Mawhiba Universal Enrichment Program 2024

Course outline proposal for Cybersecurity & Cryptography

1. Course content

Core topics

Core topics should add up to 30 hours for STEM courses, and 16 hours for Academic English.

No.	Topic description	Hours
1	**Basic Coding (30 minutes at the end of each day):** This should be focused on almost daily. It is among the most sought-after and transferable skills, even in non-programmer fields. Students should have a strong foundation and understanding before we touch on any specialist advanced coding topics. We will use this time to further explore and investigate core topics, enhancing the students' programming abilities and giving them practical experience. **This time will include any activities from other core topics, so it is not required to do this every day.**	~0.5/day
Week 1		
2 Day 1	**Course Introduction**	0.5
	Career Paths in Cybersecurity: Begin by potentially instilling additional motivation in students. I would discuss certifications, courses, competitions and potential jobs and salaries. This should be in the first lecture (immediately following my introduction to students) and take roughly 15 minutes, with opportunities for questions and answers.	0.25
	Critical Cybersecurity Case Studies: This should also be in the initial lecture, which should take roughly 15 minutes. This is to provide additional motivation and to refer back to it as we teach future core topics. This way, students will clearly understand that the theory they are learning is entirely applicable in practice and potentially give some excitement at the beginning (I mean, *how does the US Pentagon get hacked?…*). As a side point, the rest of this lecture can be used to discuss some bases around *security*, *hackers* and *threats* and perhaps *authentication* (including *entropy*, *physical tokens*, *two-factor authentication* (*2FA*) and *biometrics*).	0.25
3 Day 2	**Architecture of Computers:** This is an integral core part of computer science. I would spend at least 3 hours of lectures discussing computer architecture (*Boolean logic*, *truth tables*, *number systems*, *memory*, *operating systems*, *CPUs*, *GPUs*). I would love to have a budget and time allocated to show students how to **build actual computers**, from beginning to end, including installing an operating system. This should involve a hands-on lab and workshops on assembly code examples and buffer overflow activities.	3
4 Day 3	**Database Systems:** I would spend 3 hours of lectures regarding database systems (*database architecture*, *ethics*, *relation model in SQL*, *relational algebra*, *ER modelling*, *normalisation* and *database design*). This would be supplemented with exercises in SQL and Python. Discussions on *big data*, *database management systems*, maintaining *data privacy regulations*, and *visualising*. We would learn how to exploit vulnerabilities and perform SQL injections. We will discuss ethics and look at cyber case studies.	3

5 Days 4/5	**Cryptographic Techniques (Encryption and Decryption):** One of the most important topics. I would dedicate 30 minutes to an introduction to this area, followed by 2 lectures discussing encryption, 2 classes discussing decryption and a practical activity combining all of this (for example, an exercise in pairs to encrypt and decrypt a message). We will discuss *symmetric cyphers*, the *RSA algorithm*, *advanced encryption standards* (AES), *electronic code books* (ECB), *public key cryptography*, and issues with *randomness*.	7
Week 2		
6 Days 1/2	**Network Security Concepts:** *We will dedicate two entire days to this topic.* One of the most important topics. This would include discussing *TCP/IP networking threats* and *architecture*, *network defences (TLS and firewalls), packet filtering, defence against man-in-the-middle attacks, firewalls* and *domain name systems (DNS)*. Introduction to the internet infrastructure and network architecture.	6
10 Day 2	**Cybersecurity Legislation and Regulation:** Based on all the information, we should discuss legislation and regulations in major countries (Saudi Arabia, the United Kingdom, and the United States). This would involve a group debate with all students.	1
Hacking and Cyber Attacks: Many core topics revolve around attacks. We would combine this all under *Hacking and Cyber Attacks*, initially beginning with an introduction, followed by 30 minutes on each of the following topics, which include analysing case studies or performing practical activities.		
7 Day 3	**Cyber Attacks**: These include **Brute Force Attacks, DDoS Attacks, Buffer Overflow** and **SQL Injection.**	3
8 Day 4	**Malware and Viruses:** This is similar to the previous area but involves much more detail. Thus, I suggest more time dedicated to this. Malware discussed would be **logic bombs, backdoors, viruses, supply chain tracks, Trojan horses** and **worms**. This would also include methods to prevent malware and viruses, such as **anti-viruses, command injections** and **input validation**.	3
9 Day 5	**Threat-Vectors and Threat Agents:** Vector-based threats involve **Cross-Site Scripting, Insider Threats, and Social Engineering**. Agent-based threats involve **Hacktivists, Script Kiddies, Cyber Crime Gangs** and **Black & White Hat Hackers.**	3

Specialist topics

Specialist topics should add up to 9 hours for STEM courses, and 20 hours for Academic English.

Week 3			
No.	**Topic description**	**Hours**	**Tutor**
Advanced Coding: Integrated development environments, multiple programming languages, procedural programming, object-oriented programming, web development, coding libraries and packages, version control.			Omar
1a	I would spend half an hour introducing integrated development environments (IDEs, specifically *VSCode*) and how they can help improve efficiency whilst programming. This would be supplemented with students downloading and using them for themselves.	0.5	
1bi	An hour to explain different programming languages, their uses and the best places to learn them. This would include *C* for briefly explaining procedural programming and *C++* for object-oriented programming.	0.5	
1bii	I would touch on web development (*HTML*, *CSS* and *JavaScript*) and the different libraries and packages used (with examples in *React*).	0.5	
1c	Half an hour to introduce and use version control. Students would make *GitHub* accounts, create repositories using the IDE, attempt to create branches, make commits, and perform push and pull requests. All students could use this repository to store all their previous work and future work from this point on.	0.5	
Data Science and Artificial Intelligence (AI): (data processing, data visualisation, machine learning, deep learning). Essentially, it is a crash course that gives a strong foundation. Machine learning and the whole artificial intelligence field are so important that it is integral to teach students what this is and how it works so that it isn't just a mystery. The applications are endless, and we will show how we can apply ML and AI to cybersecurity and cryptography.			
2a	A crash course on data processing, learning about collecting data online, using existing datasets, types of data, and issues with data, with a final focus on data cleaning and manipulation with practical exercises. All of this will be supplemented with the Python programming language.	1	
2b	Another crash course on data visualisation. I would discuss different visualisation libraries in Python and methods of understanding how to use them. We will focus on data visualisation methods using datasets they would have collected in the previous crash course on data processing. We will learn how to create nice visualisations and components of plots and graphs.	1	
2c	A machine learning (ML) crash course. As this field is vast, the main focus will be to introduce what ML is and actually understand this domain. We will focus on some basic fundamentals and then look at different Python models with practical	2	

	examples. Models will include *linear regression*, *lasso regression* and *decision trees* (including *random forest* models). There will be a discussion on *loss functions* and how we develop, train, test and evaluate ML models. I would provide students with excellent resources they can use to continue learning about this field.		
2d	A deep learning (DL) crash course. Like the previous ML course, we will operate similarly in the approach. The main focus will be *neural networks* and using *backpropagation* and *gradient descent*. We will look at practical examples using the *MNST* dataset to classify hand-written numbers. We may introduce more models, such as *transformers* (how models like *ChatGPT* work) and *convolution neural networks* (for operating on images with a practical exercise on image segmentation). We will conclude by presenting DL applications within the cryptography and cybersecurity field.	2	
The actual final hour will be kept for extension activities and as a buffer in case circumstances lead to the planned classes not being completed. I would expect this set of topics to be interesting for the students, so it should be more interactive and have more questions and answers from students.			
Cyber Incident Response and Threat Intelligence: Threat intelligence and forensics are major components of successful incident response in defensive cybersecurity. The area requires significant connection-building and critical thinking skills. This module will provide an introduction to current approaches, technologies, and techniques of defensive and investigative cybersecurity.			Quincy
1a	Threat Intelligence frameworks: MITRE Att&ck, Cyber Kill Chain, Diamond model. Information vs. Intelligence. Introduction to cybercrime studies. Introduction to risk management.	1	
1b	Introduction to incident response triage tools: SIEM, Intrusion Detection System (IDS)/Intrusion Prevention System (IPS), and access controls. Explain activities associated with asset and vulnerability management. Exercises associated with alerting and monitoring.	1	
1c	Case Studies in Threat Intelligence and Forensics.	1	
1d	Game of APT Clue and small group presentations.	1	
Digital Forensics: Malicious software is a significant component of many computer security incidents. Building off the discussion and activities in the malware module, this section of the incident response module will address the forensic background and dissection of malware.			
2a	A crash course on malware types and analysis techniques. Introduction to reverse engineering and safety procedures associated with analysis. Learn how to use virtual machines and reverse engineering tools.	2	
2b	Learn about static analysis techniques such as anti-virus, hashes, and file header assessment. Basic static analysis demonstration and labs.	1.5	
2c	Set up a sandbox and monitor processes to understand malware. Basic dynamic analysis demonstration and lab.	1.5	

2. Assessment components

Only indicate structured, more significant assessment items, such as a mid-term test, or an ongoing project; there's no need to indicate mini quizzes or brief knowledge checks. Indicate how much each component contributes towards the final grade (only applicable for STEM courses).

No.	Assessment description	Weight	Week
1	**Case Study Practical:** Given a case study, answer some relevant questions.	5%	1
2	**Coding Practical:** Simple exercises in Python.	5%	1
3	**Architecture Practical:** Pick a component in a computer and explain its role. Describe its inputs and outputs and what it interfaces with. Take a question with some logic gates and describe what it does, as well as a complete truth table.	10%	1
4	**Databases Practical:** Design an SQL schema from a list of requirements and perform ER modelling. Question on SQL query.	10%	1
5	**Cryptographic Techniques Practical:** encrypt and decrypt a message.	15%	1
6	**Network Security Concepts Practical:** describe how the internet works, why we need security and how we implement it.	15%	2
7	**Hacking & Cyber Attacks Practical:** The mid-term (end of week 2) test will ask questions about different attacks and potential mitigations.	20%	2
8	**Written Assessment:** One question on each previous major topic.	20%	3

3. Recommended textbooks and learning materials

Indicate at least one essential and one supplementary textbook. Highly recommended to include at least one useful online resource.

No.	Full Reference	Type of reference
1	Neil, I. (2024). CompTIA Security+ SY0-701 Certification Guide: Master cybersecurity fundamentals and pass the SY0-701 exam on your first attempt. Packt Publishing. https://books.google.co.uk/books?id=veXvEAAAQBAJ	Essential Textbook
2	Andress, J. (2019). Foundations of information security: a straightforward introduction. No Starch Press.	Supplementary Textbook
3	Anderson, R. (2020). Security engineering: a guide to building dependable distributed systems. John Wiley & Sons.	Supplementary Textbook
4	Codecademy Learn python 3: https://www.codecademy.com/learn/learn-python-3	Online Resource
5	Hack the Box *Easy* Labs: https://www.hackthebox.com/	Online Resource