Security Assessment & Testing 12%

Assessments

- **NIST 800-53A**
- **System Specifications -** policies, procedures, reqs, & designs
- **Mechanisms -** HW, SW, Firmware controls
- **Activities –** Performing backups, exporting logs, reviewing accounts
- **Individuals –** implementers of specs, mechs & activities

Audits

- **COBIT –** Control Objectives for Info & related Tech
- **Security –** Assessments by independent auditors
- **Internal –** by & for internal audiences
- **External –** auditing/accounting firms
- **3rd Party –** by or on behalf of another org

- **Type 1 -** description of controls. Single point in time. No auditing.
- **Type 2 -** min of 6 months & auditor's opinion of effectiveness.

Penetration Testing

1. **Planning** Get Permission 1st
2. **Discovery –** Footprinting & info gathering
3. **Vulnerability scanning –** ID vulnerabilities
4. **Exploitation –** attempt to gain access
5. **Report to management**

- **White Box** – deep system knowledge
- **Gray Box** – partial system knowledge
- **Black Box –** no system knowledge, external hacker sim
- **Metasploit -** automated testing

Vulnerabilities

1. **Detect**
2. **Validate**
3. **Remediate**

- **CVE -** Common Vul & Exposures
- **CVSS -** Common Vul Scoring System
- **CCE -** Common Config Enumeration
- **CPE -** Common Platform Enumeration
- **OVAL –** Open Vul & Assessment Lang

Scans

- **Network Discovery**
  - TCP Syn, TCP Connect, TCP ACK, XMAS (FIN, PSH, URG)
  - **Open –** actively accepting connections
  - **osed -** FW allowed, but no apps accept
  - **Filtered -** FW blocked
- **Network**
- **Web App**
- **DB Scans**
- **Banner grabbing –** ID variant & version running on system
- **Scanners –** Nessus, QualysGuard, NeXpose, OpenVAS

SW Testing

- **Code Review/Fagan Inspection – Plan -> Overview -> Prep ->**
- **Inspect -> Rework - > Follow-up**
- **Static –** source code, lexical, data flow
- **Dynamic –** runtime eval, synthetic transactions
- **Interface testing -** APIs, UIs, Physical interfaces
- **Use vs Misuse cases**

Fuzz

- **Mutation -** dumb, use old inputs & manipulate
- **Generational -** intelligent, data models, new inputs
- **Zzuf –** opensource, automated tool

Coverage

- **Branch -** every statement if & else
- **Condition -** logical test of inputs
- **Function -** all returned results?
- **Loop -** executed multiple times, 1x, never
- **Statement -** every line of code