Iteration Security Guideline

FIT3047/3048 | Student guidance for security evidence in each iteration

Audience	Format	Frequency
Student project teams	Formal security report in the PGP	Every iteration

1. Purpose

The Security Report provides evidence that the cumulative build has been reviewed, tested and strengthened before it is considered for deployment or client acceptance. It must address the current iteration and all functionality, fixes and data handling included in the build so far.

> **Timing:** Complete the report before or after mentor integrity testing, as directed by your mentor. Update it after testing if new issues or fixes are identified.

2. Goals and learning outcomes

By completing this activity, students will be able to:

- identify security, privacy, legal and copyright obligations relevant to a real client system;
- analyse threats, vulnerabilities, likelihood, impact and risk;
- select, implement and justify controls appropriate to the system and its users;
- conduct authorised security and penetration testing and evaluate the results;
- document evidence, remediation and remaining risk in a professional security report; and
- work constructively with other cybersecurity students while acknowledging their contribution.

3. The required deliverable

Create a formal Security Report as part of the project artefacts in the PGP. The report must be specific to your project and show what was analysed, implemented, tested and verified. A generic security checklist is not sufficient.

- Link the report and supporting security evidence from the PGP Security folder.
- Ensure the mentor's Integrity Testing Run Sheet links directly to this folder.
- Record the names and contributions of students from other teams who assisted with testing.
- Protect credentials, personal information and confidential client information; do not place secrets in publicly accessible files or repositories.

4. Security and privacy analysis

Explain how the system protects its users, information and business processes. Identify the legal, privacy, copyright and security requirements that are relevant to this client and system, and provide evidence that appropriate controls have been implemented.

Area to address	Questions and controls to consider
Data protection and access	Password hashing; authentication and authorisation; least-privilege access; users restricted to permitted data; sensitive documents protected from public access; secure session and account handling.
Input, files and application threats	Validation and sanitisation; protection from injection; safe file types, sizes, names and storage; output encoding; error handling that does not expose sensitive details.
Encryption and secrets	Encryption in transit and, where appropriate, at rest; safe storage of credentials, keys and environment settings; no secrets committed to source control.
Privacy and legal obligations	Only necessary data collected; appropriate consent, retention and deletion; lawful use of third-party content, code, licences and services; project-specific client obligations.
Deployment and configuration	Secure permissions and configuration; unsupported or unused components removed; dependencies reviewed and updated; logs and backups protected where relevant.

5. Risk analysis and solutions

Identify project-specific threats and vulnerabilities. For each risk, document the affected asset or business process, likelihood, impact, overall risk, proposed control, implementation status, evidence and any residual risk. Explain the reasoning behind the ratings and prioritise the most significant risks.

> **Required outcome:** Where a risk can reasonably be treated within the project, implement and verify the mitigation. Clearly record any risk that remains, why it remains, and what further action is recommended.

6. Authorised penetration testing

Conduct one or more appropriate, controlled security tests against your own authorised project environment. Select tests based on the system's actual threat model rather than running tools without a clear purpose.

> **Safety boundary:** Brute-force attacks are strictly prohibited. Test only systems, accounts and data for which the team has explicit permission. Do not test client production systems, third-party services or other teams' systems without documented authorisation. Stop immediately if testing could affect availability, privacy or real data.

7. Documenting testing and findings

For each security or penetration test, provide enough information for a mentor to understand and verify what was done without exposing sensitive information.

1. Purpose and scope: the risk, function, role, component or data being tested.
2. Authorisation and environment: where the test was performed and confirmation that it was permitted.
3. Method and tools: the approach, configuration and test data used.
4. Evidence and result: dated screenshots, logs or other relevant proof; pass, fail or observation.
5. Finding: the vulnerability, weakness or potential data exposure discovered.
6. Evaluation: likelihood, impact, severity and effect on the client's business or users.
7. Remediation: the control or fix implemented, with links to relevant artefacts or changes.
8. Retest and residual risk: evidence that the fix was verified and any remaining limitation.

8. Result evaluation and reinforcement

Analyse the overall testing results rather than only attaching tool output. Summarise what the results demonstrate about the build's security, which vulnerabilities were discovered, how likely and serious they are, what was fixed, and what further reinforcement is required.

9. Minimum PGP evidence

- The current formal Security Report and risk analysis.
- Evidence of implemented security and privacy controls.
- Security and penetration-test records, findings and results.
- Evidence of fixes and retesting, linked to the relevant build or Trello work.
- Acknowledgement of any students from other teams who contributed.
- Clear links that mentors can access from the Integrity Testing Run Sheet.

10. Final quality check

- ☐ The report is specific to the current cumulative build, not a generic template.
- ☐ Relevant privacy, legal, copyright and security requirements are addressed.
- ☐ Risks are rated, prioritised, mitigated where possible and supported by evidence.
- ☐ Testing was authorised, safe and appropriate; no brute-force attack was performed.
- ☐ Findings, fixes, retesting and residual risks are clearly documented.
- ☐ Sensitive information is protected and all PGP links are accessible to mentors.