Analysis of Botnet Samples Related to Attacks on Deepseek

Antiy CERT

First published time: February 5, 2025

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

1 Overview

Recently, the online service of DeepSeek, a domestic AI model, was attacked by a large-scale cyber attack, resulting in multiple service interruptions. This has attracted the attention of the domestic security industry. According to the monitoring report of Qianxin XLab, it was found that the botnets RapperBot and HailBot[1]launched DDoS attacks against DeepSeek. In order to more effectively assess risks and support the prevention of related attacks, Antiy CERT extracted the zombie Trojan samples used by the above two botnets from the "Cyber Brain" platform sample library and conducted further analysis.

2 Sample Analysis

2.1 Rapperbot and HailBot's Previous Life: Mirai

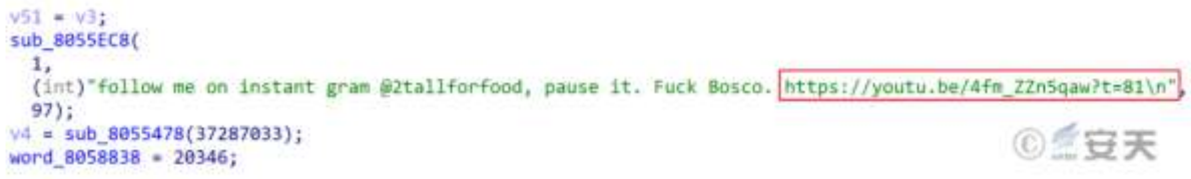
RapperBot and HailBot are both products of the Mirai botnet source code leak. The Mirai botnet was first discovered in 2016 and quickly attracted widespread attention. Its name comes from the Japanese word for "future". Unlike traditional botnets that mainly use Windows system zombies, Mirai infects and controls network cameras, home routers and other IoT devices to build a botnet system[2]. In 2016, it surfaced due to the "DYN incident"[3]. The three authors of Mirai, Paras Jha, Josiah White and Dalton Norman, are all Americans. The three operated a company called Protraf Solutions LLC, claiming to provide DDoS attack protection, but in fact used botnets to launch DDoS attacks and conduct extortion and other profit-making activities. In 2018, the three people were sentenced to five years of probation, 2,500 hours of community service, and compensation of US$127,000 by an Alaska court in the United States. They were also required to voluntarily give up the cryptocurrency they obtained during the crime.

Mirai specifically targets Internet of Things (IoT) devices for automated penetration and implantation, such as routers, network cameras, and digital video recorders (DVRs). These devices are often outside the security operations perspective or are home devices that often have problems such as unmodified default passwords or simple passwords, and outdated firmware versions. Mirai gains access to devices through password cracking and vulnerability attacks, and uploads malicious code to the devices to run, turning them into controlled zombie nodes[4]. A prominent feature of this botnet is its modular design and self-update capability, which enables it to quickly adapt to the ever-changing security environment and add new attack methods. Once a device is infected by Mirai, it will automatically perform scanning and detection tasks and attempt to spread malware to other devices, thereby building a large botnet[5].

On September 30, 2016, the source code of the Mirai botnet was publicly leaked on the GitHub platform. Attackers have customized and modified the source code to derive multiple variant families, including RapperBot and HailBot. The attackers' modification methods include but are not limited to replacing the master domain name (to avoid security vendors' bans), disguising the login authentication mechanism (impersonating legitimate device traffic), and obfuscating communication protocol fields (such as modifying the structure of the online heartbeat packet to bypass detection rules). Due to the high reusability of the source code, global black industry gangs are able to build "homogeneous" botnet clusters at a low cost. Although these variants show differences in surface functions, their core infection logic, C2 command system and attack modules are all inherited from the original Mirai architecture, making it difficult to trace the association with the manipulation organization behind them.

2.2 RapperBot Botnet

RapperBot is a botnet developed based on Mirai source code. It has multiple versions and can run on processors of different architectures, such as ARM, MIPS, SPARC, and x86. It was named "RapperBot" because its early samples embedded a link to a rap video.



```
v51 = v3;
sub_8055EC8(
  1,
  (int)"follow me on instant gram @2tallforfood, pause it. Fuck Bosco. https://youtu.be/4fm_ZZn5qaw?t=81\n",
  97);
v4 = sub_8055478(37287033);
word_8058838 = 20346;
```

Figure 2-1 2

The string "follow me on instant gram @2tallforfood, pause it. Fuck Bosco." in the code is translated as "Find me on Instgram @2tallforfood, pause it. FuckBosco." @2tallforfood is the account of a singer on the YouTube channel ALL URBAN CENTRAL. The account has only two videos on YouTube before 2021: @2TallForFood - Diamonds Is Lit (Official Video) [6] and @2tallforfood - I Am Da Bag (Official Video). Fuckbosco [7]is the name of another song by the singer. The singer is a niche singer, and the number of views of his videos did not exceed 500 at the time of the report.

The YouTube channel ALL URBAN CENTRAL is an American music entertainment channel established in 2014. Its main genres include music rap and hip hop as well as celebrity news. It has about 3 million subscribers. Most of the videos on this channel are less than 6 minutes long, and the total number of views on the channel is more than 2 billion. It charges fees through subscriptions. It ranks around 4,000 in the American music category.

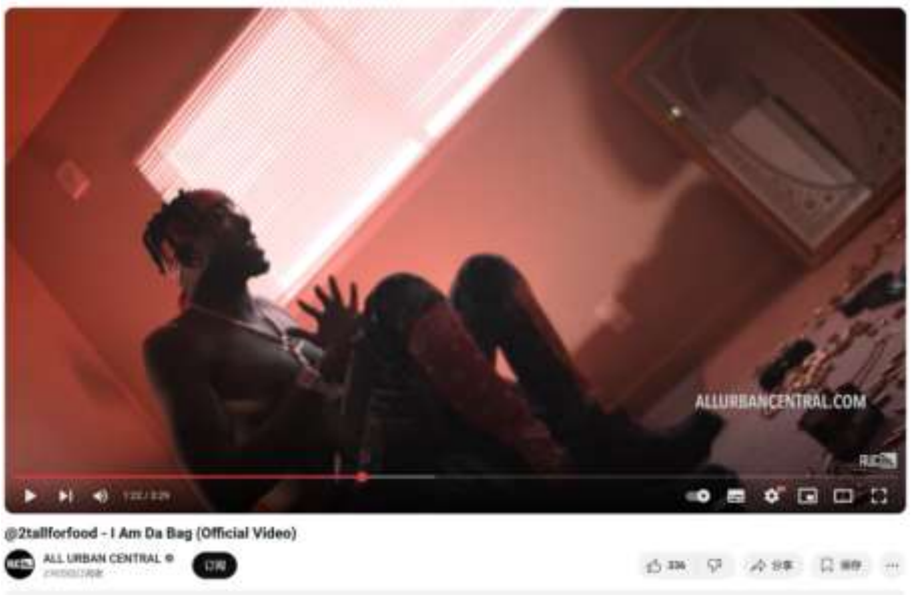


Figure 2-3Videos pointed to by embedded links in early RapperBot samples

2.2.1 Sample Labels

Table 21early versions of RapperBot

Virus name	Trojan/Linux.Mirai[Backdoor]
MD5	9E331675D780AF4585857B1F95B40CBB
Processor architecture	i386
File size	66.47 KB (68068 bytes)
File format	ELF
Digital signature	None
Packer type	None

VT first upload time	2022-06-17 08:10:19
VT test results	38/64

Table 2-2Sample labels of RapperBot new variant

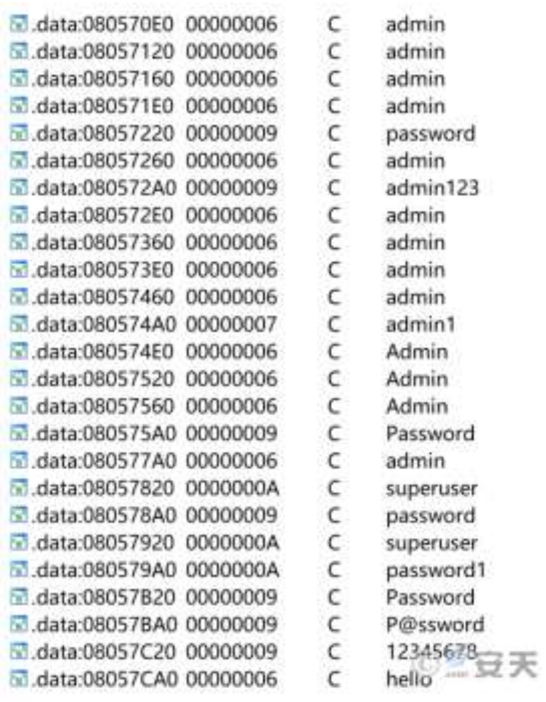
Virus name	Trojan/Linux.Mirai[Backdoor]
MD5	BEC7596CFB1225900673398ABB24FFA8
Processor architecture	i386
File size	80.47 KB (82400 bytes)
File format	ELF
Digital signature	None
Packer type	None
VT first upload time	2024-07-02 02:21:30
VT test results	37/63

Note: As the Antiy AVL SDK anti-virus engine has strong pre-processing capabilities, it can detect deformed derivative samples with fewer high-quality rules. As of the time of writing this article, the detection results of RapperBot samples are all Mirai. This is hereby stated.

2.2.2 How the Virus Spreads

2.2.2.1 SSH Brute Force

Some variants of the RapperBot botnet family spread through SSH brute force. In early samples, the credential list was hardcoded in the file, and later new variants changed to obtain the credential list from the C2 server. After successfully brute-forcing the SSH server, RapperBot executes a shell command to replace the ~/.ssh/authorized_keys file in the server, thereby maintaining remote access to the victim server.

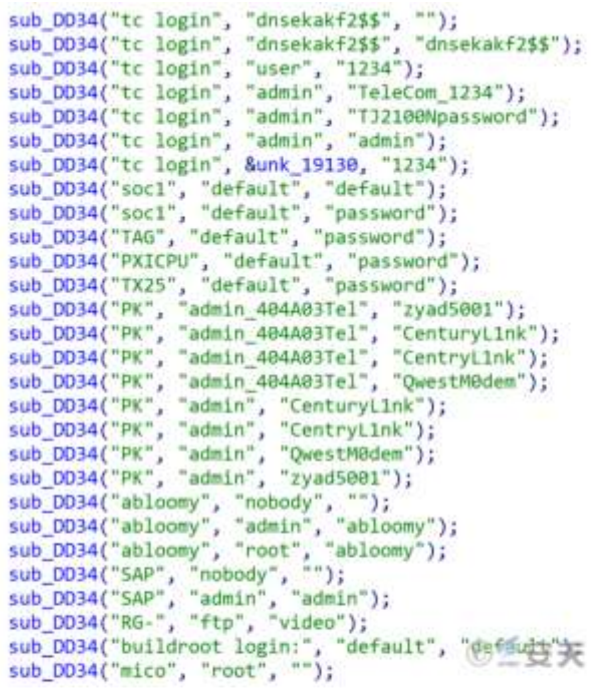


```
.data:080570E0 00000006    C    admin
.data:08057120 00000006    C    admin
.data:08057160 00000006    C    admin
.data:080571E0 00000006    C    admin
.data:08057220 00000009    C    password
.data:08057260 00000006    C    admin
.data:080572A0 00000009    C    admin123
.data:080572E0 00000006    C    admin
.data:08057360 00000006    C    admin
.data:080573E0 00000006    C    admin
.data:08057460 00000006    C    admin
.data:080574A0 00000007    C    admin1
.data:080574E0 00000006    C    Admin
.data:08057520 00000006    C    Admin
.data:08057560 00000006    C    Admin
.data:080575A0 00000009    C    Password
.data:080577A0 00000006    C    admin
.data:08057820 0000000A    C    superuser
.data:080578A0 00000009    C    password
.data:08057920 0000000A    C    superuser
.data:080579A0 0000000A    C    password1
.data:08057B20 00000009    C    Password
.data:08057BA0 00000009    C    P@ssword
.data:08057C20 00000009    C    12345678
.data:08057CA0 00000006    C    hello
```

Figure 2-4 Part of the SSH brute force credentials list hardcoded in the file

2.2.2.2 Telnet Default Password Detection

Some RapperBot botnet variants will perform detection through Telnet based on the default password of the device. The target device keywords, default user name and password are hardcoded in the file.



```
sub_DD34("tc login", "dnsekakf2$$", "");
sub_DD34("tc login", "dnsekakf2$$", "dnsekakf2$$");
sub_DD34("tc login", "user", "1234");
sub_DD34("tc login", "admin", "TeleCom_1234");
sub_DD34("tc login", "admin", "TJ2100Npassword");
sub_DD34("tc login", "admin", "admin");
sub_DD34("tc login", &unk_19130, "1234");
sub_DD34("socl", "default", "default");
sub_DD34("socl", "default", "password");
sub_DD34("TAG", "default", "password");
sub_DD34("PXICPU", "default", "password");
sub_DD34("TX25", "default", "password");
sub_DD34("PK", "admin_404A03Tel", "zyad5001");
sub_DD34("PK", "admin_404A03Tel", "CenturyL1nk");
sub_DD34("PK", "admin_404A03Tel", "CentryL1nk");
sub_DD34("PK", "admin_404A03Tel", "QwestM0dem");
sub_DD34("PK", "admin", "CenturyL1nk");
sub_DD34("PK", "admin", "CentryL1nk");
sub_DD34("PK", "admin", "QwestM0dem");
sub_DD34("PK", "admin", "zyad5001");
sub_DD34("abloomy", "nobody", "");
sub_DD34("abloomy", "admin", "abloomy");
sub_DD34("abloomy", "root", "abloomy");
sub_DD34("SAP", "nobody", "");
sub_DD34("SAP", "admin", "admin");
sub_DD34("RG-", "ftp", "video");
sub_DD34("buildroot login:", "default", "default");
sub_DD34("mico", "root", "");
```

Figure 2-5 6username/password table hardcoded in the sample file

From the hard-coded information that the relevant samples attempt to brute force, we can see that the targets of this type of RapperBot variant are mostly common network devices and IoT devices.

Table 2-3RapperBot built-in user detection login services, usernames, passwords and possible associated devices (The table content is based on DeepSeek output and manual revision, please note)

Service/Module	Username	Password	Possible associated services/brands/device types
tc login	dnsekakf2$$	""(null)	DASAN customized network equipment
tc login	dnsekakf2$$	dnsekakf2$$	DASAN customized network equipment
tc login	user	1234	DASAN customized network equipment
tc login	admin	TeleCom_1234	**China Telecom** customized equipment
tc login	admin	TJ2100Npassword	Tejas Networks TJ2100N optical modem or gateway
tc login	admin	admin	A variety of mainstream network devices
tc login	&unk_19130	1234	It may be some cameras and other IoT
soc1	default	Default	Various industrial products and software
soc1	default	password	Various industrial products and software
TAG	default	password	Suspected **IoT device**
PXICPU	default	password	Some **industrial embedded controller** devices
TX25	default	password	Probably some kind of wireless device
PK	admin_404A03Tel	zyad5001	**ZyXEL Router**
PK	admin_404A03Tel	Centurylink	**ZyXEL Router**
PK	admin_404A03Tel	QwestM0dem	**ZyXEL Router**
PK	admin	Centurylink	**ZyXEL Router**
PK	admin	QwestM0dem	**ZyXEL Router**
PK	admin	zyad5001	**ZyXEL Router**
abloom	nobody	""(null)	**Abloom brand IoT devices**
abloom	admin	Abloom	**Abloom brand IoT devices**
abloom	root	Abloom	**Abloom brand IoT devices**
SAP	nobody	""(null)	**SAP system test environment** or **IoT devices**
SAP	admin	Admin	**SAP NetWeaver Application Server**
RG-	ftp	Video	**Network Video Recorder (NVR) or IP Camera**

buildroot login	default	Default	**Various embedded Linux systems**
mico	root	""(null)	**Embedded Systems**

2.2.3 Behavior Analysis

Early versions of RapperBot supported fewer DoS attacks, including TCP STOMP attacks and UDP flood attacks. The new variant of RapperBot supports receiving commands similar to those of the early samples, but can support executing more types of DoS attacks.

Table 2-4Comparison of command functions of early versions and new variants

Instruction code	RapperBot early version features	RapperBot new variant features
1	Stay connected	Online package
2	Stop DoS attack and terminate operation	Response Packet
3	Performing a DoS attack	Heartbeat Packet
4	Stop DoS Attacks	Perform a DoS attack
5	None	Stop DoS attack and terminate operation
6	None	Close C2 connection

When an attacker launches a DoS attack, they select a pre-set sequence number to execute the corresponding function, thereby executing a specific DoS attack. From this, it can be seen that the command functions supported by RapperBot-related samples are mainly to launch DoS attacks, and from its early development to the present, its developers have gradually improved RapperBot's DoS attack function, thereby supporting the completion of a large range of DDoS attack activities.

Table 2-5 RapperBot new variants support multiple DoS attacks

Attack Command	Types of DoS Attacks	Attack Description
0	UDP flood attack	Consume the victim's network bandwidth by sending a large number of UDP packets.
1	UDP packet forgery	The attacker sends a large number of forged UDP packets to the target server, deceiving the server to respond and consuming the victim's network bandwidth.
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	GRE-Eth flood attack	Consume the victim's network bandwidth through a large amount of GRE protocol data encapsulated with Eth network packets.
4	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.

5	ACK flood attack	Consume the victim's network bandwidth by sending ACK packets with random source port, destination port, and data information.
6	ACK-PSH flood attack	It establishes a connection with the server through an ACK response with a PSH flag and sends a large number of requests to consume the victim's network bandwidth.
7	TCP flood attack	Consume the victim's network bandwidth by sending a large number of TCP packets.
8	HTTP Flood Attack	The attacker sends a large number of HTTP messages to the target server, consuming the victim's network bandwidth and server resources.

2.3 HailBot Botnet

HailBot is a botnet developed based on Mirai source code, which can run on processors of different architectures, such as ARM, x86, x64, and MIPS. Since it outputs "hail china mainland" to the console when it is running, it is named HailBot botnet.

2.3.1 Sample Labels

Table 2-6Samples labels of early version of HailBot

Virus name	Trojan/Linux.Mirai[Backdoor]
MD5	C4526600A90D4E1EC581D1D905AA6593
Processor architecture	x64
File size	68.6 KB (70,295 bytes)
File format	BinExecute/Linux.ELF[:X64]
Digital signature	None
Packer type	None
VT first upload time	2024-02-23 06:43:16
VT test results	41/66

Table 2-7 Samples labels of HailBot new variant

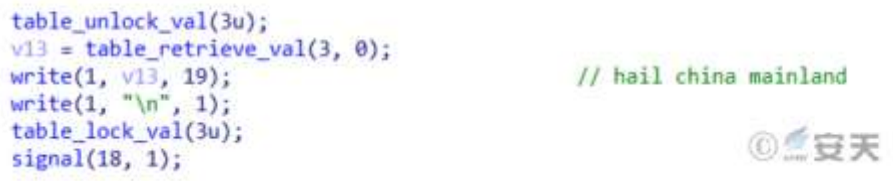
Virus name	Trojan/Linux.Mirai[Backdoor]
MD5	2DFE4015D6269311DB6073085FD73D1B
Processor architecture	ARM
File size	74.7 KB (76,572 bytes)
File format	BinExecute/Linux.ELF[:ARM]
Digital signature	None
Packer type	None

VT first upload time	2024-09-23 18:35:26
VT test results	42/63

Note: As the Antiy AVL SDK antivirus engine has strong pre-processing capabilities, it can detect deformed derivative samples with fewer high-quality rules. As of the time of writing this article, the detection results of HailBot samples are all Mirai. This is hereby stated.

2.3.2 Behavior Analysis

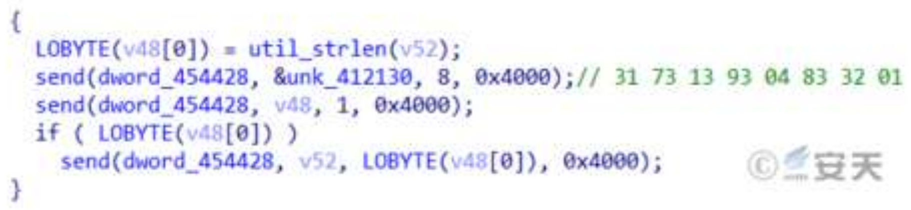
When HailBot is running, it will output "hail china mainland" to the console, which obviously has the purpose of framing China and covering up its own origin. Its expression is also obviously inconsistent with the logic of the Chinese language.



```
table_unlock_val(3u);
v13 = table_retrieve_val(3, 0);
write(1, v13, 19);                              // hail china mainland
write(1, "\n", 1);
table_lock_val(3u);
signal(18, 1);
```

Figure 2-7HailBot outputs a specific string to the console

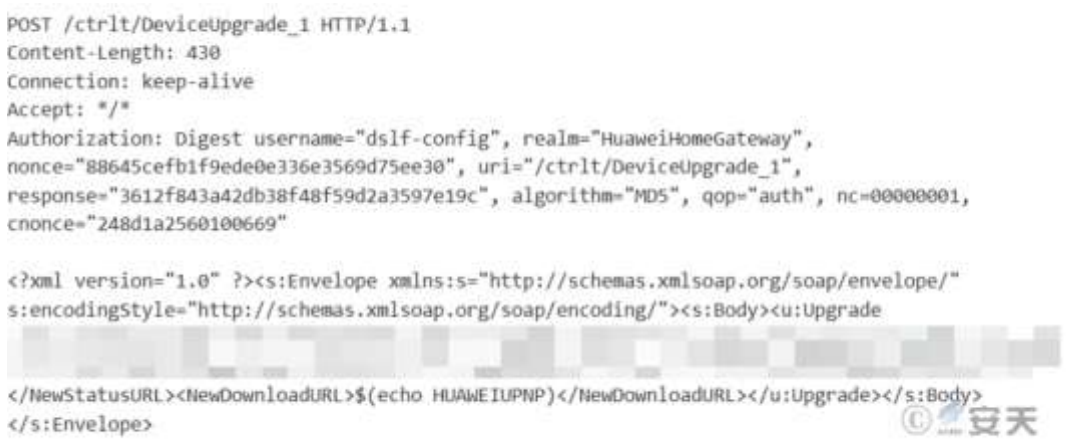
When Mirai series of botnets go online, they send online data packets to the C2 server. The original online data packets of Mirai are four bytes with the content |00 00 00 01|. HailBot modifies them into eight bytes |31 73 13 93 04 83 32 01|, allowing the C2 server to recognize that its traffic comes from HailBot. At the same time, it also avoids being detected by the security scanning mechanism using the original online packets.



```
{
  LOBYTE(v48[0]) = util_strlen(v52);
  send(dword_454428, &unk_412130, 8, 0x4000);// 31 73 13 93 04 83 32 01
  send(dword_454428, v48, 1, 0x4000);
  if ( LOBYTE(v48[0]) )
    send(dword_454428, v52, LOBYTE(v48[0]), 0x4000);
}
```

Figure 2-8 9

HailBot spreads by exploiting vulnerabilities, including the long-term use of CVE-2017-17215, which exists in the UPnP (Universal Plug and Play) service of a specific version of routers. Attackers can exploit this vulnerability by sending specially crafted HTTP requests to execute arbitrary code on the device.



```
POST /ctrlt/DeviceUpgrade_1 HTTP/1.1
Content-Length: 430
Connection: keep-alive
Accept: */*
Authorization: Digest username="dslf-config", realm="HuaweiHomeGateway",
nonce="88645cefb1f9ede0e336e3569d75ee30", uri="/ctrlt/DeviceUpgrade_1",
response="3612f843a42db38f48f59d2a3597e19c", algorithm="MD5", qop="auth", nc=00000001,
cnonce="248d1a2560100669"

<?xml version="1.0" ?><s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"><s:Body><u:Upgrade
[redacted]
</NewStatusURL><NewDownloadURL>$(echo HUAWEIUPNP)</NewDownloadURL></u:Upgrade></s:Body>
</s:Envelope>
```

Figure 2-10Partial payload of HailBot exploit

Early version of HailBot has 3 TCP attack methods and 1 UDP attack method, while the latest version has been upgraded to 5 TCP attack methods and 3 UDP attack methods, which gives attackers more options and combinations of attack methods, posing a greater threat.

Table 2-8The old version of HailBot commands

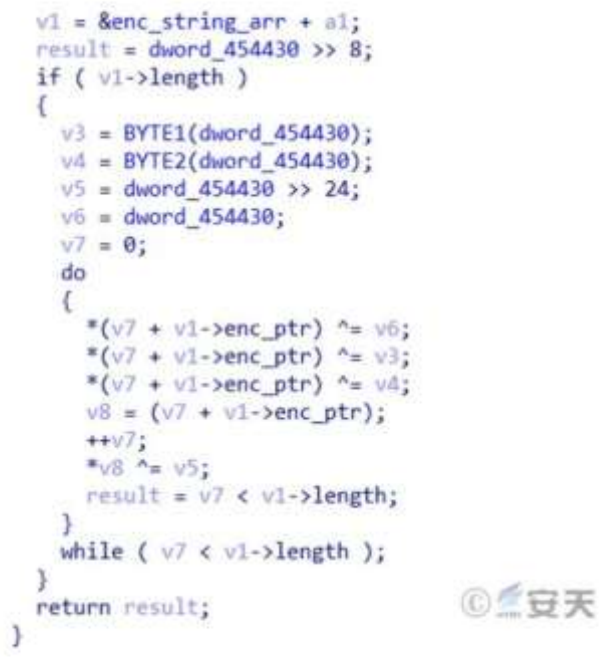
Instruction number	Function	Influence
0	TCP flood attack	Consumes the victim's network bandwidth by creating connections and sending large numbers of 500 to 900 bytes of TCP requests.
1	UDP flood attack	Consume the victim's network bandwidth through a large number of UDP requests.
2	GRE IP flood attack	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.

Table 2-9The new version of HailBot commands

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	It uses 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.

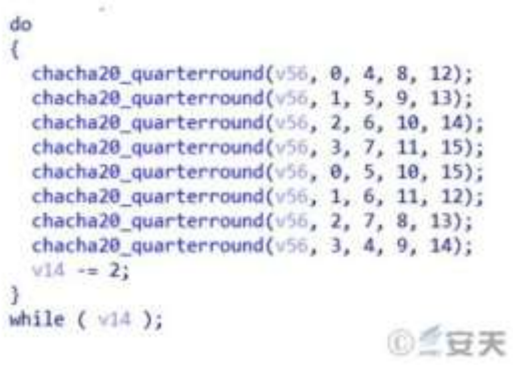
In addition, HailBot has also changed its encryption. In early versions, HailBot used a simple XOR algorithm to encrypt strings such as C2 addresses and attack payloads, where the bytes used for XOR were constants and hard-coded in the program.



```
  v1 = &enc_string_arr + a1;
  result = dword_454430 >> 8;
  if ( v1->length )
  {
    v3 = BYTE1(dword_454430);
    v4 = BYTE2(dword_454430);
    v5 = dword_454430 >> 24;
    v6 = dword_454430;
    v7 = 0;
    do
    {
      *(v7 + v1->enc_ptr) ^= v6;
      *(v7 + v1->enc_ptr) ^= v3;
      *(v7 + v1->enc_ptr) ^= v4;
      v8 = (v7 + v1->enc_ptr);
      ++v7;
      *v8 ^= v5;
      result = v7 < v1->length;
    }
    while ( v7 < v1->length );
  }
  return result;
}
```

Figure 2-11Early versions of HailBot encryption algorithm

In subsequent versions, HailBot switched to using the chacha20 stream cipher encryption algorithm to encrypt strings, which increased the strength of the string encryption and reduced the static features of the strings, making them more difficult to detect and analyze.



```
do
{
  chacha20_quarterround(v56, 0, 4, 8, 12);
  chacha20_quarterround(v56, 1, 5, 9, 13);
  chacha20_quarterround(v56, 2, 6, 10, 14);
  chacha20_quarterround(v56, 3, 7, 11, 15);
  chacha20_quarterround(v56, 0, 5, 10, 15);
  chacha20_quarterround(v56, 1, 6, 11, 12);
  chacha20_quarterround(v56, 2, 7, 8, 13);
  chacha20_quarterround(v56, 3, 4, 9, 14);
  v14 -= 2;
}
while ( v14 );
```

Figure 2-12HailBot's chacha20 rounds operation code

3 Summarize

Comprehensive sample analysis shows that the attack mode of HailBot and RapperBot mainly relies on the massive botnets under their control, which continuously consume the target host's bandwidth resources, TCP connection pool capacity, and CPU computing power required for connection processing by sending forged request packets at a high frequency. Although this attack method belongs to the category of traditional DDoS (distributed denial of service), it is still one of the core threats to Internet infrastructure. The fundamental contradiction lies in the fact that the defender's resources naturally have an upper limit (such as computing performance and network throughput), while the attacker can continue to expand the scale of the botnet at almost zero marginal cost through automated scanning, malicious code injection, and other means, forming asymmetric resources between the attacker and the defender.

After DeepSeek became popular around the world, it experienced exponential growth in the number of users, API calls, and concurrent requests in a short period of time, causing the underlying infrastructure to always be in a critical state of high load. In this context, coupled with large-scale DDoS attacks (such as launching massive text generation requests through botnets), it directly caused a surge in service response delays, API flow limiting and circuit breaking, and even cluster overload and downtime, seriously affecting user experience and business continuity.

For Internet resource service providers, the methods of preventing DDoS attacks are relatively mature, requiring in-depth integration of resource investment and normalized security operations, and requiring cooperation and coordination among service providers, infrastructure providers, and regulatory agencies. This includes deploying a more flexible distributed, multi-regional, multi-link service architecture, using load balancer equipment and strategies, enhancing bandwidth and hardware facilities, and improving system throughput; it also includes improving security monitoring, traffic cleaning, and dynamically adjusting relevant security policies.

For the security operation of government and enterprise organizations, it is necessary to strengthen protection to prevent devices from becoming zombies of botnets, which also contributes to curbing the spread of botnets. Timely discovery and disposal of infected nodes is very important. From the analysis of this article, the basic security management of terminals and IoT devices is the key, including the modification of default passwords, access policies for managing network devices and IoT device management ports, and timely firmware patch upgrades. In addition, the end-cloud side needs to deploy antivirus, EDR, CWPP and other security products with effective protection capabilities to build the security cornerstone of the system.

For regulators, DDoS governance involves a large amount of resource linkage, especially international governance coordination, which will further increase the difficulty of governance. It is necessary to improve more powerful technical resources and system capabilities at the national security and public security levels.

While improving basic protection and security governance, we need to pay further attention to the risk evolution of new technologies. The development of new technologies is always tied to the dynamic evolution of security threats in three ways: bringing new threats, promoting the escalation of traditional threats, and becoming an attack target itself. Generative artificial intelligence and large model technology are no exception. They have promoted the automation level of traditional attack technologies, brought about the rapid maturity of attack technologies such as deep fakes, and themselves have become high-value targets.

Compared with traditional Web services (such as CGI dynamic pages or search engines), the computing power consumption of generative AI for a single interaction is significantly higher, and the open API interface can be easily abused by attackers as a computing resource black hole. The business characteristics and risk scenarios of large model platforms are significantly special, so we need to be more vigilant about the risk of computing resource attacks. To support high-concurrency reasoning requests and long-context interactions, the platform needs to deploy large-scale GPU clusters and real-time scheduling systems. Attackers can design low-traffic and high-lethality precision attack chains for such computing-intensive and low-latency sensitive characteristics, such as maliciously constructing model parameter queries (such as triggering high-dimensional tensor calculations), where a single request can consume GPU resources several times that of regular tasks; context injection attacks, by implanting specific prompts (prompts), force the model to perform recursive parsing, causing CPU/memory resource exhaustion. The cost-effectiveness of such attacks far exceeds that of traditional DDoS (services can be paralyzed without massive zombie nodes), and it is easier to bypass protection strategies based on traffic thresholds.

At the same time, we also need to pay more attention to the data security risks of big data platforms: due to the problems of multi-tenant data interweaving storage and fine-tuning parameter residue involved in the training and reasoning process of large models, sensitive information leakage may occur (such as user privacy data leakage through the model output side channel). In the process of data column labeling, files with malicious code are not cleaned, which may cause the risk of virus infection and spread in the working environment of the labeling engineer and the data platform, system performance loss, and even data being blackmailed and stolen. It is necessary to consider strengthening the virus cleaning of the corpus file data to be labeled, enhancing the fine-grained isolation control capabilities and security detection and protection capabilities in the east-west direction, and thus enhancing the threat resistance capabilities of the large model platform.

Therefore, the security construction of large model platforms needs to be carried out on two tracks: on the one hand, improve the security of the infrastructure: strengthen defense, monitoring, resource isolation and other mechanisms at the cloud host, container cluster, API and other levels, that is, effectively defend against the risk of penetration and intrusion, and also use elastic expansion and contraction and real-time fuse mechanisms to resist resource exhaustion attacks; on the other hand, it is necessary to improve security capabilities from the architecture, design, business logic and coding optimization levels, including but not limited to: through prompt word injection detection, reasoning process sandboxing, data lineage tracking and other technologies, build a deep defense system at the model interaction layer. Deeply embed security capabilities into the technical architecture and business flow.

History has proven that the security gains in dealing with new technology risks often come from the new technology itself. Historically, the Internet has comprehensively improved the accessibility of attacks and has become a hotbed for large-scale attacks; but it has also improved the agility of security operations. Cloud computing platforms have introduced overall overturning risks, but they have also brought greater resource elasticity and unified and efficient security governance. Artificial intelligence technology is also rapidly changing the capabilities and appearance of network security. Antiy itself is also a practitioner in the network security industry who actively embraces big model technology. Antiy LanDi VILLM [8]focuses on binary sample analysis feature engineering scenarios, breaks through token context restrictions, and can already run in CPU scenarios. Antiy Computer Virus Classification Encyclopedia is also the result of our automated operation with the help of our own feature engineering and knowledge system. In areas where we are not good at, we have also adopted a strategy of actively embracing excellent domestic big models. In the preparation of this report, we used DeepSeek as an aid.

From the launch of Black Myth: Wukong to the explosion of DeepSeek, China's information technology is constantly creating new legends. At the same time, it is also accompanied by cyber attacks. Overcoming these risks is precisely the verification of the invincibility and bright future of new things. As a private enterprise national team that has long provided common security capabilities for the cybersecurity industry system, Antiy is willing to provide more common security genes for strategic emerging industries and escort great new things.

4 IOC

MD5
71B4C3FE502E6C6D5EF5E420D52D2729
C4526600A90D4E1EC581D1D905AA6593
6C6D1CCCE5946F0AA68F9E0C438C1E21
B1E0B2C046D4CB7F0A0DD87054A17AC4
6B8A9D6335D056E20DCD794B265074E3
AB2C4A13D1FE946003FFCB7DDEC064D0
9E331675D780AF4585857B1F95B40CBB
EFA786F2B6F0F267F717145FAF48A95B
EF9EBF4D5A1A44D0DB92DE06D3DCE7A1
BEC7596CFB1225900673398ABB24FFA8

Appendix 1: References

[1]. Qianxin. Botnet Enters the Scene, And the Network Attack Against DeepSeek Escalates Again [R/OL]. (2025-01-30)

https://mp.weixin.qq.com/s/NM-zCyA4m5WJeAjPwUmYYg

[2]. ANTONAKAKIS M, APRIL T, BAILEY M, et al. Understanding the Mirai Botnet[R/OL]. (2017-08-16)

https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis

[3]. Wikipedia. DDoS attacks on Dyn[R/OL]. (2016)

https://en.wikipedia.org/wiki/DDoS_attacks_on_Dyn

[4]. Antiy. Computer Virus Encyclopedia Mirai Corresponding Entry [R/OL]. (2016)

https://www.virusview.net/malware/Trojan/Linux/Mirai

[5]. Antiy. Antiy PTA Team Analyzes New Propagation Methods of Mirai Variants[R/OL].(2016-12-19)

https://www.antiy.cn/research/notice&report/research_report/608.html

[6]. Diamonds Is Lit (Official Video)

https://www.youtube.com/watch?v=fPu9hTClNWQ

[7]. Fuckbosco

https://soundcloud.com/xxdannyflandersxx/fuckbosco2-a-danny-flanders-special-release

[8]. Antiy. Artificial intelligence technology empowers network security application testing in 2024: Antiy LanDi VILLM Begins to Show Its Edge in Malware Detection Scenarios[R/OL]. (2024-09-23)

https://www.antiy.cn/About/news/20240923.html

Appendix 2: 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.