Advanced Unix System Administration

Lecture 20
April 12, 2007

Steven Luo
<sluo+decal@OCF.Berkeley.EDU>

Digital Cryptography

- What is crypto?
  - Most prominently, set of algorithms, but these algorithms are bundled in a system!
- What can crypto do for you?
  - Confidentiality – ensures that other people can't read your data
  - Authentication – ensures that a message comes from its claimed source
  - Integrity – ensures that data hasn't been modified or corrupted

Digital Cryptography

- What does crypto NOT do for you?
  - Address unrelated weaknesses in your security model
  - Fix weaknesses in your implementation of a cryptosystem
  - Prevent user errors
  - Make users actually use your security system!
- Cryptography is NOT a magic bullet!
  - Crypto is well-studied; it's usually easier to find another way in

Digital Cryptography

- Cryptographic primitives
  - Ciphers
    - Makes data unreadable except by holder(s) of a certain secret (the secret part is important!)
    - Symmetric ciphers share a secret for encryption and decryption, public-key ciphers only use a secret at decryption time
      - Public-key ciphers: RSA, ElGamal
      - Symmetric ciphers: DES, AES, RC4, ...
    - Block ciphers work on chunks of data, stream ciphers work on individual bytes
      - Block ciphers: DES, CAST family, Blowfish/Twofish, AES
      - Stream ciphers: RC4

Digital Cryptography

- Cryptographic primitives con't
  - One-way hash functions
    - Take data and produce some short stream of bytes which somehow “identifies” that data
    - Important properties:
      - Small changes in the input should produce large changes in the output (avalanche effect)
      - Collisions should be difficult to generate
      - Should be difficult to calculate the original datastream from the hash
    - Not as well-studied as ciphers
    - Examples: MD5, RIPEMD-160, SHA family

Digital Cryptography

- Applications of crypto
  - Encrypting data
    - Apply a cipher to the data
    - Doesn't provide for integrity!
  - Data verification
    - Compare the hash of the data with a known hash
    - Where does one get the hash from?
  - Digital signatures
    - Concept: encrypt something with your private key, so that people can identify it as coming from you

Digital Cryptography

- Applications of crypto con't
  - Hybrid cryptosystem
    - Hash the data, sign the hash, encrypt the data
    - Provides confidentiality, authentication, and integrity verification
  - Communications security
    - In general, you want to verify who you're talking to before you talk to them, so verify their public key
      - Has to be some infrastructure to verify this key!
    - Public-key algorithms are slow, so select a symmetric key via the PK-secured channel and switch to the symmetric cipher

Digital Cryptography

- Applications of crypto con't
  - Secure authentication
    - Lots of possible schemes:
      - Require the user to sign some data of your choosing
      - Use hash functions on a secret
      - Send a secret over an encrypted channel
    - Carelessly designed schemes can be quite insecure!
      - Just accepting the hash of a secret alone leaves you open to replay attacks
      - What happens if one of the endpoints is malicious?

Digital Cryptography

- Attacks on crypto
  - Cipher attacks
    - Data recovery: ability to read a message faster than brute force
    - Key recovery: ability to find the key used from the encrypted messages alone
  - Hash function attacks
    - Collision: generate two reasonably-related things that hash to the same value
    - Reversing the hash: find possible inputs to the hash from the output alone

Digital Cryptography

- Attacks on crypto con't
  - Brute force
    - When your secrets are too small, it's possible to reverse a computation in a “reasonable” amount of time
    - What values provide “enough” security?
      - Symmetric ciphers: 128 bits
      - Public-key ciphers: 2048 bits
      - Hash functions: 256 bits
  - Side-channel attacks
    - Lots of creative ways: measure power draw, time to run algorithm, etc.