

SECURE SOFTWARE APPLICATION

ISCED UNIT CODE: 0612554 06A

TVET CDACC UNIT CODE: SEC/CU/CS/CR/06/5/MA

Relationship to Occupational Standards

This unit addresses the unit of competency: Secure Software Application

Duration of Unit: 150 hours

Unit Description

This unit covers the competencies required to secure software application. It involves identifying software to be secured, establishing tools for application security assessment, perform application security assessment, hardening software application, monitoring application security performance and preparing of reports on software security.

Summary of Learning Outcomes

Learning Outcomes	Durations (Hours)
1. Identify software to be secured	20
2. Establish tools for application security assessment	20
3. Perform application security assessment	30
4. Harden software application	30
5. Monitor application security performance	30
6. Prepare a report on software security	20
Total Hours	150

Learning Outcomes, Content and Suggested Assessment Methods

Learning Outcome	Content	Suggested Assessment Methods
1. Identify software to be secured	1.1 Meaning of Terms 1.2 Types of software 1.3 Classification of software and their application 1.4 Factors influencing software selection 1.5 Identify Software That Needs Security 1.6 Identify existing list of installed software 1.7 Check software security updates 1.8 Research CVE Vulnerabilities for listed software	▪ Observation ▪ Written tests ▪ Oral questioning ▪ Practical tests
2. Establish tools for application security assessment	2.1 Types of tools used in software application security assessment 2.2 Assessing software application 2.2.1 Input Validation 2.2.2 Session Management 2.2.3 Error Handling 2.3 OWASP Security Knowledge framework (SKF) Threat Modelling 2.4 Perform common vulnerabilities. 2.5 Asses the security posture of a web application 2.6 Conduct security assessment using tools	▪ Observation ▪ Written tests ▪ Oral questioning ▪ Practical tests
3. Perform application security assessment	3.1 Introduction to application security 3.2 Phases of application security assessment 3.3 Reconnaissance and information	▪ Observation ▪ Written tests ▪ Oral questioning ▪ Practical tests

Learning Outcome	Content	Suggested Assessment Methods
	gathering 3.3.1 Passive information gathering 3.3.2 Active information gathering 3.4 Threat modelling 3.4.1 STRIDE model 3.4.2 PASTA model 3.5 Vulnerability Assessment 3.5.1 Manual Testing 3.5.2 Automated Scanning Tools 3.6 Exploitation and verification 3.7 Best Practices	
4. Harden software application	4.1 Introduction to Software Hardening 4.2 Basic security principles for software applications. 4.3 Software configuration 4.4 Common threats to applications. 4.5 Software Vulnerabilities 4.5.1 Injection Attacks (SQL Injection, Command Injection). 4.5.2 Broken Authentication and Session Management. 4.5.3 Cross-Site Scripting (XSS). 4.5.4 Insecure Deserialization. 4.5.5 Misconfigured Security Headers. 4.6 Security measures in software application 4.7 Hardening techniques 4.7.1 Secure coding practices 4.7.2 Applying least privilege principle 4.7.3 Secure configuration of software	▪ Observation ▪ Written tests ▪ Oral questioning ▪ Practical tests

Learning Outcome	Content	Suggested Assessment Methods
	components 4.7.4 Secure deployment and monitoring	
5. Monitor application security performance	5.1 Factors to consider in monitoring of application security performance 5.2 Implementation of monitoring solutions 5.3 Logs management and monitoring 5.4 Key Metrics to Monitor 5.4.1 Failed Login Attempts 5.4.2 Unusual API Requests 5.4.3 Changes in Application Files 5.5 Web applications logs and log management tools 5.5.1 Apache/Nginx logs -Access error and Security logs for web server Monitoring 5.5.2 IIS logs 5.5.3 ELK Stark 5.6 Advanced monitoring tools and techniques. 5.6.1 Security Information and Event Management (SIEM) tools 5.6.2 Web application firewall (WAF) and security monitoring. 5.6.3 Threat hunting with AI and Machine learning	▪ Observation ▪ Written tests ▪ Oral questioning ▪ Practical tests
6. Prepare a report on software security	6.1 Application summary 6.1.1 Overview of the application 6.1.2 Security goals	▪ Observation ▪ Written tests ▪ Oral questioning

Learning Outcome	Content	Suggested Assessment Methods
	6.1.3 Key findings 6.2 Methodology 6.2.1 Assessment approach 6.2.2 Tools used 6.2.3 Testing environment 6.3 Vulnerabilities and Risks 6.3.1 Identified vulnerabilities 6.3.2 Severity and impact 6.3.3 Risk rating methodology 6.4 Security Controls 6.4.1 Existing security measures 6.4.2 Effectiveness 6.5 Recommendations 6.5.1 Security improvements 6.5.2 Best practices 6.5.3 Remediation timeline 6.6 Conclusion 6.7 Appendices 6.7.1 Detailed findings 6.7.2 References	Practical tests

Suggested Methods of Instructions

- In Instructor led facilitation of theory
- Demonstration by trainer
- Practical work by trainee
- Viewing of related videos
- Group discussions

Recommended Resources for 25 trainees.

S/No.	Category/Item	Description/ Specifications	Quantity	Recommended Ratio (Trainee: Item)
A	Learning Materials			
21.	Textbooks		13 pcs	13:1
22.	Installation manuals		5pcs	5:1
23.	Charts			
24.	PowerPoint presentations	For trainer's use		
B	Learning Facilities & infrastructure			
25.	Lecture/theory room		1	25:1
26.	Computer Laboratory		1	25:1
27.	Internet Connection			
C	Consumable materials			
28.	Printing papers		1 ream	1:20
29.	Toners		2 pcs	13:1
30.	Assorted colour of whiteboard markers			
D	Tools and Equipment			
20.	Computers/Smartphones		25 pcs	1:1
21.	Projector		1 pc	25:1
22.	VMware/Oracle virtual box		25 pc	1:1
23.	Kali Linux or Parrot OS		25 pc	1:1