Life-C2ycle: Tracking the Lifecycle of Malware Command and Control Services

Aidan Delwiche
University of Michigan
Ann Arbor, MI
delwiche@umich.edu

Ben Schwartz
University of Michigan
Ann Arbor, MI
scben@umich.edu

Abstract*—Command-and-Control (C2) servers are foundational to modern malware campaigns, enabling attackers to maintain persistent access, exfiltrate data, and issue remote commands to compromised systems. Blocking access to these servers is critical for mitigating cyber threats, but traditional blacklisting methods struggle to balance efficacy and collateral damage due to the dynamic, short-lived, and often ephemeral nature of C2 infrastructure. In this paper, we present a data-driven methodology for characterizing and predicting the lifecycle of malware C2 servers. Leveraging threat intelligence from ThreatFox and historical scan data from Censys, we analyze hosts associated with five prevalent malware families: AsyncRAT, SectopRAT, ERMAC, HookBot, and Covenant. We extract a range of behavioral metrics—including SSH and RDP fingerprints, DNS records, Autonomous System (AS) data, and Censys-applied labels to infer the continuity and ownership of malicious infrastructure over time. Based on this analysis, we develop heuristic rules tailored to each malware family that estimate optimal blocking durations, enabling defenders to more confidently retire IPs from blocklists while minimizing risk. Our findings show the importance of persistent infrastructure indicators in assessing ongoing threats and demonstrate that even intermittent or superficially benign behavior can potentially mask sustained malicious control. This work helps create smarter, more flexible ways to block suspicious IP addresses and encourages future research into how cybercriminal servers behave over time.

I. Introduction

Cyberattacks leveraging malware occur daily, threatening individuals, businesses, and critical infrastructure around the globe. Once an attacker has successfully compromised a victim's system, their goals can range drastically—from deleting all of the user's data, to encrypting data and requiring a ransom to recover it, to achieving persistent access and enrolling the system in a botnet with remote command execution (RCE).

Among these goals, sensitive data exfiltration and persistent RCE definitely represent the attacker's highest-valued objectives. Both of these activities require consistent and reliable communication between infected victim systems and the attackers' own infrastructure. These infrastructures are commonly known as Command-and-Control (C2) servers. C2 servers represent the backbone of modern cyber threats; they enable attackers to control infected devices, issue commands remotely, receive stolen data, and are heavily relied on by botnets [15].

C2 infrastructure is also shockingly easy for attackers to set up due to the prevalence of C2 frameworks. There are many open-source and closed-source projects available that can facilitate the desired communication with infected devices, often claiming to be designed for legitimate purposes. For instance, Supershell is an open-source C2 remote control platform that provides a web interface for users to control connected devices, including the ability to run reverse shells and manage files [16]. Supershell, along with virtually any C2 framework, is marketed as a tool for red-teamers to simulate real-world attacks and test the effectiveness of defensive measures. Obviously, an attacker can just treat this as a plug-and-play solution for deploying and managing their own malicious infrastructure with minimal effort.

Given their nature, C2 infrastructure typically requires exposure to the Internet to effectively coordinate malicious operations on infected systems across a wide geographic area—a substantial challenge for defenders trying to mitigate their threat. To properly execute such an attack, malicious actors need only three components: (1) malware, (2) a C2 framework, and (3) an Internet-facing server. (1) Remote Access Trojans (RATs) are widely available, easy to build from scratch and deploy to victims, such as through email phishing campaigns. (2) A C2 framework is typically well documented, user-friendly and sometimes even actively maintained, making setup and operation trivial. (3) Virtually anyone can spin up a server with a public IP address using affordable cloud services, with providers rarely scrutinizing how these servers are used.

Of course, the effects of C2 attacks could be mitigated by preventing a system from getting infected with malware in the first place. However, this is a long and well-researched battle, with the consensus usually being that under the current conditions, it is almost impossible to prevent infections with malware [3]. This forces defenders to pivot their goal to block communications with known C2 infrastructure at network boundaries, such as Internet Service Providers (ISPs) or enterprise firewalls.

Unfortunately, effective blocking requires accurate identification of C2 server IP addresses, a non-trivial task given the highly dynamic nature of malicious C2 servers. Attackers employ various tactics to evade detection, such as rapidly rotating IP assignments, domain fronting (using different domain names in the TLS SNI header and the HTTP Host

header), Domain Generation Algorithms (creating thousands of pseudo-random domain names for accessing the server), communicating over obscure channels other than HTTP, such as DNS or peer-to-peer protocols, thereby making traditional detection and filtering approaches significantly less effective. In this paper, improving C2 detection is considered out of scope.

Because of this, blocking IP addresses of known C2 servers indefinitely is neither practical nor sustainable, as previously malicious IP addresses may subsequently host legitimate services. This calls for a deep analysis of the behavior of servers that have hosted C2 infrastructure at some point in time and developing a robust heuristic to accurately estimate the lifespan of malicious IP addresses, which can allow defenders to efficiently manage their blocking policies and mitigate false positives. Developing this heuristic requires a comprehensive analysis of historical data of known C2 infrastructure, identifying consistent patterns indicative of the status of malware servers.

In this paper, we address this problem by leveraging historical scanning data to track the lifecycle of the infrastructure of different C2 malware families. Our contributions include:

- A methodology for tracking and analyzing the lifecycle of malware C2 infrastructure using historical scanning datasets.
- Development of a heuristic approach for determining optimal blocking periods for previously identified malicious IP addresses based on their observed behavior and characteristics.

II. Related Work

Accurate identification and tracking of C2 servers are crucial to effectively mitigating cyber threats. Due to their evasive nature, C2 servers frequently exhibit short lifespans and intermittent availability. For instance, studies focusing on IoT botnets have found that approximately 80% of observed C2 IP addresses remain active for only a single day, with an overall average lifetime of just four days [6]. Additionally, these servers often exhibit inconsistent responsiveness to external probes; one study found that 95% of identified C2 servers responded to fewer than 12% of scans conducted over a week. This sporadic behavior does not reliably indicate that a C2 server has permanently ceased operations [12]. Continuous and widespread scanning, such as what Censys does [17], is therefore critical to maintaining an accurate understanding of active C2 infrastructure.

Previous work has attempted to develop tactics for identifying active C2 infrastructure. Techniques utilizing machine learning can highlight indications of C2 activity based on network and host behaviors [9]. Another approach involves using historic malware samples to actively probe and confirm whether a suspected host is operating as a C2 server [5]. We describe our fingerprinting methods in Section III-D, with the goal of aligning closely with existing fingerprinting methods implemented by scanning services such as Censys [11].

A significant challenge, however, lies in determining the long-term malicious nature of an IP once identified as hosting a C2 server. Studies indicate that C2 servers frequently reuse infrastructure; for example, approximately 60% of IoT malware C2 servers have been contacted by multiple distinct malware variants over time [6]. Furthermore, certain IP addresses host multiple different malware families simultaneously or successively, indicating their role as persistent hubs of malicious activity [12]. This reuse suggests that historical detection of malicious activity on an IP address may serve as a reliable indicator of future malicious usage, which can validate IP blocking as an effective defense strategy.

Nonetheless, permanently blocking IP addresses based on historical detection alone is not viable due to the dynamic nature of IP allocation, especially within cloud environments. Previously malicious IP addresses can be reassigned to legitimate services, rendering indefinite blocking impractical and potentially disruptive. Thus, it is crucial to analyze multiple additional data points to robustly determine the current state of a server. Our approach integrates these various data sources alongside fingerprint-based identification, to formulate a heuristic capable of accurately determining the period during which IP addresses identified as malicious remain a threat.

III. Methodology

In this section, we provide a detailed overview of our approach to analyzing and tracking the lifecycle of malware C2 infrastructure. Our analysis leverages the selection of C2 frameworks to analyze, finding Indicators of Compromise (IOCs), collecting historical scan data, analyzing and comparing certain metrics, and developing a heuristic for the lifetime of a specific malware family. We outline this workflow through the following subsections.

A. Selection of C2 Frameworks

Initially, we planned to study widely used and popular C2 frameworks such as Cobalt Strike [10] and Supershell [16]. However, as stated before, many C2 frameworks market themselves as being built for legitimate red-team purposes, and this actually seems to be the case for these two C2 frameworks. To minimize false positives within our dataset, we pivoted to analyzing C2 frameworks that can reasonably be assumed to be used solely for malicious purposes.

For this paper, we have chosen to analyze instances of the following malware families:

- **AsyncRAT** — A Windows-based remote access trojan (RAT) that enables attackers to monitor and control infected systems via encrypted channels, often used for keylogging and data exfiltration [2].
- **SectopRAT** — A .NET-based RAT for Windows, featuring stealth capabilities like hidden desktops and browser session hijacking, commonly employed for credential theft [14].
- **ERMAC** — An Android banking trojan that utilizes overlay attacks and accessibility service abuse to steal

credentials from financial and cryptocurrency applications [8].
- **HookBot** — An advanced Android banking trojan derived from ERMAC, incorporating features such as keylogging, screen capture, and remote control to harvest sensitive data [13].
- **Covenant** — A .NET-based command-and-control (C2) framework designed for red-team operations, which has been repurposed by threat actors for malicious activities [4].

Of course, these selections are somewhat arbitrary and may still include false positives; they represent our best effort to identify C2 malware families that are both popular and predominantly used for malicious purposes. To accommodate potential shifts in the C2 landscape, we designed our analysis scripts to be adaptable and applicable to any malware family, enabling future work to build on this foundation.

B. Identification of Malicious Hosts

To be able to look at historical scan data, we first need to generate a list of $\langle \text{IP}, \text{timestamp} \rangle$ tuples corresponding to instances of C2 malware found "in the wild".

To accomplish this, we utilize ThreatFox, a community-driven threat intelligence platform that shares Indicators of Compromise (IOCs) [18]. This allows victims and researchers to provide real-time reports of malware, clearly tagging malicious software observed on specific IP addresses. ThreatFox provides structured reports that include not only the observed IP address but also associated metadata such as the malware family, timestamp of detection, confidence level, and contextual evidence.

By focusing on entries with clearly attributed C2 activity and aligning them with our chosen malware families, we were able to construct a list of $\langle \text{IP}, \text{timestamp} \rangle$ tuples (also referred to as "IOCs" in this paper) that were likely acting as a malicious C2 server at some point in time. We use this list to conduct historical analysis in the following step.

C. Historical Data Collection

Now, with a list of IOCs to analyze, we want to be able to "look around" the timestamp of each IOC to see how that service was behaving before and after the report occurred.

To do this, we leverage Censys, an internet-wide scanning service that performs daily scans of every active host in the IPv4 address space [17]. For each host we analyze, we look at two components that the Censys dataset gives us:

- **Raw Scan Data** — For each host on the Internet, and for every active port on that host, Censys collects exhaustive details daily, including HTTP headers, TLS certificates, DNS records, HTML content, and Autonomous System (AS) data.
- **Labels** — Censys maintains a wide range of fingerprints for different types of software. After collecting raw scan data for a given service, it applies all relevant fingerprints, labeling any matches with the corresponding software identifiers. For instance, a researcher at Censys may notice that a certain malware family includes the string "4rg5q8g3" in the HTTP body of its running services. They can then create a fingerprint so that any service matching this condition is labeled as 'C2', making it easy for users to identify potential C2 servers. Our work addresses a similar problem; however, since Censys fingerprints are proprietary, we instead use the presence of the Censys 'C2' label as a point of comparison for validating our own findings.

For each IOC we have from ThreatFox, we use Censys to generate a "look around" report, where we look at how certain properties of a host changed before and after the IOC timestamp. We describe these properties (termed "metrics" in this paper) in the following step.

D. Lifecycle Analysis and Metric Extraction

With the historical dataset assembled, we systematically analyze each C2 host's behavior over time. Specifically, for each IOC's historical data (the daily raw scan data N days before and N days after the IOC timestamp), we extracted several key metrics from each and monitored how they change over time. These metrics help us infer when a host was likely acting as a C2 server, and when it transitioned into or out of that role. We describe each metric and exactly what we look for below.

1) SSH Fingerprint: The SSH host key fingerprint presented by a server can provide insight into potential changes in ownership or configuration of the host. For example, if an IOC-associated host presents a specific SSH fingerprint at the time of detection, but this fingerprint changes in the days that follow, it may indicate that the original malicious operator has abandoned the host and it has since been reassigned—possibly to a benign user on a shared or cloud-based infrastructure. On the other hand, a consistent fingerprint over time may suggest that the host has remained under control of the same entity.

2) RDP Fingerprint: Remote Desktop Protocol (RDP) fingerprints, when available, offer additional insights into the configuration of Windows-based systems. Like SSH, a change in an RDP fingerprint may indicate that the host has been reassigned. Likewise, a stable RDP fingerprint may imply continued access by the same operator.

3) DNS Records: DNS data can include A records, CNAMEs, MX records, and other related information. These can help identify when a host is tied to C2 activity. For instance, if a domain suddenly resolves to a known IOC IP shortly before the IOC timestamp, this may indicate the setup phase of a malware campaign. Conversely, the removal of DNS associations with a C2 host may suggest the relocation or shutdown of the C2 host.

4) Autonomous System (AS) Information: ASes are networks operated by a single organization or entity, and certain ASes have a history of being complicit in hosting malicious infrastructure. By tracking the AS number (ASN) and name associated with a host over time, we can detect if a C2 server remained within a single AS or if it changed ASes. We understand it is unlikely for an IP to change to a different

AS—this information can still be fruitful since it can help us map out the behavior of malicious hosts based on which AS they are hosted on.

5) Geolocation Data: Similar to the AS information, we may be able to gain insights or notice patterns of C2 activity based on what country they are hosted in. This distinction is important to make, since certain countries (such as the Seychelles or the Netherlands), have many privacy laws that allow malware campaigns to remain on their infrastructure with little interference.

6) Censys Labels: As discussed earlier, Censys applies proprietary fingerprinting techniques to label services, including those suspected of being C2 servers. We use these labels as a point of comparison and validation for our own analyses. For instance, if a host has the C2 label on Censys during or near the IOC timestamp, this provides confidence, and on the other hand, mismatches call for deeper manual inspection to understand why Censys failed to detect it.

E. Data Aggregation and Mapping

Given these categories of information we knew about the host, we were then able to extract timeline information for each one of the metrics. We did this by examining the information from the $2N$-day span of scanning data and identifying when particular values in each category appeared, disappeared, or changed. Each unique value and its associated time interval was recorded, forming a cumulative timeline for that specific metric.

We denoted that any metric changed based on a normalization and comparison approach drawn from our analysis code. Each metric's value was first normalized (e.g., lowercased, truncated, or filtered for emptiness) to ensure consistency across scan records. Then, we applied a metric-specific change detection function. For example:

- For AS information, a change was logged if either the AS number or description differed from the previous scan, ignoring placeholder or null entries.
- For server product names and country codes, a change was identified when the normalized value differed from the previous one.
- For SSH public keys, we compared the entire fingerprint dictionaries (truncated for readability), recording a change whenever the key structure altered.
- For DNS names, we compared them as sets: if two successive sets were disjoint (i.e., no shared names), this indicated a significant DNS change.
- For Censys labels, we used a simplified boolean flag to track if the host was tagged with either a `c2` or `bulletproof` label. A change was recorded when this flag flipped between true and false.

One special case of this was the Censys service labels themselves. Rather than forming a timeline of the labels, we instead tracked a boolean flag indicating whether the host was labeled as having C2 or bulletproof hosting characteristics at each point in time. This binary labeling helped us quickly identify suspicious activity windows. In the future work section, we note how looking at adjacent Censys service labels could be another indicator of malicious ownership to explore.

From these timeline views, we then visualized each host's behavior using `matplotlib`, representing persistent values as horizontal bars. This provided valuable insight into the lifecycle of C2 infrastructure. Through this visual exploration, we observed hosts that stopped responding and later returned, servers that frequently rotated SSH keys, and transitions from benign to malicious behavior (and vice versa). These timelines gave a clearer picture of how C2 servers evolve over time, and any quirks to look out for, allowing us to adjust our detection heuristic accordingly.

We also used the timeline data to calculate mean lifetimes for each metric, along with aggregate counts for AS usage and hosting countries. We used overlapping intervals to determine which metric time spans to include in our averages. Specifically, if a metric interval overlapped with the IOC timestamp or any period during which the host was marked as malicious, we included it in the calculation. These averages and totals were incorporated into our final heuristic to help generalize findings across hosts.

F. Development of Heuristic

By utilizing the extracted metrics and how they changed during the period "around" the IOC timestamp and C2-tagged time spans, we then develop heuristics for predicting the expected duration of malicious activities for specific C2 malware families. We describe the process and results of this in Section IV.

IV. Results

From the timeline analysis of the measured metrics, we were able to generate timeline graphs for every host we queried on Censys. We also created graphs that looked at the host metrics in aggregate, including average life span of metrics when a host was marked as malicious, and even raw counts of AS numbers and countries. We will begin by looking at the time span graphs as case studies, and add insight into how we used this to validate and build out our detection heuristic. If you find that the figures are too small to read in the paper, you can find them under the `paper/` directory in the following repository: github.com/btschwartz12/life-c2ycle.

A. Timeline Graphs

1) Case Study 1: ERMAC host 193.35.17.242: Figure 1 illustrates the timeline of a suspected C2 server exhibiting a period of inactivity followed by a potential "revival." At the outset of the Censys observation window, the server is briefly tagged as a C2. During this time, several persistent indicators of ownership are present—most notably, the SSH public key and DNS records. These consistent artifacts suggest that despite the brief C2 designation, the server likely remained under the control of the same threat actors, and thus potentially remained malicious.

Subsequently, from IOC offsets -3 to -2, the server appears to stop responding to Censys probes, indicating a period of

inactivity. At offset -2, the server begins responding again. Initially, it is not marked as malicious, but by offset 0, the C2 label reappears. This pattern indicates a potential reactivation or repurposing of the same infrastructure.

The key takeaway from this timeline is that the absence of a C2 label does not necessarily indicate the absence of malicious activity. If persistent indicators of ownership remain, the host should continue to be treated as potentially malicious. Furthermore, even after these indicators vanish, caution is warranted for some time thereafter, especially in the event of a "revival" such as the one illustrated in the graph.

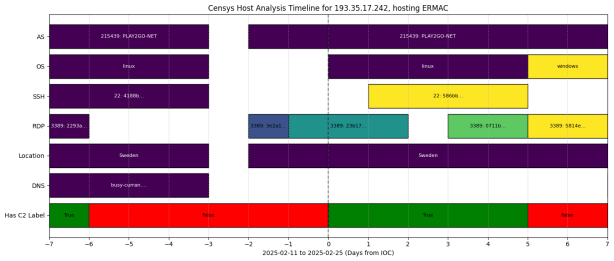

Fig. 1: Temporal analysis of network and system attributes for IP 193.35.17.242 displaying potential C2 activity and indicators of ownership

2) Case Study 2: HookBot host 185.39.206.11: Figure 2 provides insight into potential naïve detection avoidance behavior associated with a C2 service. The timeline reveals a recurring pattern in which the C2 service fingerprint appears and disappears in roughly three-day intervals. This cyclical behavior suggests that the service may be intentionally toggled on and off, or perhaps is configured to intermittently respond to Censys probes. In either case, such a pattern may indicate an attempt to evade consistent detection or fingerprinting.

Notably, throughout the entire observation period, the SSH public key remains unchanged. This persistent indicator of ownership implies that the same entity maintains control of the host, regardless of whether the C2 label is actively assigned. This shows why it's important to pay attention to long-term signs of who controls a host, not just when it happens to be marked as a C2.

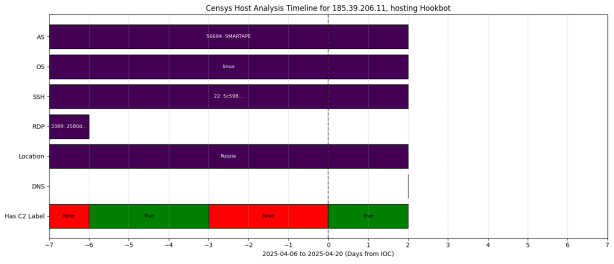

Fig. 2: Temporal analysis of network and system attributes for IP 185.39.206.11 displaying potential C2 activity and indicators of ownership

3) Case Study 3: Covenant host 152.42.251.12: Figure 3 takes a look at an instance of a previously benign host flipping over to a malicious host. In this graph, we can see that during the initial time spans, the host is tagged with specific DNS hostnames, consistent SSH keys, and identified as running Linux. After a brief period of downtime, however, we observe the host being reprovisioned—now running a different operating system, presenting a new RDP certificate, and tagged as hosting a C2 service.

It is important to note that this host belongs to an autonomous system operated by DigitalOcean, a widely used cloud service provider [7]. This figure highlights how malicious actors may leverage large-scale cloud platforms to "blend in" with benign users. It is also possible that DigitalOcean reallocates IP addresses when instances are rebooted or rotated at set intervals. For example, in AWS, public IP addresses assigned to an instance are released when the instance is stopped and started again, unless an Elastic IP is used [1].

Whatever the case may be, this demonstrates the importance of considering how IP addresses are provisioned in cloud environments, as changes in IP usage can complicate attribution and detection efforts.

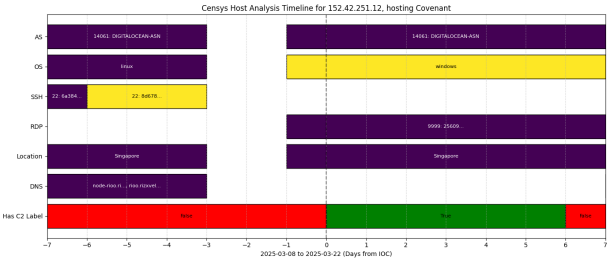

Fig. 3: Temporal analysis of network and system attributes for IP 152.42.251.12 displaying potential C2 activity and indicators of ownership

B. Aggregate Results

Using the same data that supported our timeline analysis, we also computed aggregate statistics to compare behaviors across different malware families. Specifically, we examined the average lifetimes of DNS records, RDP certificates, and SSH keys. For these calculations, we included only the time spans that overlapped with either the IOC timestamp from ThreatFox or periods when the host was marked as operating a C2 service. This filtering ensured we focused solely on time frames when the host was likely malicious, avoiding noise from potentially benign periods.

One limitation of this approach is the restricted observation window: we only had access to data spanning seven days before and after each IOC. As a result, some lifetime metrics may be truncated. We address this in the future work section and emphasize the importance of expanding the scan window—especially for high-prevalence malware like HookBot and AsyncRAT.

Metric Combination	AsyncRAT	Covenant	ERMAC	HookBot	SectopRAT	Aggregate
C2 Label	8.00/8.38 (13)	8.00/6.79 (29)	5.00/5.46 (28)	8.00/6.31 (13)	N/A	6.00/6.52 (83)
SSH Host Key	12.00/12.00 (2)	12.00/10.80 (20)	5.00/6.04 (23)	3.00/5.29 (7)	6.00/9.00 (3)	8.00/8.05 (55)
RDP Certificate	15.00/12.69 (13)	9.00/9.33 (3)	3.00/4.67 (3)	N/A	15.00/15.00 (1)	11.00/11.10 (20)
DNS Record	1.00/3.71 (17)	1.00/3.90 (29)	1.00/2.71 (28)	1.00/2.08 (13)	1.00/1.00 (5)	1.00/3.09 (92)
OS Type	15.00/13.71 (17)	14.00/11.86 (29)	5.50/6.50 (28)	10.00/10.38 (13)	10.00/10.00 (5)	10.00/10.26 (92)
C2 Label or SSH Host Key	9.00/9.00 (1)	12.00/11.10 (20)	7.00/7.35 (23)	9.00/7.71 (7)	N/A	9.00/8.90 (51)
C2 Label or RDP Certificate	15.00/13.25 (16)	15.00/14.55 (29)	15.00/13.61 (28)	15.00/13.54 (13)	15.00/15.00 (1)	15.00/13.86 (87)
C2 Label or DNS Record	8.00/7.88 (17)	8.00/7.97 (29)	5.00/6.50 (28)	8.00/6.85 (13)	1.00/1.00 (5)	7.00/6.97 (92)
C2 Label and SSH Host Key	4.00/4.00 (1)	8.00/7.05 (20)	3.00/4.35 (23)	3.00/4.57 (7)	N/A	4.00/5.43 (51)
C2 Label and RDP Certificate	8.50/9.30 (10)	4.00/4.67 (3)	2.00/2.67 (3)	N/A	N/A	5.50/7.19 (16)
C2 Label and DNS Record	1.00/3.15 (13)	1.00/2.72 (29)	1.00/1.68 (28)	1.00/1.54 (13)	N/A	1.00/2.25 (83)
C2 Label and OS Type	8.00/8.23 (13)	8.00/6.45 (29)	4.00/5.14 (28)	8.00/5.54 (13)	N/A	5.00/6.14 (83)

TABLE I: Lifetime (days) of Malware Families by Metric Combination (`<median>/<mean> (N)`) . Each cell represents the median and mean lifetimes observed looking at the specific combination of metrics, as well as the number of data points from our dataset. For example, AsyncRAT, when just basing the lifetime off of the existence of the Censys C2 label, has a median lifetime of 1.50 days, a mean lifetime of 3.83 days, with 12 hosts in our dataset having the Censys C2 label at the IOC timestamp. We only measured a 15-day span, so a lifespan of 15 implies that the service showed no indications of being taken down during our analysis period.

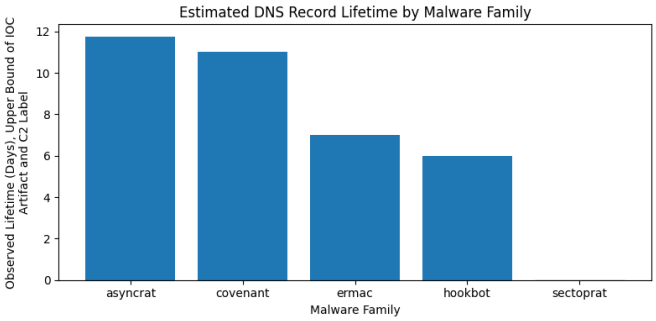

Fig. 4: Mean DNS record lifetimes (in days) for selected malware families, pulled from a fourteen day query time span.

Figure 4 shows how long DNS records associated with malicious hosts tend to persist. Notably, Covenant and AsyncRAT have average lifetimes close to eleven days, while others cluster around seven, with SectopRAT at the bottom at zero days. These differences provide insight into how various malware families use DNS infrastructure, and how long this ownership signal might persist once a DNS record appears.

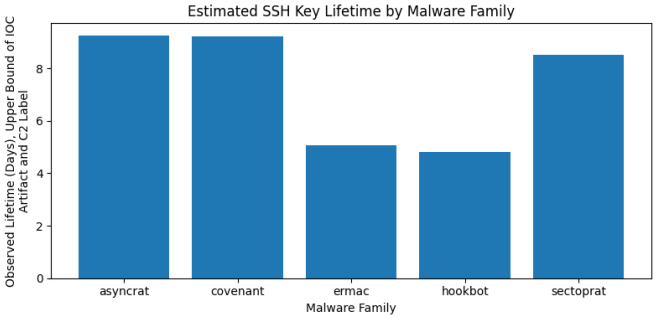

Fig. 5: Mean SSH public key lifetimes (in days) for selected malware families, pulled from a fourteen day query time span.

Figure 5 presents similar data for SSH public keys. Here again, Covenant, AsyncRAT, and SectopRAT exhibit longer key lifetimes compared to ERMAC and HookBot. This suggests a more stable or prolonged control over the host infrastructure in some families, whereas others may rely on more ephemeral setups. Since SSH keys are strong indicators of ownership, this distinction may reflect differing operational strategies between malware operators.

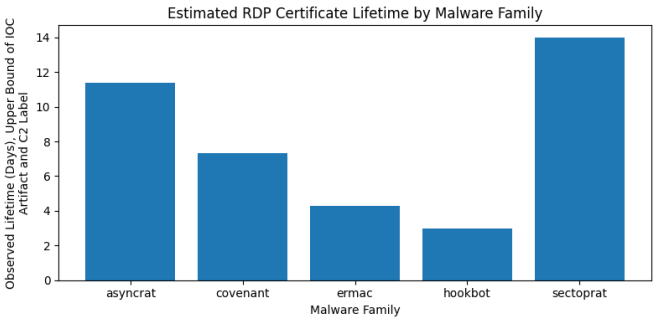

Fig. 6: Mean RDP certificate lifetimes (in days) for selected malware families, pulled from a fourteen day query time span.

Figure 6 can draw similar conclusions for the RDP data, where longer lifetimes suggest more stable control over the infrastructure. RDP certificates are a similarly strong indicator of ownership, so once again we might see differing hosting strategies between malware owners here as well.

The timeline analyses and aggregated metrics highlight the complex, often inconsistent, behavior of C2 infrastructure in the wild. Some servers cycle in and out of activity, others remain under consistent ownership even when not actively labeled as malicious, and some transition rapidly from benign to malicious use. By taking a more holistic view of a host's lifecycle, defenders can make more informed decisions about when to treat an IP address as a continuing threat and when it might be safe to let it go.

V. Recommendations & Conclusion

Malware incorporating C2 infrastructure will continue to be a difficult threat to defend for the foreseeable future. Due to their nature, the best that network operators can do is block communication with hosts flagged for malicious C2 activity. However, C2 servers tend to have substantially short and sporadic lifetimes, which makes IP address blocking non-trivial. We do not want to have a particular IP on a blocklist that only hosted malicious services for a few days long ago in the past—an eventual reassignment of an IP to a legitimate service would be unnecessarily blocked! This is why we need to take a dynamic approach to blocking IPs, which includes understanding how these C2 servers behave in the wild and devise some heuristic for when a particular malicious host is "safe" again (i.e. should not be on a blocklist).

As shown in Table I, the lifespans of malicious hosts vary greatly depending on both the malware family and the metrics considered. For instance, looking at the prevalence of the Censys C2 label alone, AsyncRAT-tagged hosts had an observed median lifetime of 8 days, while ERMAC-tagged hosts had an observed median lifetime of 5 days. When we look at different metrics, along with their union and intersection, the observed lifespan drastically changes, highlighting the nuanced nature of each C2 malware family's lifecycle.

It is essential to understand that a larger data set with more hosts and a longer observation window can greatly boost the confidence of a suggested heuristic—we simply did not have the resources to do this, and we leave it up to future work. Additionally, understanding the current behavior of malware operators is extremely relevant when acting upon our heuristic, as they tend to evolve their strategies rapidly.

We proceed to suggest a low-confidence heuristic for each malware family in our dataset:

- **AsyncRAT** — Block IPs labeled as C2 for approximately 8 days from initial labeling; monitor SSH host key changes closely, and extend if the same key remains at the time of labeling.
- **Covenant** — Block IPs labeled as C2 for 8 days from initial labeling; extend to 12 days from initial SSH host key observation if the same key persists; if RDP certificate or OS type is unchanged, extend up to 14 days from when first observed.
- **ERMAC** — Block IPs labeled as C2 for 5 days from initial labeling; if SSH host key remains stable, extend up to 7 days from initial key observation.
- **HookBot** — Block IPs labeled as C2 for 8 days from initial labeling; if SSH host key persists, extend up to 9 days from initial key observation.
- **SectopRAT** — Due to limited data, preliminary recommendations suggest blocking IPs for 6 days from initial SSH host key observation; if OS type remains unchanged, extend up to 10 days from when first observed.

VI. Future Work

Despite our comprehensive approach, several inherent limitations remain. We note them here and offer suggestions for overcoming the limitations in future work.

- **Attacker Deception** — Attackers might deliberately rotate certificates or alter DNS records to create false perceptions of legitimate ownership, complicating lifecycle tracking.
- **Incomplete Scanning Coverage** — While Censys' scanning coverage is extensive, complete and continuous global coverage is never entirely guaranteed, potentially missing transient malicious activity.
- **Cloud Hosting Inconsistencies** — Cloud providers' dynamic IP assignment and policy variations introduce ambiguity into interpreting IP lifecycle signals, potentially skewing heuristic accuracy.
- **Lack of Adjacent Service Analysis** — Our current heuristic does not incorporate analysis of adjacent service tags. This could be a valuable addition, as the presence of related services—such as databases, web servers, or file-sharing services—may suggest that the C2 remains active even if the C2-specific indicators have stopped responding to scanning packets. Monitoring these adjacent services could improve detection of persistent or re-emerging C2 activity.
- **Limited Scanning Time Span** — Our heuristic calculations are based on a fourteen-day window centered around when ThreatFox tagged these hosts as malicious. As a result, we only capture activity within that specific time frame. It's likely that some timeline metrics are truncated due to this restricted window. We chose the fourteen day window as a good faith estimate of the amount of time we would want to look at; that being said, future research should collect a broader range of scanning data to gain deeper insight into malicious host behaviors.

Acknowledgment

The authors would like to thank Aidan Holland and Ariana Mirian of Censys for their invaluable guidance in structuring this project, identifying C2 frameworks for investigation, and providing expertise on effectively leveraging the Censys dataset.

References

[1] Amazon Web Services. *Amazon EC2 Instance Addressing - Public IPv4 Addresses*. 2025. URL: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-instance-addressing.html#concepts-public-addresses (visited on 04/23/2025).

[2] *AsyncRAT (Malware Family) - Malpedia*. https://malpedia.caad.fkie.fraunhofer.de/details/win.asyncrat. Accessed: 2025-04-22.

[3] Rohith Cheerala and Gagandeep Kaur. "A Comprehensive Study on Malware Detection and Prevention Techniques used by Anti-Virus". In: Apr. 2021, pp. 429–434. DOI: 10.1109/ICIEM51511.2021.9445322.

[4] Ryan Cobb. *cobbr/Covenant: GitHub*. https://github.com/cobbr/Covenant. Accessed: 2025-04-22.

[5] Ali Davanian, Michail Faloutsos, and Martina Lindorfer. "C2Miner: Tricking IoT Malware into Revealing Live Command & Control Servers". In: *Proceedings of the 19th ACM Asia Conference on Computer and Communications Security*. ASIA CCS '24. Singapore, Singapore: Association for Computing Machinery, 2024, pp. 112–127. ISBN: 9798400704826. DOI: 10.1145/3634737.3644992. URL: https://doi.org/10.1145/3634737.3644992.

[6] Ali Davanian and Michalis Faloutsos. "MalNet: a binary-centric network-level profiling of IoT malware". In: *Proceedings of the 22nd ACM Internet Measurement Conference*. IMC '22. Nice, France: Association for Computing Machinery, 2022, pp. 472–487. ISBN: 9781450392594. DOI: 10.1145/3517745.3561463. URL: https://doi.org/10.1145/3517745.3561463.

[7] DigitalOcean, LLC. *DigitalOcean: Cloud Hosting for Developers*. 2025. URL: https://www.digitalocean.com/ (visited on 04/23/2025).

[8] *ERMAC (Malware Family) - Malpedia*. https://malpedia.caad.fkie.fraunhofer.de/details/apk.ermac. Accessed: 2025-04-22.

[9] Mitsuhiro Hatada and Matthew Scholl. "Method for Effective Measurement, Labeling, and Classification of Botnet C2s for Predicting Attacks". en. In: 27th Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA, 2020-02-23 2020. URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=929437.

[10] HelpSystems. *Cobalt Strike*. 2025. URL: https://www.cobaltstrike.com (visited on 02/26/2025).

[11] Aidan Holland. *fingerprints.yaml*. https://github.com/censys-workshop/threatfox-censys/blob/main/fingerprints.yaml. 2023. (Visited on 02/26/2025).

[12] Vivek Jain et al. "C2Store: C2 Server Profiles at Your Fingertips". In: *Proceedings of the ACM on Networking* 1 (2023), pp. 1–21. URL: https://api.semanticscholar.org/CorpusID:265509949.

[13] Joshua Kamp and Alberto Segura. *From ERMAC to Hook: Investigating the technical differences between two Android malware variants*. https://www.nccgroup.com/us/research-blog/from-ermac-to-hook-investigating-the-technical-differences-between-two-android-malware-variants/. Accessed: 2025-04-22.

[14] *SectopRAT (Malware Family) - Malpedia*. https://malpedia.caad.fkie.fraunhofer.de/details/win.sectop_rat. Accessed: 2025-04-22.

[15] Abhishek Sidhardhan, Keerthana S, and Jinesh M. Kannimoola. "Weaponizing Real-world Applications as C2 (Command and Control)". In: *2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA)*. 2023, pp. 458–463. DOI: 10.1109/ICIDCA56705.2023.10100279.

[16] Tdragon6. *Supershell*. 2025. URL: https://github.com/tdragon6/Supershell (visited on 02/26/2025).

[17] Censys Team. *Censys: Internet-wide scan data*. 2025. URL: https://censys.io (visited on 02/26/2025).

[18] ThreatFox. *ThreatFox: Sharing of malicious URLs, IPs and hashes*. 2025. URL: https://threatfox.abuse.ch/ (visited on 02/26/2025).