Chapter 2: The Proactive Security Development Process

1. Why secure systems are not built
    a. Developers are ignorant of security issues (they have not been educated)
    b. It is boring
    c. It restricts functionality
    d. It inhibits performance
    e. Its benefits can be difficult to measure
2. Security should be included as an integral part of any software model and considered in all phases of that model
3. Educate
    a. Teach the ability to build secure systems – not just how to build security features into software: Security Features != Secure Features
    b. Teach how to incorporate technology to mitigate threats
    c. Teach how to alleviate security threats
    d. Teach security features and how to use them to mitigate threats
    e. How much education in school?
        i. Three semesters!
            1. general security and threat analysis
            2. understanding and applying mitigation techniques
            3. practicing and designing a real system
        ii. Doses of security theory and technology should be balanced
4. A little bit of security knowledge can go a long way in understanding vulnerabilities (C example)
    a. Many eyes on something are no good if those eyes are not trained in what to look for
5. Define product security goals: by doing this up front you avoid feature creep and YAGNI
    a. Who is the audience?
    b. What does security mean to the audience?
    c. Where will the app run/live?
    d. What needs protected?
    e. What are the implications if app is compromised?
    f. Who will manage the app?
    g. What security infrastructure services do the OS and environment already provide that can be utilized?
    h. How much of a need is there to protect the audience from their own actions?
6. Security (not a bug!) is a product feature: do not add it as an afterthought
7. Threat modeling leads to secure design
8. Features that are insecure should ultimately be retired
9. Discovery of all bugs is impossible, but many more can be identified when security is at the forefront of the development process
10. A security team should review any code before it is made public (SWI at MS)
11. Secure coding guidelines should be established
12. Internal and external code review should be done (@Stake for MS)

13. Response process to reported vulnerabilities
    a. http://www.microsoft.com/technet/security/bulletin/policy.asp for security bulletins at Microsoft
    b. http://www.wiretrip.net/rfp/policy.html for full disclosure policy
    c. Acknowledge, fix, inform customers in as timely a fashion as possible
14. Accountability: the person(s) responsible for a security flaw should be the ones required to fix it
    a. They may need some help
    b. If they aren't involved with the fix, they'll make the same mistake again