SQL注入漏洞

base on pikachu 

Sql-Inject漏洞解析-课程目录

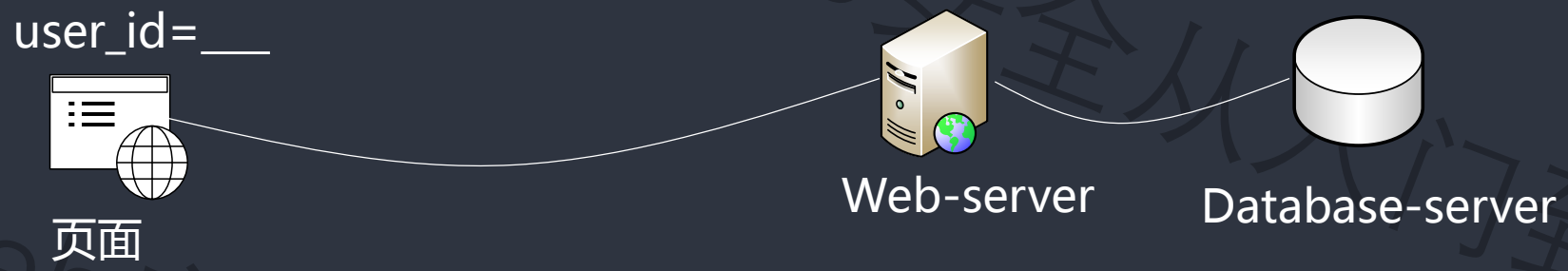
- SQL Inject 漏洞原理概述
- 如何判断注入点类型及常见注入类型讲解
- 注入方式get&post的区别
- SQL Inject漏洞手工测试：基于union联合查询的信息获取（ select ）
- SQL Inject漏洞手工测试：基于报错的信息获取(select/delete/update/insert)
- SQL Inject漏洞手工测试：操作系统权限获取
- SQL注入漏洞-盲注（ boolian base ）原理及测试
- SQL注入漏洞-盲注（ time base ）原理及测试
- SQL注入漏洞-基于http header的注入
- SQL注入表列明猜解-暴力破解在sqli上的应用
- SQL注入-宽字节注入原理及演示
- 如何使用SQL-Map进行SQL Inject漏洞测试
- SQL注入漏洞常见防范措施

SQL Inject漏洞概述

在owasp发布的top 10漏洞里面，注入漏洞一直是危害排名第一，其中主要指SQL Inject漏洞。

数据库注入漏洞，主要是开发人员在构建代码时，没有对输入边界进行安全考虑，导致攻击着可以通过合法的输入点提交一些精心构造的语句，从而欺骗后台数据库对其进行执行，导致数据库信息泄漏的一种漏洞。

SQL Inject漏洞概述



正常输入: 1

非法输入: 1 or 1=1

select email from users where id=1;

select email from users where id=1 or 1=1;

SQL Inject漏洞攻击流程

1

第一步：注入点探测

自动方式：使用web漏洞扫描工具，自动进行注入点发现

手动方式：手工构造sql inject测试语句进行注入点发现

2

第二步：信息获取

通过注入点取期望得到的数据。

1.环境信息：数据库类型,数据库版本,操作系统版本,用户信息等。

2.数据库信息：数据库名称,数据库表,表字段,字段内容（加密内容破解）

3

第三步：获取权限

获取操作系统权限: 通过数据库执行shell,上传木马

SQL Inject漏洞-常见注入点类型

- 数字型　　user_id=$id
- 字符型　　user_id=　‘$id’
- 搜索型　　text LIKE '%{$_GET['search']}%'"

如何判断注入点的类型？

SQL Inject 漏洞-注入点类型-数字类型判断

数字型注入判断：http://192.168.1.4/pikachu/vul/sqli/sqli_id.php　　post: id=1

正常提交：1

猜测SQL：select 字段名 from 表名 where id =1;

页面回显：id为1的查询结果

#演示：pikachu—SQL-Inject-数字型注入#

SQL Inject 漏洞-注入点类型-数字类型判断

使用and逻辑进行判断：

测试提交：1 and 1=1　猜测SQL：select 字段名 from 表名 where id =1 and 1=1;
测试提交：1 and 1=2　猜测SQL：select 字段名 from 表名 where id =1 and 1=2;

比较页面变化
判断输入是否被执行！

或者提交：’ (单引号)　猜测SQL：select 字段名 from 表名 where id = ‘; //造成SQL语法报错

web安全从入门到放弃

SQL Inject 漏洞-注入点类型-数字类型判断-代码解析

```php
if(isset($_POST['submit']) && $_POST['id']!=null){
    $id=$_POST['id'];//这里没有做任何处理，直接拼到select里面去了
    $query="select username,email from uname where id=$id";
    $result=execute($link, $query);
    if(mysqli_num_rows($result)>=1){//这里如果用==1,会严格一点
        while($data=mysqli_fetch_assoc($result)){
            $username=$data['username'];
            $email=$data['email'];
            $html.="<p class='notice'>hello,{$username} <br />your email is: {$email}</p>";
        }
    }else{
        $html.="<p class='notice'>您输入的user id不存在，请重新输入! </p>";
    }
}
```

#演示：pikachu—SQL-Inject-字符型注入#

SQL Inject 漏洞-注入点类型-字符类型判断

正常提交：1

select 字段 from 表名 where name= ‘kobe’；

提交: kobe and 1=1

Select字段from 表明 where name= ‘kobe and 1=1’；

提交：kobe’ and 1=1--

select字段from 表名 where name= ‘kobe’ and 1=1-- ’；

提交：kobe’ and 1=2--

select字段from 表名 where name= ‘kobe’ and 1=2-- ’；

或者提交：’ (单引号)　猜测SQL：select 字段名 from 表名 where id = ‘; //造成SQL语法报错

SQL Inject 漏洞-注入点类型-字符类型判断-代码解析

```php
if(isset($_GET['submit']) && $_GET['name']!=null){
    $name=$_GET['name'];//这里没有做任何处理，直接拼到select里面去了
    $query="select id,email from uname where username='$name'";//这里的变量是字符型，需要考虑闭合
    $result=execute($link, $query);
    if(mysqli_num_rows($result)>=1){
        while($data=mysqli_fetch_assoc($result)){
            $id=$data['id'];
            $email=$data['email'];
            $html.="<p class='notice'>your uid:{$id} <br />your email is: {$email}</p>";
        }
    }else{

        $html.="<p class='notice'>您输入的username不存在，请重新输入！</p>";
    }
}
```

#演示：pikachu—SQL-Inject-搜索型注入#

SQL Inject 漏洞-注入点类型-搜索型判断

输入：k
select 字段 from 表名 where username like ‘%k%’ ;

输入：k%’ and 1=1 -- 或者 1%’ and ‘%1%’ = ‘%2
select 字段 from 表名 where username like ‘% k%’ and 1=1 -- %’ ;
select 字段 from 表名 where username like ‘% 1%’ and ‘%1%’ = ‘%2 %’ ;

web安全从入门到放弃

SQL Inject 漏洞-注入点类型-搜索型判断-代码解析

```php
if(isset($_GET['submit']) && $_GET['name']!=null){
    $name=$_GET['name'];//这里没有做任何处理，直接拼到select里面去了
    $query="select username,id,email from uname where username like '%$name%'";//这里的变量是模糊匹配，需要考虑闭合
    $result=execute($link, $query);
    if(mysqli_num_rows($result)>=1){
        $html2.="<p class='notice'>用户名中含有{$_GET['name']}的结果如下: <br />";
        while($data=mysqli_fetch_assoc($result)){
            $uname=$data['username'];
            $id=$data['id'];
            $email=$data['email'];
            $html1.="<p class='notice'>username: {$uname}<br />uid:{$id} <br />email is: {$email}</p>";
        }
    }else{

        $html1.="<p class='notice'>0o。..没有搜索到你输入的信息! </p>";
    }
}
```

SQL Inject 漏洞-注入点类型-xxx型判断

不管是啥型

总而言之，就是对SQL中的各种类型的输入进行闭合测试，构造合法SQL，欺骗后台执行！

web安全从入门到放弃

补充：MYSQL小知识：注释符号

因为在SQL注入测试中，需要经常对多余的内容进行消除，以保证SQL语句语法正确，比如上面的# 。

使用注释符号直接对多余内容进行注释是比较有效的方法。

MySQL服务器支持3种注释：

- ✓ 从‘#’字符从行尾。
- ✓ 从‘--’序列到行尾。请注意‘--’(双破折号)注释风格要求第2个破折号后面至少**跟一个空格符**(例如空格、tab、换行符等等)。该语法与标准SQL注释语法稍有不同.
- ✓ 从/*序列到后面的*/序列。结束序列不一定在同一行中，因此该语法允许注释跨越多行。
- ✓ 下面的例子显示了3种风格的注释：

```
mysql> SELECT 1+1;     # This comment continues to the end of line
mysql> SELECT 1+1;     -- This comment continues to the end of line
mysql> SELECT 1 /* this is an in-line comment */ + 1;
mysql> SELECT 1+/*this is amultiple-line comment*/1;
```

注入方式get&post区别

Get方式中使用URL提交注入数据;

Post方式中使用抓包工具修改post数据部分提交注入;

不管是get方式还是post方式，都可能会出现SQL注入漏洞，本质其实是一样的！

web安全从入门到放弃

谢 谢

“听”而不思则罔，思而不“练”则殆

web安全从入门到放弃

通过information_schema拿下数据库

手工测试完整案例演示