MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية					
Module Title	**Python Programming in Cybersecurity / برمجة بايثون في الأمن السيبراني**			**Module Delivery**	
Module Type	**Core**			☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	CYBT202				
ECTS Credits	**5**				
SWL (hr/sem)	**125**				
Module Level			**Semester of Delivery**	3	
Administering Department	Department of Cybersecurity Technical Engineering		**College**	Technical Engineering College of Computer and Artificial Intelligence - Kirkuk	
Module Leader			**e-mail**		
Module Leader's Acad. Title			**Module Leader's Qualification**	Ph.D. in Cybersecurity / Computer Science	
Module Tutor			**e-mail**		
Peer Reviewer Name			**e-mail**		
Scientific Committee Approval Date	2/11/2026		**Version Number**	1.0	

Relation with other Modules العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	CYBT105	**Semester**	2

Co-requisites module	Information Security & Cryptography	**Semester**	5 & 6

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	This module aims to introduce students to using Python programming for automated cybersecurity operations, network penetration testing, vulnerability assessment, digital forensics, and security scripting. It focuses on sockets programming, packet sniffing and crafting using Scapy, web security automation, password brute-forcing, cryptography automation, log parsing, malware analysis automation, and constructing custom command-line security tools.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Understand the role of Python in automating security tasks, threat intelligence, and penetration testing. Develop custom network scripts using sockets and Scapy for port scanning, packet sniffing, and traffic analysis. Implement web security scripts for vulnerability assessment, web scraping, and automated security testing. Write scripts for brute-forcing authentication mechanisms (SSH, HTTP, FTP) and cryptography automation. Automate log analysis, incident response tasks, and digital forensics evidence extraction using Python libraries. Understand basic reverse engineering and automated malware analysis techniques in Python. Design and deploy end-to-end security tools and automation scripts for defense and offensive testing.
Indicative Contents المحتويات الإرشادية	The module will cover the following key areas: Introduction to Python for Security Automation & Ethical Hacking Environment Setup Python Network Programming: Socket API, Client/Server Architecture, and Banner Grabbing Packet Sniffing and Packet Crafting with Scapy Network Scanning: Custom Port Scanners, ARP Spoofing, and Packet Injection Web Security Automation: Requests, BeautifulSoup, HTTP Headers, and Web Vulnerability Scanning (XSS/SQLi) Cryptography and Hashing in Python: Hashlib, Cryptography Library, and Password Cracking Brute-Forcing Tools Development (SSH, FTP, Web Forms) System and OS Security Scripting: Subprocess, Sys, OS modules, and Registry Analysis Log Parsing, Threat Intelligence Scraping, and Automated Forensic Evidence Extraction Basics of Python for Malware Analysis, YARA Rules Integration, and Security Tool Building

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	The module will employ a combination of strategies including: Lectures: To provide theoretical foundations on cybersecurity concepts, scripting logic, and security automation. Group Discussions: To facilitate understanding through code reviews and analyzing attack/defense scenarios. Problem-Solving Sessions: To write and debug Python scripts for network scanning, brute-forcing, and parsing. Practical Lab Sessions: Hands-on experience developing Python security tools in isolated Kali Linux environments. Tutorials: To delve into advanced topics like Scapy packet manipulation and digital forensics scripting. Mid-term and final exams: To assess understanding and practical application of security scripting knowledge. Preparatory Week: To give students time to revise and clarify any difficulties before the final exam.

Student Workload (SWL) الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	63	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	2
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	62	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	2
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	125		

Module Evaluation تقييم المادة الدراسية					
		Time/Number	**Weight (Marks)**	**Week Due**	**Relevant Learning Outcome**
Formative assessment	**Quizzes**	2	5% (5)	5, 10	LO #1, 3 and 5
	Assignments	2	5% (5)	2, 12	LO #2, 4 and 6
	Projects / Lab.	1	10% (10)	Continuous	LO #7
	Report	1	10% (10)	13	LO #3, 5 and 7
	Midterm Exam	2 hr	20% (20)	8	LO #1 - 4

Summative assessment	**Final Exam**	2hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Introduction to Python for Cybersecurity: Environment Setup (Kali Linux), Security Scripting Fundamentals.
Week 2	Network Programming with Sockets: Socket Creation, Connection Handling, and Banner Grabbing.
Week 3	Advanced Network Scripting: Multithreaded Port Scanners and Service Detection.
Week 4	Packet Sniffing with Scapy: Sniffing Packets, Filter Specifications, and Protocol Analysis.
Week 5	Packet Crafting and Network Attacks: ARP Spoofing/Poisoning, SYN Flooding, and Packet Injection using Scapy.
Week 6	Web Security Scripting: Interacting with Web APIs, Handling HTTP Headers, Cookies, and Sessions.
Week 7	Web Vulnerability Automation: Automated Scraping for Vulnerabilities (SQL Injection & XSS Payloads).
Week 8	Mid-term Exam (الفصل منتصف اختبار) & Cryptography Automation: Hashlib, Symmetric/Asymmetric Encryption in Python.
Week 9	Automated Password Cracking: Hash Identification, Dictionary Attacks, and Rainbow Tables.
Week 10	Brute-Forcing Protocols: Writing Brute-Force Tools for SSH, FTP, and Web Authentication Forms.
Week 11	OS Interaction and Automation: Subprocess Module, System Commands Execution, and File System Analysis.
Week 12	Digital Forensics Scripting: Extracting Metadata (EXIF), Registry Analysis, and Memory Dump Parsing.
Week 13	Log Analysis and Threat Intelligence: Parsing Web/System Logs, IP Geolocation, and API Threat Feeds.
Week 14	Basics of Python for Malware Analysis: PE File Analysis (pefile module) and YARA Rule Matching.
Week 15	Security Tool Integration: Building a Complete Command-Line Security Testing Suite / Tool.
Week 16	**Preparatory Week & Final Examination (النهائي والاختبار التحضير أسبوع).**

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	Lab 1: Python Environment Setup on Kali Linux, Virtual Environment, and Basic Security Script Execution.
Week 2	Lab 2: Building a Custom Port Scanner and Service Banner Grabber using Python Sockets.
Week 3	Lab 3: Packet Sniffing and ARP Poisoning Script Implementation using Scapy.
Week 4	Lab 4: Building a Web Vulnerability Scanner (Detecting Form Vulnerabilities & Header Security).
Week 5	Lab 5: Developing a Multithreaded SSH/FTP Brute-Force Script and Hash Cracker.
Week 6	Lab 6: Automated Log File Parser and Forensic Evidence Extractor (EXIF & Browser History).
Week 7	Lab 7: Revision, Building an Integrated Security CLI Tool, Practical Exam, and Project Assessment.

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts	Seitz, J. (2021). "Black Hat Python: Python Programming for Hackers and Pentesters" (2nd ed.). No Starch Press.	Yes
Recommended Texts	O'Connor, T. (2012). "Violent Python: A Cookbook for Hackers, Forensic Analysts, Penetration Testers and Security Engineers". Syngress.	Yes
Websites	Yes	

Grading Scheme مخطط الدرجات				
Group	**Grade**	التقدير	**Marks (%)**	**Definition**
Success Group (50 - 100)	**A** - Excellent	امتياز	90 - 100	Outstanding Performance
	B - Very Good	جيد جدا	80 - 89	Above average with some errors
	C - Good	جيد	70 - 79	Sound work with notable errors
	D - Satisfactory	متوسط	60 - 69	Fair but with major shortcomings
	E - Sufficient	مقبول	50 - 59	Work meets minimum criteria

Fail Group (0 – 49)	**FX –** Fail	راسب (قيد المعالجة)	(45-49)	More work required but credit awarded
	F – Fail	راسب	(0-44)	Considerable amount of work required

Note: Marks Decimal places above or below 0.5 will be rounded to the higher or lower full mark (for example a mark of 54.5 will be rounded to 55, whereas a mark of 54.4 will be rounded to 54. The University has a policy NOT to condone "near-pass fails" so the only adjustment to marks awarded by the original marker(s) will be the automatic rounding outlined above.