UMBC UGC New Course Request: CMSC449 – Malware Analysis

Date Submitted: 2/14/2020 Proposed Effective Date: 8/23/2020

	Name	Email	Phone	Dept
Dept Chair or UPD	Jeremy Dixon	jdixon@umbc.edu	5-8866	CSEE
Other Contact	Mohamed Younis	younis@umbc.edu	5-3968	CSEE
Other Contact	Charles Nicholas	nicholas@umbc.edu	5-2594	CSEE

COURSE INFORMATION:

Course Number(s)	CMSC449
Formal Title	Malware Analysis
Transcript Title (≤30c)	Malware Analysis
Recommended Course Preparation	
Prerequisite :: Unless otherwise indicated, a prerequisite is assumed to be passed with a "D" or better.	CMSC313 and CMSC 341 both with a "C" or better
# of Credits Must adhere to the UMBC Credit Hour Policy	3
Repeatable for additional credit?	Yes ☒ **No**
Max. Total Credits	3 This should be equal to the number of credits for courses that cannot be repeated for credit. For courses that may be repeated for credit, enter the maximum total number of credits a student can receive from this course. E.g., enter 6 credits for a 3 credit course that may be taken a second time for credit, but not for a third time. Please note that this does NOT refer to how many times a class may be retaken for a higher grade.
Grading Method(s)	☒ Reg (A-F) ☒ Audit ☒ Pass-Fail

PROPOSED CATALOG DESCRIPTION (Approximately 75 words in length. Please use full sentences.):

Malicious software (malware) is a constant threat to the information and intellectual property of organizations. By analyzing malware using both static and dynamic methods, students will be introduced to these increasingly sophisticated attacks. This course will provide a foundation for understanding emerging trends in malware design, including efforts to deter analysis. Discussions and hands-on exercises will cover object file formats, and the use of tools such as debuggers, virtual machines, and disassemblers. Finally, obfuscation and packing schemes will be discussed, along with various issues related to operating systems internals.

RATIONALE FOR NEW COURSE:

a) Why is there a need for this course at this time?
We are continuing to grow our offerings in the field of cybersecurity as the demand continues to grow. We have offered this course several times previously as a special topics course with great success.

b) How often is the course likely to be taught?
We hope to offer it every semester if possible but at least once per year.

c) How does this course fit into your department's curriculum?
It fits well as an elective for all CS majors or as a course in our cyber security track.

d) What primary student population will the course serve?

e) Why is the course offered at the level (ie. 100, 200, 300, or 400 level) chosen?
It builds on many concepts in CMSC 313. As such, it is perfect as a 400 level course.

f) Explain the appropriateness of the recommended course preparation(s) and prerequisite(s).
This course builds on concepts in CMSC 313 which has prerequisites of CMSC 201, CMSC 202, and CMSC 203. Therefore, students should be familiar with programming in C++ and assembly.

g) Explain the reasoning behind the P/F or regular grading method.
Students can take this course for a letter grade, P/F, or audit although we continually have problems with large waitlists.

h) Provide a justification for the repeatability of the course.
This course cannot be repeated.

ATTACH COURSE SYLLABUS (mandatory):

UMBC
AN HONORS UNIVERSITY

CMSC 449: Malware Analysis

Prerequisites:

CMSC 313 and CMSC 341 each with a C or better.

Instructor:

Name: TBD
Office: TBD
Office Hours: TBD
Phone: TBD
Email: TBD

Course Description:

Malicious software (malware) is a constant threat to the information and intellectual property of organizations. By analyzing malware using both static and dynamic methods, students will be introduced to these increasingly sophisticated attacks. This course will provide a foundation for understanding emerging trends in malware design, including efforts to deter analysis. Discussions and hands-on exercises will cover object file formats, and the use of tools such as debuggers, virtual machines, and disassemblers. Finally, obfuscation and packing schemes will be discussed, along with various issues related to operating systems internals.

Credits:

Three credits: not repeatable

Learning Outcomes:

At the end of the course, the student will:

- Analyze modern malware samples using both static and dynamic analysis techniques.
- Understand a variety of executable formats including platform-specific executables (binaries) and APIs.
- Examine object file formats, and how to examine those files using tools such as debuggers, virtual machines, and disassemblers.
- Apply techniques and concepts to unpack, extract, decrypt, or bypass new anti-analysis techniques in future malware samples.

Course Format and Procedures:

As reverse engineering malware involves a deep analysis of the code, structure, and functionality, the goal of this class is to give students a hands-on introduction to static and dynamic malware analysis. A variety of tools such as host- and web-based are presented at both the basic and advanced levels. Additional utilities will be examined to provide summary information as well as disassemblers and debuggers. Most of the course content will emphasize realistic malware specimens from traditional sources such as textbooks, articles, or governmental releases. Homework assignments and exams consist of preparation of malware analysis reports such as those used in industry.

Required Textbook:

Sikorski, M., & Honig, A. (March 3, 2012). Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software. San Francisco, CA: No Starch Press. (ISBN-13: 978-1593272906, ISBN-10: 1593272901).

Readings:

Ligh, M., Adair, S., Harstein B., & Richard, M. (2010). Malware Analyst's Cookbook and DVD. Wiley. (ISBN-13: 978-0470613030).

Eilam, E. (2005). Reversing: Secrets of Reverse Engineering. Wiley. (ISBN-13: 978-0764574818).

Yosifovich, P., Ionescu, A., Russinovich, M. E., & Solomon, D. A. (2017). Windows Internals – Part 1. Redmond, WA. Microsoft Press. (ISBN-13: 978-0735684188).

Russinovich, M. E., Solomon, D. A., Ionescu, A., & Allievi, A. (2019). Windows Internals – Part 2. Redmond, WA. Microsoft Press. (ISBN-13: 978-0135462409).

Course Topics:

Students in Malware Analysis will participate by:

- Introduction: Introduction to malware analysis and setting up your environment.
- Using virtual machines: Using virtual environments to examine and analyze malware safely.
- Static analysis: Analyzing file properties, metadata, strings, and disassembled code.
- Dynamic analysis: Using tools such as sandboxes, packet analyzers (such as Wireshark), and debuggers (such as Immunity Debug).
- Malware analysis: launching, encoding, and network signatures of malware.
- Ethical considerations: White hat vs. Black hat. Identifying global implications of malware.

Grading:

Description	Number	Points	Total
Homework Assignments	6	11	66
Exam 1	1	14	14
Final Exam	1	20	20

Grading is on a standard 100-point scale, so you will get an A for 90.0 or more total points, a B for 80.0 or more but less than 90.0 points, and so on.

The homework assignments will be a blend of practical exercises and questions that cement conceptual knowledge. The homework assignments will be a combination of research activities, programming activities, reverse engineering exercises, and generating malware analysis reports. The two exams are take home and will require significant analysis and may require code-based solutions.

Academic Integrity

By enrolling in this course, each student assumes the responsibilities of an active participant in UMBC's scholarly community in which everyone's academic work and behavior are held to the highest standards of honesty. Cheating, fabrication, plagiarism, and helping others to commit these acts are all forms of academic dishonesty, and they are wrong. Academic misconduct could result in disciplinary action that may include, but is not limited to, suspension or dismissal. To read the full Student Academic Conduct Policy, consult the Academic Integrity Resources for Students page (https://aetp.umbc.edu/ai/resources-for-students/) or the Faculty Handbook (http://provost.umbc.edu/faculty-handbook/), specifically Sections 14.2-14.3.

If you need help with a project, see your instructor, your TA, or tutors provided by the Learning Resource Center. We also encourage you to consult textbooks and code examples provided on Blackboard. Consult Blackboard for additional Academic Integrity policies for projects.

Any act of dishonesty will be reported to the University's Academic Conduct Committee for further action, which may include, but is not limited to, academic suspension or dismissal from the University.

We will be using special software to check for cheating. The software is quite sophisticated and has surprised many students in the past. There is no difficulty in comparing every pair of assignments – even assignments submitted to other sections of this course, or from previous semesters.

This is a *non-exhaustive* list of restrictions for completing your assignments in this course.

- **If you have questions about what is acceptable, please contact a professor or TA.**

You may not look at, access, download, or obtain anyone else's work.

- You should think carefully about the assignment, and the assignment you turn in should be entirely a product of your own understanding of the material.

- You may not use any online resources to request additional help. Please contact a professor or TA for additional help.
- You may not post any part of a course document online. Posting any slides, projects, or labs will be considered a violation of this course policy and will result in an "F" for the course.
- You may not look at someone else's code "for reference," even if you put it aside before programming, and even if that person is not a CMSC student.
- You may not Google or search for the solution to an assignment, even if it's "only for reference."
- You may not copy code other than that provided in the course materials (slides, book, labs, etc.).
- You may not let someone else explain a solution to you in such detail that they are effectively dictating the code to you line by line. It does not matter if this person has never taken this course, or if they are not looking at their own code while doing so!

Student Disability Services:

UMBC is committed to eliminating discriminatory obstacles that may disadvantage students based on disability. Services for students with disabilities are provided for all students qualified under the Americans with Disabilities Act (ADA) of 1990, the ADAAA of 2009, and Section 504 of the Rehabilitation Act who request and are eligible for accommodations. The Office of Student Disability Services (SDS) is the UMBC department designated to coordinate accommodations that would allow students to have equal access and inclusion in all courses, programs, and activities at the University.

If you have a documented disability and need to request academic accommodations for access to your courses, please refer to the SDS website at sds.umbc.edu for registration information and to begin the process, or alternatively you may visit the SDS office in the Math/Psychology Building, Room 212. For questions or concerns, you may contact us through email at disAbility@umbc.edu or phone (410) 455-2459.

If you require accommodations for this class, make an appointment to meet with your instructor to discuss your SDS-approved accommodations.

Disclosures of Sexual Misconduct and Child Abuse or Neglect

As an instructor, I am considered a Responsible Employee, per UMBC's Policy on Prohibited Sexual Misconduct, Interpersonal Violence, and Other Related Misconduct (located at http://humanrelations.umbc.edu/sexual-misconduct/umbc-resource-page-for-sexual-misconduct-and-other-related-misconduct/). While my goal is for you to be able to share information related to your life experiences through discussion and written work, I want to be transparent that as a Responsible Employee I am required to report disclosures of sexual assault, domestic violence, relationship violence, stalking, and/or gender-based harassment to the University's Title IX Coordinator. As an instructor, I also have a mandatory obligation to report disclosures of or suspected instances of child abuse or neglect (www.usmh.usmd.edu/regents/bylaws/SectionVI/VI150.pdf).

The purpose of these reporting requirements is for the University to inform you of options, supports and resources; you will not be forced to file a report with the police. Further, you can receive support and resources, even if you choose to not want any action taken. Please note that in certain situations, based on the nature of the disclosure, the University may need to act.

If you need to speak with someone in confidence about an incident, UMBC has the following Confidential Resources available to support you:
The Counseling Center: 410-455-2472
University Health Services: 410-455-2542
(After-hours counseling and care available by calling campus police at 410-455-5555)

Other on-campus supports and resources:
The Women's Center, 410-455-2714
Title IX Coordinator, 410-455-1606
Additional on and off campus supports and resources can be found at:
http://humanrelations.umbc.edu/sexual-misconduct/gender-equitytitle-ix/

Tentative Schedule:

Date	Topic	Assignments	Readings	Notes
1/27/2020	Introduction	Install Flare VM (~18gigs)		
1/29/2020	Virtual Machines	HW1 Out	Basic Static Analysis	Malware Research Group meets Fridays 2-3pm, ITE 366, starting Friday January 31!
2/3/2020	Basic Tools		Basic Static Analysis	
2/5/2020	Packing and Unpacking	HW1 Due	VMs and Sandboxes	The UMBC Cyberdefense Team, aka the Cyberdawgs, will be meeting after this class, on Wednesdays through the semester! The location is ITE 237.
2/10/2020	Configuring Virtual Machines	HW2 Out		Malware analysts should know Python, since the pefile module in Python can be used to make lots of useful tools.
2/12/2020	Basic Dynamic Analysis		Triage vs. in-depth analysis	McAfee CTO Mike Fey has started a series of blog posts entitled Seven Myths on Advanced Malware
2/17/2020	Registry			A sandbox at http://www.hybrid-analysis.com
2/19/2020	Dynamic Analysis		A sandbox at http://www.hybrid-analysis.com	PMA recommends ApateDNS for monitoring DNS requests. The Mandiant tool ApateDNS requires .NET 3.5, which you can get here
2/24/2020	Dynamic Analysis			Look at Norman Sandbox
2/26/2020	Dynamic Analysis	HW2 Due	Book - Chapter 3	Running DLL with rundl32.exe
3/2/2020	Guest Speakers	HW3 Out		Ryan Warns and Chris Gardner from Fireye
3/4/2020	Review x86 Assembly		Book - Chapter 4	
3/9/2020	Introduction to IDA Pro		Book - Chapter 5	Demo of IDA Pro, disassembler and malware analysis tool
3/11/2020	IDA Pro		Book - Chapter 6	Here is a malware example, as a password-protected zipfile (zip) with password "malware" without the quotes
3/16/2020	Spring Break			No Class
3/18/2020	Spring Break			No Class
3/23/2020	Control Structures in Malware	HW3 Due HW4 Out		
3/25/2020	Debugging		Book - Chapter 8	
3/30/2020	Analyzing Malicious Windows Programs	HW4 Due	Book - Chapter 7	Exam and Project Discussed
4/1/2020	Debugging	Exam 1 Out (take home)	Book - Chapter 7	Peter Drucker's article "Managing Oneself" appeared in the January, 2005 issue of Harvard Business Review. The paper is not being assigned as part of this course, but if you as an authorized UMBC library patron and wish to read it, here it is. The link is supposed to work from a UMBC IP address only.
4/6/2020	Exam			Use classtime to work on exam
4/8/2020	ImmDbg	Exam 1 Due		Finish demo of Immunity Debugger
4/13/2020	Malware Behavior		Book - Chapter 11	Demo Lab 9-2
4/15/2020	Android Malware	HW 5 Out	Book - Chapter 18	
4/20/2020	Malware Behavior		Book - Chapter 12	Covert Malware Launching
4/22/2020	Packing and Unpacking		Book - Chapter 18	Sorokin's paper on structural entropy (pdf)
4/27/2020	Malware Behavior		Book - Chapter 12	Report from FireEye
4/29/2020	YARA	HW 5 Due	Download Yara from Github	YARA is a tool aimed at (but not limited to) helping malware researchers to identify and classify malware samples.
5/4/2020	Data Encoding	HW 6 Out	Book - Chapter 13	Release the YARA Homework. The docx file, and the malware specimen 7z.
5/6/2020	Guest Speaker			Dr. Robert Brandon
5/11/2020	Malware on UNIX		Book - Chapters 17, 19, and 20	Memory forensics using Volatility
5/13/2020	Anti-Debugging	HW 6 Due	Book - Chapters 17, 19, and 20	Overview of malware on a Mac.
5/18/2020	Review	Final Exam Out Due on 5/21/2020		Review of materials for take home exam.

This schedule is subject to change without notification from the professor.