Beyond The Gates: An Empirical Analysis of HTTP-Managed Password Stealers and Operators

Athanasios Avgetidis*[1], Omar Alrawi*[1], Kevin Valakuzhy[1], Charles Lever[1], Paul Burbage[2], Angelos D. Keromytis[1], Fabian Monrose[1], and Manos Antonakakis[1]

[1]Georgia Institute of Technology
[2]MalBeacon

Abstract

Password Stealers (*Stealers*) are commodity malware that specialize in credential theft. This work presents a large-scale longitudinal study of *Stealers* and their operators. Using a commercial dataset, we characterize the activity of over 4,586 distinct *Stealer* operators through their devices spanning 10 different *Stealer* families. Operators make heavy use of proxies, including traditional VPNs, residential proxies, mobile proxies, and the Tor network when managing their botnet. Our affiliation analysis unveils a stratified enterprise of cybercriminals for each service offering and we identify privileged operators using graph analysis. We find several *Stealer*-as-a-Service providers that lower the economical and technical barrier for many cybercriminals. We estimate that service providers benefit from high-profit margins (up to 98%) and a lower-bound profit estimate of $11,000 per month. We find high-profile targeting like the Social Security Administration, the U.S. House of Representatives, and the U.S. Senate. We share our findings with law enforcement and publish six months of the dataset, analysis artifact, and code.

1 Introduction

The impact of credential theft is inescapable. Verizon's 2020 Data Breach Investigations Report finds credential theft to account for over 80% of breaches [1]. More concerning, the stolen credentials are resold on the underground markets to cyber-criminal groups with unknown motives [2], [3]. In one instance, ransomware attacks leveraged stolen credentials from *Stealer* malware to obtain access to their target network and ransom critical services [4]. Given these factors, studying the *Stealer* ecosystem and their operators can help security researchers and law enforcement understand the nature, trends, and tactics of this rampant threat.

Prior works cover different facets of the underground economies by studying phishing [5], keyloggers [6], exploit kits [7], spam botnets [8], [9], and social network abuse [10]. Additionally, researchers have uncovered misconfigured dropzones [6], taken over botnets [11], and seized command-and-control (*C&C*) infrastructure [9] to further understand how malware operators conduct their business [12]. More active approaches even include hiring cybercriminals from underground forums to attack honey accounts so researchers can empirically document illicit services [2]. A prime commodity for malware operators in many of these diverse attacks is credential theft [13].

Prior studies provide fascinating insights into the underground markets, which motivates us to study the role of *Stealers*. However, the security community has not thoroughly investigated how cybercriminals manage, operate, and profit off of *Stealers*. On the other hand, technical blogs provide anecdotal insights about how *Stealers* operate, but they only focus on specific attack instances and lack deep analysis of their economies, service offerings, and victim targeting. Understanding the nature and tactics of *Stealer* operators can aid researchers in developing better defenses. In addition, law enforcement can leverage the insights to prioritize their resources when pursuing *Stealer* operators [14]–[16].

In this work, we examine a unique dataset that tracks 10 distinct *Stealer* families and their operators. We partnered with *MalBeacon*, a threat intelligence company, to study the activities of *Stealer* operators that span 20 months (Apr 2019 - Dec 2020). Using this dataset and other sources, we seek to understand the trends, nature, tactics, and service offering revenue of *Stealers* and their operators. These insights can help law enforcement pursue cybercriminals more effectively by targeting the operator's tactics and revenue streams. In summary, we seek to answer the following research questions:

- **RQ1:** How do *Stealers* contribute to cybercrime?
- **RQ2:** How do *Stealers* operate on the Internet?
- **RQ3:** What are the nature and tactics of *Stealer* operators and their service offerings?

In answering these questions, we make the following contributions: i) analyze the source code of leaked *Stealer* kits and

*Authors contributed equally.

document their features and offerings as advertised in underground forums, ii) formulate and characterize the operator activities by implementing and evaluating a clustering algorithm that resolves unique entities of the operator's device for cookie churn, iii) conduct measurements of *Stealer* hosting and victims on the Internet, and vi) empirically investigate *Stealer* service offerings and estimate their profit margins.

Our analysis of the leaked *Stealers* source code shows that they offer a wide range of functionality, including DDoS, keylogging, dropper, reverse shell, and screenshot capabilities. Hosted *Stealer* services appear to require little upfront cost and can potentially offer a large return on investment from the resale of credentials. The hosting infrastructure require minimal resources and operators often abuse free infrastructure services like country code top-level domains (ccTLD) and cloud-fronting. Our estimates show that *Stealer* services enjoy profit margins between 81% and 98%. Moreover, a lower bound estimate shows that the highest netting service provider profits approximately $10,910.55 per month. Unfortunately, we find newly registered *Stealers* domains to appear on public blocklists on average 74 days after registration. The detection lag can allow operators time to exercise other malware capabilities (i.e., install ransomware [17]).

Although we find that the highest *Stealer* activities appear to originate from Nigeria, *Stealer* operators rely heavily on proxy networks to masquerade their real IP addresses. When profiling the operators, we find that operators use proxy services ranging from traditional VPNs to mobile and residential proxies, to Tor networks, where the mobile and residential proxies can be harder to identify. We also find operators have varying privileges and access forming a stratified organization for *Stealer* services. Our targeting analysis identifies sensitive government networks with potential *Stealer* infections, including the U.S. Social Security Administration, the U.S. House of Representatives, and the U.S. Senate. We have shared our findings with law enforcement, and we discuss the ethical considerations in Section 4. To foster reproducibility and transparency, our paper accompanies six months of the *Stealers* dataset and the implementation code [1].

2 Background: Stealers & Cybercrime

Stealers are specialized commodity malware that harvest credentials from infected hosts. *Stealers* utilize many attack vectors, including drive-by download, application repackaging, remote exploitation, social engineering, and phishing. However, security reports [18]–[20] show that business email compromise (BEC) attacks are the most popular infection vector. Upon infection, *Stealers* harvest the operating system (OS) information, the system's settings, the user's profile, and stored credentials. These credentials belong to applications and services, including websites (browser stored passwords), remote management tools (FTP clients), and messenger applications. Furthermore, *Stealers* can steal cryptocurrency, install keyloggers, exfiltrate files, and drop other malware. Note that *Stealers* target *stored* credentials while keyloggers *log* keystrokes, which may include credentials. In summary, *Stealers* specialize in credential theft but may overlap in their features with remote access tools (RATs), spyware, downloaders, worms, and ransomware.

Credential Theft Lifecycle. There are four phases in the credential theft lifecycle [21]. In the first phase, cybercriminals harvest credentials through various channels, including phishing [5], social engineering, data breaches, and *Stealers* (malware) [13]. In the second phase, cybercriminals sort credentials like email, social network, financial, and corporate accounts. In phase three, automated tools verify the credentials to ensure a high-quality batch. In phase four, the credentials are sold to other cybercriminals. The pricing for each type of credential varies from $1.50 up to $9.

Cybercriminal Roles. Within the *Stealer* enterprise, there are varying roles ranging from low to high technical competency. Figure 1 depicts this relationship. We identify three primary roles, namely developers, service providers, and operators [21]. Developers are the most technical and they are responsible for writing the *Stealer* code. The next tier are service providers who typically buy a license from developers to offer *Stealer* as a service. The service providers can be the developers themselves or other cybercriminals who may be less technical (non-developers). Developers are incentivized to sell licenses of their *Stealer* malware to increase their revenue and market share. Lastly, *Stealer* operators can be the developers, the service providers, or other cybercriminals. Less technical cybercriminals may use *Stealer*-as-a-service offering to participate in the credential theft ecosystem. Note that highly technical cybercriminals can assume all three roles, while less technical cybercriminals can assume only the operator role.

Table 1: A list of top password stealers found in our dataset.

Family	First Sold	Price	Leaked	Panels ($N = 5,295$)	Hosts ($N = 2,602$)
LokiBot [22]	2015	$80-$300	✓	3,613 (68.23%)	1,952 (75.01%)
Formbook [23]	2016	$29-$299		1,195 (16.62%)	285 (5.32%)
Amadey [24]	2018	$600	✓	56 (1.05%)	44 (1.70%)
Baldr [25]	2019	$100-$150		32 (0.6%)	32 (1.22%)
Blacknet [26]	2019	Open Source		12 (0.22%)	12 (0.46%)
AZORult [27]	2016	$100	✓	8 (0.15%)	8 (0.31%)
Neutrino [28]	2013	$200-$500	✓	9 (0.17%)	8 (0.31%)
Agent Tesla [29]	2014	$12-$69		5 (0.09%)	5 (0.19%)
Nexus [30]	2020	$100		5 (0.09%)	5 (0.19%)
KPOT [31]	2018	$85		2 (0.03%)	2 (0.08%)

Stealer Management Interface. *Stealers* have two main components, namely the bot program and the management interface. The management interface for *Stealers* can be implemented as a web or desktop application. This work focuses only on *Stealers* with web-based management interfaces. While there are other popular *Stealer* malware families

[1] https://github.com/Astrolavos/stealer-sec23

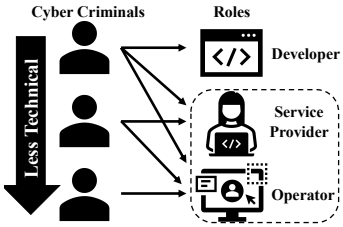

Figure 1: Cybercriminal roles in the *Stealer* ecosystem.

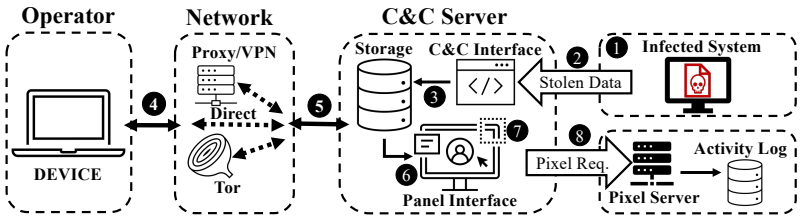

Figure 2: An overview of *Stealer* data collection.

with desktop management interface (i.e., Redline [32]), they are out of scope. Table 1 documents the *Stealer* family name, first advertised/sold year, the offering price, leaked source code, and counts found in the *Stealers* dataset. We manually analyze the source code for the open-source and leaked *Stealer* kits. We assess the technical barrier to entry and the capabilities of *Stealers* through the installation method, encryption functions, panel authentication, and malware control. All of the analyzed panels are built with PHP, HTML, and JavaScript, and their core functionality focuses on credential theft. The panels use SQL-based databases for storage and they are storing information about the bots and stolen data. Table 9 in Appendix A summarizes the source code analysis for the leaked *Stealer* panels. Panel setup can be manual, scripted, and guided, which vary in technical difficulty.

For authentication, *AZORult* only requires a password for login, whereas *LokiBot* checks the username, password, user-agent, and captcha. *LokiBot* uses captcha and randomizes the admin panel login path to make it harder to find. *BlackNet* allows users to enable 2FA using Google Authenticator as an additional layer of security. *Neutrino* bans IP addresses that attempt to enumerate files on the *C&C*. *Stealers* use varying degrees of defense to hide from internet scanners. *Stealers* provide numerous functions (bot commands) including DDoS, DNS spoofing, download or load executable (load/drop), shell command, open a browser and visit a page (visit page), screenshot, message (msg) victim, and keylogger. From Table 1, we can see offerings as low as $12 per week, which lowers the technical and financial barrier required for any cybercriminal to participate.

Takeaway-1: *Stealers* contribute to the credential harvesting phase. *Stealers* have a mature and competitive market that lowers the financial and technical barrier and caters to a wide range of cybercriminals. Hosted *Stealer* services appear to require little upfront cost and can potentially offer a large return on investment from the resale of credentials.

3 Data and Methodology

In collaboration with *MalBeacon*, we had initially set out to answer our research questions and gain insights that can help researchers develop better defenses (detection and prevention)

Table 2: A list of data sources used in this study.

Dataset	Description	Source
Stealers	Stealer tracker	*MalBeacon*
Active DNS	Domain reg./resolution	ActiveDNS Project [33]
Passive DNS	Recursive and authority domain lookups	US ISP, Global Recursives, Nameserver Authority, TLD Authority
Threat Intelligence	Malware and domain intel.	URLScan [34], VirusTotal [35] IP Reg. [36], bot tracker [37]–[39] residential and mobile proxies [40], [41]

Table 3: *Stealer* dataset fields summary.

Field Name	Description	Unique
Timestamp	The time a tracking event was observed.	202,538
IP Address	IP address used by the operator to access the panel server.	21,812
User-Agent	User-agent string associated with the operator's device.	1,484
Cookie ID	A session identifier set by the tracker for the operator's browser.	5,552
Panel Web Address	The referrer field sent to the tracker.	27,823

and aid law enforcement to pursue cybercriminals more effectively (deterrence). Unfortunately, the *Stealer* dataset alone does not allow us to fully explore these questions; therefore, we must augment the dataset with external data sources. We rely on DNS and threat intelligence. The DNS dataset characterizes DNS records, volumetrics, and client resolutions. The threat intelligence datasets enrich, validate, and identify additional artifacts of malicious infrastructure. Table 2 summarizes our data sources.

Scope. Our work investigates the *harvesting phase* of the credential theft lifecycle. The resale and distribution of the credentials throughout the underground forums or other illicit markets are out of scope. Specifically, this work studies one harvesting channel, namely *Stealer* malware, their *Stealer* operators, and the service providers, which we highlight with a dotted box in Figure 1. Readers can refer to prior works [2], [5], [9], [12], [13], [18] on credential theft profits.

3.1 Data Sources

Stealers Dataset. *MalBeacon*, a threat intelligence company, provided us with access to their commercially available *Stealers* dataset. *MalBeacon* tracks many *Stealer* families, which are listed in Table 1. In our initial analysis, we noticed a skewness in the dataset that can potentially be attributed to the malware's (Lokibot, Formbook, AZORult) popularity in the wild [23], [27], [42], limitation of the data collection process, or a combination of both. *MalBeacon* uses a *proprietary* pixel-tracking technique, similar to email marketing, embedded into artificial credentials, documents, and other sensitive information that *Stealers* target. When the operator views the stolen information, the browser will request the embedded pixel from *MalBeacon*'s server and reveal information about their device (IP address, user-agent, etc.).

Figure 2 is an overview of how *MalBeacon* collects the *Stealer* dataset. Step ❶ the *Stealer* infects a system and sends stolen artificial data with the embedded pixel (❷) to the *C&C* server, which is committed to the backend storage (❸). Next, when the operators use their device (❹) to connect to the *C&C* server (❺), they log in to the management panel (❻) where the embedded pixel gets rendered (❼). Before the pixel can render, the operator's browser will connect to the pixel server (❽) to retrieve the pixel. The pixel server logs the HTTP request from the operator's browser into an activity log database and generates a unique random long-lived cookie ID that is sent back in the response header. Any subsequent requests by the operator would include the cookie ID, which enables tracking operators across different panels. Table 3 summarizes the dataset fields and their counts.

MalBeacon did not disclose the proprietary implementation details for their system, but we demonstrate how to collect a similar dataset using the approach found in Nachum et al. [43]. In brief, Nachum et al. modify stolen system artifacts by inserting an HTML image tag alongside the original in the following format: Original Value + Image Tag, i.e. "DESKTOP-UU1KCDG<img/src=//domain.tld/name.gif>." When the stolen artifacts are rendered in the HTTP panel interface (*C&C*), the operator's browser will callback to the image hosting server and the hosting server will log the IP address, user-agent, and HTTP headers. To test this hypothesis, we implemented the system found in Nachum et al. and tested five *Stealer* malware families (Amadey, Azorult, BlackNet, LokiBot and Neutrino) for the following browsers: Chrome 96.0.4664.45, Firefox 94.0.2, and Edge 95.0.1020.44. We collected the same fields (IP address, user-agent, HTTP header) by using a Windows 10 virtual machine and hooked system calls to modify values such as the IP address (Amadey, Neutrino), Computer Name (Azorult, BlackNet), Global Unique Identifier (Lokibot) and Bot Name (Neutrino).

Additional fields can be utilized to insert the pixel code, but we leave that for future work. We were able to induce a pixel callback and cookie ID persistence for all families and across all three browsers. When testing with private browsing, we observe the cookie IDs are cleared after each session. Our testing found that privacy features on modern browsers trim the full referrer field. Specifically, we observed that starting with Firefox 87 and Chrome 89 the path and the query string information of the referrer field are missing [44]. The privacy feature impacts future collection of similar dataset and limits our cookie merging and malware labeling methodology. However, in this work, the *Stealers* dataset was collected before the browser privacy change (March 2021).

DNS Datasets. We use the aDNS from the ActiveDNSProject [33]. The project resolves over 1,100 different zones and includes resolutions for Alexa's Top 1M and public blocklists. Each domain is resolved two times during a period of 24-hours. We use aDNS to investigate *Stealer* infrastructure by enumerating relationships between observed IPs and domains. Furthermore, we use three passive DNS (pDNS) datasets from a US-based internet service provider (ISP), geographically distributed local and global DNS resolvers, and an authoritative nameserver responsible for several zones and a top-level domain (TLD) authority. The pDNS datasets are anonymized to exclude any customer-related information. We use pDNS to amplify the coverage of the stealer domain resolutions and estimate potentially infected networks resolving the stealer domains. Combining these datasets we get global visibility from over 80 million internet-connected devices.

Threat Intelligence Datasets. We use eight threat intelligence sources, namely URLScan [34], VirusTotal [35], IP Registry [36], residential and mobile proxy dataset [40], [41], and botnet trackers [37]–[39]. URLScan implements a website scanning engine to analyze JavaScript, HTML, and embedded content to detect malicious code. VirusTotal (VT) is a threat-sharing platform used by hundreds of commercial companies and thousands of security researchers to share malicious indicators. IP Registry is an IP intelligence service that collects and correlates data from partner networks and public sources like BGP tables, regional internet registry databases, internet service provider data, geofeeds, and latency measurements. The data provides coverage for 99.9% of the IPv4 space but excludes loopback, link-local, multicast, private, site-local, and wildcard IPs. The botnet trackers use open source threat intelligence to track *C&C* servers. The residential and mobile proxy datasets are sourced from an academic study [40], [41] that includes 6.42M residential IPs collected between May 2017 and February 2018 and 8M mobile proxy IPs collected between April and August 2019.

3.2 Data Validation

The raw pixel server logs contain HTTP request records where each record has a timestamp, the source IP address, and the HTTP header. *MalBeacon* processes the HTTP headers into three fields, namely the user-agent (UA), cookie ID, and refer

field. The final dataset format is a JSON file that contains the fields in Table 3. Our initial analysis of the *Stealer* dataset aims at validating the dataset by inspecting the consistency of user-agents, the persistence of cookie IDs, the identification of *C&C* instances, and the labeling by malware families.

User-Agent Validation. To investigate if the UA strings are potentially spoofed, we analyze the number of unique browsers and operating systems per cookie ID. If UA spoofing was present, the browser and operating system of the UA per cookie ID would change. We found six (0.01%) cookie IDs with more than one unique browser and 25 (0.45%) cookie IDs with more than one operating system. Manual inspection of those records reveal six cookie IDs with multiple versions of the Windows OS, four cookie IDs with multiple versions of macOS, and 12 cookie IDs with other operating systems (Linux, Android, etc.), which suggests potential UA spoofing.

On the other hand, 99.55% cookie IDs have only one operating system and browser with 73.23% having only one browser version. The rest change their browser version but they are consistent with the release of browser updates. For example, we find 50% of the devices update their browser within 21 days or sooner and 75% update their browser version within 41 days or sooner. However, a set of records from Firefox have versions before the update release, which can indicate spoofing or beta/early testing. Those UAs were associated with 145 cookie IDs and 6,068 records. In total, the potentially spoofed UAs account for 6,243 (3.0%) records associated with 170 (3.0%) cookie IDs. We discard those records when we perform operator device measurements.

Table 4: Top 10 user agents and related statistics.

OS	Browser	Cookie IDs	C&C	Update (Days)
Windows 7	Chrome 75.0.3770.100	116	119	22.50
Windows 10	Chrome 79.0.3945.130	112	110	24.15
Windows 10	Firefox 68.0	112	140	36.32
Windows 10	Firefox 69.0	111	113	47.23
Windows 10	Chrome 75.0.3770.142	109	120	53.54
Windows 10	Chrome 75.0.3770.100	108	122	21.74
Windows 10	Chrome 73.0.3683.103	95	88	28.57
Windows 10	Chrome 74.0.3729.169	88	109	22.31
Windows 10	Firefox 70.0	82	96	24.95
Windows 7	Chrome 75.0.3770.142	80	72	17.46

Lastly, we analyzed the top 10 UAs found in the dataset and present the results in Table 4. We group by OS and browser and count the associated cookie IDs, *C&C*, and the average days between a browser update release and a UA change. The most popular OS is Windows and the most popular browsers are Chrome and Firefox. We found on average there are 1.25 cookie IDs associated per *C&C*, while 75% of the *C&C* instances are associated with a single cookie ID. Although, these statistics imply that the overwhelming majority of the UAs are not spoofed, an operator can still spoof the most popular UAs to masquerade their true device fingerprint. This is an artifact limitation that we can not verify from the dataset. Realistically, to spoof a popular UA, an operator must know the most popular UAs in use with a particular *C&C* panel.

Cookie ID Persistence. We refer to the ephemeral cookie IDs as *cookie churn*, where a device is assigned multiple cookie IDs over time because they are not persistent. We find the ratio of cookie IDs per *C&C* panel to be on average 1.59 with a median of one and a maximum of 67, which implies that cookie churn is present in a subset of the dataset. We address the *cookie churn* problem by applying a similar technique to the work of Dasgupta et al. [45]. Briefly, Dasgupta et al. address cookie churn for *user-modeling* and *reach-frequency* in the context of online advertisement. User-modeling refers to estimating how many users visit a particular site (users per *C&C* panel), whereas reach-frequency refers to how often an individual user visits a particular site. In our study, we focus on user-modeling to address the cookie churn problem.

We use the OS, browser, and panel URL as device profiles. In addition, we use two *cannot-link* constraints, namely cookie lifespan overlap and browser version. Cannot-link constraints are logical constraints that can disambiguate distinct but similar device profiles. For example, the cookie ID's lifespan interval (last seen - first seen) cannot overlap. If two device profiles use Windows 10 and the Chrome browser, but the lifespan of their cookie IDs overlap, then we assume that those two devices are distinct since they access the same *C&C* from similar devices but using different cookies. The browser version constraint merges cookie IDs if and only if the browser version in later records are greater than or equal to the browser versions in earlier records per *C&C* panel.

We design and implement Algorithm 1 to analyze and reconcile multiple cookie IDs belonging to the same device. The input takes a set of *C&C* panels and retrieves a set of devices that access the panels (line 2). A device is a tuple of UA string and cookie ID, where the UA is parsed for the OS, browser, and browser version. Once we have a set of devices (D), we group the records by the OS and browser and sort them by the first seen date (lines 3 and 4). For each group (g), we iterate through the cookie IDs and either allocate a new cluster (line 9) or merge on the profile features and cannot-link constraints (line 16). Since we lack the ground truth to evaluate the accuracy of Algorithm 1, we define an error metric called *ambiguous merge error* to quantify missed merges. Our merge policy coalesces cookie ID candidates with the earliest cluster (first seen) and therefore, the metric captures how many other clusters the candidate cookie ID could have merged with.

We calculate the *ambiguous merge error (AME)* using the following formula: $AME = \frac{|collision|}{|g_i.GetClusters()|}$. More specifically, we calculate the AME per group (g_i) since the merge error can only occur when the profile features and cannot-link constraints are met for more than one cluster per group. We found 872 groups with at least two cookie IDs. We skip groups with one cookie ID since they cannot be merged. Out of the 872,

Algorithm 1: Merge device's cookie IDs.

Input: A set of unique *C&C* (*C2*)
Result: Merged Cookie ID Clusters

```
Merged ← {}
D ← GetAssociatedDevices(C2)
G ← Group(D , by=[OS ,Browser])
for g in G.sortAsc(firstSeen) do
    for i=0 to g.size do
        if g_i in Merged
            continue
        Merged.addNewCluster(g_i)
        for j=i+1 to g.size do
            for c in Merged.GetClusters() do
                if g_j.lifespan not overlap c.lifespans
                and |g_j.C2 ∩ c.C2| ≥ 1
                and g_j.browser_ver ≥ c.browser_ver
                    MergeWithCluster(c , g_j)
return Merged
```

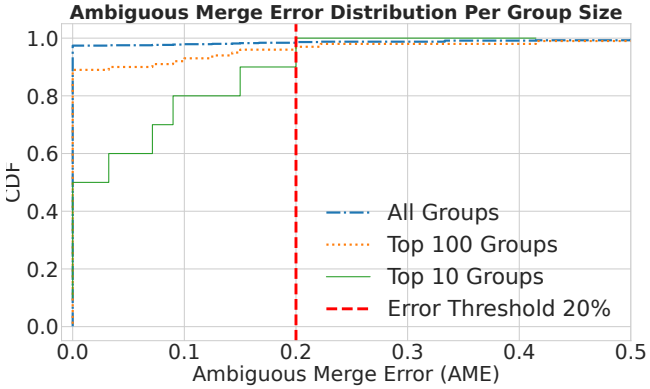

Figure 3: Distribution of *AME* per top largest group.

we detected merge misses in only 29 groups. Furthermore, 19 out of the 29 groups that have merge misses are in the top 100 largest groups. The largest *AME* value is 1.22, which indicates that the merge is ineffective, i.e merge error over 100%. This merge error belongs to the 89^{th} largest group, which had nine unique cookie IDs and 11 possible merge combinations (ambiguous merges).

We discard groups that have large *AME* values (more than 0.20) for the analysis. We summarize the distribution of *AME* for the largest top 10, 100, and all groups in Figure 3. We find eight out of the 10 largest groups have less than 0.1 *AME* rate. Additionally, five out of the 10 largest groups have 0.0 *AME* rate, which gives us confidence in the results since these groups have many cookie ID nodes. For example, group two has 68 unique cookie IDs and a merge collision count of 0. Beyond the *AME* metric, we manually inspected the top 100 groups to ensure that Algorithm 1 correctly coalesced cookie IDs and accounted for merge misses.

C&C Instance Identification and Labeling. The *Stealer* dataset does not contain any malware family labels or panel instance distinction, which makes our analysis challenging. Identifying and labeling the panel instances is an important

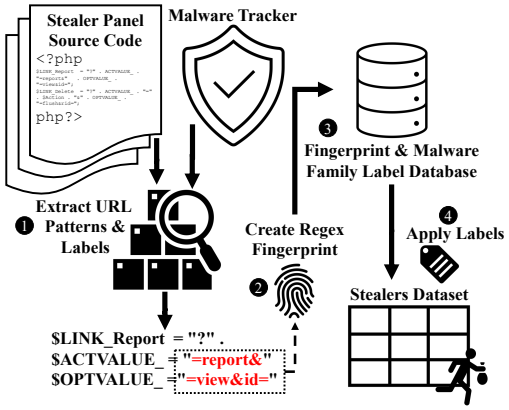

Figure 4: Panel signature generation and identification.

task that allows us to discern between different malware families and hosting infrastructure. We perform three labeling tasks, namely the identification of panel instances, panel malware families, and panel dynamic DNS domains. A single host can serve multiple panels. We define a panel instance (Π) by the domain or IP address (δ) and URL path (ρ). More formally, $\Pi = \{\delta, \rho\}$ where the δ can be a domain or an IP address and ρ is the URL path starting from the domain/IP to file name and extension (γ). For example, the following illustrates the components of a panel URL address:

$$\underbrace{\text{http://domain.tld}}_{\delta}\underbrace{\text{/path/file.ext}}_{\gamma}\underbrace{\text{?param=1}}_{\rho}$$

For records that do not contain URL paths, we label as unknown and exclude. Next, we assign a malware family label to the panel instances. We rely on the panel's URL components such as the path (ρ), file name and extension (γ), and parameters. We manually create *Stealer* family label signatures based on leaked source codes and panel tracker services [37]–[39]. Figure 4 presents our labeling process. In step one (❶), we extract URL patterns and labels from our source code and panel trackers. Next (❷), we use the strings and their order to generate a fingerprint for each *Stealer* family. In step three (❸), we store the signatures and the family labels in the database. Finally, in step four (❹), we label the panel instances based on the derived signatures. The signatures are in the form of regular expressions. From the 202,538 records in the *Stealers* dataset, there are 15,237 (7.5%) records associated with 357 (6.7%) panel instances with unknown labels. We attempted to use the AV labels from the malware files associated with each panel instance; however, we found them to be unreliable and noisy [46]. For Effective Second-Level (E2L) Dynamic DNS domains (DDNS), we manually verify them to ensure there are no false positives and we use pDNS to identify domains with 50 or more subdomains.

3.3 Affiliation Modeling and Analysis

To identify business affiliations, we model the operator devices and *C&C* panel interactions as a bipartite undirected graph $G(V,E)$ and perform link analysis. We create a vertex for each operator device (D_i) and panel (Π_j), i.e. $D,\Pi \in V$. We construct edges ($e \in E$) between vertices for each record in the *Stealer* dataset. Next, we extract connected components (subgraphs) from the global graph. For each connected component, we calculate the operator device (D_i) centrality in the subgraph using *eigencentrality*.

Eigencentrality measures the influence of a node in a graph. Intuitively, eigencentrality value is calculated based on connections to other high-scoring nodes. Since the bipartite graph has only edges between different node types (operator and panel node), the operator device node's eigencentrality will be calculated based on the collective scores of all neighboring panel nodes. An operator will have a relatively larger eigencentrality value (influence) if they are associated with more panel nodes in a connected component. We calculate the eigencentrality using the adjacency matrix of a graph $A = (a_{i,j})$, such that the eigencentrality x_i of node i is:

$$x_i = \frac{1}{\lambda}\sum_k a_{k,i}x_k$$

where $\lambda \neq 0$ is a constant. We calculate λ from the largest eigenvalue associated with the eigenvector of the adjacency matrix A, such that

$$\lambda x = Ax$$

Lastly, we treat each connected component as a potential *Stealer* service provider and the most influential operator devices (highest eigencentrality) are most likely the service administrators. We base this assumption on the conjuncture that the influential operator device nodes have privileged access to many panels, but the service customers do not have the same access. In summary, the bipartite connected component and eigencentrality are meant to identify service providers, associated infrastructure, and customers of the service.

4 Ethical and Legal Considerations

We take our ethical and legal responsibility seriously and ensure our study does not violate widely accepted norms. Our institute reviewed our request for an IRB and concluded that we do not require an IRB review. We also presented our study to the institute's Office of Cybersecurity for compliance and they did not have any concerns. This study uses data collected by *MalBeacon*, which is a US-based commercial company that operates and adheres to the Computer Fraud and Abuse Act (CFAA). The collection technique *does not* actively scan, exploit, or social engineer the malware operators in any way, and an external legal review committee reviewed *MalBeacon* 's tracking method and deemed it compliant with the Computer Fraud and Abuse Act (CFAA) and the Directive on attacks against information systems. The approach relies on honey tokens that are used in many studies [47]–[53] dating back to 2004. Moreover, our analysis of the dataset follows the precedence of prior works that study similar malware operator activities [6], [11], [54].

Research of criminal activity often involves deception or clandestine research activity [55], [56], so requests for waivers of both informed consent and post-hoc debriefing may be relatively common as compared with research studies of non-criminal activity. Support for such waivers is recommended when the research involves no more than minimal risk to the subjects, and the research could not be carried out without the waiver. For the *Stealers* dataset, deception is necessary to obtain data that characterizes the *Stealer* ecosystem. Such studies are considered permissible when (1) the research addresses important questions of public concern, (2) the research cannot be conducted if the subjects must provide consent, and (3) involving subjects in the research without their permission does not significantly compromise their autonomy. This study meets all three criteria and the scope follows well-established Menlo guidelines. Furthermore, our study is an analysis of a commercial dataset (passive observations) and does not directly implicate any malware operators or cause direct harm.

Finally, the data does not contain any personally identifiable information (PII). The IP address can be considered as PII with additional auxiliary data, but not by itself. From a law-enforcement perspective, an IP address can be subpoenaed from the ISP to get PII information about the person leasing the IP address at a given time. We do not have legal authority or access to auxiliary information to identify individuals. Despite those well-established guidelines on deceptive studies and issues regarding PII, we note that computer security research is more like behavioral research in the sense that the risks typically are not physical, and they can be difficult to quantify. Although evidence indicates that harm resulting from deceptive experiments is minimal and transient, it is still incumbent upon us to identify and minimize potential harm. We reiterate that we take our responsibility seriously and ensure our study does not violate the ethical norms.

5 Analysis Results

To answer our second research question (RQ2), we study how *Stealers* use internet infrastructure and analyze how *Stealer* operators administer their botnets by characterizing their devices, networks, and activities.

5.1 Stealers on the Internet

Our analysis of the *Stealers* public code shows that *Stealers* require minimal hosting infrastructure. We further seek to characterize *Stealer* hosting on the internet. More specifically, we characterize the domains and hosting networks of *Stealers*, quantify the detection delay between infrastructure

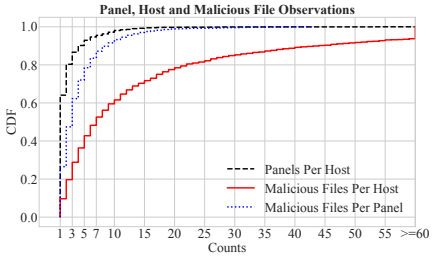

Figure 5: Distribution of panels and associated malware.

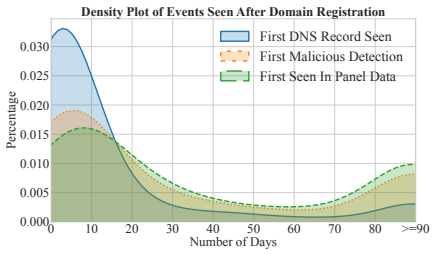

Figure 6: Distribution of domain events and detection.

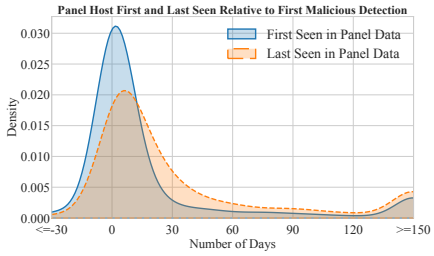

Figure 7: Distribution of the time delta for events and detection.

setup and blocklist detection, and assess the potential infections *indirectly* through the DNS dataset.

Internet Infrastructure. The *Stealers* dataset contains 2,187 registered domains, out of which 78 are DDNS and web hosting domains, and 281 panel hosting IP addresses for a total of 2,468 unique panel servers (hosts). This count excludes the two bogon panel IP addresses and three popular non-malicious domains in the Alexa top 100K [57]. Table 5 summarizes the top 10 top-level domains (TLD) count for effective second-level domains (E2LD)s of the *C&C* panels. For the panel domains, we find 41.4% use the COM TLD followed by 19.0% that use free country code domains (ccTLDs) like TK, ML, CF, and GA. Free ccTLDs are known to be heavily abused by malware [58]. The right side of Table 5 summarizes the top ten network names for the *C&C* panels, which account for 70% of the hosts. About 30.9% use US-based hosting (Cloudflare, Namecheap and Unified Layer), 15.8% use Russian-based hosting (Reg.ru, SelecTel, Mail.Ru, The First and IHOR-AS), and 12.2% use Chinese-based hosting (Alibaba cloud and Tencent).

In Figure 5, we present the distribution of panels and associated malware files per host and per panel, respectively. Note, that we differentiate between the host and the panel since a host can serve multiple panel instances. We observe that 64% of the hosts serve a single panel, 26% of the hosts serve between two and four panels, and 9.8% of the hosts serve five or more panels; the largest host has 71 panel instances. We find 61.5% of the hosts have 10 or fewer malicious files associated with them. The number of malware files per panel and host has a maximum value of 43 and 249, respectively.

Table 5: Top 10 TLDs and hosting networks for panel hosting server domains.

TLD	Domain (%)	Type	Reg. Cost	Network	Domain (%)
COM	874 (41.5%)	Commercial	$8.38	CLOUDFLARENET	308 (14.1%)
GA	107 (5.0%)	Country Code	$0	NAMECHEAP-NET	263 (12.0%)
XYZ	105 (4.9%)	General	$0.99	CNNIC-ALIBABA-US-NET-AP	197 (9.0%)
ML	97 (4.6%)	Country Code	$0	UNIFIEDLAYER-AS-1	105 (4.8%)
INFO	94 (4.4%)	Information	$2.99	SELECTEL OOO	86 (3.9%)
TK	79 (3.7%)	Country Code	$0	AS-REGRU	79 (3.6%)
ICU	73 (3.5%)	Business	$1.99	TENCENT-NET-AP-CN	71 (3.2%)
CF	66 (3.1%)	Country Code	$0	Mail.Ru LLC	64 (2.9%)
TOP	61 (2.9%)	General	$0.99	THEFIRST-AS JSC The First	61 (2.8%)
GQ	56 (2.6%)	Country Code	$0	IHOR-AS Ihor Hosting	57 (2.6%)

Detection of Stealer Hosting. Next, we want to assess if public blocklists detect *Stealer* infrastructure and if they do what is the time difference between the domain setup and detection. This will help us understand if current defenses against *Stealers* are effective and identify limitations that researchers can improve on. We find that 95% of the *Stealer* hosts appear on VT historical blocklist. Surprisingly, 123 hosts do not appear on public blocklists. We investigated the 123 hosts and did not find any notable difference from the detected domains. Figure 6 quantifies the detection timeline for 52.58% of the newly registered *Stealer* domains that had no prior DNS history (first-time registration). The plot shows the distribution of the events for new DNS records (solid blue line), malicious detection (dotted orange line), and the first operator activities in the *Stealers* dataset (dashed green line).

The average and median time for the first observed DNS record is 15 and two days, respectively. The pDNS data shows that DNS records are set within the first week after registration for 77% of the domains. We find the average and median time for detection is 74 and 11 days, respectively. Notably, the operators continue to access the *Stealer* hosts even after detection for an average of 74 days. On the other hand, 53.26% and 69.03% of the *Stealer* hosts stop operating 14 and 30 days after appearing on blocklists, respectively. For 43% and 28% of the newly registered panel domains, we find that they are detected within one week and after two months, respectively. For the remaining *Stealer* domains go undetected for an average of 64 days and a median of six days after their first DNS resolution. Within the undetected domains, 33% remain undetected for more than a month.

We observe, on average, 87 days between registration and first appearance in the *Stealers* dataset, with a median of 20 days. *MalBeacon* integrates with VT to share samples, which may correlate with the median time to detection (20 days). Additionally, Figure 7 shows the time window distribution for the first and last seen activity from the *Stealers* dataset centered around the first malicious detection of a panel host observed in VT. We find that almost 70% of the panel hosts appear in the *Stealers* dataset within seven days or less after their first detection. In summary, *Stealer* hosts are provisioned within two weeks and they appear on blocklists within 74 days

on average and operators continue to access the *Stealer* hosts for an average of 74 days after their detection.

Assessing Victim Targeting. To understand the impact of *Stealers*, we estimate the number of targeted victims. To get an accurate estimate, we would require direct access to the *C&C* server, which we do not have. Instead, we use the pDNS dataset to estimate the number of potential infections by analyzing the DNS resolutions. We quantify the number of DNS resolutions by network types and countries during the active time frame of each domain in the *Stealers* dataset. We define a network by the EDNS Client Subnet (ECS) [59], [60] found in the DNS resource records for clients resolving domains above the recursive, where the DNS recursive query the upper DNS hierarchy (i.e., TLDs and authoritative name servers). It is important to note that the results are associated with subnets and not IP addresses, which can underestimate the number of targeted victims. Moreover, the analysis is based on potential, not confirmed, infections.

We observe a total of 255,925 unique networks, but we were only able to label 167,989 (65.6%) networks. We present the results in Table 6. The table has four parts, namely the *Client Networks*, *Residential Networks*, *Business Networks*, and *Government Networks*. The *Client Networks* is a breakdown of all 167,989 labeled networks. The *Residential Networks* is a breakdown of the networks that belong to residential subnets grouped by country. The *Business Networks* is a breakdown of the networks labeled business subnets grouped by country. The *Government Networks* is a breakdown of the networks labeled government subnets grouped by country. For each network label, we show the number of networks (Count), lookup volume (Vol), days queried (Days), and rate of lookup volume (Vol/Day).

We find that 40.5% of the resolutions originate from *Hosting* networks. These networks appear to be associated with virtual private server (VPS) providers, cloud providers (i.e. AWS, OVH, Azure), and content delivery networks (CDNs), see Table 10 in Appendix A. The rDNS records show that VPS and cloud networks account for virtual private network (VPN) services. Moreover, a portion of cloud networks and most of the CDNs appear to be internet scanners or security tools. These observations align with prior works on malicious domain sinkhole analysis [61]. However, we believe many of the hosting networks are very unlikely to be infected clients.

We observe ISP/Telco and Residential networks as the second and third most popular networks, respectively. The residential networks are more likely to be victims since ISPs designate the space for home users. For the *Residential Networks*, we observe that Chinese clients make up 14.1% of the potential infections followed by Morocco (11.2%), India (8.6%), and the United States (7.7%). Notably, we find 207 government networks resolving *Stealer* domains. We took a closer look at the 113 U.S. government networks and found a mix of federal (24), state (32), and local (58) government networks. At the federal level, we found high-profile government networks like the U.S. Social Security Administration (4), the U.S. House of Representatives (2), and the U.S. Senate (2).

Investigating further, we found a total of 107 DNS responses for 27 different *Stealer* domains from August 2019 to November 2020. More specifically for the U.S. Senate network, we observe a total of 12 distinct resolutions for nine domains from January 2020 to July 2020. These DNS resolutions originate from what appears to be the DNS recursive servers for the U.S. Senate network. This suggests that there may be more infections because the DNS resolutions are typically cached. Nevertheless, the sensitivity of these government networks, including the U.S. Social Security Administration, demonstrate the far reach and impact of *Stealers*. Finally, the infection period for all 28 domains appears to extend over a year, giving operators ample time to execute other capabilities (keylogging, drop malware, reverse shell, etc.).

Takeaway-2: We find *Stealer* infrastructure to require minimal hosting resources and abuse services such as free ccTLDs and cloud-fronting. Moreover, public blocklists detect *Stealer* domains on average 74 days after initial registration with a median of 11 days. This detection gap gives *Stealers* ample time to infect and harvest credentials from a wide range of networks. Their long-lived activities may be problematic, as they allow operators time to exercise other malware capabilities (i.e., install ransomware [17]).

5.2 Characterization of Operators

The *Stealers* dataset provides a unique vantage point to characterize how *Stealer* operators manage their botnet using the *C&C* panels. We take a closer look at how operators interact with the *C&C* panels through their devices and shed light on their tactics.

Device and Network Characteristics. Characterization of the device and network association can inform researchers about common patterns used by cybercriminals. These characteristics can help build heuristic-based defenses that profile device and network properties to flag suspicious and unauthorized access. On average, operator devices access panels using 6.66 IP addresses that belong to 1.95 autonomous systems (ASNs). The largest number of IP addresses associated with an operator device is 230 and they belong to nine ASNs. Moreover, the standard deviation for operator device IP addresses is almost double the average (12.7). When looking at how operators access their *C&C* panels, we find, on average, operator devices access 1.62 unique panel instances, 1.51 unique domains, and manage 1.04 malware families. The operator device with the most panel instances accesses 57 unique panels hosted on 42 distinct domains. We took a closer look at this particular example and found that the 42 distinct domains use algorithmically generated domains (DGA).

After applying the cookie merging algorithm (Algorithm 1), we find operator devices to be associated with 1.17 cookie IDs on average. The operator device with the most cookies has

Table 6: Networks resolving stealer domains by country for residential, business, and government networks.

Client Networks		Residential Networks					Business Networks					Government Networks				
Type	Count (%)	Countries	Count (%)	Vol.	Days	Vol/Day	Countries	Count (%)	Vol.	Days	Vol/Day	Countries	Count (%)	Vol.	Days	Vol/Day
Hosting	67,958 (40.5)	China	4,187 (14.1)	607,282	473	1,283	United States	25,315 (92.8)	1,441,020	500	2,882	United States	113 (54.6)	40,161	328	122
ISP/Telco	37,463 (22.3)	Morocco	3,313 (11.2)	47,854	351	136	Vietnam	619 (2.2)	2,004,091	348	5,758	Canada	14 (6.7)	405	25	16
Residential	29,595 (17.6)	India	2,556 (8.6)	135,815	466	291	United Kingdom	309 (1.1)	1,652,777	420	3,935	China	8 (3.8)	604	139	4
Business	27,269 (16.1)	United States	2,293 (7.7)	195,714	481	406	S. Korea	152 (0.5)	18,798	276	68	Italy	6 (2.9)	265	60	4
Education	5,143 (3.0)	Iran	1,479 (5.0)	16,929	429	39	India	117 (0.4)	5,399	287	19	Indonesia	5 (2.4)	7	6	1
Government	207 (0.1)	Mexico	1,410 (4.7)	75,469	403	187	Nigeria	108 (0.4)	7,615	212	36	Israel	4 (1.9)	235	57	4
Health	188 (0.1)	Indonesia	1,360 (4.6)	48,557	352	137	China	69 (0.2)	182,895	361	506	India	4 (1.9)	4,264	80	53

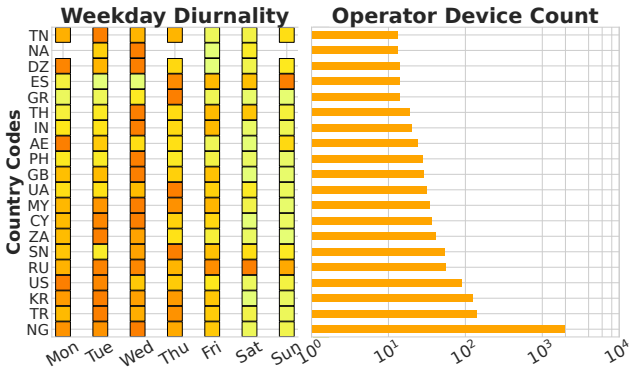

Figure 8: The diurnality for the top 20 countries of operator device activity (dark more active and light less active).

55 unique cookie IDs. This device used the same operating system, browser and browser version for over a month to access the same panel with 55 unique non-overlapping cookies, suggesting cookie churn.

In the entire *Stealers* dataset, there were 465 (10.14%) operator devices that have more than one cookie ID. We find on average 5.7 more IP address associations for these devices. Cookie merging (Algorithm 1) helped us build a complete profile for these operator devices and uncover related IP addresses and ASN associations that otherwise we would have missed. Cookie churn fragments access patterns and should be addressed to build a more complete device profile. Additionally, operator device profiles appear to be diverse and can distinguish between different operators.

Networks Access Patterns. We analyze the network types, the use of proxies, and the localized diurnal access times to investigate the access patterns. In total, operator networks originate from 135 different countries with different network classifications. The networks classifications include ISP (11.55%), ISP-Mobile (55.14%), and hosting networks (31.71%). Interestingly, we find over half of the operator networks are classified as ISP-Mobile. The bar graph in Figure 8 presents the top 20 countries for operator device networks. We find that most of ISP (80.32%) and ISP-Mobile (84.72%) networks are located in Nigeria. Revealingly, 99% of the internet broadband in Nigeria relies on mobile wireless connections [62]. Using the residential and mobile proxy dataset [40], [41], we intersect the timestamp and IP address of each operator device and find 882 (4.04%) mobile proxy records that match against the operator IP addresses. However, if we omit the timestamp field and only match against the IP, we find 1,785 (8.43%) and 5,667 (26.76%) matches for residential and mobile proxies, respectively.

Table 7: Top 10 countries of operator IP addresses and their proxy and tor networks.

Country	IPs	Mobile Proxy (%)	Residential Proxy (%)	Tor Exit Node (%)
Nigeria	11,375	4,326 (38.03%)	1,181 (10.38%)	0 (0%)
United States	1,936	161 (8.32%)	36 (1.86%)	15 (0.77%)
Great Britain	908	153 (16.85%)	65 (7.16%)	7 (0.77%)
South Korea	812	170 (20.93%)	14 (1.72%)	0 (0%)
Germany	496	40 (8.06%)	47 (9.47%)	10 (2.01%)
Netherlands	418	33 (7.90%)	31 (7.41%)	5 (1.20%)
Turkey	291	19 (6.52%)	16 (5.50%)	0 (0%)
Canada	279	23 (8.24%)	24 (8.60%)	3 (1.07%)
France	231	28 (12.12%)	21 (9.09%)	2 (0.86%)
Norway	222	4 (1.80%)	4 (1.80%)	0 (0%)

Furthermore, we analyze the number of Tor exit nodes associated with the operator networks and present the overlap per country in Table 7. Nigerian IP addresses make up about 53.73% of the operator networks and 42.55% of those were observed as proxy networks. Additionally, the top networks classified as hosting appear to be strongly associated with VPN services. For example, we find most hosting networks to be located in the US, Great Britain, Germany, and the Netherlands, and the top 3 ASNs are: AS9009 M247 Ltd, AS198605 AVAST Software s.r.o. and AS205016 HERN Labs belong to VPN services [63]–[65]. We crosschecked the hosting networks with IP intelligence feeds and found that IPRegistry [36] labels them as VPN networks. These findings suggest that *Stealer* operators make use of different proxy networks like residential, mobile, Tor, and traditional VPN services when accessing the management panel. These findings demonstrate that operator profiling can be involved and naively using the operator networks to attribute cybercriminals can be inaccurate.

Operator Device Diurnality. Diurnal analysis can provide another perspective into the nature of operator device access and can be used as an additional confluence signal for the geographical location. We quantify the access frequency for only *ISP-based* IP addresses that are not found on the proxy lists. The time zones for the diurnal analysis are *localized* to the geographical region associated with the operator's IP address. Figure 8 presents the diurnal access patterns for ISP-based (Mobile and Landline) operators. We present the

top 20 countries, which account for 95.60% of the ISP-based operator device IP addresses in the dataset, and make up 63.70% of the IP addresses of the entire dataset. The time zone localization shows higher activity on the weekdays than the weekends for most countries. For example, the Nigeria diurnal profiles have double the activity on the weekday in comparison with the weekends.

The results suggest that most operator devices are more active on weekdays regardless of the potential victim connections. Those diurnal activities can imply that operators manage *Stealer* as a full-time job as they are mostly connecting during weekdays. The higher activities observed on the weekend for some regions (Russia, Spain, Nambia) can suggest these operators use proxy networks and do not necessarily reside there. More importantly, these observations can provide higher confidence in the operator device profiles when combined with other signals (device fingerprint, network, and access profiles).

Takeaway-3: Operators use proxy services ranging from traditional VPNs to mobile and residential proxies, to Tor networks. In particular, the mobile and residential proxies can cause misdirection when characterizing operator profiles. We find that the cookie IDs are fairly persistent with the majority of the devices in the dataset, but for some operators, private browsing results in ephemeral cookie IDs. The diurnal analysis suggests that operators administer their botnet as a full-time job.

5.3 Operator Affiliations

We extend our analysis to understand operators' affiliations based on shared *C&C* panel access, i.e., distinct operator devices accessing the same *C&C* panel. Specifically, we apply the bipartite graph analysis from Section 3.3 and construct a global graph for the entire dataset. We extract connected components and study each component as an individual affiliation, which we define as an independent *Stealer* service provider.

Affiliations. The bipartite analysis found 2,449 connected components (clusters). Figure 9 shows the distribution for all nodes, operator device nodes, and panel nodes for the clusters. The cluster size ranges from two to 449 nodes. We find 98% of clusters have less than 15 nodes. The top 0.4% of clusters have 50 or more nodes. We summarize the top five largest clusters in Table 8. The Table presents the attributes for days seen, operator device nodes, and panel nodes (infrastructure). For example, the largest cluster has 285 total nodes, 127 operator device nodes, and 157 panel nodes that were observed over 689 days. The 127 operators are associated with 1,382 distinct IP addresses and 911 of the IP addresses are potential proxies. The 157 panels are associated with 92 domains and three *Stealer* families. Note that these affiliations are made up of several distinct operator devices and *C&C* panels. We find that the majority of the *Stealer* services are small and sparse; however, the top 1% appears to be more connected and active.

Influential Operators. Next, we examine the node degree to quantify the panel-to-operator ratio. Figure 10 presents the distribution of the node degree in the clusters. We observe a maximum of 57 distinct panel nodes connected to one operator node. On the other hand, we find 37 distinct operator nodes connected to a single panel node. Among all operators, we want to identify the most influential operator for each cluster by using *graph centrality analysis*. Figure 11 presents a boxplot for the top 25 largest clusters. Each boxplot represents one cluster sorted from largest (leftmost) to smallest (rightmost). We find most clusters have one or two outliers with high centrality values (>0.4). On the other hand, the majority of operators in each cluster have a centrality value of less than 0.4. This suggests that operators with high centrality values play an administrator role for the cluster (service provider). Furthermore, this suggest that influential operators may be the service providers and the remaining operators are customers. We base this assumption on the fact that an operator device with access to many *C&C* panels has more credentials than an operator device with access to a few *C&C* panels in the same cluster.

Takeaway-4: The affiliation analysis suggests that the largest 1% *Stealer* service providers account for most of the activities. Moreover, each service provider appears to have one or two influential operators that have more privileged access, which suggests those operators play an administrative role. Our analysis shows a stratified organization per cluster with different privileges, which supports our operator model role in Figure 1.

6 Analysis of Top Clusters: Services & Profits

Expanding on the top service providers from the previous section, we dive into the top five clusters and characterize their growth, operational cost, revenue, and potential infections.

6.1 Cluster Lifespan and Growth

Recall, that Table 8 presents the attributes for the five largest clusters. The Table presents the number of days seen for each cluster based on the activities in the *Stealers* dataset. The most active clusters are C1 followed by C2 and C3, respectively. We quantified the growth of each service provider by analyzing the number of new operators joining each cluster and found that, on average, one new operator joins the cluster per week. We visualize the growth in Figure 12. The growth is not uniform, and we do find some weeks with zero operators joining and some weeks with up to seven new operator devices joining. This signifies a consistent growth, in particular for C1, which suggests that it is the most stable service provider.

Next, we look at the operator access networks. Operators per cluster exhibit a similar access trend as the overall analysis

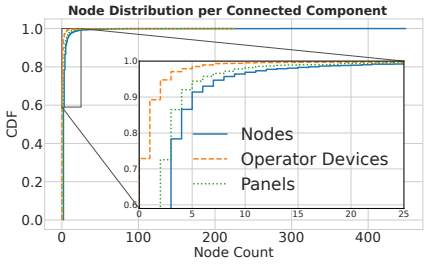

Figure 9: Distribution of the cluster size.

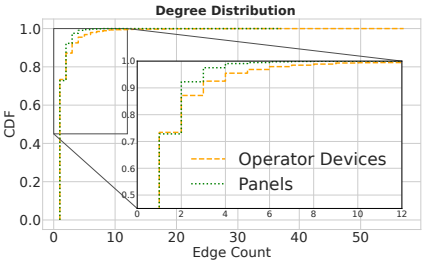

Figure 10: Distribution of graph edges.

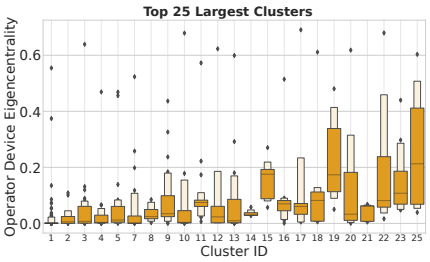

Figure 11: Distribution of centrality.

Table 8: Coarse estimates for *Stealer* services for the top five components. Estimates are in US Dollar (USD).

Name	Size	Days Seen	Operators			Infrastructure					Infected Networks		One-Off	Ongoing Monthly Estimates			
			Count	IPs	Proxy	Panels	Domains	DDNS	Hosts	ASNs	All	Residential	Cost	Hosting	Revenue	Profit	Margin
C1	285	689	127	1,382	911	157	92	3	155	34	14,247	6,795	$5,481.15	$923.45	$11,834	$10,910.55	92.2%
C2	84	468	15	99	69	68	38	1	88	76	3,931	1,076	$595.25	$199.88	$5,440	$5,240.12	96.33%
C3	72	418	37	346	257	34	19	0	35	11	1,997	1,051	$963.3	$37.45	$2,579	$2,541.55	98.55%
C4	68	332	24	167	91	43	22	0	158	47	29	4	$121.61	$638.05	$3,440	$2,801.95	81.45%
C5	57	415	26	227	139	30	21	0	65	25	23,013	8,153	$2,591.97	$88.94	$1,930	$1,841.06	95.39%

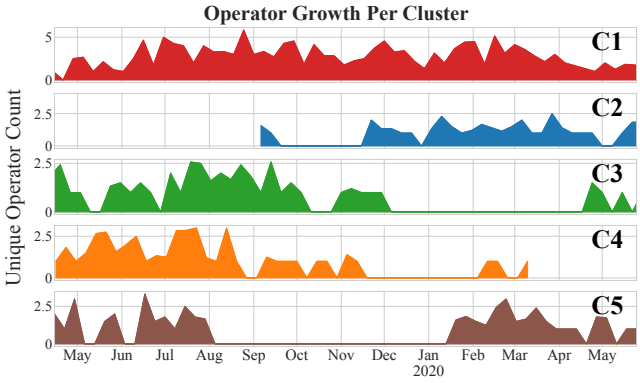

Figure 12: Operator growth per cluster over one year.

from the previous section, where operators rely on proxies as shown in Table 8. In particular, 66%, 70%, 72%, 54%, and 61% of observed operator IP addresses for clusters 1-5 are proxies, respectively. Interestingly, the influential operators in each cluster have distinct proxy access patterns. For example, in C1, the operator uses only US-based proxy networks, whereas in C2 the operator uses proxies in three different countries (US, NL, UK). On the other hand, for C3 the operator uses mobile proxies based in Nigeria, and C4 and C5 operators use network proxies based in Switzerland. Notably, for C2, the operator's proxy IP addresses appear to be static because they are reused by the operator for 200 days.

6.2 Analysis of Operational Cost and Impact

The growth results for the top clusters motivated us to investigate each cluster's cost, revenue, and potential victims. Recall, the scope of our revenue estimate is to quantify *Stealer* services monetization and we *emphasisize* that the sale proceeds of harvested credentials is out of scope.

Cost and Revenue Modeling. Our cost estimate model assumes service providers operate independently and incur two distinct costs, namely one-off and ongoing costs. *One-off* costs (annually) include the domain purchase and the *Stealer* kit license, whereas *ongoing* costs include hosting. To be conservative, we assume zero cost if the *Stealer* kit is leaked or available as open-source. Moreover, we assume multiple licenses of *Stealer* kits are required per FQDN because developers license per domain. Using IP intelligence, we map the panel IP address to a hosting provider. We then manually collect the hosting prices for shared hosting, virtual private server (VPS), and dedicated server. We exclude cloud-fronted hosts because we cannot identify their hosting provider. We use the prices per *Stealer* family in Table 1 to estimate the revenue for hosted *Stealer* panels (service offering). We assume the revenue for the cluster is generated by offering hosted *Stealer* services. For example, C1 has three malware families, namely *LokiBot*, *AgentTesla*, and *Formbook*. We multiply the lowest license cost from Table 1 by the number of panel instances and sum them up for a revenue of $11,834 per month.

Comparing Cost, Revenue, and Profits. Table 8 summarizes the cost (one-off and ongoing), revenue, and profit margin. We find the range of the one-off cost between $121.61 and $5,481.15 per year. Notably, C4 has a relatively smaller one-off cost because C4 only hosts *LokiBot* malware family, which does not have a license cost (leaked source). We find the range for the ongoing cost per month to be between $37.45 and $923.45. C3 has the lowest ongoing cost because the majority of the domains are cloud-fronted and we could not identify their hosting providers. The most expensive operation is C1, where operators use 155 distinct hosts to serve 157 panel instances, which also supports our observation for being the most stable service from the growth analysis. We

find the revenue to range between $1,930.0 and $11,834 per month. C1 has the largest operational revenue but a relatively lower profit margin. C1 offers three different malware family panel hosting, including *LokiBot*, *Formbook*, and *AgentTesla*, while the rest of the clusters only offer *LokiBot* and *Formbook* panel hosting. This suggests that *Stealer* service providers can be highly profitable with margins that range between 81.45% and 98.55%.

Surprisingly, C2 has half of the number of hosts but their monthly cost is about 10% of C1. C2's low hosting cost can be attributed to the fact that 56 out of the 88 hosts appear to be compromised residential and business devices. These networks appear to be compromised since their rDNS records map to ISP customers (business and residential) but do not appear in the proxy dataset. Most of these networks (54 out of 76 ASNs) point to one domain [2]. This domain was active for two months from Nov'19 to Jan'20 and was associated with 59 IP addresses that belong to 54 distinct ASNs in 21 countries. These results suggest that C2 uses globally infected hosts to offer panel hosting services and is relatively less stable in growth than C1. Conversely, C4 has 71% of its infrastructure geographically hosted in Russia with higher infrastructure costs than C2.

Potential Infected Hosts. To further compare the service providers, we take a closer look at their potential victims using pDNS dataset. In Table 8 under *Infected Networks*, we present the number of unique subnets from ECS [59], [60]. Recall, our earlier analysis suggested that hosting networks are less likely to be infected victims; therefore, we only quantify the residential networks. The residential IP addresses provide a lower bound for the number of potentially infected machines. We find C5 to have the largest number of potential infections with 8,153 unique residential networks followed by C1 with 6,795 networks. Although C4 appears to have a lower number of potential infections, we attribute it to the lack of coverage in the pDNS.

Takeaway-5: The top service providers appear to operate for over a year, have consistent growth, and enjoy over 90% profit margins ranging from $2000 to $11,000 per month. *Stealer* service providers use varying tactics for hosting and it is reflected in their service stability based on the growth and lifespan analysis. The hosting infrastructure varies from infected hosts to geographically concentrated to geographically distributed hosting. These observations show that no two *Stealer* service providers are the same.

7 Summary and Discussion

We set out to investigate the *Stealers* ecosystem by answering following three research questions:

RQ1: How do *Stealers* contribute to cybercrime? *Stealers* play a significant role in the credential theft lifecycle and contribute to the credential harvesting phase. *Stealers* have a mature and competitive market that lowers the financial and technical barrier for cybercriminals. Hosted *Stealer* services appear to require little upfront cost and can potentially offer a large return on investment from the sale of credentials.

RQ2: How do *Stealers* operate on the Internet? *Stealers* require minimal hosting resources and abuse services such as free ccTLDs and cloud-fronting. *Stealer* service providers use varying tactics for hosting and it is reflected by their service stability, i.e., growth. The hosting infrastructure varies from infected hosts to geographically concentrated and distributed hosting. Public blocklist detect *Stealer* domains on average 74 days after initial domain registration. This detection gap gives *Stealers* ample time to infect and harvest credentials from a wide range of networks. Their long-lived activities may be problematic, as they allow operators time to exercise other malware capabilities (i.e., install ransomware).

RQ3: What are the nature and tactics of *Stealer* operators and their service offerings? Operators use proxy services ranging from traditional VPNs to mobile and residential proxies, to Tor networks. The mobile and residential proxies can cause misdirection when characterizing operator profiles. The diurnal analysis suggests that operators administer their botnet as a full-time job. The affiliation analysis suggests that the largest 1% *Stealer* service providers account for most of the activities. Each service provider appears to have one or two influential operators that have more privileged access, suggesting those operators are administrators. Our analysis shows a stratified organization per cluster. These observations show that no two *Stealer* service providers are the same and they appear to operate independently (compete).

Actionable Insights. How can researchers and law enforcement act on these insights? For researchers, we empirically document that *Stealers* have defensive tactics to prevent active scanning and identification of *C&C* panels. Researchers can incorporate this information to build a tailored internet-wide scanning system to find *C&C* panels. For example, a scanner can scan a target host twice, once to trigger a block and a second time to check if the connection is blocked. This approach turns the *Stealer* defense system against itself and allows researchers to detect possible *C&C* panel hosts. Additional insights, such as geographical distribution of infrastructure, ASN association, and infrastructure characteristics can inform researchers to design and evaluate effective active *Stealer* infrastructure detectors.

Law enforcement can apply our operator device profiling techniques to accurately characterize cybercriminals. We show that operators use private browsing and diverse proxy services to masquerade their fingerprint. However, using our cookie churn merging algorithm and diurnal analysis, law enforcement can build a more accurate timeline of device and *C&C* panel access as forensic evidence. Moreover, the affiliation analysis can identify cybercriminal groups and pinpoint

[2] *tranpip[.]com*

their top active participants, which can help law enforcement efficiently go after influential operators. Similarly, our findings can help researchers to identify active *Stealer* infrastructure and prioritize their cleanup. For example, researchers and law enforcement can collaborate to takedown domains with large clusters of operator activities. Lastly, our infection analysis can provide a lead to law enforcement to investigate sensitive networks with potential *Stealer* infections.

Operator Attribution Attribution can be of two types, namely virtual or physical. Physical attribution requires jurisdiction and legal access to private information. Additionally, there is an ethical aspect to physical attribution that must adhere to some acceptable policies and norms. This work focuses on virtual attribution to identify operator affiliation, albeit these techniques are meant to complement and enhance existing methods instead of being used on their own. Virtual attribution deals with identification and tracking of different threat groups based on indicators of compromise (IoC). However, we caution the reader that attribution to a specific group is complex, and we avoid making any speculative judgments. For instance, the majority of indicators in our dataset point to large clusters of activities originating from Nigeria. Although this observation is suggestive, we observe that many Nigerian operator networks are mobile or residential proxies. Enigmatically, these proxies appear to be part of anonymity networks (similar to Tor), where participants may be willingly or unknowingly tunneling traffic [40], [41]. Nevertheless, law enforcement could incorporate our techniques to improve virtual and physical attribution.

7.1 Limitations and Threats to Validity

The operational nature of the *Stealer* dataset can affect the accuracy of our results. The tracking pixel may only appear on some panel pages and therefore miss activities from operator devices. Additionally, since the data collection relies on running malware in a sandbox, the malware binary collection and analysis can create a skewed view of the malware families. However, since our dataset is large (hundreds of thousands of records), we can assume the data is statistically representative of the overall population. The data validation analysis shows that operators may spoof their UA, use private browsing, or use multiple devices. It is difficult, if not impossible, to associate a virtual entity with a physical entity based on the current dataset. Nevertheless, we make conservative assumptions about the operators by framing the analysis as operator *devices* and extensively validating the dataset.

Another possible limitation is the effect of network address translated (NAT) traffic and aggregated pDNS data from recursive servers. These artifacts can impact our infection estimation and operator count. Additionally, operator network proxy use can create ambiguities about the geographical regions of the operators. For the cost and revenue estimates, the prices for hosting and service offerings are based on the time this paper was written; therefore, the prices might have changed over the past years. Nevertheless, our profits estimate should serve as a lower bound for *Stealer* services.

7.2 Related Work

Several studies have analyzed different cybercrime operations in an effort to understand their incentives. These cybercrimes include pharmaceutical spam [54], [66], spam botnets [9], spam life-cycle [5], targeted attacks [67], click-fraud bots [68], ransomware [69], and RATs [70]. Moreover, prior work [71] has explored the cybercrime business relationships and their collaboration. Franklin et al [12] investigated the financial aspect of cybercrime by analyzing transactions on IRC servers. Studying cybercrime operators requires various techniques that include honeypots [72], internet-wide scanning [61], [73], seizing malware infrastructure [9], [11], [74], tracking underground activities [2], [13], analyzing recovered credentials [6], and a combination of diverse data sources [71], [75]. Other works relied on honey tokens to study URL shortening services [47], email typosquatting [48], social media manipulation [53], detect intrusions [50], [51], and vet malicious browser extensions [52]. These works provide a valuable perspective into cybercrime tactics.

In contrast, our work examines a large commodity *Stealers* dataset from the operator's interface. Our work provides the first unique perspective into the inner workings of *Stealer* services and their operators through pixel-tracking embedded in artificial stolen credentials. Pixel-tracking allows us to quantify and distill important insights about the activities of *Stealer* operators that were not possible before. Our work leverages this dataset to analyze the nature, trends, tactics, and revenue of *Stealers* and their operators. In comparison to prior work [76] on *Stealer* panels, our work provides a more holistic and in-depth analysis of the *Stealer* ecosystem.

8 Conclusion

Our empirical analysis of *Stealers* provides a unique perspective on the nature of their ecosystem. Our research questions explored several aspects of the *Stealer* ecosystem, including how they are used for cybercrime, how they operate on the internet, and what are the tactics of their operators. We find much of the *Stealers* infrastructure to be long undetected, which gives operators time to infect the compromised networks with more serious threats like ransomware [17]. The threat posed by *Stealers* is amplified by the low barrier to entry that *Stealer* service providers enable. The *Stealer* service providers enjoy healthy profits, which financially drives additional competition further fueling this ecosystem. We believe additional work is needed to disincentive and curb the use of *Stealers*. Finally, in the spirit of scientific reproducibility, we make 6 months of the

Stealers dataset along with the cookie merging code available at: https://github.com/Astrolavos/stealer-sec23

References

[1] Verizon, "2020 data breach investigations report," *https://enterprise.verizon.com/content/verizonenterprise/us/en/index/resources/reports/2020-data-breach-investigations-report.pdf*,

[2] A. Mirian, J. DeBlasio, S. Savage, G. M. Voelker, and K. Thomas, "Hack for hire: Exploring the emerging market for account hijacking," in *Proc. of the 28th International World Wide Web Conference (WWW)*, San Francisco, CA, USA, 2019.

[3] E. Bursztein, B. Benko, D. Margolis, T. Pietraszek, A. Archer, A. Aquino, A. Pitsillidis, and S. Savage, "Handcrafted fraud and extortion: Manual account hijacking in the wild," in *Proc. of the 14th ACM SIGCOMM Conference on Internet Measurement (IMC)*, Vancouver, BC, Canada, Nov. 2014.

[4] proofpoint - Threat Insight, *New version of azorult stealer improves loading features, spreads alongside ransomware in new campaign*, https://www.proofpoint.com/us/threat-insight/post/new-version-azorult-stealer-improves-loading-features-spreads-alongside, 2018.

[5] A. Oest, P. Zhang, B. Wardman, E. Nunes, J. Burgis, A. Zand, K. Thomas, A. Doupé, and G.-J. Ahn, "Sunrise to sunset: Analyzing the end-to-end life cycle and effectiveness of phishing attacks at scale," in *Proc. of the 29th USENIX Security*, Aug. 2020.

[6] T. Holz, M. Engelberth, and F. Freiling, "Learning more about the underground economy: A case-study of keyloggers and dropzones," in *European Symposium on Research in Computer Security*, 2009.

[7] B. Eshete, A. Alhuzali, M. Monshizadeh, P. A. Porras, V. N. Venkatakrishnan, and V. Yegneswaran, "Ekhunter: A counter-offensive toolkit for exploit kit infiltration.," in *NDSS15*.

[8] S. Hao, M. Thomas, V. Paxson, N. Feamster, C. Kreibich, C. Grier, and S. Hollenbeck, "Understanding the domain registration behavior of spammers," in *Proc. of the 13th ACM SIGCOMM Conference on Internet Measurement (IMC)*, Barcelona, Spain, Oct. 2013.

[9] B. Stone-Gross, T. Holz, G. Stringhini, and G. Vigna, "The underground economy of spam: A botmaster's perspective of coordinating large-scale spam campaigns.," *4th {USENIX} Workshop on Large-Scale Exploits and Emergent Threats ({LEET} 11)*, 2011.

[10] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao, "Detecting and characterizing social spam campaigns," in *Proc. of the 10th ACM SIGCOMM Conference on Internet Measurement (IMC)*, Melbourne, Australia, Nov. 2010.

[11] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna, "Your botnet is my botnet: Analysis of a botnet takeover," in *Proc. of the 16th ACM CCS*, Chicago, Illinois, Nov. 2009.

[12] J. Franklin, A. Perrig, V. Paxson, and S. Savage, "An inquiry into the nature and causes of the wealth of internet miscreants.," in *Proc. of the 14th ACM CCS*, Alexandria, VA, Nov. 2007.

[13] K. Thomas, F. Li, A. Zand, J. Barrett, J. Ranieri, L. Invernizzi, Y. Markov, O. Comanescu, V. Eranti, A. Moscicki, *et al.*, "Data breaches, phishing, or malware? understanding the risks of stolen credentials," in *Proc. of the 24th ACM CCS*, Dallas, TX, Oct. 2017.

[14] I. F. 1, *Three arrested as interpol, group-ib and the nigeria police force disrupt prolific cybercrime group*, https://www.interpol.int/en/News-and-Events/News/2020/Three-arrested-as-INTERPOL-Group-IB-and-the-Nigeria-Police-Force-disrupt-prolific-cybercrime-group, 2020.

[15] I. F. 2, *Nigerian cybercrime fraud: 11 suspects arrested, syndicate busted*, https://www.interpol.int/News-and-Events/News/2022/Nigerian-cybercrime-fraud-11-suspects-arrested-syndicate-busted, 2022.

[16] I. Delilah, *Suspected head of cybercrime gang arrested in nigeria*, https://www.interpol.int/News-and-Events/News/2022/Suspected-head-of-cybercrime-gang-arrested-in-Nigeria, 2022.

[17] R. Future, *Initial access brokers are key to rise in ransomware attacks*, https://www.recordedfuture.com/initial-access-brokers-key-to-rise-in-ransomware-attacks, 2022.

[18] Blueliv, "The Credential Theft Ecosystem," *https://web.archive.org/web/20210107175227/https://www.blueliv.com/resources/reports/The_credential_theft_ecosystem.pdf*, 2018.

[19] P. Renals, "Unit 42 technical analysis: Silverterrier: 2019 nigerian business email compromise update," *Palo Alto Networks*, 2020.

[20] Federal Bureau of Investigation: Internet Crime Complaint Center (IC3), "2020 internet crime report," 2021.

[21] J. M. Esparza, "Understanding the credential theft lifecycle," *Computer Fraud & Security*, 2019.

[22] d00rt, "Lokibot infostealer "hijacked" version," *https://web.archive.org/web/20210117055851/https://raw.githubusercontent.com/d00rt/hijacked_lokibot_version/master/doc/LokiBot_hijacked_2018.pdf*, 2018.

[23] N. Villeneuve, R. Eitzman, S. Nemes, and T. Dean, "Significant FormBook Distribution Campaigns Impacting the U.S. and South Korea," *https://web.archive.org/web/20201211012535/https://www.fireeye.com/blog/threat-research/2017/10/formbook-malware-distribution-campaigns.html*, 2017.

[24] KrabsOnSecurity, "Analyzing Amadey – a simple native malware," *https://web.archive.org/web/20201107235827/https://krabsonsecurity.com/2019/02/13/analyzing-amadey-a-simple-native-malware/*, 2019.

[25] A. Zsigovits, "Baldr vs The World," *https://web.archive.org/web/20191217054044/https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/baldr-vs-the-world.pdf*, 2019.

[26] Partheeban, "Dark Side Of BlackNET RAT," *https://web.archive.org/web/20201224065951/https://labs.k7computing.com/?p=21365*, 2020.

[27] The BlackBerry Cylance Threat Research Team, "Threat Spotlight: Analyzing AZORult Infostealer Malware," *https://web.archive.org/web/20200920132950/https://blogs.blackberry.com/en/2019/06/threat-spotlight-analyzing-azorult-infostealer-malware*, 2019.

[28] Malware Don't Need Coffee, "Neutrino Bot (aka MS:Win32/Kasidet)," *https://web.archive.org/web/20201129222945/https://malware.dontneedcoffee.com/2014/06/neutrino-bot-aka-kasidet.html*, 2014.

[29] J. WALTER, "Agent Tesla | Old RAT Uses New Tricks to Stay on Top," *https://web.archive.org/web/20201207093051/https://labs.sentinelone.com/agent-tesla-old-rat-uses-new-tricks-to-stay-on-top/*, 2020.

[30] fr3dhk, "Nexus - Just another stealer," *https://web.archive.org/web/20201129072131/https://fr3d.hk/blog/nexus-just-another-stealer*, 2020.

[31] D. SCHWARZ, "New KPOT v2.0 stealer brings zero persistence and in-memory features to silently steal credentials," https://web.archive.org/web/20201207040629/https://www.proofpoint.com/us/threat-insight/post/new-kpot-v20-stealer-brings-zero-persistence-and-memory-features-silently-steal, 2019.

[32] Cynet, *Redline is on track, next stop – your credentials*, https://www.cynet.com/attack-techniques-hands-on/redline-is-on-track-next-stop-your-credentials/, 2022.

[33] A. Kountouras, P. Kintis, C. Lever, Y. Chen, Y. Nadji, D. Dagon, M. Antonakakis, and R. Joffe, "Enabling network security through active dns datasets," in *Proc. of the 19th RAID*, Evry, France, Sep. 2016.

[34] SecurityTrails, "URLScan - a free service to scan and analyse websites.," https://urlscan.io/about/, 2016.

[35] VirusTotal, "Virustotal-free online virus, malware and url scanner," https://www.virustotal.com/en, 2004.

[36] IPRegistry, "IP Geolocation and Threat Detection," https://ipregistry.co/,

[37] ViriBack Tracker, "C2 Tracker," http://tracker.viriback.com/,

[38] benkow, "Panel Tracker," https://benkow.cc/passwords.php,

[39] Cybercrime Tracker, "Bot Tracker," https://cybercrime-tracker.net/index.php,

[40] X. Mi, X. Feng, X. Liao, B. Liu, X. Wang, F. Qian, Z. Li, S. Alrwais, L. Sun, and Y. Liu, "Resident evil: Understanding residential ip proxy as a dark service," in *Proc. of the 40th S&P Oakland*, May 2019.

[41] X. Mi, S. Tang, Z. Li, X. Liao, F. Qian, and X. Wang, "Your phone is my proxy: Detecting and understanding mobile proxy networks," in *Proc. of the 2021 NDSS*, Virtual, Feb. 2021.

[42] *Alert (AA20-266A): LokiBot Malware*, Accessed on 08/01/2021. [Online]. Available: https://us-cert.cisa.gov/ncas/alerts/aa20-266a.

[43] S. Nachum, A. Schuster, and O. Etzion, "Detection in the dark–exploiting xss vulnerability in c&c panels to detect malwares," in *International Symposium on Cyber Security Cryptography and Machine Learning*, Springer, 2018.

[44] Mozilla, *Firefox 87 trims http referrers by default to protect user privacy*, https://blog.mozilla.org/security/2021/03/22/firefox-87-trims-http-referrers-by-default-to-protect-user-privacy/, 2021.

[45] A. Dasgupta, M. Gurevich, L. Zhang, B. Tseng, and A. O. Thomas, "Overcoming browser cookie churn with clustering," in *Proceedings of the fifth ACM international conference on Web search and data mining*, 2012.

[46] A. Mohaisen and O. Alrawi, "Av-meter: An evaluation of antivirus scans and labels," in *Proc. of the DIMVA*, London, UK, Jun. 2014.

[47] N. Nikiforakis, F. Maggi, G. Stringhini, M. Z. Rafique, W. Joosen, C. Kruegel, F. Piessens, G. Vigna, and S. Zanero, "Stranger danger: Exploring the ecosystem of ad-based url shortening services," in *Proc. of the 23rd International World Wide Web Conference (WWW)*, Seoul, South Korea, 2014.

[48] J. Szurdi and N. Christin, "Email typosquatting," in *Proc. of the 17th ACM SIGCOMM Conference on Internet Measurement (IMC)*, London, UK, Nov. 2017.

[49] B. Liu, Z. Liu, J. Zhang, T. Wei, and W. Zou, "How many eyes are spying on your shared folders?" In *Proceedings of the 2012 ACM workshop on Privacy in the electronic society*, 2012.

[50] M. B. Salem and S. J. Stolfo, "Decoy document deployment for effective masquerade attack detection," in *Proc. of the DIMVA*, Jul. 2011.

[51] J. Yuill, M. Zappe, D. Denning, and F. Feer, "Honeyfiles: Deceptive files for intrusion detection," in *Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop, 2004.*, 2004.

[52] A. Kapravelos, C. Grier, N. Chachra, C. Kruegel, G. Vigna, and V. Paxson, "Hulk: Eliciting malicious behavior in browser extensions," in *Proc. of the 23rd USENIX Security*, San Diego, CA, Aug. 2014.

[53] E. De Cristofaro, A. Friedman, G. Jourjon, M. A. Kaafar, and M. Z. Shafiq, "Paying for likes? understanding facebook like fraud using honeypots," in *Proc. of the 14th ACM SIGCOMM Conference on Internet Measurement (IMC)*, Vancouver, BC, Canada, Nov. 2014.

[54] K. Levchenko, A. Pitsillidis, N. Chachra, B. Enright, M. Félegyházi, C. Grier, T. Halvorson, C. Kanich, C. Kreibich, H. Liu, *et al.*, "Click trajectories: End-to-end analysis of the spam value chain," in *Proc. of the 32nd S&P Oakland*, Oakland, CA, May 2011.

[55] M. Bailey, D. Dittrich, E. Kenneally, and D. Maughan, "The menlo report," *IEEE Security & Privacy*, 2012.

[56] L. Gelinas, A. Wertheimer, and F. G. Miller, "When and why is research without consent permissible?" *Hastings Center Report*, 2016.

[57] Amazon Web Services, Inc., "Alexa top sites.," https://www.alexa.com/topsites, 2021.

[58] The Spamhaus Project, "The 10 Most Abused Top Level Domains," https://www.spamhaus.org/statistics/tlds/,

[59] A. Kountouras, P. Kintis, A. Avgetidis, T. Papastergiou, C. Lever, M. Polychronakis, and M. Antonakakis, "Understanding the growth and security considerations of ecs.," in *NDSS*, 2021.

[60] P. Kintis, Y. Nadji, D. Dagon, M. Farrell, and M. Antonakakis, "Understanding the privacy implications of ecs," in *Proc. of the DIMVA*, Donostia-San Sebastián, ES, Jul. 2016.

[61] M. Rezaeirad, B. Farinholt, H. Dharmdasani, P. Pearce, K. Levchenko, and D. McCoy, "Schrödinger's RAT: Profiling the stakeholders in the remote access trojan ecosystem," in *Proc. of the 27th USENIX Security*, Baltimore, MD, Aug. 2018.

[62] O. E. Agboje, S. O. Adedoyin, and C. U. Ndujiuba, "State of fiber optic networks for internet broadband penetration in nigeria-a review," *International Journal of Optoelectronic Engineering*, vol. 7, no. 1, pp. 1–12, 2017.

[63] Scamalytics, *M247 ltd - fraud risk scamalytics*, https://scamalytics.com/ip/isp/m247-ltd, 2022.

[64] Scamalytics, *Avast software s.r.o. - fraud risk scamalytics*, https://scamalytics.com/ip/isp/avast-software-s-r-o, 2022.

[65] Scamalytics, *Hern labs ab - fraud risk scamalytics*, https://scamalytics.com/ip/isp/hern-labs-ab, 2022.

[66] C. Kanich, C. Kreibich, K. Levchenko, B. Enright, G. M. Voelker, V. Paxson, and S. Savage, "Spamalytics: An empirical analysis of spam marketing conversion," in *Proc. of the 15th ACM CCS*, Alexandria, VA, Oct. 2008.

[67] S. Le Blond, A. Uritesc, C. Gilbert, Z. L. Chua, P. Saxena, and E. Kirda, "A look at targeted attacks through the lense of an NGO," in *Proc. of the 23rd USENIX Security*, San Diego, CA, Aug. 2014.

[68] P. Pearce, V. Dave, C. Grier, K. Levchenko, S. Guha, D. McCoy, V. Paxson, S. Savage, and G. M. Voelker, "Characterizing large-scale click fraud in zeroaccess," in *Proc. of the 21st ACM CCS*, Scottsdale, Arizona, Nov. 2014.

[69] D. Y. Huang, M. M. Aliapoulios, V. G. Li, L. Invernizzi, E. Bursztein, K. McRoberts, J. Levin, K. Levchenko, A. C. Snoeren, and D. McCoy, "Tracking ransomware end-to-end," in *Proc. of the 39th S&P Oakland*, San Francisco, CA, May 2018.

[70] B. Farinholt, M. Rezaeirad, P. Pearce, H. Dharmdasani, H. Yin, S. Le Blond, D. McCoy, and K. Levchenko, "To catch a ratter: Monitoring the behavior of amateur darkcomet rat operators in the wild," in *Proc. of the 38th S&P Oakland*, San Jose, CA, May 2017.

[71] G. Stringhini, O. Hohlfeld, C. Kruegel, and G. Vigna, "The harvester, the botmaster, and the spammer: On the relations between the different actors in the spam landscape," in *Proc. of the 21st ACM CCS*, Scottsdale, Arizona, Nov. 2014.

[72] T. Barron and N. Nikiforakis, "Picky attackers: Quantifying the role of system properties on intruder behavior," in *Proc. of the 33th ACSAC*, 2014.

[73] W. R. Marczak, J. Scott-Railton, M. Marquis-Boire, and V. Paxson, "When governments hack opponents: A look at actors and technology," in *Proc. of the 23rd USENIX Security*, San Diego, CA, Aug. 2014.

[74] C. Y. Cho, J. Caballero, C. Grier, V. Paxson, and D. Song, "Insights from the inside: A view of botnet management from infiltration.," *4th {USENIX} Workshop on Large-Scale Exploits and Emergent Threats ({LEET} 11)*, 2010.

[75] C. Lever, P. Kotzias, D. Balzarotti, J. Caballero, and M. Antonakakis, "A lustrum of malware network communication: Evolution and insights," in *Proc. of the 38th S&P Oakland*, San Jose, CA, May 2017.

[76] A. K. Sood, S. Zeadally, and R. Bansal, "Cybercrime at a scale: A practical study of deployments of http-based botnet command and control panels," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 22–28, 2017.

A Supplement Material

Table 9: Summary of panel installation, encryption functions, admin authentication, and supported admin commands based on source code analysis.

Malware Family	Install Type	Encryp Algo.	Panel Auth.	Bot Commands
Neutrino	Scripted		user/passwd	DDoS, shell, keylogger, DNS spoof, update
LokiBot	Guided	AES256-ECB, RC4	user/passwd, UA, Captcha	load/drop exec, keylogger, screenshot, update, uninstall
AZORult	Manual	1-Byte XOR	only passwd	
Amadey	Manual		user/passwd	drop/load exec., RAT
BlackNet	Guided		user/passwd, Captcha, 2FA	DDoS, upload, msg, visit page, mail, keylogger, shell, uninstall

Table 10: Top 10 hosting networks querying stealer domains.

Hosting AS	Networks
AMAZON-AES	30,705
AMAZON-02	12,515
CLOUDFLARENET	5,890
MICROSOFT-CORP-MSN-AS-BLOCK	4,708
OVH OVH SAS	1,623
DIGITALOCEAN-ASN	728
MAXIHOST	543
M247 M247 Ltd	536
SOFTLAYER	461
UK2NET-AS UK-2 Limited	332

Table 11: Top panel operator device types and operating systems.

Desktop			Mobile		
OS	Ver.	Count (%)	OS	Ver.	Count (%)
Windows	10	2,148 (46.84)	Android	9	22 (0.48)
	7	1,112 (24.25)		8.1	18 (0.39)
	8.1	893 (19.47)		7	17 (0.37)
MacOS	10.14	47 (1.02)		8	9 (0.19)
	10.15	29 (0.63)		10	9 (0.19)
	10.13	26 (0.56)		6	6 (0.13)
Linux	All	50 (1.10)	iOS	12	8 (0.17)

Table 12: Top 10 user agents and related statistics.

User Agent	OS	Browser	Cookie IDs	C&C	Update (Days)
Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36	Windows 7	Chrome 75.0.3770.100	116	119	22.50
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.130 Safari/537.36	Windows 10	Chrome 79.0.3945.130	112	110	24.15
Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:68.0) Gecko/20100101 Firefox/68.0	Windows 10	Firefox 68.0	112	140	36.32
Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:69.0) Gecko/20100101 Firefox/69.0	Windows 10	Firefox 69.0	111	113	47.23
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.142 Safari/537.36	Windows 10	Chrome 75.0.3770.142	109	120	53.54
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36	Windows 10	Chrome 75.0.3770.100	108	122	21.74
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36	Windows 10	Chrome 73.0.3683.103	95	88	28.57
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 Safari/537.36	Windows 10	Chrome 74.0.3729.169	88	109	22.31
Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:70.0) Gecko/20100101 Firefox/70.0	Windows 10	Firefox 70.0	82	96	24.95
Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.142 Safari/537.36	Windows 7	Chrome 75.0.3770.142	80	72	17.46