

## Book Review: Metasploit – The Penetration Tester's Guide

Review by J. Oquendo

"Metasploit – The Penetration Tester's Guide" by David Kennedy, Jim O'Gorman, Devon Kearns, and Mati Aharoni is perhaps the most enjoyable book I have come across regarding the uses and functionality of Metasploit. There were so many concepts it refreshed me on, many functions I didn't know existed and other functions I did not correctly understand even with my years of using Metasploit. Let's take an in-depth look into this stellar publication by No Starch Press.

Initially I skipped through the first chapter of the book, "The Absolute Basics of Penetration Testing." However, I went back to the chapter as I had already been in and out of reading the methodologies laid out by the Penetration Testing Execution Standard (PTES). This chapter actually made sense after the fact, since my approach was that of the technical one: Show me the meat of this book. Not everyone who uses Metasploit (and other tools like it) has a concise understanding of penetration testing, and many will assume that aiming Metasploit at an address constitutes a penetration test. The chapter is clear, summarized and offers much food for thought outside of Metasploit and into the realm of penetration testing.

After the break, look for a link to a free download of Chapter 8: "Exploitation Using Client-Side Attacks"

Discuss in Forums {mos\_smf\_discuss:Book Reviews}

[Click Here to Download Chapter 8: "Exploitation Using Client-Side Attacks"](#)

## Chapter 2 &ndash; Metasploit Basics

While one may have used Metasploit for some time, it is easy to forget how much can be accomplished with the tool. As with Chapter 1, I initially skipped this one but went back, and I am glad I did. Metasploit is written perfect enough to allow one to use tools installed on the system while inside of a Metasploit session. There are so many tools and those tools have so many options on their own that the best bet would be to dive right in and &ldquo;tab&rdquo; your way around.

The chapter introduces the reader to the different options available on the usage of Metasploit, its tools and its views. Previously I stated, &ldquo;tab your way around,&rdquo; and the statement was concerning the use of the tab key while in an msfconsole. This chapter shows you the various options available for using Metasploit outside of a simple console. For readers uncomfortable with command lines and terminals, they will likely breathe a sigh of relief.

## Chapter 3 &ndash; Intelligence Gathering

Intelligence gathering in this book expands outside of simply gathering intel under Metasploit. Because the author&rsquo;s used examples while writing, the reader can see firsthand the output of Metasploit commands along with the explanation of what is occurring. This was a nice touch and anyone who reads this book will certainly take away a lot more than solely the uses of Metasploit or other tools. A favored example of what I mean can be read on &ldquo;Advanced NMAP Scanning TCP Idle Scan.&rdquo; Rather than state the capabilities of the tool, the authors explained in detail what the scan is, what it does and how it does it. It is definitely worth reading this chapter a few times, and perhaps even keeping it around as a reference guide.

## Chapter 4 &ndash; Vulnerability Scanning

Having never been a fan of vulnerability scanning, I went back over this chapter in order to give a fair review. In order to understand where I am coming from, I would have to take up too much space explaining my history and knowledge of the security arena. With this said I will give you an analogy, so that you may understand my gripes when it comes to vulnerability scanning versus outright penetration testing.

As a building owner, you are fearful that someone is going to intrude on your premise causing some form of financial damage. You seek to determine the possible avenues of attack someone may attempt to perform in order to steal something of financial value. This individual is an assessor. He will walk around the perimeter and assess the potential that someone will cause you harm. In walking around the perimeter he will take notes of peculiarities and report them:

“You have a window in the back of your building away from visibility to most. Someone can break this window, get in your building and crack your safe.” And so goes the reporting. The reality of this reporting is that it is flawed. Everything is vulnerability like it or not.

In a real world, while someone may actually get into that window, what does it yield them? They’d have to know firsthand exactly what is valuable and where to find it. Because someone gets into a window means nothing. Perhaps there is a guard dog walking inside the perimeter or maybe cameras will alert a guard. There are too many variables to make the data from a vulnerability assessment worthwhile.

The difference in a penetration test and a vulnerability assessment is that in a penetration test, the assessor is going to find the flaw and exploit it. They will not give you data that is half-baked. It will be more to the tune of, “I got in the back window and discovered no one was watching me, so I snuck into room 101 and was able to get your family jewels,” or “I was able to get in the window but was unable to do anything, because a guard dog chased me out!” This information allows security managers to better formulate security financials. As opposed to spending say \$10k on a full blown camera system, they can spend say \$2,000.00 on another guard dog, placing security resources where it is proven to be necessary.

Because enterprises are connected to external resources, the rule of thumb should always be, “We are vulnerable to everything, so let us figure out the realities of these vulnerabilities to apply the proper fixes,” versus, “Well maybe&hellip; let’s spend money to figure it out, then more money to fix it.” I believe in an all-inclusive “clean it up in one shot” approach where the vulnerabilities and exploitation is done in parallel. With that out of the way, I tend to favor exploitation scanning versus vulnerability scanning.

## Chapter 5 – The Joy of Exploitation

This exploitation chapter offers those unfamiliar with Metasploit a detailed explanation of the parameters and settings of the Metasploit application. “You’ve discovered a target, determined whether or not something was potentially exploitable, and now you will determine whether or not the target is truly vulnerable by exploiting it.”

The chapter is rather short and is detailed, yet it is lacking. As a tester and sometimes author of security documents, I can understand why it was written this way. As an experienced professional, the chapter did not give much to learn from; however, for those unfamiliar with Metasploit, it will enable one to understand the different parameters available when exploiting a system.

## Chapter 6 – Meterpreter

Metasploit is a very flexible tool, and this chapter does a nice job of exposing the reader to the Meterpreter shell. Meterpreter is the post exploitation shell component of Metasploit. “You have a foot in the door now what?” Meterpreter allows the tester to perform a variety of commands across the different operating systems. Any system command you would perform physically sitting at a terminal can be done using meterpreter. It places you – via a terminal – right in front of the machine.

Chapter 6 covers enough topics to make the book extremely valuable. Because it covers many overlooked components, I have actually gone back and forth through the chapter as not only a refresher, but an "aha" learned something new" scope. It is well written, extremely descriptive and in my opinion, the best chapter of the book.

#### Chapter 7 – Avoiding Detection

I felt that this chapter had a misleading title; however, it is worth reading from not only a "penetration tester's" perspective, but it can also aide anyone in the forensics, incident response and network analysis realms of security. The chapter focuses on the elements of "covert" planting tactics used in penetration testing. Meaning, building one's own tainted binaries in order to bypass antivirus software. It is a short chapter; however, those who are involved in the forensics realm may find it useful in identifying whether or not a system was compromised using Metasploit. I am aware that forensics applications have been created to do just that such as the Metasploit Forensics Framework.

#### Chapter 8 – Exploitation Using Client-Side Attacks

Client-side attacks are highly underrated by far too many security "professionals." Client-sides I want to believe, are likely responsible for the vast majority of high level compromises. This is because of the nature of the attack. Why – as an attacker – would I want to beat down the door to a castle, deal with the moats, bridges, and guard towers, when I can simply ask the princess to open the gates for me?

With that tidbit out of the way, this chapter simply offers a brief summary of an attack (MS Aurora), a primer on file format exploiting, and a brief walk-through on debugging using Immunity Debugger. It must be difficult to write a book when there is so much to talk about, that I can understand why the basics were chosen. For anyone new to penetration testing, the chapter will definitely give you something to learn, but from a professional standpoint, I did not gain much from this chapter (This is not to say it is not worth reading).

#### Chapter 9 – Metasploit Auxiliary Modules

Auxiliary modules in Metasploit make using the framework interesting. Metasploit has an enormous amount of tools to use and the modules chapter gives the tester a hardcore look at some of the tools to use in correlation to testing as well as a primer on using some of them. Although this chapter is short, a lot of time can be spent on this chapter especially if you have Metasploit running while reading this chapter.

In my experience I have occasionally seen peers use certain tools that are readily available in Metasploit. While I see nothing wrong with diversity, there are times when "timing" is critical, and one won't have the option to spend time digging out tools. Also worth noticing is the fact that when exploiting a system, Metasploit gives you the ability to port some of these tools over in the shell. Why re-invent wheels.

## Chapter 10 &ndash; The Social-Engineer Toolkit

Having tinkered with The Social-Engineer Toolkit (SET) from time to time, I skipped this chapter. This is not to say it should be skipped, but when it comes to social engineering, I tend to create highly targeted methods on my own. SET, for those who are unfamiliar with it, is an interesting tool that combines Metasploit alongside some serious Python programming, which enables a tester to create some interesting attack vectors. I believe this is the longest chapter in the book and it covers a lot about SET. Seeing as how the author of the tool is also one of the authors of the book, I am sure it is well documented.

## Chapter 11 &ndash; Fast-Track

Like Chapter 10, I didn't read this chapter fully through as most of the testing I perform requires a lot of manual intervention. I have used Fast-Track once or twice but have always preferred avoiding too much automation in the hopes of not missing important events. Fast-Track as I recall contained more SQL attacks and other exploits, and, being the author also had a hand in creating this tool as well, I am sure there is a lot of worthwhile information.

## Chapter 12 &ndash; Karmetasploit

Karmetasploit is an interesting tool and the authors of the tool are well known in the industry. Dino Dai Zovi is probably one of the most down-to-earth security wizards in the game. Karmetasploit is a tool that allows a tester to create Fake APs and leverage those APs for all sorts of attacks.

The chapter is meticulously written and will instruct a tester on how to use and master Karmetasploit; however, in none of the penetration tests that I have performed have I ever had to nor felt inclined to use it. This is simply because of the testing environments and parameters in my SOWs (Scope of Work) that prevented its use or minimized it.

## Chapter 13 &ndash; Building Your Own Module

This chapter exposes the reader into the inner workings of Metasploit modules and guides the reader through building their own module. Metasploit's portability and use of programming languages and exploits, makes creating specific tool possible without having to outright learn an entire language from scratch. With that said, it can also make the testing a bit more complex if one is not well versed in systems and or networking. I state this, because I have made my own VoIP-based modules which have caused DoS attacks on my development systems.

This chapter is worth reading a couple of times to familiarize yourself with how Metasploit works not only from a &ldquo;build your own module&rdquo; perspective but also that of a troubleshooting scope as well. There will be times

when an exploit or module will falter and understanding the inner workings of Metasploit from this level will allow you to save yourself a lot of time.

## Chapter 14 &ndash; Creating Your Own Exploits

Sadly, I skimmed through this chapter, and it is not to say that it wasn't well presented or documented. It was a bit on the short side, and when it comes to discovering and or writing your own exploit, it's a bit more complicated than this chapter presented it to be. The chapter covers fuzzing, debugging, and SEH handling on a very small scale. This isn't a book on fuzzing, reverse engineering or exploit writing, so do not expect much. However, one can expect to gain a better understanding of their relation to Metasploit.

## Chapter 15 &ndash; Porting Exploits to the Metasploit Framework

Porting exploits is a bit more complex than this chapter presents; however, the authors are very keen to go through the motions. While I prefer a more dynamic approach &ndash; not keeping all of my eggs in one basket &ndash; there are times when it would be helpful to have things readily available. Exploits on their own are worthy of their own books, since there are tons of parameters that come into play, many more than the book explains. This does not mean that the chapter is not worth reading, it simply means if you believe you will snag up every exploit you can get a hold of and port it into Metasploit, you will be sadly mistaken.

This chapter lays out the groundwork, but I would advise that anyone taking this route (porting exploits into Metasploit) must understand enough programming (and enough about exploits), to make this chapter worthy of the extra focus. Again, not stating the chapter is not good, simply stating it is very basic.

## Chapter 16 &ndash; Meterpreter Scripting

Scripting is 'where it's at' for penetration testing in the sense that as a tester, if you are familiar with systems, you can accomplish a lot quickly. When it comes to testing, I always formulate a gameplan, strategy and test it using system-based scripts. They allow me to generate baselines on accomplishing my objectives. What I need to do, how I need to do it, what could be an alternative route to take and so forth.

In this chapter, you will become more familiar with post compromise commands and the automation of those commands as well as how to bring them all together. My only gripe about this chapter is that much of it (Meterpreter Mixins) is Windows specific.

## Chapter 17 &ndash; Simulated Penetration Test

I will be honest, I did not read this chapter. This is simply because of Murphy's Law and my experiences doing

testing where one has to be prepared for the unexpected and be able to improvise on the fly. Never have I been a fan of “shooting fish in a barrel” where the targets are automatically configured for a compromise. I feel that a tester needs to learn the ropes on their own outside of following a step-by-step walkthrough. A simulated test can never be real world.

## Put a Bow on This Bad Boy

Wrapping this up ;) Had enough yet? Metasploit – A Penetration Tester’s Guide is a great book and in my top 20 overall favorite books of the past few years. It is detailed, well written and has actually enabled me to recall things I had forgotten about Metasploit over the years. There are little nuances I have with not the book, but with the duration of the material, some chapters are too short. Understandably though, this is a book about Metasploit, not about building exploits, not about methodologies or specific systems.

Overall you will not be disappointed with any of the content. I may have skimmed through a few of the chapters, and that is not because they were bad. Some of the content needed further expansion, lest someone become confused and think that this book would make them the best 0day writer on the security scene. It is worth the money getting this book and my suggestion for a tester would be to skip Chapter 17, create their own labs and practice what one is learning via the book in a personal lab.

J. Oquendo is a Chief Security Architect in a New England Managed Security Services Provider, and Trainer at the Cyber Security Forum Initiative where he teaches Computer Network Attack, Computer Network Exploitation and Computer Network Defense courses. He currently possesses the GIAC Reverse Engineering Malware (GREM), Certified Penetration Tester (CPT), Certified Ethical Hacker (C|EH), Certified Hacking Forensics Investigator (CHFI), Offensive Certified Security Professional (OSCP), Real World Security Professional (RWSP), Stonegate Firewall Engineer (SGFE) and Stonegate Firewall Architect (SGFA) certifications.