Developing a Black Box Security Test Plan using Security Test Patterns

Dr. Ben Smith (IBM)

You are security experts

- Techniques
  - Penetration testing (opportunistic)
  - Automated static analysis
  - Misuse cases
  - Threat modeling
  - Planned security testing
  - ....
Software Design Patterns

**Software design pattern:** A description of a recurring problem, and a core solution to that problem, that is repeated in such a way that the solution can be used a million times over, without ever doing it the same way twice.

Software Security Test Patterns

**Software security test pattern:** A description of a generalized test case that reveals a recurring vulnerability type, that is described such that the test pattern can be instantiated a million times over, without ever doing it the same way twice.
Pattern: <Vulnerability> Tests

**Keywords:** List of words that appear in key phrases that signal the need for the use of this pattern.

**Targeted Vulnerabilities:** Vulnerability types that the type is designed to expose.

**Test Procedure Template:** A generalized form of the test case steps that should expose the targeted vulnerability.

**Expected Results Template:** The generalized expected results for a system that is not vulnerability of this type. Failures mean a vulnerability is present.

**Example Natural Language Artifact:** A natural language artifact that this pattern can be applied to.

**Example Test Procedure:** Example of the template applied to the artifact.

**Example Expected Results:** Example of the template applied to the artifact.

---

The system shall provide the ability to *generate hardcopy or electronic output of part or all of the individual patient’s medical record.*
Test Case: Ensuring the Confidentiality of Sensitive Information

**Keywords:** link, report, relay, copy, attach, identify, access, exchange…

**Targeted Vulnerabilities:** CWE311 Missing Encryption of Sensitive Data

**Test Procedure Template:**
1. Authenticate as a registered user.
2. Open the user interface for generating hardcopy or electronic output of part or all of an individual patient’s medical record.
3. If necessary, open, view, or otherwise access the generated patient medical record.
4. Log out.

**Expected Results Template:**
- The connection to the server was made using a known-safe connection protocol (e.g. HTTP over SSL).
- The patient’s medical record was encrypted with a safe encryption protocol, is password protected, or both.
- Any sensitive or personal information within the medical record is accessible only to the user who generated it, or a pre-defined set of users.

Pattern: Ensuring the Confidentiality of Sensitive Information Tests

**Keywords:** link, report, relay, copy, attach, identify, access, exchange…

**Targeted Vulnerabilities:** CWE311 Missing Encryption of Sensitive Data

**Test Procedure Template:**
1. Authenticate as a registered user.
2. Open the user interface for generating hardcopy or electronic output of part or all of an individual patient’s medical record.
3. If necessary, open, view, or otherwise access the generated patient medical record.
4. Log out.

**Expected Results Template:**
- The connection to the server was made using a known-safe connection protocol (e.g. HTTP over SSL).
- The generated object was encrypted with a safe encryption protocol, is password protected, or both.
- Any sensitive or personal information within the object is accessible only to the user who generated it, or a pre-defined set of users.
Three Types of Key Phrases

- **Key Action Phrase:** record, enter, create, update, capture, store, edit, modify, etc.
- **Key Object Phrase:** demographics, reminders for disease management, documents, files, images, credit card numbers, etc.
- **Supporting Information:** The user ID and timestamp shall be recorded.

Software Security Test Patterns

1. Attacking Session Management
2. Attacking the Authentication Mechanism
3. Testing the Safe Storage of Authentication Credentials
4. Manipulating the Redirect Header
5. Submitting a Dangerous URL
6. Uploading a Malicious File
7. Searching for Documented Passwords
8. Impersonating Trusted External Resources
9. Ensuring the Confidentiality of Sensitive Information
10. Exposing Functionality Requiring Authorization
11. Attacking Input Validation Mechanisms
MORE INFORMATION ON EACH PATTERN AT:

securitytestpatterns.org