Subject: Reverse Engineering and Malware Analysis								
Program: M.Sc. in CyberSecurity				**Subject Code:**			**Semester: II**	
Teaching Scheme				**Examination Evaluation Scheme**				
Lecture	**Tutorial**	**Practical**	**Credits**	**University Theory Examination**	**University Practical Examination**	**Continuous Internal Evaluation (CIE)- Theory**	**Continuous Internal Evaluation (CIE)- Practical**	**Total**
4	**0**	**2**	**5**	**40**	**40**	**60**	**60**	**200**

COURSE OBJECTIVES:

1. Understand the fundamentals of malware analysis, including different types of malware, their behaviors, and their characteristics.
2. Develop skills in reverse engineering, including disassembling, decompiling, and debugging software to understand its functionality and behavior.
3. Learn how to use a variety of malware analysis tools and techniques, including static and dynamic analysis, sandboxes, and debugging tools.
4. Gain an understanding of common malware techniques, including obfuscation, anti-debugging, and anti-reverse engineering techniques.
5. Develop practical skills in identifying and analyzing different types of malware, including viruses, worms, and trojans, as well as more advanced types of malware like rootkits and botnets.

Content

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	**Topics**	**W**	**T**
1	**1. Intro to course:** **1.1.** Malware, Assembly Language, Reverse Engineering 1.2. Under what circumstances is reverse engineering useful or breaking contracts? 1.3. Why is reverse engineering necessary? 1.3.1. Interoperability / Competition 1.3.2. Auditing 1.3.3. DRM 1.3.4. Analysis of Malware	**20**	**10**
2	**2. Background on Malware** 2.1. Current and Next-Generation Malicious Software 2.1.1. Viruses 2.1.2. Worms 2.1.3. Trojans 2.1.4. Botnets 2.1.5. Polymorphic and Metamorphic Malware	**20**	**10**

	2.1.6. Advanced Persistent Threats 2.2. Intro to Defensive Strategies Against Malware 2.2.1. Worm Fingerprinting / Signature Generation 2.2.2. Behavioral Approaches to Detection of Malware		
3	**3. Low level Software/Assembly:** 3.1. Overview of Intel Assembly Language 3.2. Virtual Machines for Interpreted High-Level Languages 3.3. Representation of Compiled High Level Language Structures in Assembly 3.4. Operating Systems Background 3.4.1. MS-DOS Internals Related to Malware Case Studies 3.4.2. Modern Windows Execution Environment 3.5. Executable File Formats 3.5.1. PE Files 3.5.1.1. Import Address Table	**20**	**10**
4	**4. Static Analysis of Software** 4.1. System Monitoring Tools 4.2. Dynamic Tracing: System Calls, Filesystem, and Registry 4.3. Compiler Issues 4.4. Debuggers 4.4.1. OllyDbg 4.4.2. WinDbg 4.5. Disassemblers 4.5.1. IDA Pro 4.5.2. Decompilers 4.6. Memory Analysis to Support Reverse Engineering 4.6.1. DRAM Acquisition 4.6.2. Extraction of Malware	**20**	**9**

TEXTBOOKS and REFERENCE BOOKS:

1. "Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software" by Michael Sikorski and Andrew Honig
2. "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
3. "Malware Analyst's Cookbook and DVD: Tools and Techniques for Fighting Malicious Code" by Michael Ligh, Steven Adair, Blake Hartstein, and Matthew Richard
4. "Reversing: Secrets of Reverse Engineering" by Eldad Eilam
5. "IDA Pro Book: The Unofficial Guide to the World's Most Popular Disassembler" by Chris Eagle

List of Practicals:

List of Practical	
1.	**Practical-1** Static analysis of malware samples using disassemblers such as IDA Pro or Ghidra.
2.	**Practical-2**

	Dynamic analysis of malware samples using a virtual machine and tools such as Process Monitor, Wireshark, and Sandboxie.
3.	**Practical-3** Memory analysis of malware samples using tools such as Volatility.
4.	**Practical-4** Analyzing malicious documents such as PDF files and Office documents using tools such as PDFStreamDumper, oledump, and OfficeMalScanner.
5.	**Practical-5** Reverse engineering malware using reverse engineering frameworks such as Radare2.
6.	**Practical-6** Analyzing malware that uses anti-analysis techniques to evade detection.
7.	**Practical-7** Analyzing malware that uses cryptographic techniques to hide its functionality.
8.	**Practical-8** Analyzing malware that targets specific industries or geographic regions.
9.	**Practical-9** Analyzing malware that uses rootkits to hide its presence on a system.
10.	**Practical-10** Conducting threat intelligence analysis to understand the motivations and goals of malware authors.