CS 795/895: Topic in .Net Security
Summer 2015
Course Outline

Instructor: Ravi Mukkamala
Office: E&CS 3317
Phone: 683-7083 (Voice) 683-4900 (Fax)
e-mail: mukka@cs.odu.edu
Lecture: Monday, Wednesday 5:15-9:00 PM (ECSB 2120)
Office Hours: Monday, Wednesday: 4:00-5:00 PM (OCNPS 137)
Prerequisites: CS 555 & CS 471
Programming .NET Security (Paperback), O’Reilly (Optional)
by Adam Freeman, Allen Jones
C#: How to program, Deitel, et al, Prentice Hall

Objectives: Gain a basic understanding of systems programming for the Windows System platforms. Emphasis will be placed on network application programming. This course will cover the architecture of the windows programming environment. Major areas of study will be .Net and parts of J2EE. This course will provide a practical application of material from operating systems, computer networks, data structures and object oriented design.

PreRequisites: Strong Background in C++ or JAVA and Operating Systems. Network and Communications concepts would also be useful. Because of the importance of .NET in windows programming, you should be comfortable with the principles of object oriented programming and their implementation.

Course Objectives
Understand the issues and explore solutions for building secure distributed applications. We will discuss some key issues in distributed systems and secure systems. We will then use .Net framework to explore and experiment with the ideas that we discuss. This course certainly requires students to be explorative in nature---it is not a 500-level course.

Attendance: Attendance at classes is not mandatory, but students are responsible for all material covered and announcements made in class. Consequently, if you are going to miss class, be sure to get notes, handouts, etc., from another class member. Class notes and other information will be available at the following WebSite: http://www.cs.odu.edu/~mukka

Cheating: Everything turned in for grading in this course must be your own work. The instructor reserves the right to question a student orally or in writing and to use his
evaluation of the student's understanding of the assignment and of the submitted solution as evidence of cheating. Violations will be reported to the Honor Council for consideration for punitive action. However, it is entirely appropriate seek and give assistance on procedural matters (such as how to send e-mail, how to run the debugger, how to send files from a home PC to a UNIX workstations). If there is any question on whether a particular behavior is appropriate, the student is encouraged to seek guidance from the instructor.

**Grading:**

<table>
<thead>
<tr>
<th>Course</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exam 1</td>
<td>100</td>
</tr>
<tr>
<td>Exam 2</td>
<td>100</td>
</tr>
<tr>
<td>Course Project</td>
<td>100</td>
</tr>
<tr>
<td>Homework</td>
<td>100</td>
</tr>
</tbody>
</table>

**Tentative Schedule**

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>May 18 (M)</td>
<td>Intro to .Net, distributed systems and security, and Java security</td>
<td>HW#1</td>
</tr>
<tr>
<td>May 20 (W)</td>
<td>Part 1</td>
<td></td>
</tr>
<tr>
<td>May 25 (M)</td>
<td>Memorial day --- No class</td>
<td></td>
</tr>
<tr>
<td>May 27 (W)</td>
<td>Part 2</td>
<td></td>
</tr>
<tr>
<td>June 1 (M)</td>
<td>Part 3/Part 4</td>
<td>HW#2</td>
</tr>
<tr>
<td>June 3 (W)</td>
<td>Part 5/Review</td>
<td></td>
</tr>
<tr>
<td>June 8 (M)</td>
<td>Examination I</td>
<td>HW#3</td>
</tr>
<tr>
<td>June 10</td>
<td>Part 5/Part 6</td>
<td>HW #4</td>
</tr>
<tr>
<td>June 15</td>
<td>Part 7/Part 8</td>
<td>HW#5</td>
</tr>
<tr>
<td>June 17</td>
<td>Part 9/Review</td>
<td></td>
</tr>
<tr>
<td>June 24</td>
<td>Examination II</td>
<td></td>
</tr>
<tr>
<td>Part</td>
<td>Topics</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------------</td>
<td></td>
</tr>
<tr>
<td>Part 1</td>
<td>.Net Framework</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ASP.Net examples</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C# Examples</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ADO.net example</td>
<td></td>
</tr>
<tr>
<td>Part 2</td>
<td>Types of attacks</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Who is responsible for preventing attacks</td>
<td></td>
</tr>
<tr>
<td></td>
<td>How to prevent attacks</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Storing secrets</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Password policies</td>
<td></td>
</tr>
<tr>
<td>Part 3</td>
<td>ASP.Net security Framework</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Windows authentication</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Forms authentication</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Passport authentication</td>
<td></td>
</tr>
<tr>
<td>Part 4</td>
<td>Authorization</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Web services</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Web services security</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Impersonation</td>
<td></td>
</tr>
<tr>
<td>Part 5</td>
<td>Assemblies</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Application domains</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Lifetime of a secure application</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Runtime security</td>
<td></td>
</tr>
<tr>
<td>Part 6</td>
<td>Evidence</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Permissions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Security Policy</td>
<td></td>
</tr>
<tr>
<td>Part 7</td>
<td>Code access security</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Role based security</td>
<td></td>
</tr>
<tr>
<td>Part 8</td>
<td>.Net Cryptography</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Hashing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Symmetric key cryptography</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Asymmetric key cryptography</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Digital signatures</td>
<td></td>
</tr>
<tr>
<td>Part 9</td>
<td>XACML and security policies</td>
<td></td>
</tr>
</tbody>
</table>