COMP[69]447 Course Outline

Security Engineering Workshop

Trimester 2, 2019
6 UoC — for UG + PGC

Course Staff

Course Convenor
Prof Richard Buckland

Lecturer-in-Charge
Prof Richard Buckland

Course Admin
Lachlan Jones

Lecturers and Teaching Staff
Brendan Hopper, Keiran Sampson, Adam Tanana

For confidential course enquiries, email: cs6447@cse.unsw.edu.au.

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 wargame competitions, 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. Binary exploits = hardcore.

BEWARE

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 semester 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

You need to have taken and passed COMP6441 or COMP6841 or COMP3441 or COMP9321.

Prior to commencing the course, students should have a good understanding of how computers work at a low level. It will be extremely useful to have a strong understanding of:

1. Virtual Memory
2. C Programming Language
3. Linux Operating Systems

Course 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.

Teaching Strategies

Lectures will be 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.

Assignments : Students are expected to apply the knowledge gained in practical environments known as **wargames** as well as a major assignment.

Assignments will be subject to a late penalty. Due to their diversity, the late penalty will be tailored for each individual assignment.

The **Mid-Semester Exam** will be a practical exam where students will apply their knowledge learnt over the first half of the course.

The **Final Exam** will be a theoretical and practical exam where students apply their knowledge and skills learnt over the entire course.

Supplementary exams will only be awarded in well justified cases, in accordance with School policy for *Special Consideration* , **not** as a second chance for poorly performing students. In particular, it is unlikely that a supplementary will be awarded to students who have actually sat the proper exam. **Make up your mind whether or not you are sick before attempting the exam,** as this course adheres to UNSW's 'Fit to Sit' rule.

Student Conduct

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 is defined as using the words or ideas 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.

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

- Student Code Policy
- Plagiarism Policy Statement
- Plagiarism Procedure
- Student Misconduct Procedure

Good Faith Policy

These courses expects a high standard of professionalism from its students with regard to how Application Security Testing is conducted. 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
- Always abide by the law and university regulations
- Be considerate of others to ensure everyone has an equal learning experience.
- Always check that you have written permission from the owner before performing a security test on a system
- Attacking course infrastructure is strictly prohibited. You are only permitted to attack the wargame challenges in order to obtain the ‘flag’, not to alter the challenge in any way.

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. Basically 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.

Your actions speak volumes; It is our responsibility to uphold the reputation of the course, the course staff, fellow students, the school, the university and the ICT profession. **If you are unsure whether your actions may violate this policy, ask one of the course staff for guidance.**

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

Assessment

- **Mid Semester Exam** 10%
- **Wargames** Practical homework assignments; worth 30%
    - Each set of wargames will offer a bonus 1/2 mark for submitting by 11:59pm Friday of the relevant week (Whereas the final deadline is Sunday 11:59pm)
- **Assignment** Rootkit group assignment; worth 20%
- **Final Exam** A theoretical and practical examination. During the university's exam period; worth 40%

A mark of 50% overall **AND** a mark of 50% in the final exam is required in order to pass the course. In the circumstance that a student does not pass the final exam, their mark for coursework will be capped by their exam mark. For example, a student scoring 40% in their final exam, but who previously scored 60% in the course work will have their course work mark modified down to 40%.

Course Schedule

Week	Dates	Lecture 1	Lecturer	Assignments	Wargames	Tutorial Content
1	03/06	A History of Hacking How Computers Work	Brendan		Wargame 1 (Intro challenges)	Tooling + Environment Setup
2	10/06	Buffer overflows Stack canaries Intro to Reverse Eng.	Adam		Wargame 2 (Buffer Overflows & Stack Canaries)	Buffer overflows Stack canaries Intro to Reverse Eng.
3	17/06	Shellcode Reverse Engineering	Adam		Wargame 3 (Shellcode)	How to write Shellcode Advanced Reverse Engineering
4	24/06	Format Strings Countermeasures - ASLR/PIE	Keiran		Wargame 4 (Format Strings)	Format Strings How to defeat ASLR/PIE
5	01/07	Rootkits / Source code auditing	Keiran/Guest	Rootkit Assignment Released	Wargame 5 (Source Code, Harder Binaries)	Source code auditing Assignment walkthrough
6	08/07	Mid Semester Exam	EXAM		Reversing and Source Review	Go through midsem exam - Quiet Week
7	15/07	Return Oriented Programming	Keiran	Rootkit Assignment Check-In Due	Wargame 6 (ROP)	Return Oriented Programming

8	22/07	HEAP Exploitation	Adam		Wargame 7 (Heap)	HEAP Exploitation
9	29/07	REVISION	Keiran/Adam		Wargame 8 (ROP harder)	Harder ROP - How to Pivot
10	05/08	Hacking in the real world	Brendan/Guest	Assignment Due	-	Revision

Course Evaluation and Development

These courses are still relatively new, and we strongly encourage students to actively provide feedback about the course's progress.

These courses will be evaluated by UNSW's myExperience program. You'll receive an email to your student email address with instructions on completing this; we'll also (endeavour to) send out a notification.

If you have any feedback on the course as you are taking it, please inform your tutor or course admin (cs6447@cse.unsw.edu.au) - we would rather be aware of issues early so that we may correct them than to discover issues after the course has concluded through MyExperience.

Text and Reference Books

Textbook

- Building BSD Rootkits

Reference Books

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