Agent Tesla: A Lesson in How Complexity Gets You Under the Radar

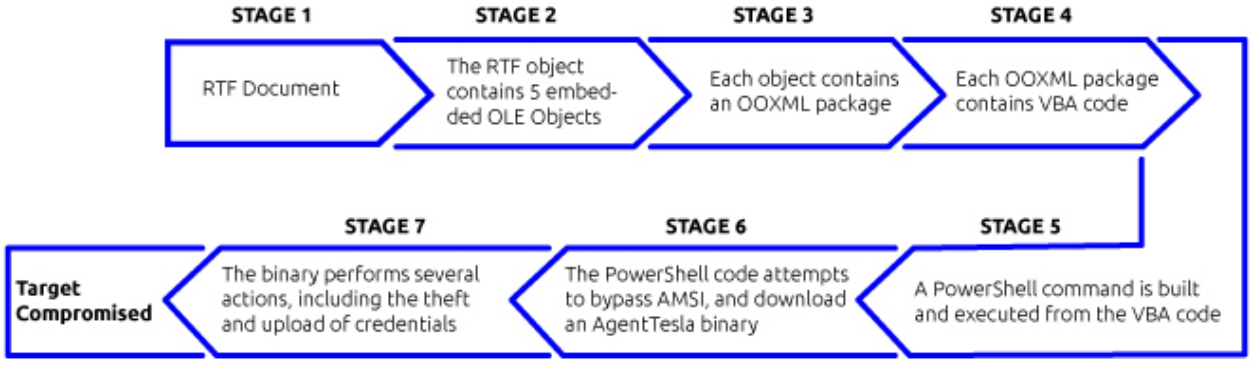
By Guy Propper

Published: 2020-07-02 · Archived: 2026-04-05 14:59:13 UTC

Agent Tesla is a prolific strain of spyware, that is being sold online since 2014. It is advertised in dark-web forums as a legitimate monitoring software not intended for malicious purposes. However, its extensive password extraction features are clearly used for malicious purposes by many actors.

Deep Instinct's Research Team recently came across a very interesting infection chain found in one of our production sites. The uniqueness of the infection chain is due to its long and inordinately complex process; starting with an RTF document attached to a phishing email, it ends with the dropping of an Agent Tesla executable on the victim machine. The multiple stages of the infection process include the use of OLE Objects within the RTF document and the execution of obfuscated VBA code contained in OOXML packages within the OLE Objects. In turn, the VBA code executes Powershell, which finally drops the malicious executable. Dozens of similar RTF droppers were found to be active very recently, possibly indicating a wide attack wave of Agent Tesla, utilizing this infection process.

The full infection flow is explained in the following diagram:

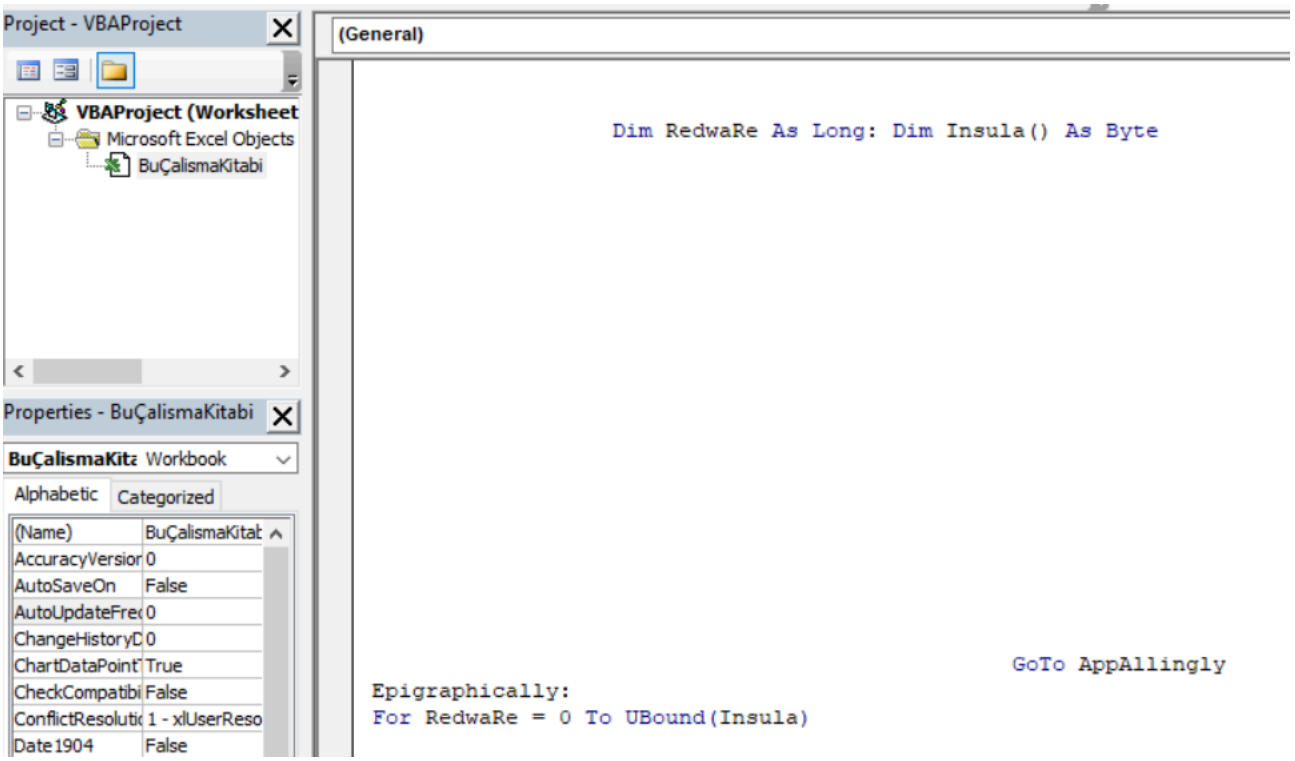


Infection flow

Phishing RTF

The infection chain begins with the execution of an RTF file, which arrives as an attachment in a phishing email. Once executed, the user is presented with five consecutive requests to enable macros. The five requests are due to the fact that the RTF contains five embedded OLE objects, which each contain an OOXML package. Inside each OOXML package lies a VBA macro, and when the user enables macros, one of the macros will execute at a time.

The VBA code contained in each of the OOXML packages is rather short and has been designed with many spaces and line breaks to make reading and organizing the code difficult



A snippet of initial spaced VBA code.

To hinder static analysis of the code, the main variable used in the code is contained in a specific cell of the spreadsheet in each OOXML package. The variable is a long obfuscated string, which can be found in xl/sharedStrings.xml in the OOXML package.

Indeed, static analysis tools such as oletools and oledump did not help in providing any details regarding the functionality of the VBA code. In addition, dynamic analysis of the VBA using ViperMonkey was not successful. However, manual debugging of each of the five VBA code parts revealed that each part is responsible for creating part of a PowerShell code, which will form the next stage of the infection process.

A snippet of Powershell code created from the execution of the VBA macro in the first OLE Object.

Once all five parts are run, the resulting Powershell code is executed.

PowerShell execution

The PowerShell code formed in the previous step is highly obfuscated

Obfuscated Powershell, with a large encoded blob).

After debugging the code, which is deobfuscated through the function af23a, it is still obfuscated, but its function becomes clear – the main purpose of this PowerShell code is to attempt to bypass AMSI, and download a file using WebClient().DownloadFile.

 Partially deobfuscated PowerShell code. Red squares are obfuscated strings responsible for AMSI bypass, and blue square is the download URL

Partially deobfuscated PowerShell code. Red squares are obfuscated strings responsible for AMSI bypass, and blue square is the download URL

The AMSI bypass is attempted through the provision of an empty buffer to the AmsiScanBuffer function – the strings which are relevant to the bypass are the red squares in the above image and can be deobfuscated using function af23a in the PowerShell script. This has already been attempted in the past by a very similar Agent Telsa infection process.[1] The URL which is accessed to download the file, which appears obfuscated as the string '09411248125b1a495b0d044707560e0753075b040c1b05570c4e5b04501804470217030e580416041950', is de-obfuscated to "hxxps://cleranoffacem[.]com/nbhyerd/bomb[.]exe" (blue square in image 4), and the file downloaded from this URL is Agent Tesla, which will be overviewed in the following section.

AgentTesla download and execution

The AgentTesla executable is download from hxxps://cleranoffacem[.]com/nbhyerd/bomb[.]exe to AppData\Roaming\u565.exe. Then, the executable starts performing several tasks:

- Creates a scheduled task using schtasks.exe, to execute the AgentTesla executable.
- Disables task manager through the registry using reg.exe. The specific command used is “REG add HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System /v DisableTaskMgr”.
- Searches for WIFI passwords using netsh wlan show profile.
- Tries to steal a variety of credentials: putty/WinSCP, browser, FTP, and Mail credentials.
- It then sends stolen credentials to dir.fb@tolipgoldenplaza.com, including the credentials Golden@#$2019.

An Earlier Precedent?

It is interesting to note that a very similar infection flow involving an RTF file that contained five OLE Objects was identified in March 2018.[2] In that instance, the file that was dropped following the infection chain was Lokibot. Considering the striking similarity between the two cases, it appears that the attack was either executed by the same actors or created using the same framework. However, we could not find data to further elucidate the greater likelihood of the two options.

In addition, despite this method being several years old, it is apparently still used effectively in the wild, with dozens of similar RTF files found in a recent attack wave.

Conclusion

The sophisticated and complex infection chain covered in this article, while not new, is still being used extensively in the wild. This indicates that a complex attack chain, involving many stages, is not only difficult to analyze but can also help attackers evade detection. In this case, evasion may be achieved through the use of multiple stages,

each responsible for only a small portion of the attack, making each stage more difficult to detect. In addition, in this attack, and many other attacks in recent years, internal Windows tools are being abused by the attackers. In this attack, the tools abused were schtasks.exe, reg.exe, and netsh, evidence of the continued trend of attackers to abuse dual-use tools.

Deep Instinct's customers are protected from this threat, which is prevented at multiple execution stages. The initial RTF dropper was prevented in production pre-execution, using deep learning-based static analysis. Moreover, this RTF dropper was prevented with Deep Instinct's prediction model (D-Brain) released more than 20 months prior to the appearance of this dropper. If the dropper were to execute, PowerShell execution would be prevented with Deep Instinct's script protection, and the Agent Tesla executable is prevented both statically using the D-Brain, and dynamically using advanced behavioral analysis protection.

IOCs

RTF: ce212984a9ed60ef6015bfb2f930a0f501a2f6f373c9fa68af54fe8f68d4de9e

Agent Tesla download URL: hxxps://cleranoffacem[.]com/nbhyerd/bomb[.]exe

Agent Tesla: 756feeaec24bcada5d473a53931ac665c2a159083f408d41e7fe1c8fcb0b9a6b

Similar RTF files

840a22c718e33120f6e47c310497148ca903912a46458fbf9f21edc8976074ce

842ad0c1407a7c87c9f76a7a55d56f36dfef501495f56dbad4d28f04b807b63a

b0f8dd641769a080b640dbaa2666b5982344642335372ee4680fa5a6e771991d

ce212984a9ed60ef6015bfb2f930a0f501a2f6f373c9fa68af54fe8f68d4de9e

c03f438d814bd52be15b47743b44519263aaeded731dcfac7e9070628a41d70a

20ae23fa54d2f997c50f85b9977899255822fbe200e17d933b430561adcd1e12

859a9f0c613775907c2cda4d946159e7991ee6f9be430fe5658e95e7e5a0388b

a60c7244206b635d18c244028c1b1dc4c07da716e0ff78529692bc667f117195

2bbc9c51a29557cf8934de723236bf2f5683391d3d57d7d86410221d30b53bd3

3fe1d15c026ad8fa1c510ac3d4982f38be59e84cef34119fff0aad6fad35bc54

f11ee07c633a0ad6a88ec9cb3e798dda02d6459b5eb35eb00d403d8445b0c554

402f2be1b65ae460898ccbf47a475430cc5c64c548228481ad062934f6a85aa2

eec9b14da6a2745f089361002429d13b044d66dedf944e951b39f9d243ae3df9

786f2eaa675e1ee953a159eb4a4ccb734b1adf16ede28dd7b801df9a612a4167

fd26d992e3014118d345027e8a3c482519d75ef0fda12241d244e3a80abeda67

2f9d34c9752df5565c79ed5d0dab3e4c48f5c3de22f54180388a90e3e0b30c9a

d8be93b858f4ddfe0f6dab717e269665a56d862b86781da908fafa31be2ec509

518eb357618f85a419cdeba49b45f8a98441a6a2df1edebb2376cd0a0e98f56f

256777b273432143492346edc89f678e386cb4569e8fd48645e28245977f5856

6d0636869e65966bbb79fb58a0af016e9af41420978a43b5c2eb1ed462a24724

a114858d777f74faafadca52424a9fca33426dc5f3c4777453348e359115ac6d

bf36d5e468b5c654a47ebf07b4a0ef9e192307674960f7fdf22d6e3cb3e85177

6189ddb04b9bbb45474ed48c6685d316c06458da3d9b430727ade08cc344f235

dc1b5e7c4aeb32c2370fc03983502639d31c2c4fdecdb12b6248351daa38129a

d7f2a3ec1aae489bc44b7819ce6f4e5029282b8f8d2064fccfe1804278c38d11

d6779d721788c2826a9cd43cb01c3279c8aaca4a3210c5331125c08a9be32557

1a8ee2fcf777abbcc6d3eda5a52f5cdb2269cc8a6e7e339b01c04d47138bb702

a16cdca08584f03a1deaefa94393914bb317e80bd2a2b9f5da7c0b4355a1fddd

52f2e17287a2f975d30fdda43b44c67b5f70a168ccf97696b7d95a962d46dd7a

167760bf97f12f6ef1d66ca2db17a5a0ed2d594f86f3d8716c83e7d66d502f3e

0d873ad2a42333ee77bb18bb92c920afe94fe3c108de28fc4bb89901eb12161c

8ac06f7b667d0ae9fc2e0940efba2d580af0dab54825275b7f85cb5ac37c6f05

e5ade604474407fc742a5b99996b1aae86695493eb71d5fc2478fb78238a0799

c4d7f76ca3ccc9a7f8763e4688cc2660a1164674f14c86fd384153b5e2fa566f

b2c6e93875ed9728da141566603ad47a71a82d3867313744ceca367158c2b20c

356c459692775dae1f20998c5d39f51a4b94ac01de509fa609844eee8adab19f

[1] https://r00tten.com/in-depth-analysis-rtf-file-drops-agent_tesla/

[2] https://twitter.com/DissectMalware/status/977087605719302144
