

## Review: SANS SEC542 - Web App Penetration Testing and Ethical Hacking

Applications are moving away from the desktop and onto the web. With technologies like AJAX and Flash and the popularity of Mash-Ups and social networks, web application penetration testing is becoming increasingly important. Pushes for penetration testing are being driven by compliance, regulation, and a desire to not end up on the evening news, so a quality web application penetration testing class has been long overdue. SANS has stepped up to the plate and re-released SEC542 Web App Penetration Testing and Ethical Hacking as a 6-day course with stronger hands-on exercises and culminating with a final day where students perform a penetration test on the classroom network. The original course was a 4-day version, but Kevin Johnson of InGuardians has updated and enhanced the content to contain many of the cutting-edge web application hacking techniques seen in the field today.

I recently had the opportunity to take the re-born SEC542 course in Orlando, Florida as part of the SANS 2009. SANS 2009 was one of the larger yearly conferences that SANS offers with quality evening talks after classes which offered additional content for no additional cost. Some of SANS higher profile members presented fresh content ranging from Josh Wright's talk on the risks associated with using personal wireless devices such as the Nike +iPod titled "Privacy Loss in a Pervasive Wireless World" to Ed Skoudis' talk on cutting-edge tricks and techniques in "Secrets of America's Top Pen Testers." The secondary benefit of the large conferences was the ability to network with instructors and peers. There were frequent opportunities to hang out and talk with SANS instructors and other students after hours, with impromptu events such as full-contact mini-golf, dinner and karaoke. It is commonly known that an event is what you want to make of it, and SANS 2009 came through in spades in providing an educationally rich environment. So if an attendee didn't take advantage of networking with those in the industry, then it certainly wasn't SANS fault.

[del.icio.us](http://del.icio.us)

Discuss in Forums {mos\_smf\_discuss:Linn}

Day One was a good overall review of the basics of HTML and JavaScript along with a good introduction to the foundations of what would be discussed for the rest of the week. In my class, there were some students who weren't trying to understand what was happening in the exercises but instead were content in just trying to type in the information to complete the exercises. This course has quality exercises, but you will only get out of them what you put into them, and the students who read the background and followed the exercise all the way through felt like they had learned quite a bit by the end. For those with experience, Day One was mostly a review with a couple of valuable gems, but a treasure trove of knowledge for the others just getting into pen testing. Either way, without this information, the rest of the week, entitled "The Attack Process" with each day incrementing in number, would be very challenging to say the least.

Day Two AKA Part I: The Attack Process was an example filled foray into reconnaissance. It began by discussing the steps of recon and discovering targets, then moved into a series of hands-on exercises dealing with everything from

discovering DNS records through web spidering and session analysis. A series of new tools were introduced throughout the day, each time discussing basic usage and then utilizing the tool to discuss one of the principles of recon. For example, Nikto was introduced during the software configuration section. After discussing some examples of webserver misconfigurations, Nikto was used against a sample web site to demonstrate how these issues can be discovered.

Day Two was very informative. A lot of information was covered in what seemed like a very short time, and each topic was re-enforced with an exercise. Many new tools were introduced, and a small amount of scripting was incorporated as well. There was a little bit of overlap initially with the SEC 560 class as far as information gathering and recon, but much of this information was fresh and all of it portable to outside the classroom.

The vulnerability discovery portion of the course began on Day Three, "Part 2: The Attack Process." Unfortunately for Day Three, Mr. Johnson was ill, so Teaching Assistant (TA) Seth Misener took over. SANS utilizes TAs not only for the larger courses with many hands-on exercises, but also as a way for SANS instructors to become familiar enough with new courses to eventually teach. The latter case was clearly apparent as Seth was energetic, knowledgeable and did a great job on short notice in the first run of the new version of the course. Because of its first run status, there were a few snags on this day, but on Day Four all of the issues were discussed and resolved. So this should not be an issue in future offerings.

Day Three began with a discussion of automated web application scanners such as Grendel-scan, w3af, and the tools on the SamuraiWTF Live Linux CD. With the introduction of each tool, there was an exercise to demonstrate the basic usage of the tool. The first portion of Day Three was a great introduction to vulnerability scanning as each tool was covered in respect to vulnerability analysis but did not dig deeply into exploitation methodologies yet. The exploit methodologies and techniques using these tools come later in the week.

The second part of Day Three was on manual verification techniques. The emphasis of these sections seemed to focus more on the basics of different types of vulnerabilities rather than manual verification, although for each type of vulnerability there was a discussion of how to verify findings by hand. Like most of this course, the second half of the day was also very heavy on exercises. There were exercises for discovering information leakage vulnerabilities, performing username harvesting, performing command injection, and leveraging SQL injection (SQLi). The Day Three portion on vulnerability discovery finished with sections on cross-site scripting (XSS) and blind SQLi. There was an additional section on Python for penetration testers, however that section was moved to Day Four of our course. So in future courses it may be either on Day Three or Day Four.

Throughout Day Three there was new material which was re-enforced with exercises, and a lot of vulnerability types were discussed. With each type, the instructor discussed how to discover the vulnerabilities, and, while the course didn't probe too deeply into any specific vulnerability type, each topic was discussed in enough depth to give the students the resources to dig further to gain a complete mastery. The instructor was very knowledgeable about each type of vulnerability and as questions came up, he was consistently able to give quality information and relate the responses to real world scenarios. Since the student makeup of each class may differ greatly from city-to-city, the depth at which the course goes may very well depend on the experience level of the students themselves.

Python for penetration testers started off Day Four, "Part 3: The Attack Process," and provided a great introduction to the language. Much like the Javascript section, there seemed to be two sets of individuals in class: the type that blindly followed and the type that tried to understand the code. Although just recently back from the dead, Mr. Johnson was nonetheless very patient when students had problems. With his easy-going demeanor, everyone in the class had working code by the end of the exercise.

The main content of Day Four focused on client-side vulnerabilities. Day Four had the fewest exercises of the five days of lectures, however there was a tremendous amount of material covered. Each of the concepts was covered in enough detail to understand the concepts, but because of the sheer quantity, not every subject was covered in an extremely in-depth manner. A number of tools were introduced throughout the day as well, but, unlike some of the other days of the course, many of the applications of these programs did not have exercises. Although the trade-off is understandable, those tools without hands-on exercises were accompanied by screenshots in the provided courseware.

An early portion of Day Four was also devoted to AJAX. This section was full of quality information for detecting vulnerabilities in AJAX applications along with tools to assess AJAX problems. RatProxy was introduced and included an exercise that proved to be critical in tying together information presented earlier in the course. RatProxy meshed well with the Javascript section from Day One, and covered many of the common AJAX vulnerabilities that penetration testers cover while doing real-world assessments for Web 2.0 applications.

Next on our list was the second half of the Web 2.0 equation, web services. The section began with a discussion of web services, how they work, and frequent attack vectors. New tools to deal with web services were presented as well as strategies for utilizing these tools. Unfortunately for the web services section, there were no exercises to solidify the understanding of these tools. The explanations were quite good, and, by the end of the section, students were provided with a good starting point for assessing web services. I'm sure some of the other students appreciated the break from all of the exercises, but, being a glutton for some Kevin Fu, I wanted more.

The second half of Day Four concentrated on client-side code. These sections focused mainly on Flash and Java, and there were some strong exercises included in these sections to demonstrate how to analyze this code. Evaluating Flash and Java applets has been a hot topic lately as the technologies have increased in popularity and security professionals are focusing on them more. It was great to see some of these current issues incorporated into the course so quickly.

The section on Flash introduced the technologies that make up Flash and then discusses how to take them apart to evaluate the Flash files for security vulnerabilities. There were two tools with accompanying exercises that investigate different methods of pulling apart Flash applications into their components in order to review source. The Java section discussed how Java applets are handled and called within web pages. After the basics, the instructor discussed how to decompile Java class files into Java source in order to investigate what was going on within the application itself. There was a very nice exercise that involved decompiling class files to review the Java source and find vulnerabilities.

Day Four was scheduled to have PHP as the final module, however, like the Python section, it was pushed to the next day. It was unclear if this will be the standard for the class, but it did provide some extra time to work through the questions without feeling the pressure of finishing the exercises before having to leave the classroom.

Overall, Day Four contained a lot of cutting-edge material, even though the first half of the day was light on exercises. This can probably be attributed to the fact that all of the exercises are self-contained on the student's computer until the last day of class, so it may have been impractical and difficult to set up these types of services. The exercises in the second half of the day were good and provided a basis for individuals who are comfortable looking through code to find vulnerabilities. Those that weren't as comfortable looking through code should have been able to use these examples to get started, and even find some basic vulnerabilities after the course's discussion of potential attack vectors.

The day that everyone was looking forward to was Day Five, unofficially named the exploitation day. It began with the PHP chapter from the previous day's book. The PHP section was a good, quick introduction to PHP with the final result utilizing aspects from the Javascript section and creating code that would be very useful outside the class. The programming sections were a bit difficult to sit through because of the variety of skill levels in the class, but the instructor helped guide those who needed it to the final product. The idea of the exercise was great, and the fact that there were intermediate steps and checkpoints helped the exercise feel like an actual development process. After the PHP section, the day jumped right into exploit techniques.

We began with a discussion of authentication bypass, and an exercise demonstrating its principles. The rest of the morning through lunch consisted of SQLi discussions with exercises demonstrating different types of SQLi. The SQLi section was good, discussing many different methods as well as how to leverage SQLi in other attacks. There was also good time spent on blind SQLi, determining vulnerabilities and eventually exploiting them. There were a variety of tools that were discussed and utilized in the examples.

The second half of Day Five, "Part 4: The Attack Process," focused primarily on Cross Site Scripting (XSS) and session flaws. There were examples for both persistent and reflective XSS as well as examples using the Browser Exploitation Framework (BeEF) to leverage sites vulnerable to XSS. The BeEF example was great and, as client-side exploitation is one of the commonly discussed topics today, this was a very relevant exercise and discussion. The day finished out with a discussion of session flaws and determining if a site was vulnerable. The instructor covered how to exploit the site, although there were no examples. The final topic was on chaining attacks to get maximum benefit, and a teaser discussion of the upcoming exercises for Day Six.

Day Five was the day that I had looked forward to the most. There were some great examples and a lot of time was spent on two major discussion points, XSS and SQLi. While I felt like there were a lot of tools that could have helped in executing SQLi attacks, I wish that there had been more hands-on, manual exercises, so that the students would have had a better feel for how to exploit more complex applications. The tools that were used though were effective at detecting problems and exploiting them, and proved to be good preparation for what lied ahead on Day Six.

Day Six consisted of a brief introduction, some network setup, and then a sample web penetration test affectionately known as "Capture the Flag"(CTF). The scenario was setup well, and the requirements for the exercise matched well with the course content and mirrored many of the expectations from a real-world penetration test. The class was grouped into teams with each team determining the duties of its members to find sample data (flags) on a variety of servers and services. The nice part about this exercise was that the difficulty of the placement of the flags varied. The variation made it possible for everyone in the class to make some progress, while only a handful of individuals were able to capture all of the flags. There was a final debriefing that mirrored the presentation of findings to a client. There wasn't any new content presented in Day Six, but it definitely brought together everything from the previous days of the class.

If you have ever heard a phrase such as "BeEF injection through persistent XSS due to blind SQL injection," and thought that you were the victim of Mad Libs gone awry, then this class will help you sort out your attack methodologies and teach you the basics of web application penetration testing along the way. You will learn methodologies for approaching web app penetration testing and many tools to help you along the way. SANS has consistently provided quality instructors for the classes that I have taken, and even the day the primary instructor was ill, the alternate instructor picked up the reigns with very few hiccups. Kudos go to SANS for the quality of their farm system like process of finding and grooming instructors.

In a world where "software as a service" and "cloud computing" are buzzwords on everyone's lips, corporations are rapidly trying to secure computing environments that don't live as close to the firewall as they used to. Security professionals are reaching out to gain the knowledge to detect and resolve the security problems inherent in Web 2.0 deployments such as Flash, Java, PHP, SQL and more. This course infuses that security knowledge and the skills needed to get security professionals and developers alike familiar with those technologies and the techniques centered on web application penetration testing. You won't walk out of this class a web pen testing ninja, but you will gain a solid foundation in the tools and techniques used by the criminal element to properly assess the real risks in your own environment. If you are ready for a fast-paced knowledge transfer of quality techniques, cutting-edge tools, and real life experiences by qualified instructors, then SANS 542 Web App Pen Testing and Ethical Hacking should strongly be considered as the class to get you started.

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.