**1. Unauthenticated Arbitrary SQL Execution**

이 취약점은 inc/csv.php 파일에서 발생합니다. 이 파일에서는 사용자로부터 전달받은 POST 변수 중 $q를 통해 임의의 SQL 쿼리를 실행할 수 있도록 되어 있습니다.

<?php

*require\_once*('../../../../wp-load.php');

global $wpdb;

global $current\_user;

wp\_get\_current\_user();

*//   echo user\_login . "'s email address is: " . $current\_user->user\_pass;*

*//get admin pass for integrity check*

*// extract query*

$q = stripslashes($\_POST['q']);

$auth = stripslashes($\_POST['auth']);

$integ=stripslashes($\_POST['integ']);

*if*(wp\_automatic\_trim($auth == '')){

      echo 'login required';

*exit*;

}

*if*(wp\_automatic\_trim($auth) != wp\_automatic\_trim($current\_user->user\_pass)){

      echo 'invalid login';

*exit*;

}

*if*(md5(wp\_automatic\_trim($q.$current\_user->user\_pass)) != $integ ){

      echo 'Tampered query';

*exit*;

}

$rows=$wpdb->get\_results( $q);

$date=date("F j, Y, g:i a s");

$fname=md5($date);

header("Content-type: application/csv");

header("Content-Disposition: attachment; filename=$fname.csv");

header("Pragma: no-cache");

header("Expires: 0");

  echo "DATE,ACTION,DATA,KEYWORD \n";

*foreach*($rows as $row){

    $action=$row->action;

*if* (stristr($action , 'New Comment Posted on :')){

            $action = 'Posted Comment';

        }*elseif*(stristr($action , 'approved')){

            $action = 'Approved Comment';

    }

*//format date*

    $date=date('Y-n-j H:i:s',strtotime ($row->date));

    $data=$row->data;

    $keyword='';

*//filter the data strip keyword*

*if*(stristr($data,';')){

        $datas=explode(';',$row->data);

        $data=$datas[0];

        $keyword=$datas[1];

    }

      echo "$date,$action,$data,$keyword \n";

}

*//  echo "record1,$q,record3\n";*

?>

 **취약점 원인:**  
사용자로부터 전달받은 $q (SQL 쿼리)를 그대로 $wpdb->get\_results($q)를 통해 실행할 수 있습니다.

 **검증 우회:**  
wp\_automatic\_trim() 함수는 단순히 trim()만 수행합니다.

취약코드

function wp\_automatic\_trim($str)

{

    if (is\_null($str)) {

        return '';

    } else {

        return trim($str);

    }

}

기본적으로 $current\_user->user\_pass는 인증되지 않은 사용자의 경우 빈 문자열("")입니다.  
첫 번째 검증은 $auth가 빈 값인지 체크하는데, 이를 우회하기 위해 단일 공백(" ")을 넣으면 인증 체크를 통과할 수 있습니다.  
이후 $integ 값은 MD5 해시로 계산되는데, $current\_user->user\_pass가 빈 문자열이므로 우리가 공급한 SQL 쿼리의 MD5 값을 그대로 제공하면 검증을 우회할 수 있습니다.

원문링크

https://patchstack.com/articles/critical-vulnerabilities-patched-in-wordpress-automatic-plugin?\_s\_id=cve