Vulnerability Classification

Coleman Kane
Coleman.Kane@ge.com

January 21, 2015
Classify the nature of a vulnerability based upon the component affected. While the class will not be comprehensive, it will explain a number of common vulnerability vectors and the factors which impact discovery and remediation.
Covered vulnerability types:

- Software (userland & supervisor)
- Language design concerns
- Network / Protocol vulnerabilities
- Hardware Architecture
- Configuration
- User / Social Engineering
Key considerations:

- Sometimes, design/impl under your control
- Exposure risk can be limited to user’s access
- Code review where available
- Static analyzers, run-time profilers
- May use production environment
Key considerations:

- Rarely design/impl under your control (unless you are OS/driver dev)
- Exposure risk can lead to complete access
- Typically must recreate clone of run-time environment
Key considerations:

■ Lead to frequent unintended vulnerabilities
■ Influence coding habits / styles
■ Educational
■ Example: C vs. Ada
Network / Protocol Vulnerability

Key considerations:

- Lack of input control by default
- Streaming rather than finite data source
- Concurrency
- Pass-through / side-channel
Key considerations:

- Development cycles
- Replacement cost
- Rarely design/impl. under control
- Opacity
- Physical limitations
Key considerations:

- Low remediation cost
- Typically entirely within control
- Documentation
- Training / experience
Key considerations:

- Human Behavior
- Sophistication
- UX pain-points
- Documentation / training
- Awareness
- Skepticism
Typically comprises two or more of these vulnerability vectors. The term "exploit" is used to describe when an attacker leverages a vulnerability for unauthorized access or effect. As a noun, it is commonly used to describe a program or method which can reproducibly exploit one or more vulnerabilities.