Syllabus for CS 491/591 Computer Security and Privacy, Spring 2009 (offered over ITV)

Instructor: Jed Crandall, crandall@cs.unm.edu

Prerequisites: None formally. If you don't know C and the basics of a UNIX environment, that's okay, but you should talk to me early in the semester about it so we both know what to expect.

If you are not a CS major, that's also okay, just talk to me early in the semester and we can tailor the class to what you are interested in learning.

TA: You're all TAs! Just kidding, but I think we have a lot to learn from each other so I hope you'll all post your questions and ideas to the chat mailing list for the class and reply to any emails on that list that you have thoughts about or answers for. This class may be assigned a TA early in the semester, but for getting help on the labs I hope you'll help each other since we all have a lot to learn from each other.

Mailing lists: You are required to join the main class mailing list that I will use for announcements, etc. You are strongly encouraged to join the chat list for the class and participate in discussions. This list is for sharing interesting articles you come across, asking for help from your classmates on one of the projects, posting ideas and thoughts, or anything you want to use it for. If you're one of those people that's vigilant about your inbox there is a digest mode. Discouraging other students from using the list with replies like, "why are you spamming us with this stupid question" will not be tolerated. I'll post a link and instructions for joining both lists on the course web page.

Course Web Page:

http://www.cs.unm.edu/~crandall/591spring09/

I'll post test keys, supplementary materials, readings, *etc.,* here. The planned course schedule is also there.

Required text: *Computer Security: Art and Science* by Matt Bishop. This is the brown graduate version of the book, do not buy the green version which is just a stripped down version of the same thing.

Class meeting time and place: MWF 11-11:50am in MECH 214. Regular attendance is required unless you've made prior arrangements with me. ITV students are always welcome to attend the class in Albuquerque.

Office hours: My office is FEC 335 (note that I've moved if you've taken a class from me before), office hours are Mon 3-5pm and Thu 9-10am. You can call me during office hours at 505-277-0380, outside of office hours e-mail is preferred.

If you happen to be in FEC and my door is open or cracked, feel free to knock and see if I'm free. Also, you can e-mail me to make an appointment.

Never hesitate to e-mail me directly about anything.

Final: Wednesday, 13 May 2009, 10:00am to 12:00pm. Don't make travel plans to leave earlier than this, exceptions will only be made in accordance with UNM policy (*e.g.*, if you have three finals scheduled on the same day).

Readings and homework: I'll assign light homeworks and supplemental reading from sources other than the book as I feel necessary, but your main workload outside of class will be on the projects.

Grading: There will be four tests and a final. Tests are each 7.5% of your final grade and the final will be 15%. Another 5% is homework. The remaining 50% is the projects. For the final grade you'll need to break 90.0 for an A, 85.0 for an A-, 80.0 for a B+, 75.0 for a B, 70.0 for a B-, 65.0 for a C+, 60.0 for a C, 55.0 for a C-, and 50.0 for a D. Below 50.0 is an F. I don't give A+'s or D-'s, might consider a D+ out of sympathy.

The tests and the final will be very challenging. *Do not panic if you get, e.g., below a 50 out of 100.* I'll curve all tests before handing them back so that your raw score will be written in red and the curved score that will appear in the grade book will be written in blue. If you must miss a test, you might be able make it up with my permission, but the test will only be half your score and you have to make up the other half with an essay. The essay score is capped by your makeup test score so it can't possibly improve your grade. Do not talk to anybody about the test as I might use the same test as a makeup test. Always contact me in advance about makeup tests, only under extreme circumstances will I allow a makeup test when I didn't give you permission prior to the date of the test.

Security and privacy does have some core content that you should know before I give you a passing grade, but really we're all here to have fun and learn about a very interesting topic so I'd be happy to give everyone high grades in this class. Conceivably, everyone could get an A on all the tests. ***When I'm deciding the curve for each test, attendance and participation will be a huge part of my decision on how to curve it. Thus, implicitly, attendance and participation are about half of your grade.*** I won't be taking attendance but I do notice who comes to class and who doesn't. Also, the course is ITV so Big Brother is watching you. If you need to miss a lecture and watch it on video let me know that that's what you're going to do. If you miss two lectures in a row without contacting me I reserve the right to use the instructor drop feature in LoboWeb.

Do the homework. 5% doesn't seem like much but it's the difference between one grade and another. Also, my curving decisions on the test will also be based on who's doing the homework. As per university policy you get two late days total, *i.e.*, you can turn in two different assignments one day late or one in two days late but then you've used your late days. If you have some special circumstances, like a rogue dump truck had its emergency brake fail and took out the power and Internet to your house, and then came to a stop blocking your driveway, let me know and I might not charge you your late days (though that particular story won't work a second time).

You are expected to work on the projects every week. If you get them all done, great, if you only get two or three done all semester that's okay as long as you learned a lot. Your grade will be based on your weekly progress which you will record on a blog. You can use BlogSpot or whatever you want, if you use your personal blog create a tag for your project entries and give me the link with the tag. You are responsible for your own backups, *etc.*, and technical problems with the blog are your own problem.

Journal entries are due Wednesday at midnight every week. There are 15 weeks so you will do 15 entries total. You are encouraged to make multiple entries per week, but will receive one grade for all entries over the course of that week. You'll receive a score between 0 and 5. Your five lowest scores will be dropped at the end of the semester (but at most two of these that I drop can be in the last 5 weeks of the class), and the project component of the your grade will be the sum of the other 10. This way if you get a busy week at work you can take a week off from the projects. Also, I plan to give out lots of 0's in the early weeks of class until you're all calibrated to my expectations of what to put in your journal entries and how much work you need to put into the projects each week.

Your journal entry should describe what you've done, what you've learned, what you've tried, *etc*. You need to make tangible progress every week toward completing one of the projects, though. Each project also has something you need to turn into me at the end (a secret key, some code, *etc.*). Completing projects is a good way to get a 5 for that week. If you spend a long time "researching" a project and don't actually get anything done, I'm eventually going to notice.

Students that are registered pass/fail should contact me at the beginning of the semester so we can work out an agreement on what is expected of you. Particularly if you are not a CS major then you can register pass/fail and we can tailor my expectations to what you want to learn.

Cheating and collaboration: I hope that you'll help each other out on the projects. Looking at each other's code and sharing code snippets is okay. To prevent getting yourself into trouble you don't deserve to be in, though, I suggest not sharing your entire code with somebody. Unless specified otherwise, all projects and homeworks are individual efforts. Part of learning the nitty-gritty details of systems programming is that you have to suffer individually. If you'd really like to help out your classmate and are not sure if it would cross the line into cheating, ask me.

Each test will state at the top what materials you're allowed to use (book, notes, etc.). Not noticing, for example, that the top of the test says that it's not open notes is not an excuse. Anything not specified as open is closed. In other words if the test instructions don't say "open-iPod" you should assume that the test is closed-iPod, and if the test instructions don't say "open-cheat-sheet-on-the-inside-of-your-water-bottle-label", assume that the test is closed-cheat-sheet-on-the-inside-of-your-water-bottle-label. All university policies regarding these matters will be strictly enforced.

More about the projects and the purpose of this course:

This is not a course on hacking. I hope you wouldn't waste hundreds of dollars in tuition to learn how to be a white hat or grey hat or any silly kind of hat. We're here to learn how to be computing professionals. For graduate students and those considering careers in research, you need to know the principles of attacks so that your theories are firmly planted in practice. For others, you need to know this stuff to make intelligent decisions about security. If, for example, a machine gets compromised and your boss wants to know if the attacker could have sniffed some important network traffic of other machines using the compromised machine, you should be able to give your boss an intelligent answer (based on what privileges the attacker was able to gain on the system and the configuration of hubs, switches, routers, *etc.*). If you've forged ARP packets before yourself, you're more likely to be able to give an intelligent answer. Giving you these kinds of experiences is the only reason for the projects. ***That most of the projects take the form of a simulated attack is only coincidental.***

Because hacking is not what the class is about, "creative" solutions---such as gaining root and stealing other people's source code, or using ARP injection to sniff the postscript when I print the tests---are not acceptable and you will be punished under university policy and possibly prosecuted for this kind of behavior. It's okay to be creative, but within the bounds specified. If you're not sure if something will be acceptable or not, e-mail me first.

There will be one machine dedicated as *the* dedicated machine for carrying out your projects, I'll give you an account on it. The domain name is:

shasta.cs.unm.edu

No data on this machine or any of the virtual machines running on it is private, I reserve the right to view and copy your files on this machine and the virtual machines for forensic or other purposes. Keep in mind, also, that your classmates may have elevated privileges on these virtual machines, so assume nothing is private (if they steal your source code out of your directory that's their fault, not yours, though, so it's okay to keep your source code on these machines). All of the projects can be carried out in simulated environments without any need to do anything malicious, illegal, or against university policy. Thus, anything that is malicious, illegal, or against university policy falls outside of the scope of this class and you will be held personally responsible for it.

The following rules apply to all assignments and everything we learn in the class, including, but not limited to, the projects.

Rule #1: All forged packets, *e.g.*, for ARP injection or TCP fragmentation, are to be forged only on the tuntap interfaces of the dedicated machine. A tuntap is a virtual network that connects the virtual machines, so your forged packets should never actually go out on the physical Ethernet cable. We will install safeguards to protect the outside network if you should accidently route forged packets or other malicious traffic to eth0, but you may still be held responsible if these safeguards fail and you were doing something other than what the assignment says to do, so do not forge packets on the eth0 interface and do not forge packets in ways not specified in the assignment without prior permission. You are not to forge packets on any university-owned network or computer other than the dedicated machine. You take legal and ethical responsibility for any packets you forge off campus, *e.g.*, on your private network at home.

Rule #2: All malicious traffic and scanning traffic (remote exploits, nmap, *etc.*) is to be sent only from a virtual machine to another virtual machine connected by the tuntap, all of this contained within the dedicated machine. Malicious or scanning traffic other than prescribed by the assignment is not permitted without prior permission. The policies about safeguards, university-owned computers and networks, and off-campus networks from Rule #1 also apply to this rule.

Rule #3: You are to develop and execute source code for privilege escalation (*e.g.*, gaining root with a race condition or installing a kernel rootkit) only on virtual machines specified for this purpose that will be hosted on the dedicated machine. Do not exploit vulnerabilities on the other virtual machines designated for other purposes. You may use your own private systems and virtual machines that you have root access for whatever purpose you wish but then you take personal responsibility for attack containment. Do not develop or execute code, scripts, or commands to exploit vulnerabilities on university-owned computers or any computers anywhere that you are not permitted to have administrator access on. This is against the law.

Rule #4: Do not store or execute malicious code samples (viruses, worms, *etc.*) on university-owned computers other than the dedicated machine. I will place malicious code samples that you need on the dedicated machine. Do not transfer malicious code samples to or from this machine. Do not execute malicious code samples in virtual machines without my explicit permission. If you analyze and/or execute malicious code on your own personal machine you take personal responsibility for its containment, so I strongly recommend that you not do this unless you know what you are doing.

Rule #5: Do not use any cryptanalysis, cracking, sniffing or other inference algorithm, tool, or technique to violate the privacy of others or the security of any system, other than virtual machines on the dedicated machine and your own private systems. This includes dictionary attacks, keystroke logging, password cracking, and anything we learn that

involves inferring private data, passwords, or other information you are not granted access to.

Rule #6: You are responsible for any actions you take that (1) diverge from what is necessary to complete the assignment as specified and (2) violate the laws, policies, and ethical standards of the university and applicable jurisdictions, or violate the security of any system or the privacy, integrity, or availability of any person's data. This applies to all actions, not just those covered by rules 1-5.