CS-5375: Software Reverse Engineering
Fall 2025

Mohammad Saidur Rahman, Ph.D.
msrahman3@utep.edu
https://www.rahmanmsaidur.com
Department of Computer Science
University of Texas at El Paso (UTEP)

Logistics

- Course Website: https://sre.rahmanmsaidur.com/
- Lectures: Class: 40% lectures, 60% in-class activity.
- Mode of Instruction: In-Person
- Class Schedule: Wednesdays 6:00PM - 8:50PM Mountain Time (MST)
- Location: College of Business Admin 301
- Office hours: Thursdays 03.00PM - 04.00 PM Mountain Time (MST)
- Contact: In addition to the office hour meeting, students are encouraged to ask all course-related questions on our *Microsoft Teams Channel* (Team Code: ask instructor) so that your peers can benefit from the answers. If you have a specific question that requires privacy, you may *email* directly.
- Academic accommodations: If you need an academic accommodation based on a disability, you should initiate the request with the Center for Accommodations and Support Services (CASS). The CASS will evaluate the request and recommend accommodations. Students should contact the CASS as soon as possible since timely notice is needed to coordinate accommodations.
  - *CASS-approved students are granted a 3-day extension for any submissions that are individual tasks.*
- Honor code and Academic Integrity: Very very important. See Section 9.
- **Important Dates**:
  - Late Registration: August 25-29, 2025
  - Census Day: September 10, 2025
  - Fall Course Drop/Withdrawal Deadline: October 31, 2025
    - *Withdrawal after this date has to go through UTEP Withdrawal Committee*
  - Last Day of Class: December 4, 2025

1 Course Overview

This graduate-level course integrates hands-on training with research-driven exploration to prepare students as future experts in software reverse engineering and malware analysis. Students will gain practical expertise in uncovering malicious behavior on compromised systems and analyzing the techniques adversaries employ to evade detection. The course features extensive use of industry-standard tools such as Wireshark, Ghidra, IDA Pro, Dshell, Cuckoo Sandbox, Volatility, Metasploit Framework, Armitage, and Google Rapid Response. In addition to classical methods, students will engage with emerging approaches in automated reverse engineering, including large language model (LLM)-assisted techniques. The course also introduces cutting-edge strategies for red teaming Artificial Intelligence (AI) systems, focusing on black-box testing and adversarial evaluation.

Finally, the course emphasizes research engagement by surveying state-of-the-art developments in malware analysis and machine learning for intelligent malware classification and detection. Students will critically analyze current literature, identify open challenges, and design solutions that advance the field of reverse engineering and cybersecurity.

Why Software Reverse Engineering (SRE)?

SRE involves deconstructing software to understand its components, functionality, and behavior without access to the original source code. As cyber threats become more sophisticated, SRE provides the tools needed to effectively dissect and mitigate cyber threats, ensuring the integrity and security of software systems across various industries. SRE is essential for identifying security vulnerabilities, understanding malware, ensuring compliance, and recovering lost or legacy code.

- **The Perfect Time to Study Automated Reverse Engineering with LLM and AI.** The rapid advancements in artificial intelligence (AI) and large language models (LLMs) have revolutionized many fields, including software reverse engineering. Automated reverse engineering powered by AI and LLMs can significantly enhance the efficiency and accuracy of analyzing complex software. These technologies can automate repetitive tasks, identify patterns, and provide insights that might be missed by human analysts. Now is the perfect time to study these advanced techniques because the tools and resources are more accessible than ever, and the demand for professionals skilled in automated reverse engineering is on the rise. As AI and LLMs continue to evolve, their applications in reverse engineering will become even more integral, making early adopters of these technologies invaluable assets to their organizations.

- **The Crucial Role of AI Red Teaming.** AI red teaming is an emerging field that focuses on testing and evaluating the robustness of AI systems. As AI becomes increasingly integrated into critical infrastructure and decision-making processes, ensuring its security and reliability is paramount. AI red teaming involves simulating attacks on AI systems to identify weaknesses and improve their defenses. This is crucial now more than ever because AI systems are being deployed in high-stakes environments where failures or vulnerabilities can have significant consequences. By studying AI red teaming, students can learn how to anticipate and mitigate potential threats to AI systems, making them more resilient and trustworthy.

- **The Importance of Studying State-of-the-Art Malware Analysis Research** Malware is continuously evolving, with new variants and sophisticated attack techniques emerging regularly. Staying ahead of these threats requires a deep understanding of the latest research and developments in malware analysis. By studying state-of-the-art malware analysis research, students can learn about the newest tools, techniques, and methodologies used to detect, analyze, and combat malware. This knowledge is critical for developing effective defense strategies and keeping up with the fast-paced nature of cyber threats. Engaging with cutting-edge research ensures that students are well-prepared to tackle current and future challenges in the field of cybersecurity.

2 Course Objectives

This course is designed to:

- **Develop Expertise in Malware Analysis:** Train students to analyze and interpret malicious behavior on compromised systems using industry-standard tools such as Wireshark, Ghidra, IDA Pro, and Cuckoo Sandbox.
- **Advance Automated Reverse Engineering Skills:** Introduce state-of-the-art methods that leverage large language models (LLMs) and AI to automate software deconstruction and analysis.
- **Explore AI Red Teaming Practices:** Equip students with methodologies to conduct red teaming against AI systems in black-box settings, simulating adversarial threats to assess robustness.
- **Engage with Research Frontiers:** Encourage critical engagement with current research in malware analysis and machine learning to identify open challenges and contribute to innovative detection and classification methods.
- **Build a Comprehensive Foundation in Software Reverse Engineering (SRE):** Provide a systematic understanding of SRE principles and practices for uncovering functionality, architecture, and vulnerabilities in complex software.

3 Course Learning Outcomes

By the end of the course, students will be able to:

- **Apply Advanced Malware Analysis Techniques** *Outcome:* Use tools such as Wireshark, Ghidra, IDA Pro, and Cuckoo Sandbox to identify malicious behaviors and evasion strategies. *Assessment:* Hands-on labs and in-class activities.
- **Leverage AI for Automated Reverse Engineering** *Outcome:* Employ LLM- and AI-driven approaches to enhance efficiency and accuracy in reverse engineering without access to source code.
- **Conduct Critical Research Analysis in Malware and Security** *Outcome:* Review and critique state-of-the-art research, identifying limitations and proposing novel approaches for malware detection and classification.

- **Design and Execute AI Red Teaming Exercises** *Outcome:* Develop and implement black-box red teaming scenarios to test AI system security, detect vulnerabilities, and suggest mitigations.
- **Integrate Approaches to Solve Complex Security Challenges** *Outcome:* Synthesize techniques from malware analysis, reverse engineering, AI, and cybersecurity to design comprehensive solutions for advanced threats.

4 Prerequisites

Students are expected to have the following background:

- Knowledge of basic computer science principles and skills, at a level sufficient to write a reasonably non-trivial computer program.
- Understanding of basic cybersecurity threats, attacks, and countermeasures.
- Basic understanding of machine learning algorithms.
- Familiarity with at least one machine learning framework, such as TensorFlow or PyTorch, is a plus.

5 Grade Breakdown

The final grade for this course will be based on the following components:

- 7 Labs, Hands-On Activities, and In-Class Exercises (55%)
- Surprise quiz (5%)
- Research Paper Review and Presentation (10%)
- Final Research Project and Presentations (25%)
- Attendance (5%)
- Bonus Lab Activities/Quizzes (5%)
- Bonus Micro-credential (5%)

5.1 Grading Policy

We will follow a standard UTEP grading policy. Grades in this course will be assigned based on the following scale:

- A: 90 – 100 %
- B: 80 – 89 %
- C: 70 – 79 %
- D: 60 – 69 %
- F: 0 – 60 %

6 Submission Policy

- **Deadlines**: All labs, assignments, and reports, must be submitted by the specified due date and time.
- **Late Submission and Penalties**:
  - 0-24 Hours Late: 10% deduction.
  - 24-48 Hours Late: 50% deduction.
  - 48+ Hours Late: 100% deduction.
- **Extensions**: Extensions may be granted for serious and extraordinary circumstances if requested before the deadline. Documentation may be required.
- **Technical Issues**: Technical difficulties are not valid reasons for late submissions. Plan ahead and submit early.

7 Course Syllabus

<table>
<tr><th>Week & Date</th><th>Description</th><th>Materials</th><th>Events</th></tr>
<tr><td>Week 1 (08/27)</td><td>Course Overview
• Fundamentals of Software Reverse Engineering
• Developing Research Mindset
Malware Research
• Research Overview
• Understanding Security and Privacy Problems
• Literature Review?
• Analyzing a Paper
• Security and Privacy Conferences</td><td>- Lecture Note</td><td>- Lecture</td></tr>
<tr><td>Week 2 (09/03)</td><td>Static Analysis
• Wireshark
• YARA</td><td>- Lecture Slides
- Lab</td><td>- Lecture
- In-Class Activity</td></tr>
<tr><td colspan="4">Week 3 (09/09), Lab 1 Due, Wireshark, YARA</td></tr>
<tr><td>Week 3 (09/10)</td><td>Static Analysis and Automated RE
• Ghidra
• LLM4Decompile</td><td>- Lecture Slides
- Lab</td><td>- Lecture
- In-Class Activity</td></tr>
<tr><td colspan="4">Week 4 (09/16), Lab 2 Due, Ghidra, LLM4Decompile</td></tr>
<tr><td>Week 4 (09/17)</td><td>Automated RE
• GhidraMCP, OGhidra
• Autonomous Privilege Escalation, RamiGPT</td><td>- Lecture Slides
- Lab</td><td>- Lecture
- In-Class Activity</td></tr>
<tr><td colspan="4">Week 5 (09/23), Lab 3 Due, GhidraMCP, OGhidra, RamiGPT</td></tr>
<tr><td>Week 5 (09/24)</td><td>Code Analysis
• IDA and IDA Pro
• Bandit</td><td>- Lecture Slides
- Lab</td><td>- Lecture
- In-Class Activity</td></tr>
</table>

Week 6 (09/30), Lab 4 Due, IDA, Bandit			
Week 6 (10/01)	**Intelligent Malware Analysis** • Supervised Learning • Malware Classification and Detection	- Lecture Slides - Lab	- Lecture - In-Class Activity
Week 7 (10/08), Paper Review Due by Group			
Week 7 (10/08)	**Paper Review Presentation by Group**		
Week 8 (10/15)	**Intelligent Malware Analysis** • Semi-Supervised Learning • Unsupervised Learning • Malware Classification and Detection	- Lecture Slides - Lab	- Lecture - In-Class Activity
Week 9 (10/21), Lab 5 Due, Intelligent Malware Analysis			
Week 9 (10/22)	**Project Proposal Development**	**NO CLASS**	
Week 10 (10/28), Project Proposal Due			
Week 10 (10/29)	**Project Proposal Presentation**		
Week 11 (11/05)	**Dynamic Analysis** • Any dot Run	- Lecture Slides - Lab	- Lecture - In-Class Activity
Week 12 (11/11), Lab 6 Due, Any dot Run			
Week 12 (11/12)	**AI Red Teaming**	- Lecture Slides - Lab	- Lecture - In-Class Activity
Week 13 (11/18), Project Update Due			
Week 13 (11/19)	**Reserve for Guest Lecture 1**		
Week 14 (11/25), Lab 7 Due, AI Red Teaming			
Week 14 (11/26)	**Reserve for Guest Lecture 2**		
Week 15 (12/03), Final Project Report Due			
Week 15 (12/03)	**CTF Competition**		

Week 16 (12/10)	**Exam Week and Final Project Presentation**

8 Reference Resources

The course does not require any textbook.

Relevant books to the course (Optional):

- "Practical Malware Analysis: A Hands-On Guide to Dissecting Malicious Software," Michael Sikorski and Andrew Honig. ISBN-13: 9781593272906. [Available on UTEP Library]
- "The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory," Michael Hale Ligh, Andrew Case, Jamie Levy, Aaron Walters. ISBN: 978-1118825099. [Available on UTEP Library]
- "The Rootkit Arsenal: Escape and Evasion: Escape and Evasion in the Dark Corners of the System," Bill Blunden. ISBN: 978-1598220612.

9 POLICIES

EXCUSED ABSENCES AND/OR COURSE DROP POLICY

According to UTEP Catalog, "*At the discretion of the instructor, a student can be dropped from a course because of excessive absences or lack of effort. A grade of "W" will be assigned before the course drop deadline and a grade of "F" after the course drop deadline.*" See Policies and Regulations in the UTEP Catalog for a list of excuse absences. Therefore, if I find that, due to non-performance in the course, you are at risk of failing, I will drop you from the course. I will provide 24 hours advance notice via email.

ACCOMMODATIONS POLICY

The University is committed to providing reasonable accommodations to students with documented disabilities. Students who become pregnant may also request reasonable accommodations, in accordance with state and federal laws and regulations and University policy. Accommodations that constitute undue hardship are not reasonable. If you need an academic accommodation, you should initiate the request with the Center for Accommodations and Support Services (CASS). Contact CASS at 915-747-5148, email them at `cass@utep.edu`, or apply for accommodations online via the CASS portal. The CASS will evaluate the request and recommend accommodations. Students should contact the CASS as soon as possible since timely notice is needed to coordinate accommodations.

INCOMPLETE GRADE POLICY

Incomplete grades may be requested only in exceptional circumstances after you have completed at least half of the course requirements. Talk to me immediately if you believe an incomplete is warranted. If granted, we will establish a contract of work to be completed with deadlines.

HONOR CODE AND SCHOLASTIC INTEGRITY

Academic dishonesty is prohibited and is considered a violation of the UTEP Handbook of Operating Procedures. It includes, but is not limited to, cheating, plagiarism, and collusion. Cheating may involve copying from or providing information to another student, possessing unauthorized materials during a test, or falsifying research data on laboratory reports. Plagiarism occurs when someone intentionally or knowingly represents the words or ideas of another as ones' own. Collusion involves collaborating with another person to commit any academically dishonest act. Any act of academic dishonesty attempted by a UTEP student is unacceptable and will not be tolerated. All suspected violations of academic integrity at The University of Texas at El Paso must be reported to the Office of Community Standards for possible disciplinary action. To learn more, please visit HOOP: Student Conduct and Discipline.

Furthermore, see below:

Honor Code and Academic Integrity

Permissive but strict. If unsure, please ask the course staff!

- **OK** to search for information and ask questions publicly about the systems we're studying.
  - Always cite all resources you reference, including papers, online articles, and any information obtained from AI tools.
  - When using AI tools, include a link to your tool's search history (e.g., share your ChatGPT workspace) and mention this in your reports, labs, and final projects.
  - If you engage in public discussions, such as on Reddit, include the link to the discussion.
- **NOT OK** to copy solutions directly from AI tools or other sources.
  - Solutions should be your original work, reflecting your understanding and analysis.
- **NOT OK** to ask someone else to complete your assignments, labs, or projects.
  - Academic integrity requires that all submitted work is your own.
- **OK** to discuss questions and ideas with classmates.
  - Collaborative learning is encouraged, but you must disclose your discussion partners in your submissions.
- **NOT OK** to copy solutions from classmates.
  - While discussion is permitted, all submitted work must be completed independently.
- **OK** to incorporate existing solutions as part of your projects or assignments.
  - Properly cite these solutions and clearly distinguish your contributions from those of others.
- **NOT OK** to present someone else's solution as your own.
  - Always attribute credit where it is due, and clearly identify your original work.
- **OK** to publish your final project after the course is over.
  - We encourage you to share your work with the broader community, contributing to the field's body of knowledge.

- **NOT OK** to post your assignment solutions online during or after the course.
  - This protects the integrity of the course for future students and maintains academic standards.

GUIDANCE ON ARTIFICIAL INTELLIGENCE

Some AI technologies or automated tools, particularly generative AI such as ChatGPT, Gemini, or DALL-E, can be beneficial during the early brainstorming stages of an activity, and you are welcome to explore them for that purpose. However, keep in mind that AI-generated ideas are not your own and may hinder your ability to think critically and creatively about a problem. It is also important to remember that these technologies often "hallucinate" or produce materials and information that are inaccurate or incomplete—even providing false citations for use. That said, you are not allowed to submit any AI-generated work in this course as your own. If you use any information or materials created by AI technology, you are required to cite it like you would any other source. Consider how this will affect your credibility as a writer and scholar before doing so. Any direct use of AI-generated materials submitted as your own work will be treated as plagiarism and reported to the Office of Community Standards.

AUDIT POLICY

We welcome auditing requests from anyone affiliated with UTEP. As an auditor, you will have access to all course lectures but will not receive grades for labs, homework, or final projects. Due to limited resources, we are unable to provide feedback on assignments or projects for auditors. If you are interested in auditing this course, please contact the Computer Science department to make the necessary arrangements.

Please note that external requests for auditing will not be considered, as the course is conducted in-person on campus. All course materials, including lecture slides, detailed notes, assignments, and final project instructions, will be made publicly available on the course website for your reference.

10 Course Policy Modifications

The Professor reserves the right to modify, change, or alter the syllabus, assignments, labs, and grading criteria of the course as deemed necessary to enhance the learning experience, accommodate unforeseen circumstances, or address the progress of the class. Any changes will be communicated promptly to students through official channels, such as email, Blackboard, and Microsoft Team.