The Future of Internet Security

Steven M. Bellovin
http://www.cs.columbia.edu/~smb
Columbia University

October 4, 2007

The Past

Almost 20 Years Ago

```
      X
     X X
    X X X
   X X X X
  X X X X X
 X X X X X X
X X X X X X X
      X
      X
      X
```

```
A very happy Christmas and my best wishes for
the next year. Let this run and enjoy yourself.
Browsing this file is no fun at all.
Just type Christmas.
```

The IBM Christmas Card “Virus”

- A simple executable script
- When invoked, it displayed an animated Christmas tree
- It also scanned the user’s alias file for the addresses of contacts, and sent out a copy of itself to each one
- Sound familiar?

The Internet Worm

- 11 months later, it happened again
- A worm was unleashed that spread by many different mechanisms, including password-guessing and buggy software
- Multi-protocol, multi-platform — took out *lots* of machines

What Have We Learned?

What Have We Learned?

The Past
Almost 20 Years Ago
The IBM Christmas Card “Virus”
The Internet Worm
What Have We Learned?
What Would Have Stopped Those?
Human Nature
Security Standards
Older and Wiser Heads
These Attacks Weren’t New
Hoare’s Turing Award Lecture, 1980
The Present
The Future

?

What Would Have Stopped Those?

What Would Have Stopped Those?

?

Human Nature

- There are no patches available for *H. sapiens* 1.0

Human Nature

- There are no patches available for *H. sapiens* 1.0
- The release date for 2.0 has not been announced

Human Nature

- There are no patches available for *H. sapiens* 1.0
- The release date for 2.0 has not been announced
- There are no betas in sight

Security Standards

- The U.S. Defense Department had been worried about computer security since at least the 1970s
- They developed a series of standards, most notably the *DoD Trusted Computer System Evaluation Criteria*
- Intended for 1970s-style time-sharing systems. . .

Older and Wiser Heads

Me "I just reread the Orange Book; nothing in it would have prevented the worm.

Older and Wiser Heads

Me "I just reread the Orange Book; nothing in it would have prevented the worm.

Him "No, that's not so; a B2 system would have stopped it."

Older and Wiser Heads

Me	“I just reread the Orange Book; nothing in it would have prevented the worm.
Him	“No, that’s not so; a B2 system would have stopped it.”
Me	“How so?”

Older and Wiser Heads

The Past
Almost 20 Years Ago
The IBM Christmas Card "Virus"
The Internet Worm
What Have We Learned?
What Would Have Stopped Those?
Human Nature
Security Standards
Older and Wiser Heads
These Attacks Weren't New
Hoare's Turing Award Lecture, 1980

The Present

The Future

Me "I just reread the Orange Book; nothing in it would have prevented the worm.

Him "No, that's not so; a B2 system would have stopped it."

Me "How so?"

Him "B2 requires a thorough search for bugs."

Older and Wiser Heads

The Past
Almost 20 Years Ago
The IBM Christmas Card "Virus"
The Internet Worm
What Have We Learned?
What Would Have Stopped Those?
Human Nature
Security Standards
Older and Wiser Heads
These Attacks Weren't New
Hoare's Turing Award Lecture, 1980

The Present

The Future

Me "I just reread the Orange Book; nothing in it would have prevented the worm.

Him "No, that's not so; a B2 system would have stopped it."

Me "How so?"

Him "B2 requires a thorough search for bugs."

Me "Oh. . ."

These Attacks Weren’t New

- I saw my first fake login prompt (not “login screen”!) in 1969
- I caught my first hackers in 1971
- My friends and I discussed “rabbit jobs” circa 1972, and we knew we weren’t inventing them
- Hoare warned about buffer overflows as a security problem in his 1980 Turing Award lecture

Hoare's Turing Award Lecture, 1980

The first principle was security: . . . A consequence of this principle is that every occurrence of every subscript of every subscripted variable was on every occasion checked at run time against both the upper and the lower declared bounds of the array. . . . I note with fear and horror that even in 1980, language designers and users have not learned this lesson. In any respectable branch of engineering, failure to observe such elementary precautions would have long been against the law.

The Present

Progress?

- We see phishing, come-and-get-it attacks, and pump-and-dump spam
- We still see buffer overflows
- The only reason we don't see more "bring down the Net" worms is because they're bad for (evil) business
- Have we made progress?

Problem Areas

- Most security problems are due to buggy code
- Very few problems are due to lack of cryptography
- The most important form of cryptography is authentication, but presenting the results to people is very hard

Threat Models

- The military has always been worried about serious enemies
- Most software, though, has been built for casual attacks
- The Internet grew up in an era of "joy hackers" — mostly bored kids doing it for fun
- That's changed...

Systems Are Somewhat Better

- I'm no longer surprised to see a computer that's been up for a year or more
- We no longer reboot our computers daily because they really need it; even ordinary desktops can (usually) survive overnight
- But it doesn't seem to take much of a push to knock things over

Why Do Problems Seem Worse?

- Systems are more complex; the complexity has grown faster than our technology
- The net has gotten much bigger
- The threat model has changed

Fudd’s First Law of Opposition

The Past

The Present

Progress?

Problem Areas

Threat Models

Systems Are Somewhat Better

Why Do Problems Seem Worse?

Fudd’s First Law of Opposition

The Future

“Push something hard enough and it will fall over.”

Firesign Theater, 1971

The Future

Where Now?

- We're not going to fix human nature
- We're not going to abolish buggy software
- Is there a solution?

Where Now?

- We're not going to fix human nature
- We're not going to abolish buggy software
- Is there a solution?
- Not a perfect one, but we can (probably) achieve "good enough"

Changing Threat Models

- Most hacking now is done for profit
- The spammers pay the hackers
- The password theives pay the hackers
- The stock swindlers pay the hackers
- Most major militaries have cyberwarefare/cyberespionage arms

What Won’t Do It

- The problem isn’t the Internet architecture
- Vint Cerf, Robert Kahn, et al., weren’t idiots
- Assertion: *any network with power comparable to the Internet’s will have most of its problems*

- The problems are on the hosts and with the users

Strategies

- Paying for good code
- Human-centered design
- Proper cryptography
- System architecture
- Harden the right pieces

How is it Done?

- Disciplined software engineering
- We must understand our goals
  ⇒That may be “keep flying” or “keep the switch up”
- Understand that people are part of the system
- The software architecture must be aimed first at those goals

High-Quality Software

- We've been able to produce high-quality software when people cared enough (and paid enough)
- Yes, there have been plenty of problems with phone switches, avionics, and the like, but on the whole such software has a much better track record
- What does it take?

The Human Element

- Human beings are an essential part of our computer systems
- We must design our systems to be resilient in the face of human error
- We must also design them to minimize such errors

Words of Wisdom

"If we assume that the people who use technology are stupid ('Bubbas') then we will continue to design poorly conceived equipment, procedures, and software, thus leading to more and more accidents, all of which can be blamed upon the hapless users rather than the root cause – ill-conceived software, ill-conceived procedural requirements, ill-conceived business practices, and ill-conceived design in general."

Don Norman, *Risks Digest 23.07*, 2003

Proper Cryptography

- The hard problem isn't the primitives
- We have more than enough protocols to solve today's problems
- Cryptography has to be easy to use
- Ideally, it should be transparent to users
- It should be decentralized and not require complex infrastructure
- It shouldn't create more human vulnerabilities

Architecture

- We are never going to have absolutely-correct programs
- We can design systems to protect what really counts
- This requires a change in architecture
- Coding practices, no matter how good, won't help

There Isn't One Architecture

- A home LAN isn't a bank
- A bank isn't the CIA
- A hospital needs very strong protection of patient records — but has to be able to cope with unconscious patients who can't give consent
- SCADA networks need strong isolation, but they also need to permit access from proper points in the corporate net
- One design doesn't fit all

Design Principles

- Some components *will* fail
- The bigger the code base, the more likely the failure
- Identify and isolate crucial sections
- Design the *system* to minimize the impact of a failure
- Detect the problem and recover

One Oversimplified Example

The Past
The Present
The Future
Where Now?
Changing Threat Models
What Won't Do It
Strategies
How is it Done?
High-Quality Software
The Human Element
Words of Wisdom
Proper Cryptography
Architecture
There Isn't One Architecture
Design Principles
One Oversimplified Example
Caveats
Cautious Optimism

- For an e-commerce site, store credit card numbers on a seperate machine
- Two per-account commands can be sent: store credit card #N; pay for this order with card #N
- No command to read card numbers
- Authentication from end user to database
- Result: security problem probably doesn't result in bulk compromise of card numbers

Caveats

Caveats

- “You can’t make systems foolproof because fools are so smart”

Caveats

The Past
The Present
The Future
Where Now?
Changing Threat Models
What Won't Do It
Strategies
How is it Done?
High-Quality Software
The Human Element
Words of Wisdom
Proper Cryptography
Architecture
There Isn't One Architecture
Design Principles
One Oversimplified Example
Caveats
Cautious Optimism

- "You can't make systems foolproof because fools are so smart"
- You have to *know* what really counts

Caveats

- "You can't make systems foolproof because fools are so smart"
- You have to *know* what really counts
- You'll still get it wrong sometimes — but less often

Cautious Optimism

- I think we can do better (or at least well enough)
- It won't be easy, it won't be cheap, and it won't involve business as usual
- We can do this — but will we?