Adventures with SSL

hitting one wall at a time

Setting expectations

Almost every software engineer knows what SSL is...

Setting expectations

…and what SSL is for…

Setting expectations

...but not so much about what kind of headaches it brings

Setting expectations

I have ~110 slides — this is going to be fast!

Setting expectations

No boring cryptography: real world issues only

Questions like…

Should I secure the whole site or just a few pages?

Questions like…

How large is performance overhead?

Questions like…

Will Flash, Java applets and API clients work with HTTPS?

Questions like…

What about browsers support?

Questions like…

What certificate should I buy?
What's the right certificate price for my app?

Questions like…

Is it really impossible to host multiple SSL-enabled sites on a single IP address?

Questions like…

How do I inspect encrypted traffic during development?

Hitting one wall at a time

Lets break it down one by one

Securing the whole site

Is a bit of extreme approach

Securing the whole site

May be worth it for apps that work
with really sensitive data

Securing the whole site

Like PayPal. Or something works with intellectual property. And so on.

Securing the whole site

Gives people a warm fuzzy feeling of “real security” \m/

Securing the whole site

This is what we are talking about…

Securing the whole site

```
rewrite ^/signin$            https://myapp.local/signin              permanent;
rewrite ^/signup$            https://myapp.local/signup              permanent;
rewrite ^/dashboard$         https://myapp.local/dashboard           permanent;
rewrite ^/people/(.*)/edit   https://myapp.local/people/$1/edit      permanent;
rewrite ^/people/(.*)        https://myapp.local/people/$1           permanent;
```

Securing the whole site

“It is going to be sloooow...”

Securing the whole site

~~How soon is now?~~

How slow is “slow”?

Securing the whole site

- 60%?
- 70%?
- 200%?
- I am fre-e-e-a-a-king out! (c) South Park 708

Performance overhead

From my experience, ~ %5-30

Performance overhead

Rule of thumb is…

Performance overhead

…keep number of HTTPS connections low

Performance overhead

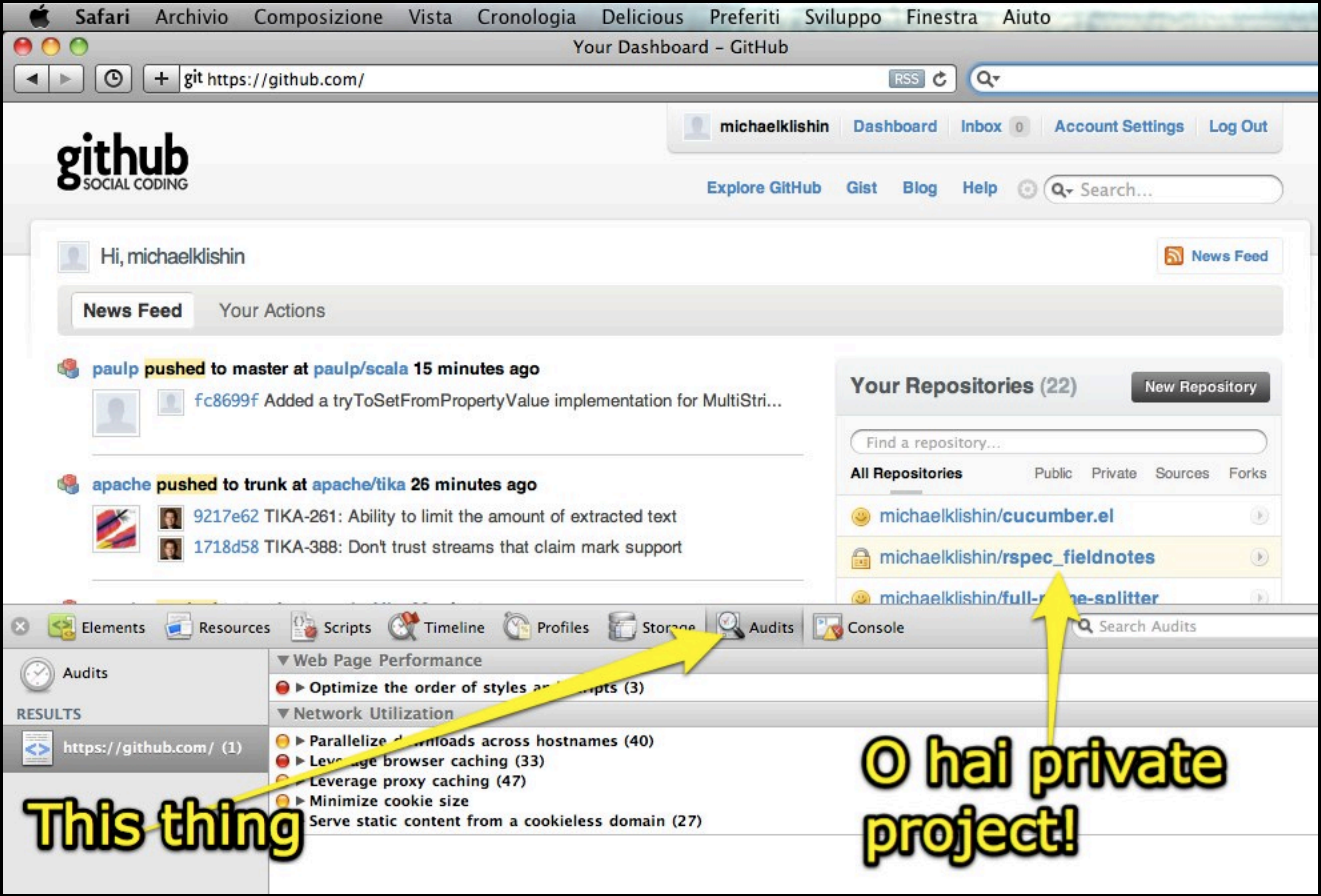
Rich clients (a la GMail) are hit the most

Performance overhead

Go for 99+ in YSlow

Performance overhead

WebKit Nightly and Chromium builds both have new Audits tab in Web Inspector


Safari Archivio Composizione Vista Cronologia Delicious Preferiti Sviluppo Finestra Aiuto
Your Dashboard – GitHub
https://github.com/
RSS
github SOCIAL CODING
michaelklishin Dashboard Inbox 0 Account Settings Log Out
Explore GitHub Gist Blog Help Search…
Hi, michaelklishin
News Feed
News Feed Your Actions
Your Repositories (22)
New Repository
Find a repository…
All Repositories Public Private Sources Forks
michaelklishin/cucumber.el
michaelklishin/rspec_fieldnotes
michaelklishin/full-r…e-splitter
Elements Resources Scripts Timeline Profiles Storage Audits Console
Search Audits
Audits
RESULTS
https://github.com/ (1)

paulp pushed to master at paulp/scala 15 minutes ago

fc8699f Added a tryToSetFromPropertyValue implementation for MultiStri...

apache pushed to trunk at apache/tika 26 minutes ago

9217e62 TIKA-261: Ability to limit the amount of extracted text

1718d58 TIKA-388: Don't trust streams that claim mark support

▼ Web Page Performance

- ● ▶ Optimize the order of styles an… …ripts (3)

▼ Network Utilization

- ● ▶ Parallelize d…wnloads across hostnames (40)
- ● ▶ Lev…age browser caching (33)
- ● ▶ Leverage proxy caching (47)
- ● ▶ Minimize cookie size
- Serve static content from a cookieless domain (27)

This thing

O hai private project!

Performance overhead

Is 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

in1999

Bandwidth overhead

30% to 40%

Bandwidth overhead

Only really matters for mobile web

Bandwidth overhead

GMail is served via HTTPS on my iPhone

Bandwidth overhead

And I am happy with that

HTTPS clients

Browsers handle HTTPS fine,
what about Flash?

HTTPS clients

Flash does too, if you take care of cross-domain policies and friends

HTTPS clients

API clients must use libraries that handle HTTPS as transparently as possible

HTTPS clients

…and not all of them do…

HTTPS clients

So you keep supporting non-HTTPS version too :(

HTTPS clients

Unless you are a big ass bank with lots of toxic assets and legalese bullcrap

HTTPS clients

mostly suck at handling SSL errors

HTTPS clients

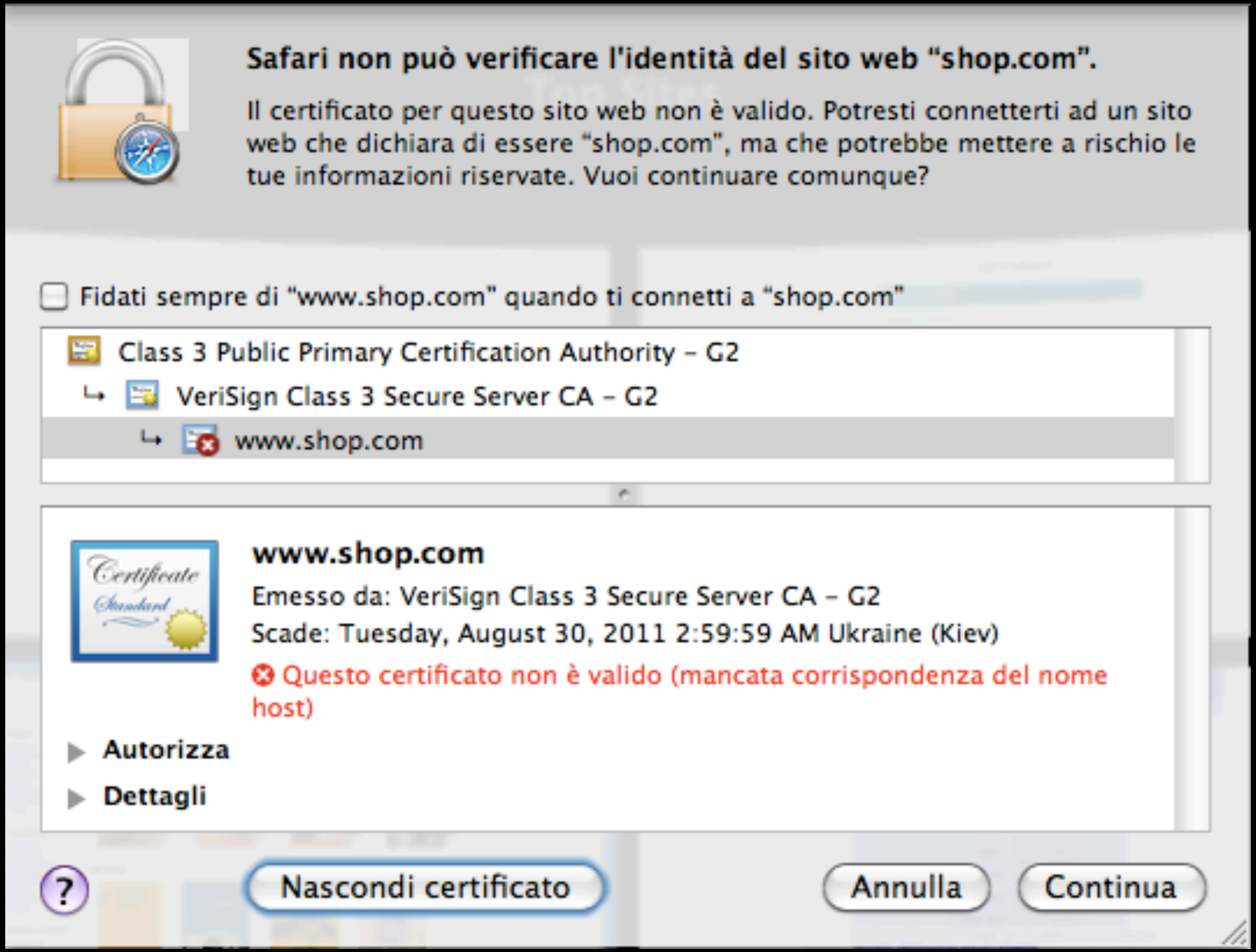
library authors are overly optimistic

HTTPS clients

can fuck you and your customers up

Red screen of death

Decision by the Firefox team
that does as much harm as it does good



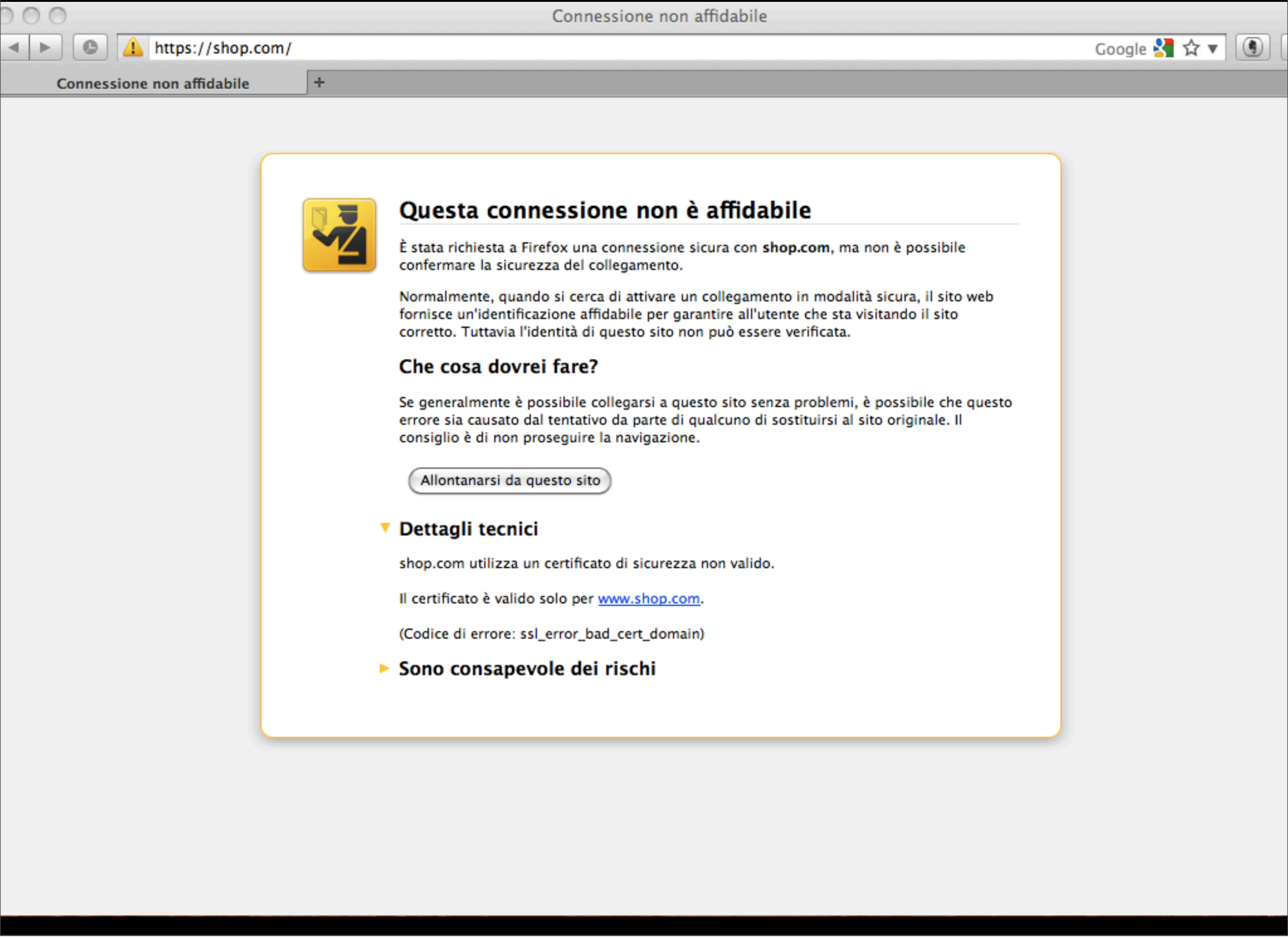
Safari non può verificare l'identità del sito web "shop.com".

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)



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.

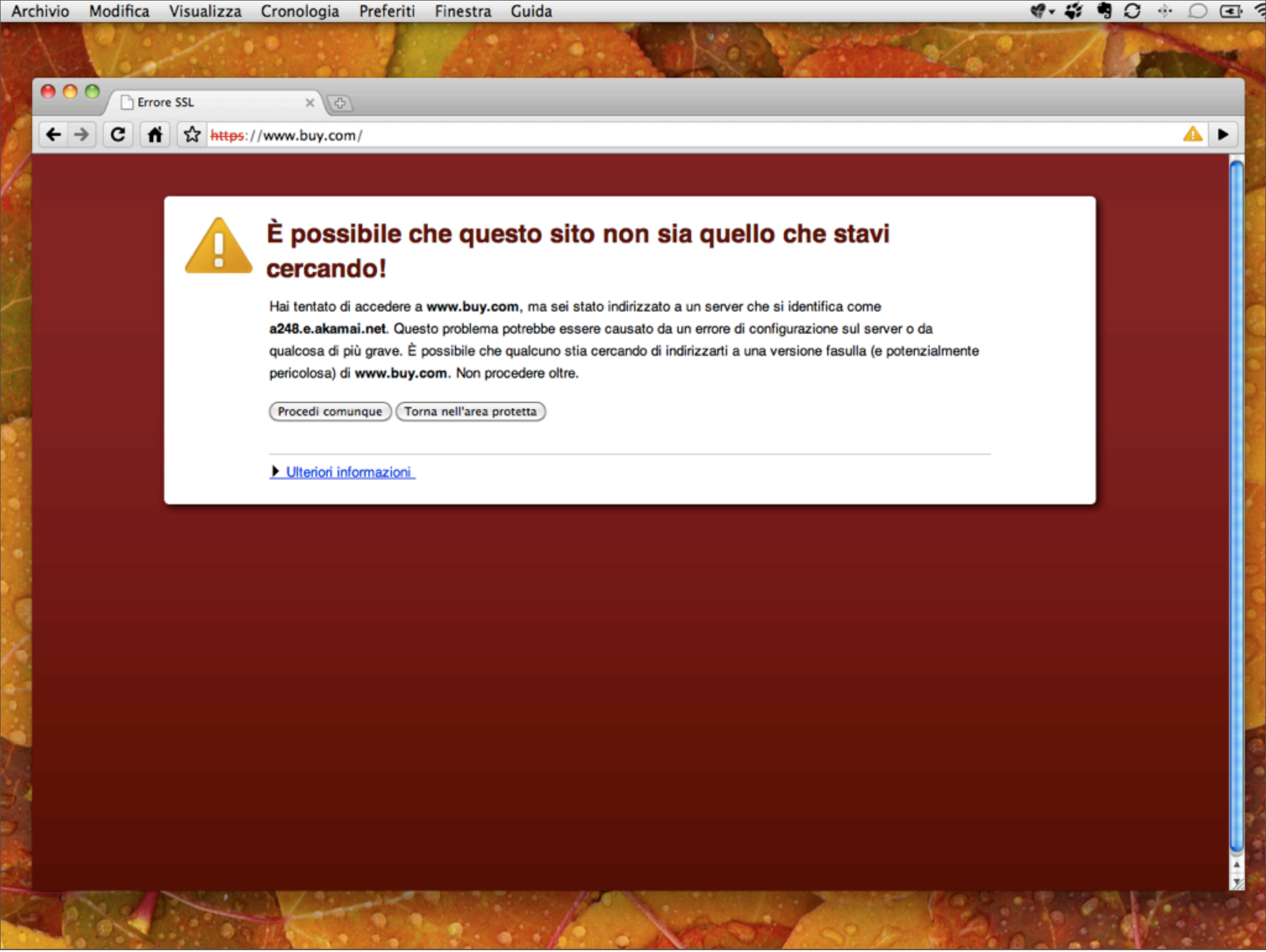
▼ Dettagli tecnici

shop.com utilizza un certificato di sicurezza non valido.

Il certificato è valido solo per www.shop.com.

(Codice di errore: ssl_error_bad_cert_domain)

▶ Sono consapevole dei rischi



È possibile che questo sito non sia quello che stavi cercando!

Hai tentato di accedere a **www.buy.com**, ma sei stato indirizzato a un server che si identifica come **a248.e.akamai.net**. Questo problema potrebbe essere causato da un errore di configurazione sul server o da qualcosa di più grave. È possibile che qualcuno stia cercando di indirizzarti a una versione fasulla (e potenzialmente pericolosa) di **www.buy.com**. Non procedere oltre.

Кинул пацана — по ебалу на!

Browsers

Asset hosts (assets*.myapp.com)
add insult to injury

Browsers

Browsers display SSL exception dialog when requesting a web page, but simply close network connection when requesting a CSS or JavaScript file.

Browsers

Browsers usability (when it comes to self-signed SSL certificate) is broken

Browsers

SSL is not just a mean of identification, it is also a mean of connection encryption

Browsers

Browsers completely ignore this part and act as drama queens when stumble upon a self-signed certificate

Browsers

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

Browsers

WebKit has some issues, too

```
+Changes with nginx 0.8.33                                  01 Feb 2010
+
+    *) Security: now nginx/Windows ignores trailing spaces in URI.
+       Thanks to Dan Crowley, Core Security Technologies.
+
+    *) Security: now nginx/Windows ignores short files names.
+       Thanks to Dan Crowley, Core Security Technologies.
+
+    *) Change: now keepalive connections after POST requests are not
+       disabled for MSIE 7.0+. Thanks to Adam Lounds.
+
+    *) Workaround: now keepalive connections are disabled for Safari.
+       Thanks to Joshua Sierles.
```

Browsers

Keep-alive connections are important to keep number of HTTPS connections low :(

Multiple SSL sites on one IP address

Multiple SSL sites on one IP address

Is a pain in the ass

Multiple SSL sites on one IP address

Symptoms: random SSL errors (red screens of death) in Firefox

Multiple SSL sites on one IP address

Host: ruby-lang.org

Multiple SSL sites on one IP address

SSL connection is established before HTTP headers come in

Multiple SSL sites on one IP address

So web server cannot figure out what virtual host to use

Multiple SSL sites on one IP address

IE, Safari, Chrome seem to handle this case better

Multiple SSL sites on one IP address

My source code investigation with Nginx, WebKit and Firefox is not done yet :(

Multiple SSL sites on one IP address

http://nginx.org/en/docs/http/configuring_https_servers.html

Multiple SSL sites on one IP address

What do we do then?

Multiple SSL sites on one IP address

Buy additional IP addresses

Multiple SSL sites on one IP address

$1 or $2 at Linode, Slicehost, Rackspace

Multiple SSL sites on one IP address

Amazon EC2 won’t let you use multiple IPs with the same instance!

Multiple SSL sites on one IP address

Use separate machine to do traffic forwarding

Traffic forwarding: iptables

Pro: bare metal performance

Traffic forwarding: iptables

HTTP client’s IP is less-than-trivial to preserve

Traffic forwarding: HAProxy

Pro: HTTP client’s IP is easy to preserve

Traffic forwarding:
HAProxy

Con: overhead compared to iptables

Traffic forwarding: Nginx

Move Nginx or Apache to a separate host outside of EC2 and make it serve static content from there, proxying dynamic requests to EC2 instance

Traffic forwarding: Nginx

Pro: HTTP client’s IP is easy to preserve

Traffic forwarding:
Nginx

Pro: SSD, geographic load-balancing \m/

Traffic forwarding:
Nginx

Con: deployment complexity goes up

Traffic forwarding: Nginx

Con: nginx-upload-module assumes backend has access to web server's FS

Traffic forwarding:
Nginx

This is what Capistrano’s roles are for

SNI: Server name indication

SNI: Server name indication

An extension to SSL/TLS

SNI: Server name indication

Is around since at least 2007

SNI: Server name indication

Supported by Apache 2.2, Nginx, lighttpd, etc

SNI: Server name indication

IE 7, Firefox 2, Safari 3.2, Google Chrome…

SNI: Server name indication

…but not on Windows XP

NEW YEAR
2008
SAME
OLD CRAP
Cellfish

SNI: Server name indication

Is thus not an option for everyone

SNI: Server name indication

Wish Windows XP customers to switch to anything (Mac OS X, Windows 7, Linux)

Other issues

HTTPS traffic is not trivial to inspect

Other issues

SSL certificates are hard to test "in a sandbox" before you deploy

Other issues

Be aware of chained certificates

Other issues

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

If your app uses subdomains, make sure you buy a wildcard certificate

What certificate to buy

*.myapp.com

What certificate to buy

GoDaddy has SSL certificates
wildcard domains for $200/year

Tools

OpenSSL

Tools

CSR tools

Tools

ssldump

Tools

Certificate Patrol for Firefox

Development

Use self-signed certificates

Development

Don’t forget to add exceptions for all hosts to all the browsers

Phew! We’ve made it!

I would love to hear about your SSL-related issues
at michael@novemberain.com

GitHubz!

github.com/michaelklishin,
including slides for this talk

Thank you