ITSC 304 - Operating System Exploitation

Course Description:
This course provides students the opportunity to examine the source code of several operating systems. The emphasis will be on the overall structure, internal algorithm implementation potential weaknesses and hardening of each system.

3 credits

Time Guidelines:
The standard instructional time for this course is 90 hours.

Effective Year
2021/2022

Prerequisite(s):

- ITSC 203
- ITSC 204
- ITSC 205

Course Assessment:

Labs	20%
Lab Quizzes	30%
Theory Quizzes	20%
Final Pen-Testing Project	30%
Total	100%

SAIT Policies and Procedures:
For information on the SAIT Grading Scale, please visit policy AC 3.1.1 Grading Progression Procedure: http://www.sait.ca/Documents/About SAIT/Administration/Policies and Procedures/AC.3.1.1 Grading and Progression Procedure.pdf

For information on SAIT Academic Policies, please visit: www.sait.ca/about-sait/administration/policies-and-procedures/academic-student

Course Learning Outcome(s):

1. Assess Vulnerabilities
   Objectives:

   1.1 Describe vulnerability and its classification

   1.2 Compare attack's frameworks

   1.3 Describe the phases of vulnerability assessment life cycle

   1.4 Differentiate vulnerability scoring systems (CVSS)

   1.5 Explore databases and platforms that provide vulnerabilities information

   1.6 Detect vulnerabilities in operating systems using scanning tools.

   1.7 Describe critical elements of vulnerability assessment reports

2. Analyze basic techniques to exploit and ethically hack operating systems
   Objectives:

   2.1 Review operating systems concepts, programming languages and assembly instructions required to exploit operating systems

   2.2 Describe offensive exploitation techniques

   2.3 Outline general purposes and steps of exploiting computer operating systems

   2.4 Implement techniques to create clean shellcode and ethically exploit operating systems

   2.5 Compare current techniques implemented to exploit operating systems such as stack buffer overflow

   2.6 Develop payloads to ethically exploit operating systems

   2.7 Analyze operating systems exploits using debuggers

   2.8 Generate automatically shellcode and payloads via framework such as Metasploit to exploit operating systems

3. Analyze advanced techniques to ethically exploit and post-exploit operating systems
   Objectives:

   3.1 Differentiate user and kernel mode operating systems' exploitation

   3.2 Compare current advanced techniques to exploit operating systems

   3.3 Exploit ethically kernel structures by implementing bypass methods to circumvent operating systems' defenses

   3.4 Differentiate post-exploitation and persistence methods

   3.5 Hack ethically operating system by implementing post-exploitation and persistence methods that prevent detection

   3.6 Generate automatically post-exploitation code via framework such as Metasploit to post-exploit operating systems

4. Evaluate methods to manipulate kernel structures:
   Objectives:

   4.1 Examine kernel data structures and functions.

   4.2 Describe the purpose and construction of kernel module.

4.3 Create kernel modules to execute as part of the operating systems

4.4 Modify kernel data structures to avoid detection.

4.5 Describe Windows kernel device drivers and how can be exploited

4.6 Create kernel device drivers to execute as part of the operating systems

5. Analyze Bootkits in firmware and hardware

Objectives:

5.1 Differentiate rootkits and bootkits

5.2 Compare exploits in firmware and hardware

5.3 Hack ethically functions to access kernel data.

5.4 Demonstrate methods of hooking into kernel objects.

5.5 Demonstrate methods to infect startup (boot/firmware) code.

5.6 Experiment with firmware exploits to affect hardware.

5.7 Defend the boot process using trusted process modules.

6. Examine detection and preventions techniques

Objectives

6.1 Validate the presence of exploits using monitoring tools

6.2 Demonstrate methods to discover the presence of exploits

6.3 Compare common detection and evasion techniques

6.4 Use operating system tools to generate logs and monitor the system

6.5 Implement open-source infrastructure to monitor and analyze data(logs) generated by different Sources in different formats

7. Explain defense mechanisms to harden the operating systems against exploitations.

Objectives:

7.1 Describe software that can be used to harden operating systems.

7.2 Discuss operating systems implementation strategies that can be used to harden operating systems

7.3 Compare existing operating systems attestation technology.

7.4 Describe evasion techniques and breaching defenses

7.5 Describe operating systems' threat mitigations to protect it against exploitation

7.6 Configure operating systems built-in security features that harden and protect it to mitigate threats

8. Apply Ethical Hacking and Operating Systems Pen-Testing.

Objectives:

8.1 Describe penetration testing methodologies

8.2 Differentiate penetration testing categories

8.3 Describe structure of penetration testing reports

8.4 Implement vulnerability assessment, ethical exploitation and post-exploitation of operating systems

8.5 Write reports for specific audience such as: executive, management and technical