Global Information Assurance Certification Paper

An Introduction to Metasploit Project for the Penetration Tester

GSEC – GIAC Security Essentials Certification
Version 1.4c

Brandon Greenwood
Date: March 09, 2005

Table of Contents

Abstract	**3**
Introduction	**3**
Metasploit Project	**4**
Installation of the Framework	**4**
Windows	**5**
Linux	**5**
Pre-Metasploit Groundwork	**5**
Using the msfconsole for testing	**7**
Global settings and temporary environments	**14**
Using msfweb for testing	**14**
Advanced Features	**17**
Conclusion	**18**
References	**19**

Abstract

This paper sets out to inform the reader about the benefits and uses of the Metasploit Project (MP) focusing mainly on the Metasploit Framework version 2.3 and its many uses. An example of a successful vulnerability exploit will be presented with the approach being that of not knowing anything about an environment/workstation or a black box test.

The paper will begin with the installation of the framework (Microsoft and Linux), to implementation and on down to the features that are still actively in development. It is hoped that the reader will not only be able to come away with an awareness of the power of the framework, but also be able to make the tools work for them in their own environments.

Introduction

Penetration testing can be considered somewhat of a black art in the information security realm. Ask any number of security practitioners what methods they might use for penetration testing, and you will get answers like Nessus[1] or Internet Security Scanner[2] {ISS). While they are good vulnerability assessment tools, they can't truly be considered penetration testing tools. Others might tell you that they roll their own, (i.e. take pre-existing code or exploits and either compile or modify them to their liking). But then there are also those who actively seek out and discover vulnerabilities and then find ways of exploiting them. Not everyone in the security field has the time, dedication, or the know how of the latter group. Some practitioners might have one or two, but not all three of these qualities. Those same people now have a tool to aid in whatever area they are lacking. The MP aims to assist those people.

Network vulnerability assessment has long been an area within information security where there have been tools capable of allowing a vulnerability/penetration tester (VPT), to verify whether systems and or services are available and, depending on the tool, whether or not there is the possibility of a vulnerability associated with it. These tools have ranged from standard system tools such as ping, telnet, finger, and netstat, to more customized tools whose features allow the VPT to test for specific vulnerabilities. These tools include but are not limited to the open source standards such as SATAN[3], SARA[4], Nmap[5], and of course Nessus to the commercial offerings from ISS, eEye (Retina)[6], CA (eTrust Vulnerability Manager)[7], and GFI (LNSS)[8].

Once results have been generated from any of these tools, a VPT may have the tendency to take them at face value and not take into consideration the possibility of false positives. While using a second or even a third scan using a different product is an option, this can be very time consuming depending on the types of tests and the level of knowledge needed per tool to become proficient enough to have a high level of accuracy and confidence within the results of those tests.

How can a VPT make sure that the results they present to an organization, an employer, or a boss are valid and limit the likelihood of erroneous results? By testing the results of a vulnerability assessment a VPT can add another separate method of determining whether or not vulnerability truly exists.

In-house applications are notorious for their lack of security and exploitable features. Even with commercially available tools, it could take a VPT countless hours to prove or disprove the existence of such flaws. The MP is the open source answer to address this dilemma.

In addition to the VPT benefits, an Intrusion Detection Sensor's (IDS) signatures or an Intrusion Prevention Signature's (IPS) features can be tested using the same methodology illustrated below.

Metasploit Project

So just what exactly is the Metasploit Project? The Metasploit homepage defines exactly what the project is trying to accomplish, "The goal is to provide useful information to people who perform penetration testing, IDS signature development, and exploit research. This site was created to fill the gaps in the information publicly available on various exploitation techniques and to create a useful resource for exploit developers. The tools and information on this site are provided for legal penetration testing and research purposes only."[9] Some of the tools that the project makes available on the MP Homepage are the Framework, the Shellcode, and the Opcode Database.

The Framework is pretty much the meat and potatoes of the project. This is the actual user interface into the project. From here the VPT can develop their exploit, payloads and actually craft a penetration test to their own requirements. One of the major values of the Framework is its extreme flexibility. The Framework is written in Perl allowing it to be easily ported between multiple operating systems. The included components were coded in C, assembler and Python.

The Shellcode is an archive of payloads that have been created and are ready for use from within the Framework. At this time, the payload collection contains payloads for the following systems; Windows, Mac OS X, Solaris, Linux, BSDi, and BSD.

The Opcode Database is a goldmine as far as penetration testing is concerned. It provides a method of searching opcodes[10] for use in modules, showing the opcode types, listing supported operating systems as well as modules, and being able to display module information. This set of resources in a single easy to use location saves an inordinate amount of time when researching and or testing an exploit. Having the opcode or module page immediately available will save countless hours searching the Internet for it.

Installation of the Framework

The installations presented here will go through the installation of the Framework for Microsoft Windows 2000 and Red Hat Linux 9 in a VMware environment. The reason for these two operating systems is that there are a plethora of exploits that we can use against them in an unprotected state making for easy examples in demonstrating the MP's ease of use. Both installations were built solely for the illustration of the Metasploit Framework. They are clean installs with no patches, no antivirus, and no personal firewall of any type. On the Linux box, Nmap 3.81, Nessus 2.2.3, Snort 2.3[11], and Perl 5.8.6[12] have also been installed.

The current version of the Framework at the time of this writing is framework-2.3 for both Win32 and UNIX.

Windows

The installation package for Win32 is quite a bit larger than that of the UNIX Tar Archive. The reason for this is that the MP development team in working with the port for Microsoft systems, had issues in getting ActiveState Perl to work correctly with the multiple components of the Framework. For this reason it was decided to use a stripped down version of the Cygwin[12] environment. Windows users, who are used to a Unix/Linux variant or Cygwin for that matter, should have no problems with this environment. For those who are not as familiar with a shell, don't fret as the syntax is really quite easy to learn.

The framework-2.3.exe can be downloaded from http://www.metasploit.com/projects/Framework/downloads.html. Once downloaded, locate and launch the executable. By accepting the defaults, reading and agreeing to the licensing, you can have the application installed rather quickly. Most of the installations size and time is related again to the Cygwin environment. The total size of the installation is 83.2M. Once the installation is completed you should be presented with the msfconsole (one of three user environments included).

Linux

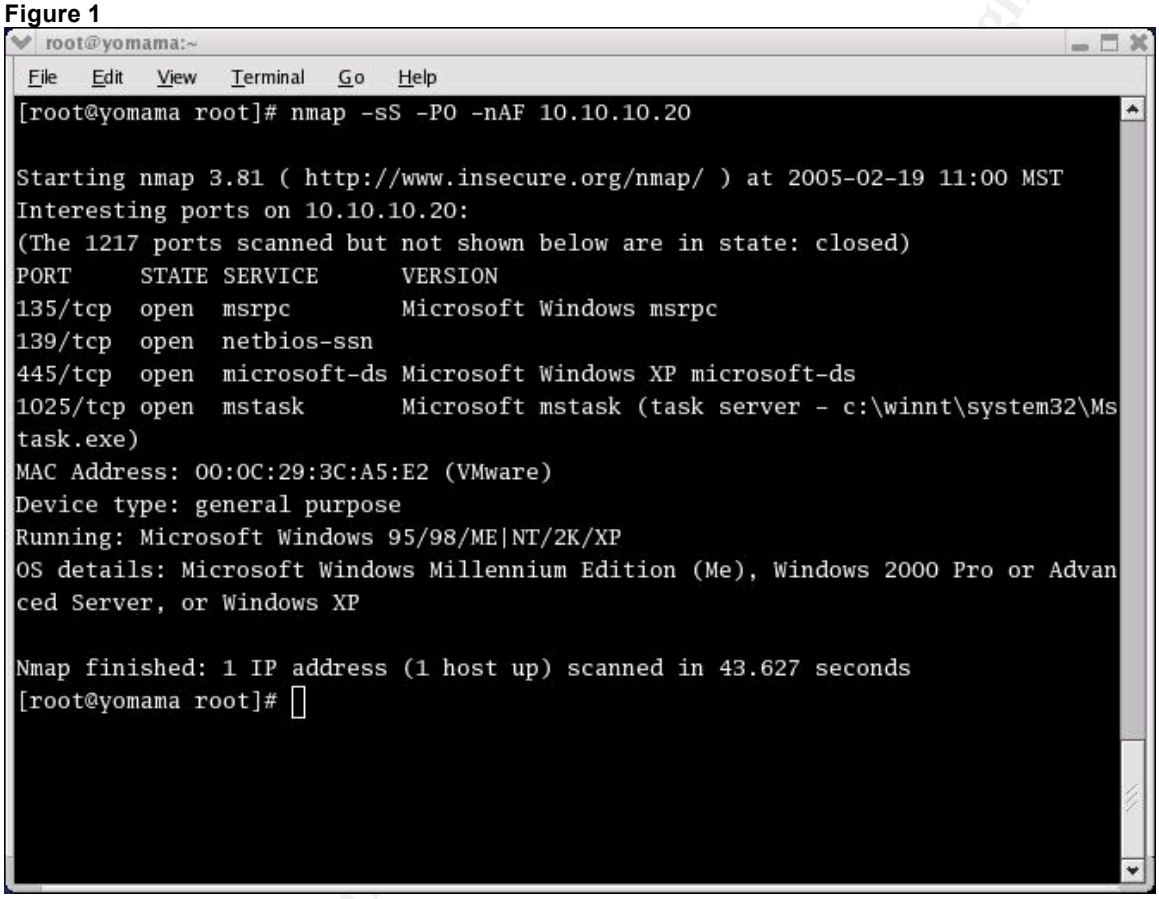
The installation Tar Archive can be downloaded from the same URL as the Windows binary. Extracting the tarball will produce a framework-2.3 directory that will contain everything pre-compiled for use. The directory can now be utilized with the option of deploying it so that only an individual VPT can make use of its functionality or make it accessible system wide.

Pre-Metasploit Groundwork

This assessment begins as it would for a VPT performing black box testing or no prior knowledge about the system except for in this case the network segment where the box resides. The network segment that both virtual machines are configured in is the 10.10.10.0/24 subnet. The Linux machine or the VPT box is

10.10.10.10. The test is done in a fashion that the rest of the segment is unknown and must be discovered. By starting out with an ICMP scan on the local segment, a 10.10.10.20 address is located. Using Nmap to scan this address for services and possibly identify the OS shows that this has a very good chance at being a Microsoft Windows machine. (Figure 1)

Figure 1



By the looks of the results, the VPT can now custom tailor a vulnerability analysis scan so that they are only looking for vulnerabilities inherent in the Microsoft operating systems when scanning this particular machine.

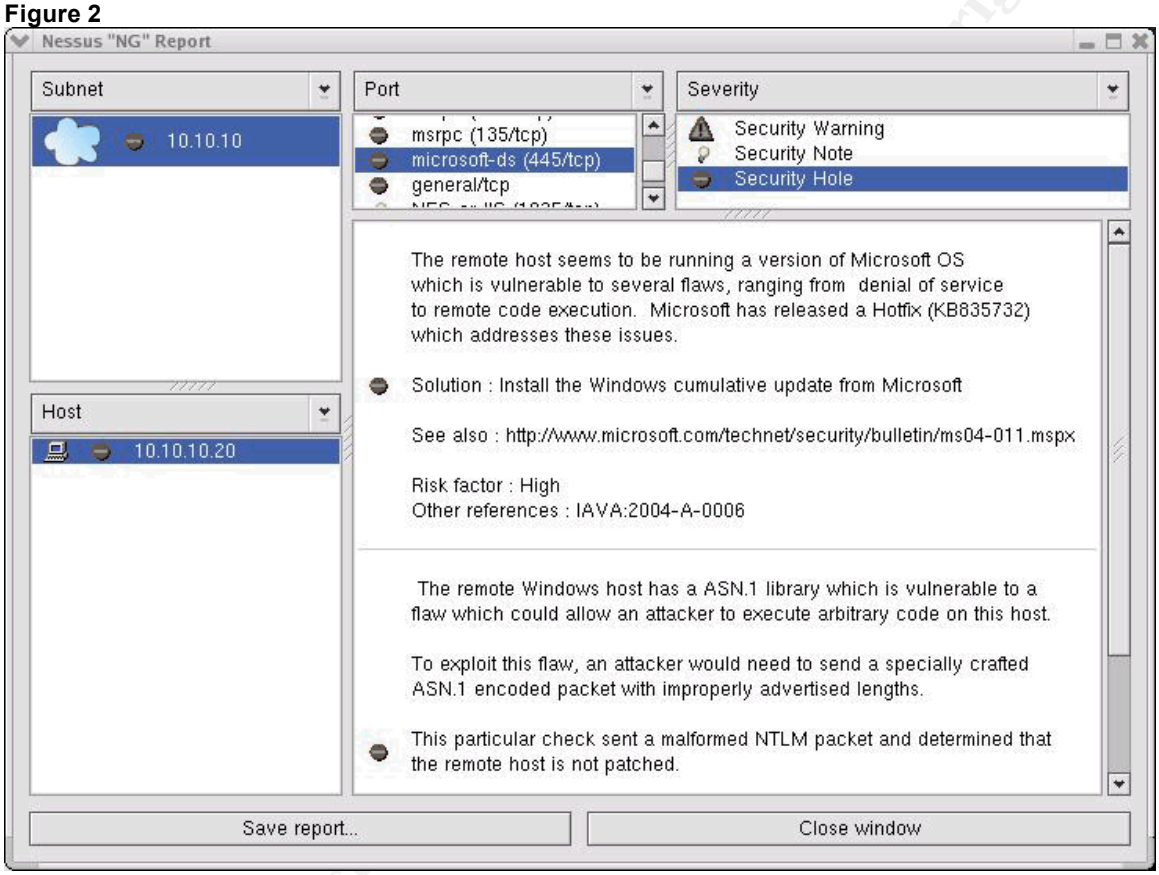
Any vulnerability scanner can be used at this point in the analysis. Nessus will be used here to perform the vulnerability assessment. The Nessus scan, without any user login credentials to the target system, should give us a basic understanding of the OS, any services, any warnings, or any holes that are not being blocked in some fashion on the system, Nessus will be able to show those services for and allow for a deeper inspection of the services.

Nessus can provide a very detailed scan if given the right options. With only the Nmap results available, the Nessus Scan was configured with all windows tests available. If the VPT wanted a more thorough scan of the environment or wanted more information from the vulnerability analysis, I highly recommend

reading “Nessus Network Auditing”[14] from Syngress. This book is a goldmine when working with the powers of Nessus.

The results from the analysis (Figure 2), show that Nessus believes it is possible that there are a couple of security holes over port 445 on the Windows machine. This is the vulnerability that the Metasploit Framework will work to exploit.

Figure 2

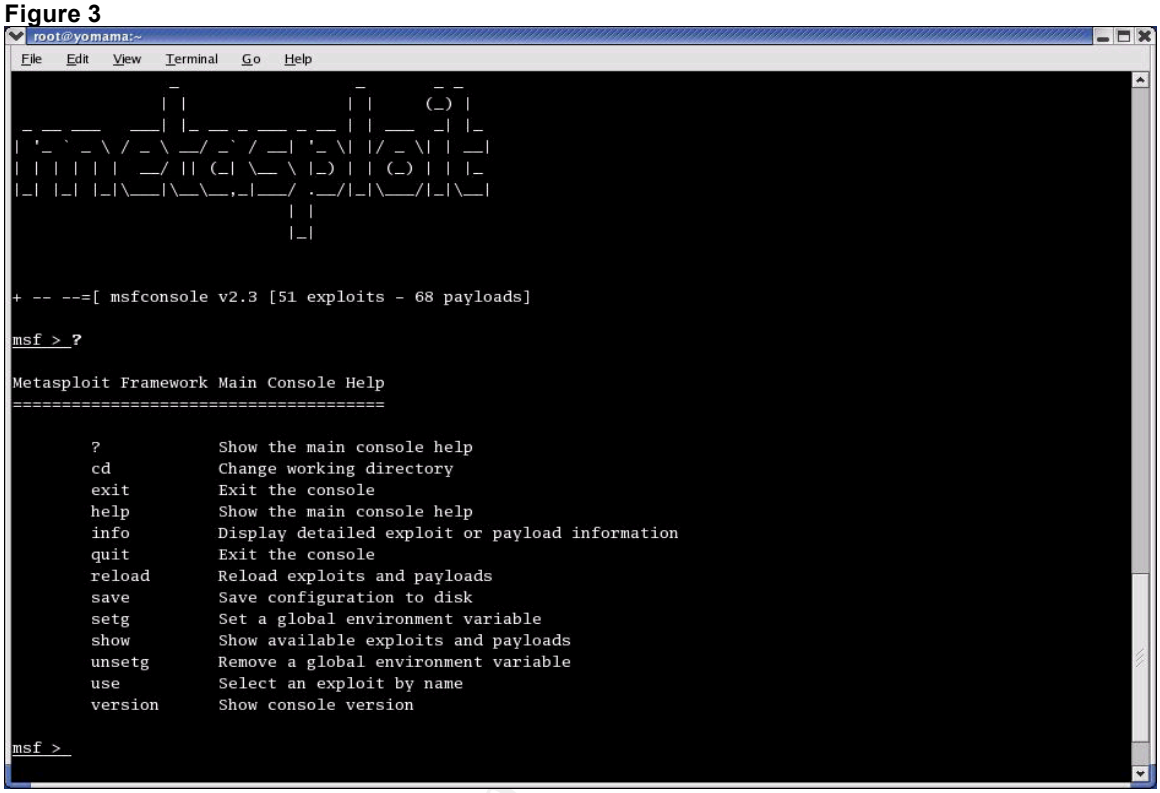


Using the msfconsole for testing

There are three user interfaces for the Metasploit Framework. msfconsole and msfweb are presented here. msfcli is another user interface that works well with scripting. You can script into msfcli any variables, options, or settings you can supply using the msfconsole.

The msfconsole is the method most often used when working within the Framework. Learning the syntax is not very difficult and after a couple of sessions in the environment it almost becomes second nature. If a command is unknown or forgotten, a simple ‘help’ command will show the environment console help or ‘?’, (Figure 3) at any time will show the main console help.

Figure 3



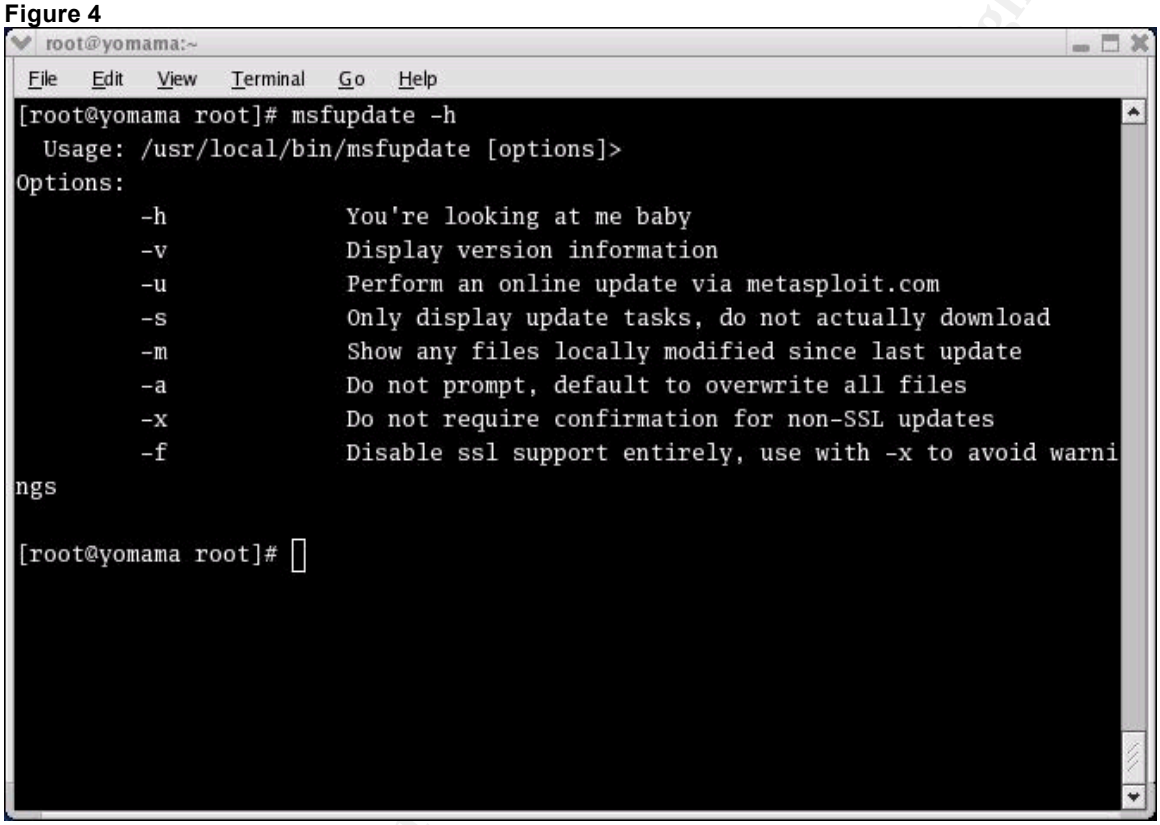
As is illustrated, the version of msfconsole being used is 2.3. This particular version contains 51 exploits and 68 payloads.

There are two options to bring the framework up to date with any exploits or payloads introduced since a particular release or previous update. The first option is to update the framework by downloading the exploit or payload from the MP site and place them in their respective directories, (this depends on how you have deployed the MSF package and where the '/exploit' and '/payloads' directories are located). To be sure that your packages are loaded after placing any modules or payloads in by hand, run the 'reload' command from msfconsole.

The second option is to automate this procedure. The developers have included the msfupdate tool with versions 2.2 and greater. (Figure 4) There are many options here regarding the use of msfupdate, but for a standard update via metasploit.com the '-u' flag is used. The msfupdate command will by default work utilizing HTTPS if the Net::SSLeay Perl module is installed (Net::SSLeay.tar.gz and Term-Readline-Gnu-1.14.tar.gz are included in the '/extras' directory); otherwise the update takes place over HTTP. There is a '-f' flag that disables SSL, but this can introduce other issues relating to hostile code injection. The update will go out to the MP site and download the module

names and checksums. It then compares these results of those existing on the machine that is running the update. An Online Task Update Summary that contains all of the updates is then displayed. The option is then given as whether or not to continue. If 'yes' is selected the updates are processed and the local file database is then regenerated for use with future updates.

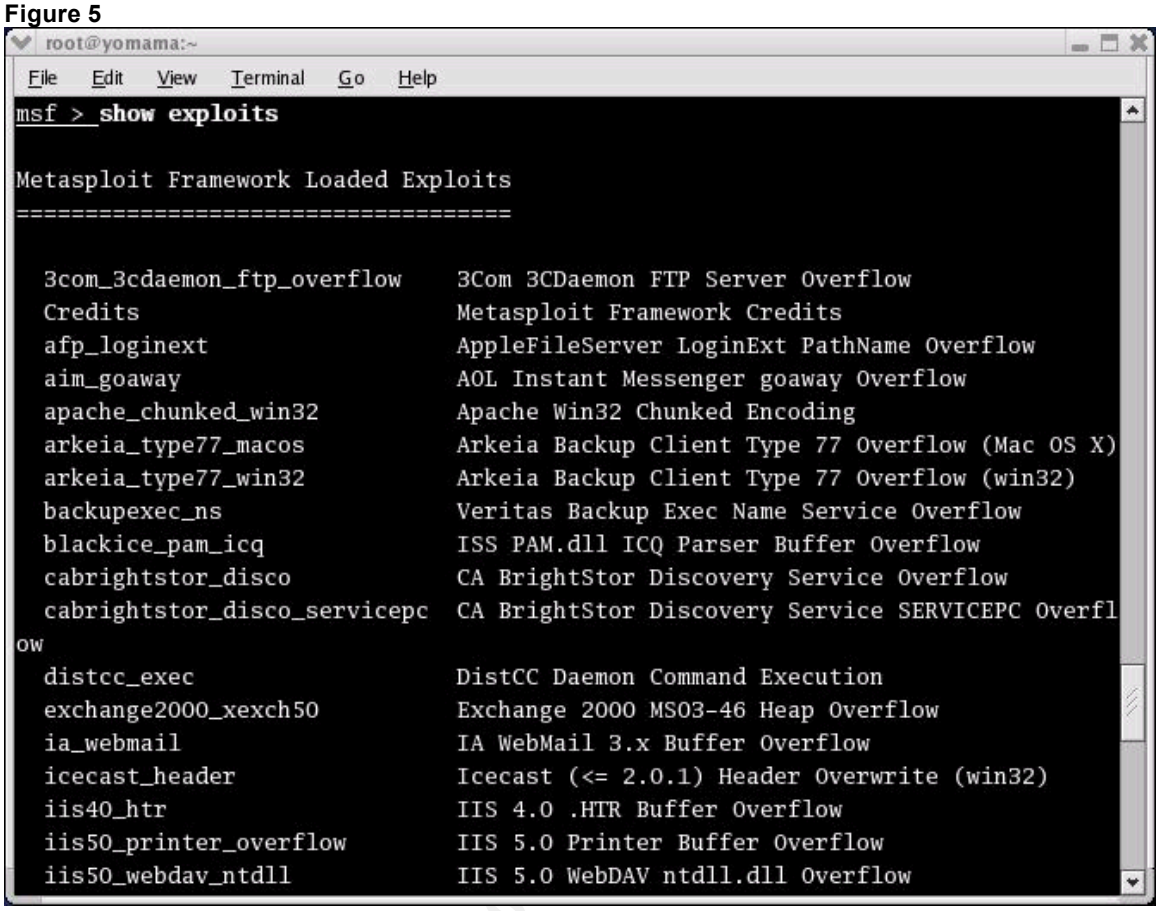
Figure 4



Starting the msfconsole again should now produce different values for the exploits and/or payloads on the metasploit splash screen if updates were available. Now that the msfconsole is up to date, the analysis is ready to begin.

To test the possible vulnerability presented by Nessus, msfconsole is brought into play. A quick check via the 'show exploits' (Figure 5) command, displays some possible avenues of exploitation via this vulnerability. The exploit used in this instance is the 'lsass_ms04_011'.

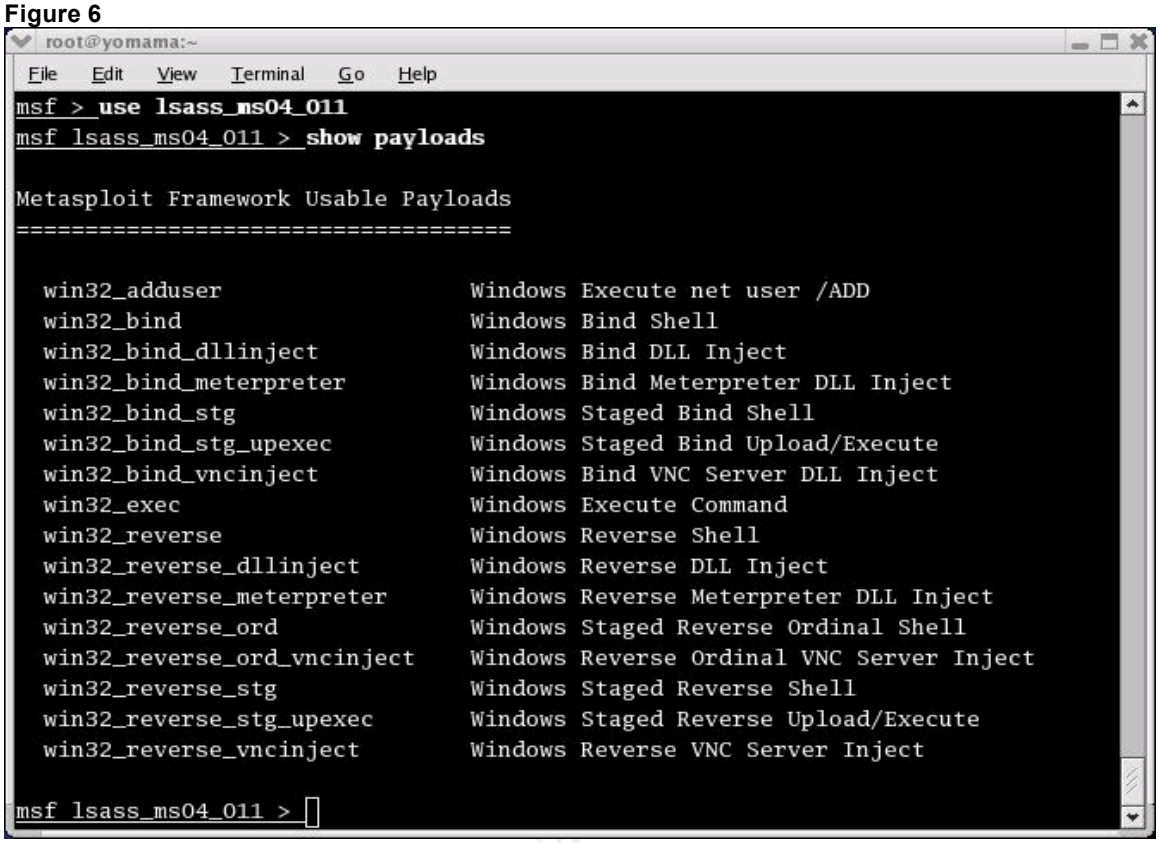
Figure 5



For information regarding this or any other exploit or payload, the 'info *exploit/payload*'' command can be issued to verify that it will do what the VPT intendeds.

In order to use the exploit of choice, think of it as working like a Cisco router or switch. On a Cisco router or switch you can look at and verify certain settings in exec, but to make real changes you have to get into privileged exec. With the msfconsole you can get in and look at settings or payload/exploit information, but to really get work done you must enter the exploit shell. This is achieved by issuing the 'use' command. In this example the 'use lsass_ms04_011' command is issued. This changes the prompt from the 'msf>' to the 'msf lsass_04_011>' prompt. From here we can now check the available payloads available with this exploit by issuing the 'show payloads' (Figure 6) command.

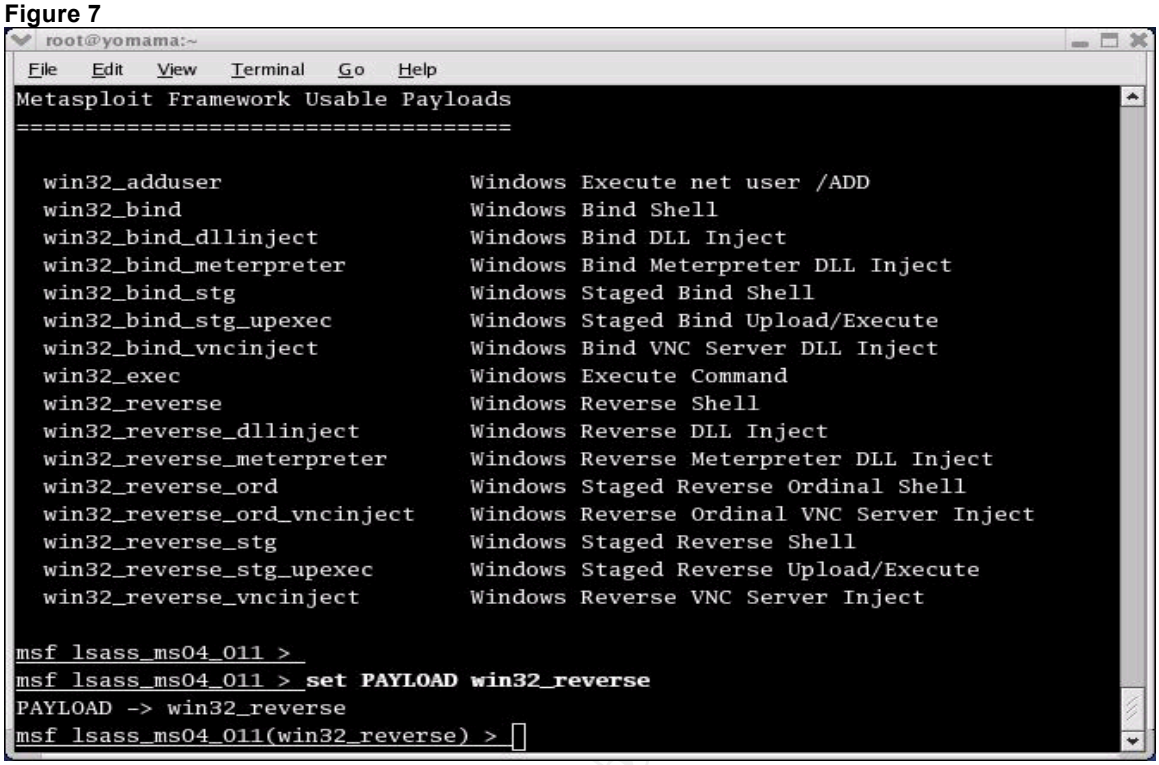
Figure 6



The ‘win32_reverse’ payload will be used to illustrate that the exploit will allow a remote shell with system privileges (full access). The ‘info’ command can be used if at any time the VPT might have a question of what an exploit or payload does, or would require as arguments. The ‘show advanced’ will also show additional information regarding a plugin that ‘show’ may not. For example ‘info win32_reverse’ would tell the VPT that the payload will connect back to the attacker and spawn a shell and list available options. That word ‘attacker’ is a strong word nowadays, but in reality, that is what the VPT is doing.

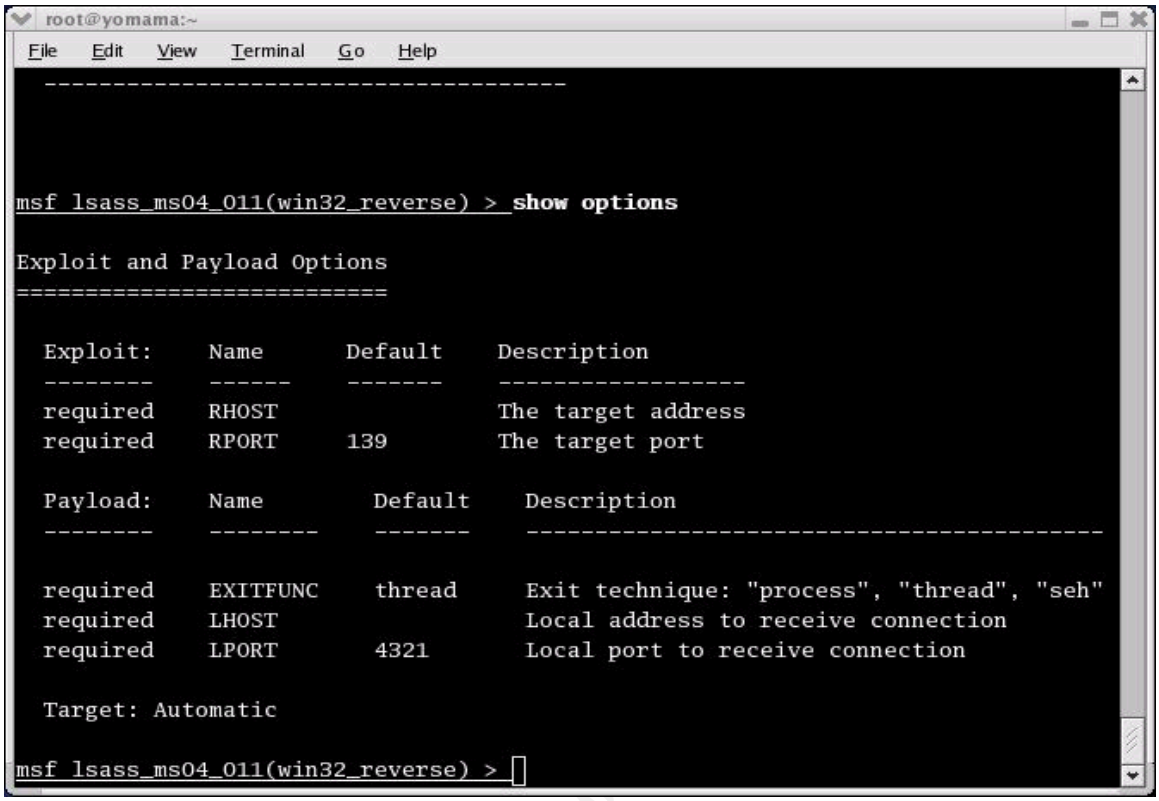
To utilize the ‘win32_reverse’ payload, the ‘set’ command is entered followed by the PAYLOAD keyword and win32_reverse as option, ‘set PAYLOAD win32_reverse’ (Figure 7). Notice that the command prompt now identifies the exploit and the payload.

Figure 7



The 'show options' command (Figure 8), will display the current required options loaded with the payload.

Figure 8



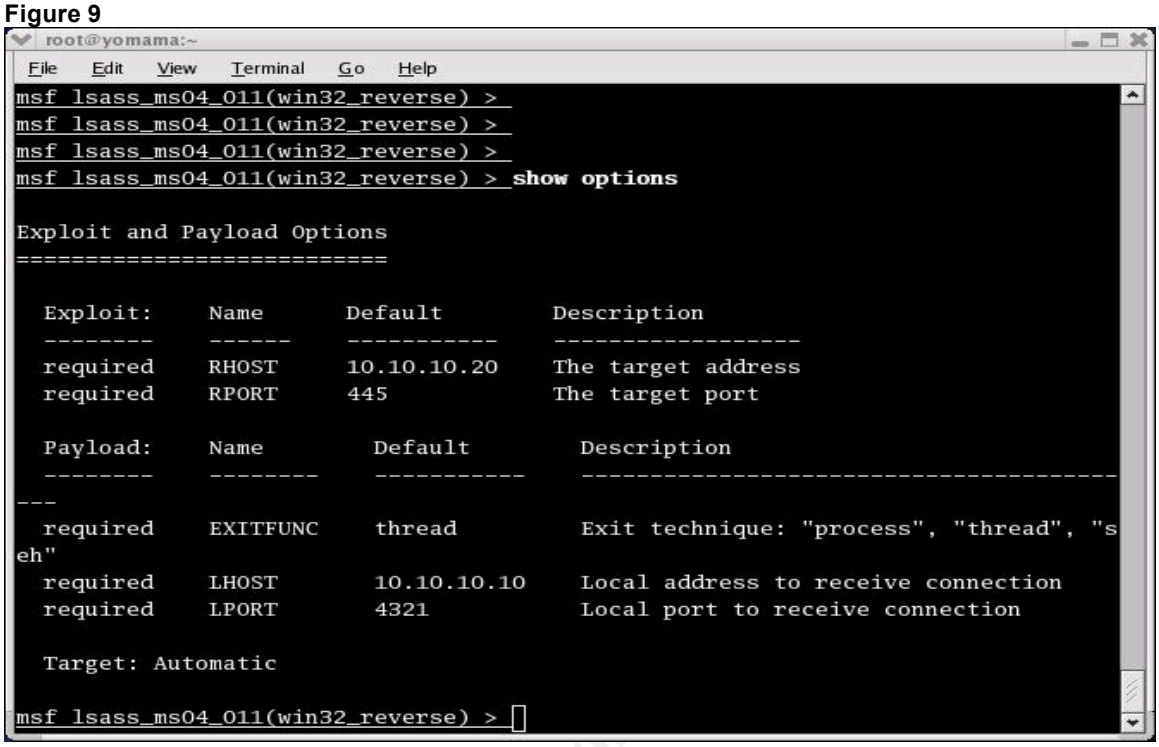
All of the options listed can be configured outside of the default. For instance, the Nessus scan indicated that the vulnerability was discovered over tcp/445. The exploit default configuration of the payload is over tcp/139. To change this to tcp/445, the 'set RPORT 445' command is entered. A confirmation will display to screen 'RPORT -> 445' indicating that the change was successful.

The exploit will run against the host at 10.10.10.20. To add this variable to the exploit, the 'set RHOST 10.10.10.20' command is entered and another confirmation is displayed confirming the remote host. The local host is also required to send the shell back to, 'set LHOST 10.10.10.10'. The LPORT can also be changed from the default 4321 to another port of the tester's choice as long as there is not another listener existing on that port.

Notice that the target is also set to Automatic. In this instance, the Target refers to the operating system that the exploit will attempt to run against. The 'show targets' command will display a list of possible targets and a selection can be made from there if the OS was known or the exploit was not working with Automatic set. Since it is assumed that we only know what we have gathered from the Nmap and Nessus scans, we will leave it as Automatic because there is not 100% certainty that it is a specific OS yet.

A quick run of the 'show options' command (Figure 9), will verify that all of the new settings are active for this session and ready for testing. Some modules allow for the use of advanced options. These can be viewed per module with the 'show options' command. This exploit will run with only the defined options.

Figure 9

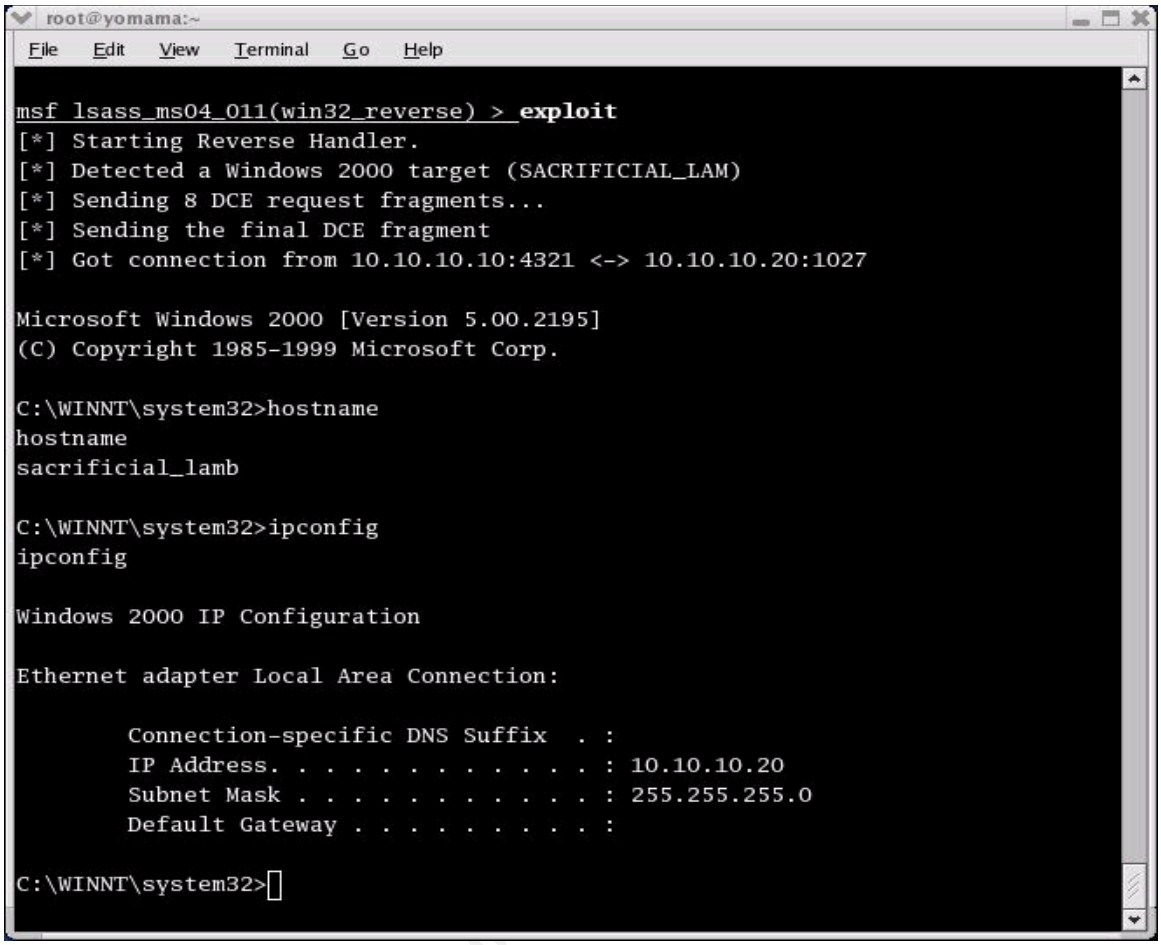


Some modules allow the use of a check command to check the configuration of the variables or options and remote system for the vulnerability. The lsass_ms04_011 does not have this functionality, so jumping right to the test is the next option.

Now the development team for the MP really tried to make this as easy as possible. The command to launch the exploit against the remote machine is 'exploit'. As soon as the command is entered the exploit begins the process of the exploitation, the payload is launched, and the VPT is presented with updates of where the process is in the exploitation process. We are presented with the 'C:\WINNT\system32>' prompt upon successful completion.

In order to verify this is the correct box, the 'hostname' and 'ipconfig' commands are entered and the results verify that this is the box in question (Figure 10). To leave the exploited session and perform a graceful teardown, the exit command is issued and the VPT is returned to the msfconsole. This should be all the information that is required when dealing with a system admin or user who adamantly disputes the results of any scan results indicating there are missing patches or other issues needing to be addressed.

Figure 10



Global settings and temporary environments

All settings up to this point are volatile or in other words, when you exit the console all changes made up to this point are lost and would need to be re-created the next time the msfconsole is run even if it is the exact same scan against the exact same box. This may be fine for the consultant who travels from site to site and only hits the same box as part of an annual or quarterly scan, but what about an organization that may have the framework set up on a box in their monitor segment?

You can define global properties by issuing the 'setg' command once the settings have been created in the temporary environment. Take for example the exploit shown above. If this machine were in a monitoring segment for an organization and its IP address was not going to change in the foreseeable future and this is the IP that the reverse shell would always come back to, then once the LHOST was defined, the 'setg' command would be invoked and a config file would then be written to $HOME/.msf/. Those settings are loaded by default when the msfconsole is started with that config file present in a VPT's profile. The settings are not set in stone and there is nothing special needing to be done to change these settings once the msfconsole has been started. All changes again will be temporary until either the 'setg' command is entered with

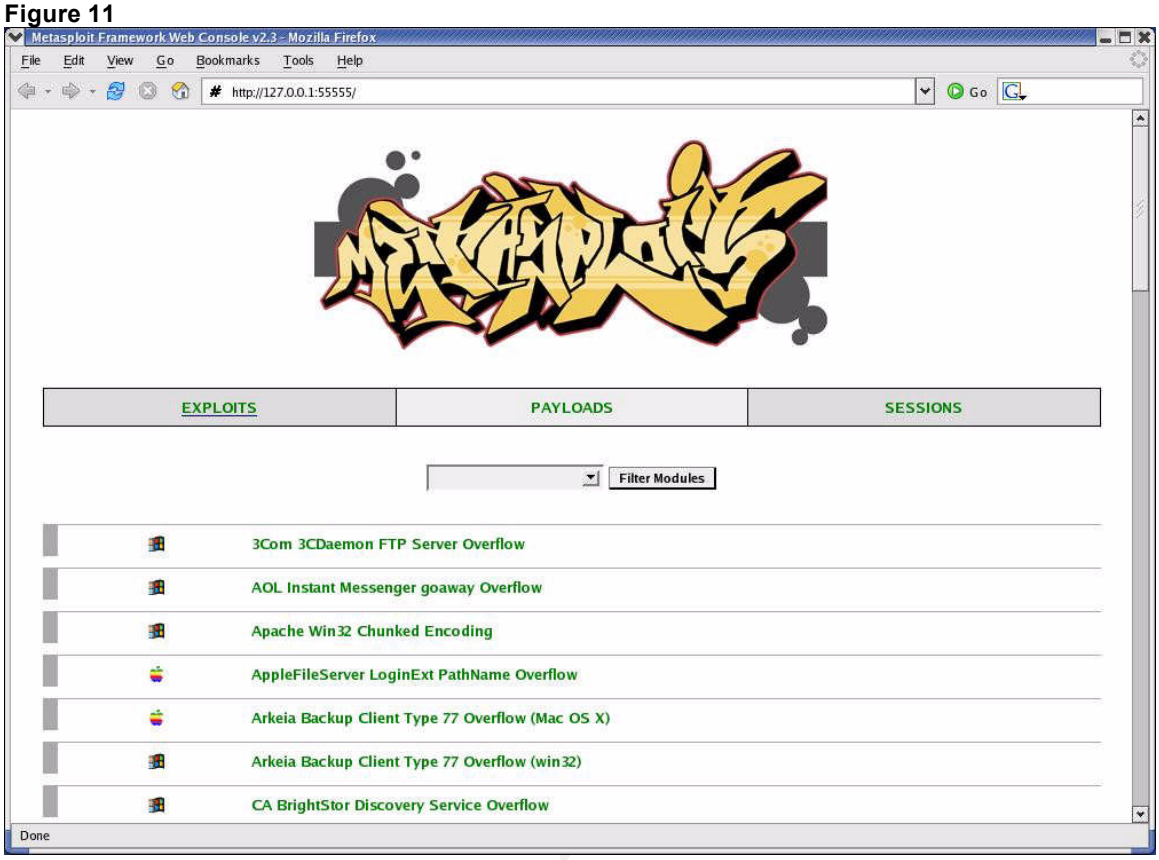
new variables or the ‘unsetg’ will return the setting to undefined.

Using msfweb for testing

There is another interface still in the process of being developed but is available as an option. The developers warn against the possible misconfiguration of this interface. Anyone who tests in this fashion should take heed as accidentally configuring the web based interface to the world could make a VPT’s life real miserable, real fast.

The ‘msfweb’ command with no options will open tcp/55555 on the local host. By connecting to this port via a browser, the VPT is presented with a listing of all available exploits and an icon depicting exploits (this is the default page), payloads, and session’s hyperlinks along with a list of available exploits and an icon depicting the OS that the exploit is designed for. (Figure 11)

Figure 11



The dropdown box presented on the exploits page will allow the VPT to select an exploit based on application, OS, and machine architecture.
By selecting the 'Microsoft LSASS MS04-011 Overflow' hyperlink, a target list is displayed. Selecting the target hyperlink (0-Automatic), will bring up the page for the selection of a payload all similar to the msfconsole selections. The web page now displays text boxes for configuration options. (Figure 12)

Exploiting the system is now just a click of the exploit button. If the exploit is configured to allow testing, this is also an option. When the exploit button is selected, the exploit is launched and the VPT is given the status via the browser. When the exploit has completed successfully, the VPT is either informed whether it has been a success or failure, or if there is additional functionality, such as the reverse shell in the msfconsole example, there may be another hyperlink for that feature. Figure 13 shows what the web based console control looks like. (Figure 13)

Figure 12

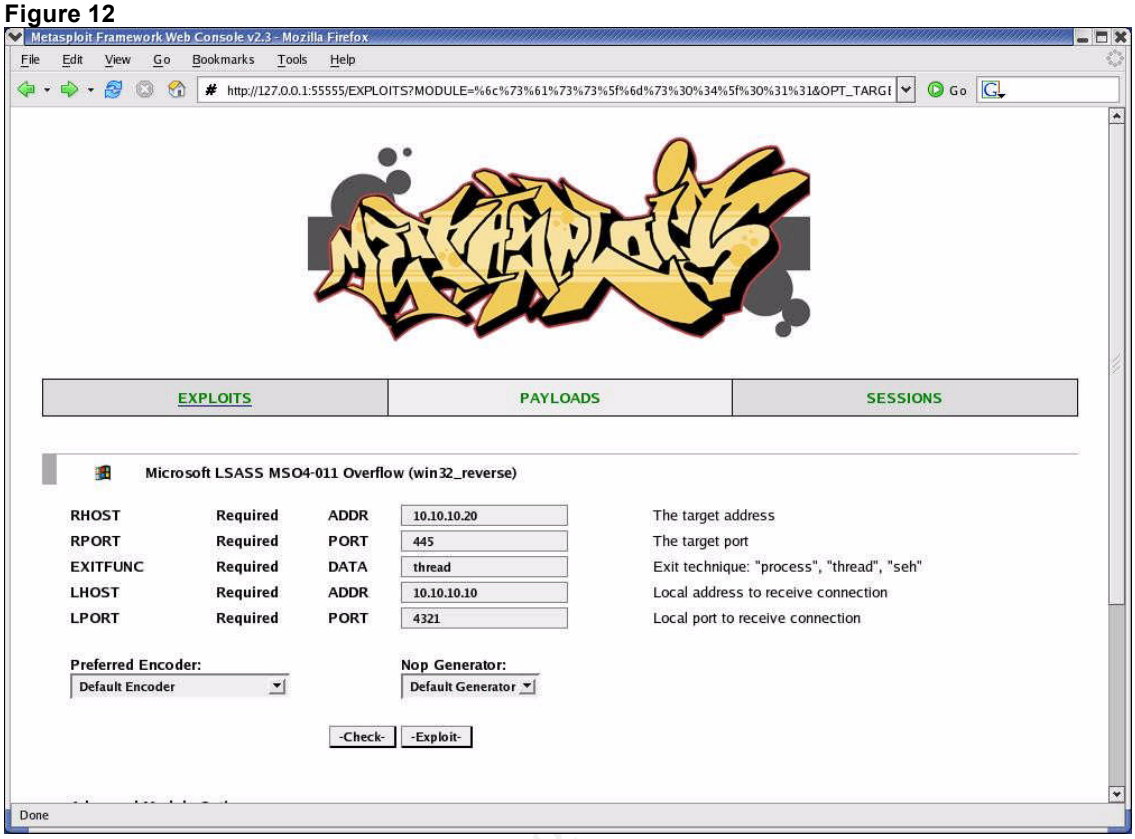
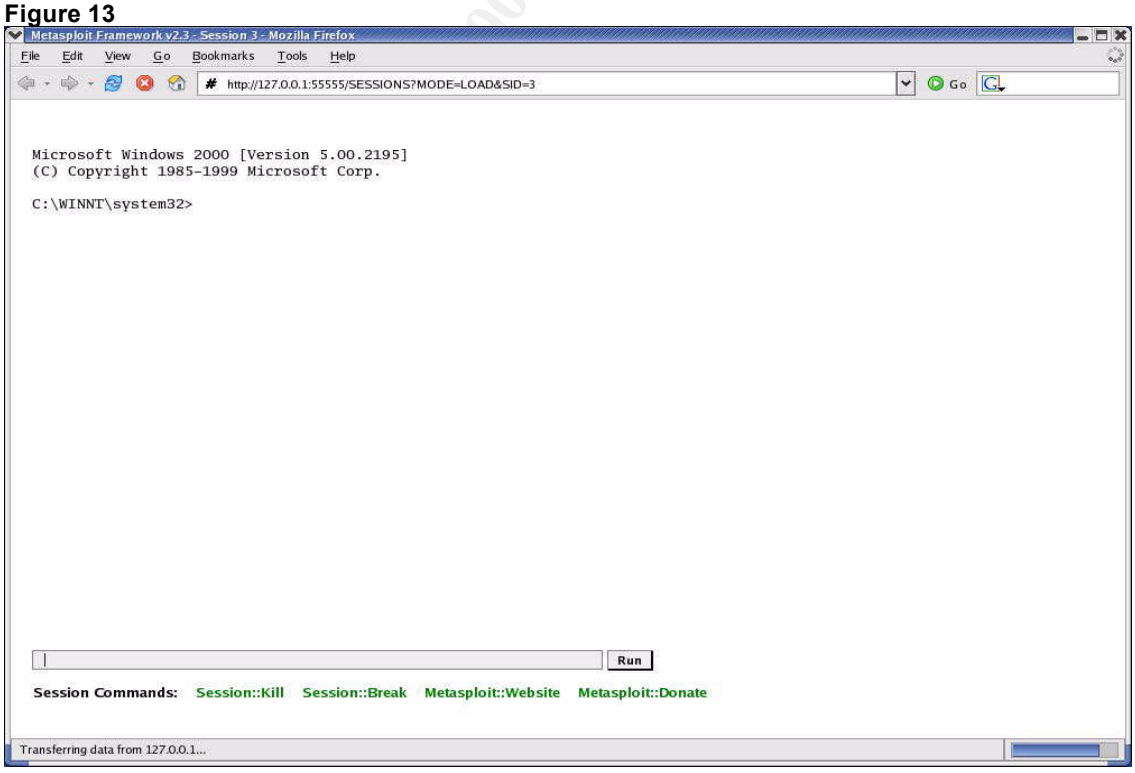


Figure 13



Advanced Features

The demonstrations above are very basic but at the same time very powerful examples of some of the functionality of the MP and in particular the Framework. For those who may be intrigued beyond the basics of what the Framework has to offer, the user guide that is available directly on the Metasploit site goes into explaining how reverse shell users can be added, .dll's can be injected into memory never being written to disk allowing further control using other methods of access or even additional avenues of exploitation, and what I believe to be the most promising is the meterpreter which "is an advanced multi-function payload that can be dynamically extended at run-time"[15]

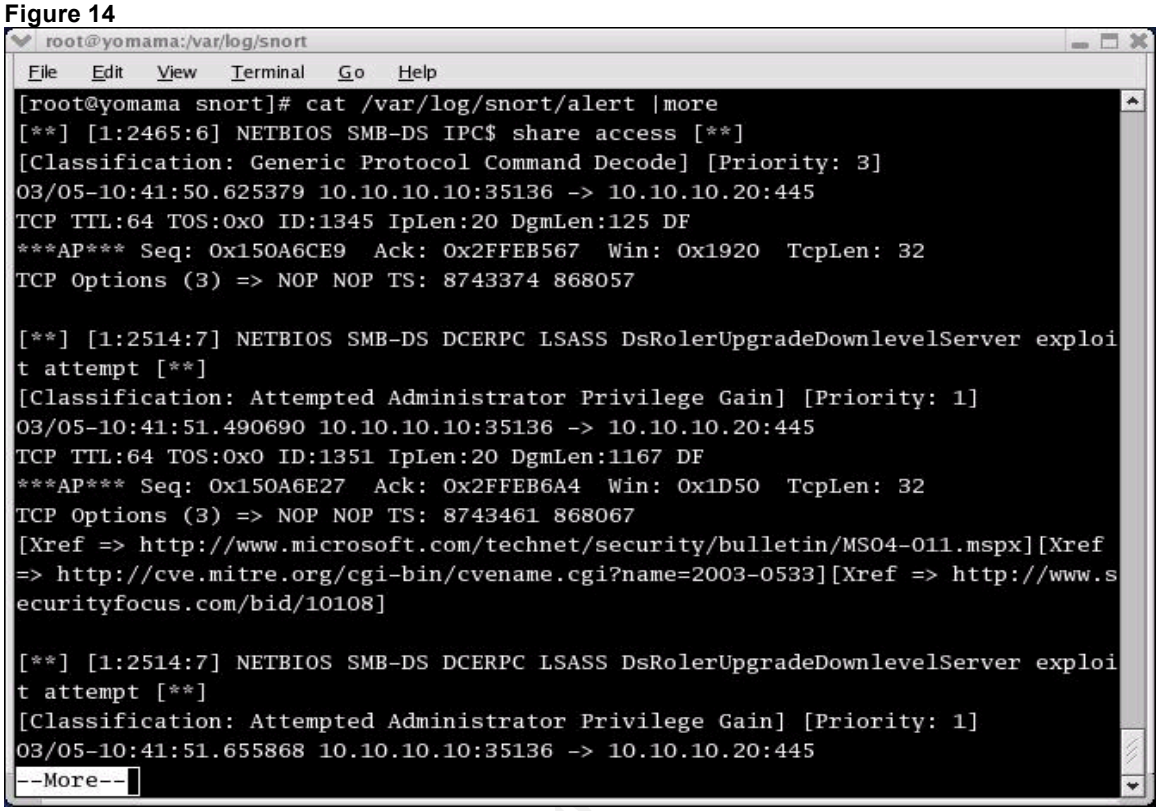
There may also be times when a manager/client/customer may not fully appreciate just what all of the command line interaction proves. For these people, presenting a GUI of the exploited box may be the best solution. The MP provides the ability of having a VNC server run on the exploited box and the VPT having a GUI session into that box.

When working with those in house applications, msfpescan allows the VPT to disassemble and examine those custom .dll and executables. By being able to search for jump equivalent instructions, pop+pop+return combinations, regex match and being able to show code at a specified virtual address is a much needed function with any penetration assessment tool..

Outside of using the framework as a penetration testing tool, it could also be utilized by an Intrusion Detection Engineer in the IDS signature development arena. All too often an IDS is set up with its default configurations and with the rash of false positives that are inherent upon that install, the person over that IDS may inadvertently reconfigure or flat out remove too many of the signatures or misconfigure variables introducing the possibility of false negatives. The Metasploit Framework will allow for the proper testing of an IDS and aid with the signature development. Figure 14 is the Snort output from the preceding exploit. (Figure 14) A signature could be created either by visiting the MP homepage and seeing if the latest vulnerability is available that the IDS is set to alert for, or crafting a quick and dirty exploit using the tools available through the MP.

But if a security practitioner, thinking here about an IDS engineer, is to use signatures that will catch all of the exploits used by the Metasploit Framework and leave it at that, a false sense of security could be just as bad as no IDS at all. The IDS could still fall prey to not only the custom crafted exploits, but also active development in the IDS evasion or countermeasure area of the exploitation process. There are exploit options available that work to blind the IDS from the impending attack. But exploit options are just one of many ideas being worked on right now. This fact in and of itself should make those same IDS engineers want to become students of the project and its capabilities.

Figure 14



Conclusion

Like many security tools, the MP has great potential with all of the features that have been presented. But again like many security tools there is the possibility of misuse. It is up to the individual end user to decide how it will be used. The bad guys already possess the tools capable of doing what is now possible with the MP. Security practitioners need to know how those same bad guys might attack and what is possible. Layered security is not just ACL's, firewalls, network segregation, IDS, etc. The most important layer in the security process is the human layer. What that human layer can bring to the table is every bit as important as the rule base on the firewall and being armed with the Metasploit Framework will only add to that human value.

References

1. Nessus Homepage. URL: http://nessus.org/

2. Internet Security Systems (ISS) Internet Scanner Homepage. URL: http://www.iss.net/

3. Security Administrator Tool for Analyzing Networks (SATAN). URL: http://www.fish.com/~zen/satan/satan.html

4. Security Auditor's Research Assistant (SARA). URL: http://www-arc.com/sara/

5. Nmap Homepage. URL: http://www.insecure.org/

6. eEye Digital Security Retina Homepage. URL: http://www.eeye.com/html/

7. Computer Associates (CA) eTrust Vulnerability Manager. URL: http://www3.ca.com/Solutions/Product.asp?ID=4707

8. GFI LANguard Network Security Scanner. URL: http://www.gfi.com/lannetscan/

9. Metasploit Project Homepage. URL: http://www.metasploit.com/index.html

10. Definition of opcode. URL: http://en.wikipedia.org/wiki/Opcode

11. Snort Homepage. URL: http://www.snort.org/

12. Cygwin Homepage. URL: http://www.cygwin.com/

13. Comprehensive Perl Archive Network (CPAN). URL: http://www.cpan.org/

14. Syngress Nessus Network Auditing. URL: http://www.syngress.com/catalog/?pid=2850

15. The Mererpreter documentation. URL: http://www.metasploit.com/projects/Framework/docs/userguide/node45.html

Metasploit Framework User Guide. URL: http://www.metasploit.com/projects/Framework/docs/userguide.pdf

The Metasploit Meterpreter documentation. URL:
http://www.metasploit.com/projects/Framework/docs/meterpreter.pdf

Security Focus writeup on the Metasploit Framework. URL:
http://www.securityfocus.com/infocus/1789
http://www.securityfocus.com/infocus/1790
http://www.securityfocus.com/infocus/1800