Review: SANS SEC709 Developing Exploits

Review by Zoher Anis, TerpSys

I had the opportunity to attend SANS 2009 in Orlando, once again as a facilitator. This time it was to tackle the toughest course SANS has to offer, SANS SEC709 Developing Exploits for Penetration Testers and Security Researchers, currently their only 700-level course. As described on SANS web site:

"In this course, we bridge the gaps and take a step-by-step look at Linux and Windows operating systems and how exploitation truly works under the hood. This four-day course rapidly progresses through exploitation techniques used to attack stacks, heaps, and other memory segments on Linux and Windows. This is a fast-paced course that provides you with the skills to hit the ground running with vulnerability research."

I would like to begin by saying that the above description is very accurate and should be taken word-for-word. It is a very tough course and very fast-paced. It does require you to know intermediate level x86 assembly programming, basic level C and python to get the most out of the course. Here's a quick day-by-day account of my experiences.

del.icio.us

Discuss in Forums {mos_smf_discuss:/root}

Day 1

The first half of the day was spent to quickly lay out the basics of x86 processors, memory registers, as well as stack and dynamic memory management. This was the only session of the class with very little hands–on exercises. It also introduced the students to tools and debuggers that would be used during rest of the course to analyze the given binary and developing exploit code for each specific example.

The rest of the day was spent disassembling x86 binary code, identifying stack-based buffer overflows and ways of

attacking them. It included hands-on exercises to demonstrate the stack-based buffer overflow, ret2libc attacks, defeating stack protection (canary) and defeating ASLR on Linux.

Day 2

This day, as described by the author and instructor himself, was the most technical day of the course. We started off with a brief discussion on understanding format strings and discovered vulnerabilities within it. This was followed by hands-on exercises on taking control of a process using format string exploitation.

Exercises and discussions on abusing the unlink() function in Linux was proceeded by discussing ways of writing efficient shell code. The day drew to a close with a discussion on understanding symbol resolution with PE/COFF object file format, the differences between AT&T and Intel x86 Assembly code format, a run-down of Ollydbg, and a brief understanding of modern OS and memory protection in Windows.

Day 3

This was the 'Windows Day.' We started off by discovering a remote vulnerability (FTP Server) on a Windows System through fuzzing. Once the vulnerability was discovered, we used Ollydbg and fuzzer to locate the exact location of the vulnerability and learned how to take control of the process. Then we looked into the ways to bypass protections added to Windows XP SP2, XP SP3 and Vista.

Next up was Windows heap exploitation. We looked at methods to abuse the Process Environment Block (PEB) and other constructs to gain control of a process. We moved from there into browser-based exploitation and how to increase the chances of exploitation through heap spraying. Day 3 ended with a look at Windows shellcode and how it differs from Linux.

Day 4

This was 'CtF Day' done DEFCON-style with a live scoreboard. After setting up the class network the previous evening, we were off capturing flags for the rest of the day. This was frustrating and tough, yet a fun exercise nonetheless. You used all the knowledge acquired in the previous 3 days to get 10 flags starting from a very easy to a very difficult level and gaining points at each stage. This exercise actually tested your patience and level of details as you progressed from one stage to the next.

I should confess here that I did not do very well and for the better part of the day was staring at the bottom, due to the fact that I was missing a little detail. I think I will attribute that to the previous night's session with Don and the EH-Net crew (search EH-Net for pics). There were no hard feelings as fun was had by everyone, even at the expense of my performance the next day. Just keep in mind that at any event, SANS or otherwise, there is always a social component. I encourage all the readers to do both, as it's not often you get to hang with like-minded people.

Conclusions

Overall, I would say that I enjoyed the course very much after overcoming the initial shock of the technical level for which I was not prepared. It is a very technical course and if anyone is planning to attend, please read up on x86 assembly, C and python. It is very fast paced (all students in our class felt that way), cutting-edge, deals with newer kernel versions (2.6) and even Windows Vista / Windows Server 2008. I think this is a one-of-a-kind course not offered by any other institution. If so, then it's not to the extent that Steve takes it, and definitely not with current platforms. I would rank it as the most technical and up-to-date course that I have ever taken bar none.

Last but not least, we had Steve Sims teaching it. Not only is he the author of the course, but a great guy with a huge amount of hands-on knowledge. He makes everything look so easy that it's scary. Then again, with a course at this level, you wouldn't want anything less.

Zoher Anis is Senior Security Engineer with TerpSys, a consulting company in Rockville, MD. He holds many technical certifications, including GSEC, GCIH, GCFA, GPEN, OSCP, CISA and CISSP. Zoher has sixteen years of IT experience, and has specialized in Information Security for over six years. He has been involved with many real-world instances of Incident Handling and Response, Computer forensics and development of policies and guidelines. He holds a Bachelor's Degree in Electronics Engineering from Bombay University.