Whitepaper

The Unknown Stealers:
From Dark Web to Log Markets

SOCRadar®
Your Eyes Beyond

socradar.io

Introduction

Stealers, initially known as InfoStealers and first seen in the late 2000s, are malware families designed to extract sensitive data from infected devices. Their main targets have been stored credentials and cookies, and more recently, cryptocurrency wallets and other parts of the crypto ecosystem.

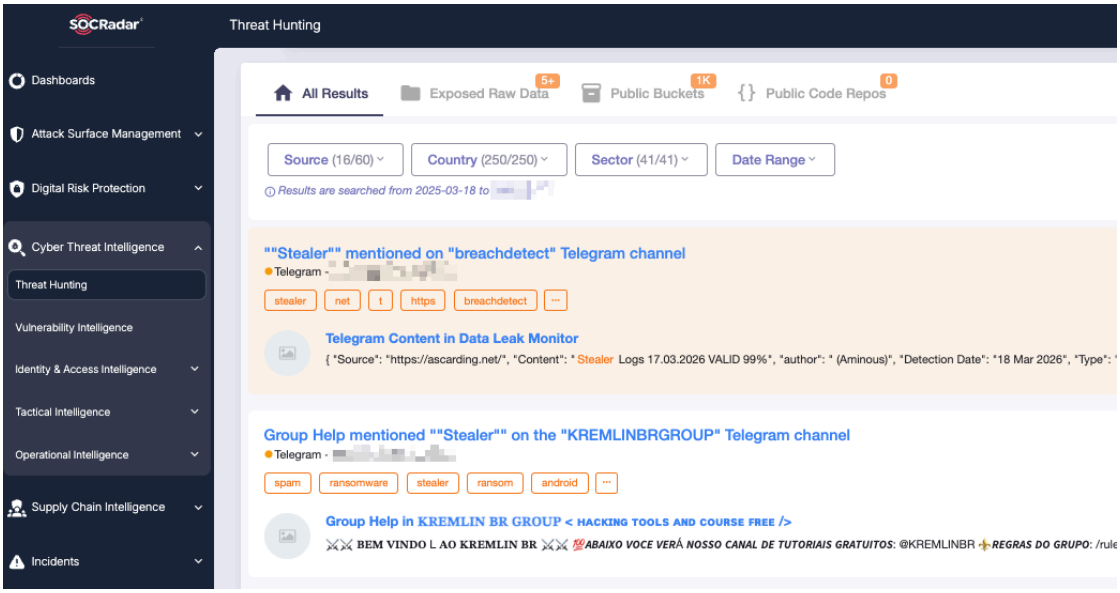
Unlike other malware, stealers operate with a low profile. They access, extract, persist, and sometimes disappear while erasing traces. They do not aim to disrupt operations or directly extort victims like ransomware. Their value lies in the data they steal and what follows after infection.

The exfiltrated data is shared in underground communities as logs, then sold on specialized marketplaces for 10 to several hundred dollars. Other threat actors (TA) buy these logs to launch more advanced attacks, including ransomware, Business Email Compromise (BEC), or Account Takeover (ATO) targeting corporate accounts.

Their relevance comes not only from volume but also from acting as an entry point for larger attacks. Their business model follows Malware-as-a-Service (MaaS).

This model allows adversaries with limited technical skills to subscribe to a stealer for 50 to 300 dollars per month. Subscribers access a web panel to configure the data to steal and build payloads using provider features.

This industrialization drives variant growth and the constant rise of new actors developing stealers.

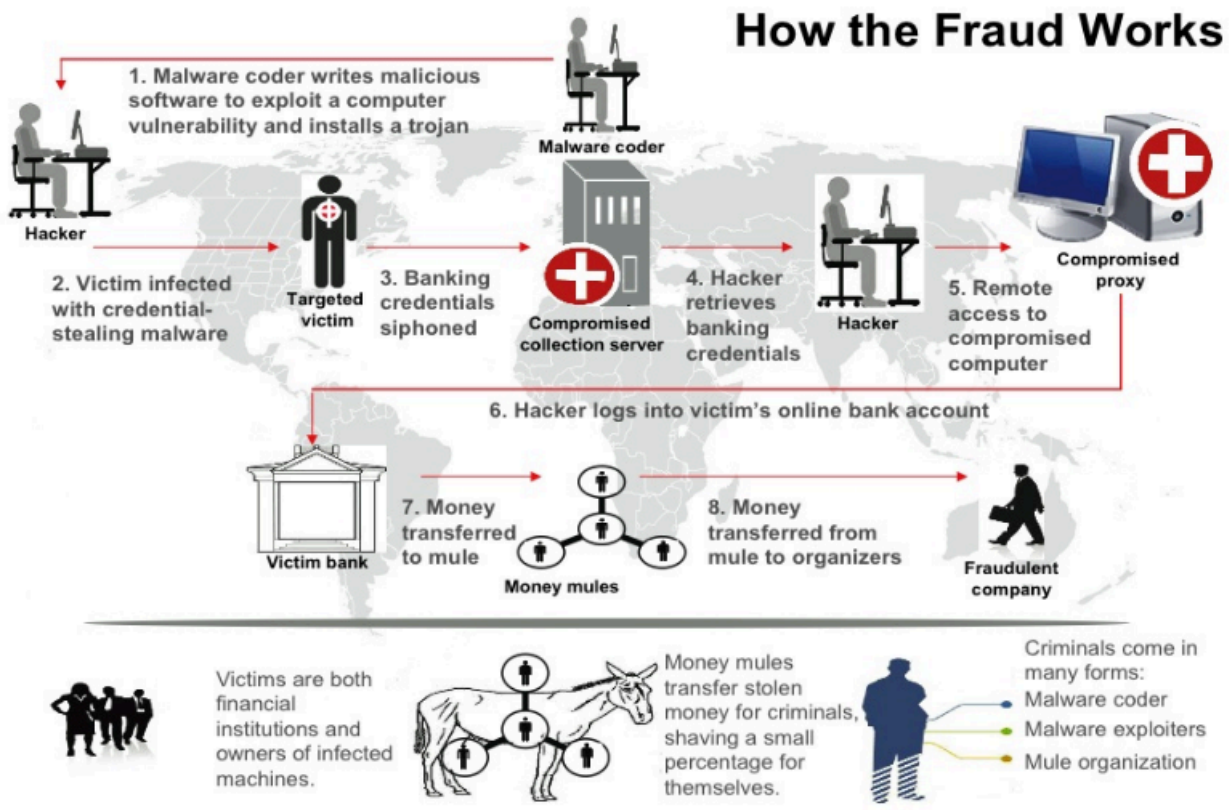


During threat hunting in the SOCRadar platform module, the Threat Research team identifies new stealers entering circulation each month that feed the MaaS market.

History & Evolution

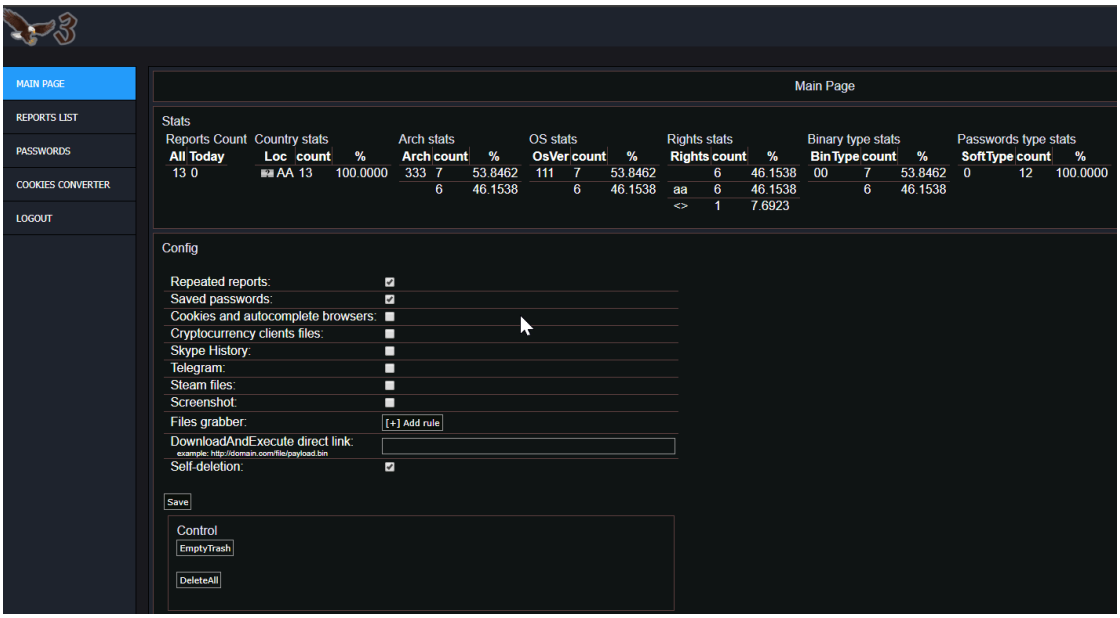
The concept of infostealers began in the late 1990s, with a focus on rudimentary methods of collecting user data and basic keyloggers that captured keystrokes. At the time, malware was often distributed through floppy disks and email attachments.

One of the most significant milestones was the emergence of **Zeus**, also known as **Zbot**, a banking malware designed to intercept banking credentials through form grabbing and screen captures. Its financial impact was considerable and redefined the malware business model. After its source code was leaked in 2011, underground markets were flooded with similar malware based on that code, acting as a catalyst for today's criminal ecosystem.



Fraud Scheme for Zeus Trojan (Source)

From 2006 to 2016, families such as SpyEye, Citadel and Dyre appeared, derived from or inspired by the Zeus model. Their primary focus remained financial, intercepting banking transactions, stealing credit card numbers and capturing online banking credentials. However, as the digital attack surface expanded with the growth of cloud services, social networks, gaming platforms and cryptocurrency wallets, the conditions were created for a shift in the objectives of stealers. AZORult was one of the early examples of this more modern era, adopting a broader approach focused on browsers, wallets and other types of sensitive information.



Example of an early AZORult-type infostealer panel used to configure data theft and manage stolen credentials (Source)

Timeline of the Stealer Ecosystem

The period from 2018 to 2026 marks the maturation of the stealer ecosystem. It shows the coexistence of smaller, artisanal tools and large-scale providers that have impacted millions of users for years.

The market has become more professional, with evolving technologies and improved techniques, alongside rapid growth in low- to mid-tier segments where new actors emerge each year.

The ecosystem timeline during this modern period reflects this consolidation and expansion.

Fragmentation Dynamics & Novel Stealers

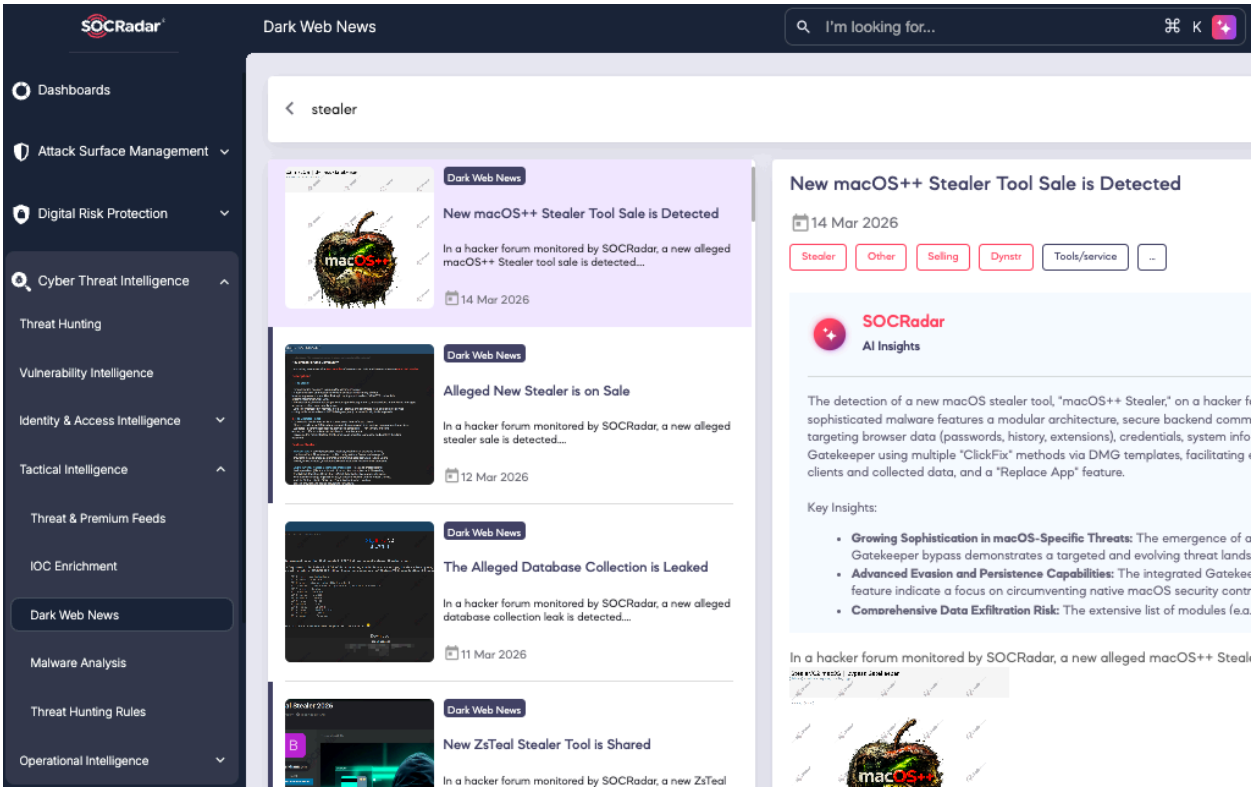
One of the main turning points in the stealer ecosystem is the high turnover and constant creation of new tools. Some malware families persist for many years, while others regroup, dissolve or rebrand annually, creating new versions that drive the growth or emergence of additional stealers.

Some examples of this phenomenon among organized groups include:

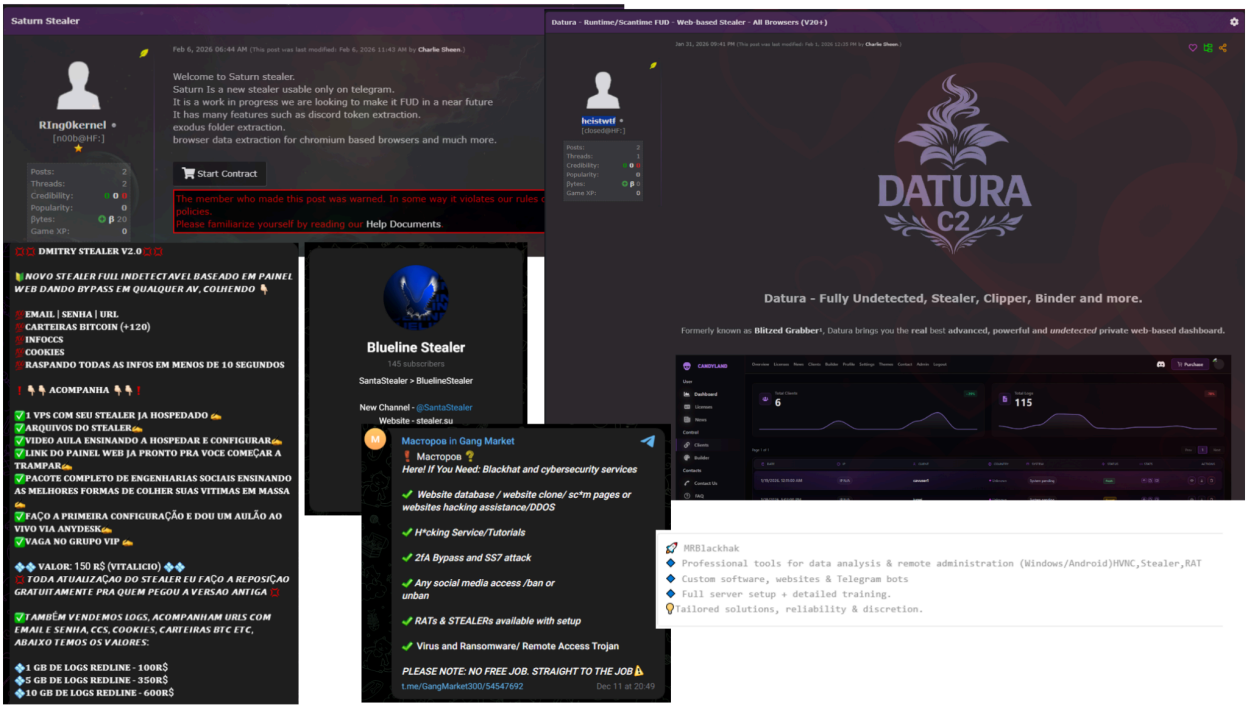
- The fall of AZORult, which accelerated the adoption and rise of **Raccoon** and **Vidar,**
- The suspension of Raccoon in 2022, which led to the adoption and increased prominence of **Lumma** and **RedLine,**
- The slowdown in Lumma's activity, which resulted in increased distribution and use of **StealC** and **Vidar.**

These dynamics are also present in smaller markets. It is common for stealers to be dismantled after a year, only for a new malware strain to emerge built on the knowledge or team members of previous projects. This makes the market difficult to track within the broader ecosystem.

The SOCRadar platform features various news items about threat actors promoting their new tools to affiliates who will be creating the campaigns that follow:



Their low public profile can result in lower detection rates and reduced visibility, increasing their success rate. Although their operational lifespan may be shorter, they can be particularly critical for companies that do not actively monitor underground markets and the emergence of new malware, especially stealers.



Threat Hunting pivots, stealer sale and promotion posts from Telegram and a hacker forum

Capabilities & Risks

Despite many stealer families and variants, most share similar capabilities and attack surfaces. Their techniques often trigger follow-on actions that support different malicious uses.

Category	Targeted Data	Attacker Follow-Up Use
Browser credentials	Usernames and passwords stored in Chrome, Firefox, Edge, Brave and others. Autofill data, browsing history, etc.	Credential stuffing, access to corporate accounts, sale on marketplaces via logs.
Cookies and sessions	Active session cookies that allow user impersonation without requiring a password or MFA.	Account Takeover (ATO), multi-factor authentication bypass, access to corporate SaaS.
Credit cards	Card numbers, expiration dates and CVV stored in browsers or web forms.	Direct financial fraud, sale in carding shops.
Crypto wallets	wallet.dat files, seed phrases, browser extensions such as MetaMask.	Direct theft of cryptocurrency funds.
2FA / MFA tokens	OTP codes, authenticator tokens, seeds from apps such as Google Authenticator.	Two-factor authentication bypass.
Messaging	Telegram sessions, Discord sessions, email client sessions.	Social engineering, identity impersonation, access to internal communications.
FTP / VPN clients	Credentials from FileZilla, WinSCP, corporate VPN clients.	Initial access to corporate infrastructure, lateral movement.
System files	Specific documents by extension (.pdf, .docx, .kdbx), configuration files.	Intellectual property theft, access to password managers, reconnaissance.
System data	Hardware details, operating system, process list, configured language, screenshots.	Environment fingerprinting for targeted attacks and initial reconnaissance.

These techniques carry clear risks. They enable the sale of sensitive data and escalation into more severe attacks. Depending on the target, compromised access can lead to broader breaches across the organization.

Threat Actors

The stealer ecosystem is best understood by analyzing the adversaries who create, maintain, and sell these tools. Unlike traditional malware, where development and distribution were handled by specific groups or individuals, the MaaS model has professionalized the stealer threat actor role, turning it into a commercial function similar to a Software-as-a-Service (SaaS) business. Marketing, customer support, product roadmaps, versioning, and affiliate or developer programs are now common.

Stealer Threat Actor Archetype

Adversaries operating within the MaaS environment vary depending on the type of tool or malware they develop. However, most professional groups and individuals share several common characteristics.

- **Geography:** Russian speaking actors predominate, particularly from Russia, Ukraine and Belarus, although diversification toward Eastern Europe, Central Asia and, more recently, English speaking actors has been observed.

- **Technical:** These developers usually possess intermediate to advanced knowledge of C or C++, C#, Assembly and reverse engineering. They are not necessarily exploit development experts, but they are highly skilled in evasion techniques, obfuscation and API hooking at the Windows level.

- **Experienced**: Many come from the cracking scene, keygen development, banking trojans or RATs. Some have documented histories in underground forums spanning more than a decade.

- **Anonymity**: They use pseudonyms or handles that they maintain for years, building reputation around that alias. They often operate from jurisdictions with limited international law enforcement cooperation.

In recent years, both low level actors and more organized groups have increasingly leveraged artificial intelligence. Campaigns have already been observed containing automatically commented code, standardized icons and indentation patterns typically generated by automated tools.

Communication & Channels

Depending on their sophistication level and the style of malware they develop, vendors adopt different communication and sales strategies. In the past, underground markets were the primary channels. Today, the widespread and almost silent migration of adversaries to Telegram is undeniable.

Telegram

Telegram has consolidated itself as the dominant communication channel for most active vendors, regardless of their sophistication level. In some cases, it is used purely for announcements, sharing proof of concept material or updates. In others, it functions as a direct gateway for communication and sales with customers.

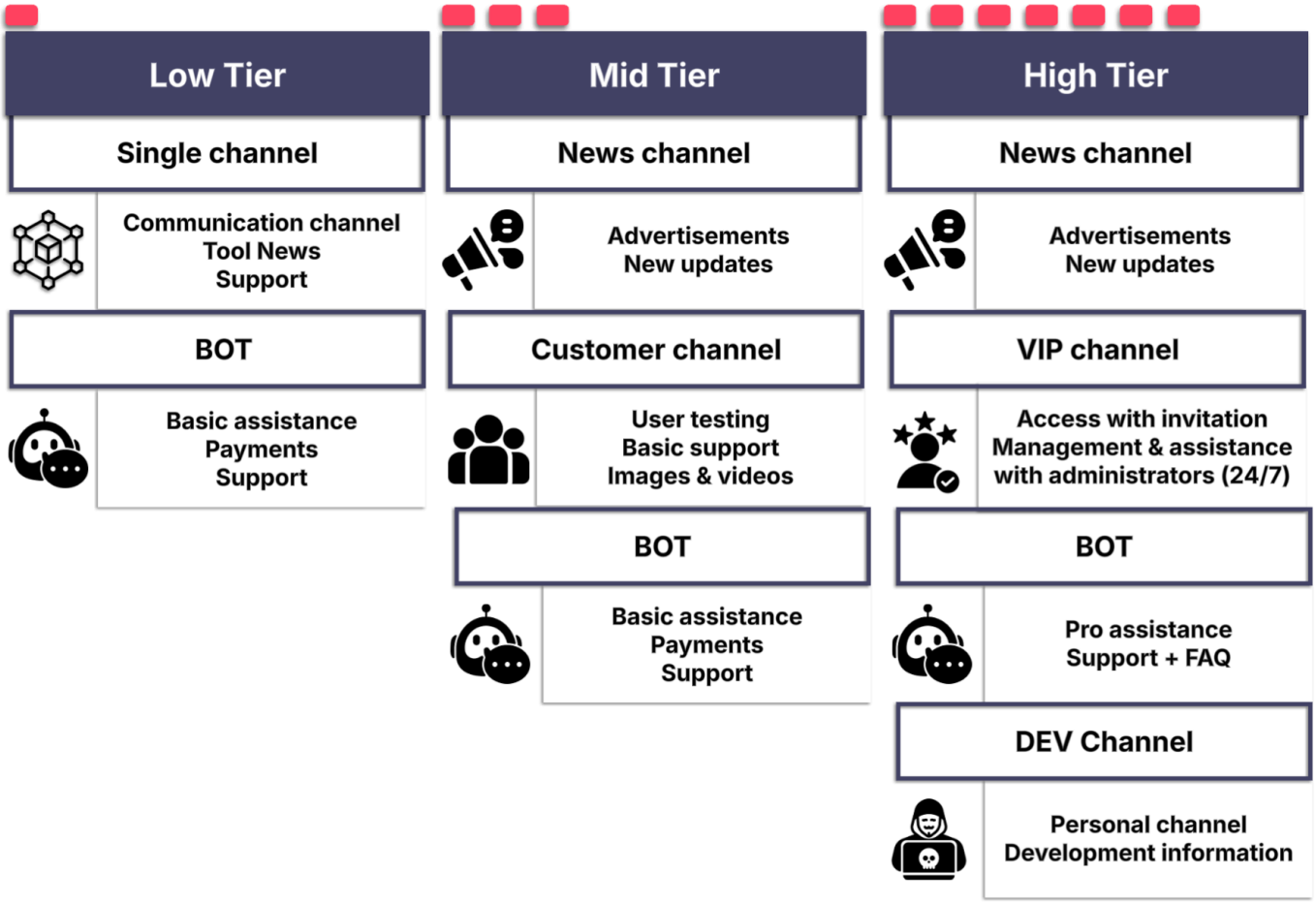
Several factors explain this shift.

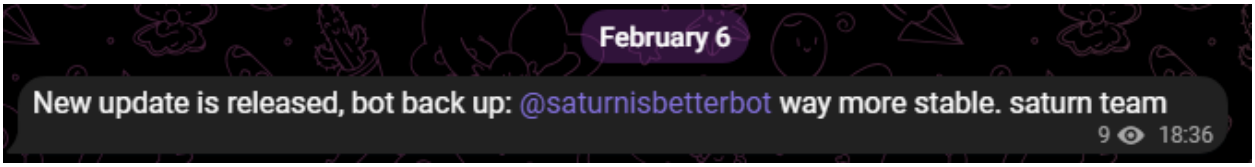
It offers secure messaging, including end to end encryption in secret chats, although most communication takes place in public channels and groups.

- Automation bots allow TAs to manage subscriptions, renewals, builder distribution and technical support without direct human interaction.
- Public channels are used for update announcements, changelogs, infrastructure incidents and ecosystem news.
- Private customer groups allow active subscribers to share experiences, report bugs and coordinate log distribution.
- Operational simplicity provides an all in one solution. Capabilities that would otherwise require custom infrastructure development or external services are integrated into a single platform suited to fraudulent markets.

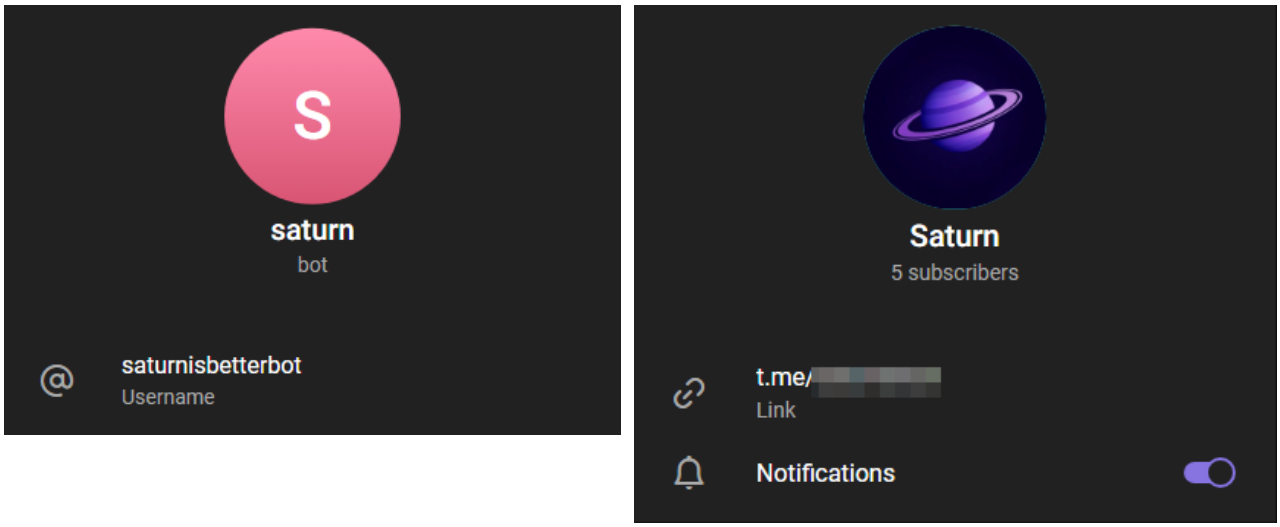
However, their structures often differ significantly depending on their maturity level.

An example of this is the low infrastructure created by Saturn Stealer, which has a channel with users, as well as a basic bot:



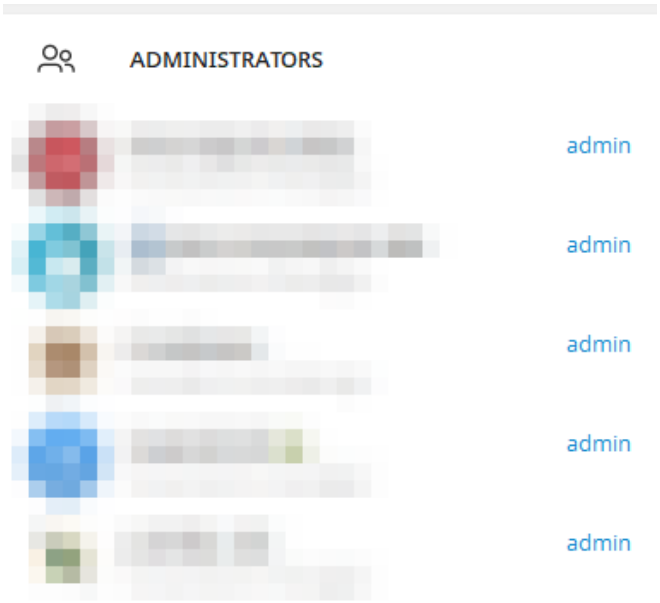


They keep their bots well-oiled with updates

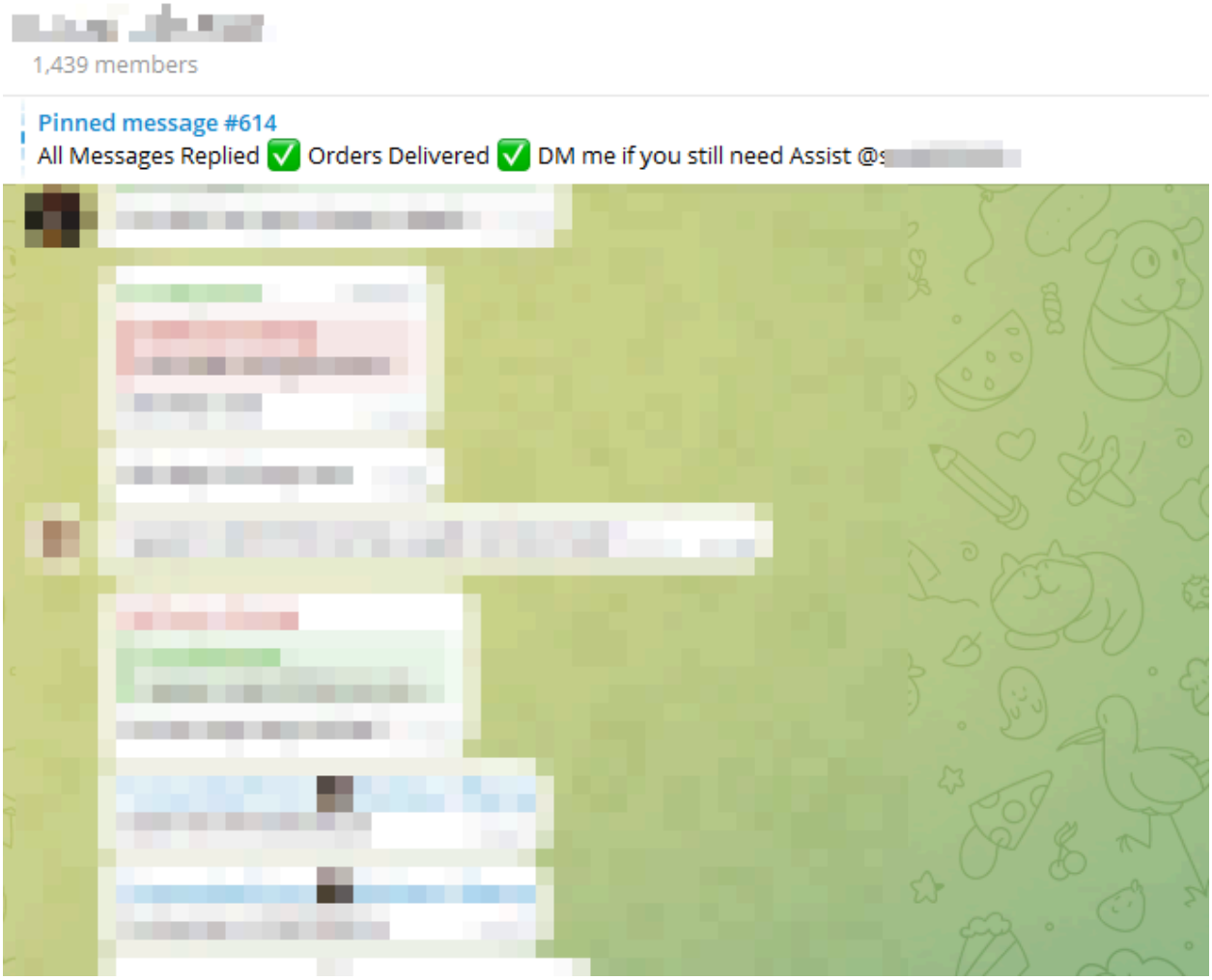


Telegram accounts of the bot and group

Completely opposite to Stealers or Malware with a consolidated infrastructure, creating several groups, with administrators, and separating them into different ones according to the purpose of each channel:



Telegram is also frequently used to showcase technical capabilities through large public groups, answer questions and support deployment via automation bots. In other cases, it functions primarily as a contact channel combined with a bot that receives stealer logs and forwards them to affiliates or creators. The platform's usage scales according to the sophistication and size of the MaaS operation.

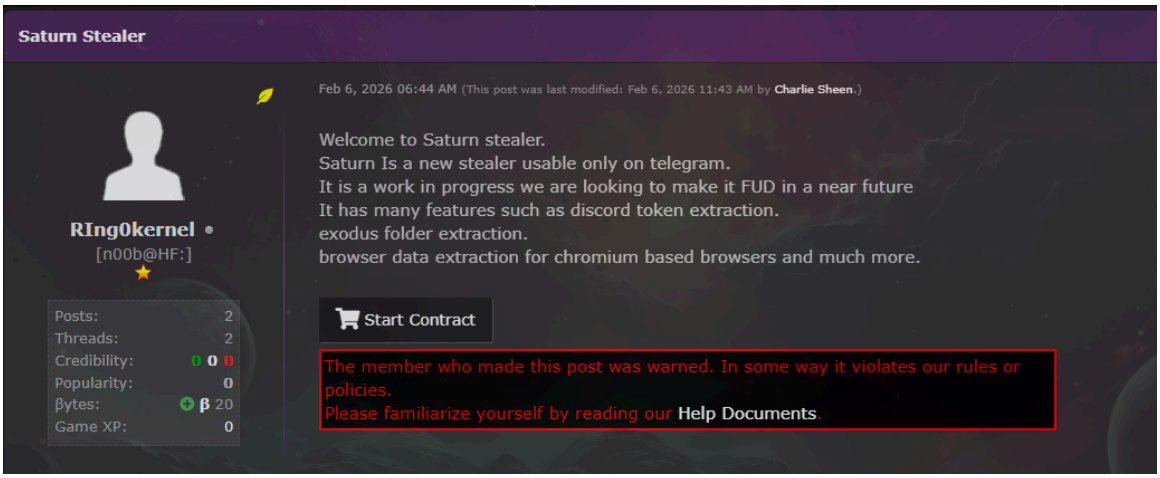


Automated channel management by the bot

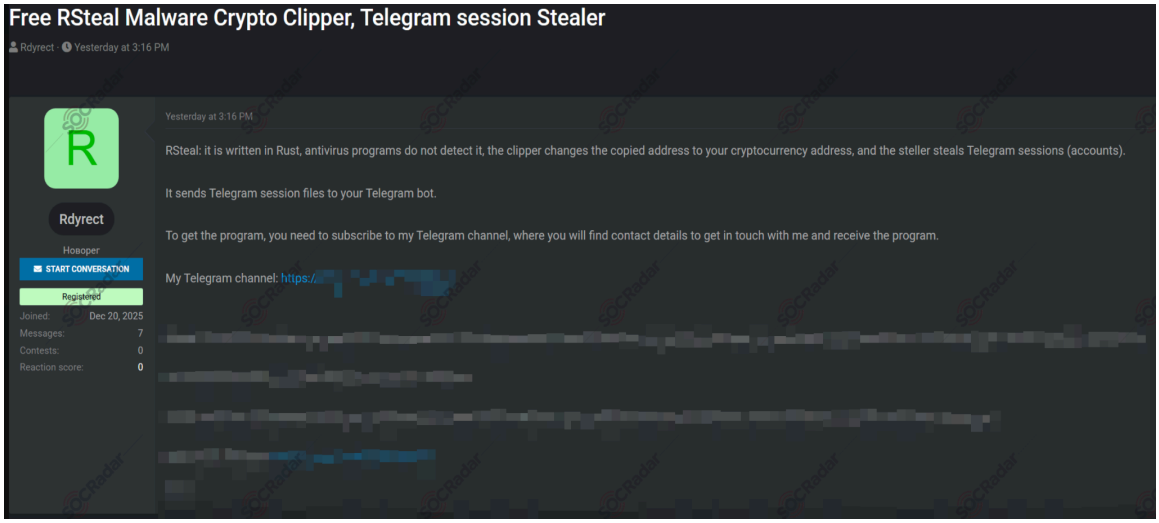
Underground Markets & Forums

As we have seen above, Telegram has displaced much of global criminal activity toward itself. However, forums remain relevant in various situations and continue to be active in the market for actions such as:

- Official launch and establishment of initial credibility, greater reputation and more potential clients,
- Sales threads including pricing, terms of service, samples, customer reviews and word of mouth,
- Dispute resolution mediated by forum administrators and business strengthening, with other developers or adversaries giving opinions about the product,
- Historical archive of the product's evolution, samples of new versions or changes.



Saturn Stealer promotion in a hacker forum



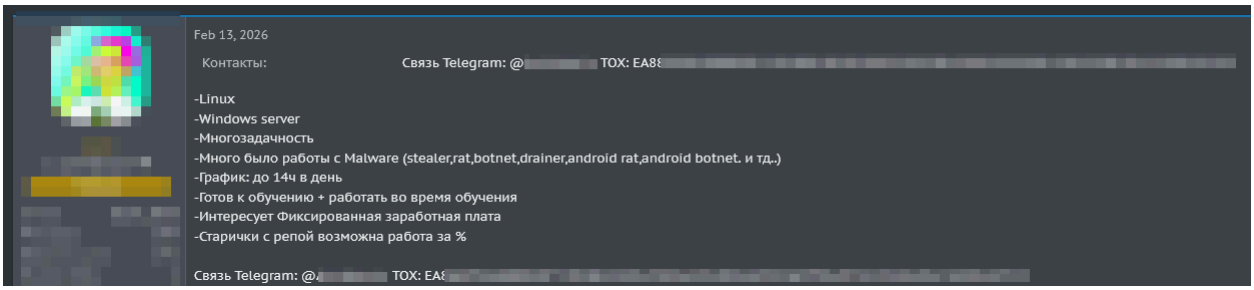
A Telegram account stealer operated in a Telegram channel that was advertised in a hacker forum

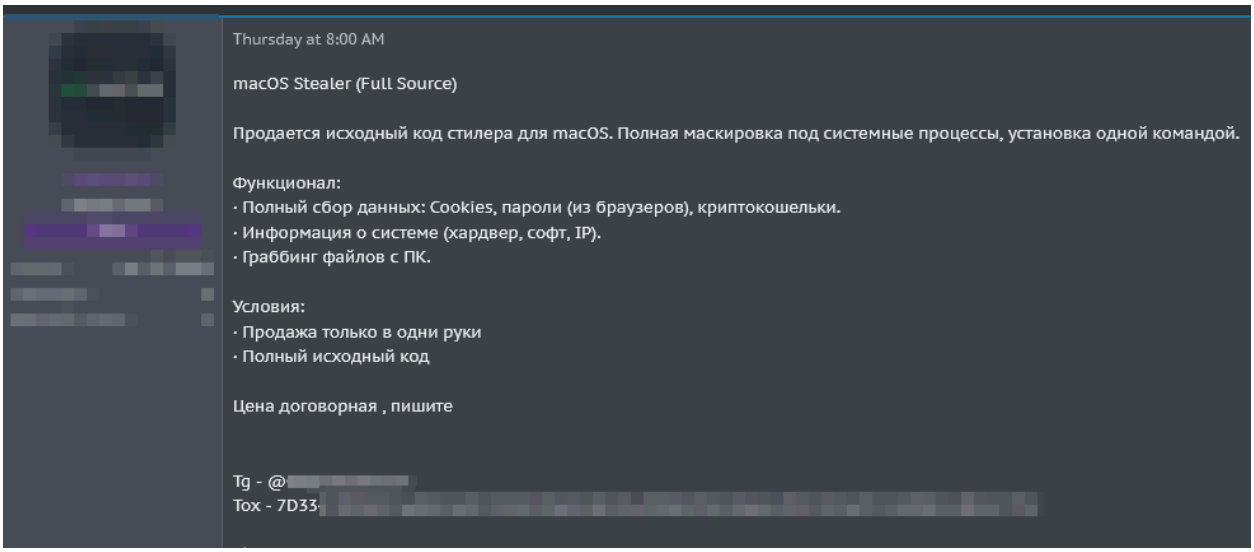
Many of these adversaries continue and will continue using these markets to redirect users to Telegram, to their accounts or even to their websites, usually accessible via TOR or through invitation or payment.

Other Platforms

In addition to the main communication methods, many adversaries use other alternatives that serve as direct substitutes for those previously mentioned:

- TOX, Jabber, Matrix: Some TAs offer support through decentralized messaging platforms for clients who require higher operational security,
- Darknet forums such as Dread or other Tor hosted forums: Marginal presence, more common among stealers oriented toward specific markets such as crypto or darknet markets,
- Discord: Largely abandoned after the increase in takedowns, although some still use it for operational support and information delivery in a similar way to Telegram bots.





Dark Web Forum posts often include Telegram links for contact

As observed, the adversaries behind the analyzed stealers are clearly centered on Telegram and secondarily on underground markets, which act as the cornerstone of their daily operations, both for communication and for publishing updates and maintaining relationships with the community of potential affiliates.

Operational Model

Unlike other types of malware where the distributor or creator may be an unknown adversary with an ephemeral presence, stealer developing threat actors operate under a long term reputation model. Although it is an unstable market and some projects are discontinued or merged with others, trust building is essential for a sustainable MaaS, because:

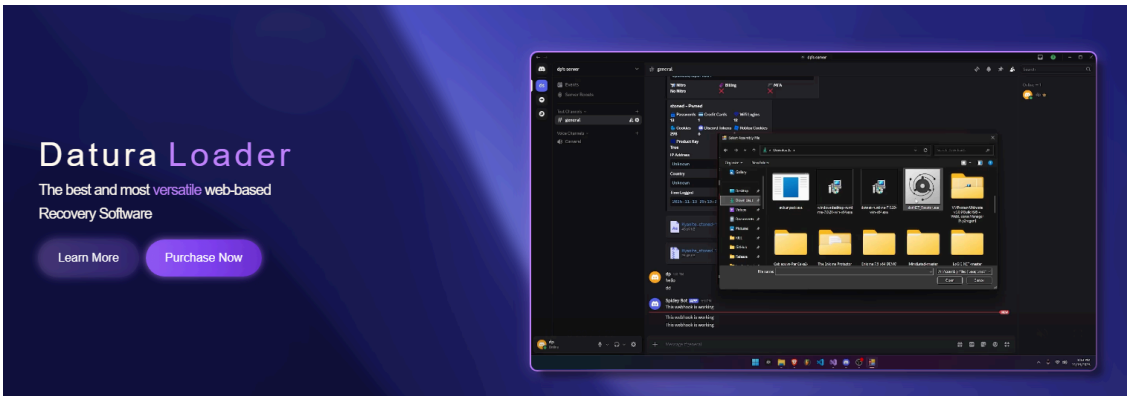
1. Clients need operational continuity. If the stealer stops working after a week, it becomes useless and trust in the developers is lost,
2. The subscription model involves recurring payments. The client or affiliate must trust that the vendor will not disappear with the money,
3. The quality of the command and control infrastructure and the builder can only be validated after the initial payment or through demonstrable proof of concepts that are often shared through official channels.

To maintain this trust building strategy, gain reputation over time and sustain a viable MaaS, adversaries carry out different activities.

Proofs of Concept (PoC) & Public Samples

Before commercial launch or after releasing the product, most TAs across all tiers publish information designed to encourage sales. These messages usually include:

- Screenshots of the panel showing real logs with sanitized data,
- Demonstration videos displaying the builder, stub configuration and infection results in a virtual machine,
- Limited samples or trial pricing lasting three to seven days for known reviewers or as launch offers.



Datura Loader PoC clip used to demonstrate capabilities and attract potential buyers

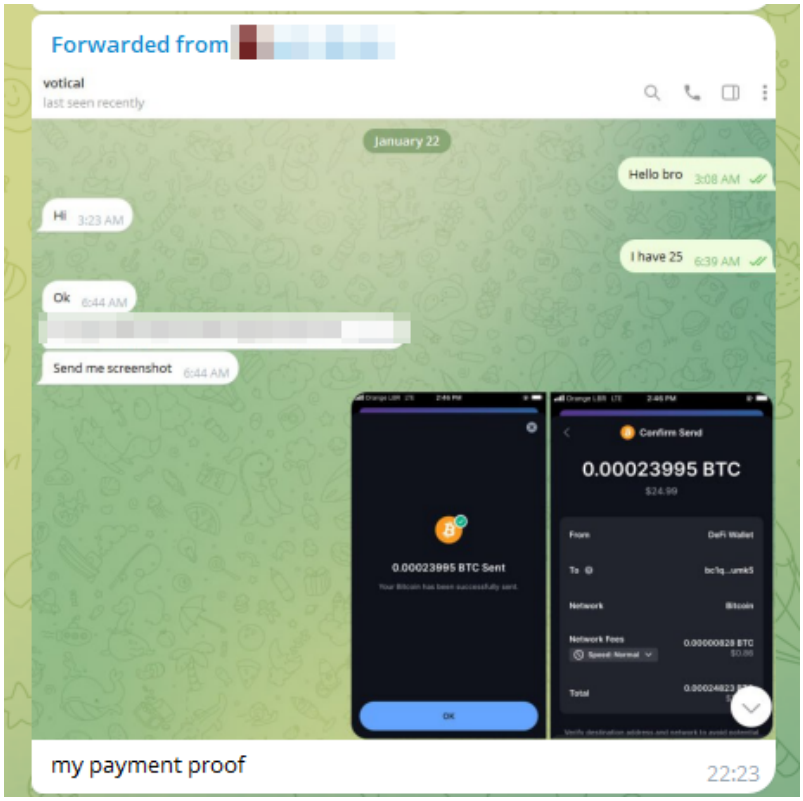
Through these strategies, users are redirected to the developer's channels or websites, if available, in order to showcase all the stealer's capabilities and promote sales.

Escrow & Guarantees

A very common practice in forums is the use of escrow systems, such as those available in forums like XSS, where initial transactions are mediated:

1. The buyer deposits funds into the forum's escrow service,
2. The Threat Actor delivers the builder and panel access,
3. The buyer has a testing period, typically 24 to 72 hours, to validate functionality and confirm that it fits their operations,
4. If everything works, the buyer releases the funds. If problems arise, the moderator mediates.

This protects both the buyer from fraudulent threat actors and the actor from buyers who make unjustified claims. To reinforce these strategies, threat actors also share images of feedback from other collaborators, including proof of payment and successful builder executions.



Payment proof in form of screenshots

The overall objective is to create the impression of a profitable business with a developer adapted to the environment and committed to keeping their word, generating guarantees that strengthen both the sales strategy and the consolidation of the product.

Customer Support

Technical support is one of the main differentiators in a market as broad as that of stealers. TAs who invest in quality support retain clients long term, can charge premium prices and create demand for higher priced packages of the tool.

The levels typically observed are:

Level	Description	Threat Actor example
Tier 1 Basic	Responses within 24-48 hours, only via Telegram, no structured documentation	Low-tier stealers under $100 per month
Tier 2 Standard	Responses in under 12 hours, documented FAQ, automated bot for common issues	RedLine, Vidar, Raccoon
Tier 3 Premium	Responses in under 4 hours, multi-language support, weekly updates, detailed changelogs, customized configuration for enterprise clients	Lumma, Rhadamanthys, Meta Stealer

This level of professionalism is often comparable to the support of a legitimate SaaS startup, demonstrating the adaptability and transformation of previously disorganized markets into a more structured strategy that results in higher sales.

Relationships for Distribution

Stealer vendors rarely distribute the malware themselves. Their role is purely technical. They develop the stealer, maintain the infrastructure including command and control servers and panels, and provide support. Distribution is usually delegated to an ecosystem of affiliates and distributors operating independently, multiplying the malware's reach without directly exposing the threat actor.

As with tiers, affiliates and the surrounding ecosystem allow several profiles to coexist simultaneously, independently from the developer.

Type	Profile	Compensation	Volume
Independent operators	Individual actors, low-effort methods such as torrents, YouTube scams, basic phishing	Standard license 100–200 per month, no revenue sharing. Monetize by selling logs directly	60-70 % of clients, 30-40 percent of infections
Professional distributors	Own infrastructure for malvertising, phishing as a service, SEO poisoning. Proven track record	Reduced license 50–100 per month plus 60/40 revenue sharing on sold logs. Customized builder and dedicated panel	10-20 % of clients, 50-60 percent of infections
Resellers	Intermediaries who buy in bulk and resell in lower-tier forums under white label branding	Resale margin, for example buy at 80 per month and sell at 120-$150 per month	5-10 % of clients, marginal impact

They may leverage the surrounding ecosystem and affiliates, delegating tasks to:

- **Malvertising Networks:** Abuse of Google Ads, typosquatting, cloaking. High volume between 10 thousand and 100 thousand infections per month, high investment.
- **Phishing as a Service Platforms**(such as EvilProxy or Caffeine): Multi factor authentication bypass combined with stealer download disguised as a mandatory update. Medium volume, high conversion.
- **Cracked Software or Torrents:** Bundled with Adobe CC, AutoCAD or AAA video games. Low cost, high execution rate around 70 to 80 percent.

Example of partnership between threat actor and a professional distributor:

Threat Actor offers:

- Customized builder with unique stub to avoid cross detection,
- Dedicated panel with exclusive subdomain and separated logs,
- Reduced fee of 75 dollars per month compared to 150 dollars standard,
- Revenue sharing 60 percent Threat Actor and 40 percent distributor on sold logs,
- Service level agreement with 99 percent uptime and priority support under 2 hours response time.

Distributor is responsible for:

- Campaign execution including malvertising, phishing as a service and search engine poisoning,
- Maintenance of burned domains and cloaking infrastructure,
- Sale of logs on marketplaces such as Russian Market or 2easy,
- Filtering of high quality logs including corporate VPNs or wallets with balance.

Mutual benefit:

- **Threat Actor:** Passive income from 60 % of sales plus fixed subscription fee without managing distribution or increasing operational exposure.
- **Distributor:** Access to proven technical infrastructure without developing a stealer from scratch, allowing full focus on distribution and monetization.

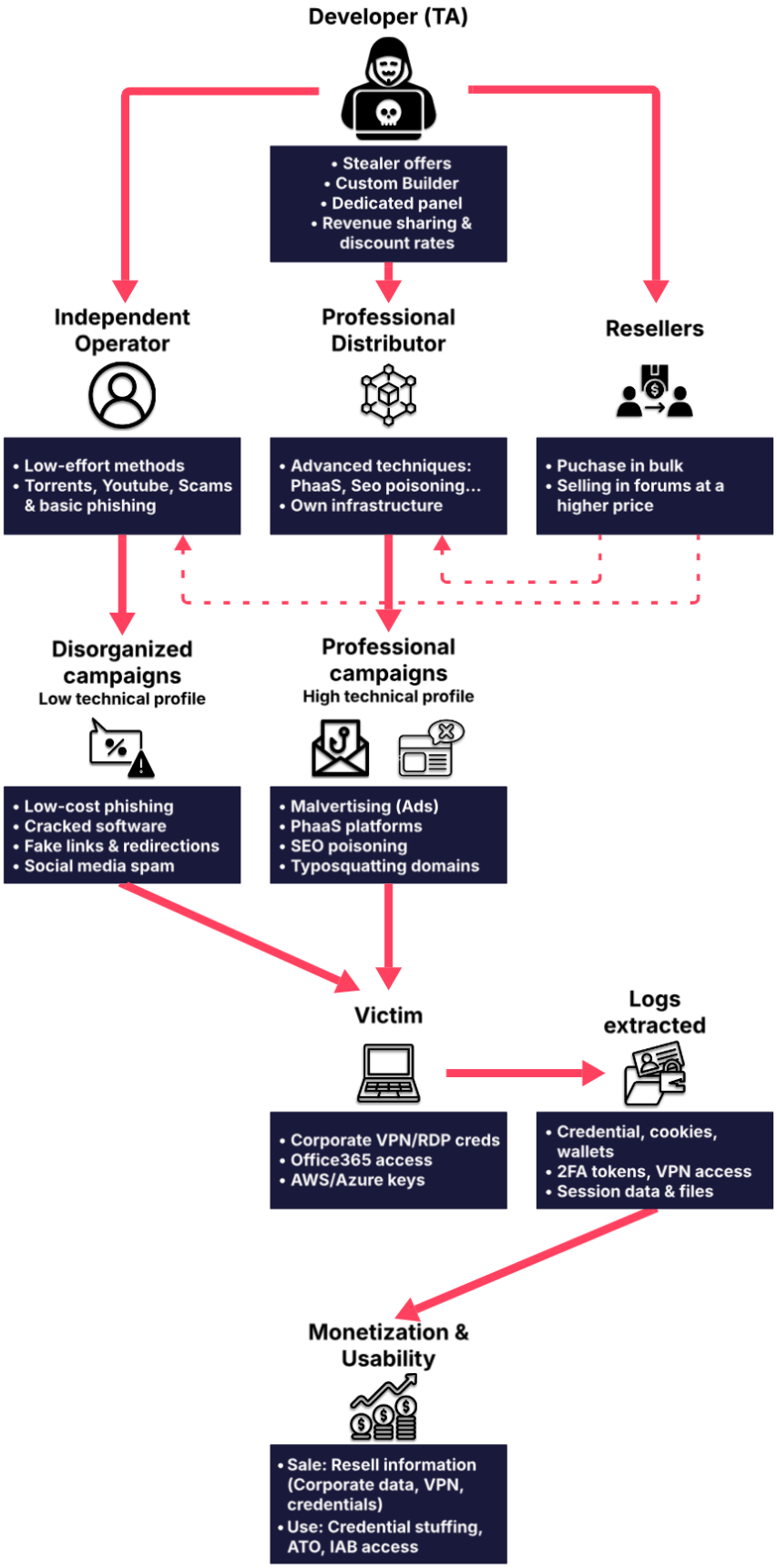
This system feeds back on itself, since the buyers and affiliates of the main tool can profit from it, while also being able to sell or use the information, which can be used for other adversaries or for the same market, perpetrating malware as a service.

Business Model - MaaS

The Malware-as-a-Service (MaaS) model has transformed the threat landscape by industrializing the near-infinite provision of malicious tooling. Instead of requiring deep technical knowledge to develop a stealer from scratch, any actor with sufficient financial capability to pay a monthly subscription can access ready-to-use infrastructure: a builder to generate payloads, a web panel that centralizes exfiltrated logs, command-and-control (C2) servers managed by the TA, and technical support.

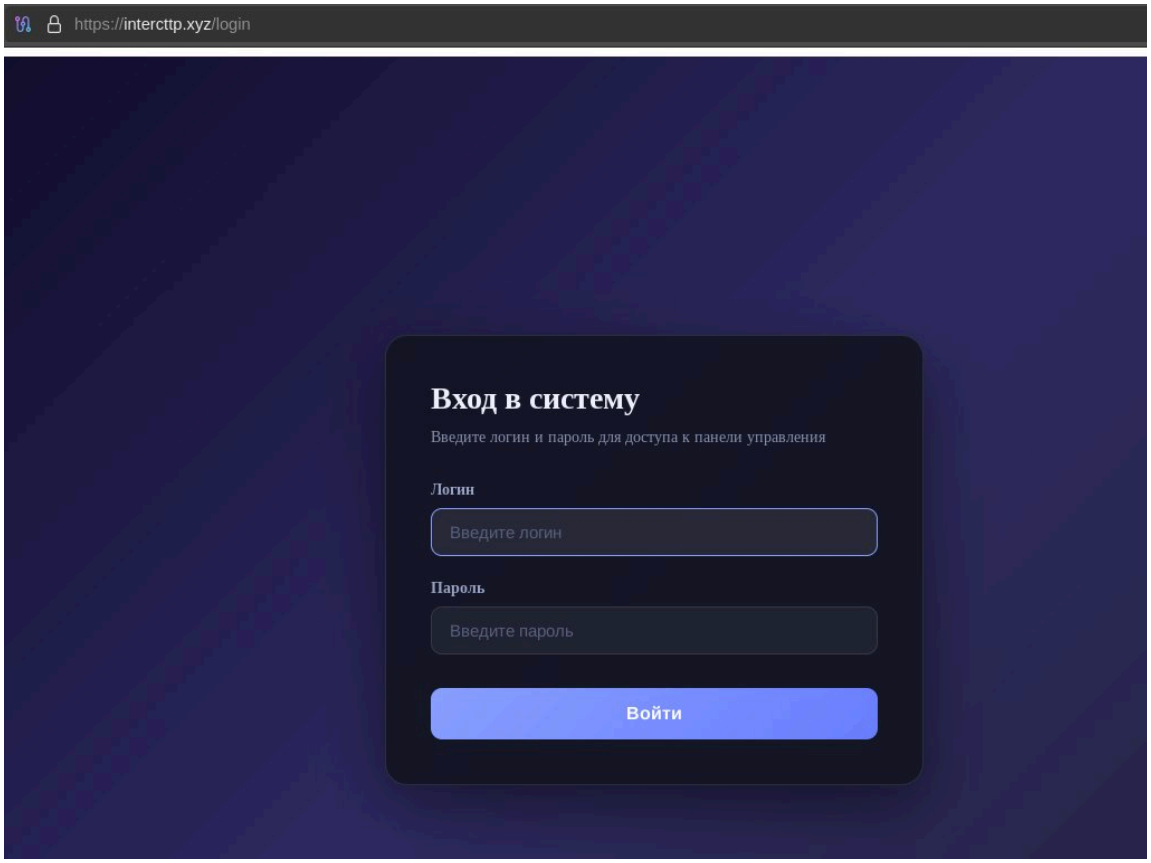
This model mirrors legitimate Software-as-a-Service (SaaS) logic, but applied to the criminal ecosystem. The stealer developer assumes the role of technical infrastructure provider, while clients, independent operators, professional distributors, or resellers, focus on malware distribution and monetization of stolen data.

This structure may shift over time, but it typically maintains key structural elements where the central and most critical actor is the developer, who creates the stealer and activates the entire operational model.



Phase 1: Development

- **Builder**: An application is created, typically with a GUI, that allows the client to generate customized stealer payloads. The builder offers configuration options such as what data to steal (browsers, wallets, specific files), which evasion techniques to enable, which C2 to use, and optionally obfuscation.
- **Panel**: A dashboard where exfiltrated logs are displayed in real time. It includes search functionality, filtering by data type, export to standard formats (CSV, JSON), and infection statistics.
- **Infrastructure**: Servers that receive exfiltrated data from infected machines. The TA manages availability, IP rotation in case of blacklisting, and traffic encryption.
- **Support:** Telegram channels, automated bots, FAQs, and in some cases 24/7 support for premium clients.
- **Service Design**: Different tiers are structured depending on the client profile, including independent operators, professional distributors, or resellers. Pricing varies based on capabilities and level of access to the tools.



C2 Dashboard, Login Page

Phase 2: Distribution

- **Disorganized Campaigns**: Typically run by independent operators who use the stealer through torrents, fake YouTube links, mass phishing, and similar techniques. These campaigns are broader in reach but generally lower in effectiveness.
- **Professional Campaigns:** Conducted by experienced operators who deploy malvertising campaigns and leverage advanced Phishing-as-a-Service (PhaaS) platforms. They may also use SEO poisoning techniques to increase victim volume. These campaigns require greater investment, coordination, and operational maturity, resulting in significantly higher effectiveness and quality of execution.

Phase 3: Infection & Extraction

- **Execution**: Depending on the campaign, the sample executes on the victim's machine while attempting to remain stealthy and avoid detection.
- **Collection**: It gathers browser credentials, cookies, wallets, 2FA seeds, specific files, and other sensitive data, storing them in structured logs.
- **Exfiltration**: It maintains communication with the TA's C2 infrastructure and, through encrypted protocols and often via Telegram channels, exfiltrates the victim's data.

Phase 4: Monetization & usability after infection

- **Sale**: The obtained logs can be resold on other marketplaces. Their value depends on factors such as whether they contain corporate credentials, VPN access, privileged accounts, association with large enterprises (e.g., Fortune 500 companies), sector relevance, and data recency.
- **Usage**: Logs are leveraged for credential stuffing attacks, testing stolen credentials across multiple services, conducting Account Takeovers (ATO) on financial portals, or selling access to Initial Access Brokers (IABs), which may subsequently enable other adversaries such as ransomware operators.

This phase (4) will be analyzed separately in greater depth, as it represents a critical post-infection stage from a threat intelligence perspective. This point can be addressed with circular restart of MaaS, as it can return information to the market, opening up other business opportunities or giving way to other tools.

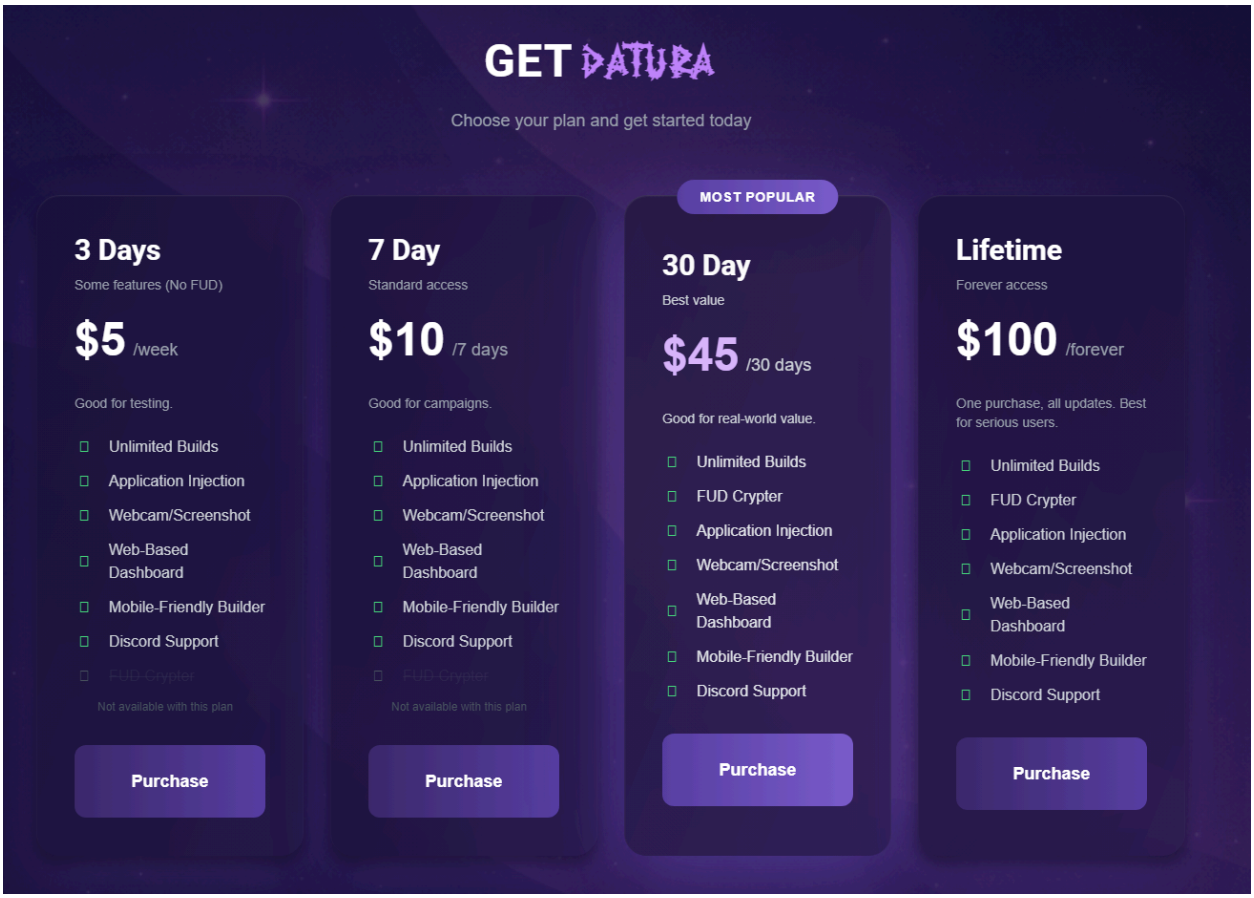
Licensing

Stealer Threat Actors offer different licensing structures depending on the customer's profile and needs. There are many models depending on the developer, but they are usually similar.

- **Monthly License:** The client pays a fixed monthly fee and receives unlimited access to the builder, panel, and support during that period. At the end of the month, renewal is required to continue using the service. If the client does not renew, access to the panel is revoked, and previous logs may or may not remain downloadable depending on the TA.
- **Lifetime**: The client pays a one-time fee and obtains lifetime access to the stealer, including all future updates. Threat actors usually offer this in limited slots, for example “only 50 lifetime licenses available”, to create artificial scarcity and prevent infrastructure saturation.
- **Weekly**: Some TAs offer 7-day licenses for clients who want to test the stealer before committing to a monthly subscription, or for low-budget operators running occasional campaigns.
- **Bulk/Reseller:** The threat actor provides volume discounts to actors who purchase multiple licenses for resale. For example: *"Buy 10 monthly licenses at $80 each instead of $150 retail and resell them at $120–$150"*.
- **Trial or Demo**: Some threat actors offer 3–7 day trials with limited functionality, for example a maximum number of builds or restricted log export, for clients evaluating the product.

Model	Duration	Typical Pricing	Target Client	Renewal Frequency
Monthly	30 days	100–300	Standard operators, distributors	Monthly
Lifetime	Indefinite	500–1,500	Early adopters, long-term actors	One-time
Weekly	7 days	50–100	Low-budget operators, testers	Weekly
Bulk/Reseller	30 days	60–100 per license	Resellers	Monthly (reseller-managed)
Trial/Demo	3-7 days	Free or low cost	Evaluators	N/A

The usual licensing fit is observed in Datura Stealer, where it complies with many models explained above:

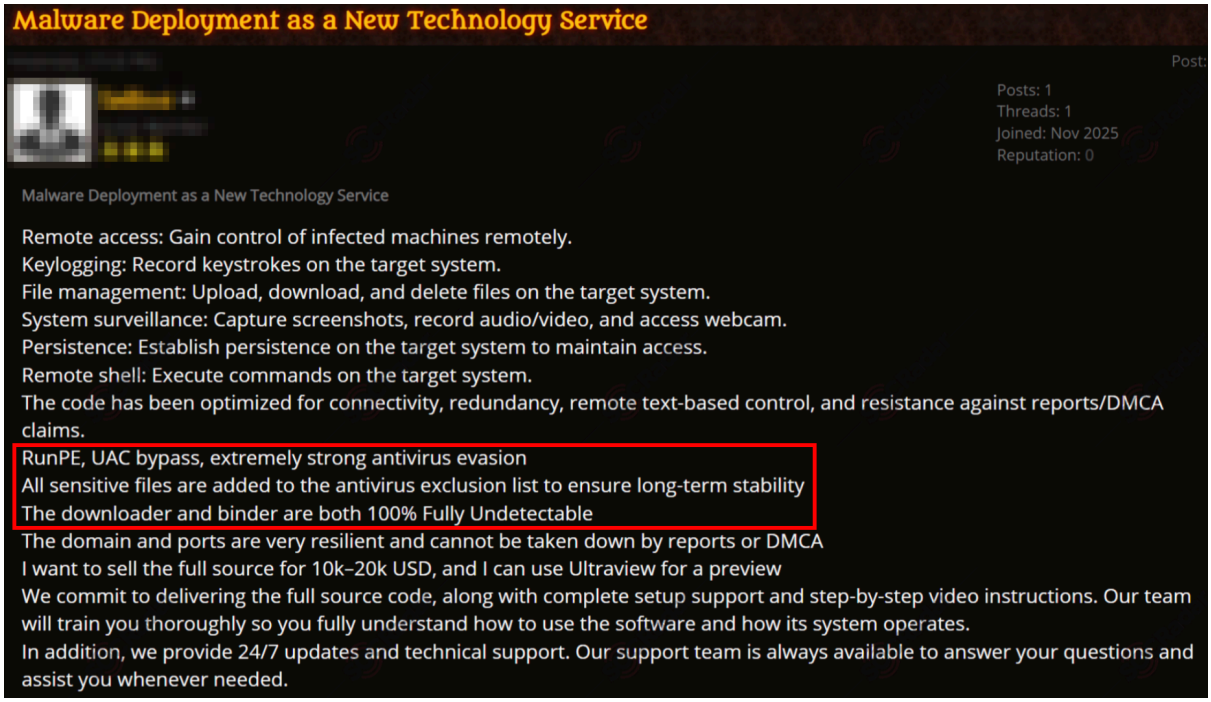


Pricing & Payment Methods

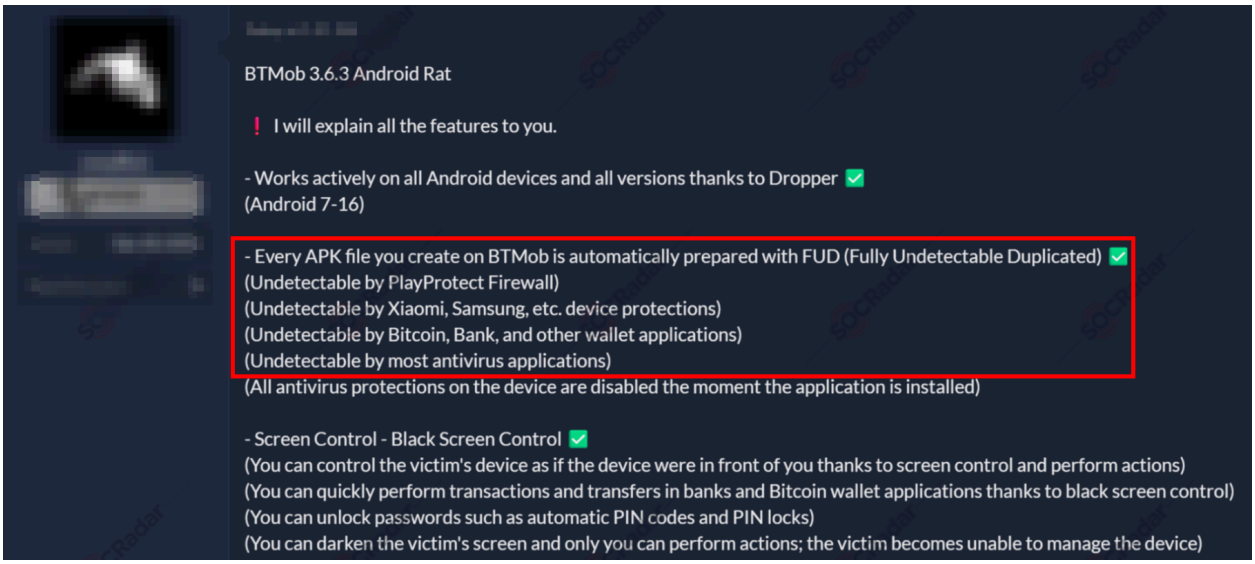
The pricing of MaaS stealers is not arbitrary. It is determined by multiple variables, as discussed previously, which reflect both the TA's strategic positioning and the technical characteristics of the product. Not all stealers operate within the same price range. Some are positioned at 80 dollars per month, while others reach 300 dollars or more per month depending on several factors.

These variables include:

- **Technical Sophistication**: Stealers with higher development investment, advanced evasion techniques such as process hollowing, AMSI bypass, and ETW patching, and stronger architecture justify higher pricing. Key indicators include detection rate and FUD duration, level of obfuscation, and data theft capabilities such as 30 or more browsers, 20 or more crypto wallets, 2FA seed extraction, and similar features. A stealer maintaining fewer than five detections on VirusTotal for weeks can charge significantly more than one that cannot.
- **Support and C2 Quality**: Technical support response time and infrastructure availability are critical differentiators. A vendor offering response times under four hours and C2 uptime above 99 percent, often backed by SLA and downtime compensation, can charge more than one providing basic 24 to 48 hour support without availability guarantees. A down C2 means lost logs for the operator, so reliability justifies premium pricing.
- **Threat Actor Reputation and Longevity**: Vendors with a track record of more than two years, no exit scams, long forum threads with hundreds of pages, and verified testimonials can charge premium prices based on trust. New TAs, typically active for zero to six months, often enter the market with aggressive pricing between 80 and 100 dollars per month to gain traction, then increase prices as reputation grows. A thread exceeding 100 pages with consistent reviews significantly increases pricing power.
- **Market Positioning**: Pricing is strongly influenced by direct competitors. When a dominant stealer faces disruption or is dismantled, competitors adjust pricing to capture demand. For example, after disruption affecting Lumma in May 2025, StealC increased pricing from 150 to 180 dollars per month, Vidar launched promotional offers, and new stealers entered the market in the 120 to 150 dollar range.



Source code sale for a Stealer Malware/Trojan, focus is on the evasion capabilities



Stealer advertisement in a hacker forum, once again the prominent function is being undetactable

Less-known stealers gain traction due to limited monitoring and are often underestimated. They may operate for months before scaling, rebranding, or disappearing, affecting many organizations before detection.

MaaS developers generate revenue not only from licensing fees but also from resale-related income streams, with profits typically coming from the following sources:

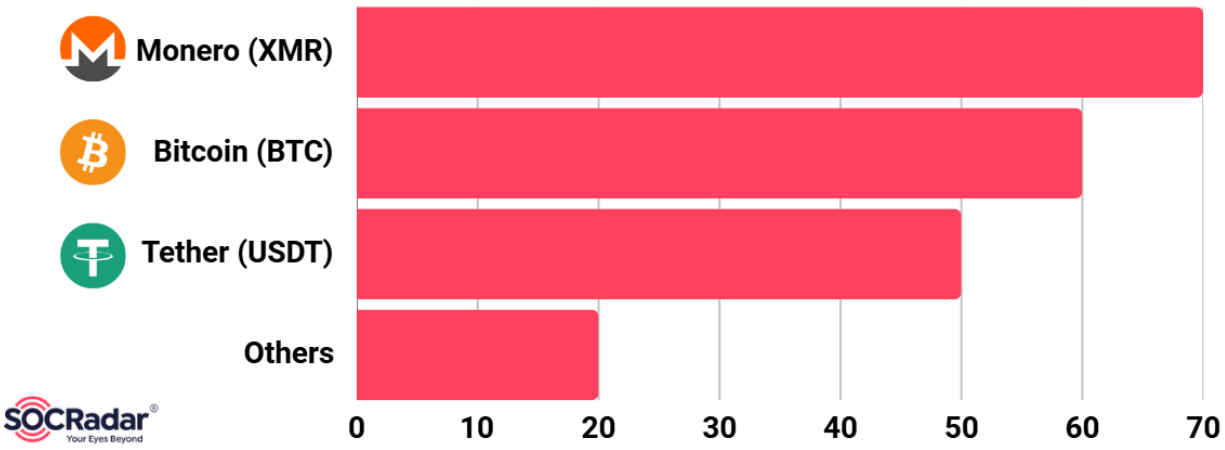
Type	Description
Monthly subscriptions	Fixed monthly fee per client. Base and more predictable revenue stream. Scales directly with a number of active clients.
Logs sold revenue sharing	Agreements with professional distributors where the TA receives 50-60% of each log sale. A distributor generating 60k/mo gross may provide to the threat actor at 60%. With 5-10 distributors, this can equal or exceed subscription revenue.
Lifetime licenses	High one-time payments between 500–1,500. Useful for capital injections during launch or promotions. Trade-off: lifetime clients do not generate recurring monthly revenue.
Source code sale	Exit strategy or pivot. Before shutting down, some TAs sell full source code to another actor for rebranding.

Payment Methods

MaaS operations rely exclusively on payment methods that prioritize anonymity. Credit cards, PayPal, and bank transfers are absent due to traceability and chargeback risks. TAs accept cryptocurrencies as the sole payment method, with preferences based on privacy levels.

- **Monero** (XMR) is considered the gold standard due to default privacy. Transactions are opaque, with sender, receiver, and amount concealed through ring signatures and stealth addresses. It remains non-traceable even with significant law enforcement resources. Approximately 70 to 80 percent of TAs accept XMR, many as the only option or with a 5 to 10 percent discount to incentivize its use.
- **Bitcoin** (BTC) is universally accepted because of liquidity and maturity. However, blockchain transparency requires vendors to use mixers or tumblers such as Wasabi or Whirlpool to obscure origin, adding 2 to 5 percent in fees and additional operational risk. Some Threat Actors apply a 5 to 10 percent surcharge for BTC payments.
- **Tether** (USDT) is accepted due to price stability, with strong preference for TRC-20 on the Tron network over ERC-20 on Ethereum because of significantly lower transaction fees. However, USDT does not provide privacy and Tether Ltd. has demonstrated the ability to freeze wallets upon request from authorities.

Other cryptocurrencies such as Litecoin, Dash, and Zcash have minor adoption.



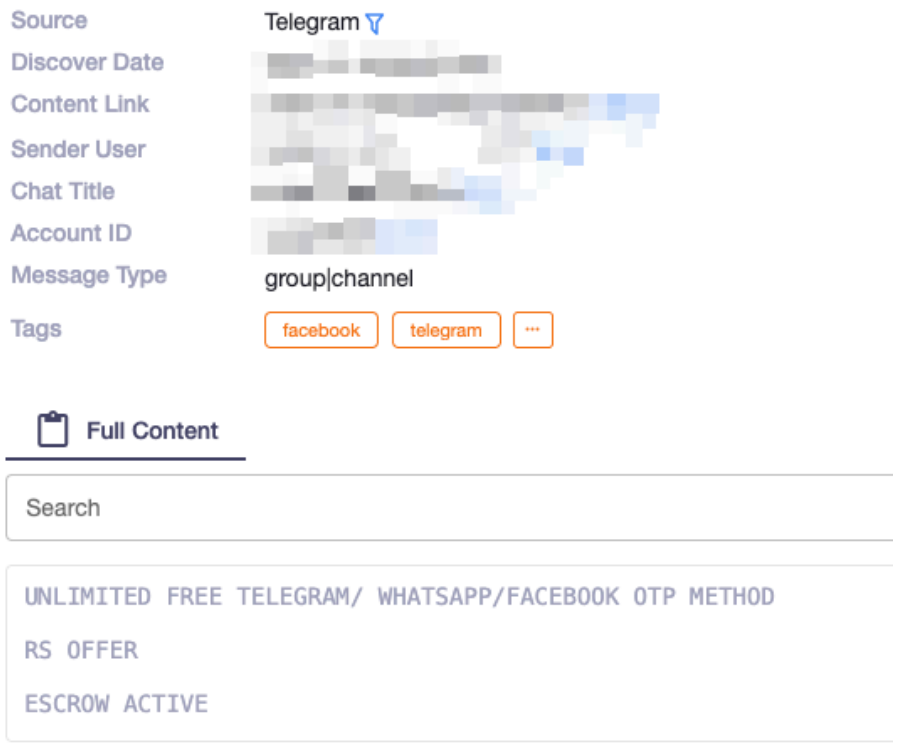
Cryptocurrency distributions

Payment mechanisms vary depending on the threat actor operating the network, although they often coexist:

- **Direct Payments, Wallet-to-Wallet**: The client requests a license, the TA provides a disposable wallet address, the client sends payment, and the Threat Actor confirms receipt manually or through a blockchain-monitoring bot before delivering the builder and panel credentials.
  Advantages: No intermediaries, no additional fees, fast confirmation in minutes for XMR and 10 to 30 minutes for BTC.
  Disadvantages: No buyer protection if the TA scams, requires prior trust or strong reputation. Common among established Threat Actors.

- **Forum Escrow**: The client deposits funds into forum escrow managed by an administrator or moderator. The Threat Actor delivers the product, the client has 24 to 72 hours to validate functionality, then releases funds or opens a dispute. The moderator reviews evidence and decides.
  Advantages: Protection for both parties, suitable for first transactions without prior history.
  Disadvantages: Escrow fee typically 3 to 5 percent, depends on trust in forum administrators. Forums with active escrow include XSS, Exploit, and RAMP.

- **Automated Telegram Bots**: The client interacts with a bot such as @StealerName_bot. The bot generates an invoice with a unique wallet address and QR code. After payment, the bot monitors the blockchain and confirms automatically, delivering panel credentials and the builder download link.
  Advantages: Fully automated 24/7, scalable for high transaction volumes, reduces manual support workload.
  Disadvantages: Requires technical development and blockchain API integration, risk of bugs such as failure to detect payment. Common among mid to high-tier Threat Actors with more than 200 clients.



SOCRadar Platform, Threat Hunting findings

SOCRadar Platform, Threat Hunting findings #2

Builder & PoC Offerings

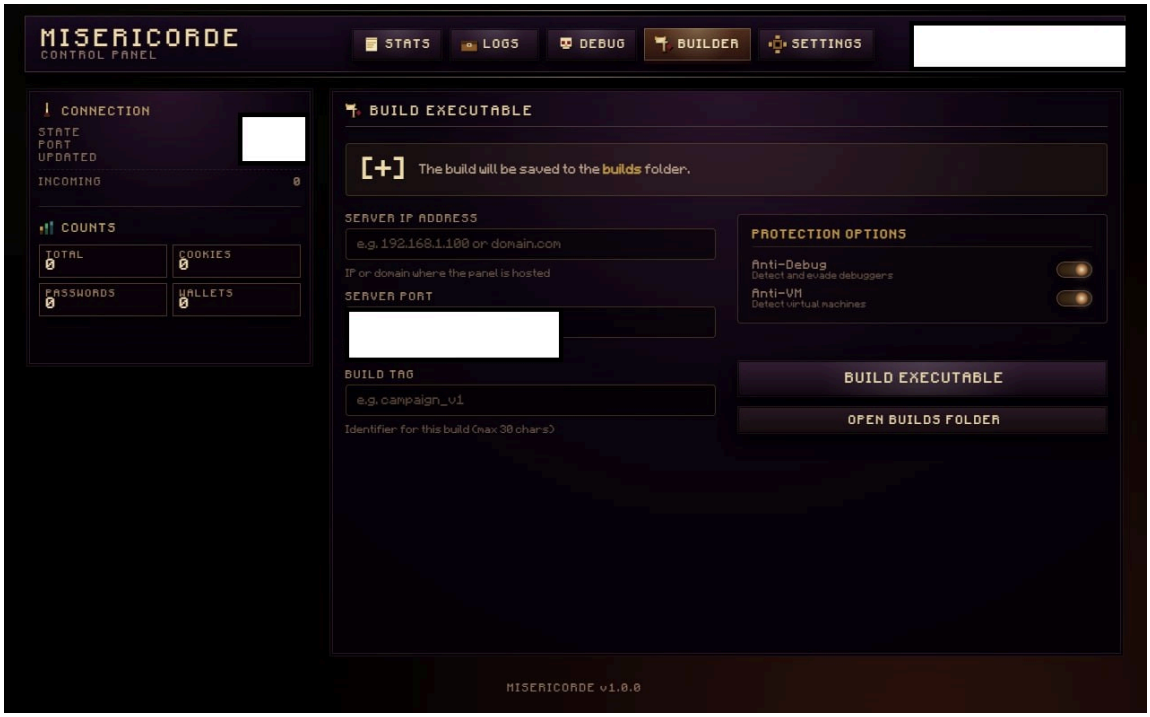
The commercialization of infostealers in the underground follows established patterns combining technical demonstration through proof of concept, product accessibility via the builder, and strategic presence in marketplaces where Threat Actors build reputation and attract clients.

The builder is the application that allows customers to generate customized payloads without programming knowledge. It is the core product delivered after payment, and its quality directly determines user experience and operational effectiveness.

Typical builder characteristics include:

- **Graphical Interface**: Usually intuitive interfaces built with Windows Forms, WPF, or Electron. Some TAs offer command line interfaces for advanced operators seeking automation.
- **Configuration Options**: Capabilities vary depending on the developer's technical maturity and ecosystem.
    - **Theft Targets**: Checkboxes allowing selection of browsers, crypto wallets, messaging applications such as Telegram and Discord, FTP or VPN clients, and files by extension.
    - **C2 Configuration**: URL of the command and control server. Some Threat Actors allow custom C2 infrastructure, others restrict usage to their own servers. Protocol selection and heartbeat interval may be configurable.
    - **Evasion and Obfuscation**: String encryption, control flow obfuscation, anti virtual machine or sandbox, anti debug techniques, sleep or junk code injection, and icon spoofing to mimic legitimate applications.
    - **Persistence**: Startup registry keys, scheduled tasks, shortcut in Startup folder. Many modern stealers do not implement persistence, following a grab-and-go philosophy with single execution and minimal footprint.
    - **Output**: Executable format such as .exe, architecture selection x86 or x64 or both, compression or packing such as UPX or custom packers.

What is received is an executable file that is ready to be distributed as an operator, each creation usually having a unique signature and different obfuscations to avoid mass detection, usually by YARAs or signatures.



Misericorde Control Panel, executable Builder

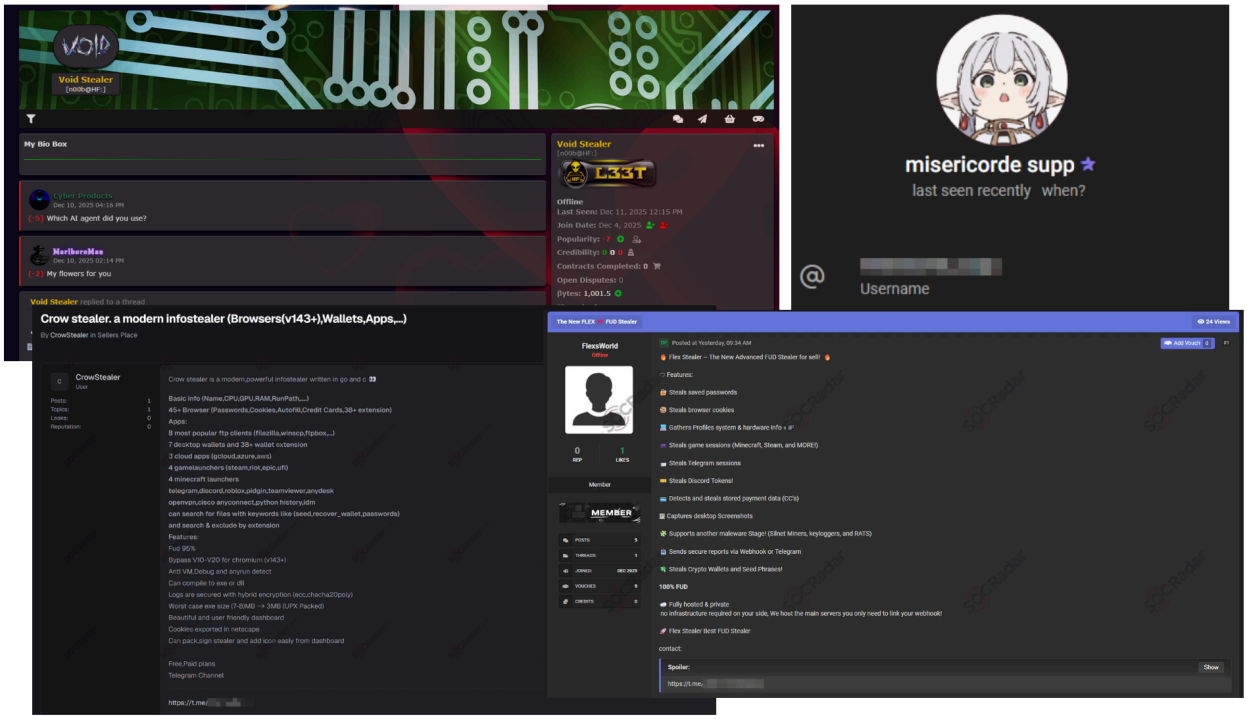
As seen above, the channels where builders communicate are also on Telegram and underground markets, where it is common to find launch offers or, after speaking with the developer, they offer tests on a PoC to verify their authenticity, as well as the videos or images mentioned above.

The accumulation of capabilities directly impacts sales strategy. The broader the feature set, the greater the appeal to individual operators and professional distributors, increasing coverage and long-term visibility for the stealer.

Technical Analysis

As discussed previously, the industry tends to focus on dominant threats that already have high media visibility. However, it often overlooks those lesser known threats that are still in development and represent invisible risks, becoming the weakest link due to the lack of monitoring and analysis around them.

This situation is tangible in stealers such as Void, Crow, Flex, Datura, Misericorde, RS or Saturn. These are some of the samples frequently found in these types of markets and that form part of the operational blind spot for many companies.

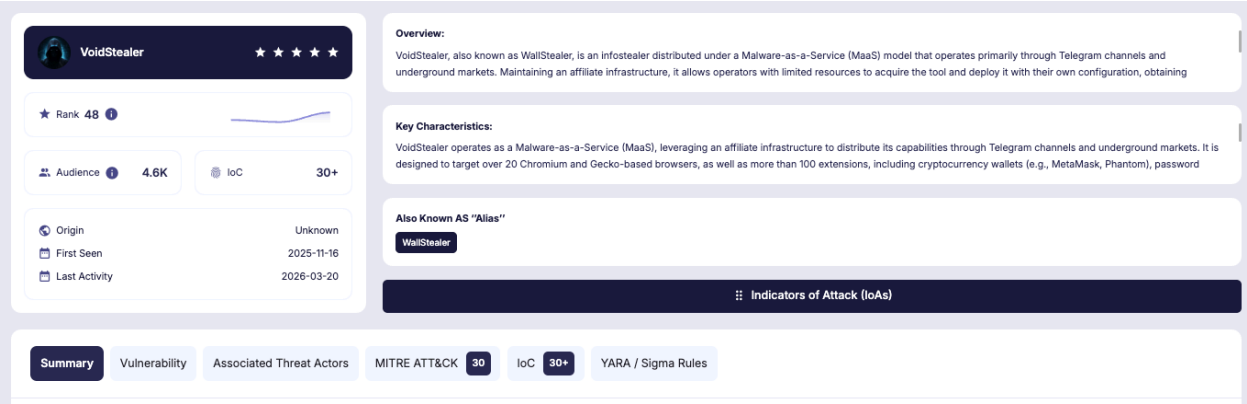


Various less known stealers and dissemination platforms

The lack of attention given to these emerging projects allows them, during their first months of life, to cause significant disruption. This creates a window of time in which these TAs can recruit affiliates and run campaigns until they gain broader recognition, leaving dozens of victims behind.

Void Stealer

A direct example of the above is Void Stealer, a project that emerged in late 2025 and already presents a fairly mature infrastructure that has been collecting victims for months. Maintaining a low profile, like the rest of the mid and low level stealers, it maintains a presence on Telegram and underground forums where its services are offered, providing builders and panels for operators.



Void Stealer in SOCRadar Threat Actor/Malware Intelligence

Core Capabilities

Void Stealer implements the standard capability set of a modern stealer, with some technical particularities that distinguish it.

- **Credential and Session Data Theft:**
  - Chromium-based browsers (Chrome, Edge, Brave, Opera, Vivaldi, and derivatives), including stored credentials, session cookies, and browsing history
  - Credentials and cookies extracted through direct access to SQLite databases
  - Browser extensions, including tokens from crypto wallet extensions such as MetaMask, Phantom, and Coinbase Wallet
- **Cryptocurrency Wallets:**
  - Desktop wallets such as Exodus, Atomic Wallet, and Electrum
  - Configuration files and seeds stored in known system paths
- **System Information:**
  - Operating system data such as Windows version, architecture (x86/x64), hostname, and current user
  - Region and language configuration, including timezone, keyboard layout, and regional settings
  - Hardware information such as CPU, RAM, and GPU extracted through registry queries
- **FTP and Application Credentials:**
  - Credentials from FTP clients and third-party applications stored in known paths

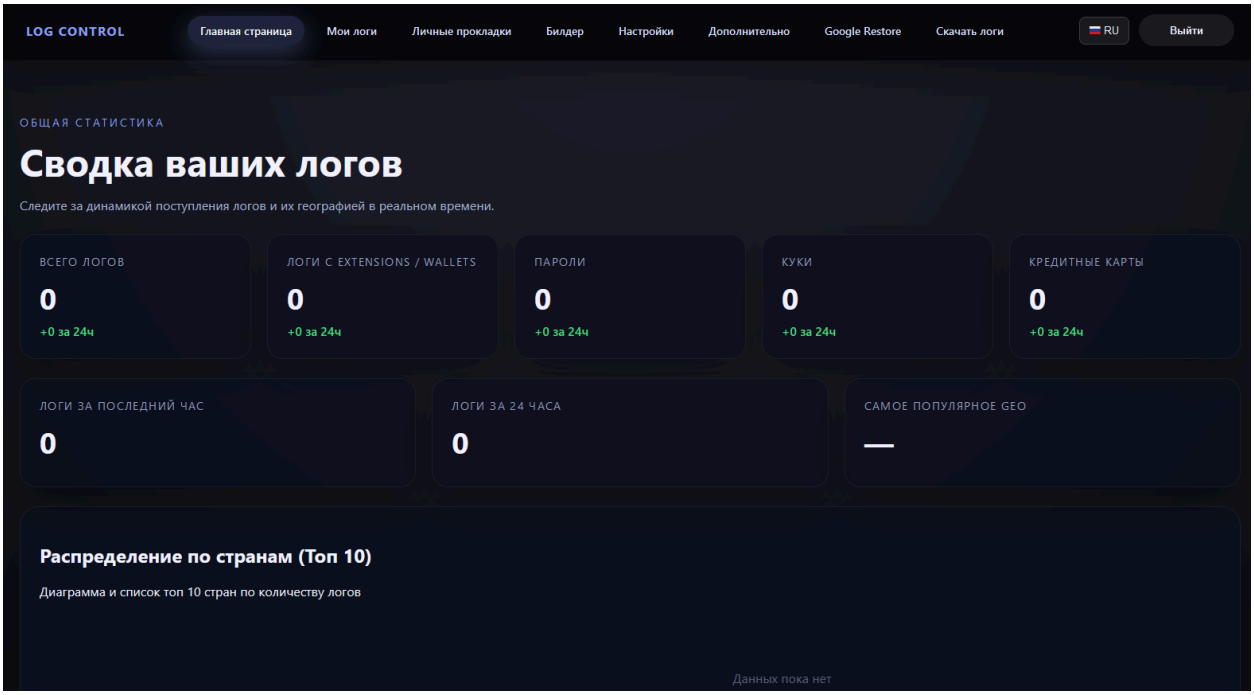
- **Optional Webcam Capture:**
  - Ability to capture images from the compromised system's webcam if available and enabled by the operator in the builder
  - This function is uncommon in stealers of this profile and suggests use in sextortion campaigns or targeted attacks where visual capture adds value
- **Messaging and Application Tokens:**
  - Telegram sessions (tdata files)
  - Discord tokens

Additionally, Void uses an intermediate C2 resolution mechanism by abusing a Steam profile, which is a relatively uncommon technique at this tier.

No persistence mechanisms were identified. Void maintains a deliberately low execution profile, it runs, collects, exfiltrates and terminates, minimizing its detection window. This behaviour is consistent with the stealer-as-a-service model where speed of exfiltration takes priority over long-term implant survival, and contributes directly to its evasion effectiveness against both endpoint solutions and user awareness.

Panel

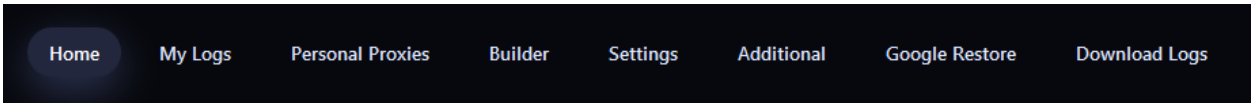
The administration panel is the interface where the Void Stealer operator visualizes, manages and downloads the exfiltrated logs. Unlike panels from more professional stealers that offer advanced analytics, multi-tenant dashboards and APIs for integration with other systems, Void's panel is functional but minimalistic, oriented toward individual operators or small teams, similar to what has been previously observed with Datura or Misericorde.



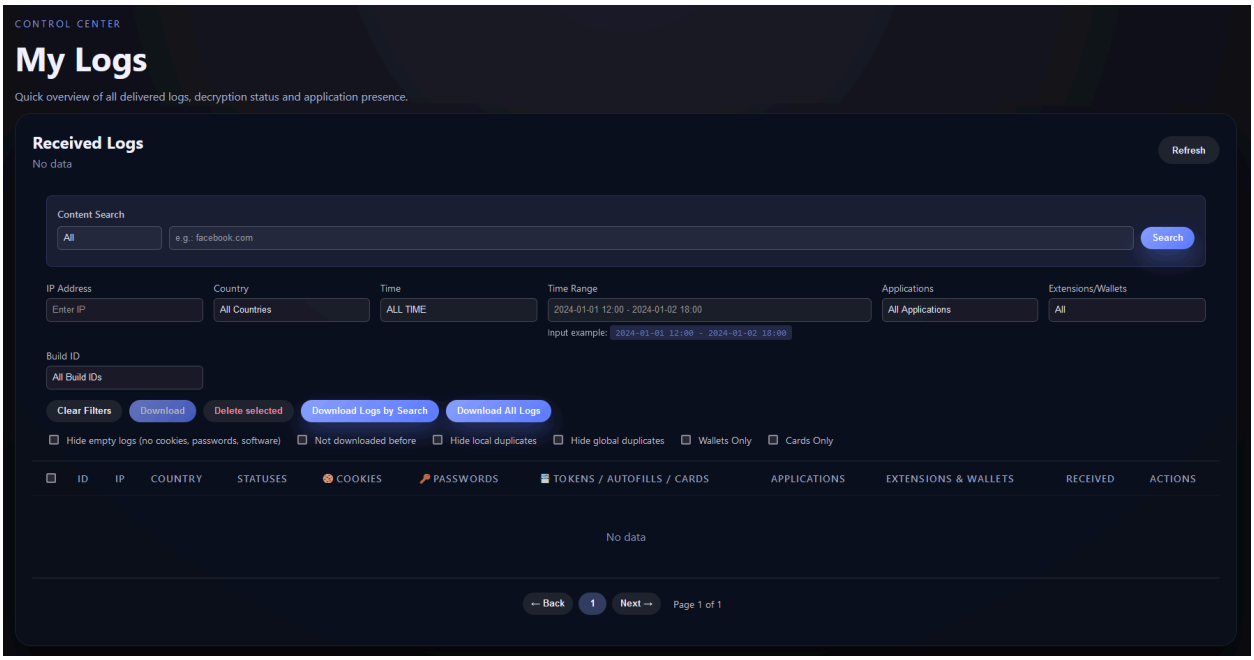
Void Stealer administration panel displaying log statistics and basic operator controls

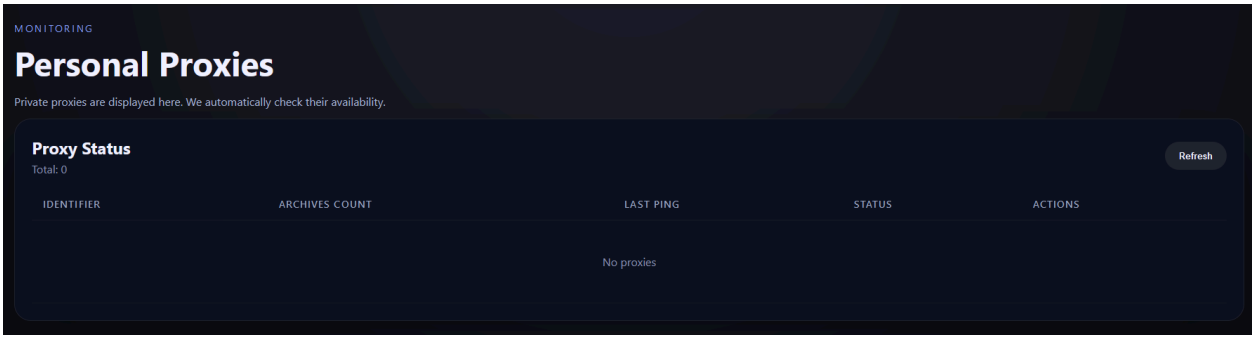
The panel is accessed through a username and password. It lacks common security mechanisms present in more professional stealers such as 2FA or TOTP.

Nevertheless, the typical functionality expected from malware in this family is present, providing the operator with access to logs, proxies, the builder and other operational aspects:



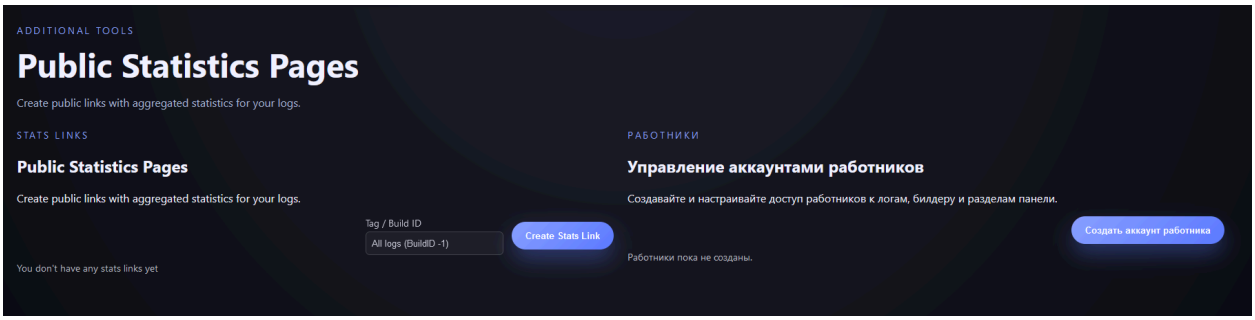
In the most common sections, log data and proxy configuration can be observed:





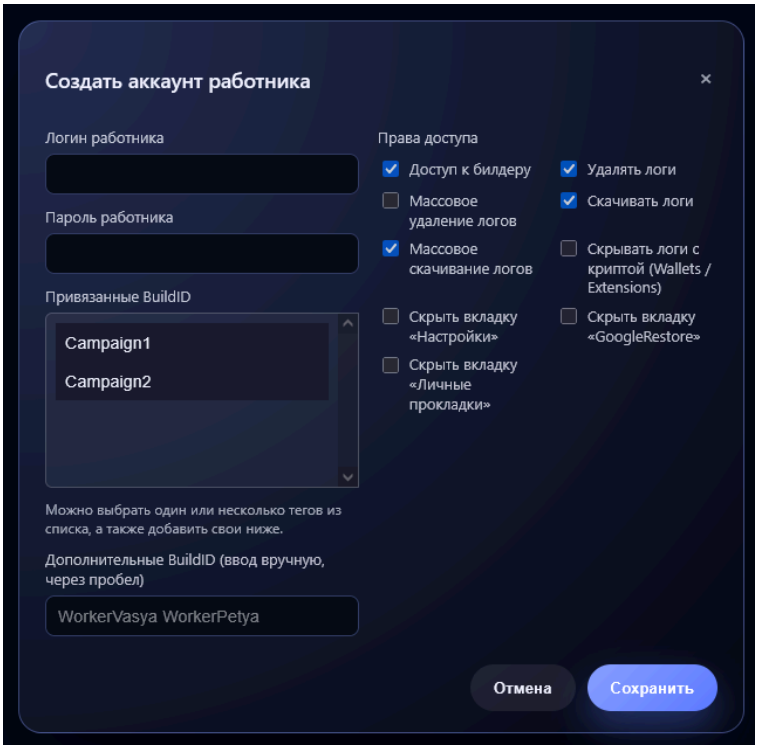
On the other hand, within the "Additional" section there is the possibility to generate statistics based on logs by ID, allowing filtering by campaign. This functionality can also be useful for showcasing results in underground markets.

It also supports multi user access for managing small teams, a typical scenario for resellers or organized groups.



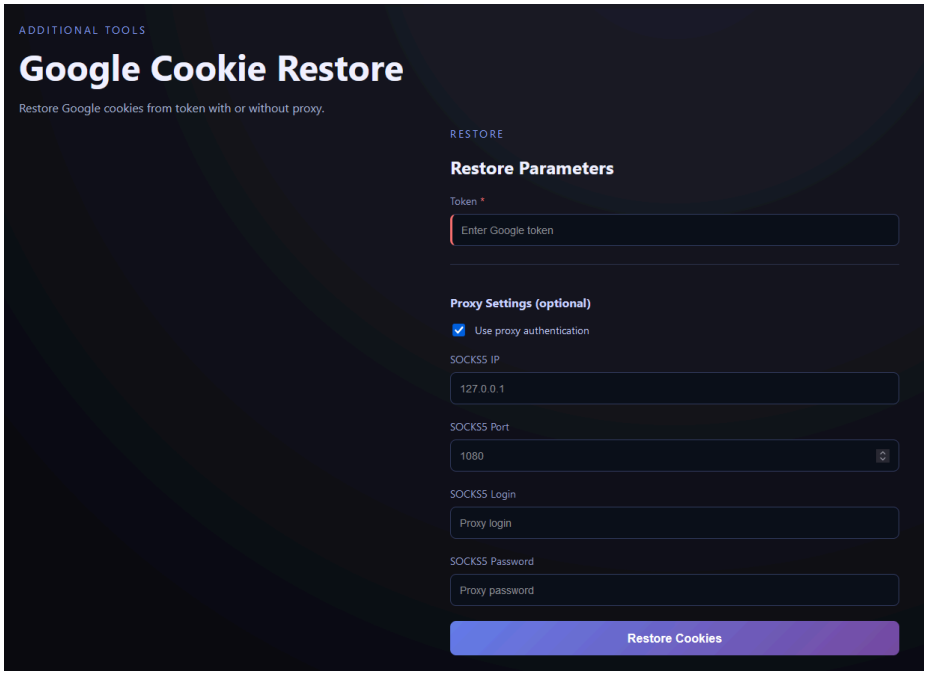
Void Stealer public statistics page for generating shareable log dashboards

Campaign management view showing generated links and log access controls

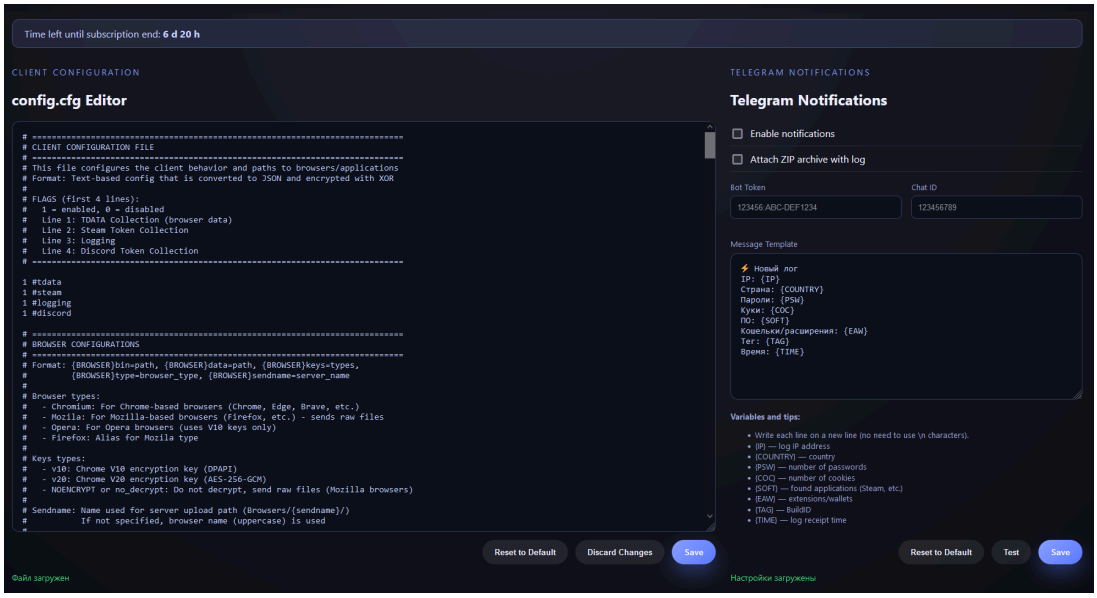


Worker account creation panel with role-based permissions and access settings

Within the Google Cookie Restore section, it is possible to regenerate active Google session cookies from a stolen token. This allows the operator to anonymize the operation using proxies and to geolocate from a location close to the victim:



Finally, the most important section is the configuration of settings and the builder:

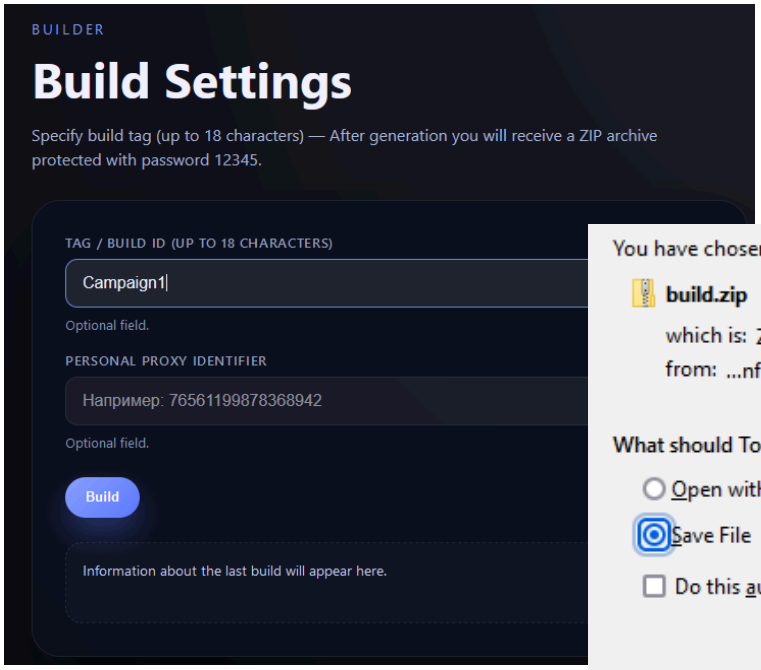


Here the operator can modify the configuration file to define what data should be collected, including the option to add a Telegram bot that will send notifications with the information that will later appear in the panel after infection.

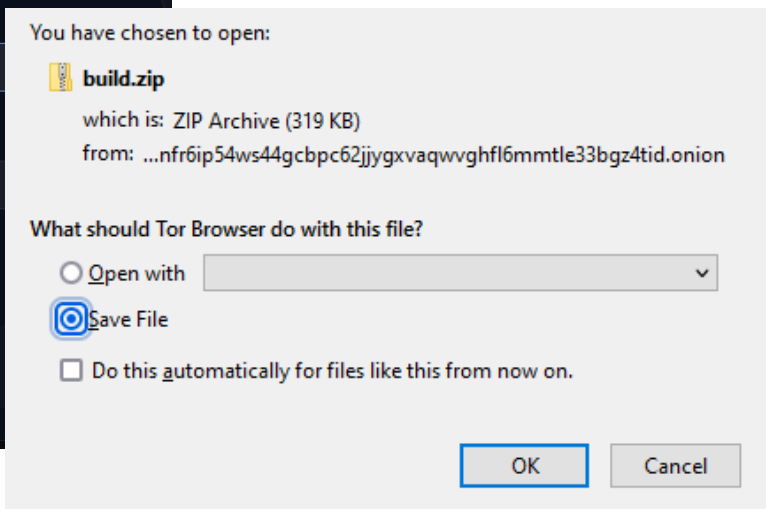
Builder

The builder is the tool used by the operator to generate customized Void Stealer payloads. Its quality and interface directly determine the usability of the stealer for operators with different levels of technical expertise.

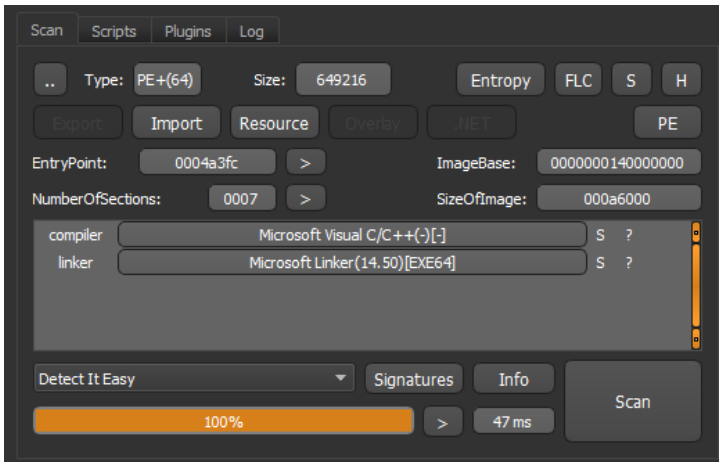
As seen previously, everything is integrated within the panel itself. Once configured, the operator can set campaign tags and even proxies, and the malware will be downloaded with the previously defined configuration.



Builder Settings section



Builder output executable

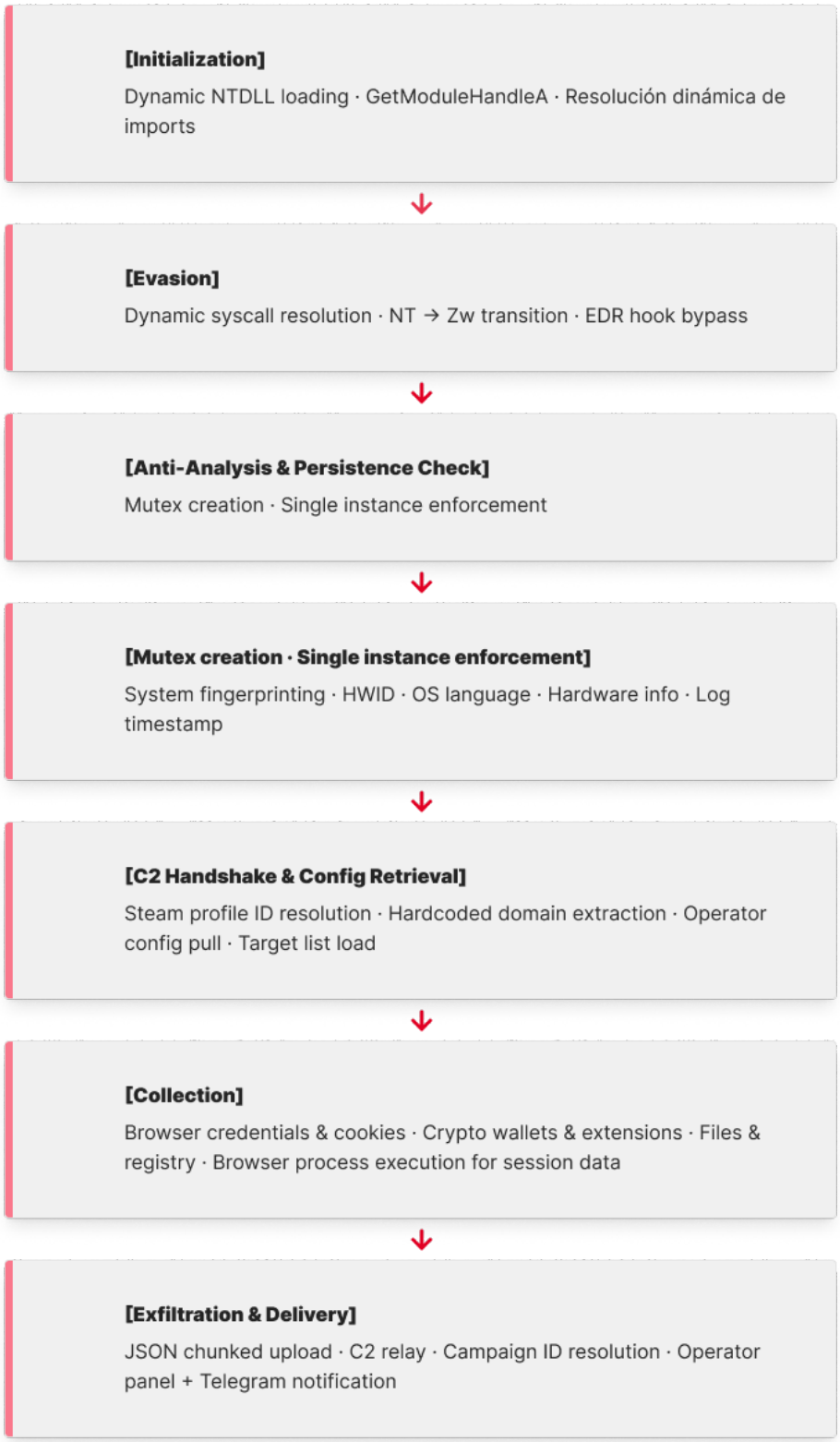


← The result will be a C++ binary belonging to Void Stealer.

Modus Operandi

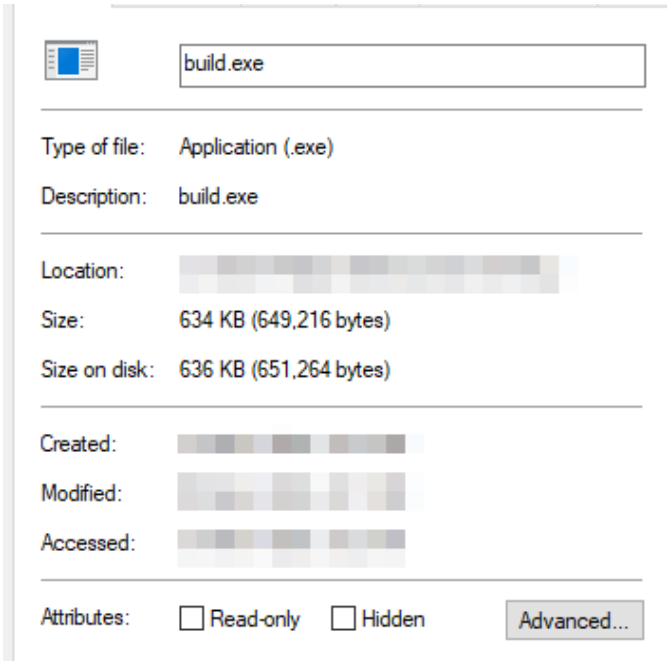
After obtaining Void and analyzing both the panel and the builder, it is possible to determine that it follows an infrastructure pattern similar to other stealers. Regarding execution, the following diagram presents the end to end malware flow, from initial execution to the final delivery of the exfiltrated data to the operator.

As can be observed, it presents several interesting peculiarities that will be examined in detail below.

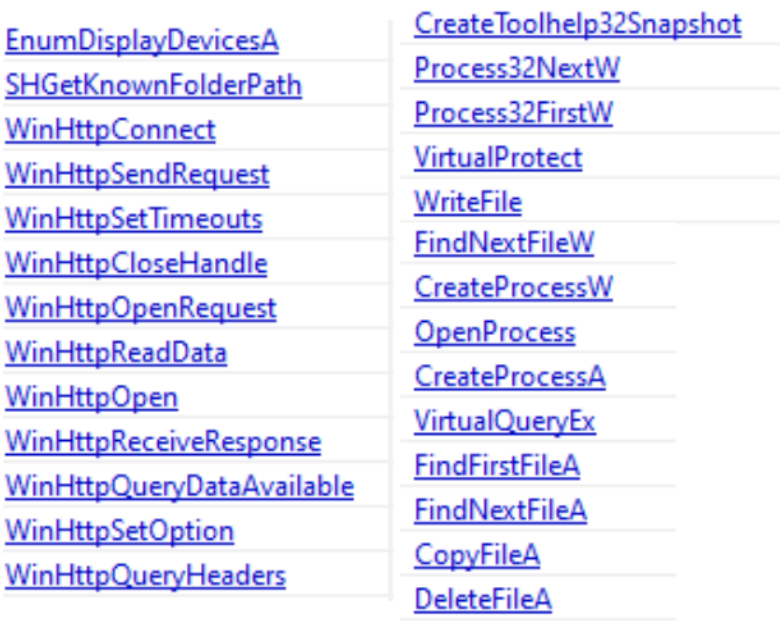


Technical Details

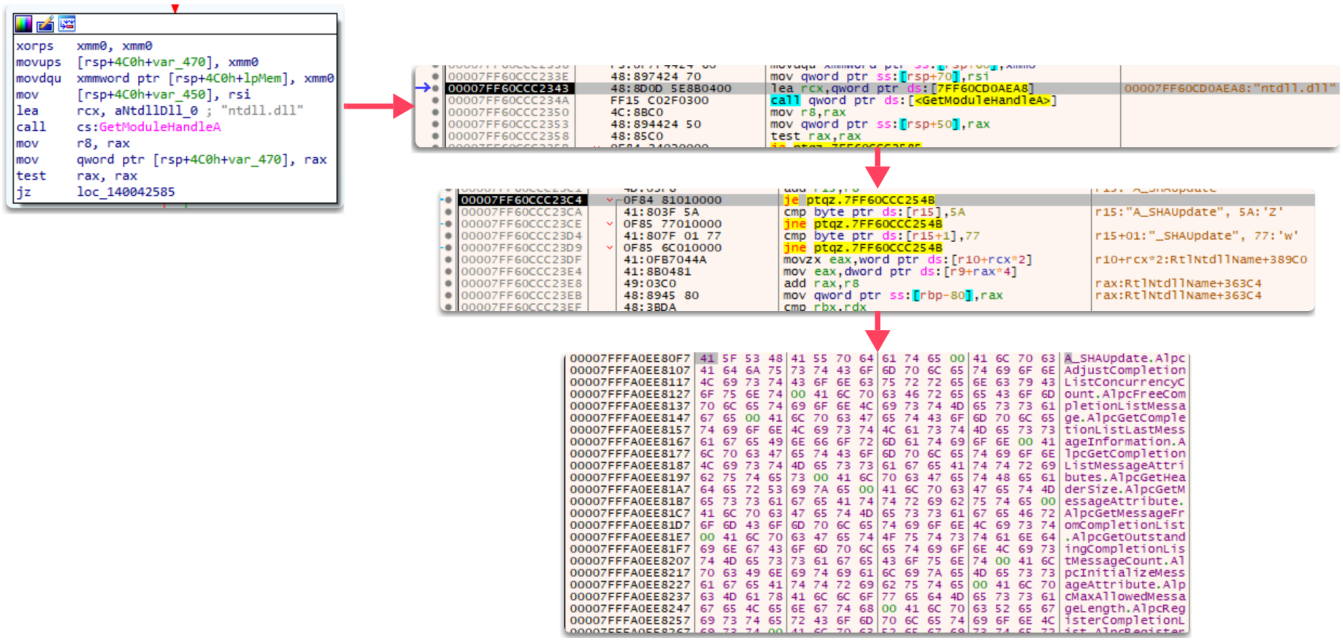
Starting from the Void Stealer binary, it can be observed that it is a PE executable targeting Windows, typically with a size ranging between 600 and 700 KB:



It imports libraries commonly used by legitimate software, but it is also clear that it will maintain external communications. Typical imports are present for process interaction, file access, communications and loading additional libraries:



During the early stages of Void Stealer, runtime loading of libraries and imports can be observed, in this case abusing ntdll. This is a common technique used to avoid triggering certain detection engines and to make static analysis more difficult.



Dynamic syscall resolution via ntdll to bypass EDR monitoring and evade static analysis

This is where the so-called Dynamic Syscall resolution and transition is implemented in order to evade or bypass EDR solutions. Modern EDRs frequently hook functions within NT, therefore the malware resolves Zw calls instead. In this technique it follows the steps below:

1. Enumerate exports from ntdll,
2. Store Zw* + address,
3. Receive "NtXXXX",
4. Convert Nt to Zw,
5. Search in the table,
6. Scan the ntdll stub,
7. Locate the syscall instruction,
8. Store the clean syscall address,

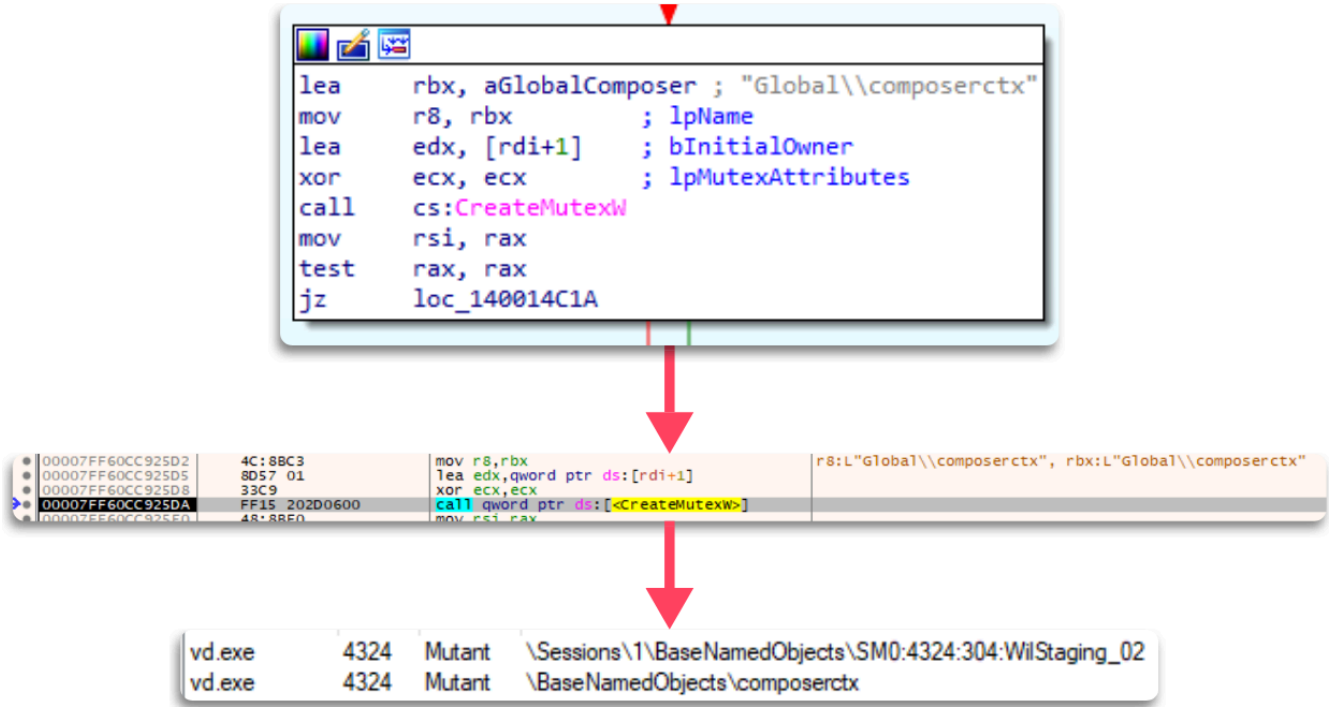
Runtime API resolution and syscall mapping via ntdll

As a result, it has now increased its capabilities and avoided potential detection.

00007FFFA0E2DF80	Export	232	NtAlpcCreateResourceReserve
00007FFFA0E2DFA0	Export	1817	ZwAlpcCreateSectionView
00007FFFA0E2DFA0	Export	233	NtAlpcCreateSectionView
00007FFFA0E2DFC0	Export	1818	ZwAlpcCreateSecurityContext
00007FFFA0E2DFC0	Export	234	NtAlpcCreateSecurityContext
00007FFFA0E2DFE0	Export	1819	ZwAlpcDeletePortSection
00007FFFA0E2DFE0	Export	235	NtAlpcDeletePortSection
00007FFFA0E2E000	Export	1820	ZwAlpcDeleteResourceReserve
00007FFFA0E2E000	Export	236	NtAlpcDeleteResourceReserve
00007FFFA0E2E020	Export	1821	ZwAlpcDeleteSectionView
00007FFFA0E2E020	Export	237	NtAlpcDeleteSectionView
00007FFFA0E2E040	Export	1822	ZwAlpcDeleteSecurityContext
00007FFFA0E2E040	Export	238	NtAlpcDeleteSecurityContext
00007FFFA0E2E060	Export	1823	ZwAlpcDisconnectPort
00007FFFA0E2E060	Export	239	NtAlpcDisconnectPort
00007FFFA0E2E080	Export	1824	ZwAlpcImpersonateClientContainerOfPort
00007FFFA0E2E080	Export	240	NtAlpcImpersonateClientContainerOfPort
00007FFFA0E2E0A0	Export	241	NtAlpcImpersonateClientOfPort
00007FFFA0E2E0A0	Export	1825	ZwAlpcImpersonateClientOfPort

Export table analysis and mutex creation for persistence control

From this point, the malware checks for and/or creates the Mutex that will allow it to avoid reinfections on the same system.

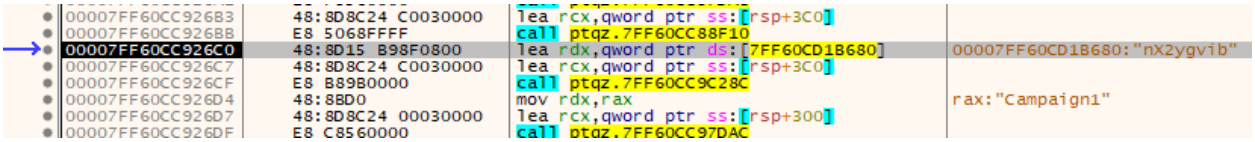


Mutex creation used to prevent reinfection and ensure single instance execution

The malware then retrieves configuration information where it contains the internal list of items it will inspect next, as well as the campaign identifiers previously defined.

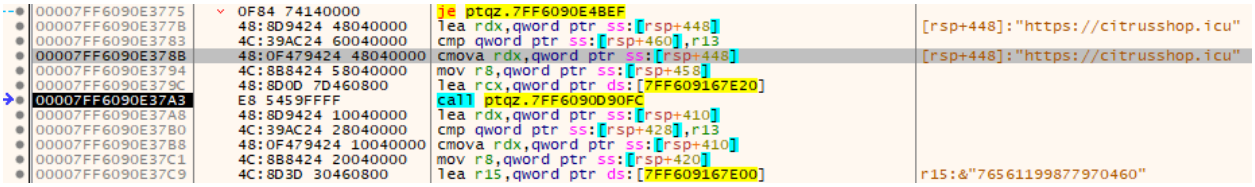
```
00000296719B6EA0  4C 6F 67 20 64 61 74 65  00 00 00 00 00 00 00 00  Log date........
00000296719B6EB0  08 00 00 00 00 00 00 00  0F 00 00 00 00 00 00 00  ................
00000296719B6EC0  48 57 49 44 00 00 00 00  00 00 00 00 00 00 00 00  HWID............
00000296719B6ED0  04 00 00 00 00 00 00 00  0F 00 00 00 00 00 00 00  ................
00000296719B6EE0  53 79 73 74 65 6D 20 4C  61 6E 67 75 61 67 65 00  System Language.
00000296719B6EF0  0F 00 00 00 00 00 00 00  0F 00 00 00 00 00 00 00  ................
00000296719B6F00  50 72 6F 63 65 73 73 6F  72 00 00 00 00 00 00 00  Processor.......
00000296719B6F10  09 00 00 00 00 00 00 00  0F 00 00 00 00 00 00 00  ................
00000296719B6F20  49 6E 73 74 61 6C 6C 65  64 20 52 41 4D 00 00 00  Installed RAM...
00000296719B6F30  0D 00 00 00 00 00 00 00  0F 00 00 00 00 00 00 00  ................
00000296719B6F40  80 30 9A 71 96 02 00 00  00 00 00 00 00 00 00 00  .0.q............
00000296719B6F50  10 00 00 00 00 00 00 00  1F 00 00 00 00 00 00 00  ................
00000296719B6F60  47 72 61 70 68 69 63 73  20 63 61 72 64 00 00 00  Graphics card...
00000296719B6F70  0D 00 00 00 00 00 00 00  0F 00 00 00 00 00 00 00  ................
00000296719B6F80  43 6F 6D 70 75 74 65 72  20 4E 61 6D 65 00 00 00  Computer Name...
00000296719B6F90  0D 00 00 00 00 00 00 00  0F 00 00 00 00 00 00 00  ................
00000296719B6FA0  44 6F 6D 61 69 6E 20 4E  61 6D 65 00 00 00 00 00  Domain Name.....
00000296719B6FB0  0B 00 00 00 00 00 00 00  0F 00 00 00 00 00 00 00  ................
00000296719B6FC0  4D 61 63 68 69 6E 65 49  44 00 00 00 00 00 00 00  MachineID.......
00000296719B6FD0  09 00 00 00 00 00 00 00  0F 00 00 00 00 00 00 00  ................
00000296719B6FE0  55 73 65 72 20 4E 61 6D  65 00 00 00 00 00 00 00  User Name.......
00000296719B6FF0  09 00 00 00 00 00 00 00  0F 00 00 00 00 00 00 00  ................
00000296719B7000  41 64 6D 69 6E 20 47 72  6F 75 70 00 00 00 00 00  Admin Group.....
00000296719B7010  0B 00 00 00 00 00 00 00  0F 00 00 00 00 00 00 00  ................
00000296719B7020  49 6E 74 65 67 72 69 74  79 00 00 00 00 00 00 00  Integrity.......
00000296719B7030  09 00 00 00 00 00 00 00  0F 00 00 00 00 00 00 00  ................
00000296719B7040  54 69 6D 65 5A 6F 6E 65  00 00 00 00 00 00 00 00  TimeZone........
```

Collected system and host profiling data

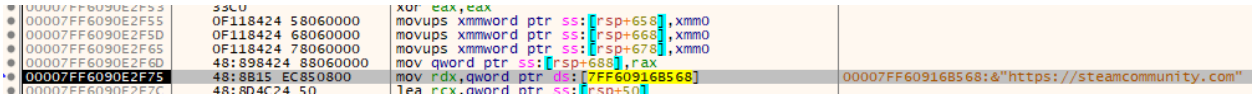


Runtime string and identifier resolution

After this step, it deobfuscates a string corresponding to a URL which, as can be observed, belongs to Steam.



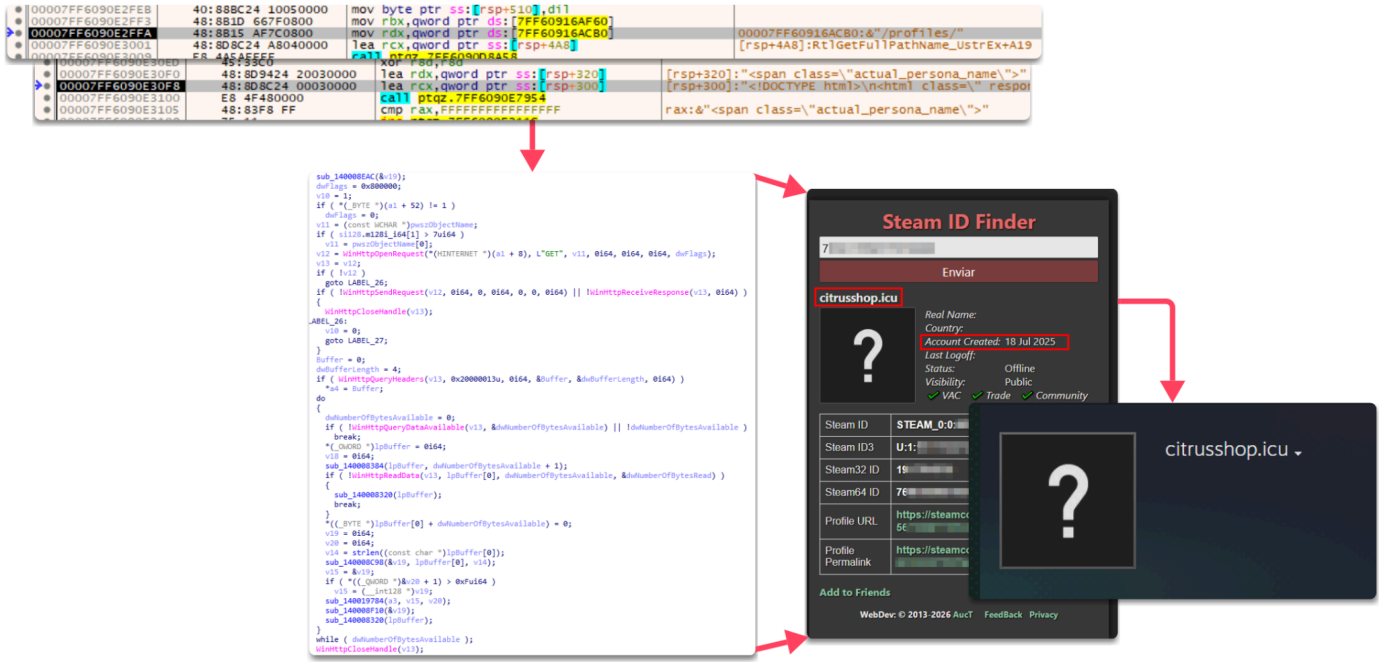
Deobfuscation of embedded URL



Resolved C2/target URL reference (Steam domain)

The threat actor has configured the retrieval of an intermediate C2 by querying Steam profiles created by the actor themselves. These profiles serve as a dynamic source of information that can be modified and rotated over time.

The malware sends a request to check the profile name based on an ID (SteamID), which returns the URL that the malware will internally verify. As shown below, this profile has been active since July 2025

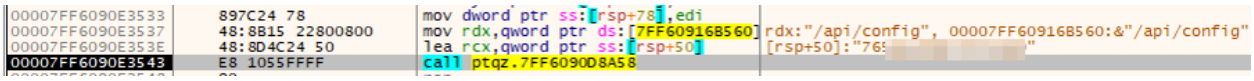


API-based access to backend infrastructure

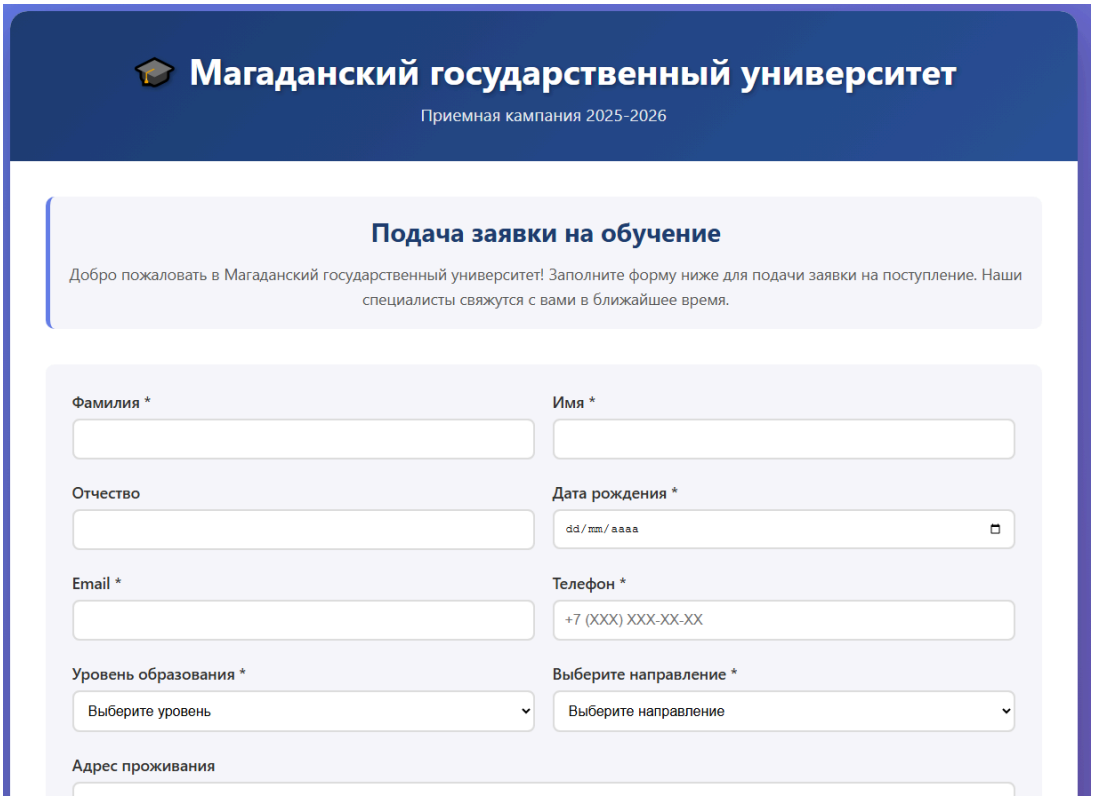
The domain leads to a fake form impersonating the University of Magadan. The malware accesses it through an API, while the visible page acts merely as a decoy portal intended to avoid blocking.

/ → Fake webpage

/api/client → intermediate C2



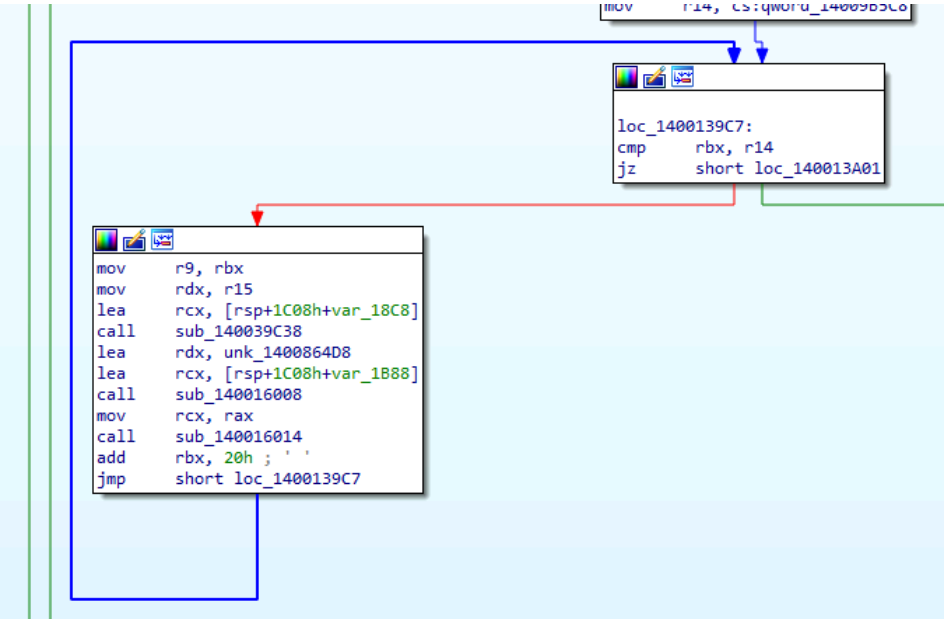
API endpoint usage for communication with intermediate C2



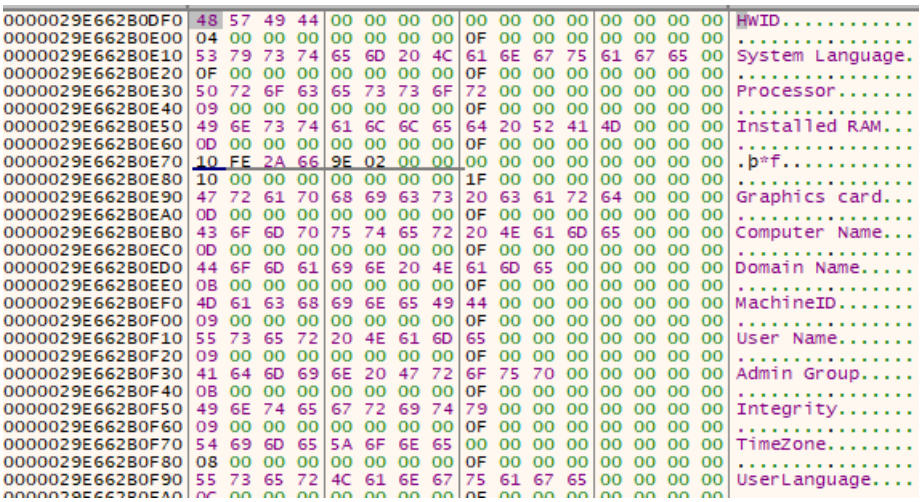
Fake university application form used as a decoy interface for data collection

Following this step, the malware attempts to gather key information from the affected system and proceeds to collect data such as:

- ➔ Log Date
- ➔ OS
- ➔ HWID
- ➔ Processor
- ➔ Installed RAM
- ➔ Graphics card
- ➔ Computer Name
- ➔ Domain Name
- ➔ MachineID
- ➔ User Name
- ➔ Admin Group
- ➔ Integrity
- ➔ TimeZone
- ➔ UserLanguage
- ➔ Wallpaper Hash
- ➔ Keyboard Language
- ➔ Screen Resolution



Control flow used to process and prepare collected data



Structured storage of system profiling information

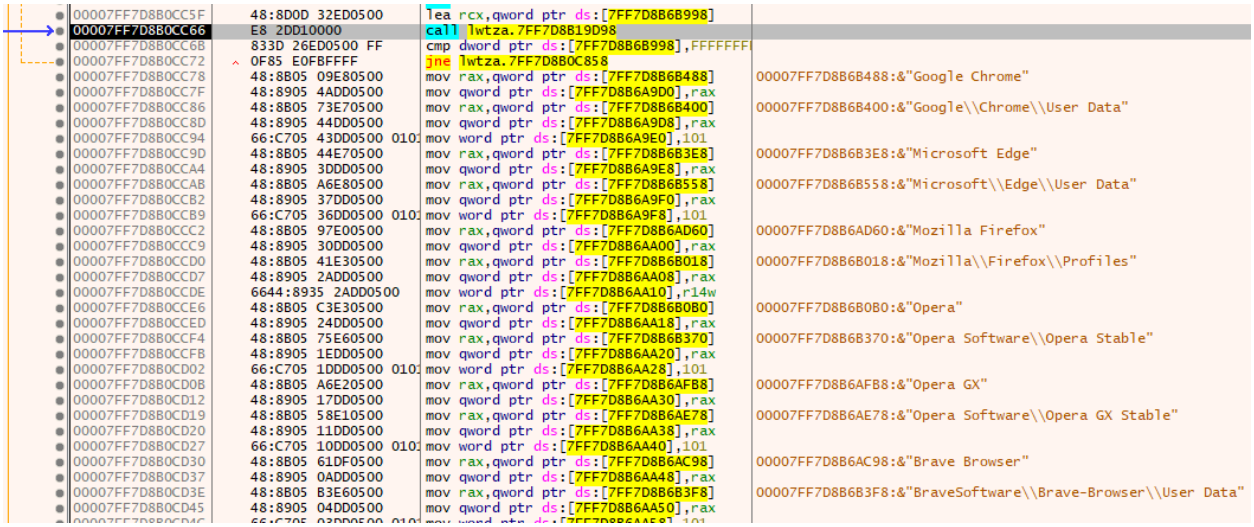
It iterates through the entire list, retrieving the information, typically from the Windows registry, encoding it in Base64 and maintaining a record of each element with a counter. The collected data is then converted into JSON, a practice that is consistently followed throughout the execution. The information is stored locally as userinfo <ID>.txt before being sent to the server.

Registry-based system profiling and packaging of collected data prior to exfiltration

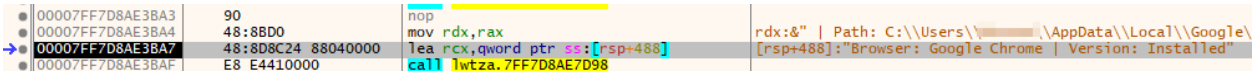
Depending on timing checks, the malware determines whether execution delays are occurring, something commonly observed during debugging. If such delays are detected, a specific mode is activated that does not affect the data collection process but changes how the data is transmitted.

Instead of sending the collected information in a single block, it is sent in smaller chunks. This defensive mechanism is common in stealers and is intended to prevent data loss. If the malware detects potential analysis activity or defensive systems interfering with execution, it may begin transmitting the collected information incrementally to ensure that at least part of the data reaches the operator before execution is interrupted.

Once the malware has obtained key system and user information, it proceeds to collect data from installed applications, primarily web browsers. It executes a routine that iterates through its internal list, defined previously in the builder configuration, and checks one by one whether each application is installed.



Enumeration of installed browsers and profile paths



Extraction of browser installation details and local data directories

It then extracts relevant data from the browsers or applications found on the system by accessing common files and registry locations. Typical paths such as "User Data", "Local", or "Profiles" are accessed, with the ultimate goal of extracting information such as:

- Credentials (Login Data)
- Cookies
- Web Data
- History
- Extensions

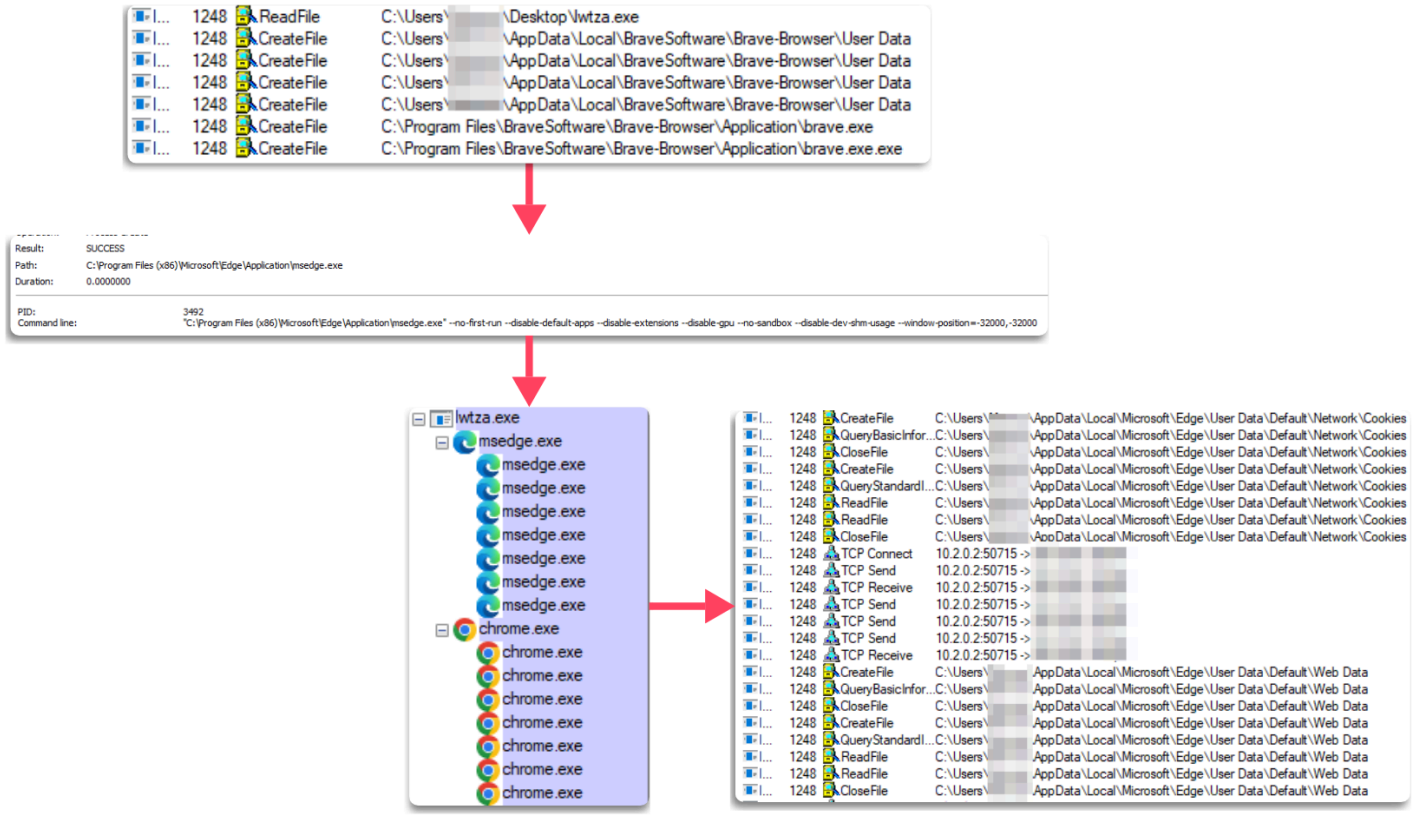
I...	1248	RegCloseKey	HKCU
I...	1248	RegQueryValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders\AppData
I...	1248	RegCloseKey	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
I...	1248	CreateFile	C:\Users\ \AppData\Roaming
I...	1248	QueryBasicInfor...	C:\Users\ \AppData\Roaming
I...	1248	CloseFile	C:\Users\ \AppData\Roaming
I...	1248	RegQueryKey	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1
I...	1248	RegOpenKey	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\KnownFolders
I...	1248	RegQueryKey	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1
I...	1248	RegQueryKey	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1
I...	1248	RegOpenKey	HKLM\SOFTWARE\Microsoft\AppModel\Lookaside\user\SOFTWARE\Microsoft\Windows\C
I...	1248	RegCloseKey	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1
I...	1248	RegOpenKey	HKCU
I...	1248	RegQueryKey	HKCU
I...	1248	RegOpenKey	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders

Registry queries used to locate user directories and AppData paths

Extraction of browser paths and profile information from the system

During this phase, the malware launches any browsers it has identified as installed. It first checks whether they are currently running (CreateToolhelp32Snapshot + Process32FirstW + Process32NextW). If they are not running, it forces their execution.

```
"C:\Program Files\Google\Chrome\Application\chrome.exe" --no-first-run --disable-default-apps
--disable-extensions --disable-gpu --no-sandbox --disable-dev-shm-usage
--window-position=-32000,-32000
```



Browser process spawning and access to SQLite databases for credential and cookie extraction

This allows the malware to read and unlock the SQLite databases used by browsers and decrypt possible cookies or stored passwords belonging to the victim, which could not be accessed if the browser was not running.

As with the device information, a browsers_profiles.txt file will also be created and sent to the intermediate C2:

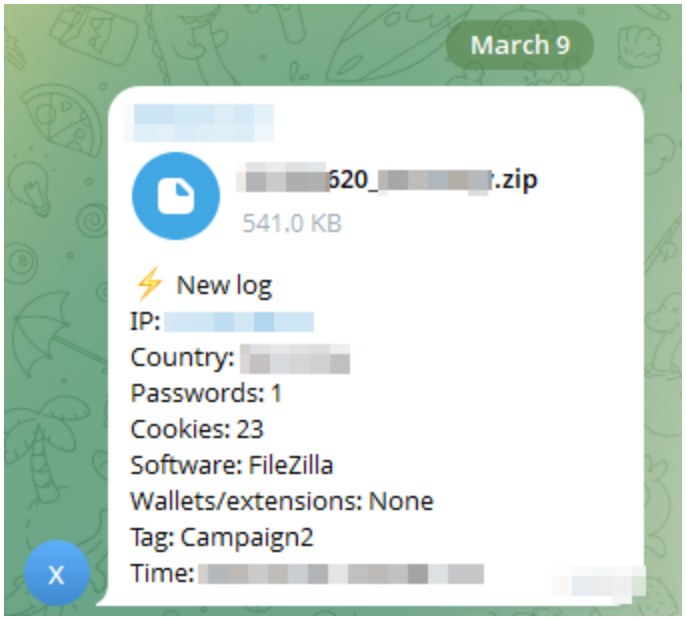
```
0x1dbdb7f460f   22    0browsers_profiles.txt
0x1dbdb81e59f   666   0{"type":"file_upload","file_name":"browsers_profiles.txt","file_category":"misc","data"
0x1dbdb820378   181   {"status":"success","message":"File saved successfully","ip":"[redacted]","path":"
0x1dbdb85645f   666   0{"type":"file_upload","file_name":"browsers_profiles.txt","file_category":"misc","data"
0x1dbdb87dae0   345   HTTP/1.1 200 OKServer: [redacted]GMT
0x1dbdb883d8f   327   erver: nginx/1.24.0 (Ubuntu)Date: [redacted]Content-Type: ap
```

The information is sent in JSON format as previously observed and is processed by the server.

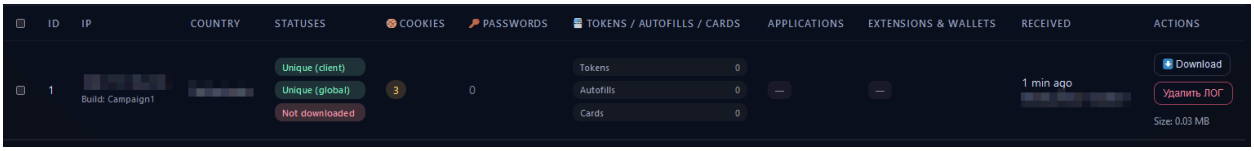
```JSON
{
 "type":"file_upload",
 "file_name":"browsers_profiles.txt",
 "file_category":"Browsers/Chrome/Default",
 "data":"encoded_data",
 "build_id":"Campaign1",
 "token":"xxxx",
 "session_id":"9..."
}
```

The destination server acts as a dispatcher, receiving the information, processing it and forwarding it to the corresponding operator panel based on the identifier assigned.

The operator will receive the information via Telegram, if enabled, as well as through the panel itself. Logs can therefore be accessed from multiple points, increasing resilience in terms of data retrieval and allowing operators to consult the data even if the panel becomes unavailable.



Data retrieval via Telegram

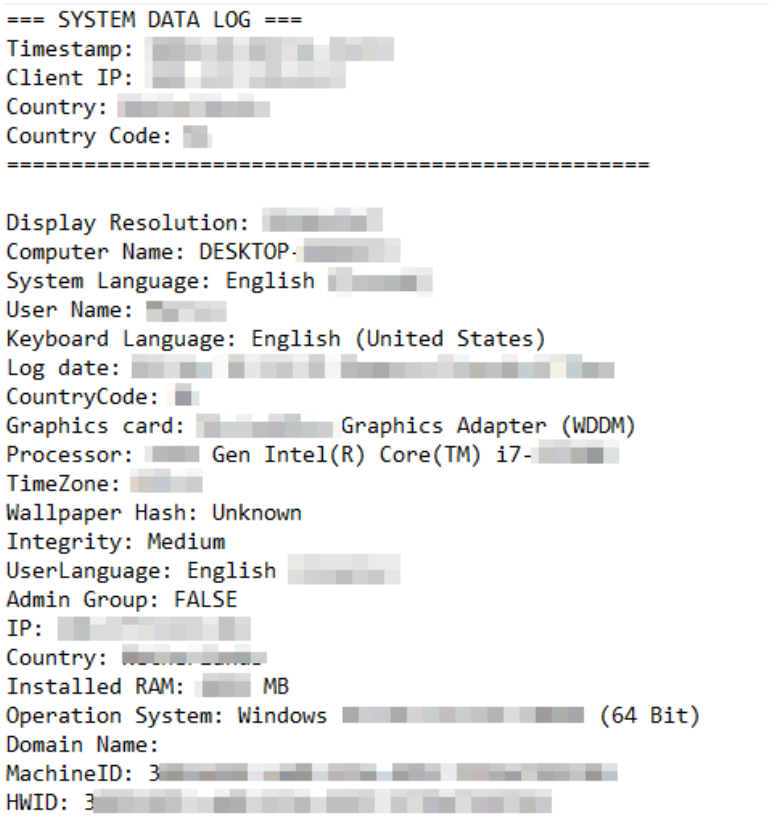


Data retrieval via Panel itself

What the operator ultimately obtains is sensitive information that can be used for different purposes, as mentioned previously in the MaaS section:

- Resale of information,
- Sale of data to an IAB,
- Personal use for access and pivoting.

Structured log output including system and user profiling data



Collected artifacts such as cookies, browsing history, and screenshots

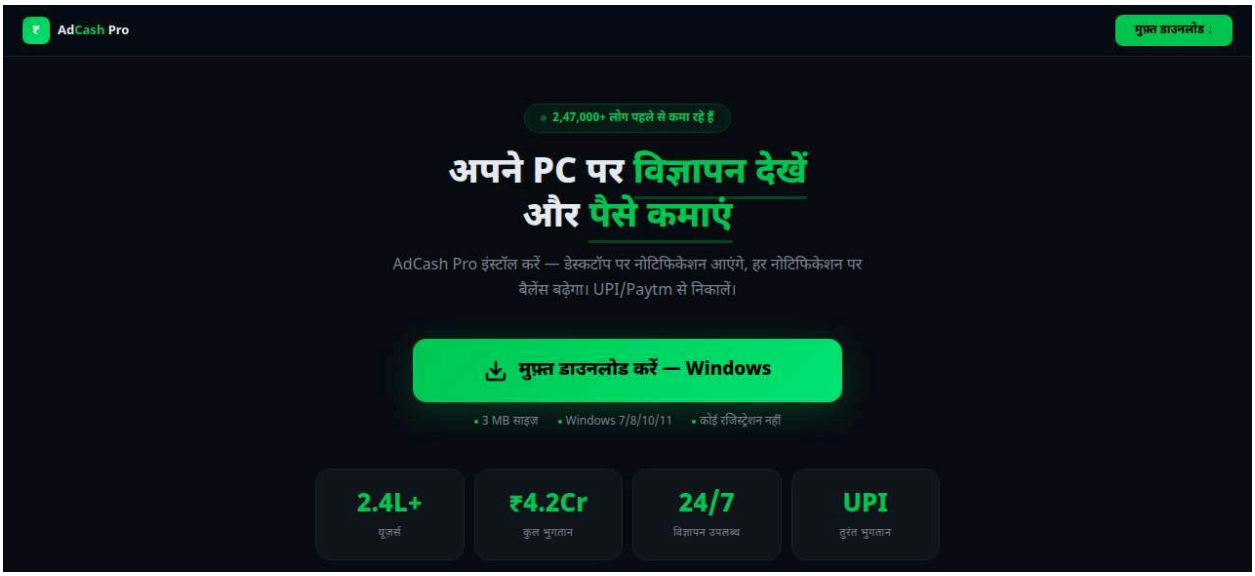
No persistence mechanisms were identified and the malware maintains a relatively low execution profile in order to avoid detection.

The quality of this stealer lies in the techniques it employs and in the amount of information it is capable of obtaining without being detected. There is a clear difference between low or mid level malware from this family and the more professional variants, as previously observed in terms of infrastructure and technical quality of their development. Nevertheless, they obtain sufficient information to have the same operational impact as more well known stealers, making their analysis and monitoring imperative.

Active Campaigns

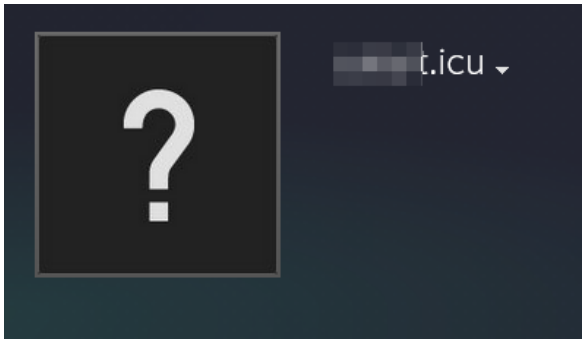
Based on execution patterns and understanding how the campaign operates, up to six active campaigns running simultaneously with VOID Stealer have been identified. This demonstrates that these low and mid tier stealers have affiliates and operators launching campaigns in parallel to target users worldwide.

Campaigns have been observed where attackers impersonate company portals. These sites are also used as apparently legitimate portals that function as intermediate C2 infrastructure, which the malware accesses through an API.



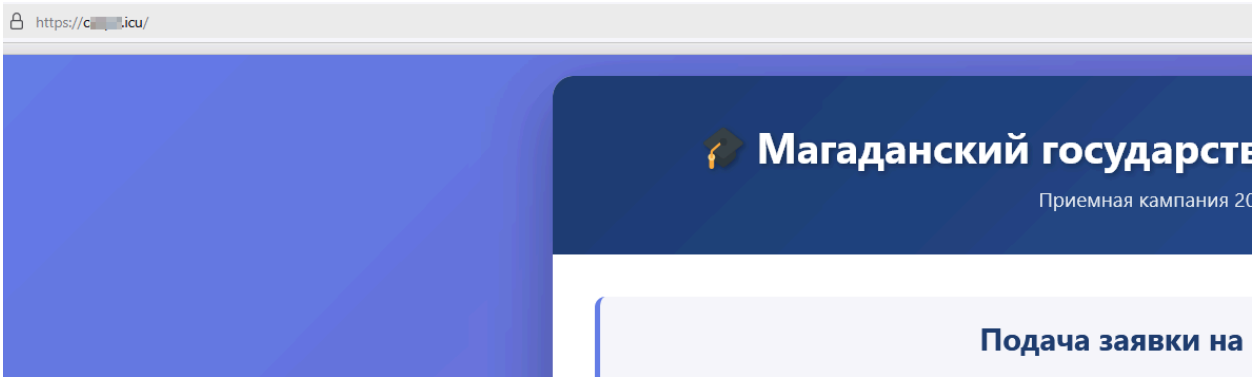
Malicious campaign page impersonating a legitimate service

The final result remains the same, the malware performs the request that connects to Steam in order to retrieve the intermediate C2, who will then be responsible for sharing the information with the panel and on Telegram.

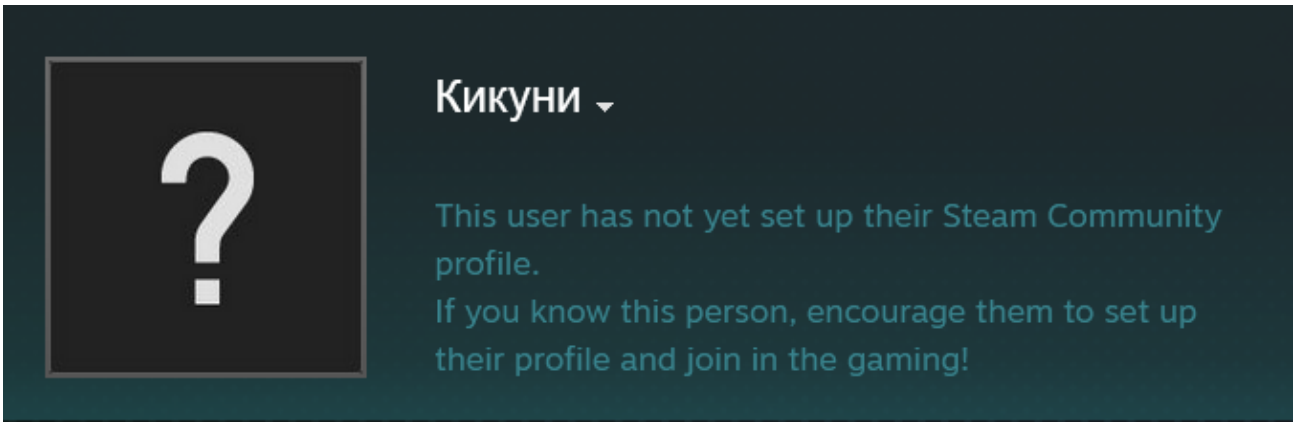


Intermediate C2 Steam endpoint used for data relay and coordination

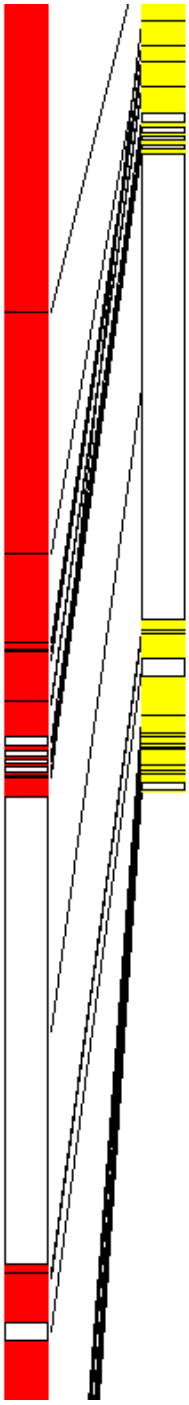
In other campaigns, similar forms have been preserved and can be found hosted on different domains. Despite the different infrastructure, they connect and behave in the same way as those already described in the technical details section:



Some of the Steam users previously used in the infrastructure have been deleted and are no longer accessible, indicating ongoing rotation within the infrastructure:



The binary responsible for triggering the execution behaves in the same way as previously analyzed samples. Nearly identical samples can be observed, consisting of executables that change depending on the builder configuration, as well as other packed samples whose behavior remains identical to the original specimen.



Multiple parallel campaigns sharing similar infrastructure with slight variations in binaries and delivery

Ultimately, even among lower tier stealers it is possible to identify multiple campaigns operating simultaneously. These campaigns present variations introduced by affiliates who purchase the malware, resulting in different websites and slightly modified binaries. However, in essence they behave in the same way and rely on the same underlying infrastructure.

Impact & Stolen Data

Once the stealer executes the exfiltration process on the compromised system, the tangible result is what the underground ecosystem refers to as a "log", a file containing the entirety of the data stolen from that specific victim. This log is essentially the unit of value in the market. Everything the malware collects, credentials, session cookies, browsing history, crypto wallet data, screenshots of the system, is packaged and ready to be sold, classified or directly exploited.

The post infection cycle follows a well defined chain that connects the moment of infection with underground markets and, from there, with the next wave of attacks.

After the Infection: Logs Market

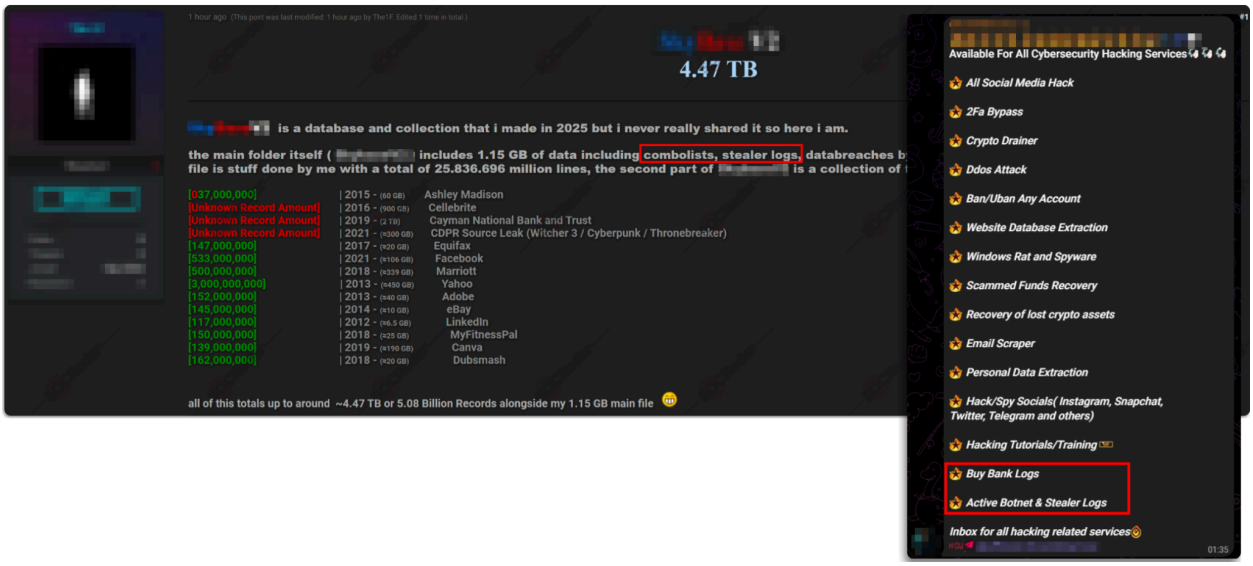
Once the operator, or the MaaS affiliate, has the logs in their possession, there are two primary monetization paths.

The first is direct sale on specialized underground marketplaces such as Russian Market, 2easy Market or similar platforms. On these markets, logs are indexed and offered with metadata that facilitates their purchase: the victim’s country, operating system, number of captured credentials, domains associated with the obtained accounts, or whether they include access to high value services such as online banking, corporate administration panels or enterprise SaaS platforms.

The price varies significantly depending on the perceived value:

- A log from a home user with few credentials may be worth less than one dollar,
- One that includes access to corporate accounts, VPNs or administration panels may exceed one hundred.

The second path is personal use or private exchange through Telegram channels or closed forums. This dynamic is precisely what distinguishes low and mid tier actors in the ecosystem. Lower level operators frequently do not have access to established marketplaces, or they simply lack the ability to correctly assess and classify the data they have obtained. As a result, they end up selling it in bulk at very low prices in Telegram groups, where more experienced actors purchase it, filter it, classify it and resell it with a higher margin.

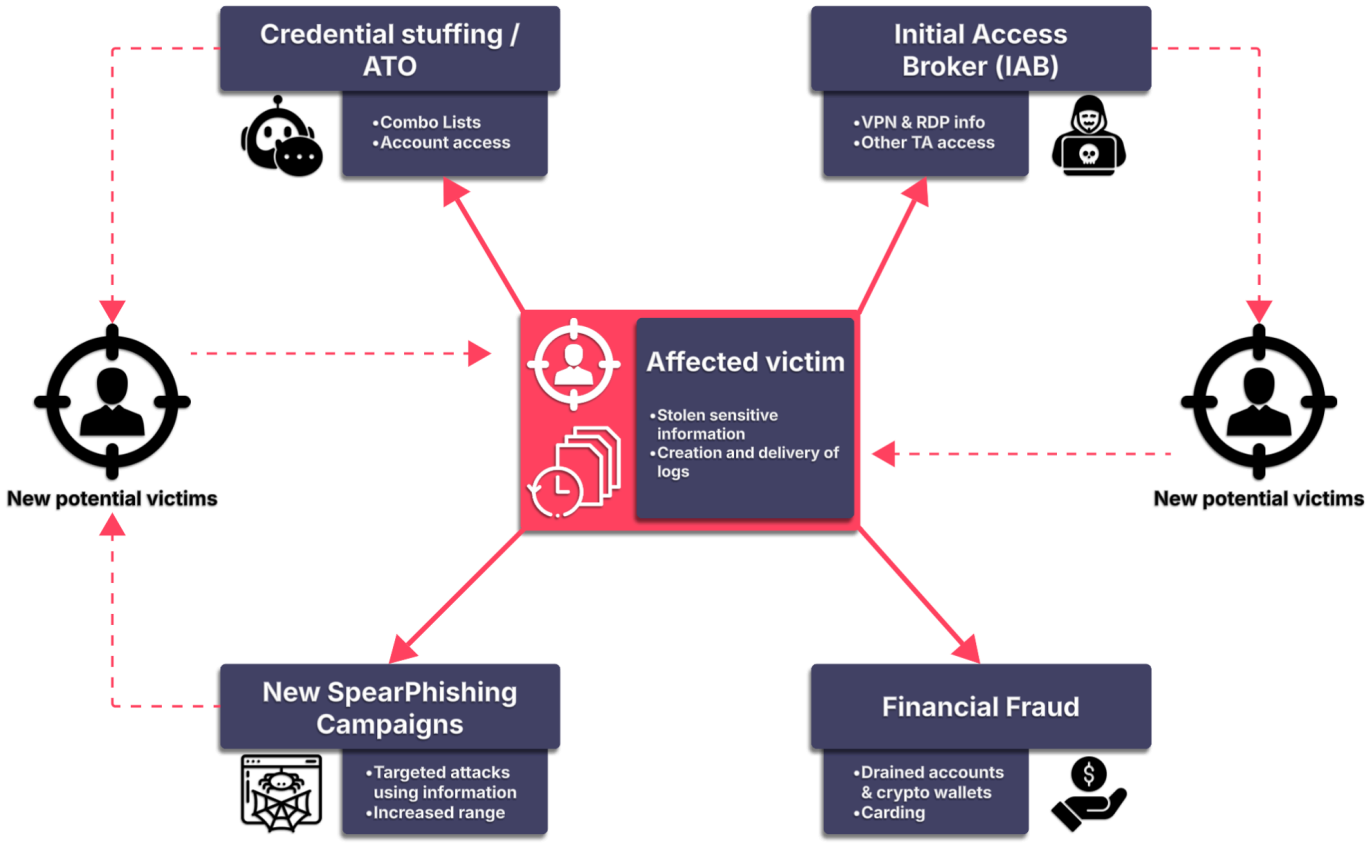


Stealer Logs & Combolists sales in Telegram and Underground Forums

Fueling Further Attacks

The log market is not an end in itself but rather part of a cycle in which the infrastructure feeds the next generation of attacks. Data obtained through stealers, once circulating within the underground ecosystem, becomes raw material for a chain of secondary threats with enormous impact. This multiplier effect is precisely what makes stealers, even low level ones, a threat that cannot be ignored.

Each exfiltrated log can be exploited simultaneously and in complementary ways by different actors within the ecosystem, generating an impact that far exceeds what the original operator could achieve alone.



Credential Stuffing & Account Takeover (ATO)

- **Description**: Stolen credentials are integrated into combo lists that feed automated credential stuffing tools. These tools test username and password combinations at scale across hundreds of services in parallel, exploiting the widespread reuse of passwords among users.
- **Result/Impact:** Takeover of legitimate accounts ranging from streaming services and online banking to corporate email accounts or administrative access panels. In enterprise environments, a single successful ATO can become the entry point to an entire network.

Initial Access Brokering (IAB)

- **Description**: Initial Access Brokers actively filter logs available on the market looking for high value credentials such as corporate VPN access, RDP access, administration panels or cloud environment keys such as AWS or Azure. Once a valid access is identified, the broker verifies it, packages it and resells it to more advanced TAs.
- **Result/Impact**: The obtained access eventually reaches ransomware groups or espionage actors who use it as the entry point for much larger scale attacks. This is the link that directly connects a low tier operator, who sold a log for a few dollars on Telegram, with a large scale ransomware incident. The operator was never aware of the real value of what they had obtained.

Spear Phishing & Targeted Attacks

- Description: Stealer logs do not only contain credentials, they contain context. Browsing history, captured documents, accessed emails or files present on the victim's desktop provide adversaries with operational intelligence that allows them to construct highly personalized phishing pretexts. The attacker knows which platforms the victim uses, which projects they are working on and which people they communicate with frequently.
- Result/Impact: Spear phishing campaigns that are extremely difficult to detect, targeting both the original victim and their professional or personal environment. The quality of the pretext, fueled by real information, exponentially increases the success rate compared to generic phishing, and can escalate to the compromise of third parties such as colleagues, clients or suppliers who were never the original target.

Financial Fraud & Crypto

- Description: Credit card data, which in this case VOID did not collect, and financial platform credentials are funneled to actors specialized in fraud and carding. Cryptocurrency wallet seed phrases, present in the panel configuration, and active exchange session cookies are particularly sought after due to how quickly they can be exploited before the victim detects the compromise.
- Result/Impact: Direct and often irreversible financial losses. In the case of cryptocurrencies, the nature of blockchain transactions makes fraudulent transfers practically unrecoverable. For the stealers analyzed in this report, the capture of crypto wallets is among the most actively developed features precisely for this reason.

The most relevant aspect of this ecosystem is its self sustaining and self referential nature. The profits obtained from selling logs finance the acquisition of new stealer licenses or the development of improved variants. Buyers of logs use them in campaigns that generate new victims. The personal information extracted from those logs enables phishing campaigns that infect a new generation of victims with the same malware. The cycle closes and begins again.

For the low and mid tier actors discussed in this report, this dynamic has an implication that should not be underestimated. Although their individual technical capability may be limited and their operations disorganized, they function as nodes within a much larger value chain. Their contribution, low cost logs sold without classification in Telegram channels, is eventually absorbed, classified and amplified by more sophisticated actors, transforming seemingly minor threats into vectors for high impact attacks.

The lack of attention these actors receive does not reflect their real impact. It reflects their visibility.

TTPs

Tactic	Technique	Procedure
TA0002 Execution	T1204.002 User Execution: Malicious File	Victim manually executes the stealer binary delivered via phishing or trojanized software
TA0005 Defense Evasion	T1027.007 Dynamic API Resolution	Dynamic NTDLL loading via GetModuleHandleA to resolve imports at runtime, avoiding static analysis and import table inspection
TA0005 Defense Evasion	T1055.015 Direct System Calls	NT to Zw syscall transition to bypass EDR userland hooks, directing calls to Zw functions to avoid monitored NT wrappers
TA0005 Defense Evasion	T1480.002 Mutual Exclusion	Mutex creation and single-instance enforcement to abort execution in sandbox or analysis environments
TA0005 Defense Evasion	T1027 Obfuscated Files or Information	Configuration file encrypted with XOR prior to runtime decryption, and the use of Base64 at different stages
TA0007 Discovery	T1082 System Information Discovery	Extraction of OS version, architecture, hostname, hardware (CPU, RAM, GPU) via registry queries
TA0007 Discovery	T1614 System Language Discovery	Collection of timezone, keyboard layout and regional settings for victim profiling
TA0007 Discovery	T1033 System Owner/User Discovery	Current username and system HWID collected as part of initial fingerprinting
TA0007 Discovery	T1518.001 Security Software Discovery	Identification of installed security software to adapt evasion strategy
TA0009 Collection	T1555.003 Credentials from Web Browsers	Direct SQLite database access to extract stored credentials and cookies from Chromium-based browsers and Firefox
TA0009 Collection	T1539 Steal Web Session Cookie	Session cookies harvested from all configured browsers including active session tokens
TA0009 Collection	T1176 Browser Extensions	Extraction of crypto wallet extension storage (MetaMask, Phantom, Coinbase Wallet and ~50 additional extensions)
TA0009 Collection	T1005 Data from Local System	Desktop wallet files collected from known paths (Exodus, Atomic, Electrum, Zcash, Ethereum keystore)
TA0009 Collection	T1113 Screen Capture	Optional webcam capture enabled via operator builder configuration
TA0009 Collection	T1125 Video Capture	Webcam image capture from compromised system when enabled by operator

TA0009 Collection	T1552.001 Credentials in Files	File grabber targeting seed phrases, .pfx and .key files in Desktop and Documents with depth 2 and 25MB max size
TA0009 Collection	T1185 Browser Session Hijacking	Browser process execution triggered to access session data only available at runtime
TA0010 Exfiltration	T1041 Exfiltration Over C2 Channel	All collected data uploaded to intermediate C2 which resolves campaign ID and forwards to operator panel and Telegram
TA0010 Exfiltration	T1020 Automated Exfiltration	Exfiltration executed automatically post-collection with no operator interaction required during runtime
TA0011 Command & Control	T1071.001 Application Layer Protocol: Web Protocols	Exfiltrated data transmitted to intermediate C2 server via HTTP in chunked JSON blocks
TA0011 Command & Control	T1573 Encrypted Channel	Data transmitted in structured JSON format; configuration protected via XOR encryption

IoCs

You can find the rest of the indicators associated with the Void Stealer campaign on the SOCRadar platform

Type	Data
MD5	0152914D34A70D620D711A5CF205D529
MD5	e62bad247c046f8f5a0c75ff52ec4bed
MD5	9a72d388dd1ff5c36a44c14c907bc3a0
MD5	79f4e0b394b2f0ca0573b983b84741f4
Domain	citrusshop.icu
SteamID	76561199877608270

Who is SOCRadar?

SOCRadar provides Extended Threat Intelligence (XTI) that combines: **"Cyber Threat Intelligence, Brand Protection, External Attack Surface Management, and Dark Web Radar Services."** SOCRadar provides the actionable and timely intelligence context you need to manage the risks in the transformation era.

Trusted by
21.000+ companies
in 150+ countries

Dark Web Monitoring: SOCRadar's fusion of its unique Dark Web recon technology with the human analyst eye further provides in-depth insights into financially-targeted APT groups and the threat landscape.

Protecting Customers' PII: Scan millions of data points on the surface, deep and Dark Web to accurately identify the leakage of your customers' Personally Identifiable Information (PII) in compliance with regulations.

Credit Card Monitoring: Enhance your fraud detection mechanisms with automation speed by identifying stolen credit card data on popular global black markets, carding forums, social channels, and chatters.

360-Degree Visibility: Achieve digital resilience by maintaining internet-facing digital asset inventory. Significantly accelerate this process by automated discovery, mapping, and continuous asset monitoring.

GET ACCESS FOR FREE

START YOUR FREE TRIAL

Discover SOCRadar's powerful tools and easy-to-use interface to enhance cyber threat intelligence efforts. Schedule a demo with our experts to see it in action, and we'll show you what SOCRadar can do.