CS 490 / ECE 492: Embedded Systems Security

Instructors: Thoshitha Gamage, Ph.D. & Tim York Ph.D.
Southern Illinois University at Edwardsville

Spring 2016 Syllabus

Course Information:

Title:	CS 490 / ECE 492: Embedded Systems Security (3 Credits)
Location:	EB 3012
Time:	M & W 12:00 – 01:15 p.m.
Course Web site:	http://www.cs.siue.edu/~tgamage/S16/CS490

Contact Information:

	Thoshitha Gamage	Tim York
Office:	EB 3053	EB 3042
Phone ☎:	650-2407	650-2615
Email ✉:	tgamage@siue.edu	tyork@siue.edu
Web Site ⌂:	http://www.cs.siue.edu/~tgamage	http://www.ee.siue.edu/~tyork/
Office Hours:	M & W 09:00 – 10:30 a.m. T & TR 02:00 – 03:30 p.m.	T & R 4:00 p.m. - 5:30 p.m. *or by appointment*

This is an upper level undergraduate course in Cyber-Physical Security that's focused on the security of embedded systems (or Internet-of-Things (IOT) as they are referred to as nowadays). The primary course objectives are:

1. to learn security through hands-on experimentation;
2. to learn the various uses and applications of embedded systems in our day-to-day lives;
3. to learn through experimentation that embedded systems, in their base form, are not secure and that security needs to be built into them through careful design and implementation;
4. to learn first hand how embedded systems can be exploited for security vulnerabilities;
5. to develop and experiment on security mechanisms for embedded systems; and
6. to develop essential problem solving and communication skills.

By the end of the semester, students are expected to have a proficient understanding of the importance of properly secured embedded systems design.

1 White Hat Agreement

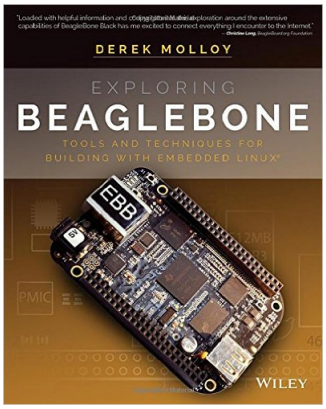
This course will most likely expose students to tools and technologies that could potentially be used to cause damage and harm to systems, infrastructures, and/or their users. The content of this course is developed with only educational and research intentions. By agreeing to this syllabus, the student **understands** that the skills he/she learns here can be dangerous if used improperly, and **agrees** to use them only in ethical ways. The student **understands** that unauthorized hacking is a crime and could get him/her into serious legal trouble which neither the instructors nor the university will be able to save him/her from.

2 Course Prerequisites

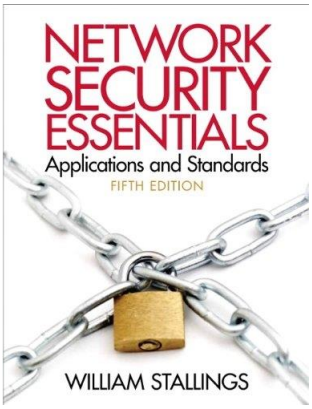
CS 314 (Operating Systems) is required for Computer Science students and CS 447 (Network and Data Communication) is also preferred. For ECE students, you must be at senior standing to take this course. Also *fluency and*

significant experience in programming (C++, Java, Python, etc.,) and Unix/Linux will be essential. If you do not meet these prerequisites, you **MUST** come and talk with the instructors the first week of class. The instructors reserve the right to drop you from the course if it becomes obvious that you do not meet the prerequisites.

3 Textbook & Resources



(a) **[BB1e]**



(b) **[NS5e]**

[Required] **[BB1e]** Exploring BeagleBone: Tools and Techniques for Building with Embedded Linux, 1^{st} ed., Derek Molloy Corman et al., Wiley, ISBN: 978-1118935125
[Supplemental] **[NS5e]** Network Security Essentials Applications and Standards, 5^{th} ed., William Stallings, Pearson, ISBN: 978-0133370430

You may find the set of useful videos on the BeagleBone by Dr. Malloy on his YouTube channel `https://www.youtube.com/user/DerekMolloyDCU/`, especially if you are new to embedded systems.

4 Assigned Work and Tentative Grading Policy

The following grade allocation breakdown is ***tentative***, and may change during the semester. Unless the circumstances change, We are **NOT** planning on curving the final grade.

Grading Allocation	
Class Participation	5%
Labs	50%
Quizzes & Homework	15%
Final Project	30%

Final Letter Grade	
90–100	A
80–89	B
70–79	C
60–69	D
below 60	F

4.1 Class Participation

You are expected to **proactively** participate in in-class discussions. This aids your learning and that of your classmates, and provides valuable feedback on the lecture. Constructive and proactive participation in in-class discussions and scribing accounts for 5% of your final grade. Each student is required to submit their scribe notes at least **two** times during the course of the semester. Scribe notes are due through *Moodle* within **48 hours** after the lecture. Only the top two scribe submissions (based on Moodle timestamp) will be counted as valid submissions.

Scribe notes serve as a baseline set of complementary notes to you and your colleagues, hence please do your due diligence to make them readable by others.

Students are also **required** to check the course website and the SIUE email account regularly for any important updates.

4.2 Final Project

The final project will be a chance for students to explore some of the security techniques learned through the class. This can be done through a variety of methods. For some examples, students could attempt to "hack" an already existing IoT device, try to develop their own exploit technique, or try to design their own device using secure techniques. Students should start thinking about the project as soon as possible. Tentatively, sometime around Week 10, the students will have to submit a written proposal detailing what they will do for the project. The expectations for a complete project must be made clear in the proposal. The instructors and students will meet individually to go over the proposal and ensure the scope of work is adequate, with a formal agreement upon what will constitute the complete final project.

5 Course Requirements and Policies

5.1 Attendance Policy

For unforeseen circumstances, there will be times when you are unable to attend the lecture. Thus, I allow you to miss at most 2 classes for the semester without any penalties. Medical emergencies are outside this "absentee allowance", but should be accompanied by proper documented proof of medical services. For planned absences, assignments should be turned in before the absence rather than after. I reserve the right to lower the grade of any student who is markedly deficient in attendance and/or in in-class participation. If you miss a class, it is *your* responsibility to find out what happened and to collect any material that was handed out in the class.

5.2 Late Policy

Unless otherwise noted or announced in-class, any leftover questions from in-class problem solving sessions are due within a week at the beginning of the next immediate class. Programming assignments typically have a 2 week deadline. Assignments may be turned within 48 hours *grace period* after the deadline with a 20% late penalty. No assignment is accepted beyond that.

5.3 Responsible Learning Policy

We expect *you* to *own* your degree of success in this class *and*, We expect you to contribute to the success of others. Examples:

- Read outside the class on your own in preparation for each lecture, jot down any questions your encounter on your reading (strongly encouraged), and bring those to the class as discussion points;
- Be respectful of the learning environment. Refrain from activities that may disturb the flow of the lecture or the environment;
  - Do not engage in disruptive *"little talk"* while We are conducting the lecture; if you have a concern, raise your hand and grab my attention. be respective of your colleagues time and desire to learn.
  - Put your cell phones to vibrate mode and refrain from using your computers for casual web browsing. Take full advantage of the opportunity to learn.
- Cooperate with other students and to share your knowledge during in-class discussions. Respect the differences in learning and understanding of each other. Seek ways of taking advantage of those differences;
- If another student is confused, help him or her out without disturbing the class;
- We enjoy engaging in technical conversations with students with the goal of helping them create an accurate understanding of course material. Participating in such conversations is very favorable for your class participation grade;
- If We are systematically doing something that inhibits your learning, tell us;

- Engage in *proactive learning*: speak up when you don't understand, question assumptions, relate course material to your experience outside class, seek out additional experience and reading related to the class. You must *construct* your understanding of the material;
- If a lecture point is unclear, ask questions and ask the instructor to repeat what he said, preferably in class, during office hours, or through e-mail. You are probably not alone in your confusion;
- Promptly review feedback you receive from me or other students to actively clarify the feedback if the material is still unclear and to incorporate the feedback in your future work;
- Spend adequate time on the course. Adequate time includes getting enough rest so that time you spend on course tasks is well-spent time. Adequate time includes proofreading and reviewing your assignments before you hand them in;
- Have high expectations of yourself: set goals for yourself and try to do your very best. Consciously think about the balance between what you do to earn a grade and what you do to learn (If we are doing something that puts these in opposition to each other, please let us know.); and,
- Check your SIUE assigned student email and the course website regularly for important class announcements.

IMPORTANT: I strongly discourage you from getting into discussions with us about grades and how you can get a better one. This includes emailing us about possible ways to "bump" your grade. Such requests only mean one thing; that you have already fallen behind on your own expectations.

5.4 Academic Dishonesty: http://www.siue.edu/policies (3C1 & 3C2)

Do your own work. Your exams, homeworks, and programming projects are subject to the academic honor code. **DO NOT CHEAT IN ANY WAY: DO YOUR OWN WORK!**. Following activities will be considered academic dishonesty:

- Submitting work (such as homework assignments projects, and code) done by somebody else (this includes any human/electronic sources (such as web sites));
- Watching and copying your neighbors' solutions during quizzes and/or exams;
- Using materials not allowed during quizzes and exams;
- Using materials not allowed for the programming projects.

It is quite acceptable to ask others things like "Have you come across this particular issue/error/exception before?," and even having them briefly look briefly at your stack trace and/or its code. To have them spend hours helping develop or seriously rearrange your program's logic, on the other hand, is not acceptable. And, of course, it is unacceptable for two or more people to collaboratively develop the solutions to assignments. If you are tempted to collaborate on such assignments, **DON'T!!**.

I expect you to know and observe the SIUE Student Conduct Code (3C1) and Student Academic Code (3C2). Copying of other students' work, working together on individual assignments, plagiarism of published sources and other forms of academic dishonesty will result in zero credit on the assignment for all students involved and a lower grade in the class. A second offense (across the University) will result in an automatic **F** in the course and exposes the violator to University sanctions up to and including expulsion. All offenses will be reported to Student Affairs.

5.4.1 Advice

a Don't wait until the last minute to do homework or projects. Labs get busy, computers break down, and people get sick. These are not sufficient excuses for an extension.
b Save early; save often!
c Contact us if you are confused. Don't wait for office hours; send an email.

5.5 Disability Support Services: http://www.siue.edu/dss

Any student inquiring about academic accommodations because of a disability should contact Disability Support Services so that appropriate and reasonable accommodative services can be determined and recommended. Dis-

ability Support Services is located in Student Success Center, Room 1270. Their phone number is 650-3726 and their email is disabilitysupport@siue.edu.

6 Tentative Schedule and List of Topics*

*Subject to adjustment and change. We reserve the right to change topics or add an item of related interest. All changes will be announced in class.

- Basics of Security, Embedded Systems, and Linux
- Low Level Protocols (I2C, SPI, UART, CAN, etc.)
- Encryption Techniques
- Network Security by Layers (TLS, IPSec, GPG, etc.)
- Wifi Security, 802.11 Communication
- Other Wireless Technologies (Bluetooth, 802.14.5 (Zigbee), NFC)
- Defenses and Penetration Testing
- Trusted Computing
- Vehicle/CAN Bus Security
- Secure Programming
- Common Attack Vectors and Sidechannels

List of Tentative Labs

- Introduction to Embedded Linux
- I^2C Communication & Memory Dumping
- Simple Symmetric Encryption of I^2C Memory
- PKI for Lab Network
- WiFi Sniffing
- WiFi Encrpytion
- Zigbee Mesh Networks (Sniffing, Spoofing, Authentication)
- Crypto Cape for Trusted Computing
- Secure Communication (Client/Server Using NRF24L01 Radios)