INT251:MALWARE ANALYSIS AND CYBER DEFENCE

L:2 T:0 P:2 Credits:3

Course Outcomes: Through this course students should be able to

CO1 :: understand major defense strategies to secure operation centers

CO2 :: explore the behavior of the malware and its interaction with the system

CO3 :: gain the basics of assembly Language and the necessary skills required to perform code analysis

CO4 :: analyze the stealth techniques used by advanced malware to hide from Forensic tools

CO5 :: apply the malware forensic techniques to investigate advanced malware

CO6 :: identify major defense strategies to secure operation centers

Unit I

Introduction to malware analysis : introduction to malware, types of malware, malware analysis, types of malware analysis

Static Analysis : determining file type, fingerprinting malware, multiple anti-virus scanning, extracting strings, determining file obfuscation, Inspecting PE header information, Comparing and classifying malware

Unit II

Dynamic Analysis : dynamic analysis steps, analysing malware, DLL analysis

Assembly language and disassembly primer : introduction to assembly language basics, registers, data transfer instructions, arithmetic operations, bitwise operations, branching and conditionals,, loops and Functions, arrays and strings, structures and x64 architecture

Unit III

Disassembly using IDA : static code analysis, dissembling Windows API

Debugging malicious Binaries : general concepts of debugging, debugging binaries

Unit IV

Malware functionalities and persistence : malware functionalities, malware persistence methods

Code Injection and Hooking : virtual memory, user mode and kernel mode, code injection techniques, hooking techniques

Unit V

Malware Obfuscation Techniques : simple encoding, malware encryption, custom encoding, malware unpacking

Hunting Malware using Malware Forensics : memory forensics steps, memory acquisition, volatility overview, enumerating processes, listing process handles, dumping executable and DLL, listing network connections and Sockets, inspecting registry, investigating service, extracting command history, listing DLL's

Unit VI

Detecting advanced malware using memory forensics : detecting code injection, investigating hollow process injection, detecting API hooks, kernel mode rootkits, listing kernel modules, I/O processing, display device tress, detecting kernel space hooking, kernel call-backs and timers

Security Operation Center : Major defense strategies, Importance of SOC, SIEM, Importance of SIEM, Case studies pertaining to SOC

List of Practicals / Experiments:

Identifying file type using manual method

- manual file identification using various methods

Identifying file type using tools

- CFF explorer
- Determining file type using python

Fingerprinting the malware

- Generating cryptographic hash using tools
- Determining cryptographic hash in python

String extraction using tools

- Decoding obfuscated strings using FLOSS

Determining file obfuscation

- packers and cryptors

Detecting and inspecting pe and exports

- Detecting inspectingpe header information file obfuscation using exeinfo pe
- inspecting file dependencies and imports
- inspecting exports
- examining pe sectiontable and sections
- examining the compilation timestamp
- examining pe resources

Comparing and classifying the malware

- classifying malware using fuzzy hashing
- classifying malware using import hash
- classifying malware using section hash
- classifying malware using yara

Dynamic analysis method

- process inspection with process hacker
- determining system interaction with process monitor
- logging system activities using noriben
- capturing network traffic with wireshark
- simulating services with inetsim

Analyzing a malware executable

- static analysis of the sample
- dynamic analysis of the sample

Dynamic-Link Library (DLL) analysis

- analyzing the dll using rundll32.exe
- analyzing a dll with no exports
- analyzing a dll with exports
- analyzing a dll accepting export arguments
- analyzing a dll with process checks

Assembly and disassembly on disk

- analyzing the program on disk
- program disassembly(from machine code to assembly code)
- analyzing 32-bit executable on 64bit windows

Static code analysis disassembly using IDA

- loading binary in ida
- improving disassembly using ida

Disassembling windows API

- understanding windows api
- windows api 32-bit and 64-bit comparison

Patching binary using IDA

- patching program bytes
- patching instructions

IDA scripting and plugins

- executing ida scripts
- ida python
- ida plugin

Malicious binaries dubugging

- debugging a binary using x64dbg
- debugging a malicious dll using x64dbg

Debugging a binary using IDA

- debugging malware executables
- debugging a malicious dll using ida
- debugger scripting using idapython
- determining files accessed by malware

References:

1. LEARNING MALWARE ANALYSIS by MONNAPPA K A, PACKT PUBLISHING