

Common Pitfalls of Key Validation

or

How to WEAKEN Security with Public Key Crypto

Andrew J. Korty
Dave Monnier
IT Security Office
Indiana University

Problems

- Users ignore certificate warnings from SSL-protected web sites
- Users ignore unknown host key warnings from SSH
- Web site maintainers use self-signed certificates without verification
- General problem: confusion between encryption and authentication (and *two-way* authentication)

Very Basic Public-key Crypto

- Every user has a keypair (public and private)
- Either key can be used to encrypt a message
- The public key decrypts messages encrypted with the private key, and vice versa
- Encrypting with the private key is called *signing* (usually, we encrypt a *hash* of the message instead)

SSL Overview

- Public-key cryptography
 - Server uses a private key to prove its identity
 - Client verifies server's identity using server's public key
- Why should we trust the public key?
 - Signed by certificate authority
 - CA certs bundled with all browsers

SSL Pitfalls

- If your browser doesn't have the CA cert or the server cert, you can't be sure of the server's identity
- Knowing the server's identity isn't the same as trusting the server
 - Untrustworthy server maintainers
 - Compromised server

SSL Solutions

- As a user:
 - Validate self-signed certs properly
 - Don't assume a site is trustworthy or secure just because you've validated its identity
- As a site maintainer:
 - Get a certificate from a well-known CA
 - If using a self-signed cert, have all your users validate it properly

Validating a Cert

- Obtain it out-of-band
 - from a floppy
 - over a separate, trusted connection
- Verify the *fingerprint*
 - from a piece of paper or business card
 - over the phone

OpenSSL Techniques

- OpenSSL (in addition to many, many other things) can generate keys, certificate requests, and signed certificates
- *Network Security with OpenSSL* (O'Reilly)
- *About Certificates* section of the *mod_ssl* FAQ
<http://www.modssl.org/docs/2.8/ssl_faq.html#ToC24>

SSH Host Key Overview

- Public key cryptography
 - Server uses a private key to prove its identity
 - Client verifies server's identity using server's public key
- Why should we trust the public key?
 - Without verification, you shouldn't
 - *Must* have a copy distributed out-of-band or verify the hash out-of-band

SSH Pitfalls

- If you haven't verified the server's host key, you can't be sure of its identity
- Knowing the server's identity isn't the same as trusting the server
 - Untrustworthy server maintainers
 - Compromised server

SSH Solutions

- As a user:
 - Validate host keys properly
 - Use strict host key checking
 - Don't assume a site is trustworthy or secure just because you've validated its identity
- As a server maintainer:
 - Distribute host keys out-of-band

Validating a Host Key

- Obtain it out-of-band
 - from a floppy
 - over a separate, trusted connection
- Verify the *fingerprint*
 - from a piece of paper or business card
 - over the phone