vb 2023 LONDON

4 - 6 October, 2023 / London, United Kingdom

INFOSTEALERS: INVESTIGATE THE CYBERCRIME THREAT IN ITS ECOSYSTEM

Pierre Le Bourhis, Livia Tibirna & Quentin Bourgue

Sekoia.io, France

pierre.le-bourhis@sekoia.io
livia.tibirna@sekoia.io
quentin.bourgue@sekoia.io

ABSTRACT

For several years, the information stealer (commonly known as stealer) threat has increasingly impacted both individuals and businesses. This evolution can be explained, in part, by the popularization of this threat within the Russian-speaking cybercrime ecosystem and by the professionalization of its activities – two aspects that will be discussed in this presentation.

First, we will detail the key concepts related to the infostealer – a type of malware that collects sensitive data on infected devices – as well as the impact related to data theft. The stolen data is used for further theft, fraud, and large-scale attacks such as ransomware campaigns.

Secondly, we will look at infostealers within the Russian-speaking cybercrime ecosystem to explain the malware-as-a-service model, which is easy to access for actors with low technical competency. The different channels of infostealer distribution used by cybercriminals and the different marketplaces used by these actors to sell stolen data (a.k.a. logs) will be detailed.

Finally, we will present elements of our methodology for monitoring, tracking and investigating new infostealers. Based on weak signals, these methods provide a way to identify credible newcomers that will spread quickly in the wild.

CONTEXT AND IMPACT

Information stealers, commonly called infostealers or stealers, are a type of malware that collect sensitive data stored on infected machines from web browsers, web browser extensions, cryptocurrency wallets, credit cards, messaging applications and stored documents.

Over the last several years the information stealer threat has increasingly impacted both personal and corporate assets – its evolution can be explained, in part, by its popularity within the Russian-speaking cybercrime ecosystem, as well as by the professionalization of infostealer-related activities.

The data stolen by infostealers – typically credentials (passwords, session cookies, banking credentials and private keys) and sensitive documents – is leveraged by cybercriminals for fraud, data exfiltration, data theft, deployment of ransomware or large-scale cyber attacks.

In 2022, several intrusions and ransomware campaigns reported in open source originated from stolen employee credentials, which were exploited to gain access to corporate networks. Globally, the infostealer ecosystem based on the malware-as-a-service[1] (MaaS) model is part of the financially motivated cybercrime ecosystem (ransomware operators, initial access brokers (IABs), data breach actors, etc.). Ransomware groups such as Money Message [1], Royal [2] and TA505 [3] have leveraged infostealers in double extortion ransomware operations. The Vidar stealer, first reported [4] to be used in ransomware campaigns in 2019, was observed being used in campaigns deploying the LockBit [5] and Djvu [6] ransomware in late May 2023.

However, the exploitation of information gathered by infostealers can involve any type of threat actors or intrusion sets, including advanced persistent threat groups (APTs). At least one stealer sold as MaaS – Rhadamanthys – was recently reported being used by the Russia-nexus Sandworm intrusion set.

Key concepts

The following is a list of some basic concepts routinely used by infostealer operators in their daily activities:

- **Traffer** (from the Russian ‘Траффер’) – a threat actor in charge of redirecting the user’s traffic to malicious content in order to distribute information stealers.
- **Build** (from the Russian ‘билд’) – a stealer sample generated for a dedicated customer with a unique identifier, linked to the traffer’s *Telegram* account, to gather all collected logs on its *Telegram* bot.
- **Crypter** (from the Russian ‘крипт’) – a piece of software used for encryption of a malicious file for anti-virus evasion.
- **Installs** (from the Russian ‘инсталлы’) – a method to get logs by persuading the user to download a malicious file.
- **Logs parsing** (from the Russian ‘чек логов’) – the process of verifying logs to identify and sort logs of interest based on a given query. This can be done manually or automatically, with open-source, paid or custom-made software.
- **Logs processing** (from the Russian ‘отработка логов’) – the process of analysing logs, also used to refer to log exploitation.
- **Knock time** (from the Russian ‘отстук’) – a term used by traffers and traffers team administrators to qualify the rapidity with which a log is received by a traffer after a build is distributed. This term is used to stress the successful execution rate of a piece of malware.
- **911** – a term used to designate an infection chain consisting of taking over a *YouTube* channel.

[1] MaaS is a model of delivering malware by renting it to multiple threat actors on a subscription basis

RUSSIAN-SPEAKING INFOSTEALERS ECOSYSTEM

Professionalization of the Russian-speaking infostealers ecosystem

Malware-as-a-service

During the last few years, the infostealer threat has become increasingly prominent as the part of the Russian-speaking cybercriminal ecosystem related to infostealer distribution has become highly specialized and opened its doors to less skilled, and consequently more numerous, threat actors.

Most of the infostealers available on Russian-speaking hacking forums are sold using the malware-as-a-service model, allowing cybercriminals to purchase a subscription for use of the malware. As a result, numerous threat actors distribute stealers at a lower cost and with little technical expertise. Every month, we observe new infostealers sold as MaaS joining the cybercrime market, rapidly gaining popularity among threat actors.

When they subscribe to this model, the buyer gets access to an infostealer build, as well as to the stealer administration panel. Finally, the MaaS seller opens a direct channel with the client to provide support. This model features two types of actors, the vendors and the buyers. The vendors operate different activities, such as developing the infostealer and the administration panel; they are also in charge of advertising and selling their packaging. The buyers are also cybercriminals, whose subscription grants them access to the infostealer for a limited time period; these actors are in charge of distributing the infostealer.

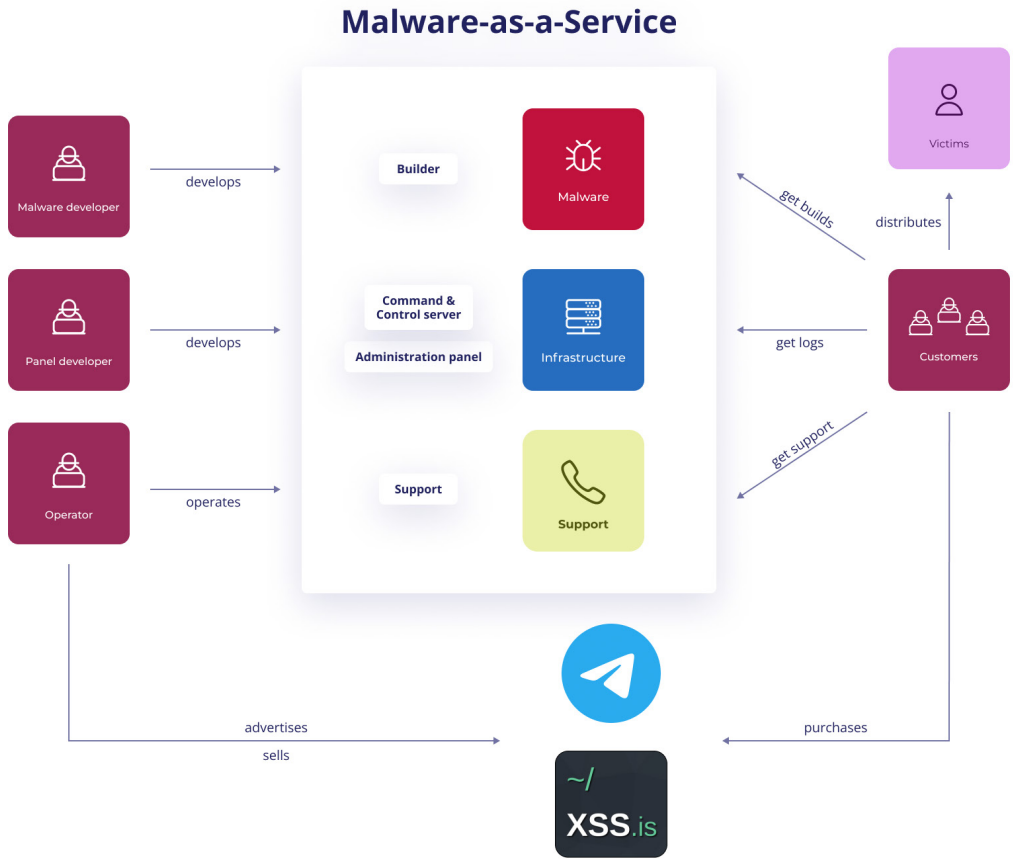

Figure 1: The malware-as-a-service (MaaS) model adapted to the infostealer threat.

The MaaS model is popular in the cybercrime ecosystem mainly because of its low entry cost: an average subscription for an infostealer ranges from US$50 per month to US$200-300 per month. The other reason for its popularity is the low technical requirement for buyers – customers do not need to have advanced skills in malware development in order to operate an information stealer campaign.

The size of the threat group behind a MaaS program depends both on its levels of sophistication and professionalization and on its goals. Generally, threat actors distributing stealers-as-a-service are organized in small teams of up to three people, with an administrator running the business and several threat actors to assist them in advertising, developing or deploying the MaaS model.

Communication and support

Infostealer developers need to promote and advertise their product for wider distribution, which is why cybercrime forums and *Telegram* channels have an important place within the ecosystem. Most emerging and established stealers are advertised in a dedicated 'commercial' thread on forums, sometimes accompanied by a second thread dedicated to collect feedback, as well as in public and private *Telegram* channels, groups and bots.

Malware developers and customers leverage these cybercrime forums to ensure the continuity of their projects, as they allow good visibility and ability to capitalize on clients' feedback. The forums usually implement features to enhance trust among forum members, including malware developers and customers. One of these features is the deposit mechanism – to demonstrate their reliability and their professionalism, malware developers make a cryptocurrency deposit.

Forums also have dedicated sections for malware advertisement, which stealer developers use to promote their products and/or related services and to establish a first contact with their customers. The advertising posts generally look the same for most infostealers sold as malware-as-a-service: the post starts with a short description of the malware, its core functionalities (for instance which language is used for the development, the build size, the protocol used for the communication with the command-and-control, etc.). Developers also briefly list the techniques they used to bypass anti-virus products or reduce the detection rate of the malware. After the description of the technical implementation, a list of targeted applications is provided, as well as a list of the data that the stealer is able to collect. Advertisement posts frequently describe access to the logs and how they are parsed and formatted. Finally, the post ends with the pricing model and often details on how to establish a further private channel of communication with the seller.

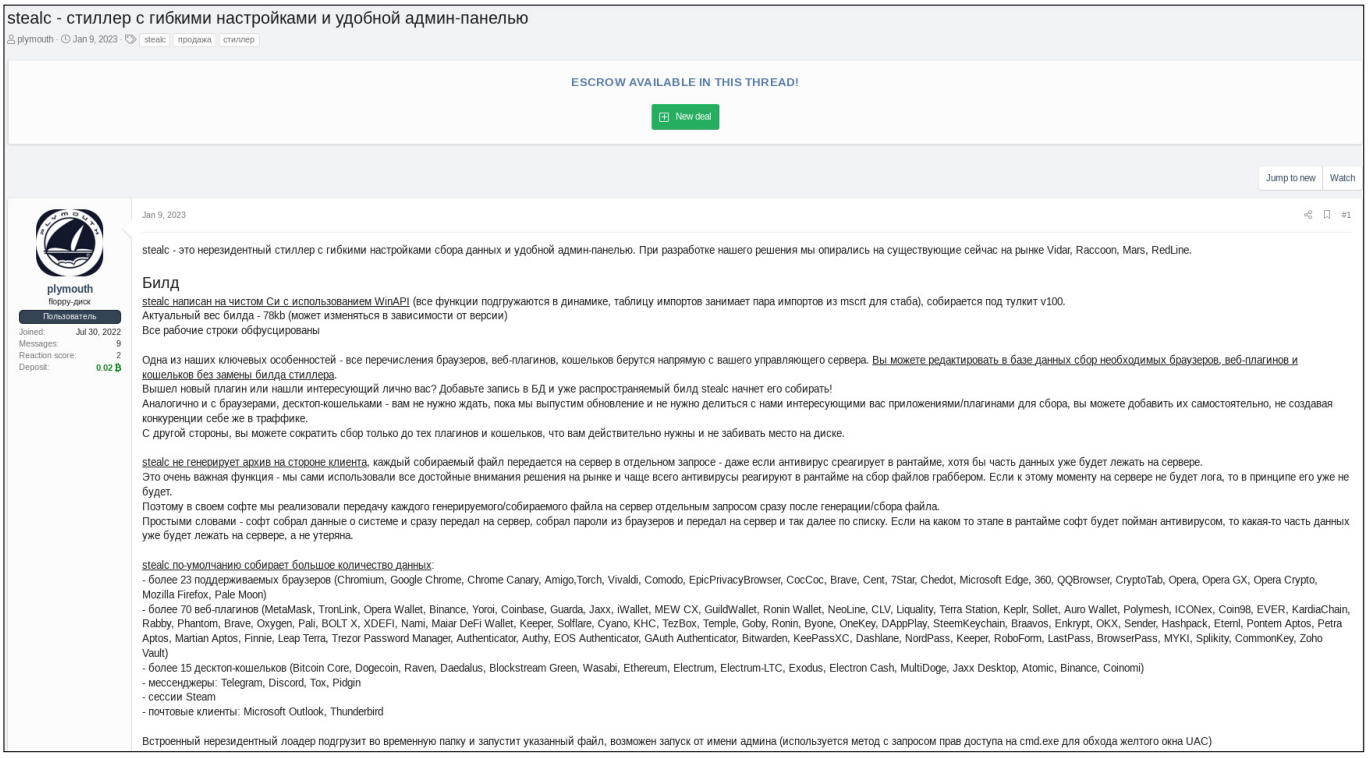

Figure 2: Advertisement for the Stealc MaaS on the XSS cybercrime forum.

Developers and other threat actors within the infostealer ecosystem frequently leverage *Telegram* for the promotion and sale of their services via dedicated channels. They use it alongside cybercrime forums for advertisement. Further, *Telegram* channels are used for posting changelogs to announce new features and/or bug fixes. The public channels are also used to offer discounts to potential or undecided customers.

Dynamics of the infostealer-as-a-service ecosystem

As mentioned previously, new information stealers regularly appear in the cybercrime ecosystem, resulting in a highly heterogeneous ecosystem. While malware such as Amadey, Redline, Vidar or even Raccoon are still very popular among members of cybercrime communities, the market attracts new and varied actors. This is emphasized by the proliferation of emerging stealers (e.g. RisePro, Eternity, Typhon Reborn, AcridRain, WhiteSnake, Stealc), and also by the threat actors switching activities – for instance, Aurora was historically known for its botnet activity but since summer 2022 its developer has started developing and selling infostealers.

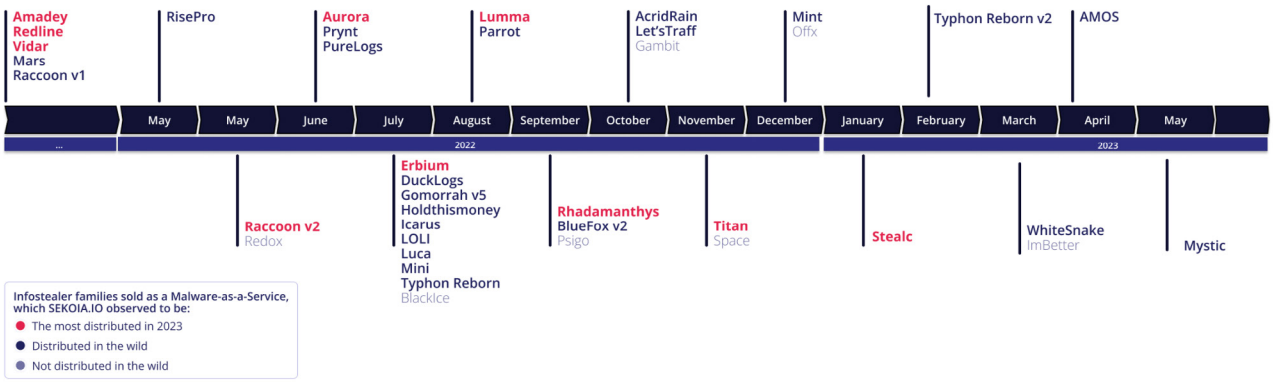

Figure 3: Timeline of the most prominent infostealers sold as MaaS on cybercrime platforms.

Emergence of traffers teams for mass distribution

Traffers play a key role in the mass distribution of infostealers, leading to the gathering of users' sensitive information and its further exploitation.

As part of the growing trend of professionalization within the cybercrime ecosystem, the organization of traffers' activity is marked by the emergence of traffers teams: a number of traffers form a team to distribute infostealers on behalf of the team administrator(s).

According to our observations, traffers teams distributing infostealers mainly recruit and operate on Russian-speaking cybercrime forums such as Lolz Guru, BHF, XSS, etc. Based on analysis of over 100 advertisements on Lolz Guru and BHF forums aiming at recruiting traffers to distribute stealers, we identified a common modus operandi.

A traffers team is an organized with a centrally managed structure headed by one or several team administrators. Team administrators hire traffers, who are in charge of generating traffic to distribute stealer builds, and provide them with resources including automatically generated builds, crypter service(s), manuals and guidelines, search engine optimization (SEO) services, a working *Telegram* infrastructure (channel, groups and bots), and dedicated services for parsing, processing, exploiting and selling logs.

Once a traffer gets a build, they are responsible for spreading the malware using the team's delivery methods or their own infection chain. Team administrators prompt recruited members to distribute the builds widely and reward them based on their performance in collecting information from victims. *Sekoia.io* assesses that the interest of traffers team administrators in hiring a large number of traffers lies in having access to large volumes of fresh data when cumulating all the submissions.

Figure 4 is an overview of a typical traffers team and its interactions within the cybercrime ecosystem:

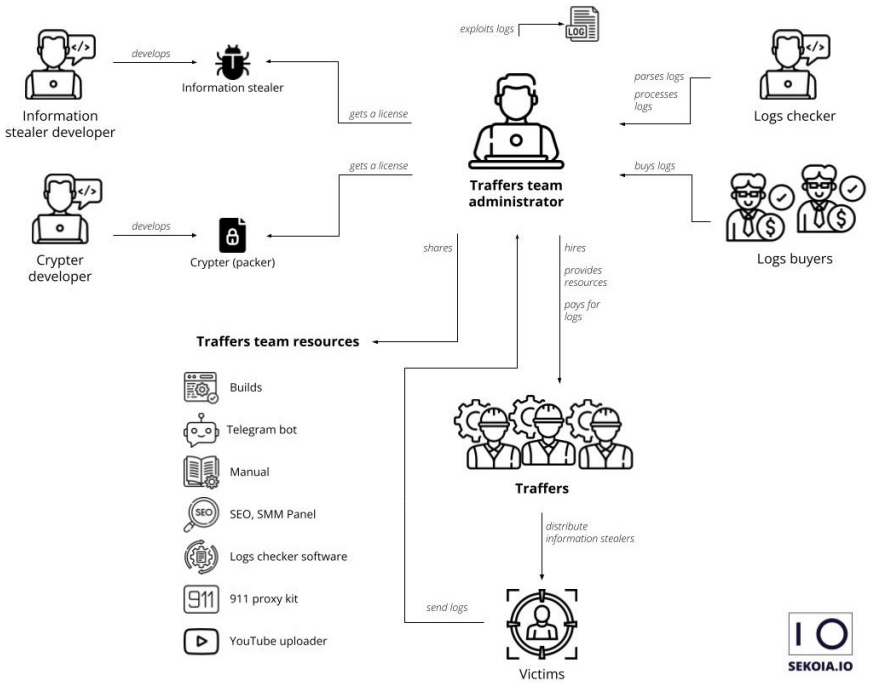

Figure 4: An overview of a typical traffers team structure and its interactions.

Distribution channels

The lures used by these actors to distribute infostealers widely are mainly cracked software, web pages imitating legitimate software sites, fake updates, and documents related to corporate activities. Techniques to redirect potential victims to these resources are usually SEO poisoning, malvertising, email spam, and redirection from community forums. Some cybercriminals offer their services to other threat actors to set up these distribution channels.

One of the most abused infection chains is the 911 chain: an attacker publishes a video on *YouTube* that explains how to install cracked software, and provides a link to a distribution website in the description of the video. Generally, the link redirects to a file transfer service (e.g. *MediaFire*, *transfer.sh*, *Google Drive*). To deter anti-virus detection, the downloaded file is an archive protected with a password, which is provided in the video description. Once the victim downloads and unzips the file, the threat actor will execute the stealer, which mimics legitimate software.

In this scenario, the video is posted either by a fake account specifically created for the campaign or via a compromised account whose credentials were bought on a cybercrime marketplace or from a previous stealer campaign. The advantage of such a campaign is the ease of publication and its low technical requirement.

Another widespread technique is SEO poisoning: attackers abuse the search engine optimization algorithm to promote a website controlled by the attacker. The website often contains a catalogue of fake or cracked software. Unwarned users are tricked and click on the first link displayed by the search engine. Once the user is on the website, they are fooled and download an archive protected with a password; the rest of the infection chain is the same as the 911 chain. This technique requires advanced knowledge on how to perform SEO poisoning; furthermore, attackers need to host their website to expose cracked software.

The third infection chain that is commonly used by attackers is malvertising. This chain uses *Google Ads* to spread a copycat of a legitimate website (e.g. video conferencing software or VPN client) to deliver their malware.

As a matter of course, diversity and creativity in the infection chains bring new roles into the cybercrime ecosystem. The monitoring of *Telegram* channels and cybercrime forums proves that copies of legitimate software websites are for sale, as well as documentation and tutorials on how to optimize for search engines so that the malicious websites appear in the top search results. These new types of service highlight an expansion and a professionalization of the cybercrime ecosystem.

Processing and sale of stolen data

The professionalization of the infostealer-related ecosystem is also reflected in the emergence of various services associated with the infostealer distribution and the processing and sale of stolen data, commonly known as ‘logs’. The logs selling phase is a key component in understanding the dynamics of the infostealer threat.

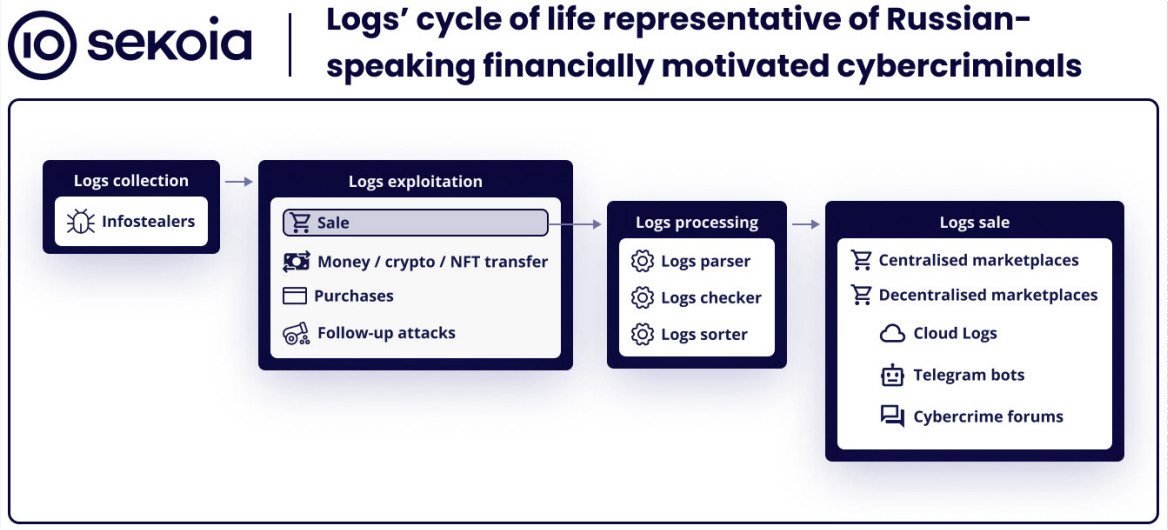

Figure 5: Life cycle of logs representative of Russian-speaking financially motivated cybercriminals.

The quality, recency and accuracy of the stolen data sold on marketplaces is very variable. This likely stems from the fact that there are a high number of logs sold on cybercrime platforms, whose format differs widely depending on the infostealer used. Moreover, manipulating databases of thousands or even hundreds of thousands of pieces of raw data and files is complex.

Multiple tools for parsing, checking and sorting logs have emerged within the Russian-speaking cybercriminal ecosystem to facilitate the qualification and exploitation of stolen data for threat actors. The tools or services on offer mainly aim at:

- Parsing logs generated by multiple stealers to normalize their format.
- Filtering logs based on domain patterns or keywords pertaining to specific topics (cryptocurrency, bank, social networks, casino, video games).
- Checking the validity of authentication information (cookies or login credentials).
- Checking for duplicates with other log databases.
- Sorting log files, folders.
- Clearing log files based on their extension.

Well-known log parsers advertised on cybercrime forums or *Telegram* channels include Crystal, BLTools, Paranoid Checker and Profit Maker.

Over the last year, *Sekoia.io* has observed the emergence of tools with more advanced capabilities, designed to facilitate, optimize and enrich the exploitation of stolen data, including:

- Checking online account balances.
- Checking the number of followers on social media.
- Checking any subscription on accounts.
- Searching seed phrases for cryptocurrency wallets.
- Checking for wallet balances.
- Verifying 2FA activation.
- Checking for valuable items in video game accounts.

The further sale of stolen information on marketplaces, whether centralized like Genesis Market (seized on 4 March 2023) or decentralized like Cloud Logs on *Telegram*, allows cybercriminals specializing in initial access to purchase qualified stolen data to conduct more sophisticated attacks.

IDENTIFY EMERGING INFOSTEALERS

In this section, we provide details of our methodology to identify and track emerging infostealers and illustrate it with a concrete application of the methodology in the case of the Stealc information stealer. The methodology is split into three parts. The first section will expose how to monitor diverse sources to spot emerging stealers. Then, we describe how to identify potential emerging stealers using a tracking approach. The final section of the methodology explains how to assign and document a threat identified in the monitoring or the tracking phases.

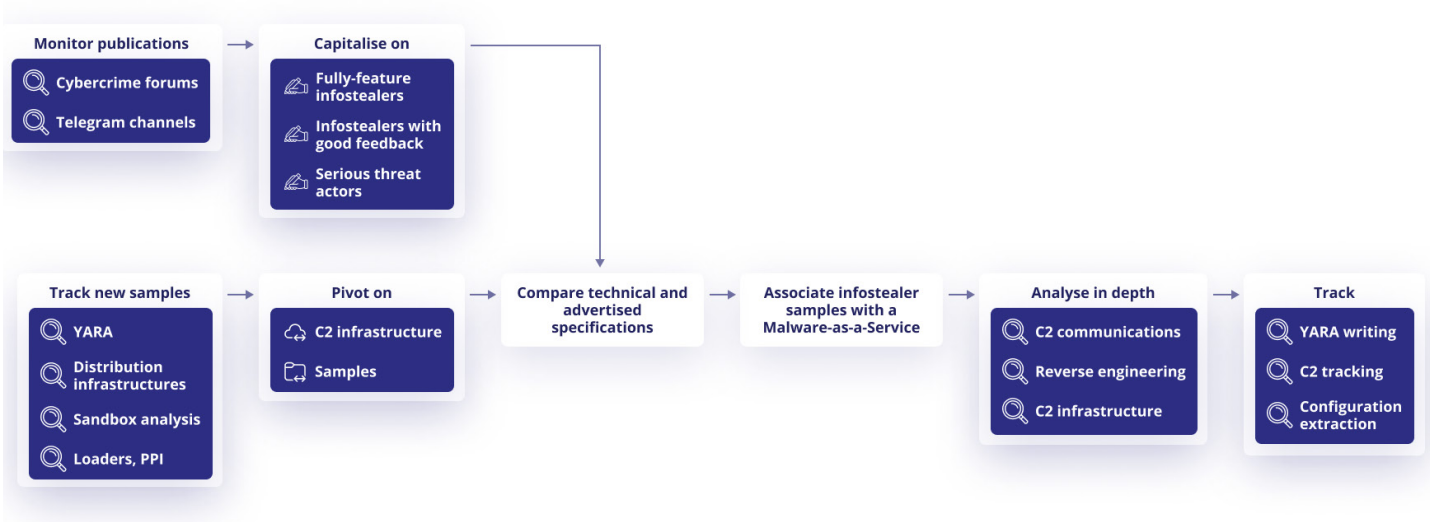

Figure 6: Our methodology for identifying and tracking emerging infostealers.

Monitoring of the advertising postings and sales of infostealers

In order to produce actionable threat intelligence, we need to identify and analyse the emerging threats that are likely to become widespread, to eventually track them down and detect them. For this purpose, we regularly monitor new MaaS

offered for sale in the Russian-speaking ecosystem. To promptly identify new infostealers sold as MaaS, we closely monitor postings of sales on cybercrime forums and dedicated *Telegram* channels and groups. Alongside monitoring sales listings, we also focus on profiling the threat actors, the activities of the alleged infostealer developers, their interactions and their selling process, and assess their credibility within the cybercrime communities. The interest (or lack thereof) for an emerging threat shown by other threat actors gives a good indication of the likelihood of it becoming widespread.

In addition, by correlating both contextual and technical analysis, we can associate an infostealer sold on the Dark Web with a malware family distributed in the wild. This pairing is used to trace the evolution of infostealer families and to focus on emerging families.

Tracking of new infostealer samples

Simultaneously, we track new samples distributed in the wild that are unattributed to known stealer families. Technical analysis of their malicious codes and the associated command-and-control (C2) infrastructures can be used to assess whether an infostealer family is widely distributed. Indeed, heuristics used to identify the C2 servers associated with a piece of malware provide an estimation of its popularity within the cybercriminal ecosystem based on the number of active servers.

There are different ways to track new stealer samples, the four following are good starting points for investigating new threats:

- Tracking the distribution infrastructure used to deliver the infostealer.
- Writing YARA rules for hunting purposes.
- Following sandbox results.
- Tracking payloads distributed by popular loaders.

These four techniques will be detailed in the next sections, as well as how they were used in the Stealc context.

Proactive research

To identify new threats, the tracking and monitoring of different types of assets is mandatory. In the case of Stealc, it was the tracking of the ‘landing page template’ used by threat actors to disguise their infostealer as cracked software that allowed its detection.

The technique of using cracked software catalogues to deliver infostealers is recurrent in the cybercrime landscape. By tracking these templates, it is possible to monitor the most commonly distributed stealers (using this technique of distribution) and also to identify new stealers. This process consists of regularly querying different search engines (e.g. *Censys*, *urlscan*) with a pattern computed specifically to track the catalogue template.

YARA rules

Information stealers share common functionalities, for instance the theft of browser passwords or cookies. These functionalities often use specific strings which can be signed in a YARA rule, for example a file path to the data: ‘\\Google\\Chrome\\User Data\\’ or ‘Mozilla\\Firefox\\Profiles’. There is a long list of paths used by information stealers that can be useful for this research rule.

Information stealers also target numerous applications, such as crypto wallet applications, messaging applications (e.g. *Discord*, *Telegram*), email clients, password managers and 2FA clients. The way data is accessed can also be signed in a YARA rule.

This technique isn’t efficient on obfuscated or packed samples, and it often identifies new infostealers that are not advanced or sophisticated. The next section will explain another tracking technique that attempts to overcome the obfuscation and packing issues.

Sandbox results

The third technique used to identify and track stealers is to follow the analysis results of different sandboxes. Based on the behaviours detected by the sandboxes, it is possible to spot information stealers and moreover ones that are not signed and/or documented.

For instance, when particular behaviour is detected in the *Hatching Triage* sandbox, tags are added to the analysis (e.g. ‘stealer’, ‘spyware’). Using this method we discovered many samples of what would later be named AcridRain.

This method is complementary to the previous technique; it covers behaviours of samples at execution time.

Distribution infrastructure

The last technique used to identify new stealers relies on tracking payloads delivered by popular loaders or pay-per-install services. The advantage of this technique is the ability to determine the duration of campaigns and to have fresh information about ongoing campaigns.

For example, we tracked GCleaner, delivered by the PrivateLoader pay-per-install service. Cybercriminals pay to have their malware build delivered by this service.

GCleaner delivered different payloads based on the geolocation of the victim. The links to the delivered payload are publicly available on the command-and-control server. One proactive method to identify new campaigns and new malware is to fetch the URLs in a sandbox. Using this technique on GCleaner allowed the identification of: Cryptbot, Redline, Laplas and Stealc as distributed payloads.

Estimation of propagation from pivot results

When a new stealer is detected during the monitoring and/or the tracking phase, we need to estimate its popularity and its extent in order to determine if a deeper analysis of the threat is necessary.

To estimate the propagation of a new piece of malware there are two available pivots: the samples themselves and the infrastructure. There are many online tools and services available to identify the common properties of a piece of malware in order to illuminate its infrastructure.

We identified that the URLs used in Stealc's C2 communication share a unique pattern: the endpoint for the C2 communication is composed of 16 hexadecimal characters followed by '.php'. Furthermore, the URLs used to download legitimate DLLs (the ones used to interact with the web browser's backend) are also composed in a unique manner: there are, once again, 16 hexadecimal characters followed by the DLL filename: sqlite3.dll, mozglue.dll, freebl3.dll, msvcp140.dll, nss3.dll, softokn3.dll, vcruntime140.dll.

http && ip.addr == 37.139.128.161

Source	Destination	Protocol	Length	Info
10.127.0.91	37.139.128.161	HTTP	466	POST /984dd96064cb23d7.php HTTP/1.1
37.139.128.161	10.127.0.91	HTTP	389	HTTP/1.1 200 OK (text/html)
10.127.0.91	37.139.128.161	HTTP	520	POST /984dd96064cb23d7.php HTTP/1.1
37.139.128.161	10.127.0.91	HTTP	226	HTTP/1.1 200 OK (text/html)
10.127.0.91	37.139.128.161	HTTP	519	POST /984dd96064cb23d7.php HTTP/1.1
37.139.128.161	10.127.0.91	HTTP	1222	HTTP/1.1 200 OK (text/html)
10.127.0.91	37.139.128.161	HTTP	1037	POST /984dd96064cb23d7.php HTTP/1.1
37.139.128.161	10.127.0.91	HTTP	220	HTTP/1.1 200 OK
10.127.0.91	37.139.128.161	HTTP	144	GET /a02fc2187db8cd88/sqlite3.dll HTTP/1.1
37.139.128.161	10.127.0.91	HTTP	285	HTTP/1.1 200 OK (application/x-msdos-program)
10.127.0.91	37.139.128.161	HTTP	144	GET /a02fc2187db8cd88/freebl3.dll HTTP/1.1
37.139.128.161	10.127.0.91	HTTP	277	HTTP/1.1 200 OK (application/x-msdos-program)
10.127.0.91	37.139.128.161	HTTP	144	GET /a02fc2187db8cd88/mozglue.dll HTTP/1.1
37.139.128.161	10.127.0.91	HTTP	485	HTTP/1.1 200 OK (application/x-msdos-program)
10.127.0.91	37.139.128.161	HTTP	145	GET /a02fc2187db8cd88/msvcp140.dll HTTP/1.1
37.139.128.161	10.127.0.91	HTTP	189	HTTP/1.1 200 OK (application/x-msdos-program)
10.127.0.91	37.139.128.161	HTTP	141	GET /a02fc2187db8cd88/nss3.dll HTTP/1.1
37.139.128.161	10.127.0.91	HTTP	1175	HTTP/1.1 200 OK (application/x-msdos-program)
10.127.0.91	37.139.128.161	HTTP	145	GET /a02fc2187db8cd88/softokn3.dll HTTP/1.1
37.139.128.161	10.127.0.91	HTTP	1157	HTTP/1.1 200 OK (application/x-msdos-program)
10.127.0.91	37.139.128.161	HTTP	149	GET /a02fc2187db8cd88/vcruntime140.dll HTTP/1.1
37.139.128.161	10.127.0.91	HTTP	964	HTTP/1.1 200 OK (application/x-msdos-program)
10.127.0.91	37.139.128.161	HTTP	921	POST /984dd96064cb23d7.php HTTP/1.1
37.139.128.161	10.127.0.91	HTTP	220	HTTP/1.1 200 OK
10.127.0.91	37.139.128.161	HTTP	607	POST /984dd96064cb23d7.php HTTP/1.1
37.139.128.161	10.127.0.91	HTTP	220	HTTP/1.1 200 OK
10.127.0.91	37.139.128.161	HTTP	519	POST /984dd96064cb23d7.php HTTP/1.1
37.139.128.161	10.127.0.91	HTTP	346	HTTP/1.1 200 OK (text/html)
10.127.0.91	37.139.128.161	HTTP	517	POST /984dd96064cb23d7.php HTTP/1.1
37.139.128.161	10.127.0.91	HTTP	541	HTTP/1.1 200 OK (text/html)
10.127.0.91	37.139.128.161	HTTP	518	POST /984dd96064cb23d7.php HTTP/1.1
37.139.128.161	10.127.0.91	HTTP	220	HTTP/1.1 200 OK

Figure 7: Extraction of the HTTP communication of Stealc with Wireshark.

Results obtained using search engines can highlight an increase in active C2 servers. In the case of Stealc we used a query based on inconsistent values in the status code and the HTML content. The status code of the request indicates a status 200 and in the HTML title there is a 404. Based on this inconsistency we were able to write a query on the different search engines to identify part of Stealc's online servers. While the results obtained using this technique are not exhaustive (due to the search engines' scanning capabilities), they are useful in identifying trends.

From our query, we were able to see that one month after its commercialization, the stealer had around 40 servers online.

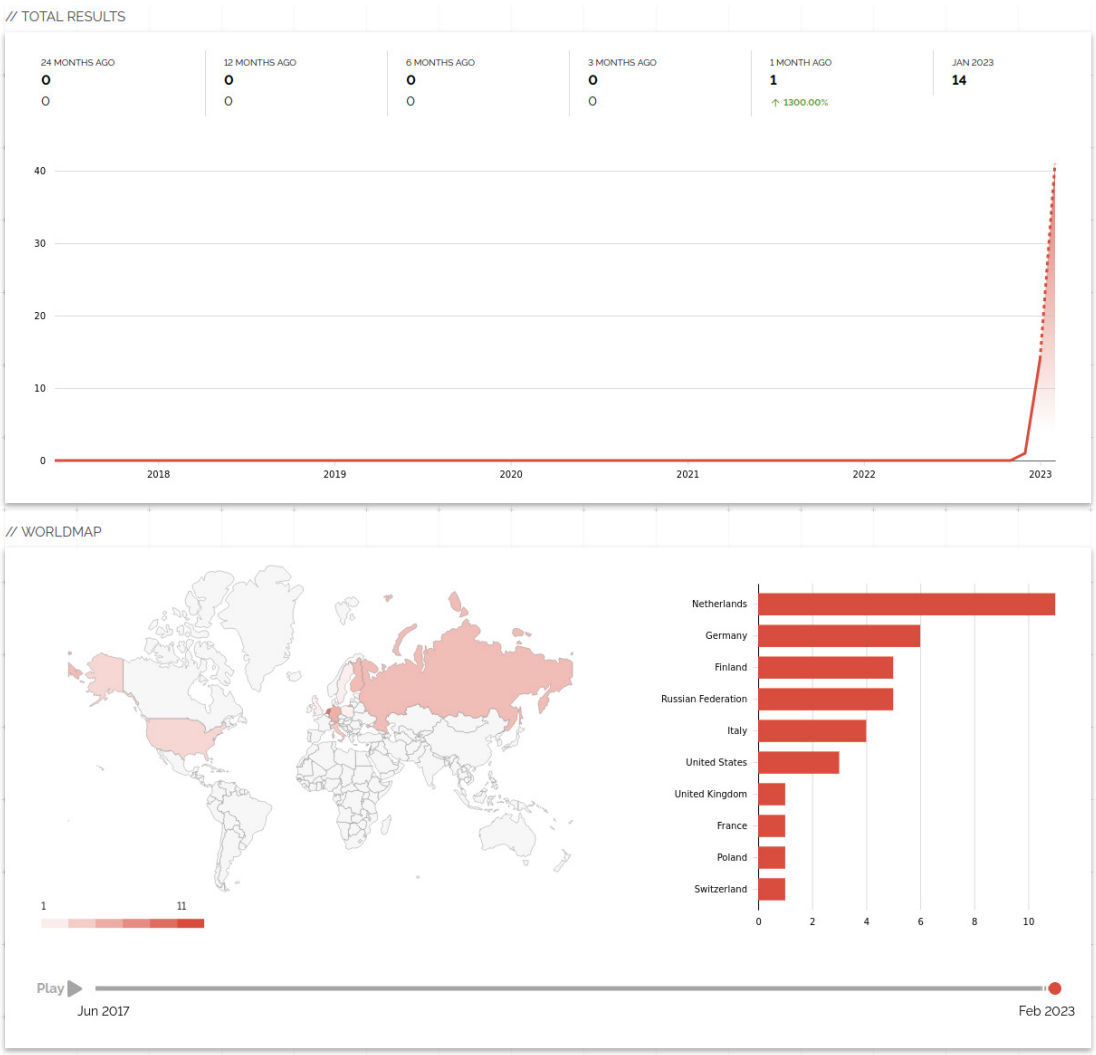

Figure 8: Shodan statistic of the query on the inconsistent HTTP status code and HTML title.

The rise in the number of observed C2 servers, together with the monitoring of cybercrime forums for mentions of Stealc, validates the hypothesis that Stealc has risen in popularity. A deeper analysis of a sample is required to improve the understanding of the threat.

However, before starting the analysis, a list of technical characteristics of the malware is required. Gathering this information is not a laborious task though, thanks to the developers having posted on cybercrime forums announcing the details of Stealc in different changelog messages.

The list of technical characteristics is used to compare the announced characteristics and those observed during reverse engineering. The objective of this deeper analysis is to validate the association between the monitored threat and the analysed payload.

In-depth analysis

In the proposed methodology, the need for an in-depth analysis arises when a piece of malware appears to become popular or widespread. The objectives of this step are:

- To compare the technical characteristics announced by the developer and those observed during the reverse engineering process.
- To provide documentation of the malware in order to improve the knowledge and coverage of the threat.
- To write advanced YARA rules.
- To identify similarities between this threat and other stealers in the ecosystem.

To set up a good environment for the analysis, a standalone sample of the malware is a plus – it avoids the noise created by loader or crypter services.

Opaque predicates

Stealc implements an anti-analysis technique named 'jump-in-the-middle', which belongs to the opaque predicate family. This technique aims to make analysis of the malware more difficult by tricking the decompiler into rendering a faulty version of the decompiled code. To work around the opaque predicate, assembly code patches are required to fix fake jumps and to get the correct version of the decompiled code.

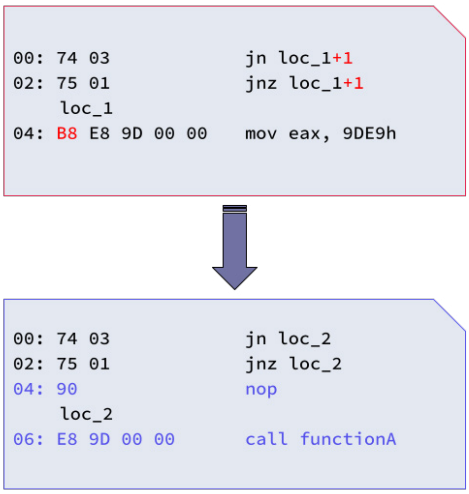

Figure 9: Example of how to patch a jump-in-the-middle opaque predicate.

Data obfuscation

The stealer stores embedded strings in Base64; these Base64 strings are encrypted using the RC4 algorithm. The deobfuscated data contains strings used for the dynamic API resolution (DLL names, API function names); it also contains strings used for data theft, as well as the command-and-control address and URLs.

```
int decrypt_string()
{
  int result; // eax

  RC4_key = (int)"74934157919546113795";
  str_04 = mw_decrypt_string("Uyk=");
  str_02 = mw_decrypt_string("Uy8=");
  str_20 = mw_decrypt_string("US0=");
  str_23 = mw_decrypt_string("US4=");
  str_GetProcAddress = mw_decrypt_string("JHgzG2hCC4cSYENcO9A=");
  str_LoadLibrary = mw_decrypt_string("L3ImL1ZECrQXdkh4");
  str_lstrcatA = (LPCSTR)mw_decrypt_string("D24zOXlMHIc=");
  str_OpenEventA = (LPCSTR)mw_decrypt_string("LG0iJV9bDagCRQ==");
  str_CreateEventA = (LPCSTR)mw_decrypt_string("IG8iKm5ILbATakV4");
```

Figure 10: Stealc strings decryption.

Dynamic API resolution

After decrypting the strings, understanding the dynamic API resolution involves re-assigning a deobfuscated string to the return value of the dynamic API resolution function. This technique is often used by malware to reduce its detection rate.

```
  lstrlenA = (int (__stdcall *)(LPCSTR))GetProcAddress(ptr_PE_header, str_lstrlenA);
  ExitProcess = (void (__stdcall __noreturn *)(UINT))GetProcAddress(ptr_PE_header, str_ExitProc
  GlobalMemoryStatusEx = (BOOL (__stdcall *)(LPMEMORYSTATUSEX))GetProcAddress(ptr_PE_header, s
  GetSystemTime = (void (__stdcall *)(LPSYSTEMTIME))GetProcAddress(ptr_PE_header, str_GetSyste
  SystemTimeToFileTime = (BOOL (__stdcall *)(const SYSTEMTIME *, LPFILETIME))GetProcAddress(
                                                                              ptr_PE_header,
                                                                              str_SystemTimeTo
}
hAdvapi32 = (HMODULE)LoadLibrary(str_advapi32_dll);
hgdi32 = (HMODULE)LoadLibrary(str_gdi32_dll);
hUser32 = (HMODULE)LoadLibrary(str_user32_dll);
hrypt32 = (HMODULE)LoadLibrary(str__rypt32_dll);
hNtdll = (HMODULE)LoadLibrary(str_ntdll_dll);
if ( hAdvapi32 )
  GetUserNameA = (BOOL (__stdcall *)(LPSTR, LPDWORD))GetProcAddress(hAdvapi32, str_GetUserNameA
if ( hgdi32 )
{
  CreateDCA = (HDC (__stdcall *)(LPCSTR, LPCSTR, LPCSTR, const DEVMODEA *))GetProcAddress(hgdi
  GetDeviceCaps = (int (__stdcall *)(HDC, int))GetProcAddress(hgdi32, str_GetDeviceCaps);
}
```

Figure 11: Stealc dynamic API resolution.

Command-and-control communication

Stealc communicates over HTTP, it uses only POST requests with HTML forms to exfiltrate data to the server and to retrieve its configuration. It uses GET requests to download additional DLLs used for the data theft (e.g. mozglue.dll to steal *Firefox* data).

To identify its victim, Stealc asks for a token during the first message exchange, which works as a handshake between the victim and the C2. The token is built from a pair consisting of the hardware ID and the build name embedded in the sample. The token is sent in each request to identify the victim.

The communication workflow is as follows:

1. C2 handshake;
2. Infected host retrieves its configuration (patterns and paths to files of interest);
3. Fingerprint of the host is sent to the C2;
4. Infected host downloads additional DLLs used to interact with the web browser's backend;
5. Stealc targets web browser data; for each type of data (cookies, passwords, credit card details) a POST request is sent to the C2;
6. Stealc targets web browser extensions; for each type of data a POST request is sent to the C2;
7. Stealc targets desktop applications, and again, for each type of data a POST request is sent to the C2;
8. A file grabber is executed, and for each match a POST request is sent to the C2;
9. Optionally, if the attacker has configured it, an additional payload is downloaded and executed. Note, at this stage, the payload is downloaded from a different server.

Targeted data

Like other stealers, Stealc targets web browser data to retrieve cookies, credit card details and passwords; it also lists web browser extensions. It targets extensions that manage crypto money assets to steal wallets or credentials to access them. Once all the data from the web browsers has been syphoned, the stealer targets desktop applications such as messaging applications *Outlook*, *Telegram* and *Discord*. It also looks for saved logins and passwords for *Steam*, and the last type of targeted applications are crypto money applications such as *Binance*. Finally, the malware has file-grabber functionality. This is increasingly popular among information stealers – it consists of a pattern (generally a regular expression) that is used to search files and directories of interest. In the context of Stealc, an attacker can configure on the C2 a list of patterns to look for on the infected host; the malware then retrieves this list during its execution from the C2.

Kill switch functionalities

Stealc has a kill switch capability: it attempts to detect its execution environment in order to avoid particular execution conditions. For example, if the local language is Russian, it stops its execution.

Other checks are executed too, to avoid virtual environments: it checks the amount of RAM of the host, it also checks if a graphics card is configured.

Stealc verifies that it is not running in the *Windows Defender* sandbox by inspecting the username and the hostname, which have specific values in the sandbox (the *Windows Defender* default username is JohnDoe and the default hostname is HAL9TH). The last check made by the malware is to verify the current date. This functionality is more related to the business model of the malware, as previously mentioned in the paper, the infostealer has a time-based subscription model.

Additional capabilities

Stealc is no exception to the stealer rule – the developer of the malware also implements functionalities shared with other infostealers. Stealc takes a screenshot of the infected host. It also has the capability to download and execute an additional payload.

CONCLUSION

The mass adoption of the MaaS model within the infostealer ecosystem is a sign of its maturity. The proliferation of the threat, observed over the last few years and consolidated over 2022, is highly likely to persist during 2023. This is partly due to a natural increase in the threat, but the growing number of infostealers sold as ready-to-use products is also evidence of a highly lucrative business model.

From our observations, campaigns involving information stealers sold as MaaS are mostly financially motivated. However, they are also regularly used in highly sophisticated attacks attributed to APTs.

The professionalization of the infostealer threat is part of a wider trend towards the professionalization of cybercrime, as recently observed across the ransomware and DDoS-related ecosystems. It allows threat actors to specialize and consequently to become more proficient, leading to faster and more effective attack campaigns on a large scale.

Threat actors related to the infostealer ecosystem are highly responsive and increasingly adaptable to evolutions within the threat landscape. They also tend to react to current events and latest trends, e.g. the use of video powered by AI to deliver infostealers by leveraging the 911 infection chain technique.

Although the exploitation of data collected by infostealers regularly leads to critical breaches and high-profile attack campaigns, the threat actors involved in the mass distribution of stealers are perceived as a form of 'low-profile cybercrime'. While this 'low-profile cybercrime' is highly specialized, it is (usually) not highly skilled. This turns some of the threat actors into 'doers' instead of perpetrators, which allows them to be less susceptible to action by law enforcement agencies.

Yet it is worth mentioning the arrest warrant issued by the US authorities against one of the Raccoon stealer operators in 2022, as well as the seizure of the Genesis marketplace in early 2023. The scrutiny from law enforcement is likely to result in threat actors tailoring their techniques to stay under the radar, while continuing to maximize their impact.

REFERENCES

[1] Cyble. Demystifying Money Message Ransomware. 6 April 2023. https://blog.cyble.com/2023/04/06/demystifying-money-message-ransomware/.

[2] Microsoft Threat Intelligence. DEV-0569 finds new ways to deliver Royal ransomware, various payloads. 17 November 2022. https://www.microsoft.com/en-us/security/blog/2022/11/17/dev-0569-finds-new-ways-to-deliver-royal-ransomware-various-payloads/.

[3] Hammond, C.; Villadsen, O. Ex-Conti and FIN7 Actors Collaborate with New Backdoor. Security Intelligence. 27 April 2023. https://securityintelligence.com/posts/ex-conti-fin7-actors-collaborate-new-backdoor/.

[4] Segura, J. Vidar and GandCrab: stealer and ransomware combo observed in the wild. Malwarebytes. 4 January 2019. https://www.malwarebytes.com/blog/news/2019/01/vidar-gandcrab-stealer-and-ransomware-combo-observed-in-the-wild.

[5] https://twitter.com/threatintel/status/1663877160497496065?s=12&t=hSMYTeIkM3tFxVAaBrKjMg.

[6] Büyükkaya, A. Polish Healthcare Industry Targeted by Vidar Infostealer Likely Linked to Djvu Ransomware. EclecticIQ. 2 May 2023. https://blog.eclecticiq.com/polish-healthcare-industry-targeted-by-vidar-infostealer-likely-linked-to-djvu-ransomware.

[7] Bourgue, Q.; Tibirna, L. Traffers: a deep dive into the information stealer ecosystem. Sekoia.io. 29 August 2022. https://blog.sekoia.io/traffers-a-deep-dive-into-the-information-stealer-ecosystem/.

[8] Bourgue, Q.; Le Bourhis, P. Stealc: a copycat of Vidar and Raccoon infostealers gaining in popularity – Part 1. Sekoia.io. 20 February 2023. https://blog.sekoia.io/stealc-a-copycat-of-vidar-and-raccoon-infostealers-gaining-in-popularity-part-1/.

[9] Bourgue, Q. Unveiling of a large resilient infrastructure distributing information stealers. Sekoia.io. 6 January 2023. https://blog.sekoia.io/unveiling-of-a-large-resilient-infrastructure-distributing-information-stealers/.

[10] Le Bourhis, P. PrivateLoader: the loader of the prevalent ruzki PPI service. Sekoia.io. 15 September 2022. https://blog.sekoia.io/privateloader-the-loader-of-the-prevalent-ruzki-ppi-service/.

[11] Bourgue, Q. Overview of the Russian-speaking infostealer ecosystem: the logs. Sekoia.io. 11 May 2023. https://blog.sekoia.io/overview-of-the-russian-speaking-infostealer-ecosystem-the-logs/.