

# Externe Skripte erstellen

Damit man bei der Installation flexibel ist, kann man mit dem [Paket Installations Skript](#) andere Skripte aufrufen, welche dann z.B ein Passwort mit einen Salt erstellen.

Bei den schon erstellen Paketen sind verschiedene Möglichkeiten genutzt worden, welche ich Euch hier mal zeigen werde.

## get\_secure\_keys.php (Wordpress)

In Wordpress benötigt man 4 verschiedene Secretkeys, welche dann in der wp-config.php eingetragen werden. Wordpress.org bietet eine API an um sich diese Key dynamisch zu erstellen lassen. Dies API nutzen wir um die Secretkeys dynamisch in die Installationen eintragen zu lassen:

```
<?php
$content = file_get_contents("http://api.wordpress.org/secret-key/1.1/");
print $content;
?>
```

Beispiel für den Aufruf aus dem [Paket Installations Skript](#):

```
my $securekeys = ` $main::cfg{'CMD_PHP'} -n ./get_secure_keys.php `;
```

## make\_password.php (Wordpress)

Es gibt sogar die Möglichkeit Funktionen von der installierten Webapplikation zu nutzen. Das funktioniert nicht immer. Solche Dinge sollte man immer vorher auf der Linux-Konsole mit einem einfachen PHP-Skript testen:

```
<?php
$user_password = $argv[1];
$absolut_path = $argv[2];
require_once($absolut_path.'/wp-includes/class-phpass.php');
$wp_hasher = new PasswordHash(8, TRUE);
$user_password = $wp_hasher->HashPassword($user_password);
print $user_password;
?>
```

Beispiel für den Aufruf aus dem [Paket Installations Skript](#):

```
my $loginpasswordhash = ` $main::cfg{'CMD_PHP'} -n ./make_password.php
$sw_install_pass_clear '$sw_dest_path' `;
```

# make\_password.php, salt.php (Joomla 2.5.1)

Joomla nutzt einen Salt bei dem erstellen eines Passwortes. Zusätzlich wird der Salt dem Passwort angehängen (Passwort:Salt).

Zuerst muss ein dynamischer Salt erstellt werden:

```
<?php
function random_str(){
    $result="";
    for ($i = 1; $i <= 32; $i++) {
        $base10Rand = mt_rand(0, 15);
        $newRand = base_convert($base10Rand, 10, 36);
        $result.=$newRand;
    }
    return $result;
}
print random_str();
?>
```

Anschließend wird was Passwort mit dem erstellen Salt erstellt:

```
<?php
$user_password = $argv[1];
$salt = $argv[2];
$user_password_hash = md5($user_password.$salt);
$user_password_hash = $user_password_hash.':'.$salt;
print $user_password_hash;
?>
```

Beispiel für den Aufruf aus dem [Paket Installations Skript](#):

```
my $salt = `$main::cfg{'CMD_PHP'} -n ./salt.php`;
my $loginpasswordhash = `$main::cfg{'CMD_PHP'} -n ./make_password.php
$sw_install_pass_clear $salt`;
```

# securekey.php (Joomla 2.5.1)

In der Datei "configuration.php" von Joomla wird auch ein Securitykey eingetragen. Dieser sollte natürlich bei jeder Installation verschieden sein.

Aus diesem Grund nutze ich dieses Skript um einen 16-stelligen Code zu erstellen, der dann in die "configuration.php" geschrieben wird.

```
<?php
function random_str(){
    $result="";
```

```
    for ($i = 1; $i <= 16; $i++) {  
        $base10Rand = mt_rand(0, 15);  
        $newRand = base_convert($base10Rand, 10, 36);  
        $result.=$newRand;  
    }  
    return $result;  
}  
print random_str();  
?>
```

Beispiel für den Aufruf aus dem [Paket Installations Skript](#):

```
my $securekey = `$main::cfg{'CMD_PHP'} -n ./securekey.php`;
```

## mail\_hash.php (phpBB 3.0.10)

Bei phpBB werden die Emailadressen in der Datenbank gehasht. Dazu nutzen wir folgendes Skript:

```
<?php  
$mail = $argv[1];  
function email_hash($email) {  
    return sprintf('%u', crc32(strtolower($email))) . strlen($email);  
}  
print email_hash($mail);  
?>
```

Beispiel für den Aufruf aus dem [Paket Installations Skript](#):

```
my $mail_hash = `$main::cfg{'CMD_PHP'} -n ./mail_hash.php  
$sw_install_email`;
```

From:

<https://wiki.i-mscp.net/> - **i-MSCP Documentation**

Permanent link:

[https://wiki.i-mscp.net/doku.php?id=de:start:howto:package\\_external\\_scripts](https://wiki.i-mscp.net/doku.php?id=de:start:howto:package_external_scripts)

Last update: **2012/03/04 14:42**

