Trust and Security

Matt Bishop
UC Davis

A Superhighway Like the Internet

"Think of the Internet as a highway."

There it is again. Some clueless fool talking about the "Information Superhighway." They don't know didley about the net. It's nothing like a superhighway. That's a rotten metaphor.

Suppose the metaphor ran in the other direction. Suppose the highways were like the net. . .

A highway hundreds of lanes wide. Most with pitfalls for potholes. Privately operated bridges and overpasses. No highway patrol. A couple of rent-a-cops on bicycles with broken whistles. 500 member vigilante posses with nuclear weapons. A minimum of 237 on ramps at every intersection. No signs. Wanna get to Ensenada? Holler out the window at a passing truck to ask directions. Ad hoc traffic laws. Some lanes would vote to make use by a single-occupant-vehicle a capital offense on Monday through Friday between 7:00 and 9:00. Other lanes would just shoot you without a trial for talking on a car phone.

AOL would be a giant diesel-smoking bus with hundreds of ebola victims on board throwing dead wombats and rotten cabbage at the other cars, most of which have been assembled at home from kits. Some are built around 2.5 horsepower lawnmower engines with a top speed of nine miles an hour. Others burn nitrogylcerin and idle at 120.

No license plates. World War II bomber nose art instead. Terrifying paintings of huge teeth or vampire eagles. Bumper mounted machine guns. Flip somebody the finger on this highway and get a white phosphorus grenade up your tailpipe. Flatbed trucks cruise around with anti-aircraft missile batteries to shoot down the traffic helicopter. Little kids on tricycles with squirtguns filled with hydrochloric acid switch lanes without warning.

NO OFFRAMPS. None.

Now that's the way to run an Interstate Highway system.

Outline

- What is trust?
- How does it relate to computer security?
- Trust in using a system
- Trust in administering systems
- Trust in programming

What Is Trust?

- An axiomatic acceptance of something as correct
- An acceptance of risk
- A system for drawing conclusions from axioms

Reasoning About Trust

- Belief logics
  - BAN
  - McEwen-Glasgow
- Careful how you use them!
  - Faulty premises give meaningless results

Example

Theorem: if a = b, then both are 0.5

Proof: Assume a = b. Then:

$a^2 - a^2 = a - a$	*follows from* 0 = 0
$a^2 - b^2 = a - b$	*follows from a = b*
(a + b) (a - b) = (a - b)	*factor left side*
(a + b) = 1	*divide by a - b*
2b = 1	*substituting a = b*
b = 0.5	*isolating b*
	Q.E.D.!

BAN Logic

- BAN logic proved public key authentication protocol correct
- Nessett switched roles of public, private keys so the private key was exposed
- BAN logic proved this authenticated correctly!
  - Why? Bogus assumption …

Trust in Using a System

- Trust system administrators and superusers
- Trust software
- Trust procedures

Trust in Administering a System

- Trust hardware
- Trust software
- Trust users

Trust in Programming

- Trust compilers, linkers, loaders, *etc*.
- Trust system administrators
- Trust users
- Trust hardware

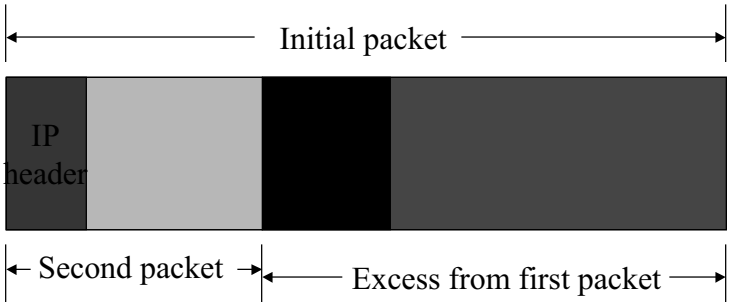
Common Themes

- Trust hardware
  - CPUs
  - Network cards
  - *etc*.
- Trust software
  - Compilers
  - Libraries
  - Loaders
  - *etc*.

Trust in Hardware

- Infamous 0xf00f instruction
  - Caused some Pentia to hang
- Error in floating-point arithmetic
  - Don't know of *security* problems from it, but easy to imagine scenarios where similar bugs could cause them
  - Intel's initial response was a *major* problem

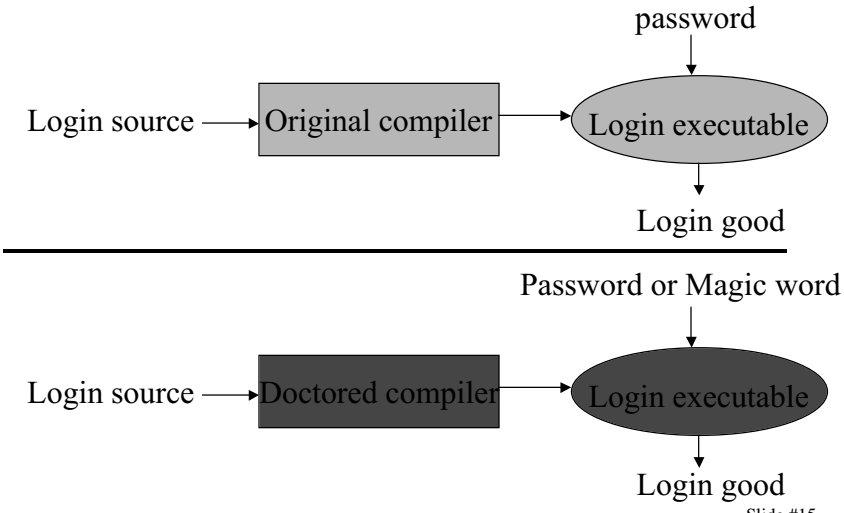
Still Trusting Hardware: Some Ethernet Cards



Trust in Software

- Vendors sell, give away buggy software
  - Human imperfection
  - Used in ways designers, *etc.* didn't intend
  - Complexity of environment makes checking hard
- Software installed, maintained incorrectly
- Users give nasty inputs
  - I consider this as buggy code, but some don't
- Modifying other system, software components may cause problems

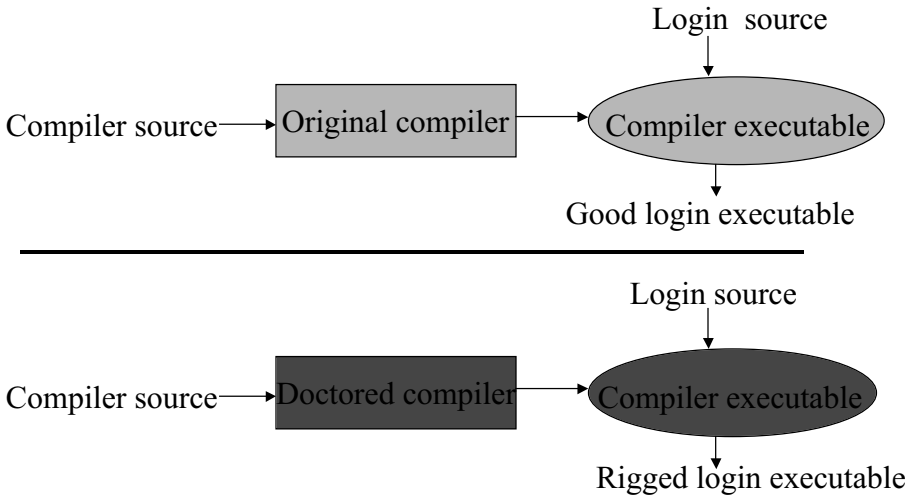
Ken Thompson's Compiler



Matt Bishop, UC Davis

Slide #15

More About the Compiler



Matt Bishop, UC Davis

Slide #16

RSAREF Library

- Basis for many implementations of RSA encipherment, authentication
- In US, had to use RSAREF library for patent reasons
- RSAREF library had at least one buffer overflow

Dynamic Loaders

When stub invoked, calls a loader

- Some implementations use PATH, IFS
  - Indirect, by spawning shell to run loader
- Define your own loader and have privileged program invoke it!

Example

```
$ cat > ld.so
#! /bin/sh
id
$ PATH=:$PATH; export PATH
$ chmod 755 ./ld.so
$ privileged_program
uid=bishop(917) euid=root(0)
$
```

More Dynamic Loaders

Load the wrong library

- As before but use LD_PATH, LD_PRELOAD, or LD_LIBRARY
  - Exact name of variable depends on your system
  - Doesn't work if kernel fixed
  - May work if path is defined in a world-writable file
    - Check your /etc, /usr/etc, … directory!

End of Dynamic Loaders

Modification changes how routine works

- Enciphered, authenticated tunnels protected traffic
- Worked for static host lookup but broke for dynamic host lookup
- Problem: socket hooked up to stdout, stderr, output packetized and enciphered, and resolver wrote “No such host” into enciphered stream
- Fix: rearrange packetizing code to handle this

Moral

If there were in justice in the world, "trust" would be a four-letter word.

Contact Information

Matt Bishop
Computer Security Laboratory
Department of Computer Science
University of California, Davis
Davis, CA 95616-8562

Phone: (530) 752-8060
Fax: (530) 752-4767
Email: bishop@cs.ucdavis.edu
WWW: http://seclab.cs.ucdavis.edu/~bishop