Software Security: Attacks, Defenses, and Design Principles

Tadayoshi Kohno

Thanks to Dan Boneh, Dieter Gollmann, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials...
Goals for Today

- Defensive Approaches
- Cryptography Overview
Genetic Diversity

- Problems with Monoculture

- Steps toward diversity
  - Automatic diversification of compiled code
  - Address Space Randomization
Principles

- Check inputs
Principles

- Least privilege
Principles

- Check all return values
Principles

- Securely clear memory (passwords, keys, etc)
Principles

- Failsafe defaults
Principles

- Reduce size of TCB
- Simplicity
- Modularity
Principles

- Open design? Open source?
- Maybe...


Vulnerability Analysis and Disclosure

What do you do if you’ve found a security problem in a real system?

Say

- Electronic voting machine?
- Boeing 787?
- iPhone?
- School grade database?
Cryptography and Security

- Art and science of protecting our information.
- Keeping it private, if we want privacy
- Protecting its integrity, if we want to avoid forgeries.

Images from Wikipedia and Barnes and Noble
Some thoughts about cryptography

- Cryptography only one small piece of a larger system
- Must protect entire system
  - Physical security
  - Operating system security
  - Network security
  - Users
  - Cryptography (following slides)

- “Security only as strong as the weakest link”
  - Need to secure weak links
  - But not always clear what the weakest link is (different adversaries and resources, different adversarial goals)
  - Crypto failures may not be (immediately) detected

- Cryptography helps after you’ve identified your threat model and goals
RFIDs in car keys:
- RFIDs in car keys make it harder to hotwire a car
- Result: Car jackings increased

Improved security, increased risk

---

Biometric car lock defeated by cutting off owner's finger

POSTED BY CORY DOCTOROW, MARCH 31, 2005 7:53 AM |
PERMALINK

Andrei sez, "Malaysia car thieves steal finger.' This is what security visionaries Bruce Schneier and Ross Anderson have been warning about for a long time. Protect your $75,000 Mercedes with biometrics and you risk losing whatever body part is required by the biometric mechanism."

...[H]aving stripped the car, the thieves became frustrated when they wanted to restart it. They found they again could not bypass the immobiliser, which needs the owner's fingerprint to disarm it.

They stripped Mr Kumaran naked and left him by the side of the road - but not before cutting off the end of his index finger with a machete.
Key Entry Pad (4-digit PIN)

- This is the key pad on my office safe.
- Inside my safe is a copy of final exam.
- How long would it take you to break in?

✦ Answer (combinatorics):
  ✦ $10^4$ tries \textit{maximum}.
  ✦ $10^4 / 2$ tries on average.

✦ Answer (unit conversion):
  ✦ 3 seconds per try --> 4 hours and 10 minutes on average.
Key Entry Pad (4-digit PIN)

- Now assume the safe automatically calls police after 3 failed attempts.
- What is the probability that you will guess the PIN within 3 tries?
- (Assume no repeat tries.)

Answer (combinatorics):
- \( \binom{10000}{3} \) possible choices for the 3 guesses
- \( 1 \times \binom{9999}{2} \) possible choices contain the correct PIN
- So success probability is \( 3 / 10000 \)
Key Entry Pad (4-digit PIN)

- Could you do better at guessing the PIN?

- Answer (*chemical* combinatorics):
  - Put different chemical on each key (NaCl, KCl, LiCl, ...)

Image from profmason.com

Key Entry Pad (4-digit PIN)

- Could you do better at guessing the PIN?

- Answer (chemical combinatorics):
  - Put different chemical on each key (NaCl, KCl, LiCl, ...)
  - Observe residual patterns after I access safe

Image from profmason.com  
Key Entry Pad (4-digit PIN)

- Could you do better at guessing the PIN?

- Answer (chemical combinatorics):
  - Put different chemical on each key (NaCl, KCl, LiCl, ...)
  - Observe residual patterns after I access safe

Image from profmason.com

Key Entry Pad (4-digit PIN)

- Could you do better at guessing the PIN?

- Answer (chemical combinatorics):
  - Put different chemical on each key (NaCl, KCl, LiCl, ...)
  - Observe residual patterns after I access safe

Lesson: Consider the complete system, physical security, etc
Lesson: Think outside the box
Thermal Patterns

Images from http://lcamtuf.coredump.cx/tsafe/
Common Communication Security Goals

**Privacy** of data
Prevent exposure of information

**Integrity** of data
Prevent modification of information

**Adversary**
passwd = foobar; transfer $100,000

**Alice**

**Bob**
$100,000
Symmetric Setting

Both communicating parties have access to a shared random string $K$, called the key.
Asymmetric Setting

Each party creates a public key $pk$ and a secret key $sk$. 

Alice

$pk_A, sk_A$

$pk_B, sk_A$

Encapsulate

Decapsulate

Adversary

Bob

$pk_A, sk_B$

$pk_B, sk_B$
Achieving Privacy (Symmetric)

Encryption schemes: A tool for protecting privacy.

Alice

Encrypted

Bob

Adversary

Message .......... M

Ciphertext .......... C

K

K

K

K
Achieving Privacy (Asymmetric)

Encryption schemes: A tool for protecting privacy.

Message \( M \) \rightarrow \text{Encrypt} \rightarrow \text{Ciphertext} \( C \) \rightarrow \text{Decrypt} \rightarrow \text{Message} \( M \)
Achieving Integrity (Symmetric)

Message authentication schemes: A tool for protecting integrity.
(Also called message authentication codes or MACs.)
Achieving Integrity (Asymmetric)

Digital signature schemes: A tool for protecting integrity and authenticity.

Message \( M \)
Tag \( T \)

Alice \( \text{pk}_A, \text{sk}_A \)
Bob \( \text{pk}_B, \text{sk}_B \)
Adversary

Sign \( \text{pk}_A \rightarrow \text{sk}_A \rightarrow T \)
Verify \( \text{pk}_B \rightarrow \text{pk}_A \rightarrow \text{valid/invalid} \)

(\( M, T \))
Getting keys: PBKDF

Password-based Key Derivation Functions

Alice

Password → PBKDF → K (Key check value)
Getting keys: Key exchange

Key exchange protocols: A tool for establishing a share symmetric key
Getting keys: CAs

Each party creates a public key $pk$ and a secret key $sk$.

(Public keys signed by a trusted third party: a certificate authority.)
“Random” Numbers

Pseudorandom Number Generators (PRNGs)

**Alice**

- **PRNG**
  - $R_1$, $R_2$, $R_3$, $R_4$, $R_5$, ...

- **Machine State**
  - **User Input**

- **Adversary**

...
Dilbert by Scott Adams

Tour of Accounting:
Over here we have our random number generator.

Nine nine nine nine nine.

Are you sure that's random?

That's the problem with randomness: you can never be sure.
IN THE RUSH TO CLEAN UP THE DEBIAN-OPENSSL FIASCO, A NUMBER OF OTHER MAJOR SECURITY HOLES HAVE BEEN UNCOVERED:

<table>
<thead>
<tr>
<th>AFFECTED SYSTEM</th>
<th>SECURITY PROBLEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEDORA-core</td>
<td>VULNERABLE TO CERTAIN DECODER RINGS</td>
</tr>
<tr>
<td>XANDROS (EEE PC)</td>
<td>GIVES ROOT ACCESS IF ASKED IN STERN VOICE</td>
</tr>
<tr>
<td>GENTOO</td>
<td>VULNERABLE TO FLATTERY</td>
</tr>
<tr>
<td>OLPC OS</td>
<td>VULNERABLE TO JEFF GOLDBLUM’S POWERBOOK</td>
</tr>
<tr>
<td>SLACKWARE</td>
<td>GIVES ROOT ACCESS IF USER SAYS ELVISH WORD FOR “FRIEND”</td>
</tr>
<tr>
<td>UBUNTU</td>
<td>TURNS OUT DISTRо IS ACTUALLY JUST WINDOWS VISTA WITH A FEW CUSTOM THEMES</td>
</tr>
</tbody>
</table>

Source: XKCD
Kerckhoff’s Principle

- Security of a cryptographic object should depend only on the secrecy of the secret (private) key.

- Security should not depend on the secrecy of the algorithm itself.

- Why?
One-way Communications

PGP is a good example

Message encrypted under Bob’s public key
Interactive Communications

In many cases, it’s probably a good idea to just use a standard protocol/system like SSH, SSL/TLS, etc...

Let’s talk securely; here are the algorithms I understand

I choose these algorithms; start key exchange

Continue key exchange

Communicate using exchanged key
Let’s Dive a Bit Deeper
One-way Communications

*(Informal example; ignoring, e.g., signatures)*

1. Alice gets Bob’s public key; Alice verifies Bob’s public key (e.g., via CA)
2. Alice generates random symmetric keys $K_1$ and $K_2$
3. Alice encrypts the message $M$ the key $K_1$; call result $C$
4. Alice authenticates (MACs) $C$ with key $K_2$; call the result $T$
5. Alice encrypts $K_1$ and $K_2$ with Bob’s public key; call the result $D$

6. Send $D$, $C$, $T$

(Assume Bob’s private key is encrypted on Bob’s disk.)

7. Bob takes his password to derive key $K_3$
8. Bob decrypts his private key with key $K_3$
9. Bob uses private key to decrypt $K_1$ and $K_2$
10. Bob uses $K_2$ to verify MAC tag $T$
11. Bob uses $K_1$ to decrypt $C$
Interactive Communications

*Informal* example; details omitted)

1. Alice and Bob exchange public keys and certificates
2. Alice and Bob use CA’s public keys to verify certificates and each other’s public keys
3. Alice and Bob take their passwords and derive symmetric keys
   4. Alice and Bob use those symmetric keys to decrypt and recover their asymmetric private keys.
5. Alice and Bob use their asymmetric private keys and a key exchange algorithm to derive a shared symmetric key
   (They key exchange process will require Alice and Bob to generate new pseudorandom numbers)
6. Alice and Bob use shared symmetric key to encrypt and authenticate messages
   (Last step will probably also use random numbers; will need to rekey regularly; may need to avoid replay attacks,...)