Security and Attacks: Thinking About the Future

John Kelsey, NIST

Overview

- Why do you care about security in your cool new application?
- What can you do with existing crypto to secure your application?
- What happens when existing crypto doesn't solve the problem?

Doing something new

- The point of this workshop is how cryptography can be used for emerging technologies.
- That is, you're doing something new, and you'd like to do it securely.
- Some of the new things being discussed at the workshop:
  - Cloud applications
  - Sensor networks
  - Smart grid
  - Mobile applications

It's common to come up with reasons your application doesn't need much security

- It's a small, obscure system
- The stakes are low
- The environment is restricted and not connected to the world.
- It doesn't have to be any better than the existing (small, obscure, low-stakes) alternative

What happens if your application is wildly successful?

It expands into new environments!

- Ultimately, it lives in an environment you never imagined.
- It is used for things you never expected or imagined.
- Examples: Credit card payment system, HTTP, WEP

Successful systems become widespread

- A widespread system is a bigger target.
- *“Successful systems attract parasites.”*
- Examples:
  - Email
  - Web content and browsers
  - Smart phones/mobile apps

Successful applications live in the future.

- Attackers know more
- Processing power, memory, bandwidth cheaper.
- Attackers adapt and evolve over time.
- Example: Malware writers in 2001 vs 2011

The result is big security headaches later.

- Big installed base = hard to retrofit fixes.
- Incompatible weak and strong versions
  - SSL 2 vs 3, WEP vs WPA vs WPA2, etc.
- Changing message lengths or performance a lot can break existing applications.

Other systems can depend on your insecurity

- *Successful systems also become part of the environment*
- Other systems grown up around yours-- sometimes expect continued insecurity to keep working!
  - Example: Packet inspection vs end-to-end encryption.
  - Redirecting connections to login screen vs. HTTPS everywhere.
  - Credit card payments and merchants using CC#s as customer identifiers

Preaching to the choir....

- Security decisions made early on can stick around for a long time.
- It's usually easy to make an argument for why you shouldn't have to worry too much about security in your system.
- That usually doesn't turn out too well.

Building security in with crypto

- *Most security problems <u>can't</u> be solved with crypto....*
- *...but some can.*
- Encryption, authentication, signatures, random number generation, etc. are pretty well-understood.
- Existing standards (see Elaine Barker's talk tomorrow) provide good tools.

When should you use crypto?

- If you're sending data over a network, and it's not intended to be read by everyone, it should probably be encrypted.
- If you're sending data over a network, and it's not intended to be altered in transit, it should probably be authenticated.
- Designing in the use of crypto from beginning will save a lot of trouble later.
- The tricky part is likely to be managing the keys.

How secure do you have to be?

- Nothing new should be fielded with less than 112 bit security level.
  - 3-key triple-DES and AES
  - SHA2, SHA3
  - RSA and DSA with 2048-bit moduli
  - DSA and ECDSA with 224-bit subgroup size
- *Designing a new application with less security for performance reasons is almost certainly a huge mistake. (DES, MD5)*
- See SP800-131A for more details

What happens when things go wrong?

- Algorithms get broken (like MD5, SHA1)
  - Design your application with the ability to use different algorithms in the future
  - Just thinking about this in message formats and stuff can save a lot of heartache later.
- Protocols get broken (WEP, SSLv2, PKCS#1 v1)
  - Often, by misusing crypto algorithm in some way.
- Keys get compromised (like SecureID token)
  - Think through how your system can recover.

What if existing crypto doesn't solve your application's problems?

- Sometimes there's existing crypto but it's not in standards yet
  - IBE, short signatures, stream ciphers
- Sometimes, there's not an existing solution.
- There are lots of crypto researchers looking for problems to solve.
  - Bad news: Tend to be academics focused on proofs and papers instead of your application.
  - Good news: They're smart and motivated and may produce something useful

Summary

- Think about what security your application will need if it is very successful.
  - It is used widely
  - It expands out into new environments
  - It is used in ways you didn't expect
  - It lives in the future
  - It may become a big target
- There are nice existing crypto tools that can help design security in.