**Julio Pochet**  
**CSD370 – Secure Software Development**  
**Module 7.2 Assignment: Security Testing**

**✅ Recommended Order of Security Tests**

1. Attack Surface Analysis
2. Scanning
3. Fuzzing
4. Penetration Testing
5. Simulation Testing
6. Testing for Failure
7. Regression Testing

**🔍 1. Attack Surface Analysis**

* **What it is:**  
  This is the first step where we figure out what parts of the system are open to the outside world, like login forms, APIs, open ports, and other places a hacker might try to mess with. It's about knowing where you could be exposed.
* **Tools:**  
  OWASP Threat Dragon  
  NIST SP 800-115
* **Expected Result:**  
  A clear list of possible attack points in your app or system.
* **Why it goes first:**  
  You need to understand what could be attacked before you try to test anything else.

**📡 2. Scanning (Vulnerability Assessment)**

* **What it is:**  
  This is when you use tools to scan your system for **known security problems**, like outdated software, open ports, or bad configuration settings. It's like doing a check-up before the real stress tests.
* **Tools:**  
  Nessus  
  OpenVAS
* **Expected Result:**  
  A report showing vulnerabilities that are already known and could be easily fixed.
* **Why now:**  
  After you know what your system exposes (from step 1), it makes sense to check those areas for obvious issues.

**🧪 3. Fuzzing**

* **What it is:**  
  Fuzzing means throwing random or weird data into your application to see how it reacts. If your app crashes or behaves strangely, it might mean there’s a deeper issue that didn’t show up during scanning.
* **Tools:**  
  AFL (American Fuzzy Lop)  
  Burp Suite Intruder
* **Expected Result:**  
  You might discover bugs that only happen with unexpected input or data.
* **Why here:**  
  Scanning checks for known problems, fuzzing helps find the unknown ones.

**🔐 4. Penetration Testing**

* **What it is:**  
  Pen testing is where you act like a hacker and actually try to break into the system using the flaws you found earlier. It’s not just about knowing there’s a vulnerability, it’s about seeing if it can be exploited.
* **Tools:**  
  Kali Linux  
  Metasploit  
  Burp Suite
* **Expected Result:**  
  Real proof that certain weaknesses can (or can’t) be exploited.
* **Why now:**  
  After finding issues, you test how dangerous they really are in a real-world scenario.

**🎯 5. Simulation Testing**

* **What it is:**  
  Simulation testing is kind of like “what if” practice—what if an attack happens or a system goes down? It helps you see how your software or team would respond in those moments.
* **Tools:**  
  MITRE ATT&CK Framework
* **Expected Result:**  
  Logs, alerts, and reactions that show the system responds correctly to threats.
* **Why here:**  
  Once you know how attacks work (from pen testing), now you simulate those scenarios to make sure the system reacts appropriately.

**⚠️ 6. Testing for Failure (Chaos Testing)**

* **What it is:**  
  This test checks how your system holds up when things break unexpectedly—like if a server goes down or if a network connection is lost.
* **Tools:**  
  Chaos Monkey
* **Expected Result:**  
  The system should stay secure, handle the error gracefully, and recover on its own if possible.
* **Why now:**  
  After security testing is done, it's important to make sure your system is also strong under random failures.

**🔁 7. Regression Testing**

* **What it is:**  
  This final step checks that nothing broke after you made changes or fixes. It also helps catch if an old bug comes back.
* **Tools:**  
  Selenium  
  JUnit  
  Jenkins
* **Expected Result:**  
  All previous test cases should pass again, and nothing should be broken or backtracked.
* **Why it goes last:**  
  It’s your final checkpoint to make sure the system is still stable and secure after everything else.

**📚 References**

OWASP. (2023). *Web Security Testing Guide (WSTG)*. <https://owasp.org/www-project-web-security-testing-guide/>

National Institute of Standards and Technology (NIST). (2008). *Technical guide to information security testing and assessment (SP 800-115)*. <https://csrc.nist.gov/publications/detail/sp/800-115/final>