I have read and understood the policy on academic dishonesty ("cheating") as outlined in the syllabus for the Fall 2015 Memory Analysis course. In particular, I understand that copying or providing to other students, in whole or in part, solutions (text *or* source code) to class assignments from *any* source (including work done by former and current CSCI students, other humans, animals, zombies, materials downloaded from the Internet, etc.) not directly sanctioned by Prof. Richard is **not acceptable**. I understand that all work must be exclusively my own, with the exception of any team projects, for which I am allowed to collaborate with my assigned partner(s).

THERE IS NO FLEXIBILITY IN THIS POLICY. IF YOU CHEAT, YOU FAIL, AND YOUR ACADEMIC CAREER IS LIKELY TO BE PREMATURELY TERMINATED. THE "REASON" YOU CHEATED IS IMMATERIAL. This policy applies equally to students "transmitting" or "receiving" answers.

Print your name: _____

Sign your name: _____

Date: _____

No grades will be assigned to your work until you sign and hand in this agreement.

Once you understand the above, please take the following survey (and be honest—this will be used to "tune" the class):

Survey: On a scale of 0 ("what is it?") to 10 ("I'm a black belt in this topic, and could kill you with a single glance…") please rate your experience with:

General familiarity with Linux/Unix: _____

General familiarity with Windows: _____

General familiarity with Mac OS X: _____

Operating systems internals (any of the above): _____

C: _____

Python: _____

Intel assembly language: _____

Malware / reverse engineering: _____

UNIVERSITY *of*
NEW ORLEANS

DEPARTMENT OF COMPUTER SCIENCE

CSCI 6623: Memory Forensics
Fall 2015 Syllabus
Prof. Golden G. Richard III

Me:

Office	☞	GNOCIA Director's Office (Math 329c)
Phone	☞	504-280-6045
Email	☞	*golden@cs.uno.edu*
Office Hours	☞	4-6pm on M/W, 2-3pm on F, or by appointment

You: Credit in CSCI 4401 or a similar operating systems course, CSCI 4623 (introduction to digital forensics), significant interest in computer security and operating systems internals, and significant experience programming in C. You will also need to learn Python in this course, but no previous Python experience is expected. This is **not** an introductory course in digital forensics!

Meeting: Section 601 ☞ 6-7:15pm on M/W in the NSSAL (Math 322).

Textbooks: The following book is required—plan on reading **all** of it this semester:

The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory by Michael Hale Ligh, Andrew Case, Jamie Levy, and Aaron Walters (Wiley, 2014)

You will also need a copy of the Intel Developer's manuals. These are free and can be downloaded via http://www.intel.com/products/processor/manuals/

Grading:

Midterm Examination	☞	35%
Final Examination	☞	35% (comprehensive)
Laboratory Assignments and Semester Project	☞	30%

More Details

BACKGROUND: Traditional digital forensics investigations focus on the contents of non-volatile storage devices—hard drives, CDs, backup tapes, etc. In the last ten years, there has been increasing interest, both in the cybersecurity research community and among practitioners, in *memory forensics*, which involves deep investigation of the contents of volatile computer memory (RAM). Careful examination of RAM can reveal hidden processes, network connections, clipboard contents, evidence of malware, and a wealth of other important evidence. This course provides a solid foundation in memory forensics, based primarily on use of the Volatility memory forensics framework, an open source toolset written in Python. The course ultimately requires you to develop significant skills in operating systems internals (Mac, Windows, Linux), since memory forensics concentrates on the data structures used internally by operating systems (as well as some userspace applications). You'll also need to develop Python programming skills, since the entire Volatility framework is written in Python.

ASSIGNMENTS: There will be a number of laboratory assignments in this course. There are dedicated machines in the Networking, Security, and Systems Administration Laboratory (NSSAL) in Math 322 for your use. It's also possible to work on your own hardware, provided you install Volatility and virtualization software. We'll discuss this further in class at a later date. There will also be a significant, semester-long project, involving creation of a new plugin for Volatility. This will also be discussed later in the semester. You should consider the due date for each assignment to be a hard deadline. When the due date arrives, turn in what you have—I do give partial credit, but…

NO LATE SUBMISSIONS WILL BE ACCEPTED. ANY SOLUTION SUBMITTED AFTER THE DUE DATE WILL RECEIVE A GRADE OF ZERO.

Submission procedures will be discussed in class.

EXAMS: The final examination is comprehensive with an emphasis on material after the midterm. Any missed test will receive a grade of zero unless arrangements are made with me. Both the midterm and final are closed book, closed notes. **The tests will be a mix of basic concepts and questions related to your laboratory assignments. Take careful notes when you are in the laboratory and study these notes before the examinations. Be prepared to explain techniques you used in the laboratory on the exams.**

CHEATING: Don't cheat. If you do, I will catch you, and I will pursue the harshest possible penalties. All submitted work must be exclusively your own. Cheating is:

- Copying, in whole or in part, the solutions of former students, current students, or any other living being, alive or dead. "Copying" includes transmission through email, port knocking, the Web, smoke signals, ESP, steganography, or any other means.
- Obtaining solutions from the Internet or other any archival source.
- Even *looking* at a solution is cheating. If you see something that looks like a solution to a class assignment, avert your eyes and *run away as fast as you can.*

The official UNO policy statement on academic integrity states:

Students are expected to conduct themselves according to the principles of academic integrity as defined in the statement on Academic Dishonesty in the UNO Student Code of Conduct. Any student or

group found to have committed an act of academic dishonesty shall have their case turned over to the Office of Student Accountability and Advocacy for disciplinary action which may result in penalties as severe as indefinite suspension from the University. Academic dishonesty includes, butis not limited to: cheating, plagiarism, fabrication, or misrepresentation, and being an accessory to an act of academic dishonesty.

Discussing assignments at a high level for clarification, discussing problems concerning the computing equipment, and studying in groups for examinations is not cheating, but every character you type for laboratory assignments, written assignments, and the examinations had better be your own!

STUDENTS WITH DISABILITIES:

The official UNO policy for handling disabilities that may impact your ability to excel in UNO courses is:

It is University policy to provide, on a flexible and individualized basis, reasonable accommodations to students who have disabilities that may affect their ability to participate in course activities or to meet course requirements. Students who seek accommodations for disabilities must contact the Office of Disability Services prior to discussing their individual needs for accommodation with their instructors.

GRADING SCALE: The following grading scale is used. I never curve. Grading in college courses is objective—please don't ask me to change your grade on an assignment unless you clearly deserve it and can demonstrate that this is the case.

A	**90-100**	**B**	**80-89**	**C**	**70-79**
D	**60-69**	**F**	**0-59**		

CLASS MATERIALS: via Moodle (http://moodle.uno.edu)

SLIDES: Lecture slides are available via Moodle (see above). Please try to view the slides online as much as possible and avoid printing them! The remaining trees will love you.

Topics and Reading

Topic	Starting Slide	AMF
Introduction / History	Part I:1	Chapters 1-3
Introduction to volshell	Part I:57	---
Memory Acquisition	Part I:85	Chapters 4, 19, 28
Address Translation	Part I:126	Chapter 1
Basic Memory Forensics Internals	Part I:140	Chapters 2-3
eBlaster Case Study	Part I:182	---
Windows Process Representation	Part I:197	Chapter 6
Linux Process Representation	Part II:2	Chapter 21
Mac Process Representation	Part II:27	Chapter 29
--- Lab 1---		
Windows Memory Allocation	Part II:39	Chapter 5, 7
Linux Memory Allocation	Part II:94	Chapters 21, 23
---Lab 2---		
Mac OS X Memory Allocation	Part II:169	Chapter 29
Swap Files and Compressed RAM	Part II:175	DFRWS 2014 paper
---Lab 3---		

Topic	Starting Slide	AMF
Windows GUI Subsystem Forensics	Part III:2	Chapters 14-15
Windows Event Logs	Part III:38	Chapter 9
Windows Registry Forensics	Part III:47	Chapter 10
Windows Network Forensics	Part III:84	Chapter 11
Linux Network Forensics	Part III:97	Chapter 22
---Lab # 4---		
Packed Malware	Part III:106	---
YARA Overview	Part III:127	---
Windows Malware Detection and Analysis	Part III:158	Chapter 8
---Lab # 5---		
Linux Malware Detection and Analysis	Part IV:2	Chapter 25
Mac Malware Detection and Analysis	Part IV:40	Chapter 30
Deeper Windows Kernel Forensics	Part IV:50	Chapter 13
Deeper Linux Kernel Forensics	Part IV:103	Chapter 26, 27
---Lab # 6---		
Deeper Mac Kernel Forensics	Part IV:165	Chapter 30
Intro to Application Memory Forensics	Part IV:180	Chapter 31
Non-traditional Malware	Part IV:198	****