		Emory Computer Security Fall 2016	Assignments	Reading	Chapters / additional reading		
1					*Note: the chapter numbers may be updated during the semester*		
Week 1 Assembly	8/22/2016	-					
		Introduction and overview		CSAPP1, 2.2	https://picocf.com/docs/asmhandout.pdf		
2							
Week 2 Assembly	8/29/2016	**x86 assembly overview: data representation, basic commands, control flow**	**Lab 1: The Bomb (7%)**				
	8/31/2016	**x86 assembly overview: Loops, procedures/functions**		CSAPP 3.1-3.16	Machine-Level Representation of programs (ch3)		
	9/2/2016						
3							
Week 3 Assembly	9/5/2016	**LABOR DAY - NO CLASS**		CSAPP 3.7-3.16			
	9/6/2016		Homework 1 due, Tuesday night				
	9/7/2016	**x86 assembly overview: C, Advanced topics**	**Lab 2: Buflab (7%)**				
4							
Week 4 Buffer overflows	9/12/2016	**Basic buffer overflow**					
	9/14/2016	**Demystifying shellcodes**			Optional reading for buffer overflows: "Smashing the stack for fun and profit"		
	9/16/2016		Homework 2 due				
5							
Week 5 Buffer overflows	9/19/2016	**NOP sleds / Jump to register / (Return to libc)**	**Lab 3: Stacklab (8%)**				
	9/21/2016	**Return to libc, return oriented programming**					
6							
Week 6 Web security	9/26/2016	**ROP, Integer overflows**					
	9/28/2016	**Integer overflows (cont.) / Heap overflows and defenses**					
			Presentation topics out				
	9/30/2016		Homework 3 due				
7							
Week 7 Potpourri	10/3/2016	**Heap exploitation cont.**				X	
	10/5/2016	**Format string attacks**				X	
					http://www.scriptjunkie.us/2014/09/exploiting-ammyy-admin-developing-an-0day/		
8							
Week 8 Advanced overflows	10/10/2016	**FALL BREAK - NO CLASS**					
			Lab 4: Joylab (13%)				
	10/12/2016	**Format string attacks cont. / ASLR and new attacks**			https://users.ece.cmu.edu/~dbrumley/courses/18487-f13/docs/Muller_2008_ASLR%20Smack%20&%20Laugh%20Reference%20Seminar%20on%20Advanced%20Exploitation%20Technique.pdf		
	10/14/2016	**Choose topic for presentation.**					
9							
Week 9 Networking / Processes	10/17/2016	**Side channel attacks**					
	10/19/2016	**Web security / OWASP TOP 10**					
	10/21/2016						
10							
Week 10	10/24/2016	**Web security / OWASP TOP 10**					
			10/25: Homework 4 due				
	10/26/2016	**Network security**			http://seclab.stanford.edu/websec/framebusting/framebust.pdf		
					http://seclab.stanford.edu/websec/csrf/csrf.pdf		
	10/28/2016				https://crypto.stanford.edu/cs142/lectures/12-csrf.pdf		
11							
Week 11	10/31/2016	**Network security cont. / Wireless security**			http://www.skullsecurity.org/blog/2013/ropasaurusrex-a-primer-on-return-oriented-programming		
	11/2/2016	**MID TERM EXAM**	**Mid term (15%)**				
	11/4/2016						
12							
Week 12	11/7/2016	**Wireless security cont. / What happened to my code?**					
	11/9/2016	**Randomness, and the RSA algorithm**	**Lab 5: Syndis OWASP lab (7%)**			(X)	
	11/11/2016						
13							
Week 13	11/14/2016	**Lock picking and physical security (Prof. Avani Wildani)**				X	
	11/16/2016	**Ethics class - With power comes responsibility**	11/16 Homework 5 due			X	
			Optional lab: Sandbox escape (+5%)				
	11/18/2016		Presentation script due				
14							
Week 14	11/21/2016	**Secure design**	**CS584: Blackjack (5%)**				
	11/23/2016	**HAPPY THANKSGIVING! (NO CLASS)** (Optional lecture online: Cryptography: How Turing defeated the Enigma code)			Optional: Read the chapter on Enigma from "The Code Book" by Simon Singh		
15							
Week 15	11/28/2016	**Student presentations (1 of 2)**	**Presentations (15%)**				
	11/30/2016	**Student presentations (2 of 2)**					
	12/2/2016						
16							
Week 16	12/5/2016	**The hacker underground scene / Interview (TBD)**	**Optional labs due**				
	12/7/2016	**EXAM PERIOD BEGINS**	**Final: 20%**				
	12/9/2016						

	Emory Computer Security Fall 2016	Assignments	Reading	Chapters / additional reading

Emory Computer Security Fall 2016	Assignments	Reading	Chapters / additional reading

Emory Computer Security Fall 2016	Assignments	Reading	Chapters / additional reading

	Emory Computer Security Fall 2016	Assignments	Reading	Chapters / additional reading

	Emory Computer Security Fall 2016	Assignments	Reading	Chapters / additional reading

Emory Computer Security Fall 2016	Assignments	Reading	Chapters / additional reading

Emory Computer Security Fall 2016	Assignments	Reading	Chapters / additional reading

	Emory Computer Security Fall 2016	Assignments	Reading	Chapters / additional reading

Done?	Presentations	URL	slides	Backup URL	Note

Who?		Presentation Topics
Elishuwon & Shachy		**Jailbreaking on latest Apple devices. How it is done, details of a particular flaw (idea behind it) and a demo.**
		Chronic Dev team, iPhone Dev Team. Look at the latest jailbreak (tethered)
		Android exploits?
		http://antid0te.com/syscan_2013/SyScan2013_Mountain_Lion_iOS_Vulnerabilities_Garage_Sale_Whitepape pdf
		http://www.idownloadblog.com/2013/04/12/evad3rs-qa-from-hitb-2013/
		http://blog.azimuthsecurity.com/2013/02/from-usr-to-svc-dissecting-evasi0n.html
Ayanna		**Defeating ROP -- and can this defense be circumvented? Read latest academic literature.**
		http://www.syssec-project.eu/media/page-media/3/gfree-acsac10.pdf
		Tutorial: https://www.corelan.be/index.php/2009/07/19/exploit-writing-tutorial-part-1-stack-based-overflows/
		http://web.mit.edu/ha22286/www/papers/conference/Systematic_Analysis_of_Defenses_Against_Return_Ori
Amanda & Kamya		**Botnets: What is going on, what are the techniques used by attackers, and what are the methods used by defenders? Give a comprehensive overview of how botnets are structured today, what they are used for, how we dismantle them, and some war stories.**
		Read up papers by Lorenzo Cavallaro, papers from WOOT, LEET, etc.
		Make sure you cover Sandworm, Flame, etc.
		ASP.NET Framework Padding Oracle (CVE-2010-3332). Give comprehensive overview of the idea behind a padding oracle, and a hypothetical or real demo of how it could be exploited.
		http://en.wikipedia.org/wiki/Pwnie_Awards
		http://pwnies.com/winners/
		https://media.blackhat.com/bh-eu-10/whitepapers/Duong_Rizzo/BlackHat-EU-2010-Duong-Rizzo-Padding-Oracle-wp.pdf
		Make sure to discuss how small leakage of information can be iterated to obtain important information
Hiren		**Identifying and Exploiting Windows Kernel Race Conditions via Memory Access Patterns**
		Pwnium award 2013: Mateusz "j00ru" Jurczyk, Gynvael Coldwind
		http://j00ru.vexillium.org/?p=1695

		The research consisted of two major parts: employing CPU-level OS instrumentation to locate potential doubl fetch vulnerabilities in the kernels of different operating systems, and discovering and testing practical means of exploiting such memory-bound race conditions in practical scenarios. Not only the topic is interesting, but bochspwn was used to find at least 37 vulnerabilities in windows kernel / drivers (plus some minor system crashes).
		OS incomplete codesign bypass and kernel vulnerabilities (CVE-2013-0977, CVE-2013-0978 and CVE-2013-0981
		Pwnium award 2013: David Wang aka planetbeing and the evad3rs team
		http://conference.hitb.org/hitbsecconf2013ams/materials/D2T1%20-%20Pod2g,%20Planetbeing,%20Musclenerd%20and%20Pimskeks%20aka%20Evad3rs%20-%20Swiping%20Through%20Modern%20Security%20Features.pdf
		According to statistics in February, the evasi0n exploit works for at least 5 million people every time they boot their iPhone. It bypasses code signing by interposing with an incomplete codesign bug in the dynamic loader. It bypasses user space ASLR by using the dynamic linker. It exploits an untrusted pointer in the kernel with some help from a heap info leak, the ARM data abort interrupt handler and some techniques by Tarjei Mandt by Mark Dowd.
		Breaking out of the Chrome Sandbox: details about how it is regularly done, what kind of effort is required, overview of the architecture and some technical detalis about a particular bug
		http://www.chromium.org/Home/chromium-security/pwnium-2
		Breaking out of the Chrome sandbox: Analyzing the exploits that have been released against Chrome
Rui?		**Rowhammer in 2016**
		https://www.usenix.org/system/files/conference/usenixsecurity16/sec16_paper_razavi.pdf
		http://arstechnica.com/security/2016/08/new-attack-steals-private-crypto-keys-by-corrupting-data-in-compute
		http://arstechnica.com/security/2016/10/using-rowhammer-bitflips-to-root-android-phones-is-now-a-thing/
		Owning the Intel System Management Mode (SMM)
		Firmware update code in the open source UEFI reference implementation was identified as containing severa vulnerabilities last year. Successful exploitation resulted in the ability for a privileged ring 3 process to stage a payload in the context of the firmware and then invoke and exploit the vulnerable UEFI firmware update code This userland (ring 3) to firmware/SMM ("ring -2") privilege escalation vulnerability is present on the majority o PC OEMs, affecting over 500+ *models* from HP alone. Other vendors have also issued patches for dozens their models, and because the UEFI reference implementation is used as the starting point by many OEMs, many other vendors are known to be vulnerable that will probably never acknowledge it, or release patches. Work by Corey Kallenberg, Xeno Kovah, John Butterworth and Sam Cornwell.
		http://blog.cr4.sh/2015/07/building-reliable-smm-backdoor-for-uefi.html

		Present an original bug
		Go forth, investigate source codes (web frameworks are a common treasure trove for bugs) and present your findings
		0wning Adobe Acrobat while bypassing ROP and breaking out of its sandbox
		Adobe Reader Buffer Overflow and Sandbox Escape (CVE-2013-0641)
		http://www.fireeye.com/blog/technical/cyber-exploits/2013/02/in-turn-its-pdf-time.html
		http://blogs.mcafee.com/mcafee-labs/analyzing-the-first-rop-only-sandbox-escaping-pdf-exploit
		http://blogs.mcafee.com/mcafee-labs/digging-into-the-sandbox-escape-technique-of-the-recent-pdf-exploit
		Just in time for last Valentine's day, FireEye found a sophisticated PDF attack in the wild that exploited Adobe Reader and escaped its sandbox. This exploit wanted to show its love for clipboard buffer lengths all in a pure-ROP payload.
Jordan		**How NSA seems to decrypt the internet: How Diffie-Hellman key exchange fails in practice. Explain the logjam approach in details, and ideally set up a hypothetical demo.**
		Credit: David Adrian et al.
		This paper introduces the Logjam attack, a vulnerability that allows a man-in-the-middle attacker to downgrade TLS connections to 512-bit export-grade Diffie-Hellman and recover the session keys. It then goes on to make a convincing case that the NSA is already doing this for 1024-bit Diffie-Hellman. Although this would require an enormous investment in computing power (perhaps the biggest secret crypto project since WW II), it would allow them to passively eavesdrop on about half of encrypted VPN and SSH traffic. This explanation precisely fits the crypto breaks described in the Snowden leaks. This paper is a landmark result, in that it uncovers a major blindspot in the relation between crypto theory and security practice, introduces a novel TLS break that is practical to exploit today, and solves a major open question about government mass surveillance capabilities.
		MS11-098: Windows Kernel Exception Handler Vulnerability (CVE-2011-2018)
		Credit: Mateusz "j00ru" Jurczyk
		http://j00ru.vexillium.org/blog/20_05_12/cve_2011_2018.pdf
		j00ru owned Windows. All of them. Ok, well just all of the 32-bit versions of Windows from NT through the Windows 8 Developer Preview. What have you done lately?
		Bug in LZC/LZH compression -- owns all SAP databases!

		Awarded to the person who discovered or exploited the most technically sophisticated and interesting server-side bug. This includes any software that is accessible remotely without using user interaction.
		SAP LZC LZH Compression Multiple Vulnerabilities (CVE-2015-2278, CVE-2015-2282)
		Credit: Martin Gallo
		SAP products make use of a proprietary implementation of the Lempel-Ziv-Thomas (LZC) adaptive dictionary compression algorithm and the Lempel-Ziv-Huffman (LZH) compression algorithm. These compression algorithms are used across several SAP products and programs. Vulnerabilities were found in the decompression routines that could be triggered in different scenarios, and could lead to execution of arbitrary code and denial of service conditions. Basically a single bug that pwns almost ALL SAP products and services.
Catherine & Rachel		**Hacking Windows and Adobe Acrobat through your TrueType fonts. Include a technical description, overview of why this is important, and a demo.**
		The "BLEND" opcode font bug was in a shared code base used both in Adobe Reader font renderer and Microsoft Windows Kernel (32-bit) font renderer. It allowed both to get code execution in Adobe Reader using a font embedded in a PDF file, and to later escape the sandbox and get SYSTEM rights by exploiting the exact same bug in the shared codebase in the Windows Kernel (ATMFD.DLL driver, part of Windows GDI).
		http://j00ru.vexillium.org/?p=2520
		http://googleprojectzero.blogspot.com/2015/08/one-font-vulnerability-to-rule-them-all_21.html
		Explain TOCTOU (time-of-check time-of-use) race condition vulnerabilities with a real-life demo
		See for instance https://googleprojectzero.blogspot.com/2016/08/a-shadow-of-our-former-self.html
Jeff and Chris		**Explain how one-byte-overwrites work, using this as an example. Create a demo that works**
		https://daniel.haxx.se/blog/2016/10/14/a-single-byte-write-opened-a-root-execution-exploit/
Robert and Brandon		**Explain the intricacies of NAND mirroring attacks, going into the academic literature to discuss evaluation etc.**
		http://arstechnica.com/security/2016/09/iphone-5c-nand-mirroring-passcode-attack/
		https://arxiv.org/abs/1609.04327
John & Ethan		**Hacking anti-virus -- show a demo of some form, check in if Metasploit has an exploit for it**
		https://googleprojectzero.blogspot.com/2016/06/how-to-compromise-enterprise-endpoint.html
		https://googleprojectzero.blogspot.com/2015/12/fireeye-exploitation-project-zeros.html