Analysis of Three Variants of the HailBot Botnet Attacking DeepSeek

Antiy CERT

First draft completed: February 8, 2025
First published time: February 8, 2025

The original report is in Chinese, and this version is an AI-translated edition.

1 Overview

Antiy CERT released the report "Analysis of Botnet Samples Related to Attacks on DeepSeek", analyzing the two active botnet systems RapperBot and HailBot and their typical samples in the attack, and analyzing their derivative relationship with the Mirai botnet source code leak. Antiy engineers relied on the feature engineering mechanism to further compare the HailBot botnet sample set with finer granularity. In the comparison of the string output by the sample to the console as the classification identification condition , it was found that some samples modified the output string "hail china mainland" of the early samples , and the two groups with a larger number of them were modified to "you are now apart of hail cock botnet" and "I just wanna look after my cats, man.". To distinguish these three groups of samples, we named the three groups of variants HailBot.a , HailBot.b , and HailBot.c , and conducted corresponding analysis on the propagation method, decryption algorithm, online package, DDoS instructions, etc. of the three groups of samples. There are also samples that modify the output string to other content, but the number is small and has not been analyzed in detail.

Table 1-1 The relationship between the three HailBot variants

	HailBot.a	HailBot.b	HailBot.c
Special strings	hail china mainland	you are now apart of hail cock botnet	I just wanna look after my cats, man.
Transmission method	CVE-2017-17215	CVE-2017-17215 CVE-2023-1389 Cracking attack (number of accounts and passwords 45)	CVE-2017-17215 CVE-2023-1389 Brute force attack (number of accounts and passwords: 96)
Decryption algorithm	ChaCha20 Algorithm key: " 16 1E 19 1B 11 1F 00 1D 04 1C 0E 08 0B 1A 12 07 05 09 0D 0F 06 0A 15 01 0C 14 1F 17 02 03 13 18 "	ChaCha20 Algorithm key: " 16 1E 19 1B 11 1F 00 1D 04 1C 0E 08 0B 1A 12 07 05 09 0D 0F 06 0A 15 01 0C 14 1F 17 02 03 13 18 "	ChaCha20 Algorithm key: " 5E 8D 2A 56 4F 33 C1 C9 72 5D F9 1D 01 6C 2F 0B 77 3D 81 94 58 40 63 0A 79 62 1F 80 5C 3E 16 04 "

	nonce : " 1E 00 4A 00 00 00 00 00 00 00 00 00 "	nonce : " 1E 00 4A 00 00 00 00 00 00 00 00 00 "	nonce : " 1E 00 4A 00 00 00 00 00 00 00 00 00 "
Online package	31 73 13 93 04 83 32 01	Most samples: 56 63 34 86 90 69 21 01 A small number of samples: 31 73 13 93 04 83 32 01 (same as HailBot.a)	56 63 34 86 90 69 21 01
DDoS Instructions	8 instructions, instruction numbers 0-7	15 instructions, instruction numbers 0-14	10 instructions, instruction numbers 0-7, 11, 14

2 Sample Analysis

2.1 HailBot.a

HailBot.a is the earliest variant. Since it outputs "hail china mainland" to the console when running, the related botnet is named HailBot. This section has some overlaps with the first analysis report, mainly to compare the differences between different versions.

The sample information of HailBot.a is shown in Table 2-1 below.

Table 2-1 Typical sample labels of HailBot.a

Virus name	**Trojan/ Linux.Mirai [Backdoor]**
MD5	2DFE4015D6269311DB6073085FD73D1B
Processor architecture	ARM32
File size	74.78 KB (76,572 bytes)
File format	ELF 32-bit LSB executable
Packer type	None
Compiled language	C /C++

2.1.1 Transmission Method

HailBot.a spreads by exploiting vulnerabilities. The long-used CVE-2017-17215 exists in the UPnP (Universal Plug and Play) service of specific versions of routers. Attackers can execute arbitrary code on the target device by sending specially crafted HTTP requests.

```
strcat(
  payload + 280,
  "POST /ctrlt/DeviceUpgrade_1 HTTP/1.1\r\n"
  "Content-Length: 469\r\n"
  "Connection: keep-alive\r\n"
  "Accept: */*\r\n"
  "Authorization: Digest username=\"dslf-config\", realm=\"HuaweiHomeGateway\", nonce=\"88645cefb1f9ede0e33"
  "6e3569d75ee30\", uri=\"/ctrlt/DeviceUpgrade_1\", response=\"3612f843a42db38f48f59d2a3597e19c\", algorith"
  "m=\"MD5\", qop=\"auth\", nc=00000001, cnonce=\"248d1a2560100669\"\r\n"
  "\r\n"
  "<?xml version=\"1.0\" ?><s:Envelope xmlns:s=\"http://schemas.xmlsoap.org/soap/envelope/\" s:encodingStyl"

strcat(payload + 280, *(dword_229C0 + 132));
strcat(
  payload + 280,

  "(echo HUAWEIUPNP)</NewDownloadURL></u:Upgrade></s:Body></s:Envelope>\r\n"
  "\r\n");
```

Figure 2-1 HailBot.a constructs vulnerability exploit payload

2.1.2 Decryption Algorithm

After HailBot.a runs, it first decrypts the domain name using the ChaCha20 algorithm. Key is "16 1E 19 1B 11 1F 00 1D 04 1C 0E 08 0B 1A 12 07 05 09 0D 0F 06 0A 15 01 0C 14 1F 17 02 03 13 18" and nonce is "1E 00 4A 00 00 00 00 00 00 00 00 00".

```
int __fastcall add_entry(int a1, int a2, int a3)
{
  int v3; // r4
  table_value *v4; // r6
  char *v6; // r0
  int result; // r0

  v3 = a1;
  v4 = table;
  v6 = strdup(a2);
  v4[v3].val_len = a3;
  v4[v3].val = v6;
  result = chacha20(key, 1, nonce, v6, v6, v4[v3].val_len);// 前三个参数为key,counter和nonce
  v4[v3].val[v4[v3].val_len] = 0;               // 使字符串以空字符结尾
  return result;
}
```

Figure 2-2 HailBot.a uses chacha20 to decrypt the string

2.1.3 Online Package

After HailBot.a is running, it sends an online data packet with the content: "31 73 13 93 04 83 32 01".

```
LOBYTE(v66[0]) = util_strlen(v58);
send(fd_serv, &unk_19118, 8, MSG_NOSIGNAL);// 31 73 13 93 04 83 32 01
send(fd_serv, v66, 1, 0x4000);
if ( LOBYTE(v66[0]) )
  send(fd_serv, v58, LOBYTE(v66[0]), 0x4000);
```

Figure 2-3 HailBot.a sends online data packets

2.1.4 DDoS Instructions

After receiving the command sent by the attacker, HailBot.a will execute the corresponding DDoS attack according to different commands. The DDoS commands supported by Table 2-2Table 2-2.

Table 2-2 DDoS commands supported by HailBot.a

Instruction number	Function	Influence
0	TCP flood attack	Create a connection and send a large number of 500 to 900 bytes of TCP requests to consume the victim's network bandwidth.
1	SSDP flood attack	Use the Simple Service Discovery Protocol (SSDP) to send a large number of "discovery message" requests to force the victim to respond, consuming the victim's memory and CPU resources.
2	GRE IP flood attack	Send a large amount of GRE protocol data encapsulated with IP network packets consumes the victim's network bandwidth.
3	SYN flood attack	Send a large number of SYN packets causes the server to create a large number of requests in a semi-connected state, consuming system memory and CPU resources.
4	UDP flood attack (512 bytes)	Send a large number of 512-byte UDP requests consumes the victim's network bandwidth.
5	UDP flood attack (1024 bytes)	Send a large number of 1024-byte UDP requests consumes the victim's network bandwidth.
6	TCP STOMP flood attack	Send Create Connection Send a large amount of 768 bytes of data to consume the victim's network bandwidth.
7	TCP ACK flood attack	Send ACK packets with random source port, destination port, and data information consumes the victim's network bandwidth.

2.2 HailBot.b

HailBot.b is also a botnet developed based on the Mirai source code. The output string is: "you are now apart of hail cock botnet".

Sample information of Table 2-3

Table 2-3 Typical sample labels of HailBot.b

Virus name	Trojan/ Linux.Mirai [Backdoor]
MD5	BB9275394716C60D1941432C7085CA13
Processor architecture	AMD64
File size	93.34 KB (95,576 bytes)
File format	ELF 64-bit LSB executable
Packer type	None
Compiled language	C /C++

2.2.1 Transmission Method

HailBot.b also exploits CVE-2017-17215 to spread .

```
.rodata:0000000000413188 aPostCtrltDevic db 'POST /ctrlt/DeviceUpgrade_1 HTTP/1.1',0Dh,0Ah
.rodata:0000000000413188                                        ; DATA XREF: sub_405620+985fo
.rodata:0000000000413188                 db 'Content-Length: 430',0Dh,0Ah
.rodata:0000000000413188                 db 'Connection: keep-alive',0Dh,0Ah
.rodata:0000000000413188                 db 'Accept: */*',0Dh,0Ah
.rodata:0000000000413188                 db 'Authorization: Digest username="dslf-config", realm="HuaweiHomeGa'
.rodata:0000000000413188                 db 'teway", nonce="88645cefb1f9ede0e336e3569d75ee30", uri="/ctrlt/Dev'
.rodata:0000000000413188                 db 'iceUpgrade_1", response="3612f843a42db38f48f59d2a3597e19c", algor'
.rodata:0000000000413188                 db 'ithm="MD5", qop="auth", nc=00000001, cnonce="248d1a2560100669"',0Dh
.rodata:0000000000413188                 db 0Ah
.rodata:0000000000413188                 db 0Dh,0Ah
```

Figure 2-4 CVE-2017-17215 exploit payload in HailBot.b

Some samples exploit the CVE-2023-1389 to spread.

In addition, usernames and passwords used for brute force attacks were also found in the Table 2-4Table 2-4.

Table 2-4Usernames and passwords used by HailBot.b brute force attacks and corresponding products and services

(The table content is based on DeepSeek output and manual revision, please note)

Username	Password	Possible associated services/brands/device types
leox	leolabs_7	Leox equipment or customized equipment (such as some industrial control systems or private network equipment)
root	wabjtam	It may be some older routers or cameras (such as small Chinese brand devices)
telnetadmin	telnetadmin	The default Telnet account of some network devices (such as switches and routers)
admin	gpon	Some fiber optic terminal equipment (such as ZTE/Huawei GPON optical modem)
admin	admin123	Common default passwords (commonly used in routers and cameras, such as TP-Link and D-Link)
e8ehome	e8ehome	Some optical modems /routers of China Telecom or China Unicom (Shanghai Bell optical modem, ZTE ZXV10 H618C router, ZXA10 F460 optical modem)

default	default	Common default configuration for some devices (such as some older routers or IoT devices)
root	root	Some devices and services have common default passwords
default	OxhL	May be a specific brand of equipment (such as some enterprise-class switches or firewalls)
root	hme12345	Hikvision related equipment (such as some cameras or NVR)
admin	aquario	Possibly Aquario branded equipment (such as temperature control systems or industrial control equipment)
root	Zte521	ZTE fiber optic modem or router
root	1234	Common default passwords
root	antslq	Possibly security equipment (such as some domestic camera brands)
default	tlJwpbo6	Complex passwords may be used on enterprise-level devices (such as firewalls or servers)
root	default	Network devices (such as the default configuration of some switches)
admin	1988	Possibly default passwords for some cameras or DVRs (e.g. year-dependent)
adtec	adtec	Adtec brand equipment (such as surveillance systems or broadcasting equipment)
root	HKIPC2016	Hikvision IPC Camera
admin	hme12345	Hikvision or related equipment
hikvision	hikvision	Default account for Hikvision devices
root	login!@ #123	Enterprise-grade equipment (such as servers or high-end routers)
telecomadmin	admintelecom	Telecom operator equipment (such as Huawei/ ZTE optical modems)
telnetadmin	HI0605v1	Possibly Telnet login to Hikvision (HI) equipment
admin	qwasz	Common simple passwords (commonly found in low-end routers or IoT devices)
support	support	Technical support accounts (such as servers or network equipment)
root	5up	Minimal passwords may be used to test equipment or embedded systems
root	a	Unknown
root	icatch99	Cameras using iCatch chips (such as some domestic camera brands)
Admin	a	Unknown
Admin	Admin	Universal administrator password
root	adminpassword	Common administrator password (such as some new routers)
root	viz	Not sure, maybe it is a custom device for a certain brand
root	unisheen	Possibly a UniSheen brand device (such as a camera or industrial control equipment)
root	a1sev5y7c39k	Complex passwords may be used on enterprise-level devices (such as firewalls or VPN devices)
root	cxlinux	Linux-based embedded devices (such as some industrial control systems)
root	sr1234	Possibly surveillance equipment (such as some DVRs or NVRs)

root	new orang	NewOrange cameras or IoT devices
root	neworange88888888	NewOrange cameras or IoT devices
root	neworangetech	NewOrange cameras or IoT devices
root	oelinux123	Default credentials for Linux systems or embedded devices
root	hslwificam	HSL Brand WiFi Camera
root	jvbz	Not sure, maybe a niche brand device
admin	stdONU101	Optical Network Unit (ONU) equipment (such as standard optical modem or operator equipment)
admin	stdONUi0i	Optical Network Unit (ONU) equipment (such as standard optical modem or operator equipment)

2.2.2 Decryption Algorithm

The domain name decryption algorithm of HailBot.b is the same as that of HailBot.a, which is ChaCha20. The key and nonce used for decryption are also the same as those of HailBot.a. The key used for decryption is "16 1E 19 1B 11 1F 00 1D 04 1C 0E 08 0B 1A 12 07 05 09 0D 0F 06 0A 15 01 0C 14 1F 17 02 03 13 18", and the nonce is "1E 00 4A 00 00 00 00 00 00 00 00 00".

```
.data:00000000005160A0 key          db 16h, 1Eh, 19h, 1Bh, 11h, 1Fh, 0, 1Dh, 4, 1Ch, 0Eh, 8, 0Bh, 1Ah, 12h, 7
.data:00000000005160A0                                  ; DATA XREF: sub_408470+44↑o
.data:00000000005160A0              db 5, 9, 0Dh, 0Fh, 6, 0Ah, 15h, 1, 0Ch, 14h, 1Fh, 17h, 2, 3, 13h, 18h
.data:00000000005160C0 nonce        dw 1Eh, 4Ah, 0Eh dup(0) ; DATA XREF: sub_408470+37↑o
```

Figure 2-5The key and nonce of ChaCha20 algorithm

2.2.3 Online Package

Among the HailBot.b samples, most of the samples have the same online package, which are: "56 63 34 86 90 69 21 01". The online packets of a few samples (such as MD5: F0E951D1ACFDF78E741B808AB6AB9628) are the same as HailBot.a , which are "31 73 13 93 04 83 32 01".

```
qmemcpy(packet, "Vc4", 3);
packet[3] = 0x86;
packet[4] = 0x90;
packet[5] = 0x69;
packet[6] = 0x21;
packet[7] = 1;
send((unsigned int)dword_51605C, packet, 8LL, 0x4000LL);// 56 63 34 86 90 69 21 01
send((unsigned int)dword_51605C, v4, 1LL, 0x4000LL);
if ( v4[0] )
  send((unsigned int)dword_51605C, a1, (unsigned __int8)v4[0], 0x4000LL);
```

Figure 2-6Send online data packets

2.2.4 DDoS Instructions

HailBot.b supports more DDoS commands than HailBot.a. The DDoS commands supported by HailBot.b are Table 2-5Table 2-5.

Table 2-5 DDoS commands supported by HailBot.b

Instruction number	Function	Influence
0	TCP flood attack	Consume the victim's network bandwidth by creating connections and sending a large number of 512-byte TCP requests.
1	UDP flood attack (512 bytes)	It consumes the victim's network bandwidth through a large number of 512-byte UDP requests without exception handling.
2	GRE IP flood attack	Consume the victim's network bandwidth through a large amount of GRE protocol data encapsulated in IP network packets.
3	SYN flood attack	By sending a large number of SYN packets, the server creates a large number of requests in a semi-connected state, consuming system memory and CPU resources.
4	UDP flood attack (512 bytes)	Consume the victim's network bandwidth through a large number of 512-byte UDP requests.
5	UDP flood attack (1024 bytes)	Consume the victim's network bandwidth through a large number of 1024-byte UDP requests.
6	TCP STOMP flood attack	Consume the victim's network bandwidth by creating a connection and sending a large amount of 768 bytes of TCP data with ACK and PSH flags.
7	TCP ACK flood attack	Consume the victim's network bandwidth by sending ACK packets with random source port, destination port, and data information.
8	None	This instruction is not implemented
9	Unknow_1	TCP packets of unknown format
10	TCP ACK flood attack	Consume the victim's network bandwidth by sending ACK packets with specific source port, destination port, data and other information.
11	UDP flood attack	Randomly send UDP packets ranging from 100 to 1312 bytes to consume the victim's network bandwidth. The packets begin with " HDR: " in an attempt to evade firewall detection.
12	Unknow_ 2	UDP packet of unknown format
13	TCP STOMP flood attack	Consume the victim's network bandwidth by sending large amounts of 1 to 71 bytes of TCP data with ACK and PSH flags.
14	Unknow_ 3	By sending a large number of UDP packets with length 0

2.3 HailBot.c

HailBot.c is also a botnet redeveloped based on the Mirai source code. The string output by the new version is: "I just wanna look after my cats, man."

Table 2-6 Typical sample labels of HailBot.c

Virus name	Trojan/ Linux.Mirai [Backdoor]
MD5	64ED4E5B07610D80539A7C6B9EF171AA
Processor architecture	ARM32
File size	66.55 KB (68 , 148 bytes)
File format	ELF 32 -bit LSB executable
Packer type	None
Compiled language	C /C++

2.3.1 Transmission Method

This sample also uses CVE-2023-1389 and CVE-2017-17215 to spread. CVE-2023-1389 is spread through the leading file, while CVE-2017-17215 is written into the sample itself.

HailBot.c also spreads by brute force. The usernames and passwords used for brute force attacks are increased compared to HailBot.b , as Table 2-7Table 2-7.

Table 2-7 Username and password used by HailBot.c brute force attack

(The table content is based on DeepSeek output and manual revision, please note)

Username	Password	Possible associated services/brands/device types
root	Pon521	ZTE router (default password for some models)
root	Zte521	ZTE router (commonly found in ZTE optical modems /routers)
root	root621	Unknown (may be a customized device for a specific manufacturer)
root	viz	Unknown (maybe a camera or IoT device)
root	oelinux123	Unknown (may be relevant for embedded Linux devices)
root	root	Generic default (Linux devices, routers, cameras, etc.)
root	wabjtam	Unknown
root	Zxic521	ZTE router (guess it is the default password format of early ZTE devices)
root	tsgoingon	Unknown
root	123456	Common default configuration for multiple devices (common on low-security devices)
root	xc3511	Unknown
root	solokey	Unknown

root	default	Common default passwords (default passwords for some IoT devices)
root	a1sev5y7c39k	Unknown (may be randomly generated or device-specific key)
root	HKIPC2016	Hikvision cameras (HKIPC is a common prefix)
root	unisheen	Unknown
root	Fireitup	Unknown (possibly custom firmware password)
root	hslwificam	Unknown (possibly a WiFi camera brand)
root	5up	Unknown
root	jvbz	Unknown
root	1001chin	Unknown
root	system	Common default passwords (some industrial control equipment or servers)
root	zlxx .	Unknown
root	admin	Common default passwords (routers, switches, etc.)
root	7ujMko0vizxv	Unknown (may be firmware-specific or custom device related)
root	1234horses	Unknown
root	antslq	Unknown
root	xc12345	Unknown (may be related to the camera chip)
root	xmhdipc	Unknown (maybe camera model abbreviation)
root	icatch99	iCatch camera (default password for some models)
root	founder88	Unknown (may be a custom device password)
root	Xirtam	Unknown (possibly a variant of "matrix" spelled backwards)
root	taZz@01	Unknown
root	/* 6.= _ja	Unknown
root	12345	Common default passwords (routers, cameras, etc.)
root	t0talc0ntr0l4!	Unknown
root	7ujMko0admin	Unknown
root	telecomadmin	Telecommunications equipment (such as optical modem administrator account)
root	ipcam_rt5350	RT5350 chip camera (MediaTek IP camera)
root	juantech	Unknown (likely JuanTech brand device)

root	1234	Common default passwords (low security devices)
root	dreambox	Dreambox satellite receiver (default password for some models)
root	IPCam@sw	Webcam (generic default or brand specific)
root	Zhongxing	ZTE equipment
root	hi3518	HiSilicon Hi3518 chip camera (commonly used in security equipment)
root	hg2x0	Unknown (may be related to Huawei HG series optical modems)
root	dropper	Unknown (possibly malware backdoor password)
root	ipc71a	Webcam (model dependent)
root	root123	Common default passwords (Extended Default Password)
root	telnet	Common default passwords (Telnet service default credentials)
root	ipcam	Webcam (Generic Default)
root	Grouter	Unknown (may be the router brand abbreviation)
root	GM8182	Unknown (possibly device model)
root	20080826	Unknown (maybe a date-related password)
root	3ep5w2u	Unknown
admin	root	Common default passwords (reverse credentials for some devices)
admin	admin	Common default passwords (routers, cameras, etc.)
admin	admin123	Common default passwords (Extended Default Password)
admin	1234	Common default passwords (low security devices)
admin	admin1234	Common default passwords (Extended Default Password)
admin	12345	Common default passwords (common on consumer devices)
admin	admin@123	Common default passwords (with symbolic variations)
admin	BrAhMoS@15	Unknown (may be a custom password)
admin	GeNeXiS@19	Unknown (may be a custom password)
admin	firetide	Firetide Wireless Network Devices (Default Password)
admin	2601hx	Unknown
admin	service	Common default passwords (service accounts)
admin	password	Common default passwords (widely used on various devices)
supportadmin	supportadmin	Generic default password (technical support account)

telnetadmin	telnetadmin	Common default password (Telnet management account)
telecomadmin	admintelecom	Telecommunications equipment (such as optical modem super administrator account)
guest	guest	Common default password (guest account)
ftp	ftp	Common default password (FTP service anonymous access)
user	user	Common default passwords (normal user accounts)
guest	12345	Universal default password (Guest account extended password)
nobody	nobody	Common default password (system account)
daemon	daemon	Common default password (system account)
default	1cDuLJ7c	Unknown
default	tlJwpbo6	Unknown
default	FqV	Unknown
default	OxhL	Unknown
default	12345	Universal default password (device default setting password)
default	default	Common default (default account password)
default	JbXj	Unknown
default	tluafed	Unknown (possibly "default" spelled backwards)
guest	123456	Universal default password (Guest account extended password)
bin	bin	Common default passwords (Linux system accounts)
vstarcam2015	20150602	Vstarcam camera (model - dependent default password)
support	support	General default (Technical Support Account)
hikvision	hikvision	Hikvision devices (default password)
default	antslq	Unknown
e8ehomeasb	e8ehomeasb	Telecom optical modem (Shanghai Bell E8-C)
e8ehome	e8ehome	Some optical modems /routers of China Telecom or China Unicom (default passwords of Shanghai Bell optical modem, ZTE ZXV10 H618C router, ZXA10 F460 optical modem, etc.)
e8telnet	e8telnet	Telnet login username and password for some telecom routers or optical modems (such as Huawei HG8245, ZTE F660, etc.)

support	1234	Common default password (simplified password for technical support account)
cisco	cisco	Cisco devices (older models default passwords)

2.3.2 Decryption Algorithm

The domain name decryption algorithm of HailBot.c is the same as that of HailBot.a, which is ChaCha20. The nonce used for ChaCha20 decryption is also the same, which is "1E 00 4A 00 00 00 00 00 00 00 00 00".

The key used by HailBot.c for ChaCha20 decryption is different from that used by HailBot.a, which is "5E 8D 2A 56 4F 33 C1 C9 72 5D F9 1D 01 6C 2F 0B 77 3D 81 94 58 40 63 0A 79 62 1F 80 5C 3E 16 04".

```
.data:00025AD1 key             DCB 0x5E, 0x8D, 0x2A, 0x56, 0x4F, 0x33, 0xC1, 0xC9, 0x72
.data:00025AD1                                       ; DATA XREF: sub_C534+64↑o
.data:00025AD1                                       ; .text:off_C5C8↑o
.data:00025ADA                 DCB 0x5D, 0xF9, 0x1D, 1, 0x6C, 0x2F, 0xB, 0x77, 0x3D, 0x81
.data:00025AE4                 DCB 0x94, 0x58, 0x40, 0x63, 0xA, 0x79, 0x62, 0x1F, 0x80
.data:00025AED                 DCB 0x5C, 0x3E, 0x16, 4
.data:00025AF1 nonce           DCB 0x1E, 0, 0x4A, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
```

Figure 2-7The key and nonce of ChaCha20 algorithm

In addition, for the decrypted domain names, some domain names of HailBot.c and HailBot..b overlap.

2.3.3 Online Package

The online packet of the HailBot. c sample is: "56 63 34 86 90 69 21 01", as shown in the following figure.

```
.rodata:0001BC3C unk_1BC3C     DCB 0x56 ; V
.rodata:0001BC3C
.rodata:0001BC3D               DCB 0x63 ; c
.rodata:0001BC3E               DCB 0x34 ; 4
.rodata:0001BC3F               DCB 0x86
.rodata:0001BC40               DCB 0x90
.rodata:0001BC41               DCB 0x69 ; i
.rodata:0001BC42               DCB 0x21 ; !
.rodata:0001BC43               DCB    1
```

Figure 2-8 The online package of the HailBot.c sample

2.3.4 DDoS Instructions

HailBot.c supports more DDoS commands than HailBot.a. The DDoS commands supported by Table 2-8Table 2-8.

Table 2-8DDoS commands supported by HailBot.c

Instruction number	Function	Influence
0	TCP flood attack	Consume the victim's network bandwidth by creating connections and sending a large number of 512-byte TCP requests.
1	UDP flood attack (512 bytes)	Consume the victim's network bandwidth through a large number of 512-byte UDP requests and does not have exception handling.
2	GRE IP flood attack	Consume the victim's network bandwidth through a large amount of GRE protocol data encapsulated in IP network packets.
3	SYN flood attack	By sending a large number of SYN packets, the server creates a large number of requests in a semi-connected state, consuming system memory and CPU resources.
4	UDP flood attack (512 bytes)	Consume the victim's network bandwidth through a large number of 512-byte UDP requests.
5	UDP flood attack (1024 bytes)	Consume the victim's network bandwidth through a large number of 1024-byte UDP requests.
6	TCP STOMP flood attack	Consume the victim's network bandwidth by creating a connection and sending a large amount of 768 bytes of TCP data with ACK and PSH flags.
7	TCP ACK flood attack	Consume the victim's network bandwidth by sending ACK packets with random source port, destination port, and data information.
11	UDP flood attack	Randomly send UDP packets ranging from 100 to 1312 bytes to consume the victim's network bandwidth, but no longer start with "HDR:".
14	Unknown	By sending a large number of UDP packets with length 0

3 Other Information, Analytical Connections and Conclusions

3.1 Summary of Sample Analysis Results

HailBot.b and HailBot.c are both variants of HailBot.a , but they have some differences. In terms of automated propagation, HailBot.a , HailBot.b and HailBot.c all exploit CVE-2017-17215. HailBot.b and HailBot.c both spread by brute-forcing common username and password files. In terms of decryption algorithms, all three variants of HailBot use the ChaCha20 algorithm. HailBot.a and HailBot.b use the same decryption key, but HailBot.c has updated the decryption key. In terms of online data packets, HailBot.b and HailBot.c have been updated compared to HailBot.a . In terms of the commands supported by DDoS attacks, HailBot.b and HailBot.c support more commands than HailBot.a , and the commands supported by HailBot.b and HailBot.c are also different.

3.2 New Vulnerability Drop Method of the Sample

Variants B and C are implanted into some target devices through CVE-2023-1389, which is a high-risk command injection vulnerability affecting TP-Link Archer AX21 (AX1800) routers. The vulnerability exists in the router's web management interface. An attacker can inject malicious commands through a simple unauthenticated POST request, which will be executed with root privileges, achieving remote code execution.

Figure 3-1 2

3.3 Comparison of Sample Password Files

It is worth comparing the password files of variant B and variant C. The password file of variant B has 45 records, and that of variant C has 96 records, but the two password files have only 24 overlapping records. In addition to the continued use of some common default passwords, some of the passwords do not seem to be the default passwords of the devices, and they are passwords with a certain strength. It can be speculated that some enterprises or operators have adopted unified password settings for devices in large-scale deployment of networks or IoT devices for convenience. After the relevant passwords are obtained by the attacker and configured in the password file, they can spread the infection to all deployed devices that adopt the corresponding strategy. However, the inheritance logic of the password files between the two botnets can be further analyzed.

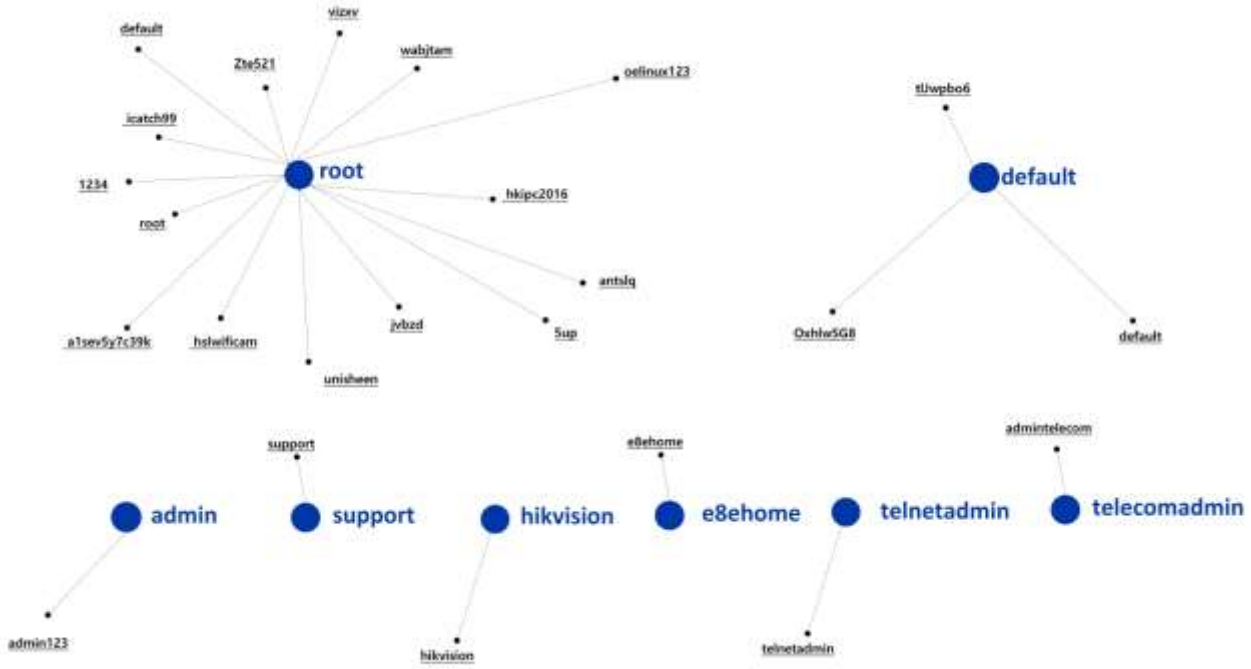

Figure 3-3 Duplicate content of HailBot.b and HailBot.c password files

3.4 Sample Life Cycle and Branch Relationship

Antiy CERT compared the three variant samples and activity times in the "Cyber Brain" platform and sorted out the activity times of the relevant samples as shown in the following figure. Since HailBot.b clearly showed the time relay characteristics with HailBot.a, it can be speculated that variant B may be an overall version update of variant A.

ID	僵尸网络	开始时间	最近时间	持续时间
1	HailBot.a	2023/9/25	2024/9/23	261天
2	HailBot.b	2024/9/30	2025/1/1	68天
3	HailBot.c	2024/11/27	2025/2/5	51天

Figure 3-4 5

However, when comparing variant C and variant B, they exploit similar vulnerabilities, but have a larger password file (but do not present a complete password file inheritance relationship), and the attack instruction encoding is basically the same. Therefore, variant C is the latest variant.

4 Summary: Sample Detection Analysis and Botnet Analysis

The ability to accurately detect threat samples is the basic capability of threat defense, and the fine-grained analysis of samples is the basic work of attack analysis.

However, the problem that emerged in this analysis is that our engine outputs precise names such as "Trojan/Linux.Mirai [Backdoor]" that strictly follow the CARO convention, but the differentiation provided is still insufficient for fine-grained analysis of botnets. The anti-virus engine is a detection capability for malicious executable objects. From the perspective of threat intelligence, this is intelligence support at the Payload/Tools level. Because there are a large number of sample version updates, encryption, anti-killing, and transformations, the anti-virus engine must curb the expansion of rules from a design perspective, and needs to hit more samples of the same family at a lower cost and rules. At this time, we added a large number of unpacking, decryption, virtual execution and other mechanisms to the AVL SDK anti-virus engine to enhance the robustness of detection. Therefore, when there are new changes in samples, as long as the anti-virus engine can detect and alert normally, we usually do not add new rules, let alone adjust the sample naming. This is basically a common practice among anti-virus companies. For example, samples related to HailBot.a are all named Mirai in many comparisons on the National Computer Virus Collaborative Analysis Platform (https://virus.cverc.org.cn/#/entirety/search) :

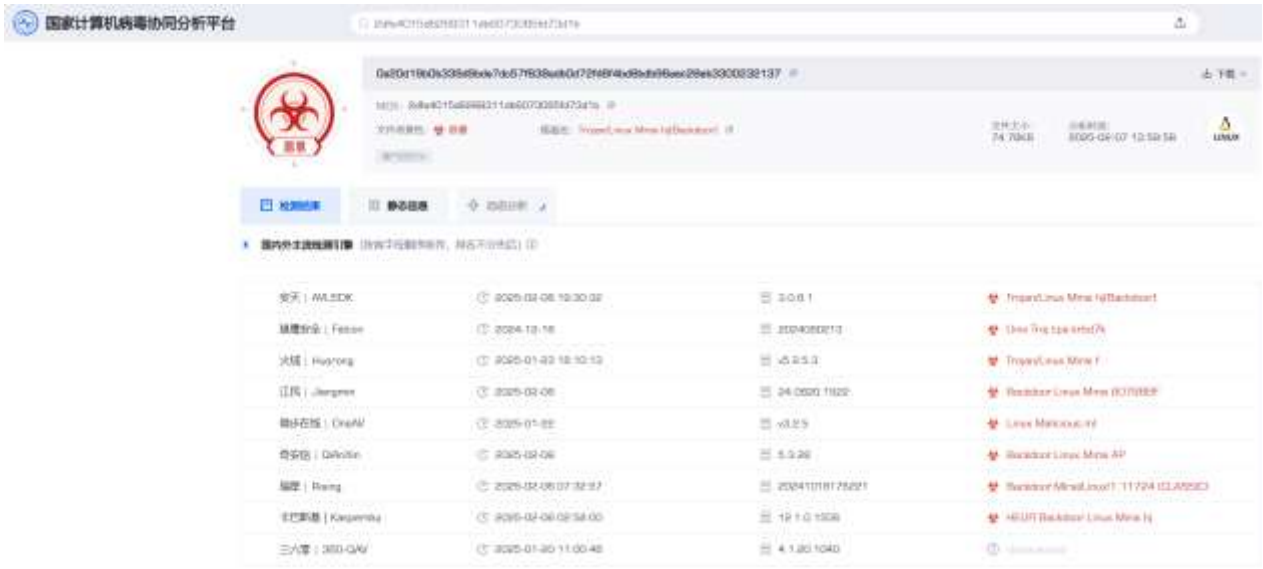

Figure 4-1Comparison of HailBot.a samples detected by the National Computer Virus Collaborative Analysis Platform

Sample analysis is an important part of botnet analysis and is also a "convergence point", but the complete analysis and tracking of botnets requires more fine-grained work, including analysis of C2, online data, attack instructions, etc., as well as analysis and evaluation of botnet distribution and scale, while attribution and tracing require more resources and greater costs.

Related analysis also proves once again that the prerequisite for government and enterprise defense against botnet infection is to do simple and boring basic skills, including changing default passwords, configuring different passwords for different devices and services, and updating system and device patches in a timely manner.

What is memorable about this analysis is that we analyzed malicious samples that attacked the big model platform with the assistance of big model technology and platform. We used the self-developed LanDi VILLM to assist the feature engineering system to achieve more convenient sample classification, clustering and feature (including characterization) discovery. DeepSeek helped us quickly output data summary tables from hard-coded information (of course, we still need to verify them one by one) and quickly compare password files when we sorted out the devices corresponding to the password files. We are integrating DeepSeek with our sample integrated analysis environment. Therefore, we are grateful for the big model technology, which can replace more repetition and basic reasoning. Let us create more valuable deep source knowledge and value.

Appendix 1: Some IoCs

Table 0-1Hash

MD5
6C6D1CCCE5946F0AA68F9E0C438C1E21
2DFE4015D6269311DB6073085FD73D1B
BB9275394716C60D1941432C7085CA13
F0E951D1ACFDF78E741B808AB6AB9628
A155F5812EA93DDEA553EA84CE28400D

Appendix 2: References

[1] Antiy. Analysis of Botnet Samples Related to Attacks on DeepSeek. (2025-0 2 - 05)

https://mp.weixin.qq.com/s/NvlVuA5urPG_r6attAiXsA

Appendix III: About Antiy

Antiy is committed to enhancing the network security defense capabilities of its customers and effectively responding to security threats. Through more than 20 years of independent research and development, Antiy has developed technological leadership in areas such as threat detection engines, advanced threat countermeasures, and large-scale threat automation analysis.

Antiy has developed IEP (Intelligent Endpoint Protection System) security product family for PC, server and other system environments, as well as UWP (Unified Workload Protect) security products for cloud hosts, container and other system environments, providing system security capabilities including endpoint antivirus, endpoint protection (EPP), endpoint detection and response (EDR), and Cloud Workload Protection Platform (CWPP) , etc. Antiy has established a closed-loop product system of threat countermeasures based on its threat intelligence and threat detection capabilities, achieving perception, retardation, blocking and presentation of the advanced threats through products such as the Persistent Threat Detection System (PTD), Persistent Threat Analysis System (PTA), Attack Capture System (ACS), and TDS. For web and business security scenarios, Antiy has launched the PTF Next-generation Web Application and API Protection System (WAAP) and SCS Code Security Detection System to help customers shift their security capabilities to the left in the DevOps process. At the same time, it has developed four major kinds of security service: network attack and defense logic deduction, in-depth threat hunting, security threat inspection, and regular security operations. Through the Threat Confrontation Operation Platform (XDR), multiple security products and services are integrated to effectively support the upgrade of comprehensive threat confrontation capabilities.

Antiy provides comprehensive security solutions for clients with high security requirements, including network and information authorities, military forces, ministries, confidential industries, and critical information infrastructure. Antiy has participated in the security work of major national political and social events since 2005 and has won honors such as the Outstanding Contribution Award and Advanced Security Group. Since 2015, Antiy's products and services have provided security support for major spaceflight missions including manned spaceflight, lunar exploration, and space station docking, as well as significant missions such as the maiden flight of large aircraft, escort of main force ships, and Antarctic scientific research. We have received several thank-you letters from relevant departments.

Antiy is a core enabler of the global fundamental security supply chain. Nearly a hundred of the world's leading security and IT enterprises have chosen Antiy as their partner of detection capability. At present, Antiy's threat detection engine provides security detection capabilities for over 1.3 million network devices and over 3 billion smart terminal devices worldwide, which has become a "national-level" engine. As of now, Antiy has filed 1,877 patents in the field of cybersecurity and obtained 936 patents. It has been awarded the title of National Intellectual Property Advantage Enterprise and the 17th (2015) China Patent Excellence Award.

Antiy is an important enterprise node in China emergency response system and has provided early warning and comprehensive emergency response in major security threats and virus outbreaks such as "Code Red", "Dvldr", "Heartbleed", "Bash Shellcode" and "WannaCry". Antiy conducts continuous monitoring and in-depth analysis against dozens of advanced cyberspce threat actors (APT groups) such as "Equation", "White Elephant", "Lotus" and "Greenspot" and their attack actions, assisting customers to form effective protection when the enemy situation is accurately predicted.