ABOUT THE COURSE: CYBER SECURITY

TOTAL DURATION :	45HRS
MODE OF DELIVERY	ONLINE TRAINING / SELF LEARNING
TOTAL MARKS:	**75**

TABLE 1	
OVERALL COURSE OBJECTIVE:	The objective of this skill-based course is to provide knowledge on the threats and vulnerabilities to web applications and help students understand the need for secure coding practices. This is very crucial due to the dependencies of today's world on web apps and digital transactions. The course also provides details on how to secure our computer network systems from malicious activities and attacks. It also provides an overview of cyber laws, Governance & risks and threat modelling techniques.
LEARNING OUTCOME:	1. Explore cyberattacks, vulnerabilities in web applications and secure coding to prevent the vulnerabilities. Practice identification of OWASP vulnerabilities and mitigation techniques. 2. Identify threat modelling and its importance in the design of web applications 3. Explore the importance of Security Standards and Regulations, cyber laws, auditing and identity governance 4. Investigate how to secure web applications written using common programming languages

TABLE 2: MODULE WISE COURSE CONTENT AND OUTCOME				
SL.NO	**MODULE NAME**	**MODULE CONTENT**	**MODULE LEARNING OUTCOME**	**DURATION (HRS)**
1	Need for cybersecurity, threats and vulnerabilities TOC - Introduction to Cyber Security | Infosys	Recent Cyber Attacks - Cyber Security Concepts - Layers of Cyber Security - Introduction to Application	Explore cyberattacks, vulnerabilities in web applications.	8

	Springboard (onwingspan.com)	Security - OWASP Top 10 – Secure Coding Practices - Secure Design [Practical demos and code on OWASP vulnerabilities and how to mitigate them]		
2	**OWASP Top 10 vulnerabilities and secure coding practices** TOC - Introduction to Cyber Security | Infosys Springboard (onwingspan.com)	OWASP Top 10 vulnerabilities - Understand the root causes, impacts, and countermeasures – Secure coding implementation for mitigating vulnerabilities – SQL injection - Cross site scripting – server side forgery etc. [Hands on Practice]	Identify vulnerabilities in web applications and secure coding to prevent the vulnerabilities. Practice identification of OWASP vulnerabilities and mitigation techniques.	16
3	Threat Modelling & Risk assessment Threat modelling	Basics of Threat Modelling - Learn Threat Modelling with a Use Case - Tool Walkthrough - MS Threat Modelling Tool – Assignment - Threat Modelling Assessment, Risk Calculations – Risk Responses – Common Vulnerability Scoring System	Explore threat modelling and its importance in the design of web applications	4

		– Understanding Threats – NIST 800-37 Risk Management – ISO/IEC 18045 – COBIT – Risk Models		
4	**Cyber Security – Audits, Laws, Security Standards and Regulations** Cyber Security Audits Security Standards and Regulations Cyber laws Risk assessments	Cyber Security Auditing – Cyber Security Assessment – Cyber Security Reporting – Network Security Auditing – Perimeter Security Auditing – Web Application Auditing – Windows Security Monitoring and Auditing – Linux Security Monitoring – Linux Security Auditing – Cyber Security Audit Strategy – Security Audit Tools – Nessus Audit Tool, Security Standards and Regulations - PCI DSS, ISMS, FIPS, NIST Special Publications, FISMA, GDPR, HIPAA and SOX	Identify the importance of Security Standards and Regulations, cyber laws, auditing, and identity governance	6
5	**Identity Governance and**	Identity, Governance, and	Identify the importance of Security	6

	Administration Identity, Governance and Administration	administration-basics concepts –Identity Administration	Standards and Regulations, cyber laws, auditing, and identity governance	

TABLE 3: OVERALL COURSE LEARNING OUTCOME ASSESSMENT CRITERIA AND USECASES

LEARNING OUTCOME	ASSESSMENT CRITERIA	USECASES
Gain awareness on cyberattacks, vulnerabilities in web applications and securre coding to prevent the vulnerabilities. Practice identification of OWASP vulnerabilities and mitigation techniques.	Completion of the assignment	Identification of basic network components, practice commands for TCP-IP architecture and subnetting. [Reference : Lab Guide - Viewer Page | Infosys Springboard (onwingspan.com)]
		Build awareness on Defensive coding practices and control such as secure configuration, error handling, and session management, cryptography, input and output sanitization, error handling, input validation, logging and auditing, and session and exception management.[Reference: https://infyspringboard.onwingspan.com/web/en/viewer/html/lex_auth_0135015696571596809160]
		Practice defensive coding practices in C/C++ such as inspections, testing, and input validation. [Reference: Defensive Coding Fundamentals for C/C++ - Viewer Page | Infosys Springboard (onwingspan.com)
		Explore the top 10 OWASP vulnerabilities, their causes, consequences, and mitigation techniques. [Reference: OWASP Top 10: Web Application Security - Viewer Page | Infosys Springboard (onwingspan.com)],OWASP.org,http://cwe.mitre.org/top25/archive/2021/2021_cwe_top25.html. Make a report of the studied material.
		Practice secure coding techniques in Python programming language [Reference: https://infyspringboard.onwingspan.com/en/app/toc/lex_auth_01350158164493107211192/overview
		Create a login page with username and password which will connect to a database which will store the name and password. You can use Java and HTML code and database as per convenience. Simulate an SQL injection attack. Write embedded SQL code to avoid SQL injection attack. Document how this is taken care in the later versions of Java.

		Create a login page with username and password which will connect to a database which will store the name and password. You can use Python as a base and database as per convenience. Simulate an SQL injection attack. Write the revised code in Python that will sanitize the inputs and help prevent an SQL injection attack.
		Read and understand the Heartbleed vulnerability. Identify the code in C++ that can simulate this vulnerability and code to fix it. Document the secure coding practices to take care of this vulnerability and the reasons for it to happen.

SL.NO	**TABLE 4: FINAL PROJECT**
1	Given a web application, demonstrate the top 10 OWASP vulnerabilities and how to mitigate them. The steps to install the Weak application will be given as a document. The students need to mitigate at least 3 vulnerabilities.