

# SSL: salt, shot, lime

No ser mexicano (cuándo se está acercando a la seguridad del Web)



# Setting expectations

- Engineers are usually aware of what SSL is...

# Setting expectations

- Engineers are usually aware of what SSL is...
- ...and what SSL is for...

# Setting expectations

- Engineers are usually aware of what SSL is...
- ...and what SSL is for...
- ...but not so much about the implications

# Setting expectations

- Securing the whole site vs. securing a few pages

# Setting expectations

- Securing the whole site vs. securing a few pages
- Performance overhead

# Setting expectations

- Securing the whole site vs. securing a few pages
- Performance overhead
- What it means for HTTP clients: browsers, Flash and API clients



# Setting expectations

- Securing the whole site vs. securing a few pages
- Performance overhead
- What it means for HTTP clients: browsers, Flash and API clients
- How browsers usability sucks

# Setting expectations

- Securing the whole site vs. securing a few pages
- Performance overhead
- What it means for HTTP clients: browsers, Flash and API clients
- How browsers usability sucks
- What certificate should I buy?

# Setting expectations

- Securing the whole site vs. securing a few pages
- Performance overhead
- What it means for HTTP clients: browsers, Flash and API clients
- How browsers usability sucks
- What certificate should I buy?
- How it affects development environment

# Securing the whole site

- Is a quite drastic approach

# Securing the whole site

- Is a quite drastic approach
- Is important for apps that host sensitive data

# Securing the whole site

- Is a quite drastic approach
- Is important for apps that host sensitive data
- Gives people a warm fuzzy feeling of “real security”

# Securing the whole site

- Is a quite drastic approach
- Is important for apps that host sensitive data
- Gives people a warm fuzzy feeling of “real security”
- Has many myths associated with it

# Securing the whole site

rewrite ^/signin\$	<a href="https://myapp.local/signin"><u>https://myapp.local/signin</u></a>	permanent;
rewrite ^/signup\$	<a href="https://myapp.local/signup"><u>https://myapp.local/signup</u></a>	permanent;
rewrite ^/dashboard\$	<a href="https://myapp.local/dashboard"><u>https://myapp.local/dashboard</u></a>	permanent;
rewrite ^/people/(.*)/edit	<a href="https://myapp.local/people/\$1/edit"><u>https://myapp.local/people/\$1/edit</u></a>	permanent;
rewrite ^/people/(.*)	<a href="https://myapp.local/people/\$1"><u>https://myapp.local/people/\$1</u></a>	permanent;



# Securing the whole site

- “It is going to be sloooow...”

# Securing the whole site

- “It is going to be slooooow...”
- Well, how slow is “slow”?

# Securing the whole site

- “It is going to be sloooooow...”
- Well, how slow is “slow”?
- 60%?
- 70%?
- 200%?



# Performance overhead

- From my experience, ~ %5-20

# Performance overhead

- From my experience, ~ %5-20
- Keep number of HTTP connections low

# Performance overhead

- From my experience, ~ %5-20
- Keep number of HTTP connections low
- Rich clients hit most

# Performance overhead

- From my experience, ~ %5-20
- Keep number of HTTP connections low
- Rich clients hit most
- Go for 100 in YSlow



# Performance overhead

- From my experience, ~ %5-20
- Keep number of HTTP connections low
- Rich clients hit most
- Go for 100 in YSlow
- It's not that bad

# Performance overhead

“Past studies have shown that cryptographic controls are too costly for performance-critical and real-time systems. This study showed that **modern processors have recently become fast enough to allow full cryptographic controls** in systems that perform large network data transfers...”

— William Freedman, Ethan Miller

# Performance overhead

“Past studies have shown that cryptographic controls are too costly for performance-critical and real-time systems. This study showed that **modern processors have recently become fast enough to allow full cryptographic controls** in systems that perform large network data transfers...”

— William Freedman, Ethan Miller  
**in 1999**

# Bandwidth overhead

- 30-40%

# Bandwidth overhead

- 30-40%
- Is only important for mobile Web

# Bandwidth overhead

- 30-40%
- Is only important for mobile Web
- GMail is served via SSL on my iPhone anyway

# Bandwidth overhead

- 30-40%
- Is only important for mobile Web
- Mobile GMail is served via HTTPS on my iPhone anyway
- And I am pretty happy with that

# HTTP clients

- Browsers handle HTTPS just fine



# HTTP clients

- Browsers handle HTTPS just fine
- Flash does too, if you take care of cross-domain policies

# HTTP clients

- Browsers handle HTTPS just fine
- Flash does too, if you take care of cross-domain policies
- API clients must use libraries that handle HTTPS as transparently as possible

# HTTP clients

- Browsers handle HTTPS just fine
- Flash does too, if you take care of cross-domain policies
- API clients must use libraries that handle HTTPS as transparently as possible
- ...and not all of them do...

# HTTP clients

- API clients must use libraries that handle HTTPS as transparently as possible
- ...and not all of them do...
- So you keep supporting HTTP version anyway

# HTTP clients

- API clients must use libraries that handle HTTPS as transparently as possible
- ...and not all of them do...
- So you keep supporting HTTP version anyway
- Unless you are a big fat bank with lots of toxic assets

# HTTP clients

- All suck at handling errors

# HTTP clients

- All suck at handling errors
- HTTP libraries are overly optimistic

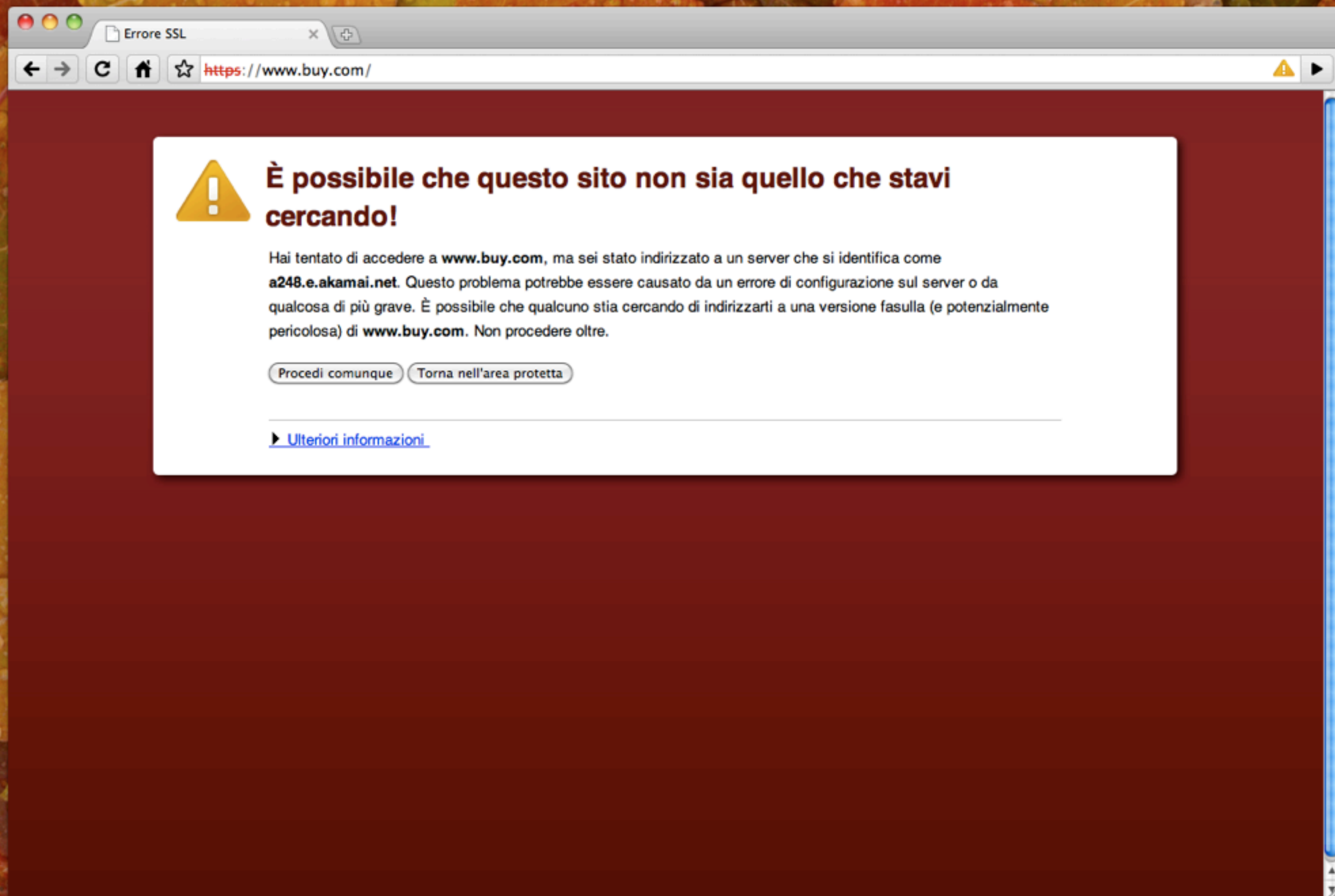
# HTTP clients

- All suck at handling errors
- HTTP libraries are overly optimistic
- Flash apps are overly optimistic



# HTTP clients

- All suck at handling errors
- HTTP libraries are overly optimistic
- Flash apps are overly optimistic
- Browsers greet you with a red screen of death





## Questa connessione non è affidabile

È stata richiesta a Firefox una connessione sicura con **shop.com**, ma non è possibile confermare la sicurezza del collegamento.

Normalmente, quando si cerca di attivare un collegamento in modalità sicura, il sito web fornisce un'identificazione affidabile per garantire all'utente che sta visitando il sito corretto. Tuttavia l'identità di questo sito non può essere verificata.

### Che cosa dovrei fare?

Se generalmente è possibile collegarsi a questo sito senza problemi, è possibile che questo errore sia causato dal tentativo da parte di qualcuno di sostituirsi al sito originale. Il consiglio è di non proseguire la navigazione.

Allontanarsi da questo sito

### ▼ Dettagli tecnici

shop.com utilizza un certificato di sicurezza non valido.

Il certificato è valido solo per [www.shop.com](http://www.shop.com).

(Codice di errore: ssl\_error\_bad\_cert\_domain)

### ► Sono consapevole dei rischi



### Safari non può verificare l'identità del sito web "shop.com".

Il certificato per questo sito web non è valido. Potresti connetterti ad un sito web che dichiara di essere "shop.com", ma che potrebbe mettere a rischio le tue informazioni riservate. Vuoi continuare comunque?

☐ Fidati sempre di "www.shop.com" quando ti connetti a "shop.com"

Class 3 Public Primary Certification Authority - G2

↳ VeriSign Class 3 Secure Server CA - G2

↳ www.shop.com



#### www.shop.com

Emesso da: VeriSign Class 3 Secure Server CA - G2

Scade: Tuesday, August 30, 2011 2:59:59 AM Ukraine (Kiev)

Questo certificato non è valido (mancata corrispondenza del nome host)

► Autorizza

► Dettagli



Nascondi certificato

Annulla

Continua

# HTTP clients

- Asset hosts add insult to injury

# HTTP clients

- Internet Explorer 7 & 8 both still do not support Keep-Alive

# HTTP clients

- Internet Explorer 7 & 8 both still do not support Keep-Alive
- Which is crucial for HTTPS in some ways

# Less trivial issues

- HTTPS traffic is not trivial to inspect



# Less trivial issues

- HTTPS traffic is not trivial to inspect
- Even locally

# Less trivial issues

- SSL certificates are hard to test “in a sandbox” before you deploy

# Less trivial issues

- SSL certificates are hard to test “in a sandbox” before you deploy
- Because certificates are tied to domains

# Less trivial issues

- Beware of chained certificates

# Less trivial issues

- Beware of chained certificates
- Nginx requires you to concat chained certificate to your own

# Less trivial issues

- Beware of chained certificates
- Nginx requires you to concat chained certificate to your own
- Apache uses a separate directive

# Less trivial issues

- Beware of chained certificates
- Nginx requires you to concat chained certificate to your own
- Apache uses a separate directive
- Safari on Mac OS X (but not on Windows) has somewhat broken list of root CAs

# What certificate to buy

- \$12.5?
- \$695?
- \$2890?
- \$1 gazillion?



# What certificate to buy

- It really depends

# What certificate to buy

- It really depends
- All most web apps care about is wildcard domains (subdomains coverage)

# What certificate to buy

- It really depends
- All most web apps care about is wildcard domains (subdomains coverage)
- GoDaddy has SSL certificates wildcard domains for \$200/year

# What certificate to buy

- It really depends
- All most web apps care about is wildcard domains (subdomains coverage)
- GoDaddy has SSL certificates wildcard domains for \$200/year
- 37signals, GitHub both seem to be happy with it

# Tools

- Certificate Patrol for Firefox

# Tools

- Certificate Patrol for Firefox
- ssldump

# Tools

- Certificate Patrol for Firefox
- ssldump
- CSR decoders

# Tools

- Certificate Patrol for Firefox
- ssldump
- CSR decoders
- OpenSSL



# Development

- Use self-signed certificates

# Development

- Use self-signed certificates
- Do not forget to add asset hosts to exceptions list in all browsers

# One more thing...

- I want to hire a couple of engineers

# One more thing...

- I want to hire a couple of engineers
- Who understand how TCP/IP, DNS and friends work, not just how to hook a plugin into a Rails app

# One more thing...

- I want to hire a couple of engineers
- Who understand how TCP/IP, DNS and friends work, not just how to hook a plugin into Rails a app
- Who reads source code instead of reading flame wars on forums

# One more thing...

- I want to hire a couple of engineers
- Who understand how TCP/IP, DNS and friends work, not just how to hook a plugin into Rails a app
- Who reads source code instead of reading flame wars on forums
- Who has real FOSS contributions

# One more thing...

- I want to hire a couple of engineers
- Who understand how TCP/IP, DNS and friends work, not just how to hook a plugin into Rails a app
- Who reads source code instead of reading flame wars on forums
- Who has real FOSS contributions
- Who has good writing skills

# One more thing...

- I want to hire a couple of engineers
- Who understand how TCP/IP, DNS and friends work, not just how to hook a plugin into Rails a app
- Who reads source code instead of reading flame wars on forums
- Who has real FOSS contributions
- Who has good writing skills
- And I don't care much about your resume



# Thank you