Syllabus - Fall 2016

Course Title: CS6813 Information, Security and Privacy

Instructor Contact Information
Instructor: Lok Yan
Office Hours: W 9-10PM
Any evening hour by appointment
Email: LokYan@nyu.edu (Always email first!)
TA: None

Course Goal Statement
The main goal of this course is to prepare the student for future studies in information assurance and computer/network security. This is done by ensuring that students understand the different concepts and terminology used in computer security and related disciplines, as well as are comfortable with extending their knowledge through reading and studying literature.

Course Description
This course is meant to introduce the student to many different topics in Computer Security. This means that the course is designed for breadth. Select topics are explored in more detail using
assignments and the discussion forums. Thus, by the end of this course, the student should be comfortable with the many terms and concepts in Computer Security from Access Control to Zero-Knowledge Protocols. The student should not expect to become an expert in any of these topics since many of the details are elucidated in the subsequent courses, or through reading the literature. To illustrate this distinction, the student should come out of this course understanding the differences and tradeoffs between secret/symmetric key cryptography and public/asymmetric key cryptography, but will not understand the specifics of different cryptographic algorithms such as AES and RSA until the Modern Cryptography course.

**Required Reading**

The course materials will consist mainly of Audio and Video clips. The length of each module varies, but the student should expect to spend at least two hours reading, listening and watching the provided materials. Each module will also contain links to additional information that the student is expected to watch/read and use in the discussion boards.

**Course Prerequisites**

The student is expected to have C programming experience. Furthermore, it is highly suggested that the student also have a background in Python as well as x86 assembly language.

**Course Objectives**

The course objectives are embedded in the schedule below

**Interaction Policy**

Since this is an online-course you are expected to interact with your classmates and the instructor on a weekly basis. I will be holding live lectures each week whenever possible. The student is expected to attend the live lecture if possible and participate by asking questions and answering questions from fellow classmates or the instructor. If the student is unable to attend the live session, then the student is expected to watch the lecture at a later time, as well as provide a short writeup that describes a current event article, news story, academic paper, online video etc. that relates to the week’s topic. This write up (about half a page) will be sent to the entire class for their consumption. Interaction is extremely important, and therefore it is worth 10% of your final grade. Interaction is also REQUIRED! Irrespective of your final average, you will fail the course if you do not participate.

The course will be spread out over thirteen (13) weeks and will consist of (2) midterms and one (1) comprehensive final. There will also be 10 assignments. You will need to program in C so get comfortable with it early!
Grading Policy

Assignments: 72% (Best 9 out of 10, 8% each)
Discussions: 8% (Best 10 out of 13, .8% each)
Midterm 1 = 4%
Midterm 2 = 8%
Final = 8%

A if >= 94%, A- if >= 90%
B+ if >= 88%, B if >= 84%, B- if >= 80
C+ if >= 78%, C if >= 70%
F otherwise

Academic Policy: See the “New York University Tandon School of Engineering Student Code of Conduct” (http://engineering.nyu.edu/files/SACCofC2-2-16.pdf) as well as the note at the end of this syllabus.

Additional Note: This is a graduate level class and I will treat you as adults. This means:

1. If I catch you cheating or plagiarising, you and your accomplice(s) will
   a. Fill out and sign out the form at the end of this syllabus
   b. Receive a 0 (zero) in the assignment or exam or course (my discretion)
2. You will not receive extensions, unless you contact me ahead of time and make the proper arrangements
3. You will not be given an opportunity to “boost” your final grade at the end of the semester
4. You are responsible for your grades, so review your returned assignments and exams in a timely manner and contact me if there are any discrepancies
5. You are expected to ask for help when you need it. While I will monitor your progress and let you know if I see any problems, I will not make sure that you do well in the course
6. In turn, I will strive to
   a. return emails and answer appointment requests within two days
   b. grade and return your assignments and exams within two weeks
   c. post new material one week in advance
   d. let you know of any changes to the schedules at least one week in advance

Course Schedule

Tentative Syllabus (The assignments and dates might change). You are expected to watch the pre-recorded videos for each Module 1 week prior (e.g. Module 1 should be completed 6-11 SEP so we can discuss them the week of 12-18 SEP)
Module 1 (12 SEP – 18 SEP): Introduction to Security and Privacy and Security Design Principles + Assignment 1 (Understanding Online Tracking: Play with the “Collusion” plugin and understand modern privacy issues. Objective: Help the student understand the connections between information, security and privacy)

Module 2 (19 SEP – 25 SEP): Threat Modeling + Assignment 2 (Find password crackers and crack a number of available hash databases. Objective: Help students get a sense of the problems with password based authentication)

Module 3 (26 SEP – 2 OCT): Cryptography Fundamentals + Assignment 3 (Understand Counter Mode for Block Ciphers and the SHA3 competition. Objective: Help students understand the tradeoffs between security and performance.)

(3 OCT – 9 OCT): Midterm 1 (Take Home)

Module 4 (10 OCT – 16 OCT): Key Management + Assignment 4 (Understand PKI and its limitations)


(31 OCT – 6 NOV): Midterm 2 (Take Home)

Module 8 (7 NOV – 13 NOV): Containerization: Virtual Machines, Software Fault Isolation and Performance Isolation + Assignment 7 (Understand software fault isolation in practice by looking into Google NaCl. Objective: Help students understand the limitations of sandboxing using software fault isolation)

Module 9 (14 NOV – 20 NOV): Software Validity and Rights + Assignment 8 (Experiment with the Adore rootkit. Objective: Help students understand what happens to a system when the reference monitor is being bypassed and system integrity has been compromised.)
THANKSGIVING BREAK (21 NOV - 27 NOV)

Module 10 (28 NOV – 4 DEC): Attacks and Defenses 1: Buffer Overflows, side channels and beyond +
Assignment 9 (Conduct a Buffer Overflow. Objective: Help students understand what a buffer overflow
vulnerability is, and modern protection mechanisms that are used to limit the impact of such
vulnerabilities.)

Module 11 (5 DEC – 11 DEC): Attacks and Defenses 2: ASLR, Return-Oriented Programming, SQL
Injections + Assignment 10 (Perform a Format String exploit. Objective: Similar to assignment 9, except
using format strings)

MAKEUP WEEK (12 DEC - 18 DEC)

(19 DEC - 23 DEC): FINAL (Take Home)

ADA and Students with Disabilities

Please let me know if you have any special requirements. We will work with the Department and
University to accommodate you. Please see the Department and University for details.

Notes on Plagiarism for CS6813

I have failed students for plagiarizing. Students have also been expelled because of it.

It disheartens me that plagiarism is still an issue at the graduate level. One would think that graduate students
would take pride in their work and strive to better themselves (at least intellectually). This is not always the case.
NYU takes plagiarism and cheating seriously and so do I.

Many do not understand that plagiarism is also cheating. Most students seem to understand cheating in the form
of multiple students working together and sharing their solutions without prior approval though. Thus, this
document focuses on plagiarism.

I, the instructor, have the final say on whether you plagiarized or cheated (the Department and University rarely
overrule instructors), I wanted to provide you with examples of plagiarism as I see it. This is NOT an exhaustive
list, but should still give you a good sense of what to avoid.
We start with the Engineering school’s definition of plagiarism and then continue on with sample answers to the question “What is a buffer overflow?” Finally this document ends with a copy of the letter that you will have to sign if I catch you cheating.

POLICIES AND PROCEDURES ON ACADEMIC MISCONDUCT Section B.3

"Plagiarism: intentionally or knowingly representing the words or ideas of another as one’s own in any academic exercise; failure to attribute direct quotations, paraphrases, or borrowed facts or information." (http://engineering.nyu.edu/files/SACCofC2-2-16.pdf. Retrieved 17 FEB 2016)

NOTE: The above definition of plagiarism contains a ; (semicolon) which should be interpreted as an “or”.

Example 1 - Direct copy and paste

In computer security and programming, a buffer overflow, or buffer overrun, is an anomaly where a program, while writing data to a buffer, overruns the buffer's boundary and overwrites adjacent memory locations.

A student who submits the sentence above is clearly plagiarizing because it was copied and pasted directly from the Wikipedia article on buffer overflows (https://en.wikipedia.org/wiki/Buffer_overflow Retrieved 17 FEB 2016) without properly citing the source.

NOTE: The student will receive zero points for the question even if he cited the source properly. That is, while he is not plagiarizing, he also failed to demonstrate a mastery of the material.

Example 2 - Lazy Paraphrasing

A buffer overflow is problem where a program, while writing data to a buffer, overruns the buffer's boundary and overwrites adjacent memory.
In this second example, the student paraphrased the quote from Wikipedia and therefore no longer needs the citation. She failed at paraphrasing though. The student’s sentence shares the same structure and essential contents as the original source. Notice how only some words were removed, and "anomaly" was changed to "problem." This is why I call this “lazy” paraphrasing. I consider this plagiarism.

**Example 3 - Quote Concatenation**

“A buffer overflow, or buffer overrun, is an anomaly where a program, while writing data to a buffer, overruns the buffer's boundary and overwrites adjacent memory locations” (https://en.wikipedia.org/wiki/Buffer_overflow Retrieved 7 SEP 2015). They are separated into stack overflows which "occurs if the stack pointer exceeds the stack bound" (https://en.wikipedia.org/wiki/Stack_overflow Retrieved 7 SEP 2015) and heap overflows which "is a type of buffer overflow that occurs in the heap data area" (https://en.wikipedia.org/wiki/Heap_overflow Retrieved 7 SEP 2015)

In this last example, the student properly cited sources but failed to make any meaningful contributions. To me, the student quickly glued together multiple quotes and presented it as their own work. Additionally, I can imagine the student Googling “buffer overflow” and arriving at the Wikipedia page. The student then quickly took the first sentence from that article and then followed the first couple of links in the “Exploitation” section to arrive at the stack and heap overflow articles where he subsequently also quoted the first sentence.

This is also plagiarism. Remember: while you may argue that this is not plagiarism and I may even agree with you, but the final decision is still mine. I also determine what “meaningful” contributions are.

**Example 4 - Borderline Good**

[Contents from example 3]. While stack and heap based buffer overflows are the most common, they can actually occur wherever there is a buffer. This includes buffers in the “global” section of a binary for example.

In the example above, I took the entirety of example 3’s answer and added just a couple of sentences. I call this a borderline good example, because the student has now added a meaningful contribution. That is, the student added a new piece of information that was not directly retrieved from the Wikipedia article on buffer overflows and required some understanding and thinking on her part.

While I will accept this above example, you can remove all doubts by rewriting the Example 3 answer as well.
To Whom It May Concern,

This form certifies that I, NAME (N#), participated in the incident (stated below) during Professor Yan’s Summer 2016 course, CS6813, INET Section, Information Security and Privacy. This incident violated the policy of this course (for which I am officially registered), the policy of NYU’s Graduate School of Arts and Science, and/or the policy of NYU.

As a result, I have accepted the penalty (also stated below) suggested by Professor Yan.

I understand that if a similar incident happens again, the penalty may be harsher, possibly even resulting in expulsion from NYU.

Incident (example):
1. Copying from another student(s), with regard to all or part of the work I submitted as a homework assignment and/or
2. Permitting another student(s) to copy and submit as a homework assignment, all or part of the work I wrote/created

Penalty (example):
A grade of 0 for Assignment #3, as well as a 0 for the class participation grade in CS6813, INET Section, Information Security and Privacy.

Student Signature: ___________________________ Date: _____________________
Course Title: CS6813 Information, Security and Privacy

Instructor Contact Information
Instructor: Lok Yan
Office Hours: W 9-10PM
Any evening hour by appointment
Email: LokYan@nyu.edu (Always email first!)
TA: None

Course Goal Statement
The main goal of this course is to prepare the student for future studies in information assurance and computer/network security. This is done by ensuring that students understand the different concepts and terminology used in computer security and related disciplines, as well as are comfortable with extending their knowledge through self-guided reading and studying literature.

Course Description
This course is meant to introduce the student to many different topics in Computer Security. This means that the course is designed for breadth. Select topics are explored in more detail using assignments and the discussions. Thus, by the end of this course, the student should be comfortable with the many terms and concepts in Computer Security from Access Control to Cryptography as well as how they relate to each other, e.g., how to use cryptography to enforce access controls. The student should not expect to become an expert in any of these topics since many of the details are elucidated in the subsequent courses, or through reading the literature. To illustrate this distinction, the student should come out of this course understanding the differences and tradeoffs between secret/symmetric key cryptography and public/asymmetric key cryptography, but will not understand the specifics of different cryptographic algorithms such as AES and RSA until the Modern Cryptography course.

Required Reading
The course materials will consist mainly of Audio and Video clips. The length of each module varies,
but the student should expect to spend at least two hours reading, listening and watching the provided materials. Each module will also contain links to additional information that the student is expected to watch/read and use in the discussion boards.

Course Prerequisites

The student is expected to have C programming experience. Furthermore, it is highly suggested that the student also have a background in Python as well as x86 assembly language. Courses such as Operating Systems, Computer Architectures and Discrete Math/Formal Modeling are core CS courses and therefore the student is assumed to have knowledge of those concepts.

Course Objectives

The course objectives are embedded in the schedule below

Interaction Policy

Since this is an online-course you are expected to interact with your classmates and the instructor on a weekly basis. I will be holding live lectures each week whenever possible. The student is expected to attend the live lecture if possible and **participate by asking questions and answering questions** from fellow classmates or the instructor. To reiterate, participation is not just attending the sessions, you must ask and/or answer questions. If the student is unable to attend the live session, then the student is expected to watch the lecture at a later time, as well as provide a short writeup that describes a current event article, news story, academic paper, online video etc. that relates to the week’s topic. This write up (about half a page) will be sent to the entire class for their consumption. Interaction is extremely important, and therefore it is worth 8% of your final grade. Interaction is also REQUIRED! Irrespective of your final average, you will fail the course if you do not participate.

The course will be spread out over thirteen (13) weeks and will consist of (2) midterms and one (1) comprehensive final. There will also be 10 assignments. You will need to program in C so get comfortable with it early!

Grading Policy

Assignments:  72% (Best 9 out of 10, 8% each)

Discussions:  8% (Best 10 out of 13, .8% each)

Midterm 1 = 4%

Midterm 2 =  8%
Final = 8%

A if >= 94%, A- if >= 90%
B+ if >= 88%, B if >= 84%, B- if >= 80
C+ if >= 78%, C if >= 70%
F otherwise

Academic Policy: See the “New York University Tandon School of Engineering Student Code of Conduct” (http://engineering.nyu.edu/files/SACCofC2-2-16.pdf) as well as the note at the end of this syllabus.

Additional Note: This is a graduate level class and I will treat you as adults. This means:

1. If I catch you cheating or plagiarising, you and your accomplice(s) will
   1. Fill out and sign out the form at the end of this syllabus
   2. Receive a 0 (zero) in the assignment or exam or course (my discretion)
2. You will not receive extensions, unless you contact me ahead of time and make the proper arrangements
3. You will not be given an opportunity to “boost” your final grade at the end of the semester
4. You are responsible for your grades, so review your returned assignments and exams in a timely manner and contact me if there are any discrepancies
5. You are expected to ask for help when you need it. While I will monitor your progress and let you know if I see any problems, I will not make sure that you do well in the course
6. In turn, I will strive to
   1. return emails and answer appointment requests within two days
   2. grade and return your assignments and exams within two weeks
   3. post new material one week in advance
   4. let you know of any changes to the schedules at least one week in advance
First Pseudo-Assignment: Write out the following agreement **by hand**, sign it, scan it and email it to the instructor.

I (INSERT NAME HERE) understand the stated course policies and agree to abide by them.

I understand that plagiarism is a serious offense and will accept full responsibility for the final grade if I was caught copying, lazy paraphrasing, concatenating quotes or performing similar dishonest actions.

I will not ask the instructor to make grade changes or create special opportunities for me to increase my final grade at the end of the semester.

(SIGN YOUR NAME HERE)

**Course Schedule**

Tentative Syllabus (The assignments and dates might change). The dates posted are for you to watch the pre-recorded videos. We will discuss the topics the following week. For example, Module 1 should be watched the first week, we will discuss during the second live session. You should also be watching the 2nd module during that second week.

Module 1 (23 JAN - 29 JAN): Introduction to Security and Privacy and Security Design Principles + Assignment 1 (Understanding Online Tracking: Play with the “Collusion” plugin and understand modern privacy issues. Objective: Help the student understand the connections between information, security and privacy)

Module 2 (30 JAN - 5 FEB): Threat Modeling + Assignment 2 (Find password crackers and crack a number of available hash databases. Objective: Help students get a sense of the problems with password based authentication)

Module 3 (6 FEB – 12 FEB): Cryptography Fundamentals + Assignment 3 (Understand Counter Mode for Block Ciphers and the SHA3 competition. Objective: Help students understand the tradeoffs between security and performance.)
(13 FEB – 19 FEB): Midterm 1 (Take Home)

Module 4 (20 FEB – 26 FEB): Key Management + Assignment 4 (Understand PKI and its limitations)

Module 5 (27 FEB – 5 MAR): Security Policies

Module 6 (6 MAR – 12 MAR): Access Control 1 – Matrices, ACLs, Linux, Windows and Android + Assignment 5 (Implement a simple reference monitor. Objective: Help students understand why security in practice is hard.)

(13 MAR - 19 MAR): SPRING RECESS

Module 7 (20 MAR – 26 MAR): Access Control 2 – Capability Systems and Information Flow Control + Assignment 6 (Attack a simple reference monitor – Related to Assignment 5. Objective: Helps reinforce the idea that security in practice is hard.)

(27 MAR - 2 APR) Midterm 2 (Take Home)

Module 8 (3 MAR – 9 APR): Containerization: Virtual Machines, Software Fault Isolation and Performance Isolation + Assignment 7 (Understand software fault isolation in practice by looking into Google NaCl. Objective: Help students understand the limitations of sandboxing using software fault isolation)

Module 9 (10 APR – 16 APR): Software Validity and Rights + Assignment 8 (Experiment with the Adore rootkit. Objective: Help students understand what happens to a system when he reference monitor is being bypassed and system integrity has been compromised.)

Module 10 (17 APR – 23 APR): Attacks and Defenses 1: Buffer Overflows, side channels and beyond + Assignment 9 (Conduct a Buffer Overflow. Objective: Help students understand what a buffer overflow vulnerability is, and modern protection mechanisms that are used to limit the impact of such vulnerabilities.)

Module 11 (24 APR – 30 APR): Attacks and Defenses 2: ASLR, Return-Oriented Programming, SQL
Injections + Assignment 10 (Perform a Format String exploit. Objective: Similar to assignment 9, except using format strings)

MAKEUP WEEK (1 MAY - 7 MAY)

(8 MAY - 14 MAY): FINAL (Take Home)

ADA and Students with Disabilities

Please let me know if you have any special requirements. We will work with the Department and University to accommodate you. Please see the Department and University for details.

Notes on Plagiarism for CS6813

I have failed students for plagiarizing. Students have also been expelled because of it.

It disheartens me that plagiarism is still an issue at the graduate level. One would think that graduate students would take pride in their work and strive to better themselves (at least intellectually). This is not always the case. NYU takes plagiarism and cheating seriously and so do I.

Many do not understand that plagiarism is also cheating. Most students seem to understand cheating in the form of multiple students working together and sharing their solutions without prior approval though. Thus, this document focuses on plagiarism.

I, the instructor, have the final say on whether you plagiarized or cheated (the Department and University rarely overrule instructors), I wanted to provide you with examples of plagiarism as I see it. This is NOT an exhaustive list, but should still give you a good sense of what to avoid.

We start with the Engineering school's definition of plagiarism and then continue on with sample answers to the question “What is a buffer overflow?” Finally this document ends with a copy of the letter that you will have to sign if I catch you cheating.

POLICIES AND PROCEDURES ON ACADEMIC MISCONDUCT Section B.3
"Plagiarism: intentionally or knowingly representing the words or ideas of another as one's own in any academic exercise; failure to attribute direct quotations, paraphrases, or borrowed facts or information." (http://engineering.nyu.edu/files/SACofC2-2-16.pdf . Retrieved 17 FEB 2016)

NOTE: The above definition of plagiarism contains a ; (semicolon) which should be interpreted as an “or”.

Example 1 - Direct copy and paste

In computer security and programming, a buffer overflow, or buffer overrun, is an anomaly where a program, while writing data to a buffer, overruns the buffer’s boundary and overwrites adjacent memory locations.

A student who submits the sentence above is clearly plagiarizing because it was copied and pasted directly from the Wikipedia article on buffer overflows (https://en.wikipedia.org/wiki/Buffer_overflow Retrieved 17 FEB 2016) without properly citing the source.

NOTE: The student will receive zero points for the question even if he cited the source properly. That is, while he is not plagiarizing, he also failed to demonstrate a mastery of the material.

Example 2 - Lazy Paraphrasing

A buffer overflow is problem where a program, while writing data to a buffer, overruns the buffer's boundary and overwrites adjacent memory.

In this second example, the student paraphrased the quote from Wikipedia and therefore no longer needs the citation. She failed at paraphrasing though. The student's sentence shares the same structure and essential contents as the original source. Notice how only some words were removed, and "anomaly" was changed to "problem." This is why I call this “lazy” paraphrasing. I consider this plagiarism.

Example 3 - Quote Concatenation
A buffer overflow, or buffer overrun, is an anomaly where a program, while writing data to a buffer, overruns the buffer's boundary and overwrites adjacent memory locations" (https://en.wikipedia.org/wiki/Buffer_overflow Retrieved 7 SEP 2015). They are separated into stack overflows which "occurs if the stack pointer exceeds the stack bound" (https://en.wikipedia.org/wiki/Stack_overflow Retrieved 7 SEP 2015) and heap overflows which "is a type of buffer overflow that occurs in the heap data area" (https://en.wikipedia.org/wiki/Heap_overflow Retrieved 7 SEP 2015)

In this last example, the student properly cited sources but failed to make any meaningful contributions. To me, the student quickly glued together multiple quotes and presented it as their own work. Additionally, I can imagine the student Googling “buffer overflow” and arriving at the Wikipedia page. The student then quickly took the first sentence from that article and then followed the first couple of links in the “Exploitation” section to arrive at the stack and heap overflow articles where he subsequently also quoted the first sentence.

This is also plagiarism. Remember: while you may argue that this is not plagiarism and I may even agree with you, but the final decision is still mine. I also determine what “meaningful” contributions are.

Example 4 - Borderline Good

[Contents from example 3]. While stack and heap based buffer overflows are the most common, they can actually occur wherever there is a buffer. This includes buffers in the “global” section of a binary for example.

In the example above, I took the entirety of example 3’s answer and added just a couple of sentences. I call this a borderline good example, because the student has now added a meaningful contribution. That is, the student added a new piece of information that was not directly retrieved from the Wikipedia article on buffer overflows and required some understanding and thinking on her part.

While I will accept this above example, you can remove all doubts by rewriting the Example 3 answer as well.
Plagiarism / Cheating Form Example

To Whom It May Concern,

This form certifies that I, NAME (N#), participated in the incident (stated below) during Professor Yan's Summer 2016 course, CS6813, INET Section, Information Security and Privacy. This incident violated the policy of this course (for which I am officially registered), the policy of NYU's Graduate School of Arts and Science, and/or the policy of NYU.

As a result, I have accepted the penalty (also stated below) suggested by Professor Yan.

I understand that if a similar incident happens again, the penalty may be harsher, possibly even resulting in expulsion from NYU.

Incident (example):

1. Copying from another student(s), with regard to all or part of the work I submitted as a homework assignment and/or

2. Permitting another student(s) to copy and submit as a homework assignment, all or part of the work I wrote/created

Penalty (example):

A grade of 0 for Assignment #3, as well as a 0 for the class participation grade in CS6813, INET Section, Information Security and Privacy.

Student Signature: ____________________________________ Date:___________________

Advisor Signature: ________________________________ Date:___________________
Course Title: CS6813 Information, Security and Privacy

Instructor Contact Information

Instructor: Lok Yan

Office Hours: W 9-10PM

Any evening hour by appointment

Email: LokYan@nyu.edu (Always email first!)

Course Goal Statement

The main goal of this course is to prepare the student for future studies in information assurance and computer/network security. This is done by ensuring that students understand the different concepts and terminology used in computer security and related disciplines, as well as are comfortable with extending their knowledge through self-guided reading and studying literature.

Course Description

This course is meant to introduce the student to many different topics in Computer Security. This means that the course is designed for breadth. Select topics are explored in more detail using assignments and the discussions. Thus, by the end of this course, the student should be comfortable with the many terms and concepts in Computer Security from Access Control to Cryptography as well as how they relate to each other, e.g., how to use cryptography to enforce access controls. The student should not expect to become an expert in any of these topics since many of the details are elucidated in the subsequent courses, or through reading the literature. To illustrate this distinction, the student should come out of this course understanding the differences and tradeoffs between secret/symmetric key cryptography and public/asymmetric key cryptography, but will not understand the specifics of different cryptographic algorithms such as AES and RSA until the Modern Cryptography course.

Required Reading

The course materials will consist mainly of Audio and Video clips. The length of each module varies, but the student should expect to spend at least two hours reading, listening and watching the provided materials. Each module will also contain links to additional information that the student is expected to watch/read and use in the discussion boards.

Course Prerequisites

CS-GY 6813 INFO SECURITY & PRIVACY, Section INET https://newclasses.nyu.edu/portal/tool/925c0a6a-abde-468f-a590-4f30ec...
The student is expected to have C programming experience. Furthermore, it is highly suggested that the student also have a background in Python as well as x86 assembly language. Courses such as Operating Systems, Computer Architectures and Discrete Math/Formal Modeling are core CS courses and therefore the student is assumed to have knowledge of those concepts.

Course Objectives

The course objectives are embedded in the schedule below

Interaction Policy

Since this is an online-course you are expected to interact with your classmates and the instructor on a weekly basis. I will be holding live lectures each week whenever possible. The student is expected to attend the live lecture if possible and **participate by asking questions and answering questions** from fellow classmates or the instructor. To reiterate, participation is not just attending the sessions, you must ask and/or answer questions. If the student is unable to attend the live session, then the student is expected to watch the lecture at a later time, as well as provide a short writeup that describes a current event article, news story, academic paper, online video etc. that relates to the week’s topic. This write up (about half a page) will be sent to the entire class for their consumption. Interaction is extremely important, and therefore it is worth 8% of your final grade. Interaction is also REQUIRED! Irrespective of your final average, you will fail the course if you do not participate.

The course will be spread out over thirteen (13) weeks and will consist of (2) midterms and one (1) comprehensive final. There will also be 10 assignments. You will need to program in C so get comfortable with it early!

Grading Policy

Assignments:  72% (Best 9 out of 10, 8% each)

Discussions:  8% (Best 10 out of 13, .8% each)

Midterm 1 = 4%

Midterm 2 =  8%

Final = 8%

A if >= 94%, A- if >= 90%
B+ if >= 88%, B if >= 84%, B- if >= 80
C+ if >= 78%, C if >= 70%
F otherwise

Academic Policy: See the “New York University Tandon School of Engineering Student Code of Conduct” (http://engineering.nyu.edu/files/SACCoFC2-2-16.pdf) as well as the note at the end of this syllabus.

Additional Note: This is a graduate level class and I will treat you as adults. This means:

1. If I catch you cheating or plagiarising, you and your accomplice(s) will
   1. Fill out and sign out the form at the end of this syllabus
   2. Receive a 0 (zero) in the assignment or exam or course (my discretion)
2. You will not receive extensions, unless you contact me ahead of time and make the proper arrangements
3. You will not be given an opportunity to “boost” your final grade at the end of the semester
4. You are responsible for your grades, so review your returned assignments and exams in a timely manner and contact me if there are any discrepancies
5. You are expected to ask for help when you need it. While I will monitor your progress and let you know if I see any problems, I will not make sure that you do well in the course
6. In turn, I will strive to
   1. return emails and answer appointment requests within two days
   2. grade and return your assignments and exams within two weeks
   3. post new material one week in advance
   4. let you know of any changes to the schedules at least one week in advance

First Pseudo-Assignment: Write out the following agreement **by hand**, sign it, scan it and email it to the instructor.

I (INSERT NAME HERE) understand the stated course policies and agree to abide by them.
I understand that plagiarism is a serious offense and will accept full responsibility for the final grade if I was caught copying, lazy paraphrasing, concatenating quotes or performing similar dishonest actions.

I will not ask the instructor to make grade changes or create special opportunities for me to increase my final grade at the end of the semester.

(SIGN YOUR NAME HERE)

Course Schedule

Tentative Syllabus (The assignments and dates might change). You are expected to watch the pre-recorded videos for each Module 1 week prior (e.g. Module 1 should be completed 6-10 SEP so we can discuss them the week of 11-17 SEP)

Module 0 (6 SEP - 10 SEP): Quick Overview & Introduction to Course Policies

Module 1 (11 SEP – 17 SEP): Introduction to Security and Privacy and Security Design Principles + Assignment 1 (Understanding The Pervasiveness of Security. Objective: Help the student understand the connections between information, security and privacy as well as how it touches every aspect of our lives.)

Module 2 (18 SEP – 24 SEP): Threat Modeling + Assignment 2 (Find password crackers and crack a number of available hash databases. Objective: Help students get a sense of the problems with password based authentication)

Module 3 (25 SEP – 1 OCT): Cryptography Fundamentals + Assignment 3 (Understand Counter Mode for Block Ciphers and the SHA3 competition. Objective: Help students understand the tradeoffs between security and performance.)

Module 4 (2 OCT – 8 OCT): MIDTERM 1 (Take Home) + Key Management

Module 6 (16 OCT – 22 OCT): Access Control 1 – Matrices, ACLs, Linux, Windows and Android + Assignment 5 (Implement a simple reference monitor. Objective: Help students understand why security in practice is hard.)

Module 7 (23 OCT – 29 OCT): Access Control 2 – Capability Systems and Information Flow Control + Assignment 6 (Attack a simple reference monitor – Related to Assignment 5. Objective: Helps reinforce the idea that security in practice is hard.)

(30 OCT – 5 NOV): Midterm 2 (Take Home)

Module 8 (6 NOV – 12 NOV): Containerization: Virtual Machines, Software Fault Isolation and Performance Isolation + Assignment 7 (Understand software fault isolation in practice by looking into Google NaCl. Objective: Help students understand the limitations of sandboxing using software fault isolation)

Module 9 (13 NOV – 19 NOV): Software Validity and Rights + Assignment 8 (Experiment with the Adore rootkit. Objective: Help students understand what happens to a system when he reference monitor is being bypassed and system integrity has been compromised.)

THANKSGIVING BREAK (20 NOV - 26 NOV)

Module 10 (29 NOV – 3 DEC): Attacks and Defenses 1: Buffer Overflows, side channels and beyond + Assignment 9 (Conduct a Buffer Overflow. Objective: Help students understand what a buffer overflow vulnerability is, and modern protection mechanisms that are used to limit the impact of such vulnerabilities.)

Module 11 (4 DEC – 10 DEC): Attacks and Defenses 2: ASLR, Return-Oriented Programming, SQL Injections + Assignment 10 (Perform a Format String exploit. Objective: Similar to assignment 9, except using format strings)

MAKEUP WEEK (11 DEC - 17 DEC)
ADA and Students with Disabilities

Please let me know if you have any special requirements. We will work with the Department and University to accommodate you. Please see the Department and University for details.

Notes on Plagiarism for CS6813

I have failed students for plagiarizing. Students have also been expelled because of it.

It disheartens me that plagiarism is still an issue at the graduate level. One would think that graduate students would take pride in their work and strive to better themselves (at least intellectually). This is not always the case. NYU takes plagiarism and cheating seriously and so do I.

Many do not understand that plagiarism is also cheating. Most students seem to understand cheating in the form of multiple students working together and sharing their solutions without prior approval though. Thus, this document focuses on plagiarism.

I, the instructor, have the final say on whether you plagiarized or cheated (the Department and University rarely overrule instructors), I wanted to provide you with examples of plagiarism as I see it. This is NOT an exhaustive list, but should still give you a good sense of what to avoid.

We start with the Engineering school's definition of plagiarism and then continue on with sample answers to the question “What is a buffer overflow?” Finally this document ends with a copy of the letter that you will have to sign if I catch you cheating.

POLICIES AND PROCEDURES ON ACADEMIC MISCONDUCT Section B.3

"Plagiarism: intentionally or knowingly representing the words or ideas of another as one’s own in any academic exercise; failure to attribute direct quotations, paraphrases, or borrowed facts or information." (http://engineering.nyu.edu/files/SACCofoC2-2-16.pdf. Retrieved 17 FEB 2016)

NOTE: The above definition of plagiarism contains a ; (semicolon) which should be interpreted as
Example 1 - Direct copy and paste

In computer security and programming, a buffer overflow, or buffer overrun, is an anomaly where a program, while writing data to a buffer, overruns the buffer's boundary and overwrites adjacent memory locations.

A student who submits the sentence above is clearly plagiarizing because it was copied and pasted directly from the Wikipedia article on buffer overflows (https://en.wikipedia.org/wiki/Buffer_overflow Retrieved 17 FEB 2016) without properly citing the source.

NOTE: The student will receive zero points for the question even if he cited the source properly. That is, while he is not plagiarizing, he also failed to demonstrate a mastery of the material.

Example 2 - Lazy Paraphrasing

A buffer overflow is problem where a program, while writing data to a buffer, overruns the buffer's boundary and overwrites adjacent memory.

In this second example, the student paraphrased the quote from Wikipedia and therefore no longer needs the citation. She failed at paraphrasing though. The student's sentence shares the same structure and essential contents as the original source. Notice how only some words were removed, and "anomaly" was changed to "problem." This is why I call this "lazy" paraphrasing. I consider this plagiarism.

Example 3 - Quote Concatenation

“A buffer overflow, or buffer overrun, is an anomaly where a program, while writing data to a buffer, overruns the buffer's boundary and overwrites adjacent memory locations” (https://en.wikipedia.org/wiki/Buffer_overflow Retrieved 7 SEP 2015). They are separated into stack overflows which "occurs if the stack pointer exceeds the stack bound" (https://en.wikipedia.org
In this last example, the student properly cited sources but failed to make any meaningful contributions. To me, the student quickly glued together multiple quotes and presented it as their own work. Additionally, I can imagine the student Googling “buffer overflow” and arriving at the Wikipedia page. The student then quickly took the first sentence from that article and then followed the first couple of links in the “Exploitation” section to arrive at the stack and heap overflow articles where he subsequently also quoted the first sentence.

This is also plagiarism. Remember: while you may argue that this is not plagiarism and I may even agree with you, but the final decision is still mine. I also determine what “meaningful” contributions are.

Example 4 - Borderline Good

[Contents from example 3]. While stack and heap based buffer overflows are the most common, they can actually occur wherever there is a buffer. This includes buffers in the “global” section of a binary for example.

In the example above, I took the entirety of example 3’s answer and added just a couple of sentences. I call this a borderline good example, because the student has now added a meaningful contribution. That is, the student added a new piece of information that was not directly retrieved from the Wikipedia article on buffer overflows and required some understanding and thinking on her part.

While I will accept this above example, you can remove all doubts by rewriting the Example 3 answer as well.

Plagiarism / Cheating Form Example

To Whom It May Concern,
This form certifies that I, NAME (N#), participated in the incident (stated below) during Professor Yan's Summer 2016 course, CS6813, INET Section, Information Security and Privacy. This incident violated the policy of this course (for which I am officially registered), the policy of NYU's Graduate School of Arts and Science, and/or the policy of NYU.

As a result, I have accepted the penalty (also stated below) suggested by Professor Yan.

I understand that if a similar incident happens again, the penalty may be harsher, possibly even resulting in expulsion from NYU.

Incident (example):

1. Copying from another student(s), with regard to all or part of the work I submitted as a homework assignment and/or

2. Permitting another student(s) to copy and submit as a homework assignment, all or part of the work I wrote/created

Penalty (example):

A grade of 0 for Assignment #3, as well as a 0 for the class participation grade in CS6813, INET Section, Information Security and Privacy.

Student Signature: ________________________________ Date:___________________

Advisor Signature: ________________________________ Date:___________________

Fall 2017 Syllabus.pdf