



PHP Security

MyGOSSCON 2010

Sumardi Shukor

Web & Mobile Apps Developer

Security? Why bother?



Loss of business

Destroy customer confidence

Legal liability

Financial loss

Cost of incident handling

Web applications Attack

How To Write Secure Code w/ <?PHP

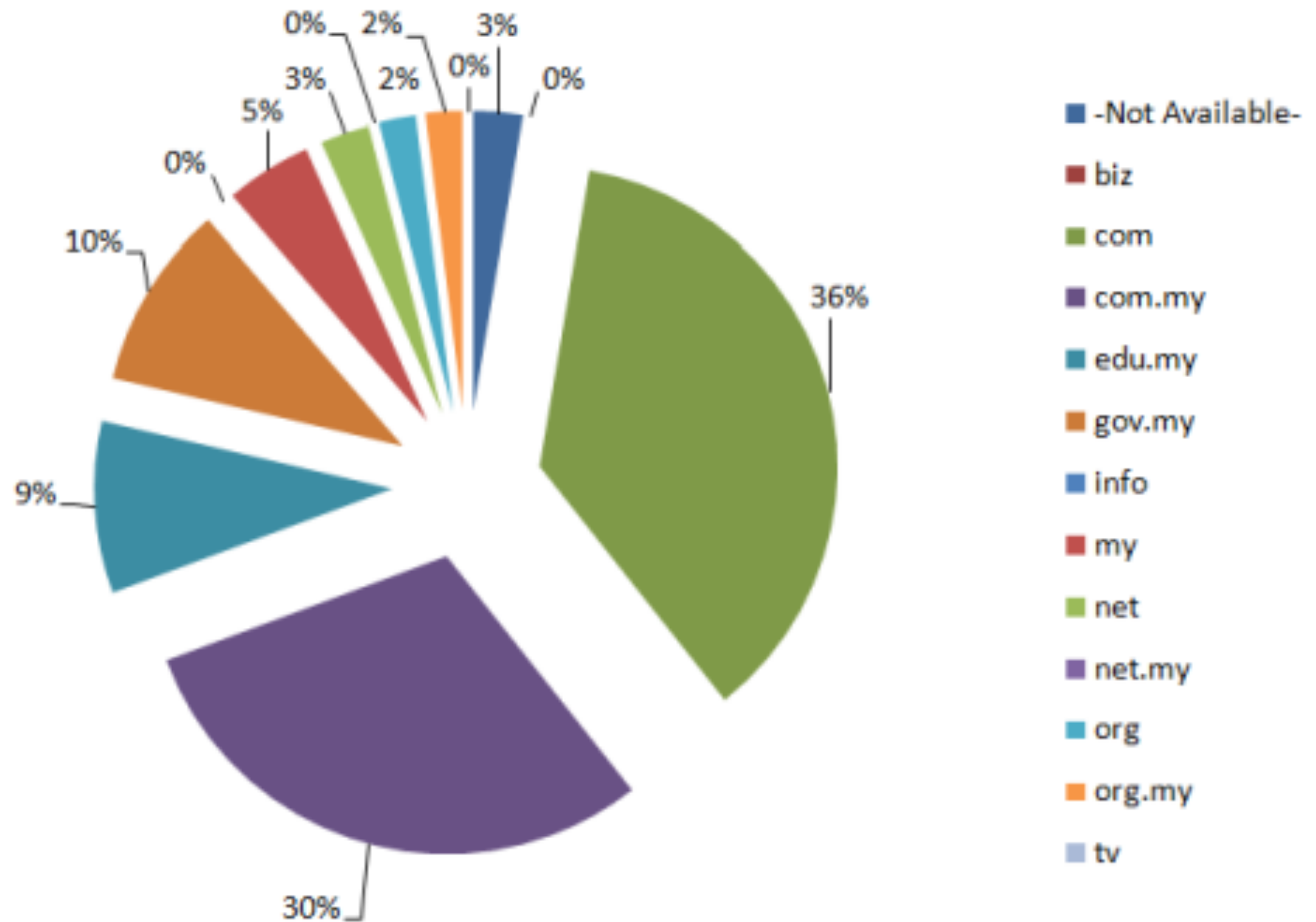
... and maybe MySQL

Categories of Incidents	Quarter	
	Q1 2010	Q4 2009
Drones Report	130	34
Denial of Service	18	2
Fraud and Forgery	446	318
Vulnerability Probing	78	28
Harassment	57	36
Indecent Contents	6	2
Malicious Codes	131	98
System Intrusion	504	404
TOTAL	1370	922

Figure 2: Comparison of Incidents between Q1 2010 and Q4 2009

* source : MyCERT

Percentage of Web Defacement by Domain in Q1 2010



* source : MyCERT

PHP Vulnerabilities



SQL Injection

Cross Site Scripting

File Includes

Unfiltered Input

File Storage & Permissions

Session Hijacking

PHP Vulnerabilities



SQL Injection

Cross Site Scripting

File Includes

Unfiltered Input

File Storage & Permissions

Session Hijacking

“SQL Injection is an old problem, so I don’t have to worry about it.”

- potential victim #1

HI, THIS IS
YOUR SON'S SCHOOL.
WE'RE HAVING SOME
COMPUTER TROUBLE.



OH, DEAR - DID HE
BREAK SOMETHING?
IN A WAY-



DID YOU REALLY
NAME YOUR SON
Robert'); DROP
TABLE Students;-- ?



OH. YES. LITTLE
BOBBY TABLES,
WE CALL HIM.

WELL, WE'VE LOST THIS
YEAR'S STUDENT RECORDS.
I HOPE YOU'RE HAPPY.



AND I HOPE
YOU'VE LEARNED
TO SANITIZE YOUR
DATABASE INPUTS.

<http://xkcd.com/327/>


Cases of SQL Injection

- (December 2009) Facebook game maker RockYou! attacked using SQL Injection, exposing 32M plaintext usernames and passwords.

Query needs a dynamic value

```
SELECT * FROM bugs
```

```
WHERE bug_id = $bug_id
```



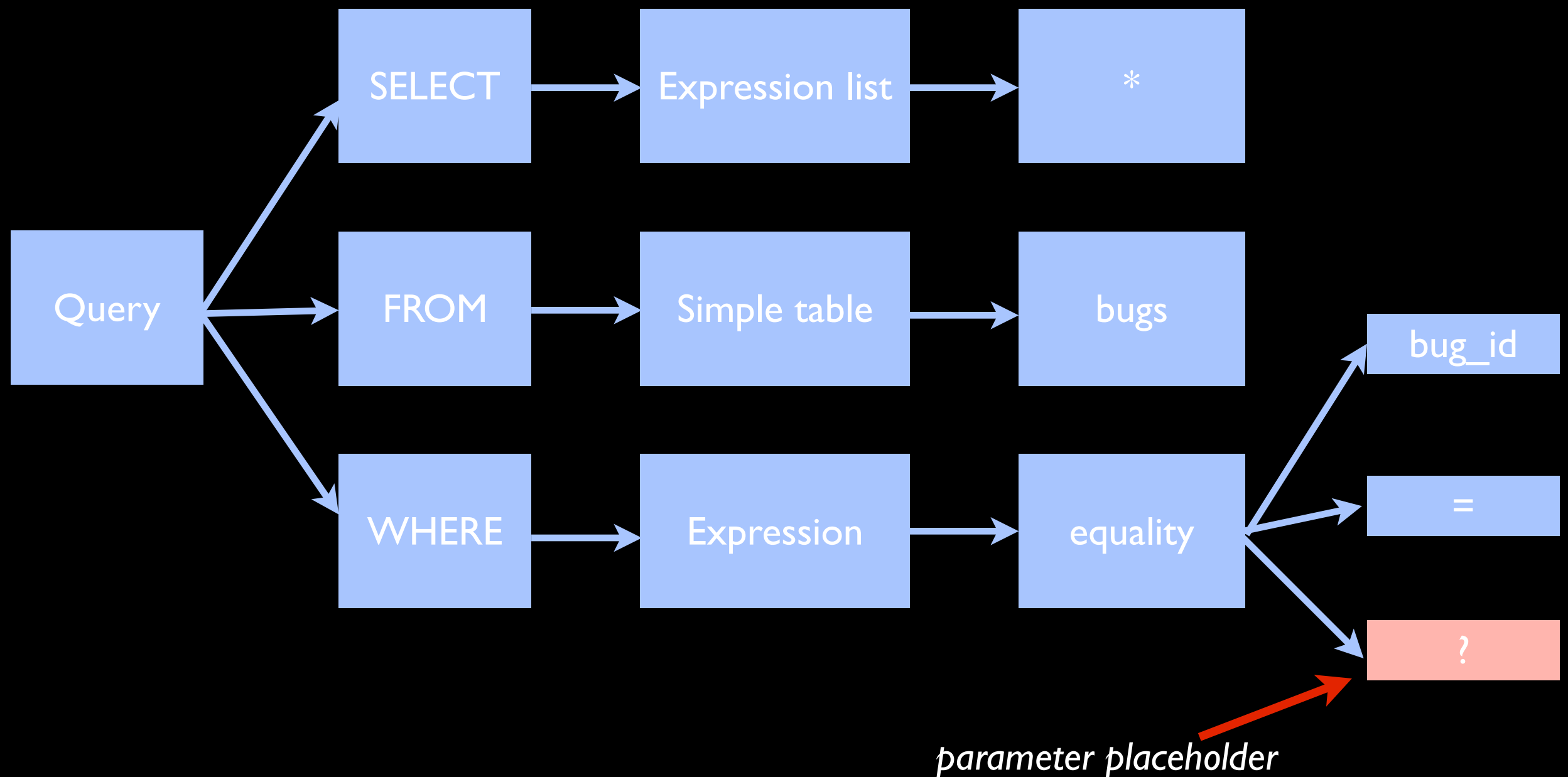
user input

Query parameter takes the place of dynamic value

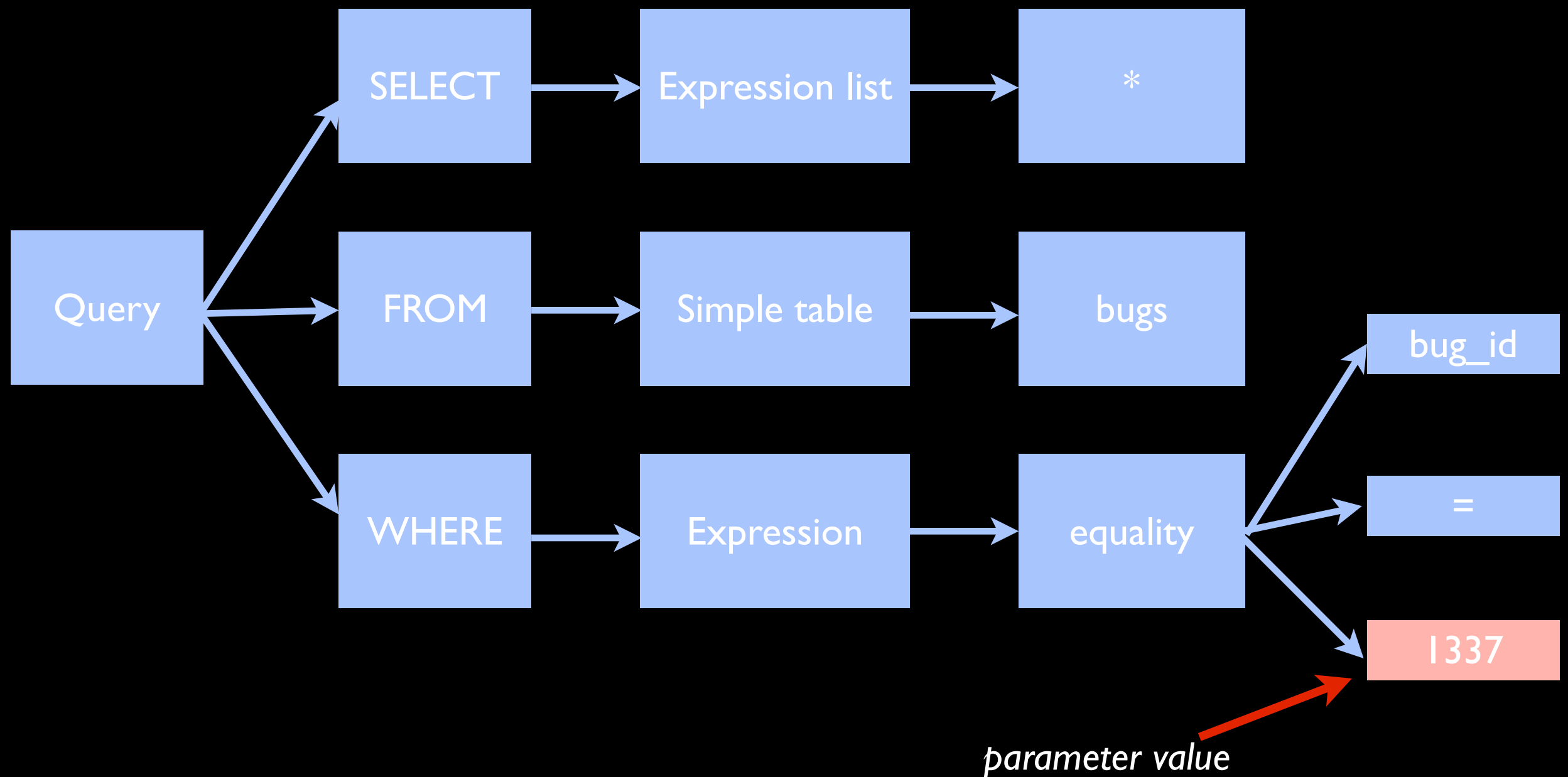
```
SELECT * FROM bugs
```

```
WHERE bug_id = 1337 OR TRUE
```

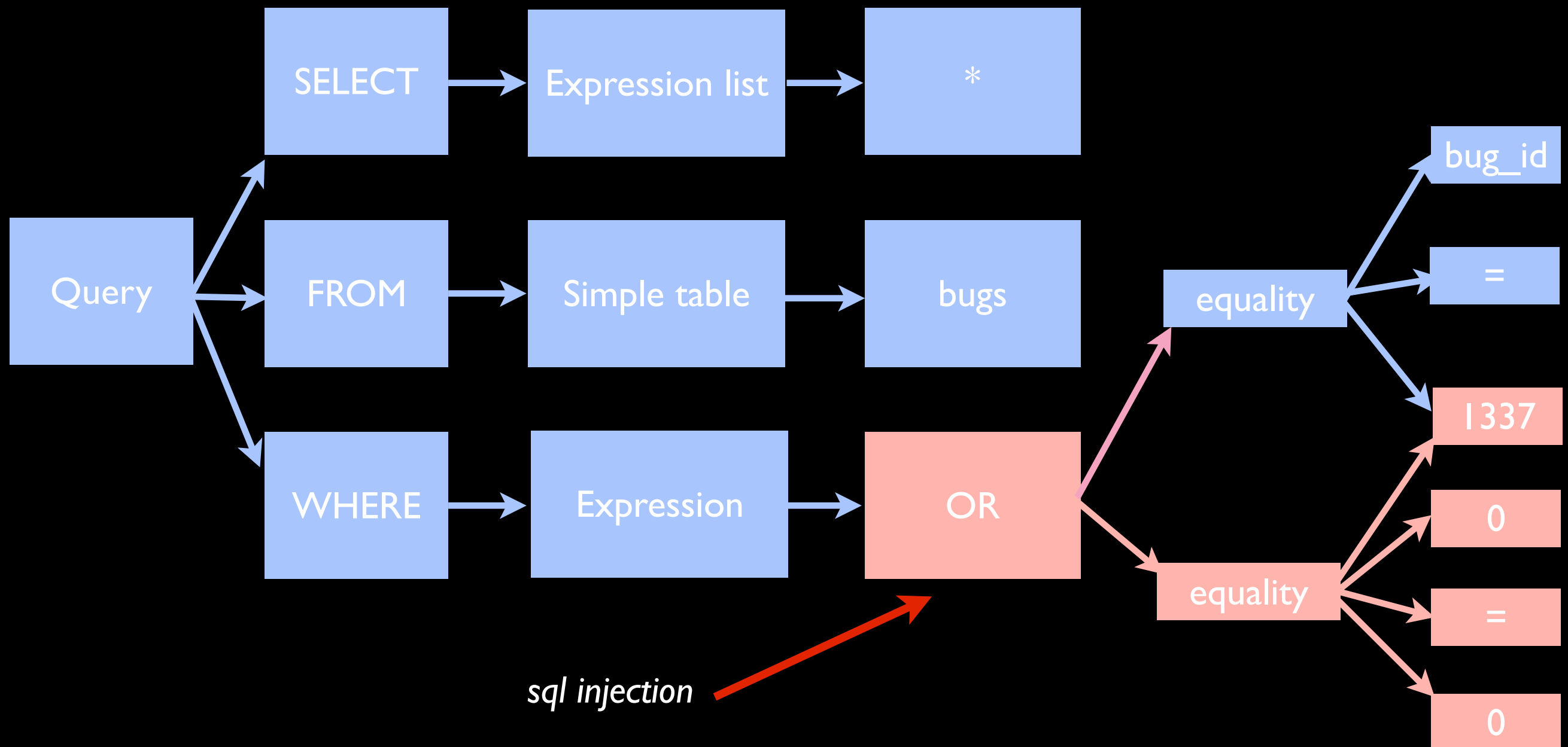
How the database parses it?



How the database execute it?



SQL Injection



<http://bit.ly/atHL2B>

DEMO

Suggested solution

`addslashes ()` isn't good enough.

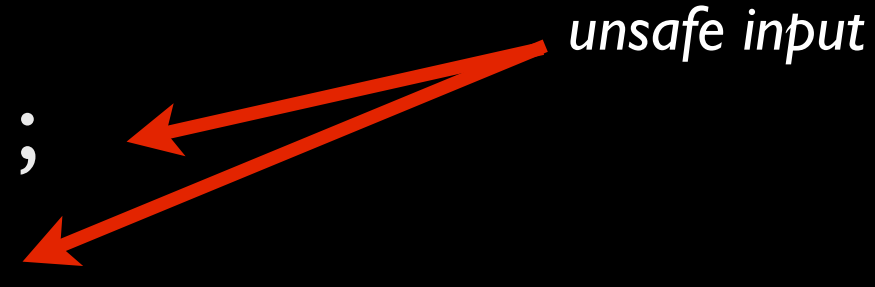
Please use driver-provided functions:

`mysql_real_escape_string ()`;

`PDO::quote ()`;

Example : Safer Solution

```
$sortorder = $_GET['order'];  
$direction = $_GET['dir'];  
  
$sql = "SELECT * FROM bugs  
        ORDER BY {$sortorder} {$direction}";  
$query = mysql_query($sql) or die(mysql_error());
```



The diagram consists of two red arrows originating from the text "unsafe input" on the right. One arrow points to the value 'order' in the first line of code, and the other points to the value 'dir' in the second line of code, indicating that these are the sources of unsafe input.

<http://example.com/list.php?order=name&dir=down>

Fix with a whitelist map

```
$sortorders = array("name" => "name",  
                    "status" => "valid","invalid");
```

```
$directions = array("up" => "ASC",  
                    "down" => "DESC");
```

```
$sortorder = array_key_exists($_GET['order'], $sortorders) ?  
                $sortorders[$_GET['order']] : "bug_id";
```

```
$direction = array_key_exists($_GET['dir'], $directions) ?  
                $directions[$_GET['dir']] : "ASC";
```

“It’s just an intranet application, it doesn’t need to be secure.”

- potential victim #2

PHP Vulnerabilities



SQL Injection

Cross Site Scripting

File Includes

Unfiltered Input

File Storage & Permissions

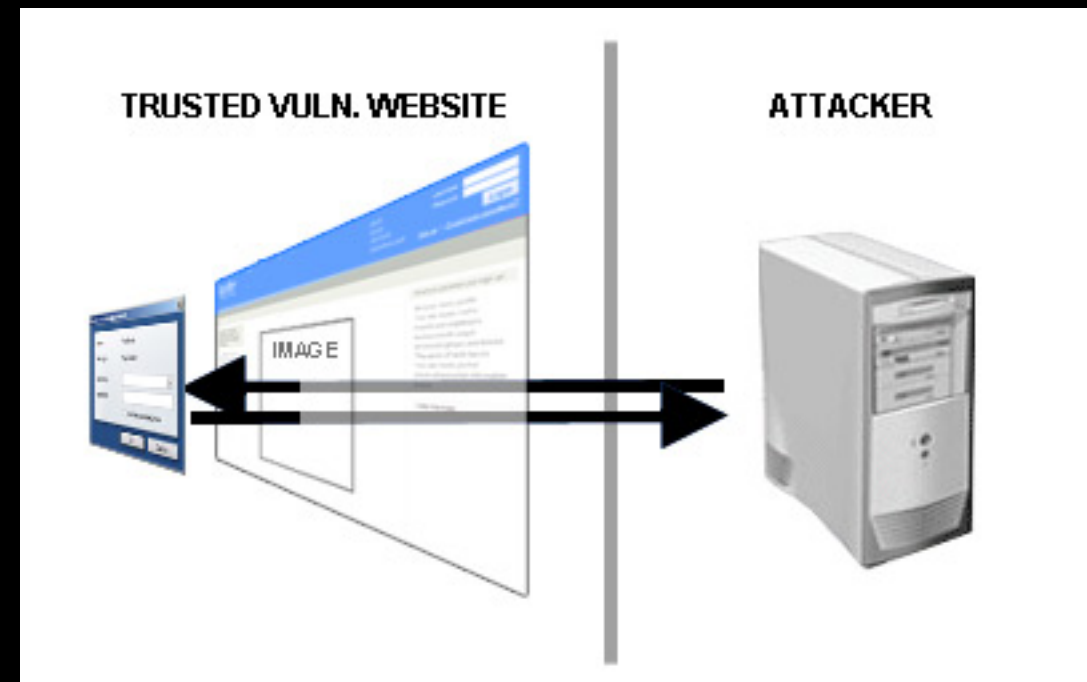
Session Hijacking

XSS Characteristics

Exploit the trust a user has for a particular site.

Generally involve web sites that display external data.

Inject content of the attacker's choosing.



What can you do?

Filter all external data.

Use existing functions. `htmlentities()`,
`strip_tags()` and `utf8_decode()`.

Use a whitelist approach.

DEMO

PHP Vulnerabilities



SQL Injection

Cross Site Scripting

File Includes

Unfiltered Input

File Storage & Permissions

Session Hijacking

Insecure include

```
$language = "english";
```

```
if(!empty($_POST['lang'])) {  
    $language = $_POST['lang'];  
}
```

```
if(file_exists("./language/{$language}/install.php")) {  
    include_once("./language/{$language}/install.php");  
}
```

DEMO

PHP Vulnerabilities



SQL Injection

Cross Site Scripting

File Includes

Unfiltered Input

File Storage & Permissions

Session Hijacking

Best Practice

Filter input, Escape Output

Input Validation

User input is unreliable and not to be trusted.

Which is why it is absolutely important to validate any user input before use.

DEMO

Numeric Value Validation

All data passed to PHP ends up being a string.

Casting is a simple and very efficient way to ensure variables do in fact contain numeric values.

```
// integer validation
if(!empty($_GET['id'])) {
    $id = (int) $_GET['id'];
} else {
    $id = 0;
}
```

```
// float validation
if(!empty($_GET['price'])) {
    $price = (float) $_GET['price'];
} else {
    $price = 0;
}
```

String Validation

PHP comes with a `ctype`, an extension that offers a very quick mechanism for validating string content.

```
if(!ctype_alnum($_GET['username'])) {  
    echo "Only A-Za-z0-9 are allowed.";  
}
```

```
if(!ctype_alpha($_GET['name'])) {  
    echo "Only A-Za-z are allowed.";  
}
```

PHP Vulnerabilities



SQL Injection

Cross Site Scripting

File Includes

Unfiltered Input

File Storage & Permissions

Session Hijacking

Sensitive Files in Webroot

Some people fail to see the danger of keeping backups, passwords and private documents in web accessible directories.



<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 BBoard/	07-May-2008 09:14	-	
 Courses/	07-May-2008 09:14	-	
 Default/	04-Dec-2008 15:40	-	
 Guest/	07-May-2008 09:14	-	
 Profile/	13-Feb-2008 10:48	-	
 Reports/	13-Feb-2008 10:48	-	
 Search/	07-May-2008 09:14	-	
 SignUp/	07-May-2008 09:14	-	
 Stu_connection.php	13-Feb-2008 10:48	509	
 Stu_connection1.php	13-Feb-2008 10:48	407	
 login/	04-Dec-2008 15:32	-	
 lostpasswd/	13-Feb-2008 10:48	-	
 newsendmail.php	13-Feb-2008 10:48	2.2K	
 phpinfo33.php	13-Feb-2008 10:48	133	
 resources/	07-May-2008 09:14	-	
 test.php	13-Feb-2008 10:48	1.8K	
 uploadmaterial/	07-May-2008 09:14	-	
 verticalscroll.php	13-Feb-2008 10:48	9.9K	
 zct8EC7.tmp	13-Feb-2008 10:48	0	

Apache/2.2.3 (Red Hat) Server at spin.medic.ukm.my Port 80

Solution

Do not keep sensitive data in web accessible directories.

Do not permit directory browsing.

Use http authentication to protect sensitive directories.

Avoid clear-text passwords.

PHP Vulnerabilities



SQL Injection

Cross Site Scripting

File Includes

Unfiltered Input

File Storage & Permissions

Session Hijacking

Securing Session ID

To prevent Session ID theft, the ID can be altered on every request, invalidating old values.

```
session_start();  
if(!empty($_SESSION)) { // not a new session  
    session_regenerate_id(TRUE);  
    // make a new session id  
}
```

Securing Session

Another session security technique is to compare the browser signature headers.

```
session_start();
$check = @md5($_SERVER['HTTP_ACCEPT_ENCODING'] .
              $_SERVER['HTTP_ACCEPT_LANGUAGE'] .
              $_SERVER['HTTP_USER_AGENT']);

if(empty($_SESSION)) {
    $_SESSION['key'] = $chk;
} elseif ($_SESSION['key'] != $chk) {
    session_destroy();
}
```

THANK YOU ;)

smd@php.net.my
twitter.com/sumardi