Ajax Security

Andrew van der Stock
vanderaj@owasp.org

OWASP AppSec Europe

May 2006

AJAX and Security

- Ajax

- Limited guidance
- New chapter in Guide

Compliance

Accessibility

- Accessibility is mandatory by law
  - Except for "justifiable hardship"
- Corporations and governments
  - No choice - do it!
- Personal web sites
  - No one will come after you... but...

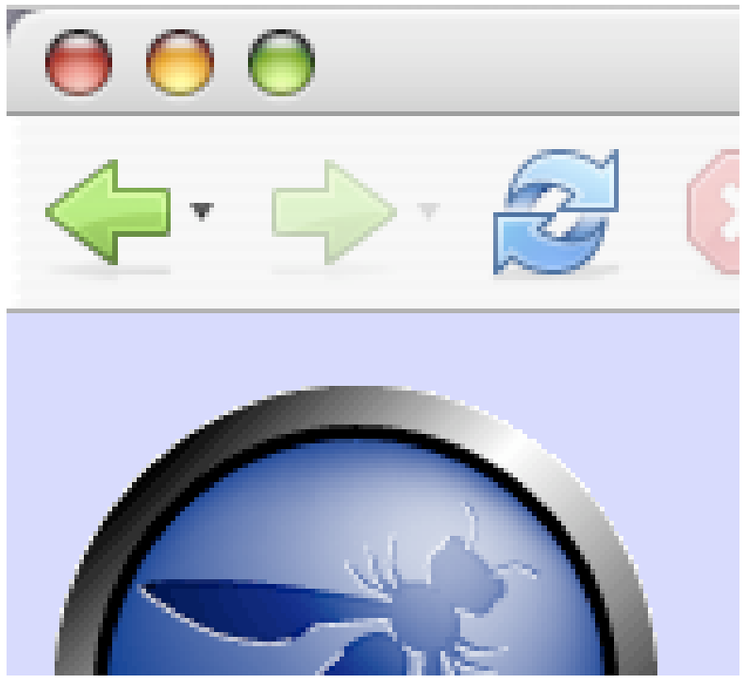
Accessibility

- Ask real users to test!

- Accessibility aides
- W3C WAI validator
- Basic tools

Back Button

- The most used button
- Ajax toolkits often destroy or hide it
- Support the Back Button!



Privacy

"You have no privacy.
Get over it."

Scott McNealy

Privacy

"Nothing that we have authorized conflicts with any law regarding privacy or any provision of the constitution."

John Ashcroft

Privacy

"Relying on the government to protect your privacy is like asking a peeping tom to install your window blinds."

John Perry Barlow

Privacy

- Ajax has client side state

- Local storage
- Caching
- Mash ups

Privacy ... not

- Javascript is clear text
  - often cached regardless of browser settings
  - Not private in any way

Privacy ... not

- DOM can be manipulated by hostile code
  - Not private in any way

Privacy ... not

- Dojo.Storage uses Flash
  - "Solution" for client-side persistent storage
  - Not private in any way
  - Often used for cross-domain postings... ARGH

Mash ups

- Who owns the data?
- Who gets the data?
- How are they going to handle it?

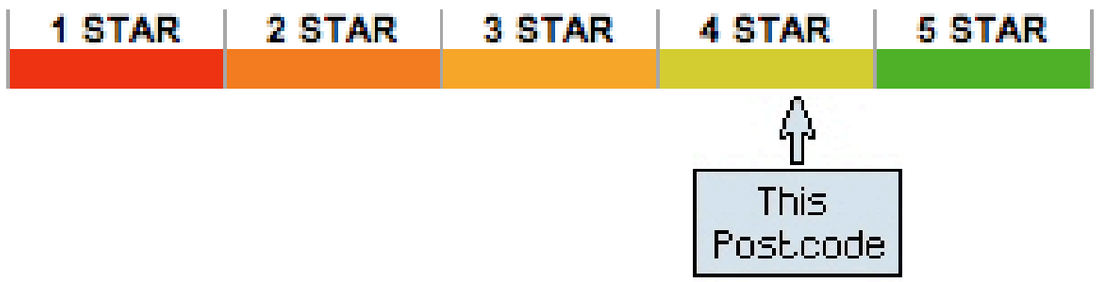
An example of a mash up

Financial Risk in L22 6AF

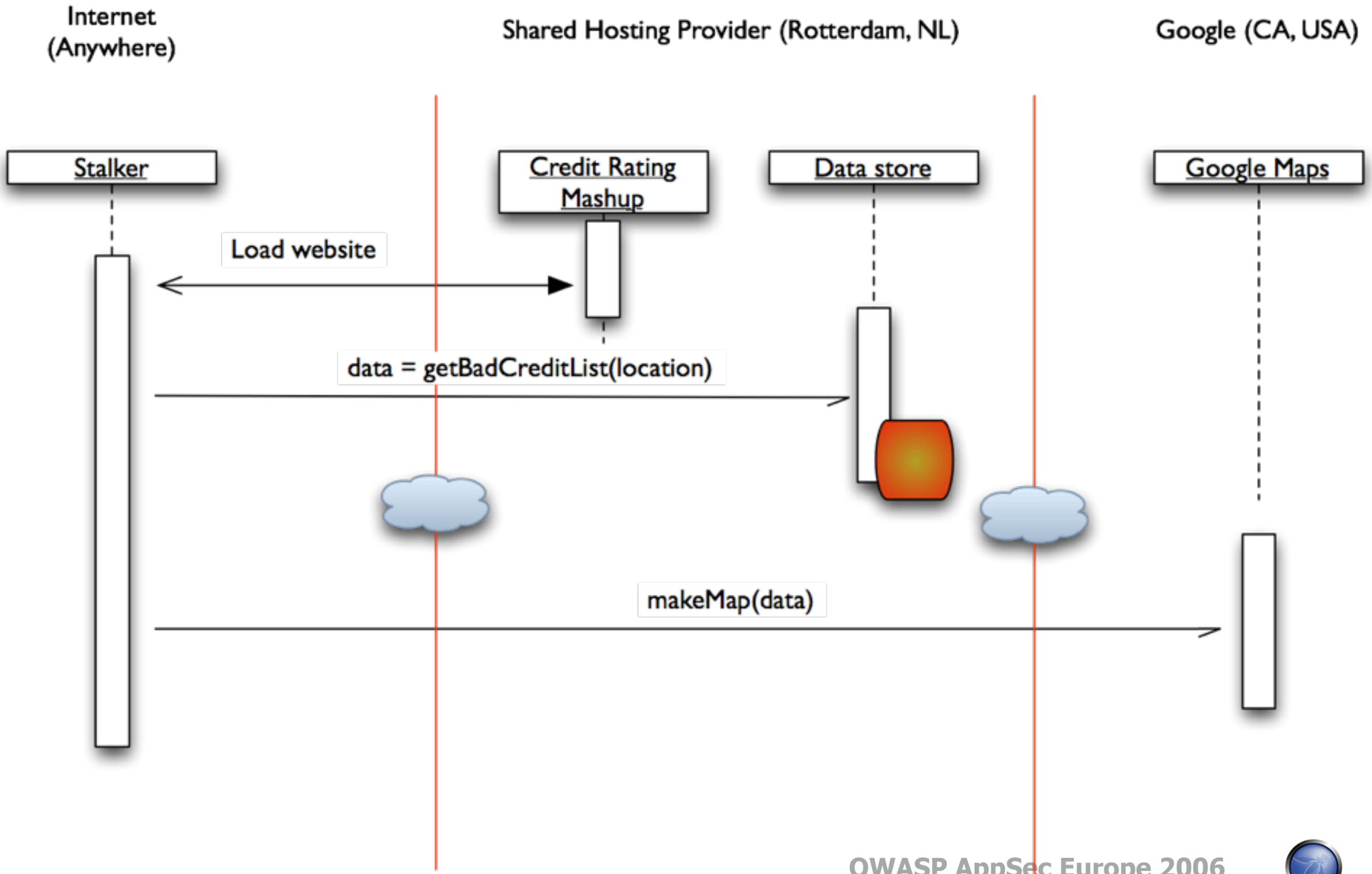
Based on Cameo financial risk, in **L22 6AF** the ratio of good credit accounts and bad credit accounts across the area is as follows. A good credit account is one with payments made on time. A bad credit account is, or has been, in arrears.

1 bad account in every 16 accounts

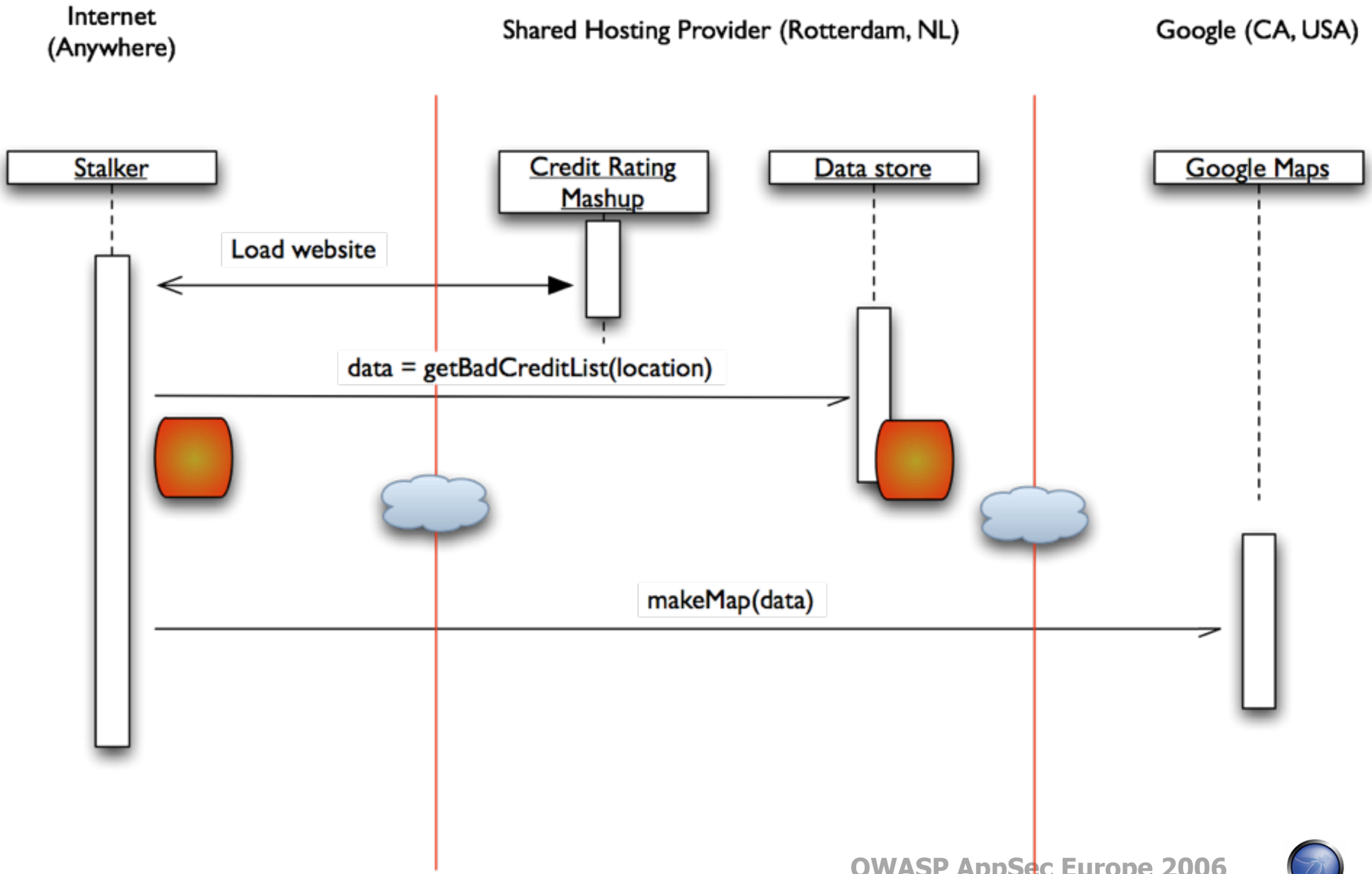
In **L22 6AF** the average checkmyfile.com Credit Score for all households, based on the Cameo financial risk is **872**



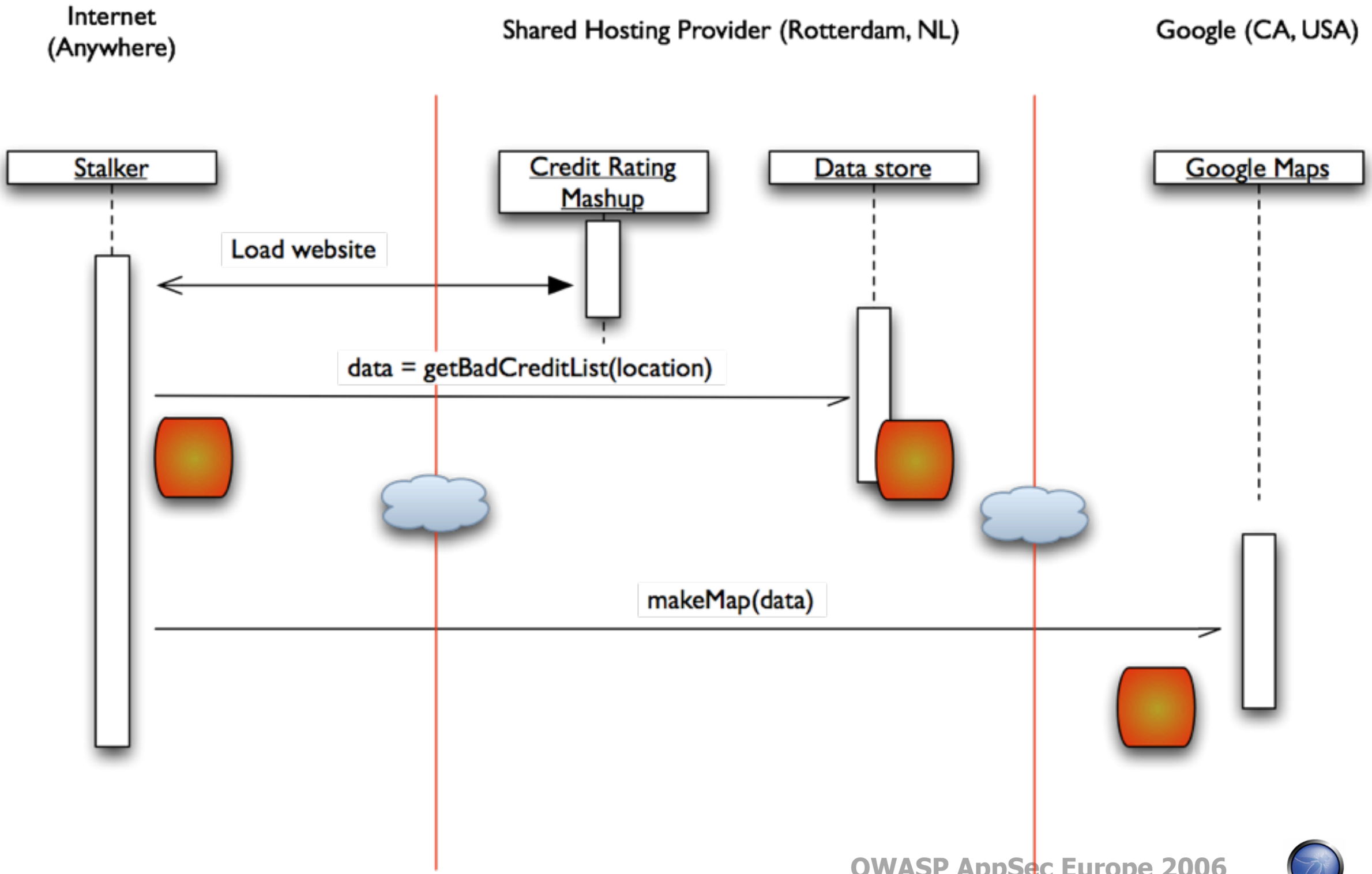
Credit Rating Mashup



Credit Rating Mashup



Credit Rating Mashup



Contentious issues

Voeg toe als favoriet! | Ook op uw site? | HELP | Slachtoffer? | Ontvang nieuwe misdaden per e-mail | Send to friend | Links | Nieuws! RSS

Misdaadkaart.nl

Al meer dan 35396 misdaden doorzoekbaar

Uw woonplaats | Uw buurt

doetinchem — Waar, bijv. Amersfoort

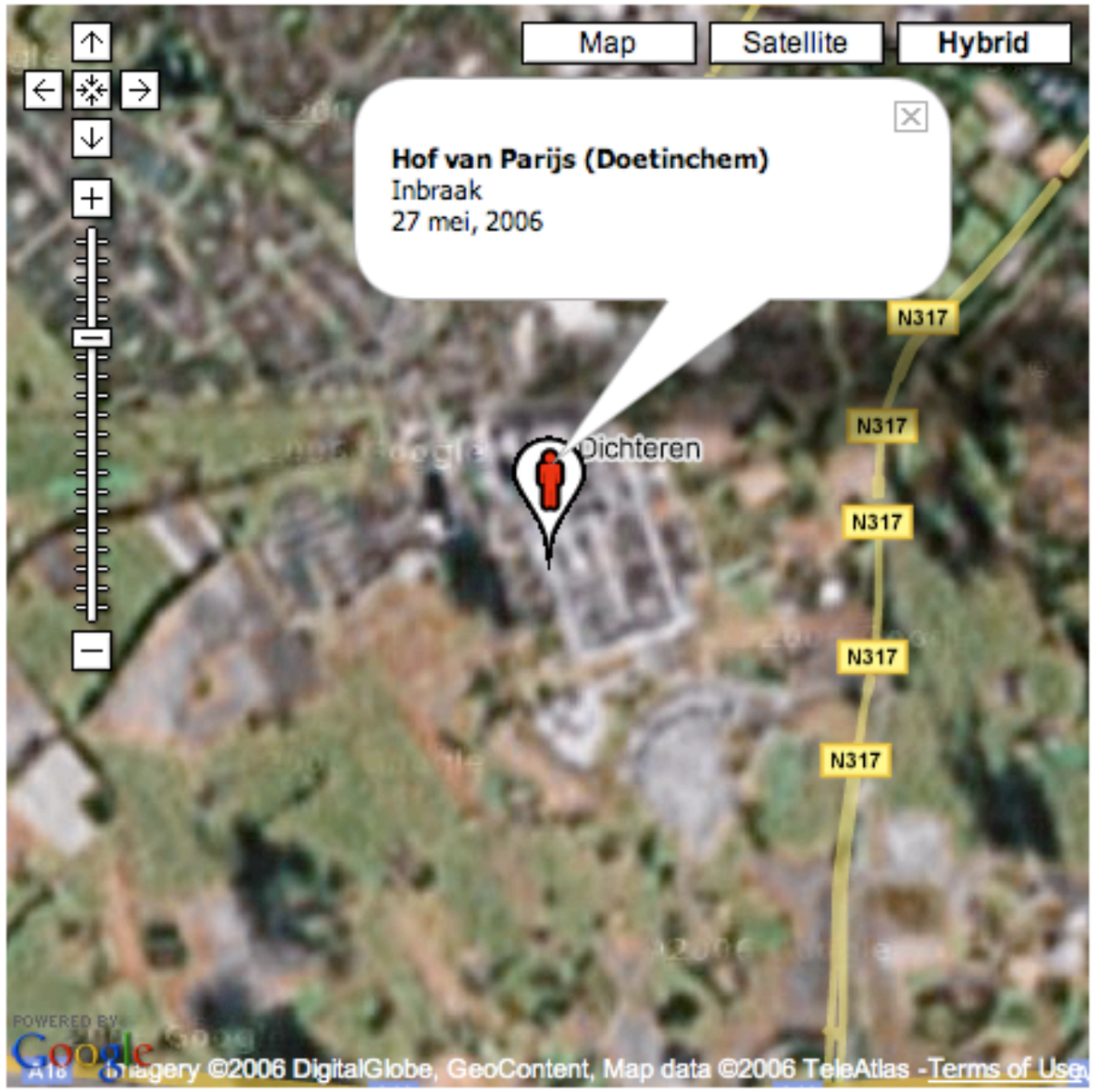
Inbraak — Wat, bijv. inbraak

zoeken

BRAND DIEFSTAL DRONKENSCHAP DRUGS INBRAAK MISHANDELING/VECHTPARTIJ MOORD/DOODSLAG ONGELUK OPLICHTING
OVERIG MISDRIJF/MELDING OVERLAST/VANDALISME OVERVAL/BEROVING SCHIETPARTIJ STEEKPARTIJ VERKEERSCONTROLE WAPEN ZEDENMISDRIJF

Wat is het hoogste gebouw in de wereld?

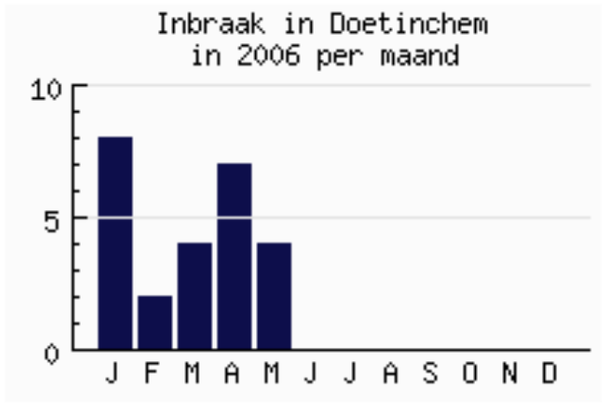
Google Trivia



Hof van Parijs (Doetinchem) - Terug
Inbraak
27 mei, 2006

Doetinchem - Verdacht gedrag op parkeerplaats

De politie heeft op 27/5 rond 01.00 uur na een melding een 20-jarige man uit Kaunas (Litouwen) aangehouden op het Hof van Parijs die zich verdacht ophield tussen de geparkeerde auto's. In de rugzak van de man bevond zich het nodige inbrekersgereedschap. De man is aangehouden en naar het politiebureau overgebracht voor verder verhoor.



Contentious issues

Welcome to Pervwatch.org

This site aims to make it easy for anyone to quickly find sex offenders in a graphical way. Many of the state registries fulfill the purpose of making the data freely accessible, however it can be difficult and time-consuming to determine just how many live near you, and how close they are. This site plots the sex offenders on a map so you can instantly see how many and where they are.

Site is closed

've decided to no longer continue updating this web site. The decision was tough, but was a result of a number of things. First and foremost I didn't have the time or resources to dedicate to the site to keep it going and the data fresh. Second, was the continuing number of threats (legal or otherwise) I was and still am receiving for keeping this site up. I don't make any money off this site and I don't have piles of cash lying around to fend off lawsuits. Sorry for those of you who found this service to be informative and useful.

Access Control

Authentication

- Don’t let any old caller in
- What’s okay without authentication?
- Authenticate new XMLHttpRequest sessions

Ask...

Request | Response

```
POST http://www.aussiefrogs.com/forum/ajax.php HTTP/1.1
Host: www.aussiefrogs.com
User-Agent: Mozilla/5.0 (Macintosh; U; PPC Mac OS X Mach-O; en-US
0.7 Paros/3.2.9
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=
.5
Accept-Language: en-us,en;q=0.5
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Proxy-Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 29
Pragma: no-cache
Cache-Control: no-cache
```

Look ma! No cookies!

```
do=usersearch&fragment=uga&s=
```


and ye shall receive

Yeah Baby!

Come to papa!

Request | Response

```
HTTP/1.1 200 OK
Date: Tue, 07 Feb 2006 05:12:40 GMT
Server: Apache/1.3.33 (Unix) mod_python/2.7.10 Python/2.2.2 mod_
_throttle/3.1.2 PHP/4.4.1 FrontPage/5.0.2.2635 mod_ssl/2.8.22 Op
X-Powered-By: PHP/4.4.1
Set-Cookie: bbsessionhash=b9369256eab3aacd64eff70550ba2525
Set-Cookie: bblastvisit=1139289160; expires=Wednesday, 07-Feb-
Set-Cookie: bblastactivity=0; expires=Wednesday, 07-Feb-07 05:12
Expires: 0
Cache-Control: private, post-check=0, pre-check=0, max-age=0
Pragma: no-cache
Content-Type: text/xml; charset=windows-1252
```

```
<?xml version="1.0" encoding="windows-1252"?>
<users>
        <user userid="2674">Uga Boga</user>
</users>
```


Authorization

Would you let Bart call your admin function?

Authorization

- Use same authorization methods
- Default deny; all actions should be denied unless allowed
- Error responses for no authorization

Sessions and State Management

Session Fixation

- Use toolkits which send session tokens
- Use proper session management to maintain the session
- OWASP Guide - Session Management chapter

Cross-domain XML Http Requests

- By security design, no browser supports this
- Many designs want to do this
  - or already do this (Google Maps, etc)
- How to do it safely?
  - Only with federated security

State management

- In the good olde days, state was on the server
- With Ajax, a lot more state is on the client
- Think "hidden fields" but so much worse

Sending state

- Validate all state before use
- Sending state to the client for display
  - DOM injections
  - HTML injections
- Only send changed state back

Exposing internal state

- Just because it’s faster doesn’t mean it’s wiser
- Keep sensitive state on the server, always
- Don’t obfuscate JavaScript - it’s hard enough now

Ajax Attack Prevention

Injection Attacks

- PHP toolkits: look for code injection attacks
- JSON injection: be careful how you decode!
- DOM injection - client side attacks now much easier
- XML injection - both client and server side
- Code injection - both client and server side

Data validation

- Data from XMLHttpRequest must be validated

- Perform validation **after** authorization checks
- Validate using **same** paths as existing code

- If you (de-)serialize, be aware of XML injection

Ajax APIs

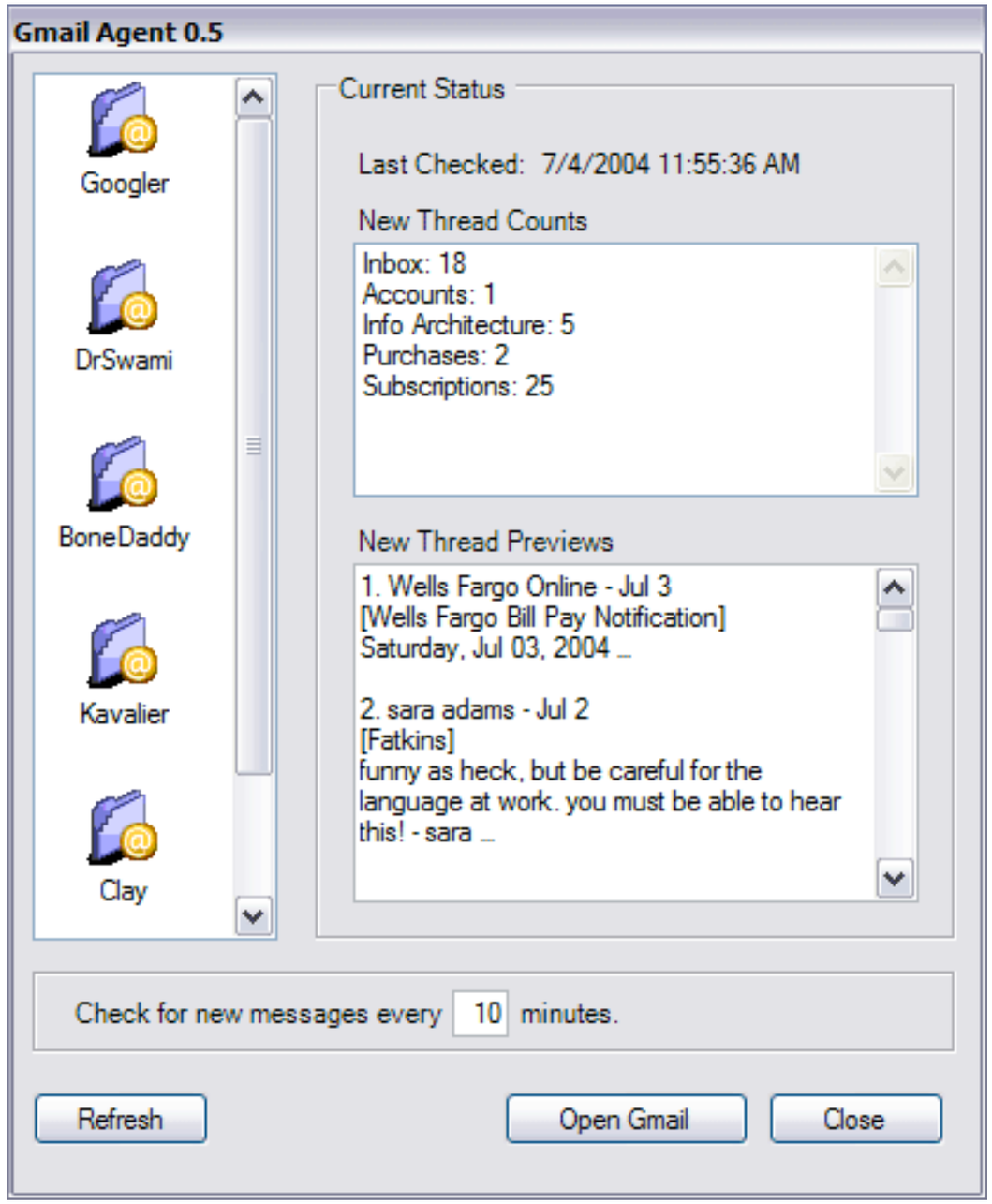
Reconstructing Ajax API

Many Ajax apps have been "decoded"

e.g. libgmail, GMail Agent API, gmail.py, etc

Spawned GMailFS, Win32 Gmail clients, etc

Do not assume your app is special - it will be decoded!



GMail Agent API in action

GET APIs

Sites

- Sites
  - http://www.jamesdam
    - GET:favicon.ico
    - ajax_login
      - GET:login.css
      - GET:login.html
      - GET:login.php(h
      - GET:login.php(t
      - GET:login_contr
      - GET:login_prese
      - GET:md5.js
      - GET:xml_http_r

Request | Response | Trap

```
HTTP/1.1 200 OK
Date: Tue, 07 Feb 2006 03:14:06 GMT
Server: Apache/1.3.33 (Unix) mod_ruby/1.2.4 Ruby/1.8.2(2004-12-25) mod_auth_passthrough/1.8 mod_log_bytes/1.2 mod_bwlimited/1.
4 PHP/4.3.10 FrontPage/5.0.2.2635 mod_ssl/2.8.22 OpenSSL/0.9.7a
X-Powered-By: PHP/4.3.10
Content-Type: text/html
```

```
false|Invalid username and password combination.
```

Raw View

```
1 GET http://www.jamesdam.com/ajax_login/login.html HTTP/1.1  => HTTP/1.1 200 OK  [32.642 s]
4 GET http://www.jamesdam.com/ajax_login/login.html HTTP/1.1  => HTTP/1.1 200 OK  [0.909 s]
5 GET http://www.jamesdam.com/ajax_login/md5.js HTTP/1.1  => HTTP/1.1 200 OK  [0.741 s]
6 GET http://www.jamesdam.com/ajax_login/xml_http_request.js HTTP/1.1  => HTTP/1.1 200 OK  [0.281 s]
7 GET http://www.jamesdam.com/ajax_login/login_controller.js HTTP/1.1  => HTTP/1.1 200 OK  [0.292 s]
8 GET http://www.jamesdam.com/ajax_login/login_presentation.js HTTP/1.1  => HTTP/1.1 200 OK  [0.289 s]
9 GET http://www.jamesdam.com/ajax_login/login.css HTTP/1.1  => HTTP/1.1 200 OK  [0.283 s]
10 GET http://www.jamesdam.com/favicon.ico HTTP/1.1  => HTTP/1.1 404 Not Found  [0.385 s]
11 GET http://www.jamesdam.com/ajax_login/login.php?task=getseed HTTP/1.1  => HTTP/1.1 200 OK  [0.509 s]
12 GET http://www.jamesdam.com/ajax_login/login.php?task=checklogin&username=user1&id=60908&hash=a666e616225c6b3b210dc81aacc4ccef HTTP/1.1  => HTTP/1
```

History | Spider | Alerts | Output

Pseudo API Injection

- Almost all Ajax toolkits use GET by default
  - Force them to use POST

- Most PHP AJAX tool kits allow remote code injection by allowing client-side server code invocation

- eg: AJason, JPSpan and CPAINT (1.x)

Psuedo API

- Guess what I can do?
- Create proxy façades

Event Management

Error Handling

- Error handling is often neglected
- Do not use Javascript alert()

Parentless window syndrome

Auditing

- Client-side auditing is a joke
- Auditing must be:
  - comprehensive
  - unavoidable
  - tamper resistant

Questions

Andrew van der Stock
vanderaj@owasp.org

Images:

John Perry Barlow image used with permission

Stock*Exchange

Image After

Andrew's OWASP EU talks sponsored by nab

OWASP AppSec Europe

May 2006

The OWASP Foundation
http://www.owasp.org/