

Course Review: Cracking the Perimeter by Offensive Security

Cracking the Perimeter (CTP) is the latest course offered by the team at Offensive Security. The course teaches expert level penetration skills including advanced tactics in web exploitation, binary manipulation and exploitation, and networking attacks. Building on material in the earlier course, Pentesting with Backtrack (PWB - Read Review), this offering provides intermediate students with a learning platform that can be used to become advanced practitioners of certain exploit methodologies. This review will attempt to provide a high-level overview of the course and set expectations for students who may be considering it.

Divided into a registration puzzle, five sections, and an exam, the course provides a more in-depth view of common web application exploits, binary analysis and backdoors, anti-virus evasion, techniques for exploitation using memory concepts, exploit writing, and network exploitation techniques. The end-of-course practical exam assures that the student has a true understanding of the course material presented, allowing employers and other security professionals to rely on the certification as a testament of capability, not only authority.

del.icio.us

Discuss in Forums {mos_smf_discuss:Linn}

In order to be allowed to register, prospective students must solve the puzzle at the challenge site: <http://www.fc4.me/> . This is a fair test of existing knowledge and problem solving skills. Those who find this puzzle to be extremely challenging may not have a firm enough grasp on foundational topics to get the full benefits of this course. In those cases, students may wish to practice more with the concepts from earlier courses or allow for additional self-study time before attempting this advanced course.

As a sample of skills requirements, it is advisable for students to have the ability to craft basic exploits, use OllyDbg, be familiar with multiple types of attacks, and have a functional understanding of network and operating system concepts. In order to finish the class within the 30 days of lab time, students should also understand web vulnerabilities including Cross Site Scripting attacks (XSS), SQL injection, remote file inclusion, and all of the basic attack types covered in the earlier course, PWB. While not all of these are mandatory, understanding these techniques well enough to expand upon them is important to grasping the concepts for many of the types of attacks covered by this course.

Course material includes instructions for connecting to the Offensive Security lab environment, a manual, and videos. The easy to follow VPN instructions are designed for use with the Backtrack Linux distribution. Setup should take less than ten minutes. The manual covers the core course concepts and contains a series of exercises for the student to do with each chapter. The videos and the course manual are complimentary. Each exercise is demonstrated in the videos with additional information to assist self-study. The exercises are well thought-out and critical to success in the course.

The course begins with "The Web Application Angle," a study of advanced web exploitation. In two scenarios, the student learns how expansion upon the basic attacks that were learned in PWB can lead to more complex and interesting attacks. These exercises require the student to expand upon current knowledge in order to take web attacks to the next level. These sections walk the student through the process of moving beyond basic JavaScript and form vulnerabilities. The student will take classic web vulnerabilities through the entire exploit process and end with shells on systems.

Once the web exercises have been completed, the course kicks into high gear with "The Backdoor Angle," a section on backdooring Windows Portable Executable (PE) files. This section walks through the steps of manually adding backdoors to working Windows binaries and keeping them working. The exercises give the student a first taste of Assembly, and, combined with some additional OllyDbg tricks, they teach students how to redirect process execution to new code, launch backdoor shellcode, and make it transparent to the individual running the file. The exercise at the end of this section is fairly straightforward thanks to the excellent videos and documentation provided.

Building on a basic understanding of process execution redirection, "The Backdoor Angle" continues with anti-virus evasion. Obfuscating portions of the executable enables payloads to run without being quarantined on systems protected by anti-virus programs. This section provides a good foundation for students to understand the role of obfuscation in anti-virus evasion well enough to conduct further independent research into other methods and easily apply them. Here, the course is heavy on Assembly and OllyDbg, but the course designers do an excellent job explaining what each Assembly statement does and presents the data in a way that makes it easy to reuse these techniques in other projects.

The next section, aptly named "Advanced Exploitation Techniques" discusses modern-day problems and solutions common to exploit development on operating systems such as Microsoft's Vista and Windows 7 with small payload sizes and Address Space Layout Randomization (ASLR). Very technical in detail, these sections explore techniques to overcome these challenges including using egghunters to find portions of shellcode in other places in memory.

"The 0Day Angle" is more complex and will probably require a bit more research and care when the student attempts the exercises at the end of the chapter. Much of the information presented until this point must be put into practice. These binary studies are two of the most difficult scenarios the course covers. The section deals with discovering vulnerabilities, determining if they are exploitable, and then following the exploit development process through creating working exploits. Here, the course will challenge the student's understanding of how to use the Assembly covered by the course materials, their ability to follow along with processes through OllyDbg, and to understand what is actually happening within a process. It is advisable to achieve the ability to complete the exercises without the assistance of the manual.

The final section entitled "The Network Angle" covers exploitation of common oversights in router security that a penetration tester may see, and then utilizing those oversights to obtain further visibility and access on the network. By modifying configurations on routers and eventually hijacking routes using GRE, a penetration tester can allow traffic that wouldn't normally be visible in order to make further access possible. Some of the principles in this section can be applied directly to a network penetration testers toolkit. The exercises at the end of this chapter test these principles in the Offensive Security lab. This environment allows not only the ability to perfect the skills from the exercises, but also offers targets for development of more generic scripts that can be added to the penetration testers arsenal for future engagements.

The final stage is the OSCE Exam, a 48-hour foray into an exploration of what the student has really learned. The four challenges initially appear to be similar to previous course exercises. However, it becomes quickly apparent that they are really components of an effective filter to differentiate between what the student perceives they know, and knowledge which the student truly understands.

The exam is scheduled when the student sends an email stating the intent to take the exam. Once a start time is agreed upon with Offensive Security and the exam has been scheduled, a new VPN configuration file and challenge document will be sent. This allows access to the exam network and provides the objectives for the exam.

Each exam objective is assigned a point value, and a passing point value is assigned in the exam documentation. Each objective has a clearly stated definition for completion, and all work needs to be documented as the student progresses. At the end of the grueling 48-hour experience, the documentation is sent back to Offensive Security, and within 72 hours the student will know whether he or she has officially passed. The result shouldn't be a surprise in most cases, however. The student will already know if he or she has completed the objectives of the exam and amassed enough points to receive a passing value.

Overall, the exam is exceptionally challenging but a good representation of the material from the course. Thorough study of the course material will ensure few surprises in the exam content, but it should help students to realize the places where they are robotically following along with the course material and doing exercises without truly understanding the material. The time allotted is not an exaggerated amount for most.

Students should expect the exam process to happen without many problems. However, the team at Offensive Security has provided a strong support structure to resolve problems both with the course and with the exam. By registering for the course, the student qualifies for a forum account and access to the CTP forums for course support. For example, using a more recent version of Metasploit presents some additional challenges and doing so may require a student to work more. Students are encouraged to visit the forums to see if others are having similar problems, if they believe something isn't working the same way it appears in the walkthrough. For the exam, the support system consists of a number of options including email, IRC, and MSN Messenger. Offensive Security staff is quick to respond to exam issues, and may grant more time to complete the exam if issues with the connection or lab prevent progress on the exam.

Overall, the course was an excellent experience with superb support, documentation, and exercises. The material and the exam were challenging but presented information in a way that was easier to understand than many other materials available. There are plenty of tutorials on the Internet about exploit development, but the team at Offensive Security has done a great job bringing these concepts together into a cohesive program. The program begins with the information from the PWB course and evolves students from practitioners into adept warriors. Successful students are able to wield the tools that they have been given, but they are also able to construct new and forward-thinking tools which better enable penetration testers to demonstrate security scenarios which represent today's landscape.

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.