SSL, CAs and keeping your stuff safe

a presentation by armin ronacher for pygrunn 2014

http://lucumr.pocoo.org/ — @mitsuhiko

SSL, CAs and keeping your stuff safe

A CAPITALISTIC AND SYSTEM CONFORMANT TALK ABOUT ENCRYPTION

a presentation by a[illegible]

http://lucumr.pocoo.org/ — @mitsuhiko

Armin Ronacher

Independent Contractor for Splash Damage / Fireteam
Doing Online Infrastructure for Computer Games

~ Epilogue ~

… The Problem with Programmers

Programmers think everything
is a technical problem

~ Chapter 1 ~

Fraud

What is the worst that can happen?

XXXX-XXXX-XXXX-1234

What makes Credit Card Numbers “secure”?

There will always be criminals

But what damage can they do?

prevented

Bitcoin

Strong Encryption

256 bit private key

decentralized

✓

A Credit Card

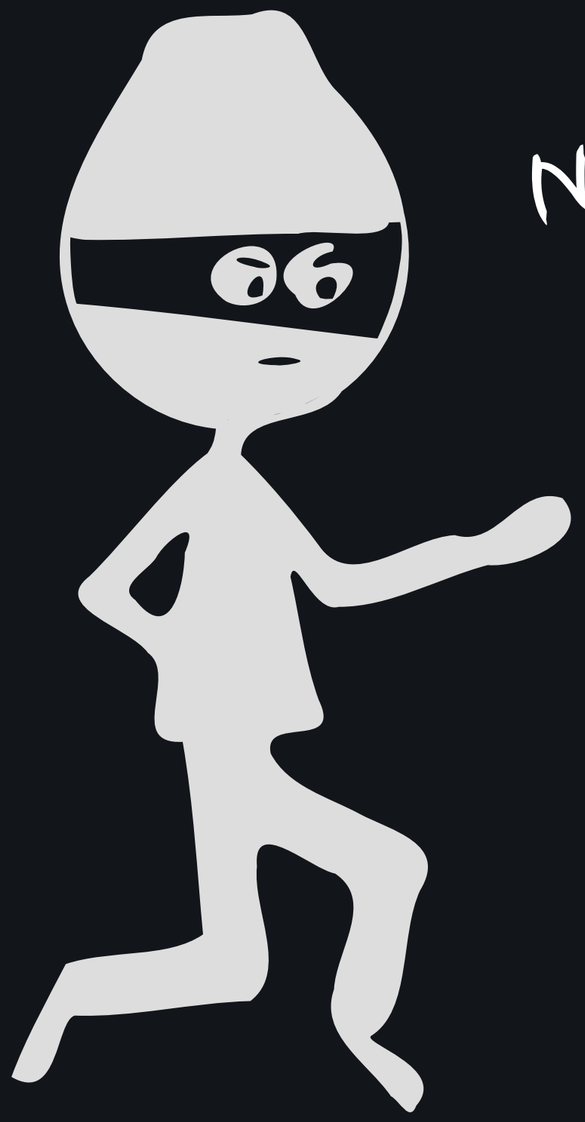
Potentially No Encryption

16 digit number + checksum

centralized

x

But I'd rather lose my credit card ...



Never mind me using this stolen card

over the counter

LOL NO SECURITY

but over the internet ...

We Accept Stolen Creditcards

The Protocol is insecure

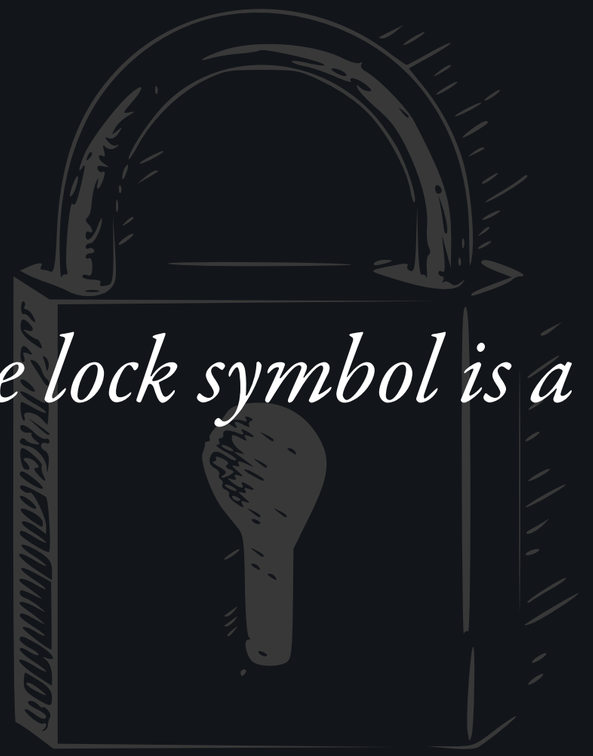
The Process is secure

If the fraud percentage is smaller than
the transaction fees we're all good.

It's too easy to forget the bigger picture

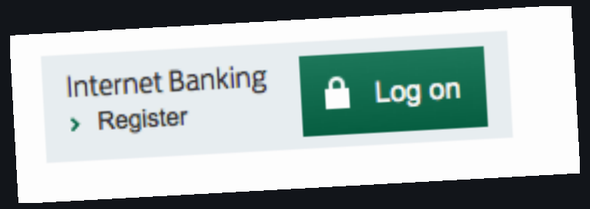
~ Chapter 2 ~

# of Lock Symbols and Encryption



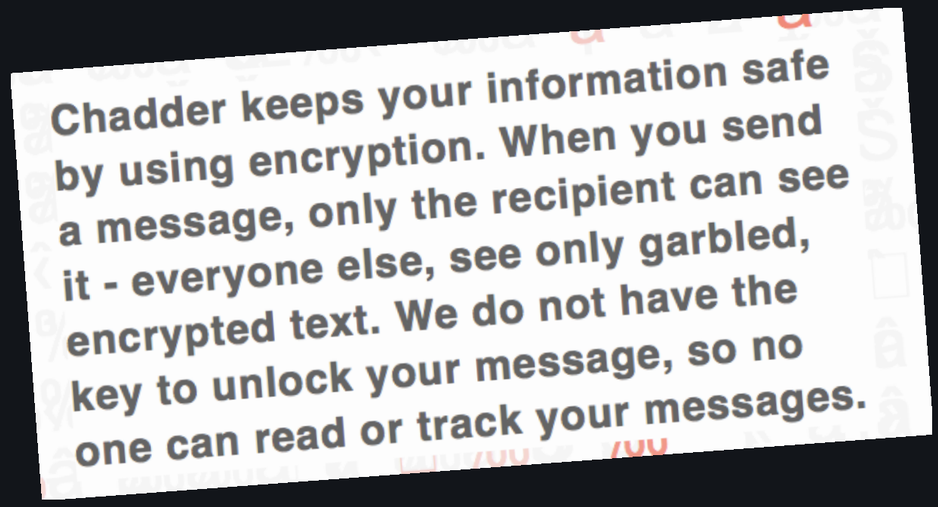
the lock symbol is a lie

the lock stands for secure



but so is encryption

such security



such buzzwords

CRIME

Heartbleed

BEAST

BREACH

PFS

users need to understand how to keep good from bad lock symbols / good from bad encryption.

but even developers are not sure yet ...

remember why you encrypt

(NSA does no care about your shitty blog)

~ Chapter 3 ~

Why do we Encrypt Traffic?


STARBUCKS
COFFEE
®
Careful, the beverage you're
about to enjoy is extremely hot.

public WiFi KILLED the unencrypted browser session

Who is the Attacker?

from secret agents to idiots

from targeted to untargeted

from low to high probability

~ Chapter 4 ~

What You Need for Encryption

passive vs active eavesdropping

```
$ ssh pocoo.org
The authenticity of host 'pocoo.org (148.251.50.164)' can't be established.
RSA key fingerprint is 14:23:83:02:45:f9:9c:d0:eb:39:c7:14:42:f5:9f:9c.
Are you sure you want to continue connecting (yes/no)?
```

your user does not check fingerprints

(your user thinks a lock symbol means security)

thus:

Certificate Authorities

let it be known that

CAs are worthless for securing APIs

~ Chapter 5 ~

Protecting APIs and Services

(non JavaScript APIs)

The Only Rule to Follow

run your own CA
issue certificates for 24 hours
trust your own CA only
screw revocations

You trust your own CA by distributing the certificate to everybody.

If your root gets compromised, distribute new root certificates.

If an individual key gets compromised,
in less than 24 hours everything is fine.

```
from requests import get
resp = get('https://api.yourserver.com/',
           verify='your/certificate.bundle')
```

“But my awesome AntiVirus says your certificate is not trusted.”

— *Windows User*

~ Chapter 6 ~

Certificate Authorities Again

Hardly news: CAs are Broken

But why are the broken?

Trusting a CA:

I Trust "TÜRKTRUST Elektronik Sertifika Hizmet Sağlayıcısı" to vouch for the identity of any domain on the planet.

trusting half the world: one shitty employee in one shitty CA is enough to break your security.

What we actually want:

I Trust "Comodo" to vouch for the identity of "Foo Owner" foo.com.

I only trust "Foo Owner" to vouch for the identity of api.foo.com

if you have seen *google.com* being from Verisign and all the sudden *google.com* becomes a StartSSL certificate you know something might be wrong.

Soon: Certificate Pinning?

~ Chapter 7 ~

Frack OpenSSL and Question “Best Practices”

Self-Signed Certificates are not bad. Just in browsers.

NEVER. EVER. LOOK AT OPENSSL'S SOURCE.

OpenSSL's "patches" are even worse:
Apple's OpenSSL always trusts system store :-/

Requests by default trusts it's own bundle :-/

(And does not even properly document how to use custom ones)

With Heartbleed SSL was less secure than no SSL :-/

~ Chapter 8 ~

Growing SSL

Credit Cards were made for thousands of people
Certificate Authorities were made for hundreds of sites

OpenSSL was probably improperly audited

See “OpenSSL Valhalla Rampage” :-(

“i give up. reuse problem is unfixable.
dlg says puppet crashes”

— tedu

~ Chapter 9 ~

Plan for Failure

what do you mean, certificate revocation does not work?

what happens to your user if he gets hacked?

(food for thought: keyloggers are still a thing)

what happens to your data

what happens to your company

encryption is hardened security
it must not be your only defense

Feel Free To Ask Questions

Talk slides will be online on lucumr.pocoo.org/talks
You can find me on Twitter: @mitsuhiko
And gittip: gittip.com/mitsuhiko
Or hire me: armin.ronacher@active-4.com