David Silva
Mohammed Raza
Walter Merfert
Adam Hernandez
CS 357-101

Metasploit

A vulnerability, in terms of network security, is an identified weakness in a controlled system. Vulnerability assessment is the first step in identifying these weaknesses. A thorough evaluation of how the system is seen, both from the untrusted network(internet), and the trusted network(intranet), should be performed on a regular basis in order to properly identify security risks.

There are basically two methods of performing vulnerability assessments, *outside looking in*, and *inside looking around*. Outside looking in vulnerability assessment involves the security administrator attempting to see their system from the attacker's viewpoint. Some of the things an attacker may be able to find are publicly routable IP addresses, firewall interfaces and devices on the DMZ or demilitarized zone, where internet accessible devices, such as DNS servers, HTTP servers, FTP servers, and SMTP servers may be located. On the other hand, an inside looking around approach involves assessing vulnerabilities within the internal system, for example, print servers, databases, and file servers. When using this type of assessment, the administrator's privileges are generally elevated to a higher level than most outsiders would be able to operate on, so it is important to take this into consideration when performing vulnerability assessment. Even today, network security for many organizations is designed to keep intruders from

penetrating their system from outside the network, while protecting the internal network's security is more limited.

There are many tools available to network security administrators that will allow vulnerability scanning on a system, however, a methodology should be set in place before such actions are taken. For example, the administrator should specify whether only certain components should be scrutinized, or whether it would be more efficient to assess the entire network as a whole. The type of tests performed will give the administrator a general overview of possible vulnerabilities, but false positives, finding vulnerabilities that do not exist, and false negatives, failure to find vulnerabilities that exist, which is particularly dangerous, must also be taken into consideration and further assessment may be required.

One vulnerability assessment tool that is commonly used is Nmap, which may be used to determine the layout of a network. It can give the administrator certain information about the network, including identifying hosts, open ports, detecting operating systems and the hardware characteristics of network devices. Another very popular assessment tool that may be used is Nessus, which is a vulnerability scanner developed by Tenable Network Security, and includes many features that are imperative to network security.

The information gathered during a vulnerability assessment can be used to perform penetration testing, or pentests, which involve safely attempting to exploit the vulnerabilities discovered on a system. These tests simulate potential attacks, and can be done both by the organization’s network security administrators or by an outside entity. There are also two types of pentests that can be implemented. A *white box* pentest takes the information from a vulnerability assessment and attempts to exploit the system based on those findings. Whereas

with a *black box* pentest, little to no knowledge of the target system is initially available to the penetration tester. Organizations that require a great deal of security, such as those that store sensitive information, should ideally use both of these techniques when testing their systems. Furthermore, these tests should be performed on a regular basis, especially if certain changes are made to the network, or if there have been changes made to the security mitigation techniques. Penetration testing is important for organizations because it helps to better identify the types security breaches that could result from possible exploits. Security risks may be prioritized depending on the organization's requirements.

Generally, a penetration test should involve reconnaissance to gather information about a system, and then utilize a set of tools in order to exploit the system and gain access to sensitive data on the trusted network. Metasploit is a tool that can perform such penetration testing, and the Metasploit framework allows administrators to create their own penetration testing modules, depending on their organization's security needs. This will allow the organization to allocate security resources and risk mitigation techniques more efficiently. Metasploit is also constantly updated, adding more information about the types of exploits different operating systems are vulnerable to, and the types of payloads that may be used. Backtrack is a distribution of Linux that includes many penetration testing tools, including Metasploit. More information about Metasploit is included later in this article.

An important thing to note about penetration testing tools, like Metasploit, is that they are available to both those protecting systems, as well as to those who wish to maliciously exploit them. Thus, making frequent assessment and tests more important for security. However, many of these tools have proprietary versions which allow more comprehensive and efficient

penetration testing for organizations that require it. Metasploit's proprietary versions may be seen as being quite expensive for the average user to acquire. Mitigation techniques should be used to properly secure controlled systems.

Security risk mitigation is the approach that attempts to reduce the impact caused by the exploitation of a vulnerability through planning and preparation. After a vulnerability has been exploited using penetration testing, the next step is to deal with the resulting impact on the system, or prepare for such an attack. Optimally, an organization should have some sort of Intrusion Detection System(IDS) in place, which will prevent and/or detect exploits before they become a problem, and log information for documentation and further analysis.

There are several types of IDS that can be used on a system. A Network-based IDS, which may use a monitoring port or SPAN port, is capable of viewing all of the traffic that moves through a device. A Host-based IDS, also known as a System Integrity Verifier, monitors important system files and detects when an attacker creates, modifies, or deletes these files. And an Application-based IDS will examine applications for abnormal events. These should all be updated on a regular basis, as new exploits are constantly discovered. Response to detected intrusions should be applied based on the organization's needs and the priority of the threat. Once again, false positives and false negatives may be an issue with IDS.

Another mitigation technique is to use anti-virus software, which will monitor a system and capture attempted exploits of files or email. This should be updated to ensure the most comprehensive list of known virus signatures is obtained and monitored. Firewalls will prevent certain types of information from moving between the internet and the trusted network. Firewalls use packet filtering to examine the header information of packets sent and received over the

network. Rules should be set for the firewall by system administrators for proper functionality. A combination of these techniques may the best approach for securing a network.

Metasploit was developed by H.D. Moore in 2003 during his spare time as a penetration tester. He originally programmed Metasploit in Perl, but later the Metasploit framework was completely rewritten in Ruby programming language. On October 21, 2009 the Metasploit project announced that it had been acquired by rapid7, a security company that provides unified vulnerability management solutions.

Metasploit framework is one of the most useful auditing tools freely available to security professionals today. From a wide array of commercial grade exploits and an extensive exploit development environment, all the way to network gathering tools and web vulnerability plugins. The Metasploit Framework is not just a collection of exploits, it is an infrastructure that you can build upon and utilize for your custom needs. This allows you to concentrate on your unique environment, and not have to reinvent the wheel. There are numerous interfaces for the Metasploit Framework, and among them the most popular interface is the msfconsole. It provides an all-in-one centralized console and allows you efficient access to virtually all of the options available in the Metasploit Framework.

There are three different types of payload modules in Metasploit: Singles, Stagers, and Stages. These different types allow for a great deal of versatility and can be useful across numerous types of scenarios. Singles are payloads that are self-contained and completely standalone. A single payload can be something as simple as adding a user to the target system or running calc.exe. Stagers setup a network connection between the attacker and victim and are designed to be small and reliable. It is difficult to always do both of these well so the result is

multiple similar stagers. Metasploit will use the best one when it can, and fall back to a less-preferred one when necessary. Stages are payload components that are downloaded by Stagers modules. The various payload stages provide advanced features with no size limits such as Meterpreter, VNC Injection, and the iPhone “ipwn” Shell.

Metasploit also has plugins such as RiskRater, Metasploitable Vulnerable Machine, Mobilisafe, ScanNow for MySQL, UpnP Router Scan, and BrowserScan. These are all available free or for trial by rapid7. These tools are used across your organization to help identify and secure risks and weaknesses to better protect your organization. For example, recent research revealed that at least 40- 50 million devices are at risk due to security flaws in the UPnP protocol. These issues potentially expose millions of users to remote attacks that could result in the theft of sensitive information or further assaults on connected machines such as personal computers.

There are two types of exploits that are in the Metasploit framework, active exploits and passive exploits. Active exploits will target a specific host device and will continuously run until the process is complete. Once the process is complete and there is nothing left for it to do the active exploit will then exit. If the exploit is a brute force module, the exploit will end if the user opens a shell or if an error occurs. An active exploit has the ability to be pushed from the foreground into the background by using the command “-j” in the command line.

Passive exploits sit in the background and wait for an incoming host to connect and then exploit the incoming hosts as they appear. Passive exploits normally target web browsers, ftp clients, and any other program that have a connection when that starts up once you open them. Passive exploits differ from active exploits because when a shell opens for a passive exploit it

does not force the exploit to close or cancel the connection. In order to have the passive exploit interact with the shell you need to enter "-l" into the command line.

A Metasploit virus can be one way of maliciously attacking someone's computer, but it can be detected by the anti-virus program, if they have one. However, if they are running an anti-virus program, it is not foolproof. There are ways for a virus to get around the anti-virus program if the virus is programmed properly. The best way to try and get around the anti-virus program is understand how the anti-virus program works. The first thing that the anti-virus should check for is a signature on the executable file template. If the anti-virus picks up a signature from the default template it will be automatically flagged for removal even if the program has no malicious code written in it. A way to get around the anti-virus from flagging your executable file because of a signature is to create a custom file.

The second technique the anti-virus uses is called the sandbox, at this point it will run your executable file for a short time and closely analyze what it is doing. If it determines that the shellcode that is running at this time is malicious, it will be flagged for removal. Some things that it looks for is if the executable is allocating a RWX memory block, or if your executable is trying to establish a reverse connection. A way to get around this part of the anti-virus detection is to have your executable file run a loop or non-malicious code during this time until the sandbox phase can time out.

The third level of detection that the anti-virus software will have is modules that are constantly monitoring and checking network traffic, web, email, and much more. This means if the connection that you are using to transfer for Dynamic-Link Library(DLL) files is not encrypted, then when you start transferring the second stage DLL, it will get detected by the

anti-virus program regardless. One way to try and get around this is to use reverse or bind HTTPS. There are some other ways of getting around the anti-virus, but they require you to increase the size of the executable file, which if increased by too much, can be a red flag.

The conficker worm uses the metasploit payload to spread through the computer. The exploit module in Metasploit also will provide the "smb_fingerprint()" function which detects and provides the attacker with the windows version, service pack information, and the language of the current operating system. By the attacker getting this information, they can cause a lot more damage. By using the metasploit module as the starting point, the programmer only needs to complement functions for automatic download and spreading.

A Trojan using the Metasploit payload can be sent to anyone using any type of file. Once that file gets opened, the attacker (if the server is up and listening) gets access to your computer through the predetermined port that they setup. The file can be as small and simple as Minesweeper for Windows. For instance, you receive a compressed .zip file with two or more files depending on how well you want to conceal the most important files, DLLs. Embedded into the code for the game is a function to call up the DLL file and once the DLL file is run, your computer is at risk while you sit there playing the game you have just downloaded.

As with all things that deal with technology, or pretty much anything for that matter, there are two sides to the same coin. Metasploit is a very powerful tool for both network administrators, and more specifically penetration testers, as well as hackers. The difference is in how that tool is used and how it can either take advantage of a system or let the user know if their system is vulnerable to a certain type of attack; more specifically, the Windows Metafile exploit.

So what exactly is a Windows Metafile? And what does it do? A Windows Metafile (WMF) is a file format for storing images. Some may know that there are many different types of image formats like JPG and PNG. However, those formats can be divided into two general categories, Bitmaps (BMP) and Vector images. Most photos on the internet are bitmap images, which are tiny pixels that when combined and viewed far away enough, will form an image. Vectors on the other hand, while made up of pixels from the monitor you are looking at, do not lose detail when zoomed in. WMF files attempt to combine elements from both worlds to allow the user to have both bitmap images and vector images in the same file using a set of function calls to the Windows Graphics Device Interface or GDI for short.

While made with the best of intentions, hackers caught on to the fact that some GDI functions accept pointers to callback functions for error handling. By including those functions within the WMF image, they are able to include any kind executable code so that when the file is opened, the code gets executed and the host computer becomes infected.

The problem with Metafiles however is that it can affect any version of the Windows operating system and the way it propagates. Let's say you are on the internet looking at photos and one of those photos is a WMF image and your computer is set to automatically preview that image file extension. If that image has malicious code attached to it then you will be infected. The worst part is that it will just seem like the image failed to load. Previewing or viewing the file in any such way will guarantee infection. You can view it through email or open it in Windows explorer and the result will be the same.

So how can you protect yourself against something that you cannot easily notice? There are quite a few things actually. Downloading and installing the official patch for your system off

of the Microsoft website is a quick fix for those not technically inclined. You can also use a workaround which requires you to disable shimgvw.dll. Maybe you want to just minimize the risk of being affected by such an attack. This can be easily accomplished by doing things like setting the default WMF application to notepad or blocking all WMF files on the network by file-header filtering. A lot of other risk reduction techniques require the user to disable things like image loading for applications, unfortunately that will disable all the other image file formats as well.

Metasploit ties into this by being a tool for penetration testers and network administrators alike to test for such vulnerabilities. By utilizing Metasploit, they can test numerous types of different exploits against their system to see if it is vulnerable to that type of exploit or not, and if it is, then they can attempt to fix the hole in the systems security. Even though Metafile exploits are not strictly network specific exploits, they are peripheral exploits. That means that a Metafile will not directly attack the network itself (routers, switches, hubs, etc.), but it can, however, attack the server that holds all the information on the network and possibly shut it down which could disrupt network usability. So by using Metasploit to test for these types of vulnerabilities on the host computers on your network, you can greatly reduce the chances of it actually happening.

Bibliography

Whitman, Michael E., and Herbert J. Mattord. Principles of Information Security, Fourth Edition. Boston, MA: Course Technology, 2011.

Phatak, Prashant. "Top 10 Security Assessment Tools." LINUX For You. 02 Feb. 2012. 04 Dec. 2013, Top 10 Security Assessment Tools

"41.2. Vulnerability Assessment." 41.2. Vulnerability Assessment. 23 Jan. 2007. Red Hat

Enterprise Linux Deployment Guide. 04 Dec. 2013, 41.2. Vulnerability Assessment

"Penetration Testing Overview." Efficient Penetration Testing. Core Security. 04 Dec. 2013 Penetration Testing Overview

Bradley, Tony. "Metasploit Framework." About.com Internet / Network Security. 04 Dec. 2013 , Metasploit Framework: Walking The Thin Line Between A Tool And A Weapon

Metasploits Unleashed. n.d. 1 Dec. 2013 Metasploits Unleashed

"Techniques for Anti Virus evasion." Metasploit. 23 Oct. 2013. 05 Dec. 2013 Techniques for Anti Virus evasion

Chen, Xiao. "Conficker Worm using Metasploit payload to spread." McAfee Conficker Worm using Metasploit payload to spread Comments. 15 Jan. 2009. McAfee Labs. 05 Dec. 2013 Conficker Worm using Metasploit payload to spread

"(Trojan Horse 2: Lesson 1)." Trojan Horse 2: Lesson 1: How to create and bundle the metasploit msfpayload reverse_tcp. Computer Security Student. 05 Dec. 2013 Trojan Horse 2: Lesson 1