Course Outline

Contents

- Course Details (https://webcms3.cse.unsw.edu.au/COMP6447/20T2/outline#details)
- Course Summary (https://webcms3.cse.unsw.edu.au/COMP6447/20T2/outline#summary)
- Assumed Knowledge (https://webcms3.cse.unsw.edu.au/COMP6447/20T2/outline#background)
- Student Learning Outcomes (https://webcms3.cse.unsw.edu.au/COMP6447/20T2/outline#outcomes)
- Teaching Strategies (https://webcms3.cse.unsw.edu.au/COMP6447/20T2/outline#strategies)
- Teaching Rationale (https://webcms3.cse.unsw.edu.au/COMP6447/20T2/outline#rationale)
- Student Conduct (https://webcms3.cse.unsw.edu.au/COMP6447/20T2/outline#conduct)
- Assessment (https://webcms3.cse.unsw.edu.au/COMP6447/20T2/outline#assessment)
- Course Schedule (https://webcms3.cse.unsw.edu.au/COMP6447/20T2/outline#schedule)
- Resources for Students (https://webcms3.cse.unsw.edu.au/COMP6447/20T2/outline#resources)
- Course Evaluation and Development (https://webcms3.cse.unsw.edu.au/COMP6447/20T2/outline#evaluation)

Course Details

Course Code	COMP6447
Course Title	System and Software Security Assessment
Convenor	Richard Buckland (/users/z9700463)
Lecturer	Brendan Hopper (https://webcms3.cse.unsw.edu.au/users/z3358973) , Adam Tanana (https://webcms3.cse.unsw.edu.au/users/adambanana)
Admin	Ash Liu (https://webcms3.cse.unsw.edu.au/users/z5161603)
Classes	**Lecture** : • Tuesday 6pm-8pm **Laboratories** : • Thurs 11am - 1pm • Thurs 1pm - 3pm • Thurs 5pm - 7pm • Fri 9am - 11am • Fri 11am - 1pm • Fri 1pm - 3pm Timetable for all classes (/COMP6447/20T2/timetable)
Consultations	In your lab time, or on the course forums at any other time.
Units of Credit	6

Course Website	http://cse.unsw.edu.au/~cs6447/20T2/ (http://cse.unsw.edu.au/~cs6447/20T2/)
Handbook Entry	http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP6447.html (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP6447.html)

Course Summary

This course looks at cyber attack and defence. Students learn how to assess and identify vulnerabilities and how vulnerabilities are exploited. Students learn the principles and theory of exploitation, the common security models, and how approaches to exploitation and defence have evolved over time.

Students from this course will engage in wargames, analyse real world case studies of vulnerabilities in complex software used on widespread systems, and gain an understanding of the technical process of finding and fixing low-level software vulnerabilities and also of the economics and causal factors involved with their real world use.

The course covers techniques and skills including vulnerability classes, source code auditing,fuzzing, security bugs, software security assurance, taint analysis, memory corruption, overflows and return oriented programming. The course coverage will be constantly updated over time to reflect emerging attack and defence methods.

There are numerous formative assessments and activities throughout the course to provide feedback and learning opportunities.

Students need a keen, devious and analytical mind.

To get the most from this course you will need to engage in independent study and act as a self-directed learner. Attending lectures alone will not be sufficient to pass the course. You will need to devote considerable practice to all the techniques we cover and read further on topics which interest you or which you do not fully understand. For a credit level result we expect you will spend 14 hours per week in total on this course.

Seek feedback from your friendly lecturers, tutors and class peers constantly over the term and closely monitor yourself to make sure you are not falling behind. Experience has shown that students who do not work hard at the course do not do well, and often express disappointment later on at the missed opportunity.

Assumed Knowledge

Before commencing this course, students should have:

- completed COMP6441 (75+) or COMP6841 (65+) or COMP3231/9201 Operating Systems; and
- a good understanding of how computers work at a low level

It will be extremely useful to have a strong understanding of:

- Virtual Memory
- C Programming Language
- Linux Operating Systems

Student Learning Outcomes

After completing this course, you will:

- Have a knowledge of the principle elements of offensive cyber security (such as vulnerability classes, source code auditing, security bugs, memory corruption, numeric overflows, heap exploitation and return oriented programming)

- Recognise and explain how these elements can be exploited by attackers, their characterising features, weaknesses and countermeasures.
- Given a system, be able to identify vulnerabilities and design and implement reliable exploits.
- Given a system, be able to identify vulnerabilities and design and implement reliable countermeasures to prevent successful exploitation.
- Have an understanding of the key legal, ethical, and professional issues of offensive-defence; and to be able to apply this understanding to design and conduct professional offensive-defence operations.

This course contributes to the development of the following graduate capabilities:

Graduate Capability	Acquired in
Scholars capable of independent and collaborative enquiry, rigorous in their analysis, critique and reflection, and able to innovate by applying their knowledge and skills to the solution of novel as well as routine problems	Lectures, Wargames, Project
Entrepreneurial leaders capable of initiating and embracing innovation and change, as well as engaging and enabling others to contribute to change	Project
Professionals capable of ethical, self-directed practice and independent lifelong learning	Wargames, Project
Global citizens who are culturally adept and capable of respecting diversity and acting in a socially just and responsible way	Lectures, Project

Teaching Strategies

- **Lectures** : used to introduce students to theoretical and practical concepts and will include live demonstrations. A detailed list of lecture topics and the slides used for the lectures will be posted on the course website as session progresses.
- **Laboratories** : self-guided activities with facilitators to assist your learning.
- **Wargames** : practical exercises to consolidate learning.
- **Project** : develop and apply in-depth knowledge on particular aspect of cyber security.

Teaching Rationale

Technical security is best learned by practice, and labs, wargames, and the project are critical components of the course. These allow students to practice design and implementation skills, and to develop teamwork skills. The project will also assist in developing your ability to reflect on your own work.

Lectures will be split between discussion of concepts, discussion of practical work (and practical demonstrations), revision work, and extension lectures. Study material will be available in both video and note form before each lecture.

Student Conduct

The **Student Code of Conduct** (Information (https://student.unsw.edu.au/conduct) , Policy (https://www.gs.unsw.edu.au/policy/documents/studentcodepolicy.pdf)) sets out what the University expects from students as members of the UNSW community. As well as the learning, teaching and research environment, the University aims to provide an environment that enables students to achieve their full potential and to provide an experience consistent with the University's values and guiding principles. A condition of enrolment is that students *inform themselves* of the University's rules and policies affecting them, and conduct themselves accordingly.

In particular, students have the responsibility to observe standards of equity and respect in dealing with every member of the University community. This applies to all activities on UNSW premises and all external activities related to study and research. This includes behaviour in person as well as behaviour on social media, for example Facebook groups set up for the purpose of discussing UNSW courses or course work. Behaviour that is considered in breach of the Student Code Policy as discriminatory, sexually inappropriate, bullying, harassing, invading another's privacy or causing any person to fear for their personal safety is serious misconduct and can lead to severe penalties, including suspension or exclusion from UNSW.

If you have any concerns, you may raise them with your lecturer, or approach the School Ethics Officer (mailto:ethics-officer@cse.unsw.edu.au) , Grievance Officer (mailto:grievance-officer@cse.unsw.edu.au) , or one of the student representatives.

Plagiarism is (https://student.unsw.edu.au/plagiarism) using the ideas or words of others and presenting them as your own. UNSW and CSE treat plagiarism as academic misconduct, which means that it carries penalties as severe as being excluded from further study at UNSW. There are several on-line sources to help you understand what plagiarism is and how it is dealt with at UNSW:

- Plagiarism and Academic Integrity (https://student.unsw.edu.au/plagiarism)
- UNSW Plagiarism Procedure (https://www.gs.unsw.edu.au/policy/documents/plagiarismprocedure.pdf)

Make sure that you read and understand these. Ignorance is not accepted as an excuse for plagiarism. In particular, you are also responsible that your assignment files are not accessible by anyone but you by setting the correct permissions in your CSE directory and code repository, if using. Note also that plagiarism includes paying or asking another person to do a piece of work for you and then submitting it as your own work.

UNSW has an ongoing commitment to fostering a culture of learning informed by academic integrity. All UNSW staff and students have a responsibility to adhere to this principle of academic integrity. Plagiarism undermines academic integrity and is not tolerated at UNSW. Plagiarism at UNSW means using the ideas or words of others and passing them off as your own.

If you haven't done so yet, please take the time to read the full text of

- UNSW's policy regarding academic honesty and plagiarism (https://student.unsw.edu.au/plagiarism)

The pages below describe the policies and procedures in more detail:

- Student Code Policy (https://www.gs.unsw.edu.au/policy/documents/studentcodepolicy.pdf)
- Student Misconduct Procedure (https://www.gs.unsw.edu.au/policy/documents/studentmisconductprocedures.pdf)
- Plagiarism Policy Statement (https://www.gs.unsw.edu.au/policy/documents/plagiarismpolicy.pdf)
- Plagiarism Procedure (https://www.gs.unsw.edu.au/policy/documents/plagiarismprocedure.pdf)

You should also read the following page which describes your rights and responsibilities in the CSE context:

- Essential Advice for CSE Students (https://www.engineering.unsw.edu.au/computer-science-engineering/about-us/organisational-structure/student-services/policies/essential-advice-for-cse-students)

SECedu Good Faith Policy

The SECedu "Good Faith Policy" applies in all of our courses. This means we expect you to act in good faith at all times. You must not act in any way so as to bring disrepute to the reputation of the course, the course staff, fellow students, the school, the university, or the ICT profession. We expect you to be a good citizen. To not invade, alter or damage the property of others including the university, invade the privacy of others, break any laws or regulations, annoy other people, deprive others of access to resources, breach or weaken the security of any system, or do or omit to do anything else which you know or suspect we would not be happy about.

Furthermore you are not to do anything which appears OK by a loophole or a strict interpretation of "the letter of the law" but which is not consistent with the spirit. Also, don't be a jerk.

If you are unsure, ask!

We expect a high standard of professionalism from you at all times while you are taking any of our courses. We expect all students to act in good faith at all times - including but not limited to:

- Respect the property of others and the university.
- Respect the privacy of others.
- Always abide by the law and university regulations.
- Be respectful and considerate of others to ensure everyone has an equal learning experience.
- Always check that you have written permission from all appropriate owners before performing a security test on any system.
- Attacking UNSW or course infrastructure is strictly prohibited.
- You are only permitted to attack a wargame challenges in order to obtain the flag, not to alter the challenge in any way or otherwise spoil the challenge for other students.
- Don’t share or publish war games or your solutions, so that others in future years can continue to enjoy them too (when solutions are shared or leak out for an exercise we have to throw away the exercise and write a new one)

Students who are found (or who have previously been found and have not disclosed this) guilty of academic or computer related misconduct or any other activity in a way which casts doubt on their ability or willingness to comply with the Good Faith Policy may be unenroled and will be not permitted to re-enrol in future offerings of the course. If you have ever been found guilty of such an activity you must disclose it to the course convener in writing immediately.

If, in our sole discretion, we feel you have violated the Good Faith Policy you will be unenroled / awarded 0 Fail for the course. Further penalties may apply also depending on the nature and severity of the violation. Students who have seriously violated the Good Faith Policy may not be permitted to re-enrol in future offerings of this or other UNSW Security courses.

Course Rules and Tips for Success

Be excellent to each other.

Have fun! Security is an extremely enjoyable and stimulating field. Approach it in a spirit of adventure and a desire to embrace challenge.

Acknowledge the contributions of others when you submit work which is not whole your own work. Except for the fuzzing project, the assessable activities in the course are to be your own work, but other study activities can be done alone or with others as you find most helpful and enjoyable. Don't do everything alone however as you'll need to demonstrate your ability to work well in teams when you apply for jobs.

Work steadily each week - don't fall behind as that can be stressful and tends to lead to surface rather than deep learning when you do get around to trying to catch up, and you’ll miss out on a lot of the potential benefit you can get from the lectures and activities if you understand things at that time. Some past students have suggested putting aside a regular scheduled time each week to work on the course, you should do whatever works for you - but don’t leave it to chance.

Join in the course community, share ideas and insights, and help others.

Read around and actively extend yourself during the course. If you already know some topics then set yourself challenges or learn about extension areas. Make sure you come out of this course substantially better than when you came in.

Assessment

Item	Due	Marks

Weekly Wargames • These will vary in difficulty	Tuesday 17:59 of weeks 2-5, 7-10	30%
Mid-term Exam	Week 5, time limited exam to be completed any time in a 24 hour window	10%
Major Assignment • Fuzzer group assignment	Sunday 17:59, end of week 10	20%
Final Exam	Exam Period	40%

> Exam marks, and the final overall course marks are scaled to ensure consistency from year to year. That is, an HD this year should mean the same as an HD in previous years, and in general we try to ensure that you would have received the same grade in any term for the same quality of work/demonstrated ability.

Note that supplementary exams will only be awarded in well-justified cases in accordance with School policy for Special Consideration. UNSW has remarkably ruthless a 'Fit to Sit/Submit' rule, which means that if you sit an exam or submit an assessment, you are declaring yourself fit to do so and cannot later apply for Special Consideration. So if you are unwell or otherwise have an extraordinary mishap in the final exam contact us in the first instance.

Course Schedule

Week	Lectures	Laboratory Content	Wargames	Major Project
1	A History of Hacking How Computers Work	Tooling Enviroment Setup	Intro to Challenges	
2	Buffer overflows Stack canaries Intro to Reverse Engineering	Buffer overflows Stack canaries Intro to Reverse Engineering	Buffer Overflow Stack Canaries	
3	Shellcode Reverse Engineering	How to write shellcode Advanced reverse engineering	Shellcode	
4	Format Strings Countermeasures - ASLR, PIE	Format Strings How to defeat ASLR, PIE	Format Strings	
5	Source Code Auditing Fuzzers	Source code auditing Fuzzer assignment walkthrough	Source Code Auditing Harder Binaries	Fuzzer Project Released
6	Flexibility Week	Review of mid-term exam	-	
7	Return Oriented Programming	Return oriented programming	Return Oriented Programming	Fuzzer Project Check-In Due

8	Heap Exploitation	Heap exploitation	Heap	
9	Revision	Harder return oriented programming - pivot	Harder Challenges	
10	Hacking in the Real World	Revision	-	Fuzzer Project Final Due

Resources for Students

- The Art of Software Security Assessment Vol 1 and 2
- Shellcoder's Handbook
- Hacking - The Art of Exploitation
- Practical Malware Analysis

Course Evaluation and Development

Of course this course is evaluated as normal using the formal myExperience system at the end of the term. However, please also give us informal feedback during the term and let us know of any problems as soon as they arise. We love suggestions! Suggestions will be listened to openly, positively, constructively and gratefully, and every reasonable effort will be made to address issues as soon as possible.

Resource created about a year ago (Friday 15 May 2020, 11:09:16 AM), last modified about a year ago (Thursday 28 May 2020, 10:18:08 AM).

Comments

(/COMP6447/20T2/forums/search?forum_choice=resource/45908)

(/COMP6447/20T2/forums/resource/45908)

There are no comments yet.