

Application Security: For Hackers and Developers

After attending AppSec, students often say stuff like:

"I haven't had my butt kicked like that since grad school. It was great!"
 Dr. Josh Pauli, Dakota State University

"You'll learn a ton. It helps to be ready with C and Assembly, but if you're not Jared will teach you what you need."
 Anonymous

"I'm not sure you could improve this two day seminar. The amount of material was more than we could cover at times, but I would rather be exposed and then go off on my own than to omit some of the material."
 Dr. George Hamer

"There was so much information that it was like drinking water through a fire hose... impossible to catch it all, but well presented."
 Shane Shellenbarger, Recent College Grad at ToorCon

Day 1	Title	Topic	Labs	Lecture Slides	OS/Tools required
8:30 - 10:00	Source Code Auditing 1	Remembering C Dangerous coding in C	Program Client/Server Audit tricky code	Intro Software Security.pptx C.pptx C_Audit.pptx	Mac, Windows, or BSD image
15 min break	Coffee				
10:15 - 12:00	Fuzzing 1, 2, 3	Fuzzing overview and basic file fuzzing Network based mutator vs. Fuzzing Frameworks Peach	File bit flipper Fuzz FTP server with GPF and Sully Pit file data definition	Fuzzing .pptx	Mac, Windows, or BSD image VDA fuzz, GPF, Sulley, and Peach tools
12:00 - 1:00	Lunch				
1:00 - 3:00	Reverse Engineering 1, 2	Using IDA pro to reverse code	Compile and compare Crack a simple and harder program	Using IDA Pro.pptx	Win & IDA Pro
15 min break	Coffee				
3:15 - 5:00	Source Code Auditing 2 Reverse Engineering3	C++ Structures and C++	Use Virtual Functions Audit bad C++ code Structure discovery RTTI reconstruction System Calls Flirt/Flare	C++.pptx Using IDA Pro.pptx	WIndows, Mac, or BSD image Win & IDA Pro
Home Work	Source Code Auditing 3	Web Apps	Input Sanitization XSS & SQL injection	Web Application Security and Penetration Testing.pptx	WebGoat

DESCRIPTION:

There are four technical skills required by security researchers, software quality assurance engineers, or developers concerned about security: **Source code auditing, fuzzing, reverse engineering, and exploitation**. All these skills and more are covered. C/C++ code has been plagued by security errors resulting from memory corruption for a long time. Problematic code is discussed and searched for in lectures and labs. Web auditing is covered using WebGoat. Fuzzing is a topic book author DeMott knows about well. Mutation file fuzzing and framework definition construction (Sulley and Peach) are just some of the lecture and lab topics. When it comes to reversing C/C++ (Java and others are briefly discussed) IDA pro is the tool of choice. Deep usage of this tool is covered in lecture and lab. Exploitation discussions and labs are the exciting final component. You'll enjoy exploiting BSD local programs to Vista browsers using the latest techniques.

Reverse Engineering

Students focus on learning to reverse compiled software written in C and C++, though half-compiled code is mentioned as well. The IDA pro tool is taught and used throughout. Calling conventions, C to assembly, indentifying and creating structures, RTTI reconstruction are covered. Students will also use IDA's more advanced features such as flirt/flare, scripting, and plug-in creation.

Source Code Auditing

Understanding how and when to audit source code is key for both developers and hackers. Students learn to zero in on the important components of each language. Automated tools are mentioned, but auditing source manually is the focus, since verifying results is a required skill even when using the most advanced tools. Spotting and fixing bugs is the focus.

Fuzzing

Fuzzing is a runtime method for weeding out bugs in software, with a growing footprint within security companies and research communities. Techniques such as dumb file fuzzing, all the way up to intelligent network protocol fuzzing will be covered. Students will write and use various fuzzers to find bugs.

Exploitation

Students will walk out of this class knowing how to find and exploit bugs in software. This is useful to both developers and hackers. The exploit component will teach each common bug type including: stack overflows, function pointer overwrites, heap overflows, off-by-ones, FSEs, return to libc, integer errors, uninitialized variable attacks, heap spraying, and more. Shellcode creation/pitfalls and other tips and tricks will all be rolled into the exciting, final component.

No hard prerequisites, but helpful if:

1. College Degree in a computer related discipline or equivalent work experience
2. If desired read "Introduction to Application Security":
http://www.vdalabs.com/tools/AppSec_Whitepaper.html
3. Programming (C/C++/.asm) and security experience will help, but you will still get a lot out of the course if you lack that, so no fears. All questions are good questions in my classes. We have a fun but instructive and *intense* learning experience. You won't walk away disappointed.

GOAL:

"By the end of this course, you will be able to: research and develop an exploit from scratch by auditing code or fuzzing an application, reverse engineering the issue, and developing an exploit for the vulnerability you discovered. This knowledge will help developers produce better code, and will help security researchers or malware analysts in their daily tasks."

Day 2	Title	Topic	Labs	Lecture Slides	OS/Tools required
8:30 - 10:00	Reverse Engineering 4, 5	Extending IDA	Decryption with IDC Plugin Creation Using an emulator to unpack UPX protected code	Using IDA Pro.pptx Windows shellcode info from Exploit1 used in malware lab	Win & IDA Pro
15 min break	Coffee				
10:15 - 12:00	Exploitation 1	Overflows, shellcode, debugging, and more	Easyd, sc, funcptr, start local	Exploitation1.pptx Using IDA Pro.pptx (sys calls section)	cygwin, FreeBSD, & Win
12:00 - 1:00	Lunch				
1:00 - 3:00	Exploitation 2	Writing Exploits in BSD and XP	Finish Local Threaded-server.exe Lab 6 (OffByOne) or Lab 7 (fse)	Exploitation1.pptx Exploitation2.pptx	BSD, Win/cygwin, IDA
15 min break	Coffee				
3:15 - 5:00	Exploitation 3 Exploitation 4 Exploitation 5	More Bug Hunting and Exploitation Heap Spraying Bypassing DEP/ASLR	Lab 8 (Ret2Libc) Lab 9 (IntErr) Exploit a Vulnerable ActiveX control Vista Protections	Exploitation3.pptx Exploitation4.pptx Windows 7 Shellcode Creation	IDA and BSD XP/IE7 Win/cygwin, IDA, .net framework Vista/IE7/8?
Home Work	Exploitation	Any unfinished exploit labs	Probably FSE and the Exploit 4/5 full implementations	There's also a lab in BSD named "final". Find/sploit bug in local pure-ftpd server	Hope you enjoyed and were challenged!

INSTRUCTOR:

Jared DeMott is a Principal Security Researcher for the Crucial Security business area at Harris Corporation and PhD candidate at Michigan State University. Crucial provides state-of-the-art technical engineering and security services to the most elite branches of the Federal Government's law enforcement and intelligence communities, engineering solutions to meet their demanding requirements. Mr. DeMott previously worked for the NSA and currently teaches computer security at university and professionally. He has spoken at security conferences such as Black Hat, Defcon, ToorCon, and Shakacon. This background provides an ideal blend of skills for teaching cutting edge security material, in a fun and instructive manner.

Students are required to provide a laptop for the course¹:
Install Ahead of Time

- Nearly all the work will be done in XP (you provide) and the BSD image (I provide)
 - Vista is not required but is referenced for the final exercise if you have it²
 - If you have Vista/7, you'll be ok for most of the exercises but will have additional pains
- Cygwin (include: vim, make, gcc, perl, python, netcat, ruby, man pages, ndisasm, and whatever else you like)
- VMware workstation/player for Windows or Fusion for the Mac
- Visual Studio (Express is fine if don't have full)
- WinDbg and Immunity Debugger
- Used only for Day 1 homework -- FireFox (optional plug-ins: Tamper Headers, Firebug, and Live headers)

¹ I have a Mac, and use fusion to include XP and the provided FreeBSD 6.3

² I do not provide an Image for XP or Vista.

Provided on Course DVD, and will be installed in class

- IDA pro 5.x (I have the 5.5 demo for install on DVD, can also get from hex-rays.com)³
- Python (From Sulley installer. pydbg works with 2.4 by default in this installer)
- 010 hex editor (trail available)
- Keep at least 1.5GB free HD space to install the course materials and FreeBSD VM

Course Material info

The course material will be provided to you at class check on day 1, normally as a DVD or thumb drive that you keep along with any printed material. As soon as you receive the course material extract⁴ and test the BSD image. There is a BSD survival guide in the AppSec_A-Z\Exploitation folder with the user and password (and more). All the material you need to do the BSD labs in already in the image so you shouldn't need to transfer any information to the image.

The course material is in 4 directories: SrcAudit, Fuzzing, Reversing, and Exploitation. In each directory you'll find a wealth of knowledge from documents, tools, labs, and lectures. There's so much we won't go over it all, but leave further study as bonus material⁵ to the student. Harris marked material cannot be directly reproduced or used for profit, but can be shared to internal co-workers within the organization that sponsored your seat in the course, if credit is noted.

There is a feedback form in the base directory that should be filled out on the final day if the conference does not provide a custom form for feedback. Any other comments can be sent directly to the instructor at jared.demott@harris.com.

SUGGESTED TEXTBOOKS:

Grey Hat Hacking: The Ethical Hacker's Handbook, 2nd Edition, Harris, Harper, Eagle, and Ness

Fuzzing for Software Security and Quality Assurance, by Takanen, DeMott, Miller

The Art of Software Security Assessment, by Mark Dowd, John McDonald, and Justin Schuh

The IDA Pro Book, by Chris Eagle

³ The demos timeout after 15 min. or so, but no worries you can keep restarting it

⁴ This is because unzipping can take some time

⁵ Not all the bonus material is complete