The Strengths of Weaker Identities: Opportunistic Personas

USENIX HotSec 2007

Mark Allman, Christian Kreibich, Vern Paxson,
Robin Sommer, Nicholas Weaver
International Computer Science Institute

"If you made them and they made you,
Who picked up the bill, who?"

Motivation

- Crypto systems are often built with the assumption that keys or certificates are tied to *authenticated identity*

- However, identities are hard...
  - to bootstrap
  - to manage

- Without hard-and-fast *identities* we can still make use of some crypto properties
  - data integrity, on-the-wire privacy

Motivation (cont.)

- So we just punt on hard-and-fast identity...

- Consider *ssh*
  - *sshd* generates a host key when first run / installed
  - on first connection the client asks user to validate host key
    - *never done!*
  - client caches the key and warns the user *when the key changes*

General Concept

- Informal crypto
- No firm notion of *authenticated identity* required
- Generate keys opportunistically == personas
- Use opportunistic persona to form an application-specific *track record*
- Manage personas by observing user reactions

Email Example

- Filtering is the only thing (sort of) saving email
- The false positive is the Achilles heel of email filters
  - Mistakenly filtering off one *crucial* email can be worse than seeing the torrent of virus / spam traffic

Email Example (cont.)

- Could develop a whitelist of email addresses

- But ...
  1. populating a whitelist is inconvenient
  2. email addresses are easy to spoof
  3. crypto identities are above most user's pain threshold

- Instead, use the informal opportunistic persona notion

Email Example (cont.)

- Sending email ...
  - When a mail client is first run / configured it generates a key-pair on user's behalf
  - All outgoing messages are signed
  - Public half of the key-pair is included in messages
    - E.g., in a new header

Email Example (cont.)

- Upon reception ...
  - Email that is correctly signed by a key on our keyring (whitelist) is not subjected to filtering
  - Email that is not correctly signed by a key on our keyring (whitelist) is processed as usual

Email Example (cont.)

- Managing the whitelist ...
  - Add keys to whitelist based on *user reaction* to email
  - E.g., if a user replies to a correctly signed email the public key is added to the user's whitelist

Email Example (cont.)

- If the machine (key) is compromised ...
  - Attacker has limited ability to bypass filtering
    - Only works against people who have whitelisted the key
  - Recipient will inevitably "junk" the incoming bogus message
    - Cue to remove the sender from the whitelist

Additional Uses

- Several others:
  - web / phishing
  - SPIM
- Speculatively, *promoting personas*

Persona Promotion

- Cheaply build a web-of-trust
- Bootstrap *authenticated identity* with personas within highly interactive apps
  - E.g., Skype
  - E.g., calendaring tools
- Leverage the fact that these tools revolve around *personal contact* to *promote* the personas to real identites

General Guidelines

- Personas vs. identities is a tradeoff
  - Personas extract strength by giving up strength
    - Greater ease-of-use and hence deployability
    - Lose ties to actual authenticated identities

General Guidelines (cont.)

- Personas open up additional attack vectors
  - Assess the implication of a compromised persona key
  - Increased opportunity for man-in-the-middle attacks?
  - Increased opportunity for social engineering attacks?

General Guidelines (cont.)

- Understand the additional work (if any) for users
- Also, need to soundly vet potential user reactions for persona management (e.g., replying to an email)

Summary

- Informal use of crypto is not ideal, but can provide for better security in real settings
- Often a hard-and-fast *track record* is enough
- We advocate developers think about how opportunistic personas might aid their applications

Summary (cont.)

Questions???