
Review: Penetration Testing with BackTrack by Offensive Security Part 4

Ryan Linn continues his insiders look at Offensive Security's online training in Part 4 of this continuing review of 'Pentesting with BackTrack.' As a reminder, PWB is described by Offensive Security as, "An online course designed for network administrators and security professionals who need to get acquainted with the world of offensive security. This penetration testing course introduces the latest hacking tools and techniques, and includes remote live labs for exercising the material presented to the students. This course gives a solid understanding of the penetration testing process, and is equally important for those wanting to either defend or attack their network. The course can be taken from your home, as long as you have a modern computer with high speed internet."

Ryan brings it all together for you next month with a complete review of the course as well as the exam experience. Stay tuned.

del.icio.us

Discuss in Forums {mos_smf_discuss:Linn}

This is the final installment of the weekly Pentesting With BackTrack (PWB) review. This week we will cover the final sections of the course which deal with post-exploitation exercises, web application penetration testing, and the final extra credit challenges. Up until this point, the course has covered recon, enumeration, exploitation creation, exploit use, and tunneling in depth. Now that those topics have been well covered and understood, PWB focuses on what happens after you have obtained a foothold in the environment.

The first section of the fourth theme covers Password attacks. The section starts out with the theory behind password attacks and explains what is happening using a Python script. This example is a good background and also shows how to create a very basic password guesser for services that may not have one already designed. The code has good feedback built into it, so that you can know what is happening as the script is running.

After the foundation for password attacks have been laid down, more automated approaches are introduced. THC Hydra is used to do the same tasks that the Python script did. Hydra is used to brute force a few services in order to show how it is flexible and how different options are utilized. Also covered is how to use wordlists differently in order to get results. One example that I use a lot was the SNMP example which will allow you to guess SNMP strings with Hydra.

Next the class goes through dealing with wordlist creation using a variety of tools. When profiling an organization it is frequently useful to gather words that are unique to that company to use as passwords and tools are introduced to gather those words from web pages in order to turn them into usable wordlists. The best part of this section is that there are tools that are on the BackTrack distribution to facilitate the creation of wordlists, and I was previously unaware of the one used in the course. I was impressed with how

easy they made this topic seem. I have seen some of this presented before, and it was presented very simply here which I liked.

Next there is a discussion and demonstration of getting password hashes out of a Windows machine using fgdump and pwdump. The course explains how to get the tools on the machine, what to expect when you run them, and what portions are important. Once you have the hashes, PWB takes you through cracking them via John the Ripper, Rainbow Tables and even online with the Offensive Security online crack tool. The final part of the password attack section covers the theory and use of the smb_relay Metasploit module, a tool that can use the victim's credentials against them in order to drop Metasploit payloads on the victim machine. This stuff is all pretty basic, but critical to a course like this, especially if the student hasn't been exposed to these types of tasks. At this point, they need it before they can even think of moving into professional penetration testing.

Next is a section that dealt with passwords when a machine is offline. If you were able to boot a BackTrack CD in a machine to which you have physical access, this section walks you through dumping SAM information on non-Domain Controllers, dealing with the restore password for Active Directory boxes, managing Linux partitions, and even dealing with Cisco device passwords. Each section is short, but gives links to a lot of additional information.

The final section for videos is the Web Application Attack section. Three basic types of attacks are enumerated for this section: SQL Injection Attacks, Using Web Proxies, and Command Injection Attacks.

The first section, the SQL Injection Attacks, covers in-depth the theory behind SQL injections, detecting SQL injections in a web page, and then the details of how the SQL injection can be used for database enumeration and even shell access.

PWB starts out by showing a vulnerable web app and then digging down into the code to show where it is vulnerable. Once we understand what a SQL injection vulnerability is, we find out how to tell if a web page is vulnerable. While the tests it lists aren't numerous, it does a good job of explaining the process.

Once you have identified the vulnerability, the class walks through table enumeration and column enumeration by hand. I have seen tons of tutorials online that cover the tool-based approaches to dealing with these attacks, but knowing how these attacks are working by executing them by hand really adds to the understanding.

Once the enumeration has been handled, a stored procedure was used and xp_cmdshell are discussed. xp_cmdshell allows for command execution, so this part of web exploitation is particularly valuable. This portion of the class is pretty brief, but the highlights are touched to give a point of reference for additional understanding in the future.

The next section deals with web proxies. This section is very brief and discusses how local web proxies can be beneficial to capture and modify data being passed between the web browser and the web server.

This section uses the Tamper Data plugin to demonstrate parameter manipulation. I thought that this section was pretty brief, and I wished that it had been longer. I use the web proxies very frequently while doing web penetration tests, and I think they are probably at least as important as understanding SQL injection. The theory is all there, I just wished that they had gone more into the practice of using web proxies.

The final portion of the web section involved command injection vulnerabilities. These vulnerabilities exist when data input from a web application are passed in some form to the shell to be executed. The course looks at some code that might exist and how it is vulnerable to this type of attack. Once the basics have been covered the limitations and some examples of exploitation are covered. This was another short section, but this time I thought the length was appropriate.

The next sections exist in the manual only, not in video form. These are all areas that are extra topics and appear to be bonus topics for those interested. The first section is on Trojan Horses. This section covers a variety of types of trojans, how they work, and include information on how to get them to try on your own. This section was interesting, and I tried one or two of the trojans, but I didn't play around with them extensively as I got what I needed from the description. I don't see myself using these very often, but it's good to know they exist in case some situation arises where they may be useful.

The next section was on Windows Oddities including NTFS Alternate Data Streams (ADS) and dealing with the Windows Registry. The ADS section describes how to use ADS to hide files and applications on the NTFS file system. Once the files are hidden this section then walks through how to access files that have been hidden. The final portion of this section discusses how to hide values in the registry, specifically ensuring that an application which is in the Run section of the Windows registry can't be seen. While I don't expect to use the second half of this section frequently, the ADS is handy when you don't want a file to be seen.

The final educational section of the class is on Rootkits. This section focused first on the background behind rootkits with some good links to more information, and then finally looks at 2 different rootkits. Each rootkit has a description of how it works and why it is interesting. The exercise is constructed to use some of these rootkits to see what it looks like on the machine, which I recommend just so see what happens when one of them is installed. Afterwards you will want to revert your lab machine, but this section was fun anyway.

Once all the teaching is done, it's time to play. The final part of the class is to exercise all of the tools and techniques you have learned in order to achieve 7 different tasks. Each task consists of using the tools you have learned to exploit a machine or a service, and each task you complete can earn you points on the OSCP exam. I thought this section was crazy amounts of fun and got to strengthen my knowledge about each of the techniques from the class. Each box reinforces a different strategy, and the fact that you have to document each step gets you ready to take the OSCP exam where documentation is critical. I spent quite a bit of time on this section and loved every minute of it. I thought each challenge was laid out well, and the fact that you take these points with you to the OSCP sealed the deal.

This is the last of the weekly updates. After this I am off to take the OSCP exam and write the final review for the class. Come back to find out what I thought of the exam, how I did, and how I feel the course fit together as a whole. Thanks for reading!

Ryan Linn, CISSP, MCSE, GPEN - Ryan is currently an Information Security Engineer at SAS Institute. Employed in the computer industry since 1997, he has held positions ranging from web developer to Unix Systems Programmer at a large university to his current position in Information Security. Ryan has been responsible for working with large scale deployments of various flavors of *nix, high availability web and database clusters, as well as for application programming in high availability environments. In the past few years, Ryan has incorporated Windows security into his responsibilities, and is now part of the team responsible for information security globally in one of the largest privately held software companies in the world.