What is Computer Security?

- Protecting computers against misuse and interference
- Broadly comprised of three types of properties
  - Confidentiality: information is protected from unintended disclosure
  - Integrity: system and data are maintained in a correct and consistent condition
  - Availability: systems and data are usable when needed
    - Also includes timeliness
- These concepts overlap
- These concepts are (perhaps) not all-inclusive
  - Spam?
  - "Non-business related" surfing?

An Example

Subject: Requesting your consent to offer professional opinion
From: mahadev satyanarayanan <mahadevsatya@yahoo.com>
Date: Sat, 5 Jan 2002 04:39:44 -0800 (PST) (07:39 EST)
To: pattrsn@cs.berkeley.edu

Dear Professor

I wish to introduce myself as an experimental computer scientist, with specialization in design, implementation, and evaluation of Systems. I have submitted my profile and a proposal by invitation for the consideration of honorary fellowship of the Jawaharlal Nehru Center for Advanced Scientific Research to support intellectual curiosity and improve technical expertise in India. I have been requested by the President JNCASR and Professor C.N.R. Rao, F.R.S to send some professional opinions for the consideration of the same. I wish to consider your name as a member of evaluation to offer your professional opinion directly on my profile and some of my referred evidence of technical accomplishment, such as: technical publications; and presentations in the design, implementation, and evaluation of systems and applications. I request you to consider that I am also a recipient of the NSF Presidential Young Investigator Award, 1987, Herbert A. Simon Award for Teaching Excellence in Computer Science, 1998, Reflections on Teaching by the Award Recipients, 1998, Carnegie Group Chair Professorship in Computer Science, 1997, Allen Newell Award for Research Excellence For CODA and ODYSSEY, 1997, Elected as a Fellow of Institute of Electrical and Electronics Engineers (IEEE) for contributions to scalable and reliable file access in large distributed systems, 2002. I shall be grateful if you kindly convey your consent early. I will submit my profile and the address for your evaluation and recommendation with your consent. Waiting for your early reply.

Yours Sincerely,
M Satyanarayanan

Types of Computer Misuse (1)

[Neumann and Parker 1989]

- External
  - Visual spying — Observing keystrokes or screens
  - Misrepresentation — Deceiving operators and users
  - Physical scavenging — "Dumpster diving" for printouts
- Hardware misuse
  - Logical scavenging — Examining discarded/stolen media
  - Eavesdropping — Intercepting electronic or other data
  - Interference — Jamming, electronic or otherwise
  - Physical attack — Damaging or modifying equipment
  - Physical removal — Removing equipment & storage media

Types of Computer Misuse (2)

[Neumann and Parker 1989]

- Masquerading
  - Impersonation — Using false identity external to computer
  - Piggybacking — Usurping workstations, communication
  - Spoofing — Using playback, creating bogus systems
  - Network weaving — Masking physical location or routing
- Pest programs
  - Trojan horses — Implanting malicious code
  - Logic bombs — Setting time or event bombs
  - Malevolent worms — Acquiring distributed resources
  - Viruses — Attaching to programs and replicating
- Bypasses
  - Trapdoor attacks — Utilizing existing flaws
  - Authorization attacks — Password cracking

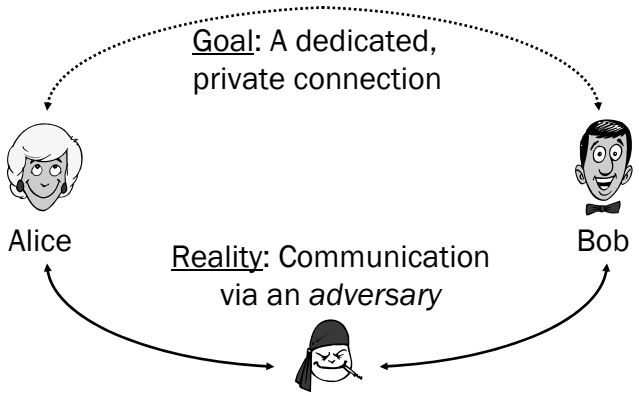
Types of Computer Misuse (3)

[Neumann and Parker 1989]

- Active misuse
  - Basic — Creating false data, modifying data
  - Denials of service — Saturation attacks
- Passive misuse
  - Browsing — Making random or selective searches
  - Inference, aggregation — Exploiting traffic analysis
  - Covert channels — Covert data leakage
- Inactive misuse — Failing to perform expected duties
- Indirect misuse — Breaking crypto keys

Cryptography

- Study of techniques to communicate securely in the presence of an *adversary*
- Traditional scenario



Adversary's Goals

1. Observe what Alice and Bob are communicating
   - Attacks on "confidentiality" or "secrecy"
2. Observe that Alice and Bob are communicating, or how much they are communicating
   - Called "traffic analysis"
3. Modify communication between Alice and Bob
   - Attacks on "integrity"
4. Impersonate Alice to Bob, or vice versa
5. Deny Alice and Bob from communicating
   - Called "denial of service"

- Cryptography traditionally focuses on preventing (1) and detecting (3) and (4)

Symmetric Encryption

- A symmetric encryption scheme is a triple $\langle \boldsymbol{G, E, D} \rangle$ of efficiently computable functions
  - G outputs a "secret key" K
    $$K \leftarrow G(\cdot)$$
  - E takes a key K and "plaintext" m as input, and outputs a "ciphertext"
    $$c \leftarrow E_K(m)$$
  - D takes a ciphertext c and key K as input, and outputs $\perp$ or a plaintext
    $$m \leftarrow D_K(c)$$
  - If $c \leftarrow E_K(m)$ then $m \leftarrow D_K(c)$
  - If $c \leftarrow E_K(m)$, then c should reveal "no information" about m

Public Key Encryption

- A public key encryption scheme is a triple $\langle G, E, D \rangle$ of efficiently computable functions
  - G outputs a "public key" K and a "private key" K^{-1}

    $$\langle K, K^{-1} \rangle \leftarrow G(\cdot)$$
  - E takes public key K and plaintext m as input, and outputs a ciphertext

    $$c \leftarrow E_K(m)$$
  - D takes a ciphertext c and private key K^{-1} as input, and outputs $\perp$ or a plaintext

    $$m \leftarrow D_{K^{-1}}(c)$$
  - If $c \leftarrow E_K(m)$ then $m \leftarrow D_{K^{-1}}(c)$
  - If $c \leftarrow E_K(m)$, then c and K should reveal "no information" about m

Message Authentication Codes

- A message authentication code (MAC) scheme is a triple $\langle G, T, V \rangle$ of efficiently computable functions
  - G outputs a "secret key" K

    $$K \leftarrow G(\cdot)$$
  - T takes a key K and "message" m as input, and outputs a "tag" t

    $$t \leftarrow T_K(m)$$
  - V takes a message m, tag t and key K as input, and outputs a bit b

    $$b \leftarrow V_K(m, t)$$
  - If $t \leftarrow T_K(m)$ then $V_K(m, t)$ outputs 1 ("valid")
  - Given only message/tag pairs $\{\langle m_i, T_K(m_i)\rangle\}_i$, it is computationally infeasible to compute $\langle m, t\rangle$ such that

    $$V_K(m, t) = 1$$

    for any new $m \neq m_i$

Digital Signatures

- A digital signature scheme is a triple $\langle G, S, V \rangle$ of efficiently computable algorithms
  - G outputs a "public key" K and a "private key" K^{-1}

    $$< K, K^{-1}> \leftarrow G(\cdot)$$
  - S takes a "message" m and K^{-1} as input and outputs a "signature" σ

    $$\sigma \leftarrow S_{K^{-1}}(m)$$
  - V takes a message m, signature σ and public key K as input, and outputs a bit b

    $$b \leftarrow V_K(m, \sigma)$$
  - If $\sigma \leftarrow S_{K^{-1}}(m)$ then $V_K(m, \sigma)$ outputs 1 ("valid")
  - Given only K and message/signature pairs $\{<m_i, S_{K^{-1}}(m_i)>\}_i$, it is computationally infeasible to compute $<m, \sigma>$ such that

    $$V_K(m, \sigma) = 1$$

    any new $m \neq m_i$

Hash Functions

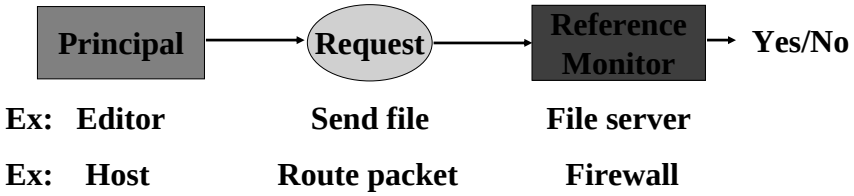
- A hash function is an efficiently computable function $\boldsymbol{h}$ that maps an input $\boldsymbol{x}$ of arbitrary bit length to an output

  $$\boldsymbol{y \leftarrow h(x)}$$

  of fixed bit length
  - Preimage resistance: Given only y, it is computationally infeasible to find any x' such that $h(x') = y$.
  - 2nd preimage resistance: Given x, it is computationally infeasible to find any $x' \neq x$ such that $h(x') = h(x)$.
  - Collision resistance: It is computationally infeasible to find any two distinct inputs x, x' such that $h(x) = h(x')$.

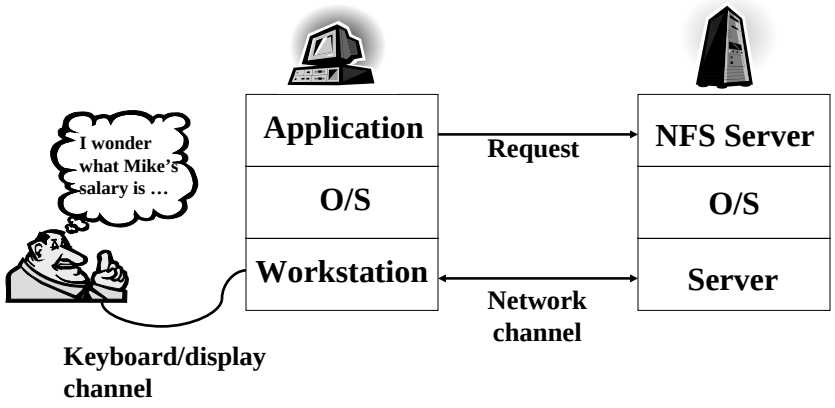
Access Control

- Principal makes a request for an object
- Reference monitor grants or denies the request



- Authentication: Determining who made request
- Authorization: Determining is trusted to access an object
  - The "decision" the reference monitor must make

The Challenge



- Who is the request "from"?
  - The user? The workstation? The application?
  - All of the above?

User Authentication

- Typically based on one or more of
  - Something you know
  - Something you have
  - Something you "are"
- "Two-factor" authentication typically refers to using two of these

Something You Know

- Password / PIN
- Social security number
- Mother's maiden name
- Pet's name
- A picture

Something You Have

- Physical key
- Proximity card
- RSA SecureID token
  http://www.rsasecurity.com/node.asp?id=1159
- Smartcard/credit card
- SecureNet token
- STU-III key
- Cell phone
- …

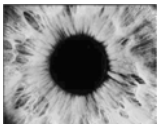
Something You Are

- Typically refers to biometrics
- Many options

Face

Fingerprint

Voiceprint

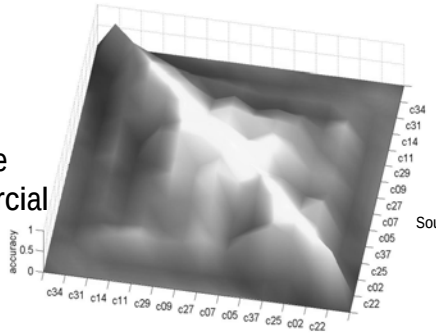


Iris

- Accuracy is more of an issue for biometrics than other user authentication technologies
  - False accepts: Accepting an authentication attempt by a person who is not the claimed person
  - False rejects: Rejecting an authentication attempt by the claimed person

The Challenge

Recognition performance of a commercial product



Source: S. Baker

Human-Generated Cryptographic Keys

- An alternative use of passwords is to generate a repeatable cryptographic key
  - Most commonly used for file encryption
  - Particularly the encryption of other keying material
- Some research has been done to generate repeatable and strong cryptographic keys from biometric information
  - Much more work left to do, though
- Key difference is the *threat model*
  - In user authentication, a trusted monitor performs the authentication and limits the number of incorrect attempts
  - In key generation, typically there is no trusted monitor to limit attempts, and so it must be computationally intractable to break

Beyond User Authentication

- User authentication is an obvious usability issue for computer systems
  - It *requires* user interaction in some form
- But it is not the only one, or even the most difficult one
- Currently there is significant debate in the community as to what extent other security mechanisms should be made visible to users or be hidden

Results from an NSF Panel on "Trust and Usability"

Usability

Usability is the extent to which users can access the functionality of a system with effectiveness, efficiency, and satisfaction to achieve specific goals. …

- Effectiveness – the degree to which a system fulfills its intended purpose and supports its users by enabling accurate and complete task performance.
- Efficiency – the resources expended by a system's users in achieving accurate and complete task performance.
- User Satisfaction – the user's perceived acceptability of the system.

Federal Aviation Administration; www.hf.faa.gov

- Note focus on "task performance" (functional properties)

Trust versus Trustworthiness

Trust is a psychological state comprising the intention to accept vulnerability based upon positive expectations of the intentions or behavior of another.

Rousseau et al. Not so different after all: A cross-discipline view of trust. *Academy of Management Review* 32(3):393-404, 1998.

Trustworthiness … asserts that the system does what is required—despite environment disruption, human user and operator errors, and attacks by hostile parties—and that it does not do other things.

Schneider, ed. *Trust in Cyberspace.* Committee on Information Systems Trustworthiness, National Research Council, 1999.

Panel Description Question 1

Is there an inherent relationship between trust and usability? Put another way are trustworthiness and usability inherently reinforcing or must they be traded off against one another?

- There are really two questions here, one pertaining to trust and one pertaining to trustworthiness
- Let's try to answer them both

Trust and Usability

- Usability promotes trust
  - Fraudsters know this well

Some people might choose to enter credit card information into a site that seems to be designed well and not into one that seems to be slapped together, making the assumption that a well-designed site costs money and could not have been afforded by a fly-by-night vendor. Because people do not spend the time and effort to investigate authenticity and the shortcut attributes that they use are well-known, they are left open to fraud at many levels.

Kent and Millett, eds. *Who Goes There? Authentication Through the Lens of Privacy* (DRAFT). Committee on Authentication Technologies and Their Privacy Implications, National Research Council, 2003.

Trustworthiness and Usability

- If a system is not usable, then it is not trustworthy
  - Example: Florida ballot in 2000 U.S. presidential election

> Soviet Union's Phobos 1 satellite … was lost on its way to Mars … It was found that the cause for the error was an operator who sent a sequence of digital commands to the satellite but mistyped a single character. … The wrong sequence set the satellite in rotation, and it was no longer possible to resume control over it; it was lost in space.
>
> Norman. Commentary: Human error and the design of computer systems. *CACM* 33:4–7, 1990. As summarized in [Kent and Millett 2003].

A Rough Claim

- Theorem ☺ : Trustworthiness ⇒ Usability ⇒ Trust
  - Implications mean slightly different things
  - Obviously a simplification, treating these notions as binary
  - Converse of the first implication doesn't hold
    - not sure about the second

- Now let's answer more questions

Are more usable devices more trustworthy …

Not necessarily.

… or more trustworthy devices necessarily more usable?

Not necessarily "more usable", but must be usable to be trustworthy.

Usability Measurement

How can we measure usability?

- Training time
- Time to reach proficiency
- Number of commands/actions per task
- Number of commands/features that are never used
- Number of times "help" is accessed
- Success vs. failure rate in task completion
- Time to complete a task
- Error recovery time
- Positive vs. negative statements recorded during observation
- …

Federal Aviation Administration; www.hr.faa.gov

Trustworthiness Measurement

How can we measure trustworthiness?

- Usability measurement, plus
- Achieved safety, liveness and security properties
  - Stronger properties increases trustworthiness
- The assumptions under which these properties are provided
  - Stronger assumptions decreases trustworthiness

This is the stuff we're good at!

- Trust measurement?
  - The degree of vulnerability that users accept

Improving Usability and Trustworthiness

What approaches are most promising for increasing the combination of these properties?

- This is where things get interesting, especially in the non-expert case
- Two schools of thought
  - Security needs to disappear
  - Security should NOT disappear, but should be presented using better metaphors
- Contrast evident in two talks at the UW-MSR-CMU summer institute on software security (June 15-18, 2003)

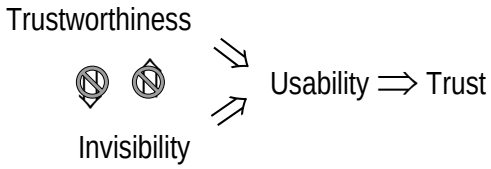
The "Security Must Disappear" Argument

[partially excerpted from D. Balfanz, *Disappearing Security*, June 2003]

- Security is hard to understand
  - What is a "public" key?
  - Does encryption make web purchases safe?
- Security is hard to use
  - What is the right Java policy file?
  - Many steps needed to get a certificate
  - Try sharing a file with (only) a group of people
- Security is annoying
  - "I can't get to your directory"
  - "I forgot my Amazon (Yahoo, E-Trade, …) password"
  - "You can't do that from behind a firewall"
- The number of devices is exploding
  - Most never see a professional admin, and so must be self-managing

The "Security Must Disappear" Argument

- We have made great strides on implementing invisible (or mostly invisible) security
  - SSH, SSL/TLS, VPNs
  - Automatic updates (e.g., Windows update)
  - Identity-based signatures and encryption
  - Wireless security tokens
- However, these sacrifice some security for the sake of invisibility in practice



The "Security Cannot Disappear" Argument

[partially excerpted from D. Simon, *A Rant About Security UI*, June 2003]

- Invisible security
  - Works only at the extremes, or at the expense of security
  - Impossible in the "fuzzy middle", where it matters
    - When is an installed/run program a "virus"?
  - Leads to things not working for reasons the user doesn't understand
- "Mostly invisible" security (augmented with "Are you sure?" warnings) yields only two realistic cases
  - Always heed the warning: same as invisible security
  - Always ignore the warning: what's the point?
- Users handle their own security in real life, all the time
  - Vehicle, home, office keys/alarms/barriers
  - Cash, checks, credit cards, ATM cards/PINs, safe deposit boxes, IDs
  - Purchases, transactions, contracts

The "Security Cannot Disappear" Argument

[partially excerpted from D. Simon, *A Rant About Security UI*, June 2003]

What works in security UI

- Clear, understandable metaphors
  - Abstract out the mechanism meaningfully for users
  - Use physical analogs where possible
- User-centric design
  - Start with the user model, design the underlying mechanism to implement it
- Unified security model
  - Across applications: "Windows GUI for security"
- Meaningful, intuitive user input
  - Don't assume things on the user's behalf—figure out how to ask so that the user can answer intelligently