Have you SYN what I see? Analyzing TCP SYN Payloads in the Wild

Dario Ferrero
d.ferrero@tudelft.nl
Delft University of Technology
Delft, The Netherlands

Enrico Bassetti
e.bassetti@tudelft.nl
Delft University of Technology
European Space Agency
Delft, The Netherlands

Harm Griffioen
h.j.griffioen@tudelft.nl
Delft University of Technology
Delft, The Netherlands

George Smaragdakis
g.smaragdakis@tudelft.nl
Delft University of Technology
Delft, The Netherlands

Abstract

TCP SYN packets are typically meant to initiate a three-way handshake for new connections and do not carry a payload. The only exception, according to the standards, is TCP Fast Open, where data is transmitted as TCP SYN payload.

In this paper, we perform an empirical analysis of other cases where TCP SYN carries a payload. We utilize a large passive and a reactive network telescope to collect pure TCP SYN packets over two years. Our analysis shows that around 75% of these payloads are HTTP GET requests either for potentially censored content performed by researchers and activists originated by a relatively small number of IPs. We also observe scouting and intrusion attempt activity related to port 0, operating systems, middleware, and edge router vulnerability exploitation. We make our data and methodology publicly available as we want to raise awareness of this type of TCP SYN that typically goes unnoticed.

CCS Concepts

• **Security and privacy → Network security**.

Keywords

Network Telescope, Internet Scanning

ACM Reference Format:
Dario Ferrero, Enrico Bassetti, Harm Griffioen, and George Smaragdakis. 2025. Have you SYN what I see? Analyzing TCP SYN Payloads in the Wild. In *Proceedings of the 2025 ACM Internet Measurement Conference (IMC '25), October 28–31, 2025, Madison, WI, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3730567.3764498

1 Introduction

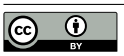
When establishing a Transmission Control Protocol (TCP) connection, a three-way handshake has to be completed between client and server before the delivery of data payloads can begin. Through the information exchanged within the initial packets, two remote operating systems (OS) can define endpoints with port numbers, synchronize sequence numbers, and negotiate connectivity options.

The TCP protocol standard does not prohibit appending payloads to the initial two messages (SYN and SYN-ACK). The only standardized instance of Zero Round Trip Time (0-RTT) data delivery is when using TCP Fast Open (TFO) [7]. TFO is an extension to speed up the opening of successive TCP connections between two endpoints by using a TFO cookie (a TCP option), i.e., a cryptographic cookie stored on the client and set upon the initial connection with the server. Other general-purpose transport layer network protocols, e.g., QUIC, enable 0-RTT for faster connection establishment following the TFO paradigm [23].

Network telescopes, i.e., reachable but inactive IP address subnets, have been used over the years by researchers to study Internet phenomena, e.g., malware propagation [11, 28–30, 35], scanning activity [14, 17], network misconfiguration [3], IPv4 space utilization [10], deployment of protocols, e.g., QUIC [31], and connectivity outages [12]. Internet traffic observed "in the wild" by network telescopes offers a unique view of unsolicited traffic, coined "Internet Background Radiation" (IBR) [3, 38], presenting characteristics which often do not match protocol standards, either because of misconfigurations or intentional bypass of the Operating System's networking stack to send custom-crafted packets. Apart from the traditional passive network telescopes, reactive telescopes such as Spoki [19] and DScope [34] have been recently proposed and studied, which provides new insights into unwanted and attack traffic on the Internet.

In this paper, we utilize both passive and reactive telescopes, and the associated IBR traffic, to study a new Internet phenomenon, namely, unsolicited TCP SYN packets carrying payloads. Our main contributions can be summarized as follows:

- We investigate a category of Internet Background Radiation traffic that, to the best of our knowledge, has never been reported in previous measurement studies, i.e., TCP SYN payloads.
- We leverage two years worth of traffic data collection by a large passive network telescope to characterize 95% of all SYN traffic carrying data payloads, providing insight and potential explanations to events and trends recorded. We observe and report on, among others, scans from research institutions, port 0 scans, and probing potentially related to vulnerability disclosures. We complement our study with data collected by a reactive telescope.



IMC '25, Madison, WI, USA

ACM ISBN 979-8-4007-1860-1/2025/10
https://doi.org/10.1145/3730567.3764498

- We determine whether the observed SYNs with payload can be linked to Operating System fingerprinting by setting up a virtualized testing environment and replaying the observed TCP SYNs with payloads to different types of OSes.

2 Related work

The majority of traffic collected by a Network Telescope consists of TCP SYN packets [17], and can be explained by more or less targeted port scans. TCP SYN packets are meant to initiate the three-way handshake for a new connection. During this initial phase, while allowed to, the first packet is not expected to carry any payload data [15], the delivery of which would need to wait for the SYN to be acknowleged and the connection established.

As stated before, the only exception to this is TCP Fast Open (TFO) [7], although this extension is still an experimental draft and the rate of its adoption is unclear. In 2015 Mandalari et al. [26] measured the availability of TFO across Internet paths by leveraging a crowdsourced setup of users across several countries and ISPs, showing that more than half of the connections attempted while enabling TFO did not reach the end server due to middlebox interference.

To the best of our knowledge, no previous work has specifically analyzed IBR events of TCP SYN packets carrying data payloads. Because of this being a rare case of the TCP protocol, it's relatively complex to find specific references to its occurrence, analysis, or adoption for any type of study. In terms of active measurement studies, Langley [22] in 2008 performed a first internet scan of one million IP addresses from a top domain list, showing that out of the hosts responding to regular SYN packets, 9% would not reply to a SYN carrying a payload, while the remainder would just acknowledge the SYN and not the data. More recently, Raman et alii [36] reported that "38% of SYN packets on port 80 contained an HTTP request payload", matching the same events we report on in Section 4.3.1, but without providing further insight into them. From the point of view of a server, the receiving host's OS kernel, in absence of a valid TFO Cookie within the TCP Options, should ignore the payload and either acknowledge with a SYN/ACK or reject with a RST [15] the SYN depending on whether there's a service listening. Processing of the payload prior to connection establishment might occur in some form of middleboxes (e.g. Firewalls) in case of targeting specific ports. Luchs and Doerr [24] performed an analysis of IBR events originating or targeting TCP port 0, which has often been associated with firewall evasion. This work does not just focus on TCP SYN packets, giving insight for example into DDoS backscatter originating from source port 0, i.e., the result of attacks targeting this port as the destination by spoofing the telescope's addresses. For the scope of our study, the authors don't mention whether data payloads are observed within the collected SYN packets. As explained in Section 4.3.1, the only case where we attribute these events to previous research relates to the works of Bock et al. [4, 5, 18, 32, 33], where TCP SYNs with payloads are explicitly sent to measure censorship evasion strategies from within censored states. In particular, with [4] the authors demonstrated how SYN packets carrying payloads can not only be a vector triggering interference by censors, but can potentially be used to performed TCP-based amplification attacks by exploiting non-TCP compliant middleboxes.

3 Datasets

We base our study on a large Passive Network Telescope located within an enterprise network and obtained from the combination of three non-contiguous /16 IPv4 subnets located in Europe, amounting to a total of ≈65,000 addresses monitored. The size of our datasets can be seen in Table 1. In this paper we focus exclusively on TCP SYN data, of which the packet and source counts are shown. Overall, from the total 292.96 billion TCP SYNs collected over the two years of measurements and originating from 17.95 million sources, we report that 1% of all observed IP addresses contact this network with more than 200 million TCP SYN packets carrying application data. Studies investigating TCP-based telescope artifacts usually do not provide insight into this layer, and as such their absence is usually listed as a limitation. In this work we show that it is still possible to observe application-layer data through the deployment of passive telescopes. These probes are present throughout the two-year measurement's duration, making them a persistent and relevant event in today's Internet.

Next to the Passive Telescope, we deploy a Reactive Telescope similar to Spoki [19] for 3 months at the end of our passive experiment, locating it within one of the providers contributing to the telescope, although in a separate network from the passive ones. As implied by its name, a Reactive Telescope replies to each incoming TCP SYN with a SYN-ACK, this way probing the scanners for connection establishments and giving an initial visibility into the application layer requests incoming from traffic in the wild. Similar to related work that performs this type of measurement, our system is responsive across the entire TCP port range [19]. The goal of this reactive network space is to determine whether hosts sending these packets would continue their connection if it were established using the TCP handshake. The number of SYN packets (with data) and of unique scanner IPs for the Reactive Telescope deployment can also be seen in Table 1. It should be mentioned that the Reactive Telescope had been deployed for a different experiment and was designed to emulate a simple non-responsive TCP service. This means that no considerations were made regarding the payloads in the TCP SYNs, such as responding to a TFO Cookie request.

As TCP SYNs with embedded payloads are not a common part of the TCP protocol, the amount of traffic we observe with these characteristics is limited. Because of the uncommon nature of these packets, the size of our vantage point and duration of data collection contribute crucially to the amount of data available, as mentioned in previous studies leveraging large network telescopes for analyses of similarly rare events [24]. While in this work we only leverage our available address space, operating a vantage point of larger size would also improve the observability and of this type of traffic, and geographically diverse deployments might reveal contrasting patterns as well as improve generalizability.

4 Case Studies

During the two years of passive measurements our network telescope collected more than 200M TCP SYNs containing a data payload. These events present high variability and require case by

Table 1: Summary of TCP SYN Packets carrying a payload ("SYN-Pay"), as seen from each Passive (PT) and Reactive (RT) Telescopes during the measurement period.

	Telescope Size	Duration	# SYN Pkts	# SYN-Pay Pkts	# SYN IPs	# SYN-Pay IPs
PT	3x /16 (≈65,000 IPs)	Apr. '23 - Apr. '25 (2 years)	292.96B	200.63M (0.07%)	17.95M	181.18K (1.01%)
RT	1x /21 (≈2,000 IPs)	Feb. '25 - May '25 (3 months)	6.82B	6.85M (0.10%)	3.28M	4.17K (0.13%)

Table 2: Shares of the total TCP SYNs w/ Payloads presenting combinations of well-known fingerprints.

High TTL	ZMap IP ID	Mirai SeqN	No TCP Options	% Packets
✓	–	–	✓	55.58 %
✓	✓	–	✓	23.66 %
–	–	–	–	16.90 %
–	–	–	✓	3.24 %
✓	–	–	–	0.63 %

Table 3: Payloads categories by identified protocol or service.

Type	# Payloads	# IPs
HTTP GET	168.23M	1.06K
ZyXeL Scans	19.68M	9.93K
NULL-start	9.35M	2.08K
TLS Client Hello	1.45M	154.54K
Other	4.98M	2.25K

case analyses to understand the intent behind them. In Section 4.1, we begin by verifying whether these packets are attributable to the adoption of less common standards or to Internet scanning by searching the TCP/IP headers for common patterns. In Section 4.2, we report on the effect of replying to such traffic in an attempt to elicit follow-up responses. Section 4.3 provides a more detailed analysis of the macro categories of payloads collected.

4.1 TCP Scanning Patterns and Fingerprints

Previous work investigating header fingerprints has allowed to track spread of botnets [1], selectively reply to scanners [19] and correlate probing campaigns [16]. Before diving deep into the content delivered within the subset of TCP SYNs under study, we provide an overall analysis of their header field values observed from the TCP/IP layers.

4.1.1 Adoption of TCP Standards. We begin by reporting on the distribution of TCP Options observed. Specific features of the TCP protocol, such as TFO [7], need to be requested by providing option data identified by a specific kind value. It is worth noting that only 17.5% out of the dataset of SYNs with payload carries some form of TCP Option, that is ≈36,000,000 packets. As observed in [19], this already shows an irregularity, since the most popular operating systems provide some option while initiating the TCP handshake. From this set, we observe that the vast majority of the options included belong to the set of those commonly adopted in the TCP Connection Establishment: (0) End of Option List, (1) No-Operation, (2) MSS, (3) Window Scale, (4) SACK Permitted, and (8) Timestamps. Packets including at least one TCP Option outside of this set comprise only 2% of those including any option (≈653,000) originating from ≈1,500 sources. Almost all of these packets are each limited to one TCP Option of a reserved kind number according to the IANA assignments [20], without presenting any relevant traffic pattern or recognizable payload protocol. The extremely limited inclusion of TCP Options excludes the explanation that this traffic could be requesting the adoption of less common TCP standards [7, 21], although the intent behind this final subset of packets remains unexplained. We notice that the TCP Fast Open Cookie option (34) appears only in ≈2,000 packets, ruling it out as a major contributor to these events.

4.1.2 Scanner Fingerprints. We want to determine whether the events observed are due to potential misconfigurations or are actually due to internet scanning attempts. To determine this, we search the headers for common fingerprints for "Irregular SYNs" as first introduced in [19]. These heuristics can indicate a likely stateless packet generation, as well as link to popular scanning tools and malware variants performing such reconnaissance. Table 2 shows the percentages of SYN-Payload traffic presenting combinations of irregularities such as a lack of TCP Options, Time To Live higher than a value of 200, Sequence Number equal to the Destination IP address (which can be linked to Mirai botnet traffic [1]), and an IP ID field with value 54321, the default for the ZMap scanning tool [14]. It is worth observing that 83.1% of this traffic presents at least one of these irregularities, with more than 75% of packets both having a high TTL and not including TCP Options in the connection request phase. The ZMap fingerprint can be observed in 23.66% of packets, pointing to explicit usage of scanning tools. Surprisingly, we do not see the original Mirai fingerprint in this dataset, while it is known to be still actively requested in basic TCP SYN scans [19]. Finally, it is worth noting that ≈ 97,000 of the hosts sending SYNs with payloads do not send any regular TCP SYN packet during their recorded activity. The entire presence of traffic irregularities can lead to the conclusion that this traffic has been generated statelessly as part of Internet scans.

4.2 Reactive Telescope Interactions

Leveraging the Reactive Telescope deployment introduced in Section 3, we investigate whether scanners sending SYNs with payload continue their interactions after receiving SYN-ACK replies. From the RT dataset shown in Table 1, we are surprised to observe that out of the collected ≈6.85M packets, only ≈500 are followed by an ACK packet completing the handshake, albeit without payload, while only few additional payloads with no particular protocol are delivered. For the almost entirety of recorded traffic, SYNs carrying data are followed by a re-transmission of the same packet. Two clarifications have to be made about our Reactive Telescope deployment. First, as already mentioned in Section 3, this reactive component was not originally intended for the SYN+Payload case, meaning that while we do acknowledge the data payload within the SYN-ACK's sequence number, we do not reply with any application-layer data or TCP Options. Second, to simplify the data collection for the scope of this secondary project, we filtered inbound traffic

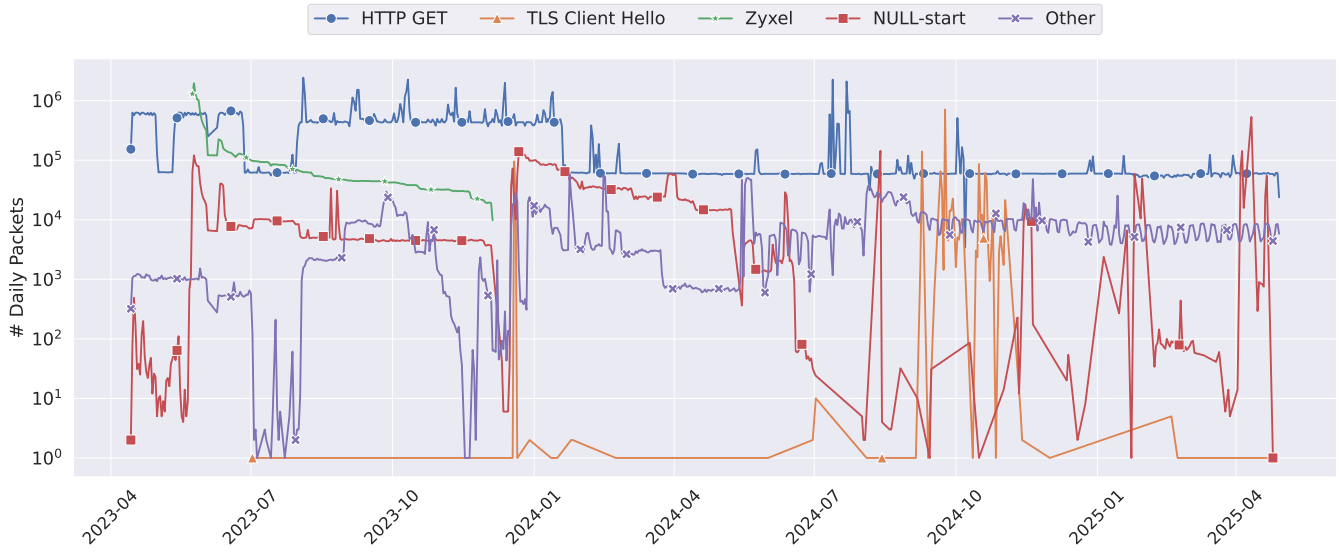


Figure 1: Daily # of Packets per Payload Type.

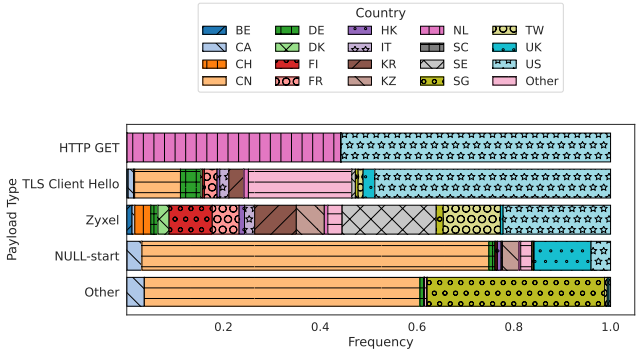


Figure 2: Shares of origin countries for each payload type.

to only accept TCP traffic including SYN or ACK flags set. This excludes TCP RST packets, which can be seen as a result of two-phase scanning [19]. While deploying a system providing higher interaction to these probes would make an interesting future work, we can only conclude that these scans are limited to a first-packet basis only, without attempting to complete handshakes or deliver data beyond that. Assuming that scans might aim at triggering unexpected end-host behavior, delivering representative data in our replies is a challenge that requires further insight into the payload contents, as we further explore in the next Section 4.3.

4.3 Payload Categories

To gain insight into the nature of the data observed in the TCP SYN payloads, we categorize them by their apparent protocol or service, as shown in Table 3. These listed categories have been determined either by inspection of the initial payload bytes (for HTTP and TLS) or by identification of more peculiar sub-patterns in the data, as detailed in the following Sections. Across the entire measurement period, we find that HTTP `GET` requests dominate in volume, with over 168 million payloads, albeit originating from a relatively small number of 1.06K IPs sources. In contrast, TLS Client Hello messages, although much fewer in total volume, are issued by a significantly larger and more diverse set of sources (154.54K IPs). Figure 1 shows the packets sent in every category. Notably, both TLS and Zyxel scanning events are temporally constrained, appearing only during specific intervals within the measurement period. Zyxel-related traffic, with nearly 20 million packets and close to 10 thousand distinct sources, is geographically distributed, originating from many countries as shown in Figure 2. We outline our observations in this section with several case-studies of the main groups of traffic observed.

4.3.1 HTTP Requests. As shown in Table 3, over 75% of all observed SYN payloads are identifiable as HTTP `GET` requests. This category represents the only persistent baseline of traffic present throughout the entire two-year measurement period at the Passive Telescope, as illustrated in Figure 1. The vast majority of these `GET` requests are minimal in form: targeting the root path, lacking body content, and omitting the `User-Agent` header. The latter is particularly noteworthy, as popular scanning frameworks such as ZGrab [37] typically insert a distinctive `User-Agent` string by default.

Among the few distinguishing characteristics in this traffic, we find notable variation in the `Host` header, with 540 unique domains recorded. The most frequently requested domains fall into categories such as adult content, VPN providers, torrenting services, social media platforms, and news outlets. While analyzing the sources of these requests, we identify a particularly strong outlier: 470 of the domains are queried exclusively by a single IP address associated with a major U.S. university, determined through reverse DNS lookups. Although no corresponding publication could be found, this likely represents an internal research effort.

Excluding this outlier, the remaining 70 domains are more evenly distributed across approximately 1,000 IP addresses, each issuing up to seven different domain requests. A curated selection of these domains is listed in Appendix B.

A second notable anomaly is the frequent appearance of requests for the path `/?q=ultrasurf`. These requests account for over half of all HTTP `GET` requests during a sustained period from April 2023 to February 2024, while only including two of the aforementioned `Host` domains (`youporn.com` and `xvideos.com`) and originating from three IP addresses belonging to a cloud hosting provider in the Netherlands. The specific path and nature of the target domains suggest the traffic may be linked to censorship evasion activities. Ultrasurf is one of the earliest tools designed to bypass Chinese internet censorship, using an obfuscated proxy network [2]. However, the presence of the term "ultrasurf" in a visible query string is unexpected, as it would likely trigger filtering rather than evade it.

These requests may originate from the Geneva research framework [5], which is designed to evolve packet sequences that evade censorship in adversarial networks. Geneva has been shown to use specific patterns (such as including the string `ultrasurf` in HTTP queries or referencing domains like `youporn.com`) to intentionally provoke censorship mechanisms [18]. Notably, some Geneva strategies involve sending a clean SYN followed by a SYN packet with payload, matching what we observe. To validate this hypothesis, we performed IP-to-country mapping using the historical MaxMind GeoLite2 dataset [27], with results shown in Figure 2. Contrary to expectations, the traffic originates exclusively from the United States and the Netherlands, not from within censored networks that have been subject to studies in related work [33]. What's more, most of the studies involving Geneva date prior to the start of our data collection period, although we do not exclude potential continuation of experiments.

Given that SYN+Payload has been documented as an effective censorship evasion method [5], we would expect to observe at least some responses from within censored countries. This absence could suggest survivorship bias, though it does not explain why such probes would be directed at entirely unresponsive networks like our Passive Telescope. These doubts may be answered by a very recent work by Nourin et al. [32], which builds on Geneva and systematically targets inactive subnets in global internet scans to study censorship interference. While the authors do not provide a period of the scans performed, the novel methodology of this work could explain why this type of traffic targets our darknet. The volume of internet scans, although visibly higher than any other event from Figure 1, is still relatively small when compared to the daily baseline of 100 million to 1 billion TCP SYN packets, amounting to less than 100 probes per monitored destination IP address.

4.3.2 ZyXel and NULL-start. The second largest group of collected SYN payloads consists of data segments containing one or more printable binary file paths. These payloads are always 1,280 bytes long, begin with at least 40 consecutive null bytes, and exhibit a consistent internal structure. Following the initial padding, we observe three to four embedded, well-formed IPv4 and TCP header pairs, separated by additional null bytes. Upon inspection, the source and destination IP addresses within these encapsulated headers are frequently set to `0.0.0.0` or fall within the `29.0.0.0/24` subnet, a block assigned to the U.S. Department of Defense and presumably used as a placeholder. After this series of internal headers, the remaining bytes follow a type-length-value format that enumerates up to 26 file path strings per payload, suggesting a deliberate encoding scheme. A sample visualization of this final part of the payload is provided in Appendix D.

While some names point to common binaries found on generic Unix-based servers (e.g. `httpd,syslog-ng`), and many appear to be truncated, we noticed how a significant portion included references to `Zyxel`, a popular brand of networking appliances and services. A list of the file paths most often encountered is available in Appendix C. Because of the frequency of these references, as well as the trend displayed in Figure 1 following the pattern of a slowly decreasing event-peak over several months, we investigate whether this campaign could be related to the release of a CVE [9] targeting specific Zyxel products. We search all available CVEs released one month before and after the beginning of this scanning peak. Most of the disclosed vulnerabilities within that time period fall into the categories of post-authentication command injections, cross-site scripting (XSS), or affect Common Gateway Interfaces. We found no explicit reference to these file paths or payload format when searching for a matching CVE or security advisory, and cannot precisely correlate this event with specific exploitation attempts, nor why these payloads are present in TCP SYN packets.

An important addition is that the vast majority of the observed Zyxel scans are targeting TCP port 0. Several studies in the past have reported on the characteristics of port 0 traffic in the wild [6, 24, 25] by making use of large network telescopes and IXP-level traffic sampling, although none of them reported insights into the data payloads carried by SYN traffic. When observing the remainder share of traffic directed to port 0, we recognize a secondary macro-category of long payloads also beginning with several null bytes. As shown in Table 3, we refer to this set as *"NULL-start"*. It is visible from Figure 1 that the initial trend of NULL-start payloads matches the one of the Zyxel scans. Despite this, for this particular subset we do not observe any listing of binary file paths, nor any discernible overall data structures or printable substrings aside from the initial null bytes. We analyse properties of the payloads to search for common patterns and find that 85% of them have fixed length of 880 bytes. For this subset, the length of the initial NULL-bytes varies between 70 and 96 bytes, although when comparing the initial non-null byte sequences that follow we do not observe any common sub-pattern, disregarding potential alignment attempts for buffer overflow attempts.

4.3.3 TLS Client Hellos. TLS Client Hello messages are the most diverse considering IPv4 source addresses, with 154.54K distinct sources as listed in Table 3. As shown in Figure 1, most of this traffic is concentrated within a short time window, similar to the Zyxel scans. However, the irregular delivery pattern suggests that this is not part of a coordinated measurement study, such as the HTTP `GET` traffic. Over 90% of TLS payloads are malformed, presenting a Client Hello length set to zero, although additional data follows in all cases. The sources are widely distributed across IPv4 /16 subnets, as also reflected in Figure 2. Given the sudden appearance, high volume, and broad address spread, and that these sources do not

complete the TCP handshake when provided with a SYN/ACK, IP spoofing would be a plausible explanation for the spread of the sources sending these packets. The complete absence of Server Name Indication (SNI) fields makes it unlikely these messages relate to censorship evasion experiments [18], unless aimed at triggering interference during the TLS handshake phase.

4.3.4 Other Payloads. Out of the 200 million initial payloads collected within SYN packets, 5 million (2.5%) either do not appear to have a distinguishable byte format, making it hard to categorize them, or belong to separate sub-categories which are difficult to explain. For example, we record separate sets of payloads comprising single-byte values such as a null-byte or the letter 'A', both in upper and lower case. Combinations of smaller sets of cases makes the analysis more complex and increasingly granular, so for the scope of this paper we conclude our analyses within the payloads mentioned in Table 3. We do note, however, that the spread over countries from this category is limited (see Figure 2), indicating that there may not be much variation in the sources sending these unknown payloads.

5 OS Behavior

One of our hypotheses is that SYN payloads might be used to do some form of fingerprinting on the OS level. The default operating system behavior when receiving SYN packets with payloads remains largely undocumented. This lack of standardized behavior prompted us to develop a dedicated experimental framework to better understand how modern OSes handle such atypical traffic.

In our approach, we implement a virtualized emulation environment similar to that presented in [13]. Within this environment, we replay a representative sample of SYN payloads, covering each type identified in Table 3. This replay-based methodology allows us to systematically observe the responses produced by various operating systems when subjected to SYN packets carrying payloads.

Each emulated operating system is equipped with dummy services running behind a designated set of "control" ports. This configuration enables us to analyze whether the response to the TCP SYN changes in presence of a service running on a port when receiving TCP SYN packets with payload. We tested a sample of various ports (80, 443, 2222, 8080, 9000, 32061). For each of these ports, we analyzed the response of various operating systems both when a service is listening to the specified port, and when it is not.

Additionally, we replayed and analyzed traffic towards the TCP port 0; in this case, we studied the response of the operating system when receiving such packets. Note that no services can listen on TCP port zero; in many network stacks, this port number is used for special purposes (e.g., in Linux, binding to port zero effectively indicates to the OS to bind the socket to an unused port). RFC 6335 and IANA indicate that the port is "reserved" and shall not be used [8, 24].

In Table 4, we report the operating system and kernel versions we tested in our replay experiments. The resulting behavior is consistent between different systems: if there is no service running on a port, the network stack responds with a TCP-RST packet, acknowledging the payload present in the TCP-SYN. On the other hand, if there is a service running on the specified port, the resulting SYN-ACK do not acknowledge the payload, and the payload is not sent to the application. As all tested operating systems behave the same, we can rule out fingerprinting as potential motivation for this traffic.

Table 4: OS types and versions tested for SYNs with payloads.

Operating System	Kernel Version	Vagrant box version
GNU/Linux Arch	6.6.9-arch1-1	4.3.12
GNU/Linux Debian 11	5.10.0-22-amd64	11.20230501.1
GNU/Linux Ubuntu 23.04	6.2.0-39-generic	4.3.12
Microsoft Windows 10	10.0.19041.2965	2202.0.2503
Microsoft Windows 11	10.0.22621.1702	2202.0.2305
OpenBSD	7.4 GENERIC.MP#1397	4.3.12
FreeBSD	14.0-RELEASE	4.3.12

6 Conclusion and Discussion

In this work, we presented a longitudinal measurement and analysis of TCP SYN packets carrying payloads observed in unsolicited traffic reaching our passive and reactive telescope networks. While the inclusion of payload in SYN packets is permitted by the TCP specification, it is generally rare and often overlooked in measurement studies. To the best of our knowledge, this work constitutes the first investigation focused specifically on this class of Internet Background Radiation (IBR) traffic.

Leveraging two years of traffic data collected by a large network telescope, we systematically characterized over 95% of all SYN packets with data payloads observed in our dataset. Through classification of payload content and contextual analysis, we uncovered a variety of behaviors ranging from clearly identifiable research scans—some of which were conducted concurrently to our own study—to more ambiguous or potentially malicious activities. In particular, we report on scans targeting TCP port 0, and probes containing signatures related to `Zyxel` software, which may be indicative of probing for known vulnerabilities, misconfigured devices, or scanning strategies attempting to evade detection. These categories of traffic appear to fly under the radar of conventional monitoring solutions that discard or ignore payload-bearing SYNs.

Another contribution of this work is a systematic analysis of how different operating systems handle incoming SYN packets with payloads. To explore whether SYN with payloads could serve as a vector for OS fingerprinting or evasion, we set up a virtualized testbed and replayed representative samples against a range of operating systems. While our preliminary results did not reveal substantial behavioral differences across the network stacks tested, our findings underscore the need for more comprehensive evaluations including a broader diversity of OS versions, firewall middleboxes, and intrusion detection or prevention systems.

Our findings suggest that SYN with payloads constitute a subtle but meaningful dimension of Internet background radiation. They merit closer attention from both the measurement and security communities, and we hope our work provides a foundation for future studies in this direction. To foster future research in this area, we are making our dataset available upon request. We hope our work encourages further scrutiny into this niche yet revealing form of unsolicited Internet traffic and inspires more comprehensive monitoring approaches.

Acknowledgment

This research was supported by the European Commission under the Horizon Europe Programme as part of the projects SafeHorizon (Grant Agreement #101168562) and RECITALS (Grant Agreement #101168490). This work was supported by the Dutch Research Council (NWO) under the ADAPTIve project.

Appendix

A Ethics and Open Science

In this study, we deploy over a /21 IPv4 subnet a service responding to incoming connections. Our infrastructure does not send any data without a request being made, but could be tricked into sending traffic to a secondary host when receiving a message with a spoofed source. In this case, our infrastructure would send a reply to the spoofed source instead, effectively reflecting (but not amplifying) a request. This is, however, no different from a regular service on the Internet, where an adversary can also send a spoofed connection packet for which the reply is sent to another host. We do not observe any abuse of our infrastructure in terms of reflection attempts.

We intend to share the data collected during the measurements. Because of the sensitive nature of this data, and to ensure the privacy of networks and hosts that have interacted with our system (even though this is not user traffic) we will only share anonymized data publicly. To allow other researchers to completely reproduce our work we are open to share the full non-anonymized dataset on request [1]. This includes the full application-layer traces collected in our infrastructure.

B Popular Domains in HTTP `GET` Payloads

Table 5 shows a curated list of domain strings contained in the `Host` headers of HTTP `GET` requests described in Section 4.3.1. The top row domains in the table comprise 99.9% of the collected requests. Other than the two related to the `ultrasurf` queries, the other three (`www.youporn.com`, `www.freedomhouse.org` and `freedomhouse.org`) are often seen within the same `GET` request within duplicated Host headers.

C Zyxel Embedded File Paths

Table 6 shows the file paths frequently observed within payloads associated with Zyxel scans. Names are mostly related to regular unix and Zyxel-specific binaries, and are often truncated.

D Zyxel Payload Structure

Figure 3 illustrates the breakdown of the final part of a sample Zyxel scan payload, as introduced in Section 4.3.2. As previously stated, this data follows a serie of three to four TCP / IP headers internal to the original payload.

References

[1] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J. Alex Halderman, Luca Invernizzi, Michalis Kallitsis, Deepak Kumar, Chaz Lever, Zane Ma, Joshua Mason, Damian Menscher, Chad Seaman, Nick Sullivan, Kurt Thomas, and Yi Zhou. 2017. Understanding the Mirai Botnet. In *26th USENIX Security Symposium (USENIX Security 17)*. 1093–1110. https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis

[2] Jacob Appelbaum. 2012. *Technical Analysis of the Ultrasurf Proxying Software.* https://blog.torproject.org/blog/ultrasurf-definitive-review

[3] Karyn Benson, Alberto Dainotti, kc claffy, Alex C. Snoeren, and Michael Kallitsis. 2015. Leveraging Internet Background Radiation for Opportunistic Network Analysis. In *Proceedings of the 2015 Internet Measurement Conference* (Tokyo, Japan). Association for Computing Machinery, New York, NY, USA, 423–436. https://doi.org/10.1145/2815675.2815702

[4] Kevin Bock, Abdulrahman Alaraj, Yair Fax, Kyle Hurley, Eric Wustrow, and Dave Levin. 2021. Weaponizing Middleboxes for TCP Reflected Amplification. In *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, 3345–3361. https://www.usenix.org/conference/usenixsecurity21/presentation/bock

[5] Kevin Bock, George Hughey, Xiao Qiang, and Dave Levin. 2019. Geneva: Evolving Censorship Evasion Strategies. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 2199–2214.

[6] Elias Bou-Harb, Nour-Eddine Lakhdari, Hamad Binsalleeh, and Mourad Debbabi. 2014. Multidimensional investigation of source port 0 probing. *Digital Investigation* 11 (2014), S114–S123.

[7] Yuchung Cheng, Jerry Chu, Sivasankar Radhakrishnan, and Arvind Jain. 2014. TCP Fast Open. RFC 7413. https://doi.org/10.17487/RFC7413

[8] Michelle Cotton, Lars Eggert, Dr. Joseph D. Touch, Magnus Westerlund, and Stuart Cheshire. 2011. Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry. RFC 6335. https://doi.org/10.17487/RFC6335

[9] CVE Program. 2025. *Common Vulnerabilities and Exposures (CVE).* https://www.cve.org

[10] Alberto Dainotti, Karyn Benson, Alistair King, Bradley Huffaker, Eduard Glatz, Xenofontas Dimitropoulos, Philipp Richter, Alessandro Finamore, and Alex C. Snoeren. 2016. Lost in Space: Improving Inference of IPv4 Address Space Utilization. *IEEE Journal on Selected Areas in Communications (JSAC)* 34, 6 (2016), 1862–1876.

[11] Alberto Dainotti, Alistair King, kc Claffy, Ferdinando Papale, and Antonio Pescapè. 2012. Analysis of a "/0" stealth scan from a botnet. In *Proceedings of the ACM SIGCOMM Conference on Internet Measurement*.

[12] Alberto Dainotti, Claudio Squarcella, Emilie Aben, kc Claffy, Marco Chiesa, Michele Russo, and Antonio Pescape. 2011. Analysis of Country-wide Internet Outages Caused by Censorship. In *Proceedings of the ACM SIGCOMM Conference on Internet Measurement*.

[13] Edoardo Di Paolo, Enrico Bassetti, and Angelo Spognardi. 2023. A New Model for Testing IPv6 Fragment Handling. In *Proceedings of the 28th European Symposium on Research in Computer Security (ESORICS 2023)*.

[14] Zakir Durumeric, David Adrian, Phillip Stephens, Eric Wustrow, and J Alex Halderman. 2024. Ten Years of ZMap. In *Proceedings of the Internet Measurement Conference*.

[15] Wesley Eddy. 2022. Transmission Control Protocol (TCP). RFC 9293. https://doi.org/10.17487/RFC9293

[16] Harm Griffioen and Christian Doerr. 2020. Discovering Collaboration: Unveiling Slow, Distributed Scanners based on Common Header Field Patterns. In *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 1–9.

[17] Harm Griffioen, Georgios Koursiounis, Georgios Smaragdakis, and Christian Doerr. 2024. Have you SYN me? Characterizing Ten Years of Internet Scanning. In *ACM Internet Measurement Conference*. Madrid, Spain.

[18] Michael Harrity, Kevin Bock, Frederick Sell, and Dave Levin. 2022. GET /out: Automated Discovery of Application-Layer Censorship Evasion Strategies. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association, Boston, MA, 465–483. https://www.usenix.org/conference/usenixsecurity22/presentation/harrity

[19] Raphael Hiesgen, Marcin Nawrocki, Alistair King, Alberto Dainotti, Thomas C. Schmidt, and Matthias Wählisch. 2022. Spoki: Unveiling a New Wave of Scanners through a Reactive Network Telescope. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association, Boston, MA, 431–448. https://www.usenix.org/conference/usenixsecurity22/presentation/hiesgen

[20] Internet Assigned Numbers Authority. 2025. *Transmission Control Protocol (TCP) Parameters.* https://www.iana.org/assignments/tcp-parameters/tcp-parameters.xhtml Accessed on 2025-08-18.

[21] Julian Kirsch, Christian Grothoff, Jacob Appelbaum, and Holger Kenn. 2015. *TCP Stealth.* Internet-Draft draft-kirsch-ietf-tcp-stealth-01. Internet Engineering Task Force. https://datatracker.ietf.org/doc/draft-kirsch-ietf-tcp-stealth/01/ Work in Progress.

[22] Adam Langley. 2008. Probing the viability of TCP extensions. *Google, Inc., Tech. Rep* (2008).

[23] Adam Langley, Alistair Riddoch, Alyssa Wilk, Antonio Vicente, Charles Krasic, Dan Zhang, Fan Yang, Fedor Kouranov, Ian Swett, Janardhan Iyengar, Jeff Bailey, Jeremy Dorfman, Jim Roskind, Joanna Kulik, Patrik Westin, Raman Tenneti, Robbie Shade, Ryan Hamilton, Victor Vasiliev, Wan-Teh Chang, and Zhongyi Shi. 2017. The QUIC Transport Protocol: Design and Internet-Scale Deployment. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication* (Los Angeles, CA, USA) *(SIGCOMM '17)*. Association for Computing Machinery, New York, NY, USA, 183–196. https://doi.org/10.1145/3098822.3098842

[1] The dataset is available on request at https://delftintellab.com/projects/

Table 5: Most frequently requested domains within the Host header of HTTP `GET` requests.

pornhub.com	freedomhouse.org	www.bittorrent.com	www.youporn.com	xvideos.com
instagram.com	bittorrent.com	chaturbate.com	surfshark.com	torproject.org
onlyfans.com	google.com	nordvpn.com	facebook.com	expressvpn.com
ss.center	9444.com	33a.com	98a.com	thepiratebay.org
xhamster.com	tiktok.com	xnxx.com	youporn.com	jetos.com
919.com	netflix.com	twitter.com	reddit.com	1900.com
www.pornhub.com	plus.google.com	mparobioi.gr	youtube.com	www.roxypalace.com
www.porno.com	example.com	www.xxx.com	www.survive.org.uk	www.xvideos.com
coinbase.com	tt-tn.shop	telegram.org	csgoempire.com	cnn.com
empire.io	bbc.com	www.tp-link.com.cn	betplay.io	bcgame.li
www.tp-link.com	bet365.com	foxnews.com	dark.fail	www.mobily.com
www.bet365.com	xxx.com	betway.com	paxful.com	

Table 6: List of file paths within Zyxel scans and their frequency over the total packets related to these events.

File Path	Percentage
/compress/usr/sbin/celld	8.0%
/util/init	5.19%
/compress/usr/sbin/zylogd	5.18%
/compress/usr/sbin/syslog-ng	5.18%
/compress/usr/sbin/zwd	5.18%
/compress/usr/sbin/secu_reporter	5.18%
/compress/usr/sbin/uamd	5.18%
/compress/usr/sbin/myzyxel_registerd	5.12%
/compress/usr/sbin/myzyxel_checkd	4.35%
/compress/usr/sbin/myzyxel_fetchurld	4.35%
/compress/usr/sbin/zyssod	4.29%
/compress/usr/sbin/rumd	4.29%
/compress/usr/sbin/link_updown	4.16%
/compress/usr/sbin/sched	3.9%
/compress/usr/sbin/zyiptcrouted	3.84%
/compress/usr/sbin/generic_timer	3.67%
/compress/usr/sbin/policyd	3.1%
/compress/usr/sbin/firewalld	2.53%
/compress/usr/sbin/zld_cdrd	2.35%
/compress/usr/local/apache/bin/httpd	2.19%
/compress/usr/sbin/sessionlimitd	2.08%
/compress/usr/sbin/ddns_had	2.01%
/util/in	1.98%
/compress/usr/sbin/signal_wrapper	1.89%
/compress/usr/sbin/proactor1.2/pro	1.41%
/compress/sbin/sshipsecpm	0.72%
/compress/bin/sleep	0.27%
/compress/bin/bash	0.27%
/compress/usr/sbin/cron	0.26%

[24] Mark Luchs and Christian Doerr. 2019. The curious case of port 0. In *2019 IFIP Networking Conference (IFIP Networking)*. 1–9. https://doi.org/10.23919/IFIPNetworking46909.2019.8999403

[25] Aniss Maghsoudlou, Oliver Gasser, and Anja Feldmann. 2021. Zeroing in on port 0 traffic in the wild. In *Passive and Active Measurement: 22nd International Conference, PAM 2021, Virtual Event, March 29–April 1, 2021, Proceedings 22*. Springer, 547–563.

[26] Anna Mandalari, Marcelo Bagnulo, and Andra Lutu. 2015. TCP Fast Open: initial measurements. *ACM CoNEXT Student Workshop*.

[27] Maxmind GeoLite2. 2025. *GeoIP2 and GeoLite City and Country Databases*. https://www.maxmind.com

[28] David Moore, Vern Paxson, Stefan Savage, Colleen Shannon, Stuart Staniford, and Nicholas Weaver. 2003. Inside the Slammer Worm. *IEEE Security and Privacy* 1, 4 (2003), 33–39.

[29] David Moore and Colleen Shannon. 2005. The Spread of the Witty Worm. *IEEE Security and Privacy* 2, 4 (2005), 46–50.

[30] David Moore, Colleen Shannon, and kc Claffy. 2002. Code-Red: A Case Study on the Spread and Victims of an Internet Worm. In *ACM Internet Measurement Workshop*.

[31] Jonas Mücke, Marcin Nawrocki, Raphael Hiesgen, Patrick Sattler, Johannes Zirngibl, Georg Carle, Thomas C Schmidt, and Matthias Wählisch. 2022. Waiting for QUIC: On the Opportunities of Passive Measurements to Understand QUIC Deployments. *arXiv preprint arXiv:2209.00965* (2022).

[32] Sadia Nourin, Erik Rye, Kevin Bock, Nguyen Phong Hoang, and Dave Levin. 2025. Is Nobody There? Good! Globally Measuring Connection Tampering without Responsive Endhosts. In *2025 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, Los Alamitos, CA, USA, 1400–1418. https://doi.org/10.1109/SP61157.2025.00153

[33] Sadia Nourin, Van Tran, Xi Jiang, Kevin Bock, Nick Feamster, Nguyen Phong Hoang, and Dave Levin. 2023. Measuring and Evading Turkmenistan's Internet Censorship: A Case Study in Large-Scale Measurements of a Low-Penetration Country. In *Proceedings of the ACM Web Conference 2023* (Austin, TX, USA) *(WWW '23)*. Association for Computing Machinery, New York, NY, USA, 1969–1979. https://doi.org/10.1145/3543507.3583189

[34] Eric Pauley, Paul Barford, and Patrick McDaniel. 2023. DScope: A Cloud-Native Internet Telescope. In *32nd USENIX Security Symposium (USENIX Security 23)*. USENIX Association, Anaheim, CA, 5989–6006. https://www.usenix.org/conference/usenixsecurity23/presentation/pauley

[35] Stuart Staniford, David Moore, Vern Paxson, and Nicholas Weaver. 2004. The Top Speed of Flash Worms. In *ACM Workshop on Rapid Malcode (WORM)*.

[36] Ram Sundara Raman, Louis-Henri Merino, Kevin Bock, Marwan Fayed, Dave Levin, Nick Sullivan, and Luke Valenta. 2023. Global, Passive Detection of Connection Tampering. In *Proceedings of the ACM SIGCOMM 2023 Conference* (New York, NY, USA) *(ACM SIGCOMM '23)*. Association for Computing Machinery, New York, NY, USA, 622–636. https://doi.org/10.1145/3603269.3604875

[37] The ZMap Project. 2025. *ZGrab 2.0: Fast Go Application Scanner*. https://github.com/zmap/zgrab2

[38] Eric Wustrow, Manish Karir, Michael Bailey, Farnam Jahanian, and Geoff Huston. 2010. Internet background radiation revisited. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement* (Melbourne, Australia). Association for Computing Machinery, New York, NY, USA, 62–74. https://doi.org/10.1145/1879141.1879149

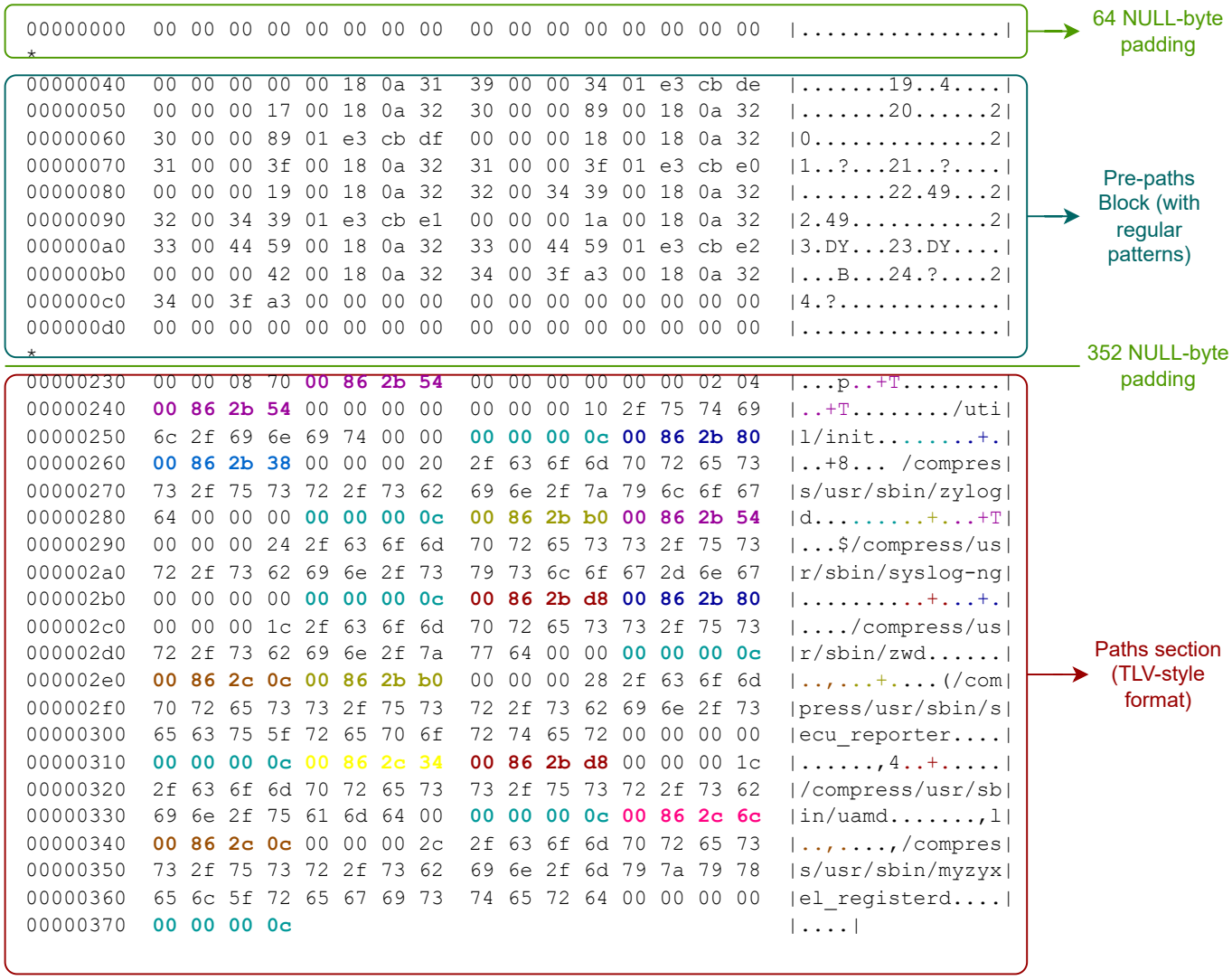


Figure 3: "Zyxel" packets (reverse engineered). The packet is divided in two macro-areas: a section with regular pattern (after the first NULL-byte padding), and a section with path of file usually found in Zyxel firmware (after a second NULL-byte padding).