

Fehler Passwort neu zusenden lassen

Betroffene Versionen:

- 4.8.16
- 4.8.17
- 4.8.18
- 4.8.19
- 4.8.20

Fehlerbeschreibung:

Wenn sich ein Benutzer (beim Loginfenster) ein neues Passwort zusenden lässt wird das neu versendete Passwort zuerst im Datenfeld tmp_pw_request gespeichert. Dies wird so gemacht damit ein Passwort neu setzen durch einen unbefugten verhindert wird. In früheren Versionen war dies wohl möglich.

Nun ist die Logik so dass beim ersten Login mit dem neu zugesandten Passwort das Passwort vom Datenfeld tmp_pw_request in das Datenfeld password kopiert wird. Dies funktioniert aber in oben genannten Versionen aber nicht da es einen Fehler in der /conlib/local.php gibt. Der Fehler ist relativ einfach (es sind sogar zwei Fehler). Beim setzen des Feldes password wurde vergessen das next_record() vor dem Update-Befehl auszuführen und zusätzlich gibt es einen SQL-Fehler wegen eines überflüssigen Kommas.

Fehlerbehebung:

Suche in der Datei /conlib/local.php (ca. Zeile 586) nach:

```
$this->db->query(sprintf("select user_id,perms,password,tmp_pw_request from %s where username = '%s' AND
(valid_from <= '". $sDate.'" OR valid_from = '0000-00-00' OR valid_from is NULL) AND
(valid_to >= '". $sDate.'" OR valid_to = '0000-00-00' OR valid_to is NULL)",
    $this->database_table,
    Contenido_Security::escapeDB($username, $this->db)));

// check if requested new password is used, if set equal password
$sRequestPassword = $this->db->if ("tmp_pw_request");
$this->auth['pwr'] = false;
if (md5($password) == $sRequestPassword) {
    $sQuery = "UPDATE ".$this->database_table." SET password = '". $sRequestPassword."',
    tmp_pw_request = NULL,
    using_pw_request = 1,
    WHERE username = '".Contenido_Security::escapeDB($username, $this->db)."'";
    $this->db->query($sQuery);
    $pass = $sRequestPassword;
    $this->auth['pwr'] = true;
}

$sMaintenanceMode = getSystemProperty('maintenance', 'mode');
while($this->db->next_record()) {
```

Ersetze ihn durch:

```
$this->db->query(sprintf("select user_id,perms,password,tmp_pw_request from %s where username = '%s' AND
(valid_from <= '". $sDate.'" OR valid_from = '0000-00-00' OR valid_from is NULL) AND
(valid_to >= '". $sDate.'" OR valid_to = '0000-00-00' OR valid_to is NULL)",
    $this->database_table,
    Contenido_Security::escapeDB($username, $this->db)));

// EDIT LW, WL - Fehlerbehebung Login nach zusenden eines neues Passworts
// neu:
$this->db->next_record();
// END EDIT LW, WL - Fehlerbehebung Login nach zusenden eines neues Passworts

// check if requested new password is used, if set equal password
$sRequestPassword = $this->db->if ("tmp_pw_request");
$this->auth['pwr'] = false;
if (md5($password) == $sRequestPassword) {
    // EDIT LW, WL - Fehlerbehebung Login nach zusenden eines neues Passworts
    // - SQL-Fehler behoben (Komma nach using_pw_request = 1 entfernt)
    $sQuery = "UPDATE ".$this->database_table." SET password = '". $sRequestPassword."',
    tmp_pw_request = NULL,
    using_pw_request = 1
    WHERE username = '".Contenido_Security::escapeDB($username, $this->db)."'";
    // END EDIT LW, WL - Fehlerbehebung Login nach zusenden eines neues Passworts

    $this->db->query($sQuery);
    $pass = $sRequestPassword;
    $this->auth['pwr'] = true;
}

// EDIT LW, WL - Fehlerbehebung Login nach zusenden eines neues Passworts
// - neu, erneute Abfrage der Userdaten:
$this->db->query(sprintf("select user_id,perms,password,tmp_pw_request from %s where username = '%s' AND
(valid_from <= '". $sDate.'" OR valid_from = '0000-00-00' OR valid_from is NULL) AND
(valid_to >= '". $sDate.'" OR valid_to = '0000-00-00' OR valid_to is NULL)",
    $this->database_table,
    Contenido_Security::escapeDB($username, $this->db)));
// END EDIT LW, WL - Fehlerbehebung Login nach zusenden eines neues Passworts

$sMaintenanceMode = getSystemProperty('maintenance', 'mode');
while($this->db->next_record()) {
```

- temp.txt - Zuletzt geändert: 2014/12/23 15:21 von admin