Top Design Flaws

Top Design Flaws?

- Our focus on **principles** and **rules** aims to **avoid security flaws**
  - We've seen several examples of flaws so far

- Recent launch: **IEEE Center for Secure Design**
  - Comprises top security professionals in industry, research, and government

- Center's initial focus: **What are the top security design flaws in software?**

http://cybersecurity.ieee.org/center-for-secure-design.html

Top 10 Flaws: Do not …

1. *Assume* trust, rather than explicitly give it or award it
2. Use an authentication mechanism that can be bypassed or tampered with
3. Authorize without considering sufficient context
4. Confuse data and control instructions, and process control instructions from untrusted sources
5. Fail to validate data *explicitly* and *comprehensively*
6. Fail to use cryptography correctly
7. Fail to identify sensitive data and how to handle it
8. Ignore the users
9. Integrate external components without considering their attack surface
10. Rigidly constrain future changes to objects and actors

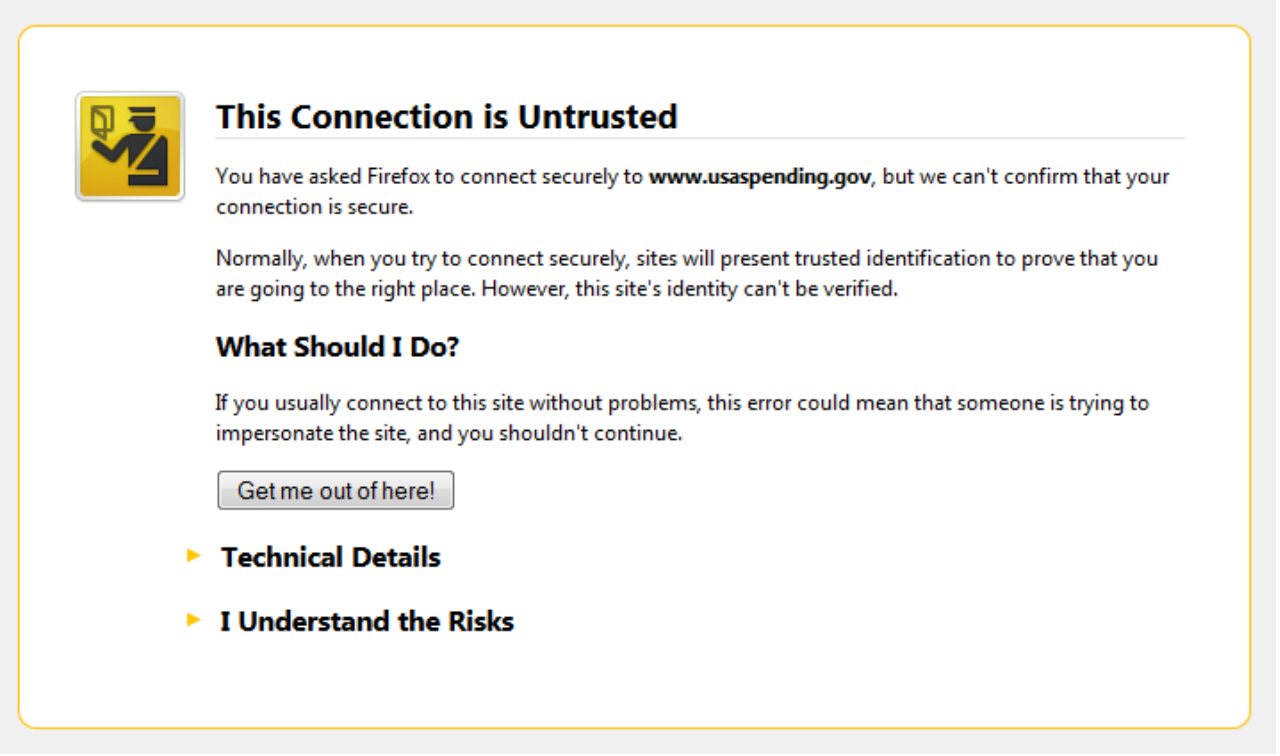
Top 10 Flaws: Do not …

1. *Assume* trust, rather than explicitly give it or award it
2. Use an authentication mechanism that can be bypassed or tampered with
3. Authorize without considering sufficie
4. Confuse data and control instructions instructions from untrusted sources
5. Fail to validate data *explicitly* and *co*
6. Fail to use cryptography correctly
7. Fail to identify sensitive data and how to handle it
8. Ignore the users
9. Integrate external components without considering their attack surface
10. Rigidly constrain future changes to objects and actors

Several of these we've covered already; will consider several in more detail

Failure: Authentication Bypass

- **Clients coerced to accept invalid SSL certificates**
  - **Bypasses client authentication of server**: Am I really talking to my bank, or a site pretending to be my bank?
- Web **browser presents a warning**
  - But how many users will "click through?"



Failure: Authentication Bypass

- **Mobile apps use SSL behind the scenes**; what happens when an app gets an invalid certificate?
  - "While it is understandable that developers **turn off SSL certificate validation** in the development phase, these developers basically forgot to remove their accept-all code when they released their apps."
    - Fahl et al, "Rethinking SSL Development in an Appified World", CCS'13 (NSA 2014 Best Cybersecurity Paper competition, honorable mention)

- **Remember: Security is *not* a feature**
  - Need to test what should *not* happen

Failure: Authentication Bypass

- **Authentication tokens with long timeouts**
  - Motivates brute-force attempts to steal session cookies
    - Recall Twitter auth_token failure from web security unit
  - But can't make it too short, or will irritate users

- **In general**: **avoid authentication bypass by developing good abuse cases**, violating assumption of unique knowledge or possession
  - How might an adversary learn a password? Spoof a biometric? Steal a session ID?

Failure: Bad (or Wrong) Crypto

- (I repeat) **Don't roll your own crypto**
  - Per use-community-resources examples: both design and implementation are hard to get right

- **Don't assume it gives you something it doesn't**
  - Encryption algorithm may protect confidentiality but not integrity. Hashing protects integrity but not confidentiality.

- **Know how to use it properly**
  - Use **properly generated keys** of **sufficient size**
  - **Protect the keys** from compromise
    - Don't hard-code them, or embed them in released binaries

Learn more!

UNIVERSITY OF MARYLAND

Cryptography

Part of the "Cybersecurity" Specialization »

This course will introduce you to the foundations of modern cryptography, with an eye toward practical applications.

Jonathan Katz

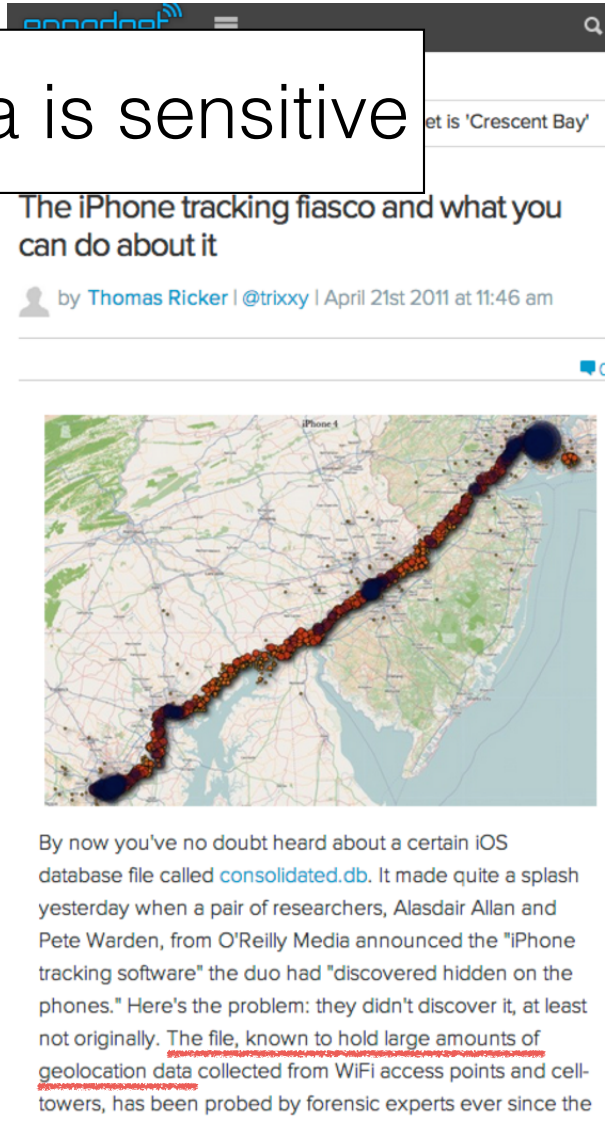
Director
Maryland Cybersecurity Center
University of Maryland, College Park

Failure: Ignore which data is sensitive

- **Think carefully about data sources**: Which require protection?
  - Personally identifiable information, sensor readings, cryptographic keys, session tokens, geolocation data, …
    - **Failure**: private data exposed to general access

- **How are these data sources exposed?**
  - When at rest, when in transmission, … what is the threat model?
    - **Failure**: embedding authentication token in exposed URL

- How does data, and its exposure, change as the **application evolves over time**?



et is 'Crescent Bay'

The iPhone tracking fiasco and what you can do about it

by Thomas Ricker | @trixxy | April 21st 2011 at 11:46 am

By now you've no doubt heard about a certain iOS database file called consolidated.db. It made quite a splash yesterday when a pair of researchers, Alasdair Allan and Pete Warden, from O'Reilly Media announced the "iPhone tracking software" the duo had "discovered hidden on the phones." Here's the problem: they didn't discover it, at least not originally. The file, known to hold large amounts of geolocation data collected from WiFi access points and cell-towers, has been probed by forensic experts ever since the

Failure: Ignore Attack Surface of External Components

- **Attack surface**: Elements of a system that an adversary can attack, or use in an attack

- **Do third-party components do *only* what I want?**
  - **Shellshock** Failure: "Bourne ag
    we
    oth
    po
    the
    - T
      n