Syllabus
Modern Binary Exploitation

Course Information
Course Title: Modern Binary Exploitation
Course Number: CSCI4967/ITWS 4967
Credit Hours: 4
Semester / Year: Fall 2019
Lectures: MTh 2-4pm in LOW 3112
Website: www.cs.rpi.edu/~milanova/csci4967
Prerequisite: CSCI 2500 Computer Organization (unofficial for this semester)
Instructors: Brian Callahan, Ana Milanova, and RPISEC

Course Description
Cybersecurity is one of the fastest growing fields in computer science. Modern Binary Exploitation will focus on teaching practical offensive security skills in binary exploitation and reverse engineering. Through a combination of interactive lectures, hands on labs, and guest speakers from industry, the course will offer students a rare opportunity to explore involved subjects in a rapidly evolving field.

The course will start off by covering basic x86 reverse engineering, vulnerability analysis, and classical forms of Linux-based userland binary exploitation. It will then transition into defenses found on modern systems (Canaries, DEP, ASLR, RELRO, Fortify Source, etc) and the techniques used to defeat them. Time permitting, the course will also cover other subjects in exploitation including kernel-land and Windows based exploitation, as well as static analysis of binaries.

Course Text(s)
Suggested Textbooks:
- Hacking: The Art of Exploitation, 2nd Edition by Jon Erickson
  - ISBN 978-1593271442
  - ISBN 978-0470080238
- Practical Malware Analysis by Michael Sikorski and Andrew Honig
  - ISBN: 1593272901

Student Learning Outcomes
Upon successful completion of this course, students will:
1. Possess the skills necessary to carry out vulnerability analysis of binary applications.
2. Understand executable formats, program control flow at the assembly level, and other low level concepts.
3. Understand classic and contemporary vulnerabilities and exploitation techniques.
4. Apply both source code auditing and binary reverse engineering to the vulnerability discovery process.
5. Be capable of exploiting vulnerabilities found in real world software as defined by MITRE’s Critical Vulnerabilities and Exposures (CVE) system.
**Course Assessment Measures**

- **Labs**: Typically, each lecture will be accompanied by a lab that will solidify understanding of the topic of the lecture. Each lab will contain at least 3 problems, a C, B, and A-level problem. There may occasionally be an A+-level problem for extra credit.

- **Term Projects**: There will be two projects over the course of the semester. In these projects students will be exploiting a known vulnerability in a realistic piece of software as selected by the instructors. Students will be expected to analyze the vulnerability and produce a working exploit with a writeup detailing the vulnerability and how it can be leveraged to gain privileged information. Students may work in groups of up to two for projects.

**Grading Criteria**

- **Labs**: 60%, 10 or 12 labs equally weighted at 6% or 5%, respectively, of the final grade
  - Labs will typically consist of 3 problems, a C, B, and A-level problems with each problem harder than the last. Lab grading works as follows.
    - Students who complete only the C problem will receive 75% (C)
    - Students who complete both the C and B problems will receive 85% (B)
    - Students who complete all three lab problems will receive 100% (A)
    - There may be one or two A+ problems throughout the semester. They can be used to raise the grade to 100% (A) of any previous lab.
  - If a student does not complete the C-level problem before the end of the associated lecture period, they will receive a letter grade reduction for the lab. For example, if a student fails to complete the C-level problem during the lecture period, but then goes on to successfully complete the C- and B-level problems for the week, the grade received for that week’s lab will be a C (i.e. B minus one letter grade).
  - In order to receive full credit for a problem the student must submit their exploit code, the flag, and may be asked to explain their work upon submission to be checked off.
  - Lab problems will be introduced by the end of lecture typically once a week. The C-level problem will be due in person by the end of the associated lecture period. All other problems become homework, and are due at the start of class, exactly one week after the associated lecture period.
  - Lab problems submitted late will receive a letter grade reduction (-10%). They MUST be submitted no later than the class following their original due date. Anything submitted later that that will not be accepted.

- **Term Projects**: 40%, two projects equally weighted at 20%
  - Project specific grading breakdowns will be given when they are assigned.
  - Project checkpoints exist to keep you on pace with projects, failing to make checkpoints will result in a -5% reduction to the term project grade.
  - Projects submitted late will receive a -10% reduction per day late, and will not be accepted for credit after 5 days.

- **Letter Grade Assignment (subject to change)**:
  - [92 – 100]: A
  - [90 – 92]: A-
  - [88 – 90]: B+
There will be servers dedicated to hosting the problems to be completed by students for both the lab and projects. The submitted exploits/solutions can be developed in any practical manner, but ultimately must work on the course servers to receive credit.

Grades and course progress will be made available to students throughout the semester. A grade can only be appealed within 5 days of the grade being made available to students.

Questions regarding a grade on an assignment should first be directed at the class instructors and TAs.

**Attendance Policy**

Although lecture attendance is not mandatory, it will be *exceptionally* difficult to keep up without attending. **Attendance for labs is required** as the first problem must be turned in in person during lab.

See Grading Criteria for more details

**Academic Integrity**

Student-teacher relationships are built on trust. For example, students must trust that teachers have made appropriate decisions about the structure and content of the courses they teach, and teachers must trust that the assignments that students turn in are their own. Acts that violate this trust undermine the educational process. The Rensselaer Handbook of Student Rights and Responsibilities defines various forms of Academic Dishonesty and you should make yourself familiar with these. In this class, all assignments that are turned in for a grade must represent the student’s own work. In cases where help was received, or teamwork was allowed, a notation on the assignment should indicate your collaboration.

Submission of any assignment that is in violation of this policy will result in a penalty of a **zero for the assignment** for all parties involved. **Repeated offenses will result in a failing grade for the course.**

If you have questions concerning this policy before submitting an assignment, please ask for clarification.

**Other Course-Specific Information**

The labs and interactive exercises used in lecture will be hosted in a variety of different series on an internal ‘Wargame’ server meant purely for teaching security concepts in a safe and educational environment. Students enrolled in the course will have SSH access to the server and are expected to use it as an aide to the learning process. Attacking parts of the server infrastructure beyond the scope of what is assigned or attempts to disrupt services for the course or other students is **not allowed** and will be considered a violation of academic integrity.

Due to the experimental nature of the course and assignments being offered, the schedule may change.

Note: This syllabus was modified by Profs. Callahan and Milanova from previous editions of Modern Binary Exploitation.