



Australian Government

Department of Education, Employment and Workplace Relations

ICAPRG507A Implement security for applications

Release: 1

ICAPRG507A Implement security for applications

Modification History

Release	Comments
Release 1	This Unit first released with <i>ICA11 Information and Communications Technology Training Package version 1.0</i>

Unit Descriptor

This unit describes the performance outcomes, skills and knowledge required to implement security for software applications. It includes code access security, security access control, cryptographic and secure input and output handling.

Application of the Unit

This unit applies to individuals responsible for coding secure software applications. These individuals may work as software developers, software engineers, system and security administrators and testers.

Licensing/Regulatory Information

No licensing, legislative, regulatory or certification requirements apply to this unit at the time of endorsement but users should confirm requirements with the relevant federal, state or territory authority.

Pre-Requisites

Not applicable.

Employability Skills Information

This unit contains employability skills.

Elements and Performance Criteria Pre-Content

Element	Performance Criteria
<i>Elements describe the essential outcomes of a unit of competency.</i>	<i>Performance criteria describe the performance needed to demonstrate achievement of the element. Where bold italicised text is used, further information is detailed in the required skills and knowledge section and the range statement. Assessment of performance is to be consistent with the evidence guide.</i>

Elements and Performance Criteria

1. Implement policy-based code-access security in an application	1.1 Demonstrate understanding of the purpose of <i>application security</i> in software development 1.2 Configure platform security configuration files using <i>security configuration tools</i> 1.3 Define a custom <i>code access permission</i> to restrict access to protected resources or run protected operations
2. Implement security access control in an application	2.1 Plan <i>authentication and authorisation strategy</i> 2.2 Develop an appropriate authentication and authorisation strategy for an application
3. Write code to encrypt and decrypt data for secure communication	3.1 Demonstrate understanding of the standard <i>cryptographic algorithms</i> 3.2 Encrypt and decrypt data using standard cryptographic algorithms
4. Protect an application against injections	4.1 Plan <i>secure input and output handling</i> to prevent vulnerabilities related to code injections 4.2 Use secure input and output handling

Required Skills and Knowledge

This section describes the skills and knowledge required for this unit.

Required skills

- analytical skills to:
 - deal with common programming security problems
 - identify security weaknesses in existing code
- literacy skills to evaluate complex and varied information and concepts in software security
- planning and organisational skills to ensure privacy for users and protect sensitive user data
- problem-solving skills to develop and refine security access control strategies
- technical skills to:
 - use security configuration tools
 - write secure code for application.

Required knowledge

- basic hardware and networking knowledge
- basic programming algorithms
- detailed knowledge of object-oriented programming
- maths at basic level.

Evidence Guide

The evidence guide provides advice on assessment and must be read in conjunction with the performance criteria, required skills and knowledge, range statement and the Assessment Guidelines for the Training Package.

Overview of assessment	
Critical aspects for assessment and evidence required to demonstrate competency in this unit	Evidence of the ability to: • create secure applications • plan security strategies • ensure safe communications • prevent security attacks.
Context of and specific resources for assessment	Assessment must ensure access to: • security configuration tools • software development environment • testing and debugging tools • network resources • appropriate learning and assessment support when required • modified equipment for people with special needs.
Method of assessment	A range of assessment methods should be used to assess practical skills and knowledge. The following examples are appropriate for this unit: • verbal or written questioning to assess candidate's knowledge of: • application security concepts • cryptographic algorithms • application protection against injections • evaluation of candidate's ability to: • create secure code and perform security configurations • apply secure communications • protect data and code.
Guidance information for assessment	Holistic assessment with other units relevant to the industry sector, workplace and job role is recommended, where appropriate. Assessment processes and techniques must be culturally appropriate, and suitable to the communication skill level, language, literacy and numeracy capacity of the candidate and the work being performed. Indigenous people and other people from a non-English speaking background may need additional support.

	In cases where practical assessment is used it should be combined with targeted questioning to assess required knowledge.
--	---

Range Statement

The range statement relates to the unit of competency as a whole. It allows for different work environments and situations that may affect performance. Bold italicised wording, if used in the performance criteria, is detailed below. Essential operating conditions that may be present with training and assessment (depending on the work situation, needs of the candidate, accessibility of the item, and local industry and regional contexts) may also be included.

<i>Application security</i> may include:	• code protection • cryptography • injections • security access control (SAC): • authentication • authorisation.
<i>Security configuration tools</i> may include:	• Java Policy Tool (policytool.exe) for setting code and principal-based security policies • .NET security configuration files: • enterprise • machine • user • .NET security configuration tools: • Caspol.exe • Mscorcfg.msc.
<i>Code access permission</i> may include:	• file system rights and authorisation • java.security.Permission or java.security.BasicPermission • NET Code Access Permission class.
<i>Authentication and authorisation strategy</i> may include:	• certificate management • login mechanism • membership provider • role-based security • user access control (UAC) • web service rights and authentication.
<i>Cryptographic algorithms</i> may include:	• asymmetric • hashes • password-based encryption • signatures • symmetric.
<i>Secure input and output handling</i> may include:	• escaping • input encoding • input validation • output encoding

	• parameterised structured query language (SQL) queries.
--	--

Unit Sector(s)

Programming and software development