**FORMAN CHRISTIAN COLLEGE UNIVERSITY**

**CSCS XXX: Cybersecurity (3 Credit Hrs)**

**Course Outline and Lesson Plan**

**Instructor Information:**

**Name:** M. Rauf Butt

**Contact:** [raufbutt@fccollege.edu.pk](mailto:raufbutt@fccollege.edu.pk)

**Office:** S-214

**Office Hours:** TBD

**Pre Requisites:**

* Data Structures & Algorithms

**Course Material:**

1. Lab/Class Activity Handouts
2. Class Handout / Lecture slides

**Text Books:**

**Course Objectives:**

The cybersecurity course aims to equip students with a comprehensive understanding of key cybersecurity principles, including the identification and mitigation of threats, vulnerabilities, and risks. Students will learn to implement security measures, develop incident response strategies, and apply ethical and legal considerations in real-world scenarios. The course will also cover secure network design, reverse engineering tools, and the integration of security into the software development lifecycle. Through hands-on exercises, students will develop practical skills in using cybersecurity tools, analyzing incidents, and effectively communicating security strategies to various stakeholders.

After completing the course, students should be able to:

1. Identify and Analyze Cyber Threats: Recognize various types of cyber threats, vulnerabilities, and attack vectors, and assess their potential impact on systems and data.

2. Implement Security Measures: Configure and deploy security technologies, such as firewalls, intrusion detection/prevention systems, and encryption methods, to protect networks and data.

3. Develop and Execute Incident Response Plans: Create and execute plans to respond to and manage cybersecurity incidents, including conducting forensic investigations to determine the scope and source of breaches.

4. Apply Risk Management Techniques: Evaluate cybersecurity risks and develop strategies to minimize or mitigate these risks, including the application of security controls and policies.

5. Understand and Apply Legal and Ethical Principles: Navigate the legal, regulatory, and ethical aspects of cybersecurity, ensuring compliance with laws and standards such as GDPR, HIPAA, and other relevant regulations.

6. Secure Networks and Systems: Design secure network architectures, implement access controls, and ensure the confidentiality, integrity, and availability of information.

7. Conduct Security Audits and Assessments: Perform security audits and assessments to identify weaknesses in systems, networks, and applications, and recommend improvements.

8. Communicate Cybersecurity Strategies: Effectively communicate cybersecurity issues, strategies, and solutions to both technical and non-technical stakeholders.

9. Stay Updated on Emerging Trends: Monitor and adapt to emerging cybersecurity threats, technologies, and trends to continuously improve security practices.

10. Collaborate in Cybersecurity Teams: Work effectively in teams to address cybersecurity challenges, share knowledge, and develop comprehensive security solutions.

**Course Learning Outcomes (CLOs)**

|  |  |  |
| --- | --- | --- |
| **CLO’s** | **Description** | **Level** |
| **CLO:1** | **Describe the functions of different components of cyber security phases.** | **C1 (Remember)** |
| **CLO:2** | **Describe how a binary gets executed and what is the main difference between a 32- and 64 bit binary.** | **C2 (Understand)** |
| **CLO:3** | **Application of different tools for scanning and pentesting of networks and web-based applications.** | **C3 (Apply)** |
| **CLO:4** | **Describe and apply knowledge to determine the software vulnerabilities based on stack overflow, return to libC and return oriented programming** | **C3 (Apply)** |
| **CLO:5** | **Learn how to write effective programs for scanners, exploits and malwares.** | **C3 (Apply)** |

**Mapping of CLO’s to PLO’s**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PLOs** | **CLO:1** | **CLO:2** | **CLO:3** | **CLO:4** | **CLO:5** |
| Computing Knowledge | √ | √ | √ | √ | √ |
| Problem Analysis |  | √ | √ | √ | √ |
| Design and development of solutions |  |  |  | √ | √ |
| Investigation |  |  |  | √ |  |
| Modern Tool Usage |  | √ | √ | √ | √ |

|  |  |  |
| --- | --- | --- |
| **Week** | **Theory Session (1 Hr 15 Min)** | **Sessionals** |
| Module 1: Penetration Testing   * Teaching penetration testing (pentesting) techniques such as   + scanning with Nmap   + using the Metasploit framework   + conducting web application pentesting | | |
| 1, 2 | * Introduction to the course * The lab environment   + Virtual Box equipped with following machines     - Kali Linux     - Metasploitable 2     - **Metasploitable-3**     - Windows 10 * Introduction to Linux environment   + Some basic Linux commands used to work with the terminal * Introduction to Networks   + Network hosts   + Network devices   + The OSI Model   + Addressing on different layers   + How data travels from one host to another in a Network   + Some well-known protocols, ports and services   + Verifying TCP/IP configurations on VMs     - Ifconfig     - Ping     - Arp     - Route     - Host     - Nslookup     - Idig   + Bind shell Vs Reverse shell   + Introduction to Wireshark | Class Task-1  Quiz 1  Assignment 1(Setting up the lab environment) |
| 3 | * Information gathering   + Tools used     - Ping command     - Whatweb     - Google Dorks     - Nslookup     - theHarvest     - Hunter.io     - ipInfo * Scanning   + Tools used     - Nmap     - Searchsploit     - Netdiscover     - Nessus     - OpenVas   + Using Wireshark to determine whether the machine is being scanned. | * Class Task 2 * Quiz 1 |
| 4 | * The Metasploit Framework   + Anatomy and structure   + Exploring auxiliary module of Metasploit framework.     - auxiliary/scanner/portscan/tcp     - auxiliary/scanner/discovery/udp\_sweep     - auxiliary/scanner/ftp/ftp\_login     - auxiliary/scanner/ftp/ftp\_version   + Performing exploitation on services (running on a Linux based vulnerable system) like     - ftp     - ssh     - telnet     - smtp     - http     - netbios-ssn * Working with some cybersecurity exploits on Windows machines.   + Eternalblue Attack   + Double Pulsar attack * Generating payloads using   + Msfvenom   + Veil |  |
| 5 | * Web App Pentesting using   + DVWA   + BurpSuite   + Kali / Metasploitable 3 / Windows 10 * Exploiting vulnerabilities   + Shellshock     - Exploiting how Bash handles environment variables.   + Command Injection     - Adding commands to an improperly sanitized user input code on a web site.   + Reflected XSS     - Returning a malicious JavaScript code.     - The malicious code comes from inside the page source code.   + Stored XSS     - Also returns a malicious JavaScript code.     - The code is stored on the server side | * Quiz 2 * Assignment 2 |
| 6, 7 | * Exploiting vulnerabilities …   + HTML Injection     - An HTML code is injected   + SQL Injection     - injecting or inserting malicious SQL code into a query through user input fields, allowing the attacker to manipulate the query and access, modify, or delete data stored in the database. * Man in the middle attack   + Fooling the host and router in a private network by making changes in the arp table.   + Tools used     - Bettercap     - Ettercap     - Manual ARP Spoofing using scapy * Anonymity   + Anonymizing our IP address using     - Tor Browser     - Proxies     - VPN * Post Exploitation * Maintaining Persistence | * Class Task 3 * Quiz 3 |
| Module 2: Binary Exploitation   * Binary exploitation in cybersecurity is the process of finding and exploiting vulnerabilities in compiled software binaries, typically written in low-level programming languages like C or C++. * These vulnerabilities often arise from programming errors or flaws in the way memory is managed. * Exploiting these flaws allows an tester to execute arbitrary code, gain access, or test for unintended behavior in the software. | | |
| 8, 9 | * An overview of C and Assembly programming languages.   + Basic programming constructs of C   + A brief introduction to gdb and gdb-gef.   + Basic programming constructs of Assembly programming   + Compiling (and linking) programs using 32 bit and 64-bit architecture.   + Understanding stack memory.   + Function calling conventions in 32 bit and 64-bit architectures. * Learning how to read and understand the assembly generated code from the binary of a C program. * Simple stack-based buffer overflow using 32- and 64-bit architecture   + Identifying a program with a buffer overflow vulnerability   + Disassembling the binary to identify functions that use vulnerable buffers on the stack   + Overflowing the buffer   + Controlling the return address | * Assignment 3 * Class Task 4 |
| 10 | * Writing basic shell code and testing the functionality of shellcodes using simple function pointer-based C program. * Shellcode based buffer overflow   + 32-bit architecture   + 64-bit architecture | * Class Task 5 |
| 11 | * Return oriented programming   + Identify the buffer overflow or similar vulnerability   + Scanning the binary to identify rop gadgets using tools like     - Ropegadgets     - Ropper   + Constructing the ROP chain   + Test the rop chain * Using ROP to bypass N0 Execution NX bit * Using ROP to handle Address Space Layout Randomization ASLR | * Quiz 4 * Course Project Proposals+ * Assignment 4 |
| Module 3: Development of scanners / exploits / malwares using Python   * Understanding and defending against malicious code, rather than creating it for offensive purposes. * Ethical guidelines, legal considerations, and a strong emphasis on responsible use will be an integral part of this module. | | |
| 12 | Development of scanners / exploits / malwares   * + Introduction to network programming using python   + Changing MAC addresses of devices in the network.   + Introduction to ARP Spoofing   + Creating and sending ARP request and response packets   + Generating and restoring ARP table   + Packet sniffing     - Extracting data from a specific layer and its fields   Analyzing and extracting passwords | * Quiz 5 |
| 13 | Development of scanners / exploits / malwares   * + Introduction to DNS Spoofing     - Filtering DNS responses     - Redirecting DNS responses   + Writing a File interceptor     - Filtering traffic based on a port     - Analyzing, intercepting and modifying HTTP requests   + Intercepting and replacing downloads on the network | * Class Task 6 |
| 14 | Development of scanners / exploits / malwares   * + Modifying HTTP responses and injecting JavaScript code in HTML pages   + BeEF (Browser Exploitation Framework) overview and basic Hook Method   + Hooking computers using code injector   + BeEF commands   + Delivering malware using BeEF | * Class task 7 * Quiz 6 |
| 15 | Development of scanners / exploits / malwares   * + Sending emails from your code   + Stealing saved passwords from remote computers   + Logging key strokes and reporting them via emails.   + Creating a listener and a backdoor   + Downloading / uploading files from/to the target machine / backdoor   + Converting Python programs to exe files |  |
| 16 | Development of scanners / exploits / malwares   * + Packaging programs   + Running programs from startup   + Creating Trojan     - Using download and executing payload     - Embedding files in program code   + Bypassing antivirus programs   + Adding icons   + Spoofing file extensions | * Course project submission |

Note that this outline is not carved on stone. Course staff / instructor reserves all rights to make appropriate changes as per needed.

**Assessment Criteria**

* In Class Quizzes (5 to 8) 15%
* Class Activities (5 to 8) 20%
* End Semester Exam (Comprehensive) 40%
* Assignments 15%
* Class Project 10%

**Class Policies:**

* This is a highly practical oriented course.
* We need to set up our working lab environment in the form of attacking and target machines on virtual box.
* The vdi files of all required machines will be provided and students MUST complete the environment on their systems within the first week of course.
* You are allowed (rather strongly recommended) to work in groups of maximum three students.
* Each group is responsible for her lab environment setup.
* Make sure that the group should be made on a permanent basis for the entire course duration.
* ***This is entirely the responsibility of the group members to always have a machine equipped with the lab environment while in class.***
* Each group must bring the system (installed with the lab environment) in every class. We may have an unannounced class activity which will need the system.
* We may have 5 to 8 quizzes. If number of quizzes is greater than 5, we will drop one quiz.
* Assignments/Home works will be uploaded on MOODLE and **MUST** be submitted within the deadline specified on handout.
* There will be **no retake for any instrument.**
* **Students are advised to attend all assigned lectures**. It is entirely the students' responsibility to recover any information presented during the lecture/s from which they were absent.
* **All work** that you submit in this course **must be your own**.
* **Unauthorized group efforts** are considered academic dishonesty.
* You may discuss assignments/project in a general way with others, but you must not consult anyone else's written work. You are guilty of academic dishonesty if:
  + You **examine another's solution** to an assignment
  + You **allow another student to examine your solution** to an assignment
  + You fail **to take reasonable care to prevent another student from examining your solution** and that student does examine your solution.
* **Cheating, plagiarism and other forms of academic fraud** are taken very seriously. University Policy of plagiarism will be applicable in the case.
* **Attendance Policy**
  + **A minimum of 69% attendance in the course MUST be observed.**
  + **Student whose attendance falls below the threshold without any acceptable reason will not be allowed to sit in the FINAL exam, which results in an F grade in the course.**

**Student Commitment Statement**

I, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, roll number \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ a student enrolled in the Cyber Security Course CSCS495 A at FCCU Lahore, hereby acknowledge the importance of ethical conduct in the field of cybersecurity. I understand that the concepts, tools, and techniques taught in this course are intended solely for educational purposes and for the defense against cyber threats.

I commit to the following:

Legal Compliance: I will not use any knowledge, skills, or tools acquired during this course for any unauthorized or illegal activities. I understand that unauthorized access to computer systems, networks, or data is against the law and may result in severe legal consequences.

Ethical Responsibility: I will uphold the highest standards of ethical behavior in all my cybersecurity endeavors. I will use the skills gained in this course to protect and secure systems, networks, and data, rather than to exploit vulnerabilities for personal gain or to cause harm.

Respect for Privacy: I will respect the privacy and rights of others. I will not engage in activities that invade the privacy of individuals or organizations, or that could lead to the unauthorized disclosure of sensitive information.

Authorized Use: I will only perform security assessments, penetration tests, or any other cybersecurity-related activities on systems, networks, or data for which I have explicit permission from the appropriate authorities.

By signing this statement, I affirm my commitment to using the knowledge and skills gained from this course in a responsible, legal, and ethical manner. I understand that any violation of this commitment may result in disciplinary action by FCCU and could have serious legal implications.

Student Name:

Course Title/code: Introduction to Cybersecurity CSCS495

Date:

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

This statement reinforces the ethical and legal obligations that come with learning cybersecurity tools and techniques and serves as a formal commitment by the students to use their knowledge responsibly.