Security Engineering (1)

Email: christian.urban at kcl.ac.uk
Office: S1.27 (1st floor Strand Building)
Slides: KEATS

This is a Misconception!

Without encryption:

With encryption:

There is some consensus that the NSA can probably not brute-force magically better than the "public".

The content of this course is very much inspired by the work of three people:

Bruce Schneier

en.wikipedia.org/wiki/Bruce_Schneier

Ross Anderson

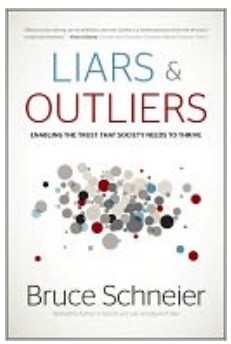
www.cl.cam.ac.uk/~rja14

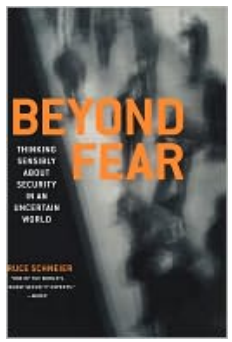
Alex Halderman

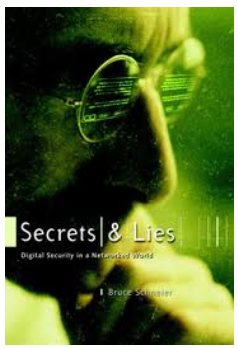
jhalderm.com

Security engineers require a particular **mindset**:

> "Security engineers — at least the good ones — see the world differently. They can't walk into a store without noticing how they might shoplift. They can't use a computer without wondering about the security vulnerabilities. They can't vote without trying to figure out how to vote twice. They just can't help it."
>
> —Bruce Schneier

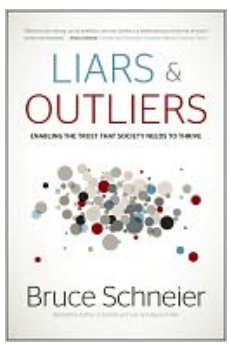


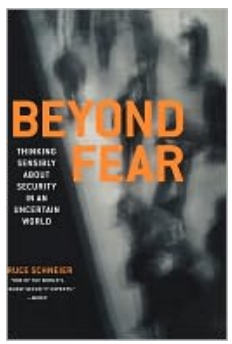


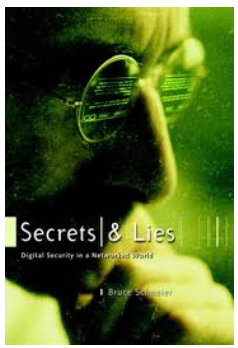


Security engineers require a particular **mindset**:

"Security engineering...requires you to think differently. You need to figure out not how something works, but how something can be made to not work. You have to imagine an intelligent and malicious adversary inside your system ..., constantly trying new ways to subvert it. You have to consider all the ways your system can fail, most of them having nothing to do with the design itself. You have to look at everything backwards, upside down, and sideways. You have to think like an alien." —Bruce Schneier







Breaking Things

For example:

Prof. V. Nasty gives the following final exam question (closed books, closed notes):

Write the first 100 digits of π:

3._____

How can you cheat in this exam and how can you defend against such cheating?

Warning

- I will be teaching techniques that can be used to compromise security and privacy.

Warning

- I will be teaching techniques that can be used to compromise security and privacy.

- Don't be evil!

Warning

- I will be teaching techniques that can be used to compromise security and privacy.

- Don't be evil!
- Using those techniques in the real world may violate the law or King's rules, and it may be unethical.
- Under some circumstances, even probing for weaknesses of a system may result in severe penalties, up to and including expulsion, fines and jail time.
- Acting lawfully and ethically is your responsibility.

Warning

- I will be teaching techniques that can be used to compromise security and privacy.

- Don't be evil!
- Ethics requires you to refrain from doing harm.
- Always respect privacy and rights of others.
- Do not tamper with any of King's systems.

Warning

- I will be teaching techniques that can be used to compromise security and privacy.

- Don't be evil!
- If you try out a technique, always make doubly sure you are working in a safe environment so that you cannot cause any harm, not even accidentally.
- Don't be evil. Be an ethical hacker.

Secure Systems

For a secure system, four requirements need to come together:

- **Policy**

  What is supposed to be achieved?
- **Mechanism**

  Cipher, access controls, tamper resistance, ...
- **Assurance**

  The amount of reliance you can put on the mechanism.
- **Incentive**

  The motive that the people guarding and maintaining the system have to do their job properly, and also the motive that the attackers have to try to defeat your policy.

Chip-and-PIN

- Chip-and-PIN was introduced in the UK in 2004
- before that customers had to sign a receipt

- **Is Chip-and-PIN a more secure system?**

(some other countries still use the old method)

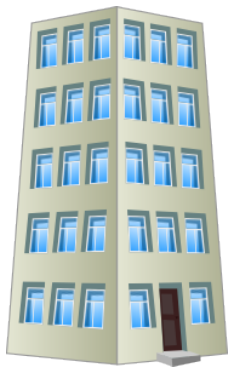
Yes ...

...if you believe the banks:

> "Chip-and-PIN is so effective in this country [UK] that fraudsters are starting to move their activities overseas,"
>
> said some spokesman for Lloyds TSB
> (in The Guardian, 2006)

- mag-stripe cards cannot be cloned anymore
- stolen or cloned cards need to be used abroad
- fraud on lost, stolen and counterfeit credit cards was down £60m (24%) on 2004's figure

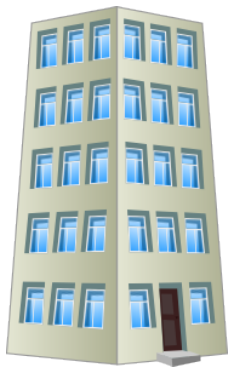
But let's see



Bank

customer / you

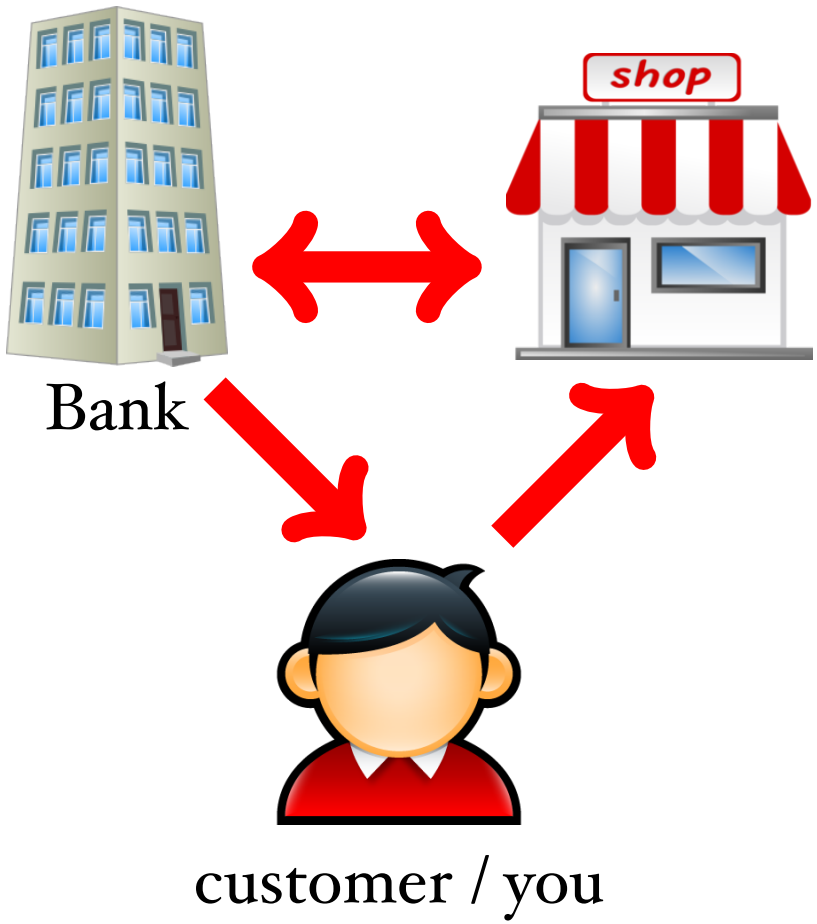
But let's see



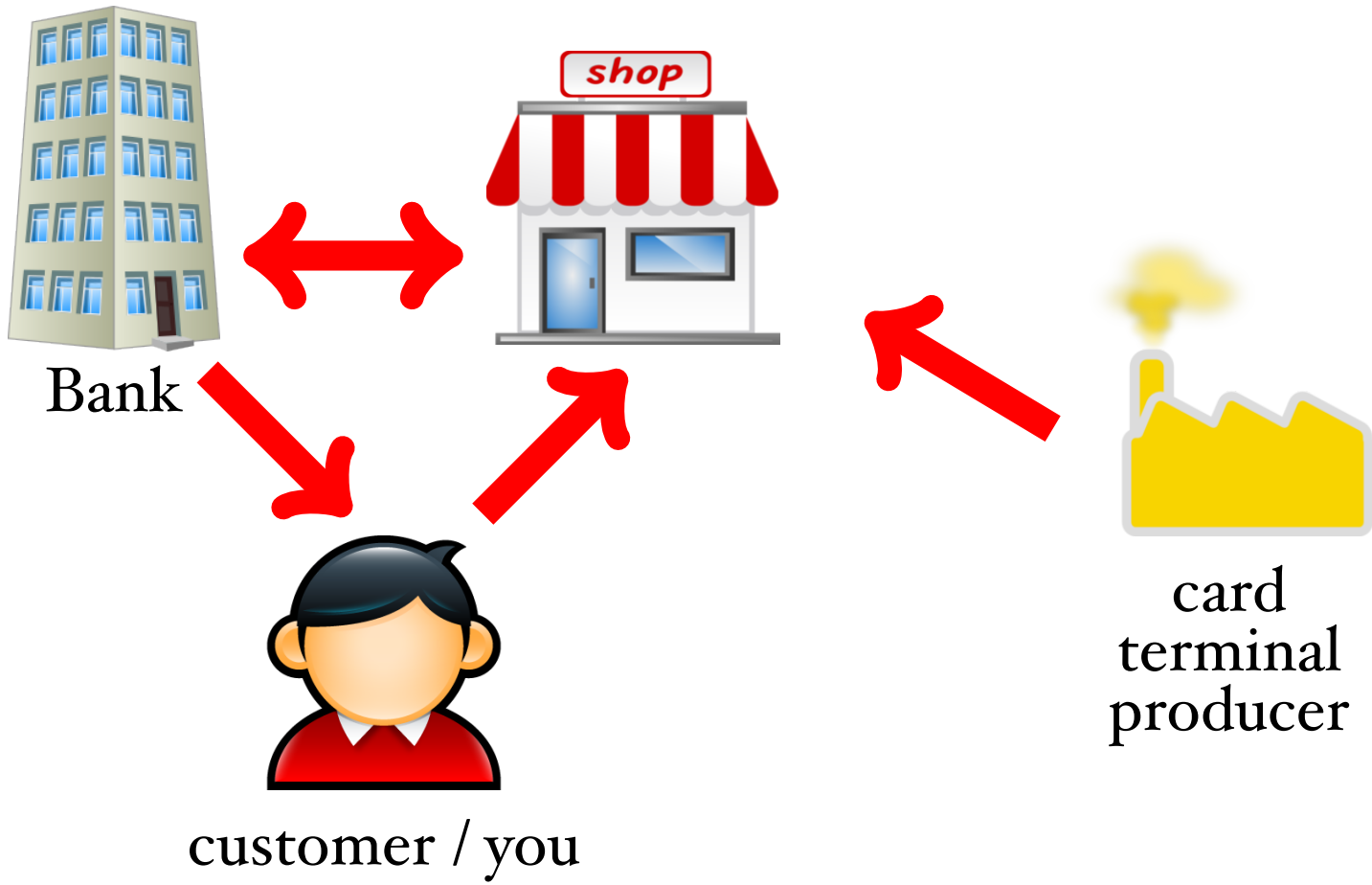
Bank

customer / you

But let's see



But let's see



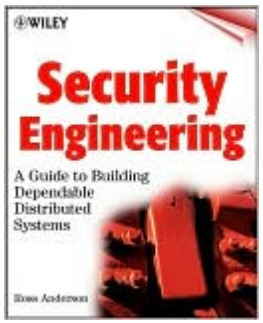
Chip-and-PIN

- A "tamperesitant" terminal playing Tetris on youtube.
  (http://www.youtube.com/watch?v=wWTzkD9M0sU)

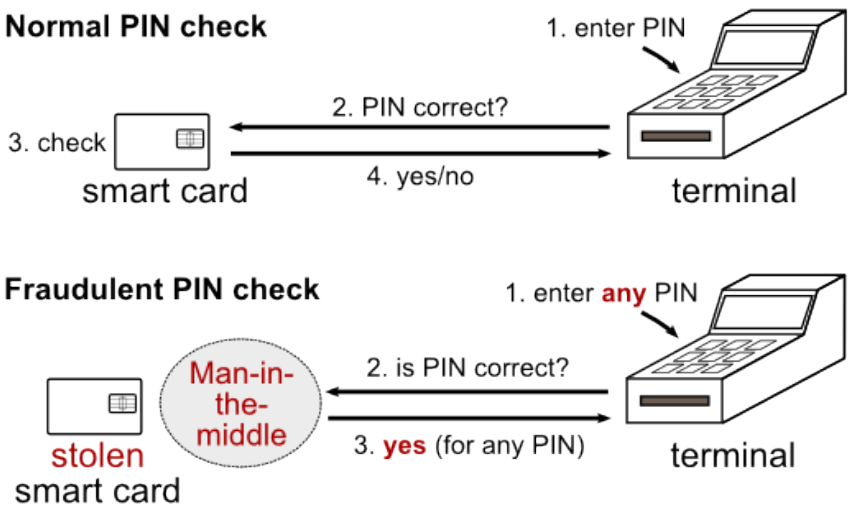
Chip-and-PIN

- in 2006, Shell petrol stations stopped accepting Chip-and-PIN after £1M had been stolen from customer accounts
- in 2008, hundreds of card readers for use in Britain, Ireland, the Netherlands, Denmark, and Belgium had been expertly tampered with shortly after manufacture so that details and PINs of credit cards were sent during the 9 months before over mobile phone networks to criminals in Lahore, Pakistan

Chip-and-PIN is Broken

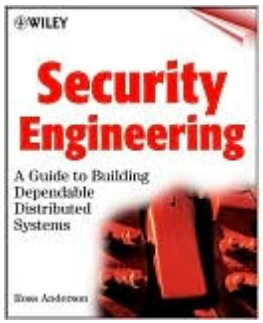


- man-in-the-middle attacks by the group around Ross Anderson



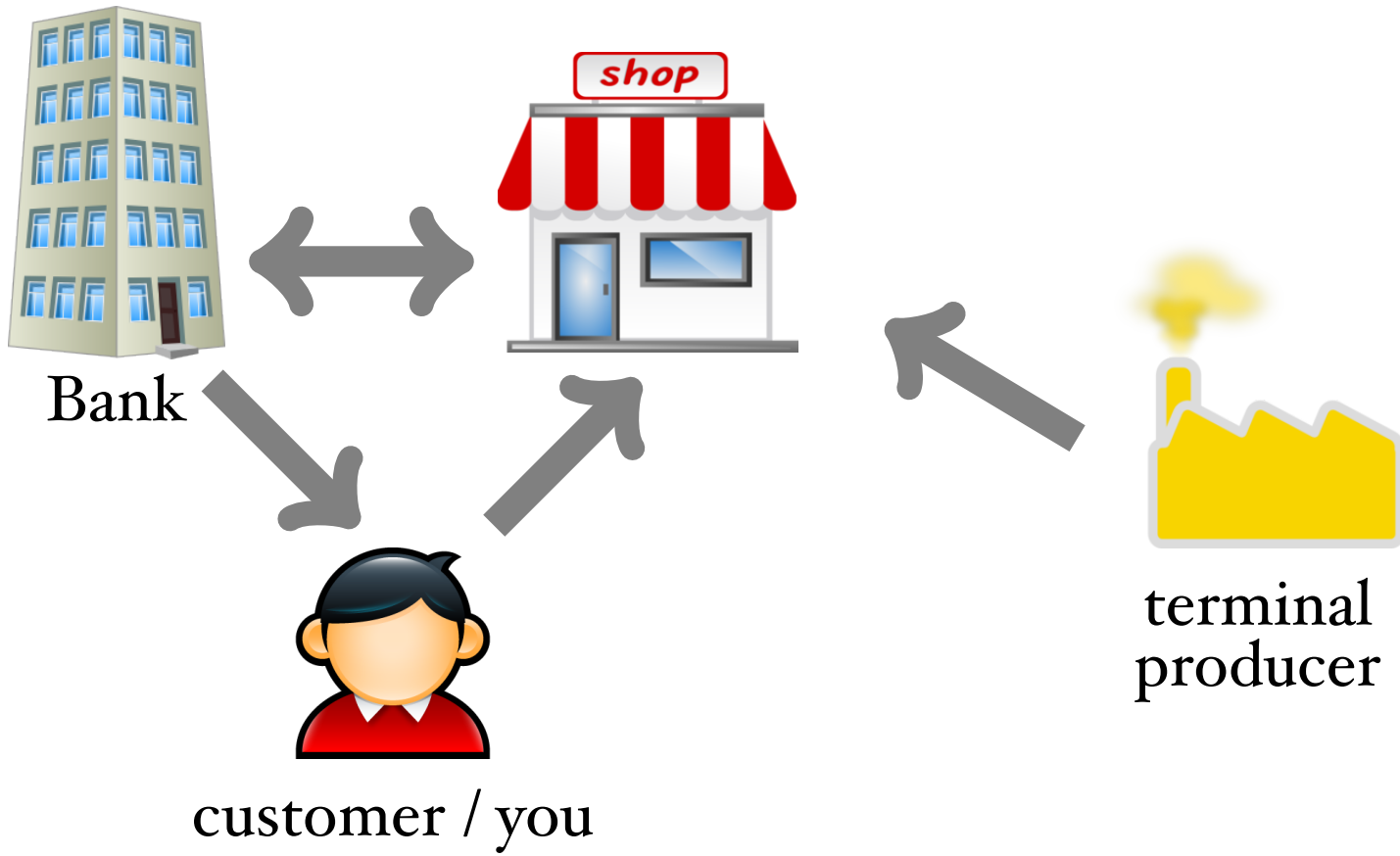
on BBC Newsnight in 2010 or youtube

Chip-and-PIN is Really Broken



- same group successfully attacked in 2012 card readers and ATM machines
- the problem was: several types of ATMs generate poor random numbers, which are used as nonces

The Real Problem ...



- the burden of proof for fraud and financial liability was shifted to the costumer (until approx. 2009/10)

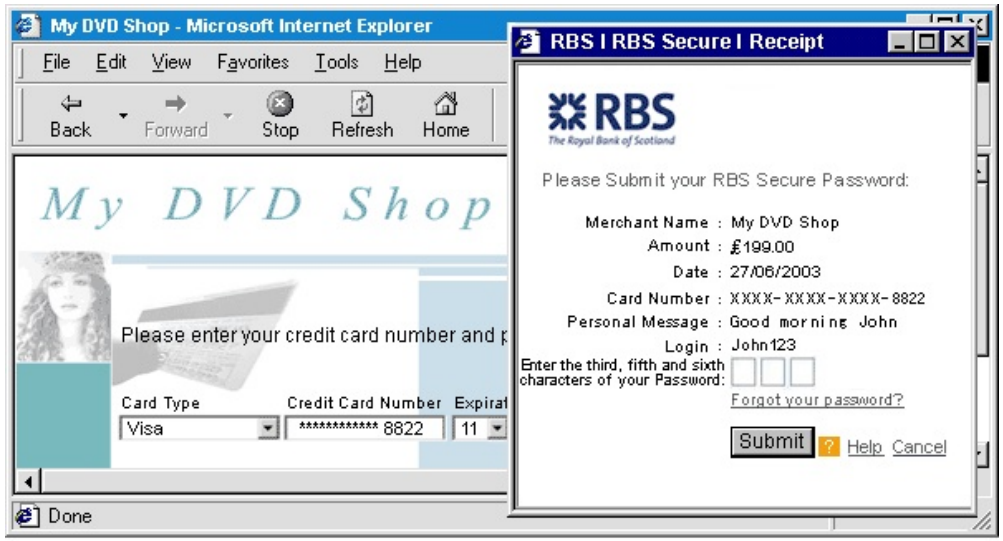
The Bad Guy Again

The anonymous hacker from earlier:

"Try to use 'Verified-By-Visa' and 'Mastercard-Securecode' as rarely as possible. If only your CVV2 code is getting sniffed, you are not liable for any damage, because the code is physically printed and could have been stolen while you payed with your card at a store. Same applies if someone cloned your CC reading the magnetic stripe or sniffing RFID. Only losing your VBV or MCSC password can cause serious trouble."

www.goo.gl/UWluh0

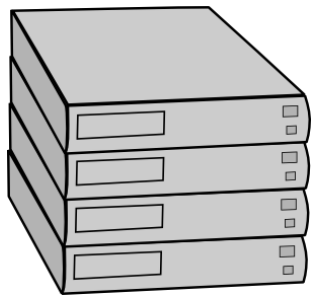
Being Screwed Again



- **Responsibility**
  "You understand that you are financially responsible for all uses of RBS Secure."

  `https://www.rbssecure.co.uk/rbs/tdsecure/terms_of_use.jsp`

Web Applications

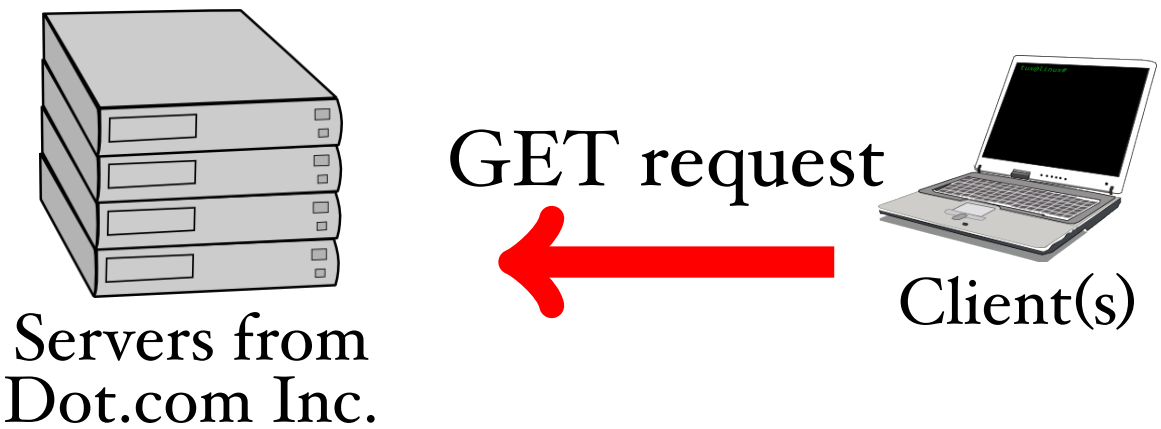


Servers from
Dot.com Inc.

Client(s)

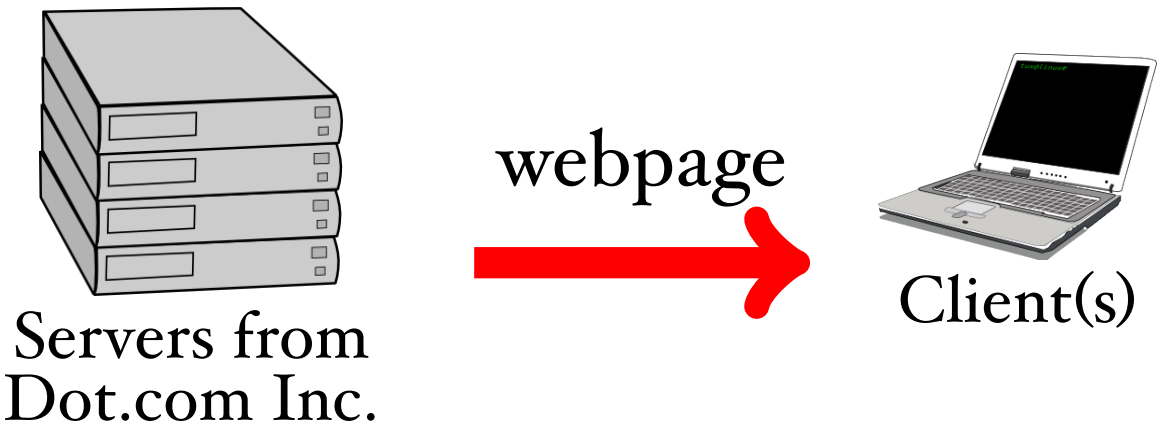
- What are pitfalls and best practices?

Web Applications



- What are pitfalls and best practices?

Web Applications



- What are pitfalls and best practices?

Web Applications

- What are pitfalls and best practices?

JavaScript + Node.js

A simple response from the server:

```
var express = require('express');
var app = express();

app.get('/', function(request, response){
    response.write('Hello World');
    response.end()
});

// starting the server
app.listen(8000);
```

JavaScript + Node.js

A simple response from the server:

```
var express = require('express');
var app = express();

app.get('/', function(request, response){
    response.write('Hello World');
    response.end()
});

// starting the server
app.listen(8000);
```

alternative response:

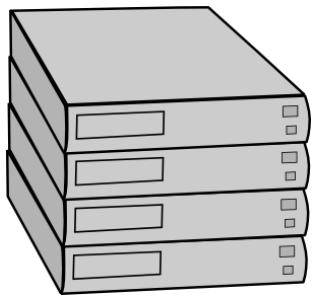
```
response.write('<H1>Hello World</H1>');
```

```
var express = require('express');
var bodyParser = require('body-parser');
var app = express();
app.use(bodyParser.urlencoded({ extended: true }));

// sending the form
app.get('/', function(req, res){
    var html = '<form action="/" method="post">' +
        'Login:    <input type="text" name="login" /><br>' +
        'Password: <input type="password" name="pass" /><br>' +
        '<button type="submit">Submit</button></form>';
    res.send(html);
});

// receiving data
app.post('/', function(req, res){
    var html = 'Received login: ' + req.body.login + '<br>' +
        'Received password: ' + req.body.pass + '<br>' +
        '<a href="/">Try again</a>';
    res.send(html);
});
```

Cookies

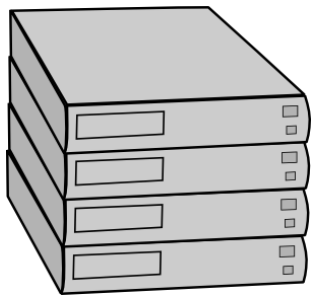


Servers from
Dot.com Inc.

GET request

Client

Cookies



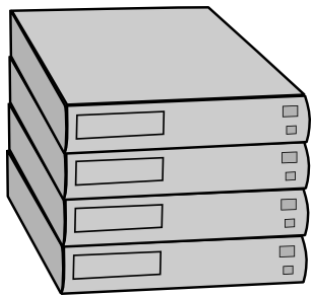
Servers from
Dot.com Inc.

GET request

read a cookie

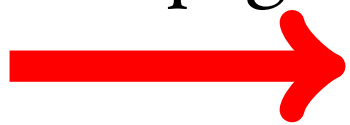
Client

Cookies



Servers from
Dot.com Inc.

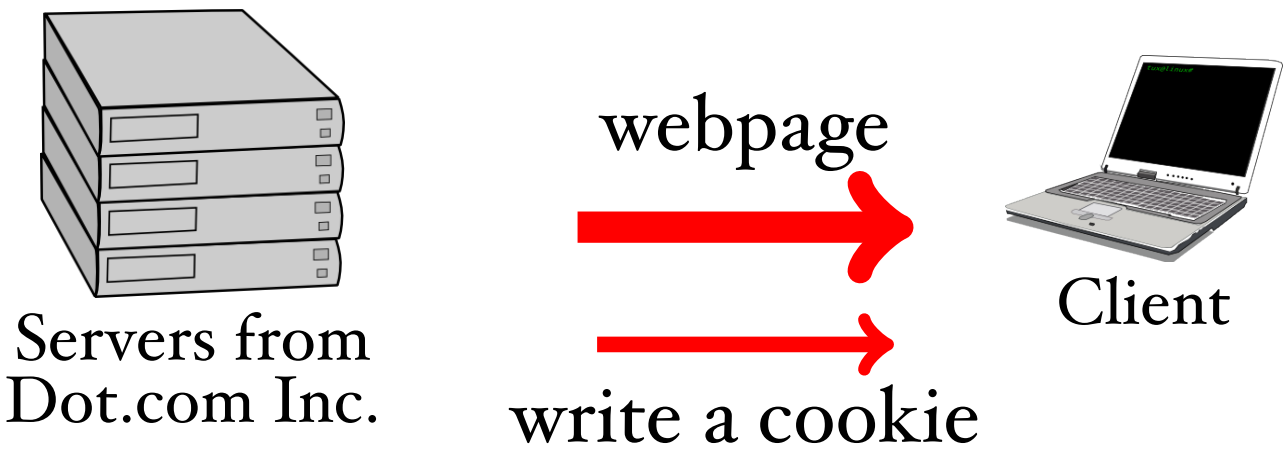
webpage



Client

write a cookie

Cookies



- cookies: max 4KB data
- cookie theft, cross-site scripting attacks
- session cookies, persistent cookies, HttpOnly cookies, third-party cookies, zombie cookies

Cookies

EU Privacy Directive about Cookies:

"In May 2011, a European Union law was passed stating that websites that leave non-essential cookies on visitors' devices have to alert the visitor and get acceptance from them. This law applies to both individuals and businesses based in the EU regardless of the nationality of their website's visitors or the location of their web host. It is not enough to simply update a website's terms and conditions or privacy policy. The deadline to comply with the new EU cookie law was 26th May 2012 and failure to do so could mean a fine of up to £500,000." →BBC News, www.goo.gl/RI4qhh

- session cookies, persistent cookies, HttpOnly cookies, third-party cookies, zombie cookies

My First Real Webapp

GET request:

1. read the cookie from client
2. if none is present, set `counter` to zero
3. if cookie is present, extract `counter`
4. if `counter` is greater or equal than 5,
   print a valued customer message
   otherwise just a normal message
5. increase `counter` by 1 and store new cookie with client

```
var express = require('express');
var cookie  = require('cookie-parser')

var app = express();
app.use(cookie());

app.get('/', function(req, res){
    var counter =  parseInt(req.cookies.counter) || 0;
    res.cookie('counter', counter + 1);
    if (counter >= 5) {
        res.write('You are a valued customer ' +
                  'visting the site ' + counter + ' times.');
    } else {
        res.write('This is visit number '+ counter +'!');
    }
    res.end();
});

app.listen(8000);
```

- data integrity needs to be ensured

```
function mk_hash(s) {
    return crypto.createHash('sha1').update(s).digest('hex')
}

function mk_cookie(c) {
    return c.toString() + "-" + mk_hash(c.toString())
}

function gt_cookie(s) {
    var splits = s.split("-", 2);
    var counter = parseInt(splits[0])
    if (mk_hash(counter.toString()) == splits[1]) {
        return counter
    } else { return 0 }
}

app.get('/', function(req, res){
    var counter = gt_cookie(req.cookies.counter) || 0;
    res.cookie('counter', mk_cookie(counter + 1));
    ...
});
```

SHA-1

- SHA-1 is a cryptographic hash function (MD5, SHA-256, SHA-512, ...)
- message $\rightarrow$ digest
- attacks exist: $2^{80} \rightarrow 2^{61}$

SHA-1

- SHA-1 is a cryptographic hash function (MD5, SHA-256, SHA-512, ...)
- message $\rightarrow$ digest
- attacks exist: $2^{80} \rightarrow 2^{61}$

- but dictionary attacks are much more effective for extracting passwords (later)

should be random

```
var salt = 'secret key'

function mk_hash(s) {
    return crypto.createHash('sha1').update(s).digest('hex')
}

function mk_cookie(c) {
    return c.toString() + '-' + mk_hash(c.toString() + salt)
}

function gt_cookie(s) {
    var splits = s.split("-", 2);
    var counter = parseInt(splits[0])
    var hash = splits[1]
    if (mk_hash(counter.toString() + salt) == hash) {
        return counter
    } else { return 0 }
}

app.get('/', function(req, res){ .... });
```

Unix Passwords

- passwords must **not** be stored in clear text
- instead `/etc/shadow` contains

```
name:$1$QIGCa$/ruJs8AvmrknzKTzM2TYE.:other_info
```

- `$` is the separator
- `1` is MD5 (actually SHA-512 is used nowadays, `6`)
- `QIGCa` is the salt
- `ruJs8AvmrknzKTzM2TYE.` $\rightarrow$ password + salt

```
(openssl passwd -1 -salt QIGCa pippo)
```

Plain-Text Passwords

Plain-Text Passwords

On 25 September 2012, a report on a data breach at IEEE:

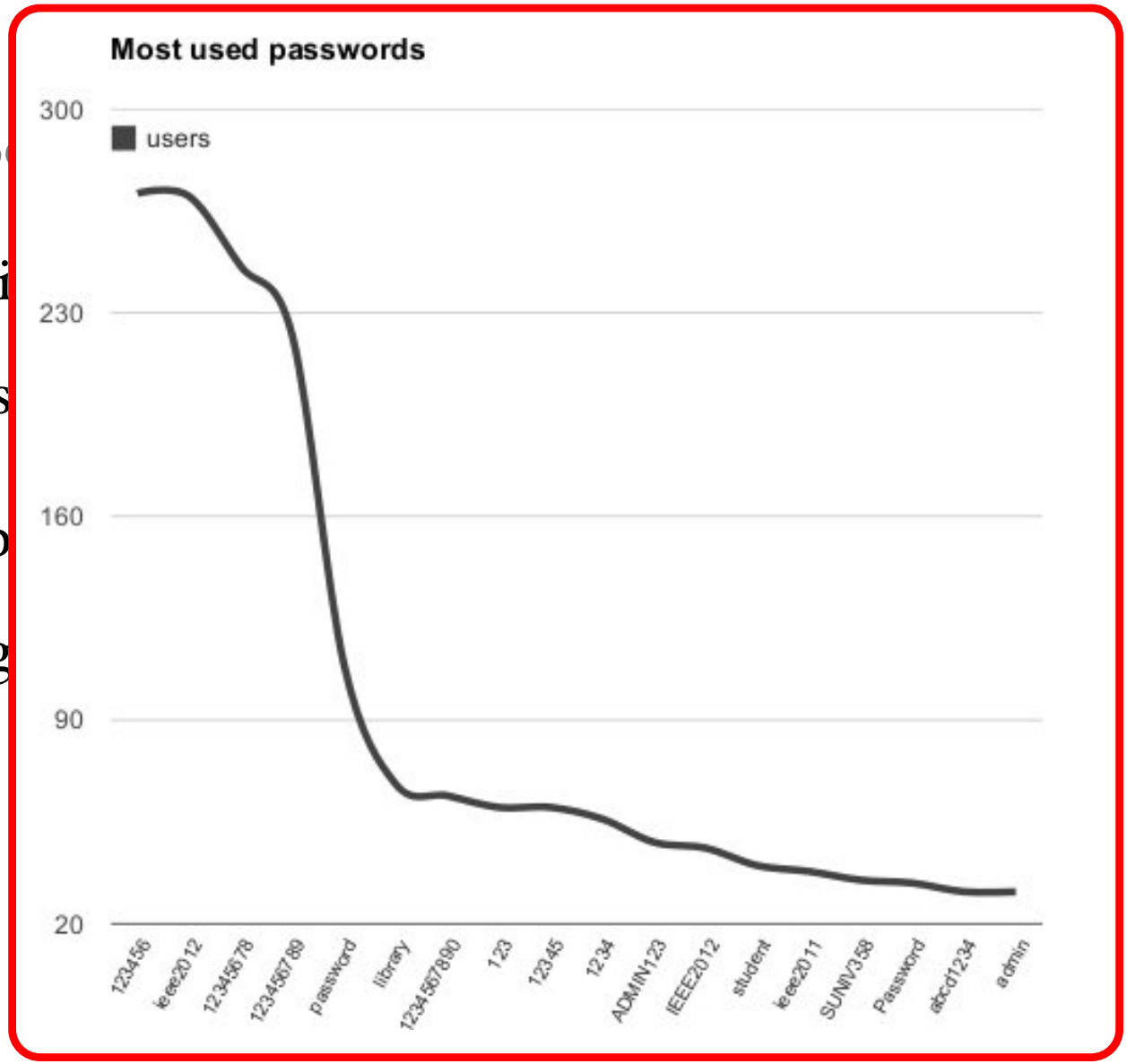
- IEEE is a standards organisation (not-for-profit)
- many standards in CS are by IEEE
- 100k plain-text passwords were recorded in logs
- the logs were openly accessible on their FTP server

`http://ieeelog.com`

Plain-Text Passwords

On 25 S… …EEE:

- IEEE i…
- many s…
- 100k p…
- the log… …log.com



Other Password Blunders

- in late 2009, when an SQL injection attack against online games service RockYou.com exposed 32 million plaintext passwords
- 1.3 million Gawker credentials exposed in December 2010 containing unsalted(?) MD5 hashes
- June 6th, 2012, 6 million unsalted SHA-1 passwords were leaked from linkedIn
- in July 2015, hackers leaked a password database from Ashley Madison containing 31 million passwords, many of them poorly hashed

(web user maintains 25 separate accounts but uses just 6.5 passwords.)

Brute Forcing Passwords

- How fast can hackers crack SHA-1 passwords?

Brute Forcing Passwords

- How fast can hackers crack SHA-1 passwords?
- The answer is 2 billion attempts per second using a Radeon HD 7970

password length	time
5 letters	5 secs
6 letters	500 secs
7 letters	13 hours
8 letters	57 days
9 letters	15 years

graphics card
ca. £300

5 letters $\approx 100^5 = 10$ billion combinations
(1 letter - upper case, lower case, digits, symbols ≈ 100)

Passwords

How to recover from a break in?

Passwords

How to recover from a break in?

- Do not send passwords in plain text.
- Security questions are tricky to get right.

This Course

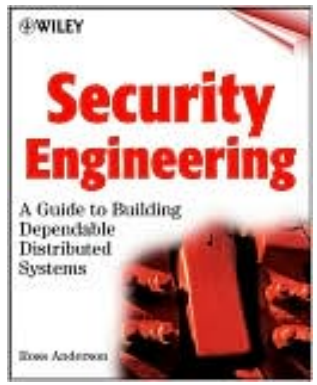
- electronic voting
- break-ins (buffer overflows)
- access control
  (role based, data security / data integrity)
- protocols
- zero-knowledge proofs
- privacy

  Scott McNealy:
  "You have zero privacy anyway. Get over it."

- trust, bitcoins
- static analysis

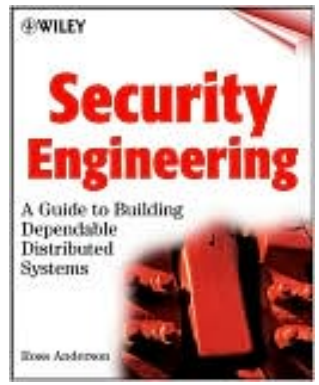
Books + Homework

- There is no single book I am following, but



Books + Homework

- There is no single book I am following, but



- The question "*Is this relevant for the exams?*" is not appreciated!

  Whatever is in the homework (and is not marked optional) is relevant for the exam. No code needs to be written.

Further Information

For your personal interest:

- RISKS mailing list
- Schneier's Crypto newsletter
- Google+ Ethical Hacker group

Take-Home Points

- Never store passwords in plain text.
- Always salt your hashes!
- Use an existing crypto algorithm; do not write your own!
- Make the party responsible for losses that is in the position to improve security.

User-Tracking Without Cookies

Can you track a user **without**:

- Cookies
- JavaScript
- LocalStorage/SessionStorage/GlobalStorage
- Flash, Java or other plugins
- Your IP address or user agent string
- Any methods employed by Panopticlick

→ `https://panopticlick.eff.org/`

Even when you disabled cookies entirely, have JavaScript turned off and use a VPN service.

User-Tracking Without Cookies

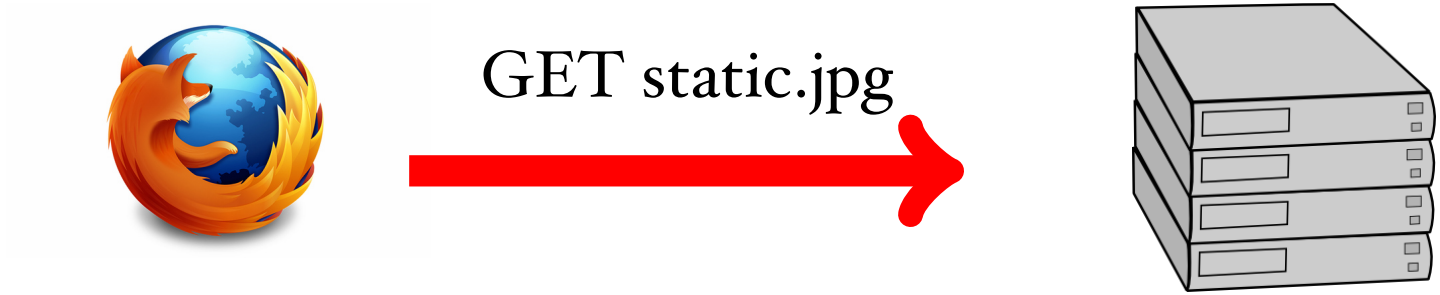
Can you track a user **without**:

- Cookies
- JavaScript
- LocalStorage/SessionStorage/GlobalStorage
- Flash, Java or other plugins
- Your IP address or user agent string
- Any methods employed by Panopticlick

→ https://panopticlick.eff.org/

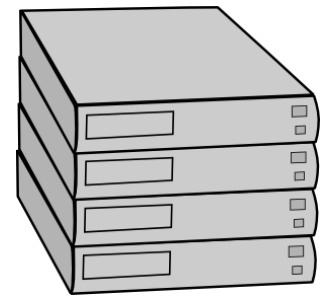
Even when you disabled cookies entirely, have JavaScript turned off and use a VPN service. (And numerous sites use it.)

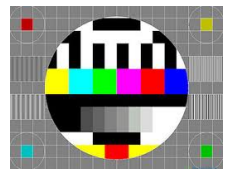
Web-Protocol

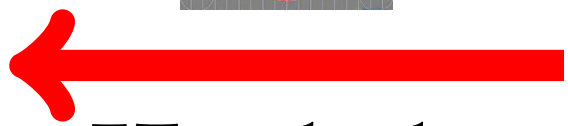


Web-Protocol

GET static.jpg



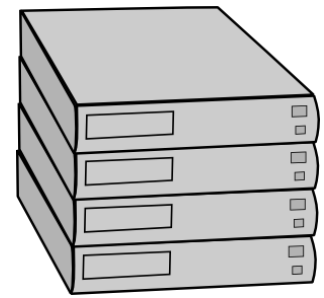


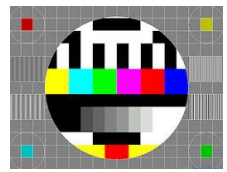


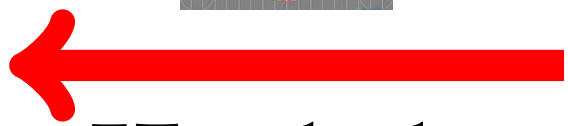
ETag: 7b33de1

Web-Protocol

GET static.jpg





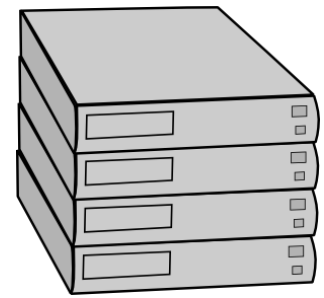


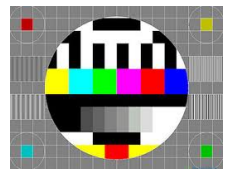
ETag: 7b33de1

GET static.jpg ETag: 7b33de1

Web-Protocol

GET static.jpg





ETag: 7b33de1

GET static.jpg ETag: 7b33de1

HTTP/1.1 304 (Not Modified)