

Advanced Web Security

Philipp Oesch

01.03.2012

Compass Security AG
Glärnischstrasse 7
Postfach 1628
CH-8640 Rapperswil

Tel +41 55-214 41 60
Fax +41 55-214 41 61
team@csnc.ch
www.csnc.ch

About Philipp Oesch



*Security
&
Software
Development*



Security Analyst - Head of Development - Compass Security AG
before 2008 ...: Software Engineer - Projectleader - ELCA Informatik AG

Agenda



- New Challenges Today
- Risks with Open Source & Standard Frameworks
- **Live Hacking Demo – Struts 2**
- Recommendations
- **Live Hacking Demo – (XXE & MySQL UDF)**
- New HTTP Headers

Evolution Of Web Technologies

HTML

XML

CSS

Ruby on Rails

Java Script
Flash

COMPASS
SECURITY

AJAX

JSON

Applet

ASP.NET

GWT

JSP

Spring

Struts CSS

Struts2

JSF

Python

Silverlight

Perl

PHP

Challenges In The Past



Situation

- ❖ Selfmade applications

Challenges

- ❖ Understanding of basic web technologies
- ❖ Knowledge about web security & secure coding



Challenges Today



Situation

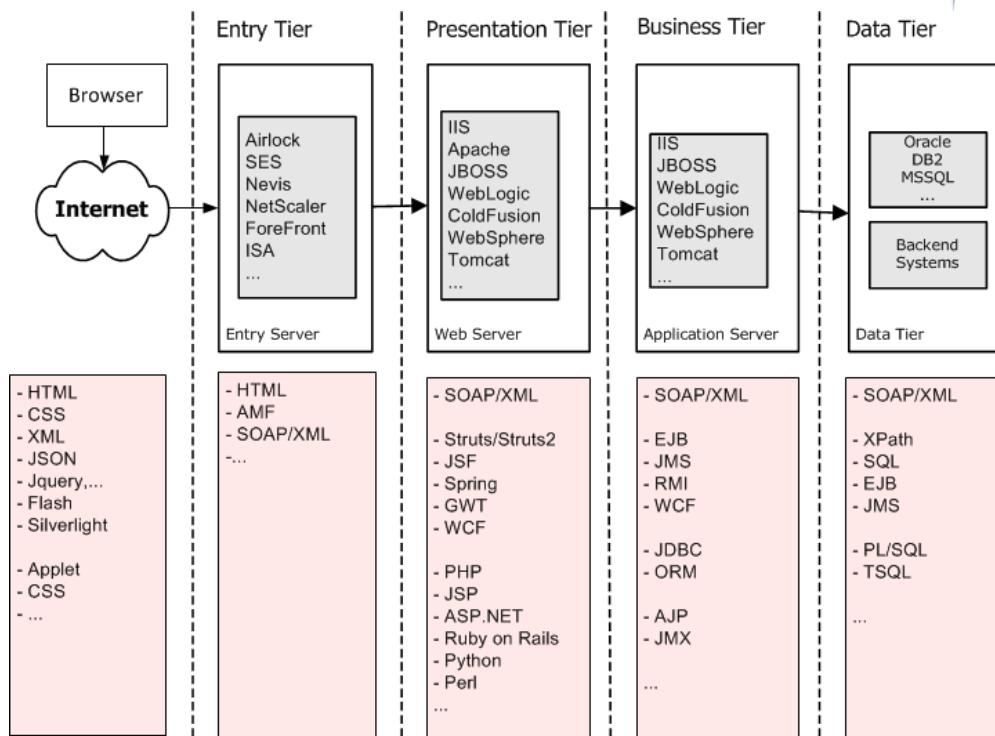
- ❖ Different technologies & libraries
- ❖ Complex frameworks

Challenges

- ❖ Knowledge of technologies
- ❖ Knowledge of frameworks
- ❖ Understanding of underlying technologies
- ❖ Knowledge about web security & secure coding



Technology Stack Today

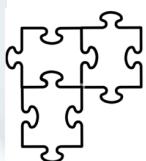


© Compass Security AG

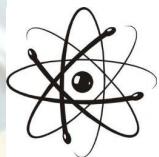
www.csnc.ch

Slide 7

Risks Today



- Growing application landscape
 - ◆ Different frameworks
 - ◆ Different technologies



- Growing framework complexity
 - ◆ Function overview
 - ◆ Understanding of underlying technology



- Quick and dirty application migrations
 - ◆ Integration of new features & technologies
 - ◆ Integration of backend & external systems

© Compass Security AG

www.csnc.ch

Slide 8

Development

- ◆ Programming of applications
- ◆ Bugfixing of applications

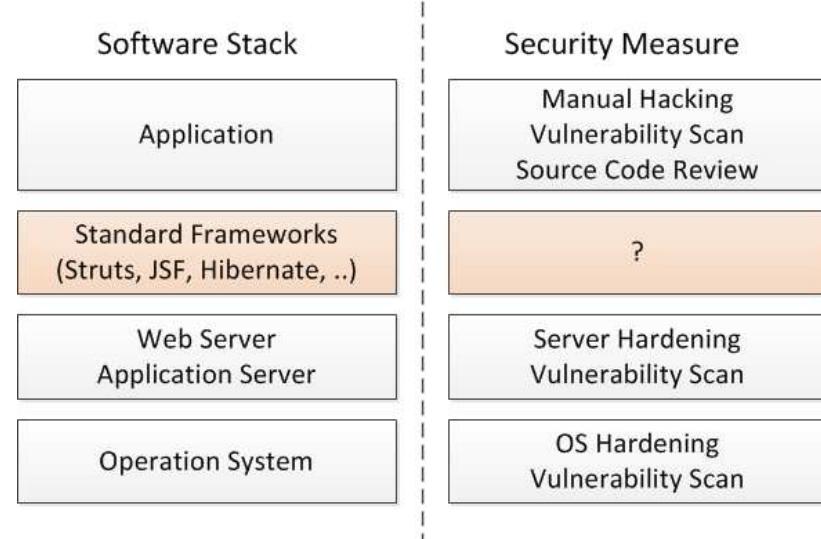
Operations

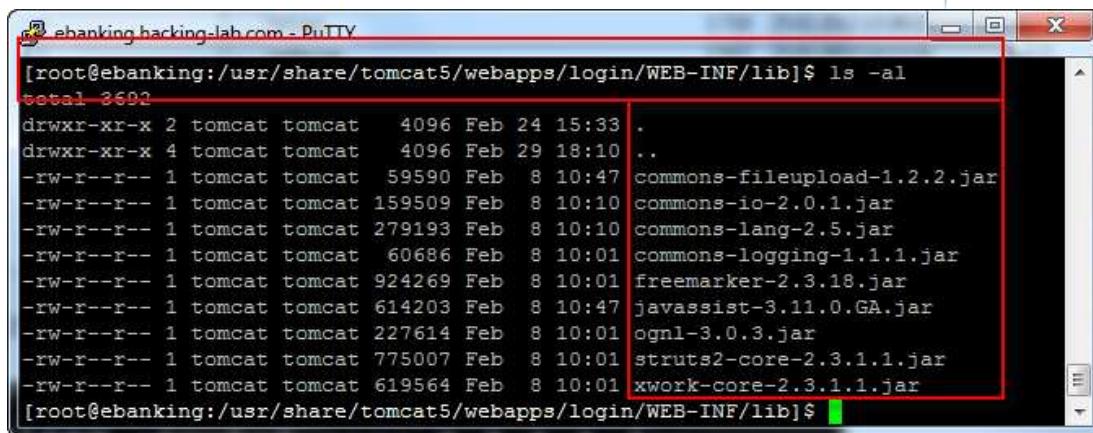
- ◆ Deployment in the productive environment
- ◆ Backup & Logging
- ◆ System updates
 - ◆ Operation system
 - ◆ Installed software
- ◆ System hardening

But Who is responsible for?

- ◆ Secure configuration of used frameworks
- ◆ Updating/Patching from used framework & libraries
- ◆ Overall security

Security Measure





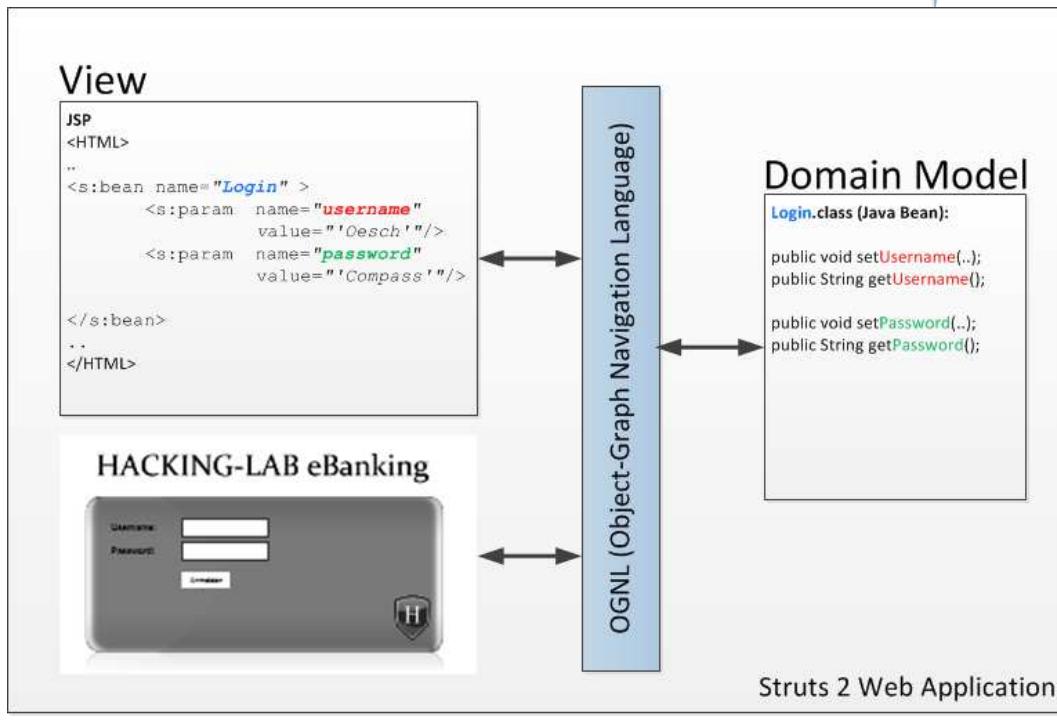
```
[root@ebanking:/usr/share/tomcat5/webapps/login/WEB-INF/lib]$ ls -al
total 3692
drwxr-xr-x 2 tomcat tomcat 4096 Feb 24 15:33 .
drwxr-xr-x 4 tomcat tomcat 4096 Feb 29 18:10 ..
-rw-r--r-- 1 tomcat tomcat 59590 Feb 8 10:47 commons-fileupload-1.2.2.jar
-rw-r--r-- 1 tomcat tomcat 159509 Feb 8 10:10 commons-io-2.0.1.jar
-rw-r--r-- 1 tomcat tomcat 279193 Feb 8 10:10 commons-lang-2.5.jar
-rw-r--r-- 1 tomcat tomcat 60686 Feb 8 10:01 commons-logging-1.1.1.jar
-rw-r--r-- 1 tomcat tomcat 924269 Feb 8 10:01 freemarker-2.3.18.jar
-rw-r--r-- 1 tomcat tomcat 614203 Feb 8 10:47 javassist-3.11.0.GA.jar
-rw-r--r-- 1 tomcat tomcat 227614 Feb 8 10:01 ognl-3.0.3.jar
-rw-r--r-- 1 tomcat tomcat 775007 Feb 8 10:01 struts2-core-2.3.1.1.jar
-rw-r--r-- 1 tomcat tomcat 619564 Feb 8 10:01 xwork-core-2.3.1.1.jar
[root@ebanking:/usr/share/tomcat5/webapps/login/WEB-INF/lib]$
```

Framework libraries are often forgotten in the update Process!

Struts 2 Framework Vulnerability

Struts²

OGNL - Vulnerability
Object-Graph Navigation Language



Struts 2 Framework - OGNL INTRO

OGNL is used to access java objects and session values:

Set java object value:

```
<s:bean name="Login" var="loginbean">
    <s:param name="username" value="'Oesch'"/>
    <s:param name="password" value="'Compass'"/>
</s:bean>
```

Read java object values:

```
Username= <s:property value="#loginbean.username" />
Password= <s:property value="#loginbean.password" />
```

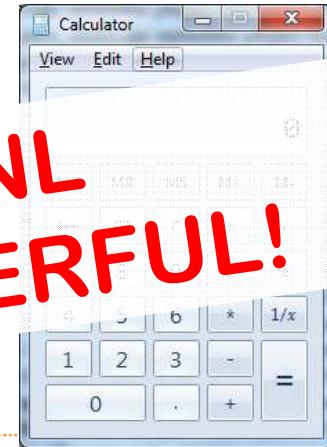
Read session values:

```
User= <s:property value="#session.user" />
```

But it is also possible to execute Java Code!

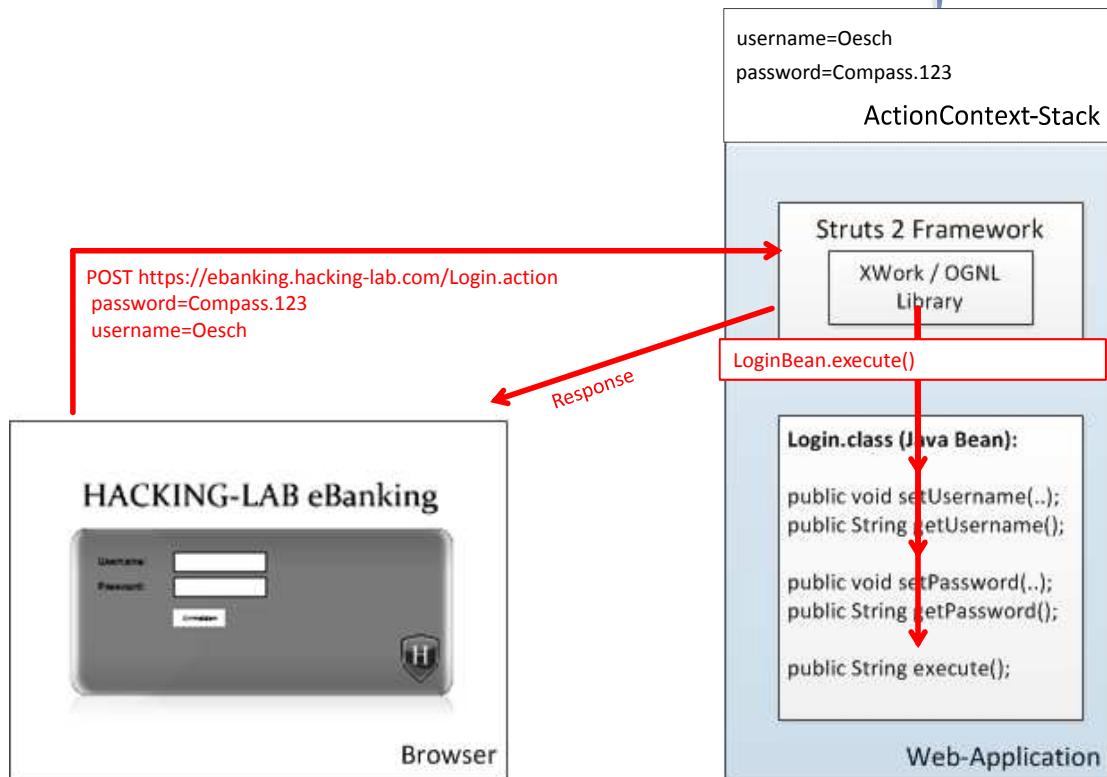
```
<s:property value="( 
    #context[\"xwork.MethodAccessor.denyMethodExecution\"]= new 
        java.lang.Boolean(false), 
    #_memberAccess[\"allowStaticMethodAccess\"]= new java.lang.Boolean(true), 
    @java.lang.Runtime@getRuntime().exec('calc.exe') 
)"/>
```

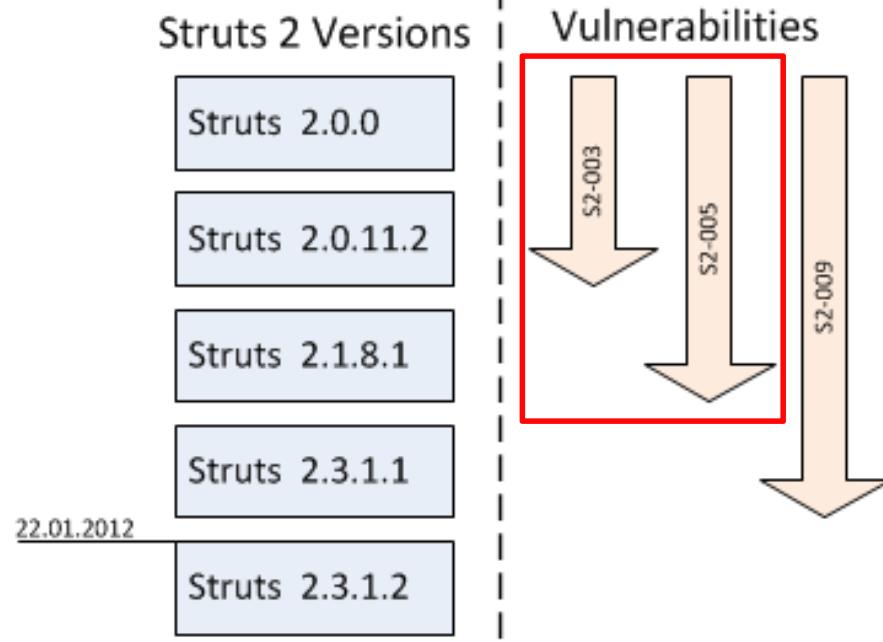
**OGNL
IS POWERFUL!**



Struts 2 - Live Hacking Demo

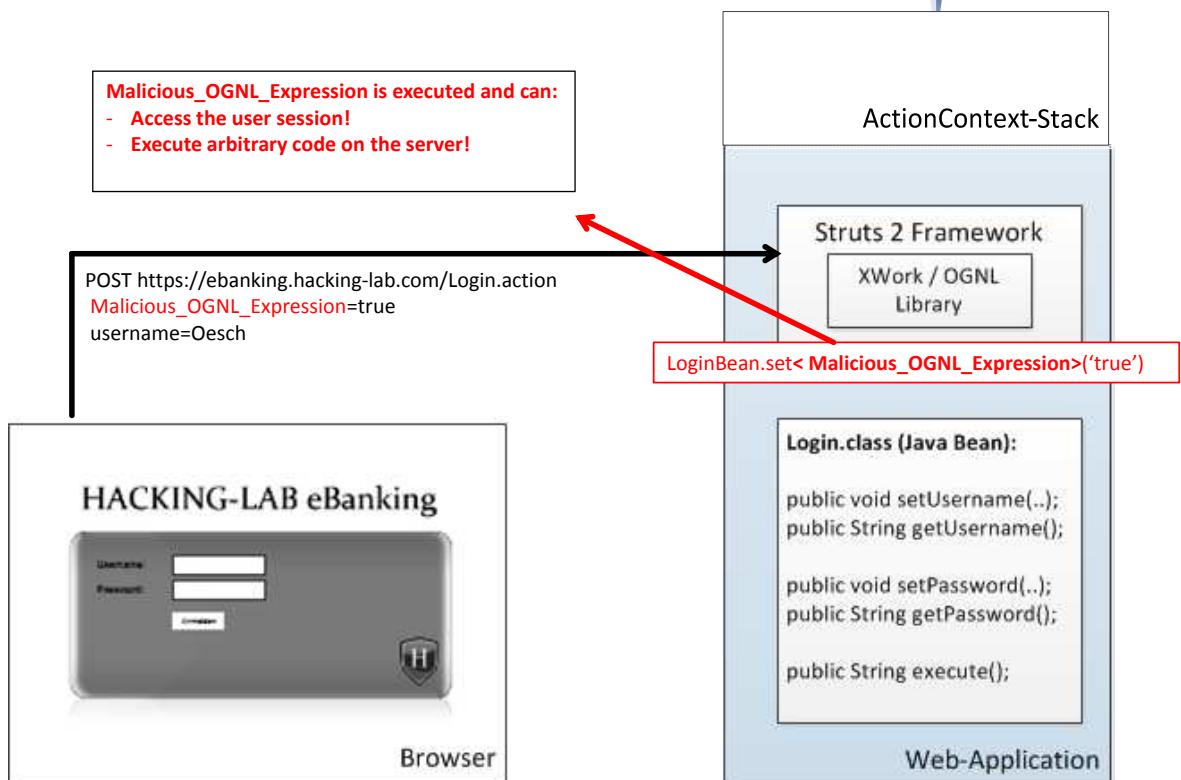
Normal Request





Struts 2 - Live Hacking Demo

Remote Code Execution Vulnerability S2-003/S2-005



Vulnerability S2-003 (Struts 2.0.0 - Struts 2.0.11.2)

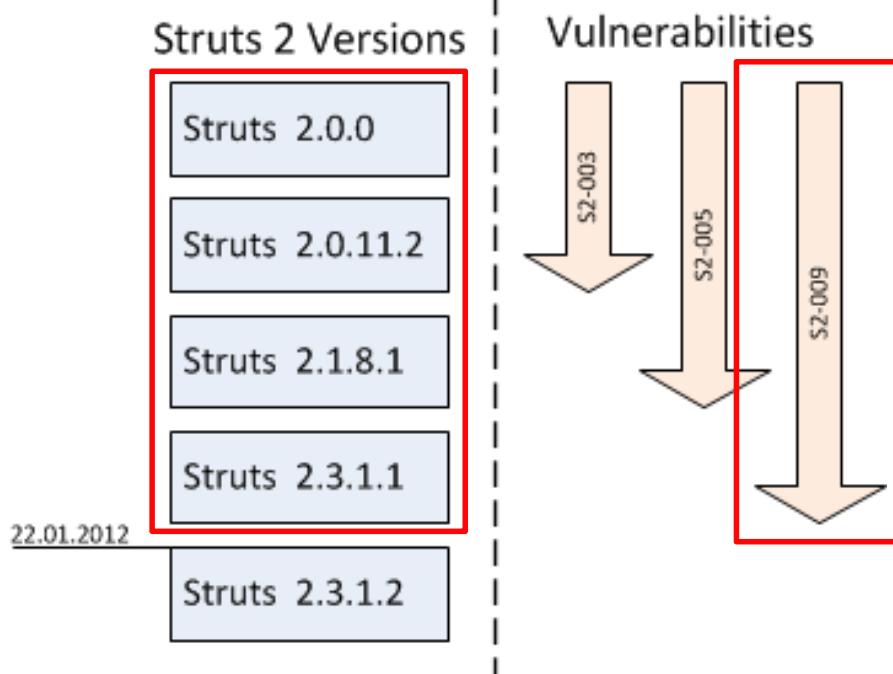
- ◆ Problem: malicious OGNL expression in parameter name
- ◆ Patch: Regexp for allowed parameter names (Whitelist)

Whitelist was not restrictive enough -> S2-005!

Vulnerability S2-005 (Struts 2.0.0 - Struts 2.1.8.1)

- ◆ Problem: malicious OGNL expression in parameter name
- ◆ Patch: Improved regexp for allowed parameter names (Whitelist)

Malicious OGNL expression in parameter name was not possible anymore but remote code execution was still possible! -> S2-009!



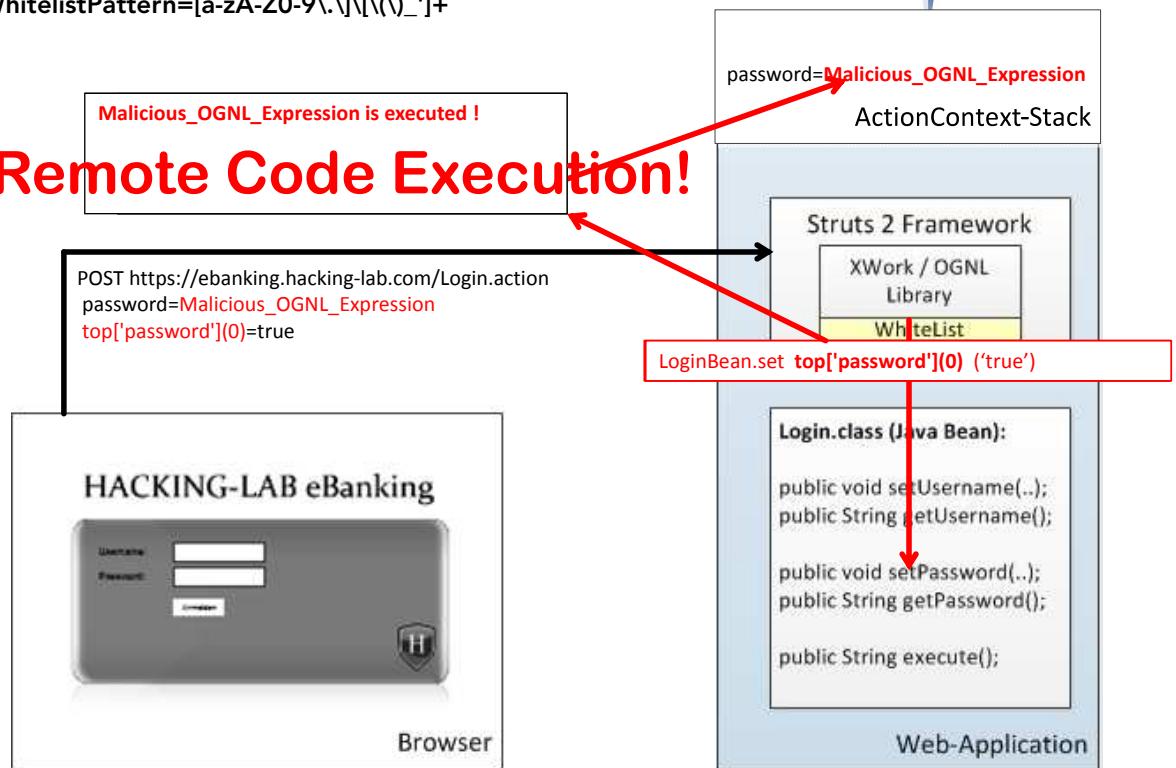
Struts 2 - Live Hacking Demo

Remote Code Execution Vulnerability S2-009

WhitelistPattern=[a-zA-Z0-9\.\]\[\(\)_]+



Remote Code Execution!



Struts 2 - Live Hacking Demo



The following security bulletins are available:

- * S2-001 - Report of an exploit on form validation error
- * S2-002 - Cross Site Scripting (XSS) vulnerability on <s:url> and <s:a> tags
- * S2-003 - ParameterInterceptors bypass allows OGNL statement execution
- * S2-004 - Directory traversal vulnerability while serving static content
- * S2-005 - XWork ParameterInterceptors bypass allows remote command execution

Struts 2 – Source Code Check



Comparing the Source Code:

- ◆ Vulnerable version (S2-009): Struts 2.3.1.1/ognl 3.0.3 /xwork-core.2.3.1.1
 - ◆ Following pattern was used to filter attack string in attribute names
 - ◆ Pattern=`[a-zA-Z0-9\.\]\N[\(\)]_+`
- ◆ Attack String in parameter name: `top['password'](0)`
- ◆ Current Version: Struts 2.3.1.2 / ognl 3.0.4 / xwork-core.2.3.1.2
 - ◆ Pattern for attribute name changed
 - ◆ New Pattern=`\w+((\.\w+)|([\d+])|(\d+))|(\['\w+'\])|(\('\w+'\))*`

Upgrade to Struts 2.3.1.2.



Struts²

Upgrade to Struts 2.3.1.2!



Struts2 Metasploit Module



Apache Struts < 2.2.0 Remote Command Execution

This module exploits a remote command execution vulnerability in Apache Struts versions < 2.2.0. This issue is caused by a failure to properly handle unicode characters in OGNL extensive expressions passed to the web server. By sending a specially crafted request to the Struts application it is possible to bypass the "#" restriction on ParameterInterceptors by using OGNL context variables. Bypassing this restriction allows for the execution of arbitrary Java code.

Rank

Excellent

Authors

bannedit < bannedit [at] metasploit.com >
Meder Kydryaliev < >

References

CVE-2010-1870
OSVDB-66280
<http://www.exploit-db.com/exploits/14360/>

http://www.metasploit.com/modules/exploit/multi/http/struts_code_exec

Dependence



springsource community

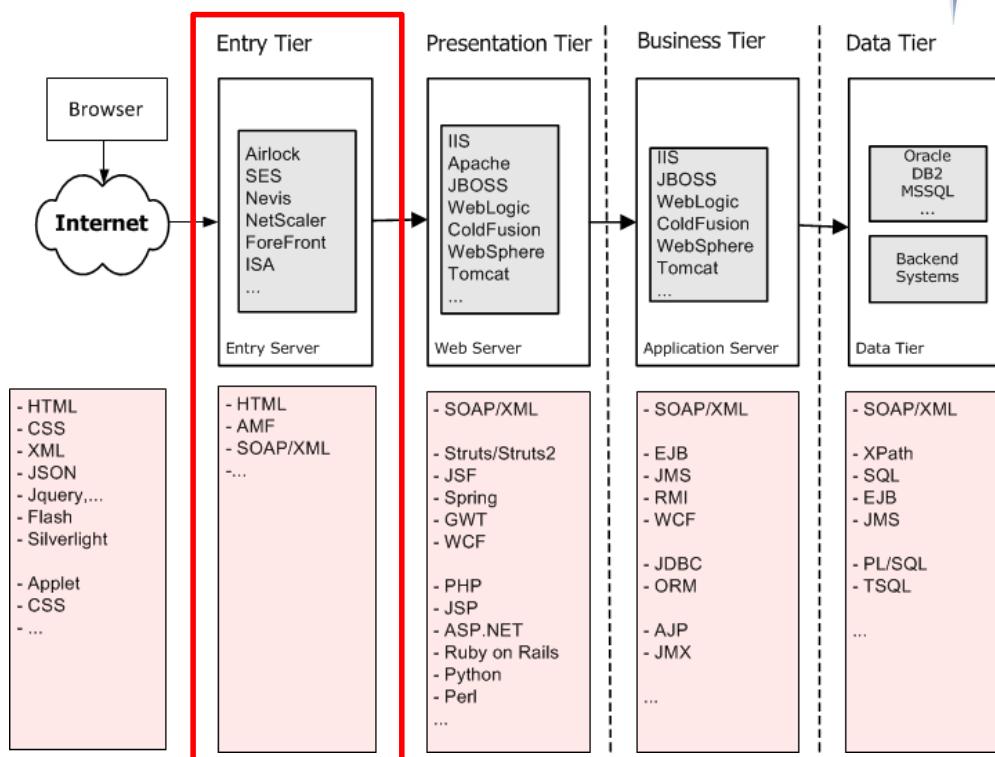


Foreign products & frameworks
are not controllable!



The Apache
Software Foundation
<http://www.apache.org>

How to protect?



© Compass Security AG

www.csnc.ch

Slide 27

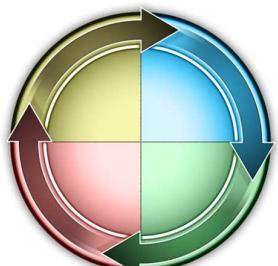
Recommendations



Entry Server (WAF) – InputFilter on parameters & requests

- ❖ Whitelisting of expected parameters/requests
- ❖ Blacklisting of known attack vectors

Improve collaboration between development & operation:



Define:

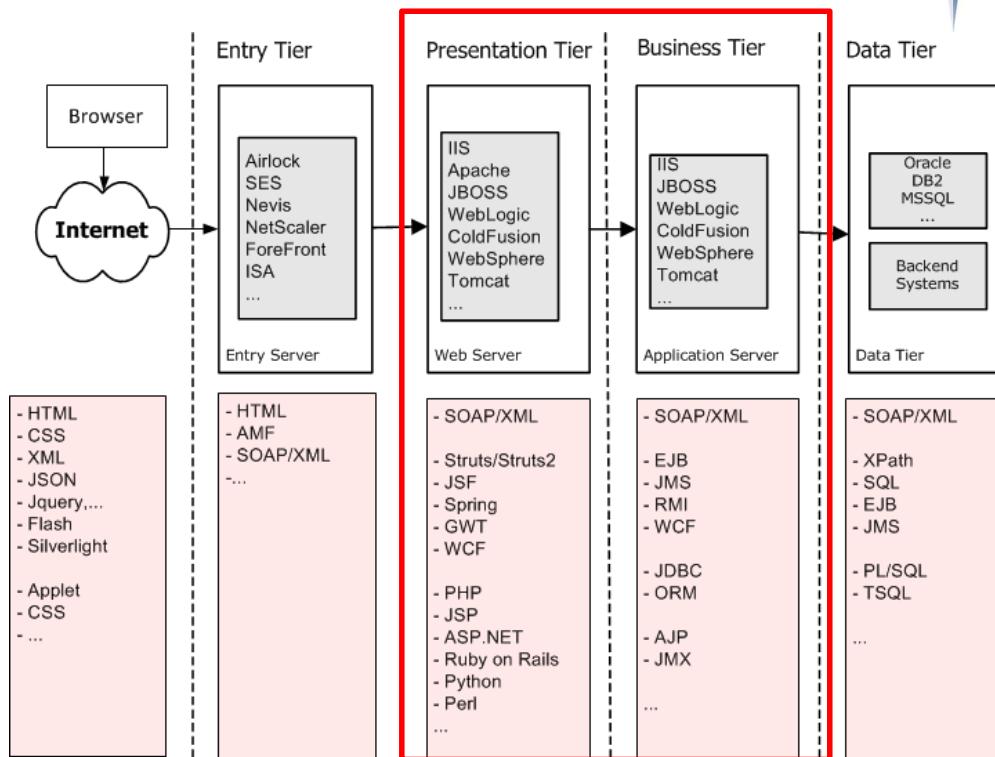
- ❖ Interfaces between different systems
- ❖ Transferred data
- ❖ Technologies and frameworks
- ❖ WAF – Input Filter Configuration

© Compass Security AG

www.csnc.ch

Slide 28

How to protect?



Recommendations



Company guidelines



- ◆ Technologies & framework guidelines
- ◆ Clear competence & responsibility

Web application



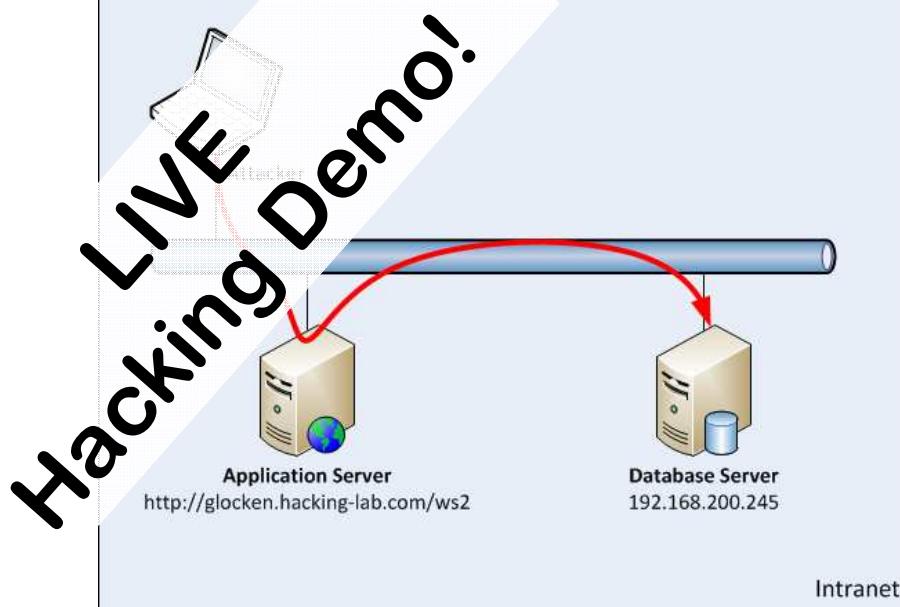
- ◆ Limit technologies & frameworks (keep it simple)
- ◆ Deploy only the required set of functions and libraries
- ◆ Track versions and update regularly
- ◆ Secure coding & configuration

Update process

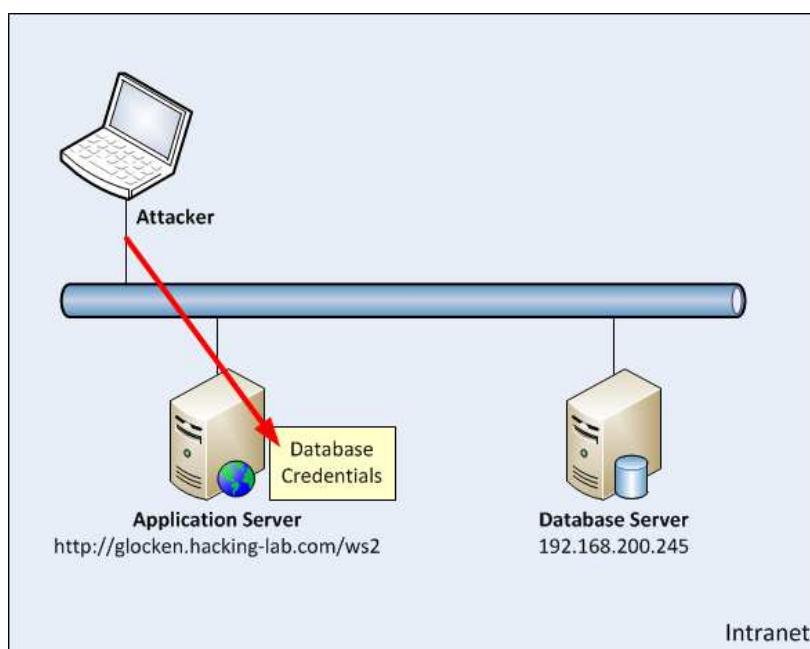


- ◆ Track the used versions and monitor new vulnerabilities
- ◆ Update all components
 - ◆ OS, Web Server, DB , ..
 - ◆ Also application libraries & frameworks!

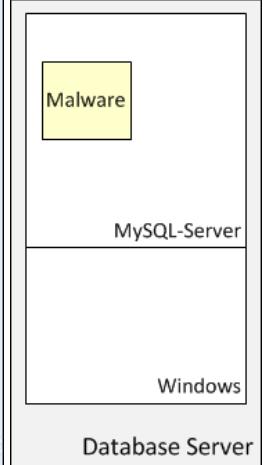
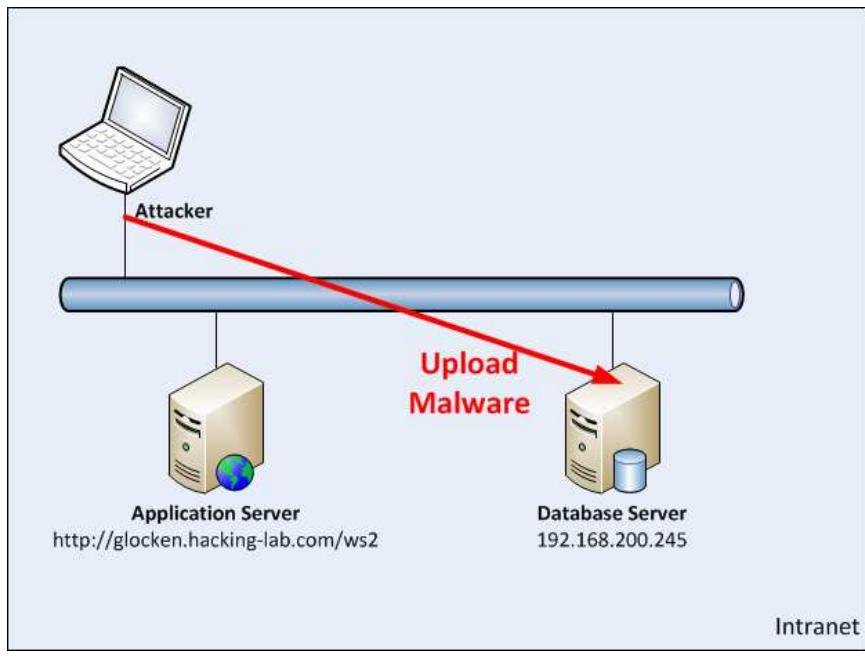
XXE UDF - Live Hacking Demo



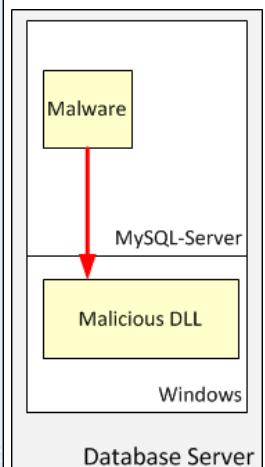
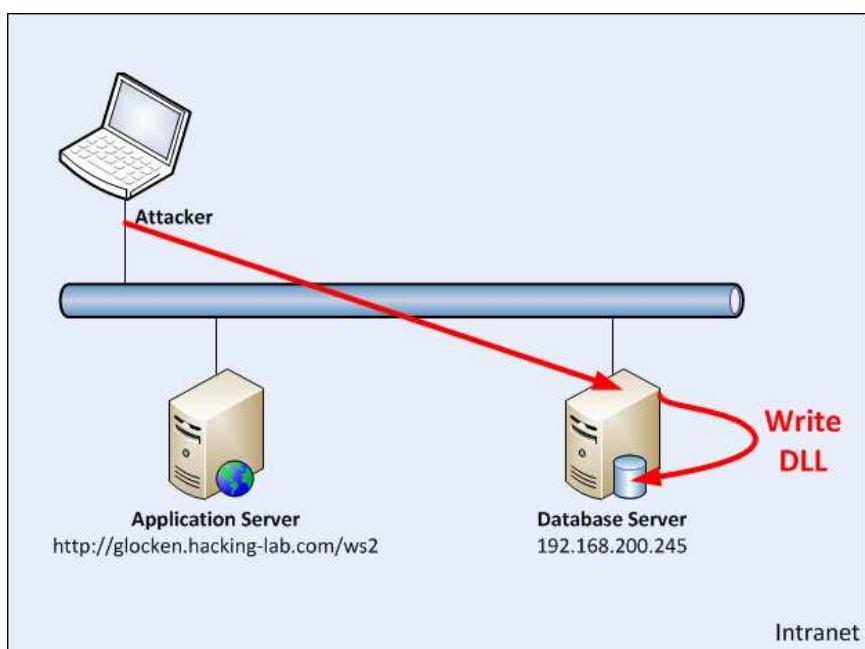
XXE UDF - Live Hacking Demo



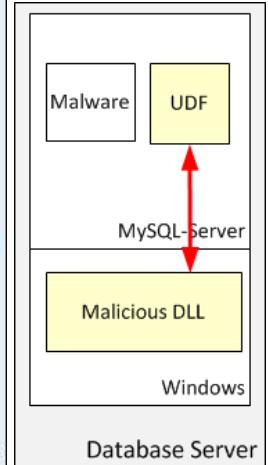
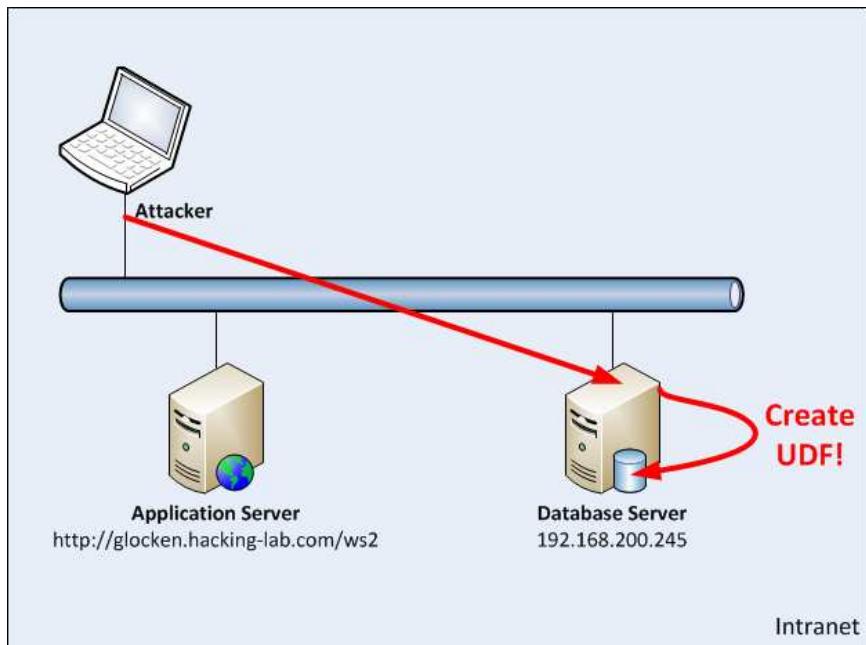
XXE UDF - Live Hacking Demo



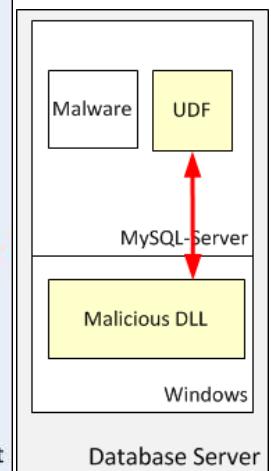
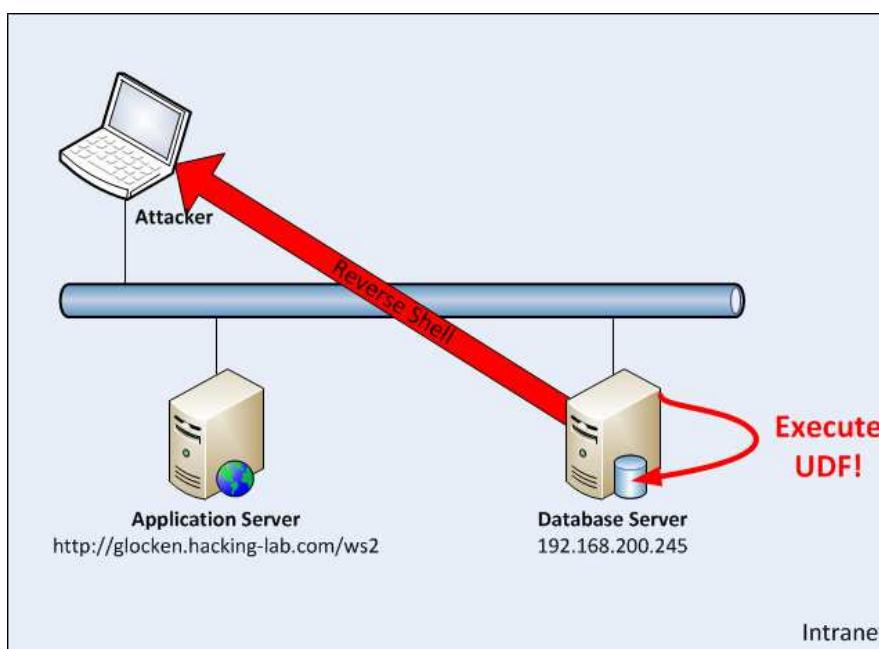
XXE UDF - Live Hacking Demo



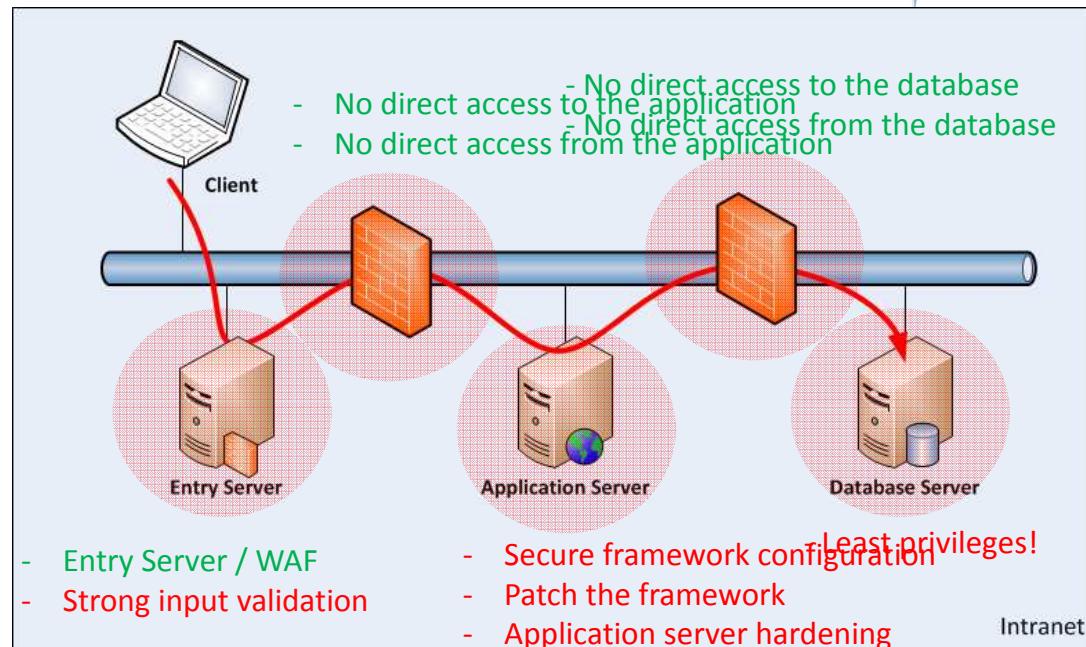
XXE UDF - Live Hacking Demo



XXE UDF - Live Hacking Demo



Where can we improve?



Try it yourself!



The screenshot shows the Hacking Lab website interface:

- Header:** H HACKING LAB, HOME, NINA, BeerTalk201203, FatTony, 0, 0, 0, 0.
- Left Sidebar:** Home, About, Volunteer, Partner & Sponsors, Job Opportunities, Events, Available Challenges, Remote Security Lab, Chat, Scoring System, Avatar, Mobile Services, Video Tutorials, Download, FAQ, Login / Sign up, My Menu, Edit My Profile.
- Challenge List:** BeerTalk201203
- User Statistics:** 5986 users, 5986 solved challenges.
- Table:** Displays solved challenges with columns: Topic, Type, Name, Level, Points, Duration, Solved by, Solution, Theory, Teacher Solution, Standard Solution, SBS Live CD, ASL Master Windows, ASL Master Linux.

Topic	Type	Name	Level	Points	Duration	Solved by	Solution	Theory	Teacher Solution	Standard Solution	SBS Live CD	ASL Master Windows	ASL Master Linux
7023 MySQL UDF Injection	SBS	7023 MySQL UDF Injection	3	0/15	15	0	☒	-	☒	☒			
7023 MySQL UDF Injection	WG	7023 MySQL UDF Injection	3	0/30	15	0	☒	-	☒	☒			
7025 Database Hijack CarGame Challenge	WG	7025 Database Hijack CarGame Challenge	3	0/30	120	0	☒	-	☒	☒			
7029 Tomcat Misconfiguration	SBS	7029 Tomcat Misconfiguration	2	0/10	20	0	☒	-	☒	☒			
7032 JSP Shell	SBS	7032 JSP Shell	2	0/15	30	0	☒	-	☒	☒			

Visit: <http://blog.csnc.ch>



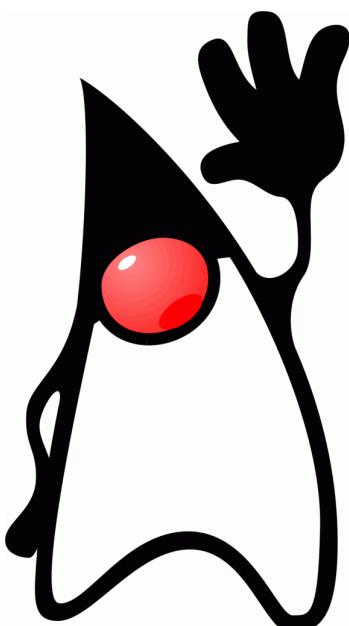
A screenshot of a Mozilla Firefox browser window. The title bar reads "New Security Enhancing HTTP Headers | Compass Security Blog - Mozilla Firefox". The address bar shows the URL "http://blog.csnc.ch/2012/01/new-http-headers/". The page content is a blog post titled "Compass Security Blog" with the subtitle "A Blog about Information Security...". The post has a dark header bar with "Home" and "Contact" links. Below the header, it says "Posted on January 31, 2012 by Alexandre Herzog" and includes "← Previous Next →" navigation links. The main body of the post is titled "New Security Enhancing HTTP Headers" and discusses proposed new HTTP Headers for web application security. A search bar is visible at the top right of the page area.

© Compass Security AG

www.csnc.ch

Slide 39

Thank You



© Compass Security AG

www.csnc.ch

Slide 40

References



<http://struts.apache.org/2.2.1/docs/s2-003.html>

<http://struts.apache.org/2.2.1/docs/s2-005.html>

<http://struts.apache.org/2.x/docs/s2-009.html>

<http://blog.csnc.ch/2012/01/new-http-headers/>