Syllabus for CS4753 / 9163: Application Security
New York University
Fall 2017
Professor Cappos
Tuesdays 6PM-8:30PM; JABS 774

To contact professor: jcappos@nyu.edu
2 MetroTech, room 10.082
Phone: 646-997-3116
Office hours: Tu 4:30PM-5:30PM or by appointment

TA:
Rajesh Kumar
Office hours: Mo 3PM-4PM
Office hour location: 2 MetroTech, cubicle area 10.098

Course Pre-requisites
Students are expected to have a strong technical background before taking this course. In particular, they must have taken a computer security class (CS3923/CS6813 preferred) and should have at least two of the following:

- Some familiarity with computer architecture (ideally being able to read x86 assembly)
- Knowledge of network security (CS3933/CS6823: Network Security preferred)
- Practice developing and maintaining programs of more than one thousand lines of code
- Familiarity with at least one scripting language (Ruby or Python preferred)
- Familiarity with command line operation of Windows and Linux

All students should also have strong programming skills in at least one programming language and be comfortable in at least one other language.

Course Description
This course teaches students the fundamental technical skills required to identify and prevent application vulnerabilities. This is further elaborated upon in the textbook, The Art of Software Security Assessment, which this course draws heavily on:

"...you'll learn about the tools you need to understand and assess software security. You'll see how to apply the theory and practice of code auditing; this process includes learning how to dissect an application, discover security vulnerabilities, and assess the danger each vulnerability presents. You also learn how to maximize your time, focusing on the most security-relevant elements of an application and prioritizing your efforts to help identify the most critical vulnerabilities first. This knowledge provides the foundation you need to perform a comprehensive security assessment of an application."

- The Art of Software Security Assessment, Chapter 1

As students learn about application security issues, we apply their knowledge to a semester-long exercise in the design and development of secure applications, covering fundamental application development practices and the software security lifecycle.

Source code analysis and testing methodologies are discussed in CS9163. Students are provided multiple hands-on sessions where they perform source code analysis across a diverse
set of projects, including web applications, C code, and mobile applications. This includes live deployed application code from startups at the university incubators.

Throughout multiple lectures and projects, issues of secure programming principles and constructive techniques are introduced, reinforced, and evaluated. For example, students learn about different variable naming styles, code review processes, managing security bug bounties, unit testing philosophy, and how to write effective documentation. These skills are sharpened through several projects, including a 6 week project where students take an existing large code base and write a substantial new piece of functionality that integrates with it. Students evaluate each others' code (and the TAs do as well), with the "root cause" of discovered bugs being discussed in class. The emphasis is on finding others mistakes and learning about programming patterns and practices that avoid these issues in the first place.

This course is unique in that videos from guest lecturers are used to present details about most of the topics in the course, adding their individual specialization and experience to their topics. In many cases, these lectures have been recorded and the appropriate class time will instead be used for hands-on experience on the relevant topic. In the application security class, the following invited experts are expected to teach classes:

- Dan Guido, Trail of Bits
- Alex Sotirov, Trail of Bits
- Dino Dai Zovi, Trail of Bits
- Brandon Edwards, Zero Day Initiative
- Marcin Wielgoszewski, Gotham Digital Science
- Chris Rohlf, Leaf SR
- Tom Ptacek, Matasano
- Shyama Rose, CBS
- Zane Lackey, Etsy
- Justin Cappos, NYU

Each week, there will be a lecture from an expert on a core topic in the course. We plan to cover the following topics this semester, along with their associated chapters from The Art of Software Security Assessment (TAOSSA). The schedule of lectures is approximate and may change depending on the pace of the course.

**Readings**

Students must have a copy of the course textbook, *The Art of Software Security Assessment*. Parts of the course draw upon material from The Tangled Web and A Bug Hunter's Diary, however excerpts will be provided if necessary.

This class is being taught as a “flipped classroom” where students watch the lectures ahead of time and use classroom time to do hands-on related to the lectures. Therefore it is essential that students read the assigned reading, watch the assigned videos, and perform other preparatory work **before the class period where the topic is discussed**.

<p>| Intro &amp; Development Practices | TAOSSA Chapters 1-4 |</p>
<table>
<thead>
<tr>
<th>Windows: Objects, the Filesystem, and IPC</th>
<th>TAOSSA Chapters 11 and 12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exploit Mitigations and Privilege Reduction</td>
<td>TAOSSA Chapter 5</td>
</tr>
<tr>
<td>Containment</td>
<td>outside material</td>
</tr>
<tr>
<td>Mobile Security</td>
<td>outside material</td>
</tr>
<tr>
<td>Security for Enterprises / Startups</td>
<td>outside material</td>
</tr>
<tr>
<td>Metacharacters, Privileges, and Files</td>
<td>TAOSSA Chapters 8, 9, and 13</td>
</tr>
<tr>
<td>C, Program Building Blocks</td>
<td>TAOSSA Chapters 6 and 7</td>
</tr>
<tr>
<td>Web Security</td>
<td>TAOSSA Chapters 17 and 18</td>
</tr>
<tr>
<td>Practical Cryptography</td>
<td>outside material</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Date</th>
<th>Video / Lecture Topic</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sept 5</td>
<td>Intro / Development Practices</td>
<td>Phubble introduced</td>
</tr>
<tr>
<td>Sept 12</td>
<td>Windows Internals</td>
<td>Phubble checkpoint</td>
</tr>
<tr>
<td>Sept 19</td>
<td>Memory Corruption</td>
<td>Phubble report due</td>
</tr>
<tr>
<td>Sept 26</td>
<td>Sandboxing</td>
<td>Startup introduced</td>
</tr>
<tr>
<td>Oct 3</td>
<td>Mobile App Sec</td>
<td>Startup milestone</td>
</tr>
<tr>
<td>Oct 10</td>
<td>Security for enterprise / startup</td>
<td>Startup final report</td>
</tr>
<tr>
<td>Oct 17</td>
<td>MIDTERM</td>
<td></td>
</tr>
<tr>
<td>Oct 24</td>
<td>Reverse engineering</td>
<td>Reprobuilds intro</td>
</tr>
<tr>
<td>Oct 31</td>
<td>Project startup</td>
<td>Reprobuilds checkpoint</td>
</tr>
<tr>
<td>Nov 7</td>
<td>Code Auditing 1</td>
<td>Reprobuilds final</td>
</tr>
<tr>
<td>Nov 14</td>
<td>Code Auditing 2</td>
<td>Main project intro</td>
</tr>
<tr>
<td>Nov 21</td>
<td>Web apps</td>
<td>Main project checkpoint</td>
</tr>
</tbody>
</table>
Course assignments
There are also four main projects in the class. In the first project, students analyze a web app and try to find issues in it. Students will write a detailed report on bugs in the web app.

The second project involves doing a security assessment for startup companies. Students will examine the websites and applications for startup companies and provide details about flaws.

The third project involves students getting experience patching flaws in widely used software. Students will work with participants from the Reproducible Builds project to address issues in widely used software.

The final project involves taking a large off-the-shelf piece of software and extending it to add substantial security functionality. As part of this process, students must build white box and black box tests that demonstrate the desired security functionality has been correctly added. Students will deploy their solutions in practice and review each other’s deployments for vulnerabilities and risks.

The midterm and final exam will each be about two hours. The exams will test course material presented in lectures and performed in assignments. There will be a mix of conceptual and hands-on (technical) questions.

- Midterm: 10%
- Final: 15%
- Projects: 75%

Extra credit is offered to students who research and deliver an in-depth presentation on leading or fundamental research in application security. Students must schedule time with the instructor to choose an appropriate topic as well as schedule such a presentation. Students are also encouraged to participate in CTF competitions, such as CSAW, for a small amount of extra credit.

Academic Dishonesty:
Students must follow the collaboration guidelines for the projects. All tests and quizzes will be individual. Students must behave ethically at all times. Academic dishonesty will be severely punished and will extend beyond failing the class.
For more details, see the Code of Conduct:
Moses Center Statement of Disability
If you are a student with a disability who is requesting accommodations, please contact New York University’s Moses Center for Students with Disabilities at 212-998-4980 or mosecsd@nyu.edu. You must be registered with CSD to receive accommodations. Information about the Moses Center can be found at www.nyu.edu/csd. The Moses Center is located at 726 Broadway on the 2nd floor.