Schneider on Security

The Security Mindset

Bruce Schneier

The Security Mindset

Blog
Newsletter
Home
Profile
Search
About Bruce

Related Entries

Top Security Mistakes

Features

News Security World Example

Search

Prepared by Bruce Schneier

About Bruce Schneier

E-mail: Bruce.Schneier@Schneier.com

TODAY'S PAPERS

Why Information Security Is Hard

An Economic Perspective

Ross Anderson

University of Cambridge Computer Laboratory, JJ Thomson Avenue, Cambridge CB3 0FD, UK

Abstract

According to one common view, information security comes down to technical measures. Given better access control policies, more careful people can protect graphical protocols, acceptable browsing, better ways of verifying information and reducing useless code, and better tools for system examination and assurance, the problems can be solved.

In this note, I put forward a contrary view: information security is at least as much due to social incentives. Many of the problems can be explained more clearly and convincingly using the language of microeconomics, network externalities, asymmetric information, moral hazard, adverse selection, negative dumping, and the tragedy of the commons.

1 Introduction

In a survey of fraud against automated machines [4], it was found that patterns of fraud depended on who was liable for them. In the US, if a customer disputes a transaction, the bank is on the hook to prove that the customer was mistaken or wrong; this gave US banks a motive to protect their systems. But in Britain, Norway and the Netherlands, the burden of proof lay on the merchant: the bank was right unless the customer could prove it wrong. When this was the situation, banks in these countries became cautious. Eventually, episodes of fraud demobilized their empathy. US banks meanwhile suffered much less fraud: although they actually spent less money on security than their European counterparts, they spent it more effectively [4].

There are many other examples. Medical payment systems that are paid for by insurers rather than by hospitals fail to prevent patient privacy whenever this conflicts with the insurer's wish to collect information about the claims. Digital signature laws transfer the burden of forged signatures from the bank that relies on the signature and that built the system to the person who has made the signature. Common Criteria evaluations are not made by the relying party, as Orange Book evaluations were, but by a commercial facility paid by the vendor. In general, the house that is in a position to audit a system is not the house that would suffer the results of security failure. Thus problems may be expected.

A different kind of innovative failure occurred in early 2000, when all international financial services against a number of high-profile web sites. AJAX\footnote{An applicable number of web sites to search a large coordinated attack.} is a risk of forged signatures from an attack that relies on the signature, and that built the system.

An economic perspective

Ross Anderson

University of Cambridge Computer Laboratory, JJ Thomson Avenue, Cambridge CB3 0FD, UK

Search

Prepared by Bruce Schneier

TODAY'S PAPERS

Why Information Security Is Hard

An Economic Perspective

Ross Anderson

University of Cambridge Computer Laboratory, JJ Thomson Avenue, Cambridge CB3 0FD, UK

Abstract

According to one common view, information security comes down to technical measures. Given better access control policies, more careful people can protect graphical protocols, acceptable browsing, better ways of verifying information and reducing useless code, and better tools for system examination and assurance, the problems can be solved.

In this note, I put forward a contrary view: information security is at least as much due to social incentives. Many of the problems can be explained more clearly and convincingly using the language of microeconomics, network externalities, asymmetric information, moral hazard, adverse selection, negative dumping, and the tragedy of the commons.

1 Introduction

In a survey of fraud against automated machines [4], it was found that patterns of fraud depended on who was liable for them. In the US, if a customer disputes a transaction, the bank is on the hook to prove that the customer was mistaken or wrong; this gave US banks a motive to protect their systems. But in Britain, Norway and the Netherlands, the burden of proof lay on the merchant: the bank was right unless the customer could prove it wrong. When this was the situation, banks in these countries became cautious. Eventually, episodes of fraud demobilized their empathy. US banks meanwhile suffered much less fraud: although they actually spent less money on security than their European counterparts, they spent it more effectively [4].

There are many other examples. Medical payment systems that are paid for by insurers rather than by hospitals fail to prevent patient privacy whenever this conflicts with the insurer’s wish to collect information about the claims. Digital signature laws transfer the burden of forged signatures from the bank that relies on the signature, and that built the system to the person who has made the signature. Common Criteria evaluations are not made by the relying party, as Orange Book evaluations were, but by a commercial facility paid by the vendor. In general, the house that is in a position to audit a system is not the house that would suffer the results of security failure. Thus problems may be expected.

A different kind of innovative failure occurred in early 2000, when all international financial services against a number of high-profile web sites. AJAX\footnote{An applicable number of web sites to search a large coordinated attack.} is a risk of forged signatures from an attack that relies on the signature, and that built the system.

An economic perspective

Ross Anderson

University of Cambridge Computer Laboratory, JJ Thomson Avenue, Cambridge CB3 0FD, UK

Search

Prepared by Bruce Schneier

TODAY'S PAPERS

Why Information Security Is Hard

An Economic Perspective

Ross Anderson

University of Cambridge Computer Laboratory, JJ Thomson Avenue, Cambridge CB3 0FD, UK

Abstract

According to one common view, information security comes down to technical measures. Given better access control policies, more careful people can protect graphical protocols, acceptable browsing, better ways of verifying information and reducing useless code, and better tools for system examination and assurance, the problems can be solved.

In this note, I put forward a contrary view: information security is at least as much due to social incentives. Many of the problems can be explained more clearly and convincingly using the language of microeconomics, network externalities, asymmetric information, moral hazard, adverse selection, negative dumping, and the tragedy of the commons.

1 Introduction

In a survey of fraud against automated machines [4], it was found that patterns of fraud depended on who was liable for them. In the US, if a customer disputes a transaction, the bank is on the hook to prove that the customer was mistaken or wrong; this gave US banks a motive to protect their systems. But in Britain, Norway and the Netherlands, the burden of proof lay on the merchant: the bank was right unless the customer could prove it wrong. When this was the situation, banks in these countries became cautious. Eventually, episodes of fraud demobilized their empathy. US banks meanwhile suffered much less fraud: although they actually spent less money on security than their European counterparts, they spent it more effectively [4].

There are many other examples. Medical payment systems that are paid for by insurers rather than by hospitals fail to prevent patient privacy whenever this conflicts with the insurer’s wish to collect information about the claims. Digital signature laws transfer the burden of forged signatures from the bank that relies on the signature, and that built the system to the person who has made the signature. Common Criteria evaluations are not made by the relying party, as Orange Book evaluations were, but by a commercial facility paid by the vendor. In general, the house that is in a position to audit a system is not the house that would suffer the results of security failure. Thus problems may be expected.

A different kind of innovative failure occurred in early 2000, when all international financial services against a number of high-profile web sites. AJAX\footnote{An applicable number of web sites to search a large coordinated attack.} is a risk of forged signatures from an attack that relies on the signature, and that built the system.
THE SECURITY MINDSET

To anticipate attackers we must be able to think like attackers

Uniquely identifiable liquid

= Proof of ownership

What would an attacker do?

Paint it on someone else’s property and then call the cops
THE SECURITY MINDSET

To anticipate attackers we must be able to think like attackers

Fill out a card with your address

⇒

They deliver a box of live ants to you

What would an attacker do?

Order them to someone else
THE SECURITY MINDSET

The ability to view a large, complex system and be able to reason about:

- What are the potential security threats?
- What are the hidden assumptions?
- Are the explicit assumptions true?
- How can we mitigate the risks of the system?

Be creative! (Attackers will be)
WHAT DOES IT MEAN TO BE SECURE?

There is no such thing as security, only degrees of insecurity.

Goal: Raise the bar for the attacker

- Too difficult
- Too expensive
- Lower ROI than the next target

Ultimately, we want to mitigate undesired behavior
WHAT ARE "UNDESIRERD" BEHAVIORS?

- Reveals info users wish to hide (confidentiality)
  - Corporate secrets
  - Private data; personally identifying information (PII)

- Modifies information or functionality (integrity)
  - Destroys records
  - Changes data in-flight (think "the telephone game")
  - Installs unwanted software (spambot, spyware, etc.)

- Denies access to a service (availability)
  - Crashing a website for political reasons
  - Denial of service attack
  - Variant: fairness

This is a subset
ATTACKS ARE COMMON
WHY ARE ATTACKS COMMON?

• Security is a property of the systems we build.

• Many attacks begin by exploiting a vulnerability.
  • Vulnerability = defect in hw, sw, protocol, design, ... that can be exploited to yield an undesired behavior.
  • Software defect = the code doesn’t “behave correctly”

• Defects arise due to
  • flaws in the design and/or
  • bugs in the implementation.
HEARTBLEED

- SSL is the de facto protocol for secure online communication
- Heartbleed was a vulnerability in the most popular SSL server
  - A malformed packet allows you to see server memory
- Fix: don’t let the user just tell you how much data to give back
- This was a design flaw
HEARTBLEED

HOW THE HEARTBLEED BUG WORKS:

USER MEG WANTS THESE 6 LETTERS: POTATO. USER LIZA WANTS PAGES ABOUT "IRL GAMES".
UNLOCKING SECURE RECORDS WITH MASTER KEY 5130985733433.
MAYBE REMEMBER TO SEND THIS MESSAGE: "POTATO".
Server, are you still there? If so, reply "BIRD" (4 letters).

Olivia from lockdown wants pages about "the bees in car why". Note: Files for IP 375.381.383.17 are in /tmp/files-3843. User Meg wants these 4 letters: BIRD. There are currently 348 connections open. User Brendan uploaded the file selfie.jpg (contents: 334ba962e25cbb9ff89b131ff9).

Hmm...

BIRD
HEARTBLEED

User passwords, private keys, personal information...

~40% of “secure” web servers vulnerable
1. **Carefully crafted Flash program.** When run by the vulnerable Flash player, allows the attacker to execute arbitrary code on the running machine.

2. This program could be **embedded in an Excel spreadsheet**, and run automatically when the spreadsheet was opened.

3. Spreadsheet **attached to an email**, masquerading as a trusted party (“spearphishing”)
   - You can forge any “From” address
WHY ARE ATTACKS COMMON?

• Because attacks derive from design flaws or implementation bugs

• But all software has bugs: so what?

• A normal user never sees most bugs
  • Post-deployment bugs are usually rare corner cases

• Too expensive to fix every bug
  • Only fix what’s likely to affect normal users
WHY ARE ATTACKS COMMON?

*Attackers are not normal users*

- Normal users avoid bugs/flaws
- Adversaries seek them out and try to *exploit* them

*This extends beyond software:*
Attacks are possible even with perfect software
WHY ARE ATTACKS COMMON?

Because it’s **profitable**

And because a system is **only as secure as its weakest link**

Figure 1: Infrastructure involved in a single URL’s value chain, including advertisement, click support and realization steps.
In order to achieve security, we must:

Be able to eliminate bugs and design flaws and/or make them harder to exploit.

Be able to think like attackers.

Develop a foundation for deeply understanding the systems we use and build.
50% of Android apps that use crypto encrypt in this manner.
UNDERSTANDING THE SYSTEMS WE USE

Three things all vulnerable websites should have done:

- Patch software: 93%
- Reissue new certificate: 27%
- Revoke old certificate: 13%

Long expiration times:
We will be dealing with Heartbleed for years
WHY IS SECURITY DIFFICULT?

Security is indeed a matter of technical reasons.

But “insecurity is at least as much due to perverse incentives”

<table>
<thead>
<tr>
<th>Percentage</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>27%</td>
<td>Reissue new certificate</td>
</tr>
<tr>
<td>13%</td>
<td>Revoke old certificate</td>
</tr>
</tbody>
</table>

Some certificate authorities give certificates for free but charge to revoke.
TOPICS OF THIS CLASS
ETHICS IN SECURITY RESEARCH

QUESTION How do we perform research such that the benefit to society outweighs the risk?

PAPERS “Encore” and “All your contacts…”
MEMORY SAFETY

QUESTION How can we safely store and process user input?

ATTACKS Software stores user input in memory. The attacker exploits this to inject code, exfiltrate data, etc.

DEFENSES Detect disallowed memory reads/writes
Taint tracking to find unintended info leakage

PAPERS Smashing the stack
Flesh on the bone
EXE Stackguard
Taint tracking
CFI
WEB SECURITY

**QUESTION**
How can we protect users from malicious websites & malicious users on benign websites?

**ATTACKS**
- Upload malicious data (XSS, CSRF, SQL injection)
- Attack visual integrity (clickjacking)

**DEFENSES**
- Secure state shared between site & user (cookies)
- Add protections at large hosting providers (CDNs)

**PAPERS**
- SQL Injection
- Clickjacking
- Defenses for CSRF
- Secure delivery networks
**USABLE SECURITY**

**QUESTION**

How do we properly account for humans? What can we expect them (not) to do?

**ATTACKS**

- password, 123456 (sigh)
- Spearphishing, bad interfaces

**DEFENSES**

- Improve understanding of user abilities/limitations
- Better interfaces and detection of attacks

**PAPERS**

- Password reuse
- Spearphishing
- Users are not the enemy
- Why Johnny can’t encrypt
**ISOLATION**

**QUESTION** How can we safely share computing resources between benign and malicious users?

**ATTACKS** Side-channel attacks
Rowhammer (exploits hardware feature)

**DEFENSES** Close side-channels
Sandboxes

**PAPERS** “Get off my cloud”
Rowhammer
Native Client
Chromium browser
Malware

**Question**
How can we detect and mitigate malicious software? What does it do? Who does it?

**Attacks**
Viruses, worms, botnets. Various attack vectors (how it infects) and payloads (what it does)

**Defenses**
Detection of malware through signatures, metadata, and driveby download nets

**Papers**
Hunting for metamorphic
Ghost in the browser
Inside Slammer
How to Own the internet
Who is actually launching these attacks? What are the weak points in these economies?

Figure 1: Infrastructure involved in a single URL’s value chain, including advertisement, click support and realization steps.

PAPERS  
Click trajectories  
Show me the money
**GOAL**

A black-box approach: this is not a crypto class
How to use it properly, how TLS works

**QUESTIONS**

Why does crypto fail in practice?
How do we use these building blocks to build more complicated systems?

**PAPERS**

<table>
<thead>
<tr>
<th>TLS/SSL</th>
<th>Diffie-Hellman atk</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTPS</td>
<td>Most dangerous code…</td>
</tr>
</tbody>
</table>
MEASURING CRYPTO USE IN PRACTICE

QUESTION
How is crypto being misapplied or mismanaged?

PAPERS
Measurements of the cert ecosystem
Crypto misuse in Android apps
NEW CRYPTO MECHANISMS

GOAL
Understand how to apply cryptographic techniques to build new systems

MECHANISMS
Property-preserving encryption
Group signatures
Blockchains

PAPERS
CryptDB
Attacking CryptDB
ANONYMITY

QUESTIONS
What is anonymity?
How can we achieve it?
How can we make it usable?

SYSTEEMS
Tor, Mixnets
Dining cryptographers (DCNets)

ATTACKS
Fingerprinting attacks on Tor
Nation-state attackers

PAPERS
Tor
Mixnets
Users get routed
Fingerprinting
CENSORSHIP RESISTANCE

QUESTIONS
Can we allow users to communicate despite powerful attackers trying to stop them? How does this relate to anonymity?

SYSTEMS
Decoy routing (now “refraction routing”)  
Alibi routing, DeTor

REPORTS
“Enemies of the Internet”  
by Reporters Without Borders
NETWORK SECURITY

QUESTIONS
What can an attacker learn about two communicating hosts?

ATTACKS
Malicious VPN apps (get on the path)
Off-path TCP attacks (side-channel attacks)

PAPERS
Off-path TCP exploits
Measurement of VPN apps
BOTNETS

IMPORTANCE
Botnets are a new, powerful force
All the more important due to IoT

QUESTIONS
How do they operate?
What do they do?
How do we measure them?

PAPERS
Your botnet is my botnet
Understanding Mirai
DENIAL OF SERVICE (DOS) ATTACKS

QUESTIONS
How do we launch them?
How do we detect/measure them?
How do we stop them?

PAPERS
OptACK
Inferring DoS activity
IP Traceback
TVA