BUUCTF [FBCTF2019] Products Manager

原创

Senimo_ 于 2021-01-08 23:49:43 发布 357 收藏 3

分类专栏： BUUCTF WEB Writeup 文章标签： BUUCTF FBCTF2019 ProductsManager writeup CTF

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。

本文链接：https://blog.csdn.net/weixin_44037296/article/details/112385568

版权

BUUCTF WEB Writeup 专栏收录该内容

65 篇文章 9 订阅

订阅专栏

BUUCTF [FBCTF2019] Products Manager

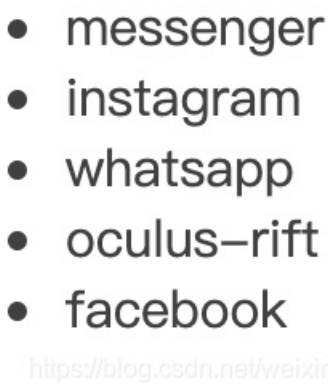
考点：

1. 基于约束的SQL攻击
2. 数据库字符串比较
3. INSERT截断

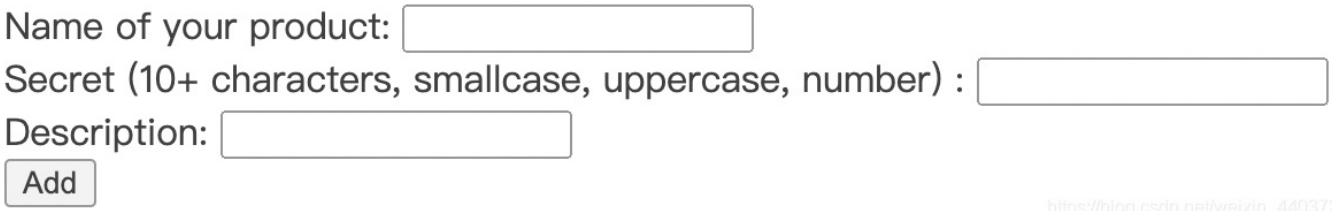
启动环境：

产品管理，其中有三个功能：

- 查看前5的产品

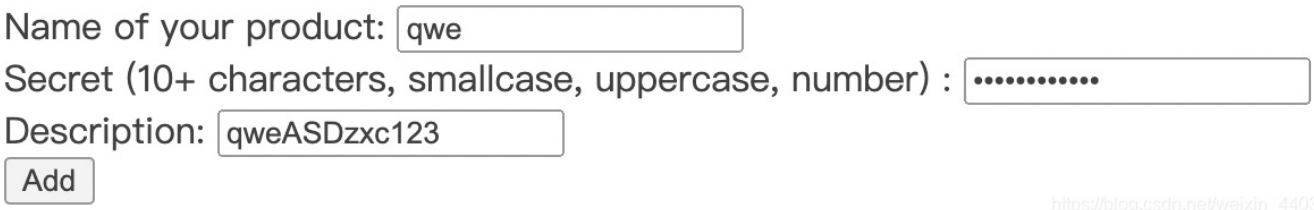


- 添加产品



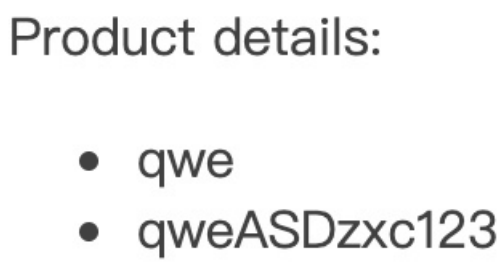
- 查看产品细节

知道了大致功能，尝试正常的业务逻辑，首先添加产品，其`Secret`值需包含`10`位以上的`大小写字母`和`数字`：

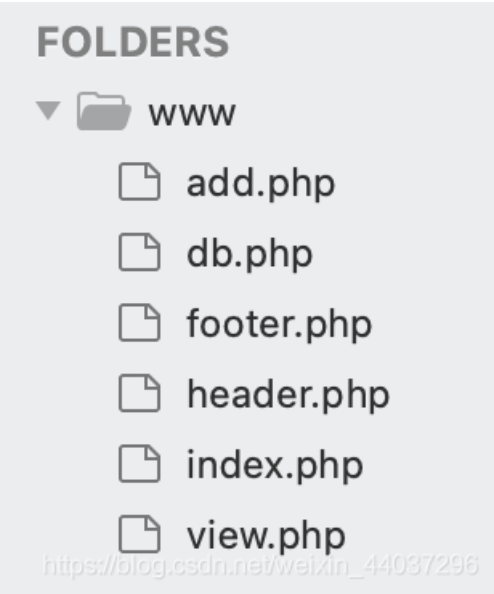


https://blog.csdn.net/weixin_44037296

添加成功，查询产品细节：



分析题目给出的源码：



https://blog.csdn.net/weixin_44037296

其中`footer.php`页面没用，`header.php`页面只是首页的三个跳转链接，`index.php`页面也没有可利用内容。
在`db.php`页面中，查看到如下**SQL**语句：

```
CREATE TABLE products (
  name char(64),
  secret char(64),
  description varchar(250)
);

INSERT INTO products VALUES('facebook', sha256(....), 'FLAG_HERE');
INSERT INTO products VALUES('messenger', sha256(....), ....);
INSERT INTO products VALUES('instagram', sha256(....), ....);
INSERT INTO products VALUES('whatsapp', sha256(....), ....);
INSERT INTO products VALUES('oculus-rift', sha256(....), ....);
```

其中给出了提示，flag在`facebook`中，若想查询产品细节，需要产品的`Secret`值，一开始猜测本题是一道SQL注入题，但未找到可用的注入点，通过查阅大佬wp，得知是道**基于约束的SQL攻击**，参考资料：基于约束的SQL攻击

1. 数据库字符串比较
在数据库对字符串进行比较时，若两字符串长度不一样，则会在较短的字符串末尾填充空格，使两个字符串长度一致。
例：
`str1`和`str`的比较，比较时会在`str`的后面添加一个`空格`以补足长度。

也就是说，对于查询语句：

```
select * from users where username='test'
select * from users where username='test '
```

查询结果是一致的。

2. INSERT截断
在数据插入时，若数据长度超过了预先设定的限制，例如：`name char(64)`时，数据库会对字符串进行截断，只保留限定的长度。

在本题`db.php`页面源码中，查看添加产品和查询产品详情函数：

```php
// 添加产品
function insert_product($name, $secret, $description) {
  global $db;
  $statement = $db->prepare(
    "INSERT INTO products (name, secret, description) VALUES
      (?, ?, ?)"
  );
  check_errors($statement);
  $statement->bind_param("sss", $name, $secret, $description);
  check_errors($statement->execute());
  $statement->close();
}
```

插入语句中`"INSERT INTO products (name, secret, description) VALUES ($name, $secret, $description)"`，并未做任何处理，直接插入数据库。

```php
// 查询产品详情
function get_product($name) {
  global $db;
  $statement = $db->prepare(
    "SELECT name, description FROM products WHERE name = ?"
  );
  check_errors($statement);
  $statement->bind_param("s", $name);
  check_errors($statement->execute());
  $res = $statement->get_result();
  check_errors($res);
  $product = $res->fetch_assoc();
  $statement->close();
  return $product;
}
```

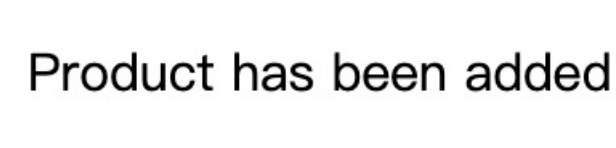
在查询语句中`"SELECT name, description FROM products WHERE name = $name"`，只是对获取的`$name`变量进行了拼接，未进行任何处理。

结合两点，产生了本题的利用点：

- 添加一个`facebook`用户，即在产品名后加大于长度限制的空格，空格后需再跟若干个字符，在添加数据时，使添加的产品名与目标一致。
- 查询时，返回的用户名是目标信息，达到水平越权

构造添加的产品信息：

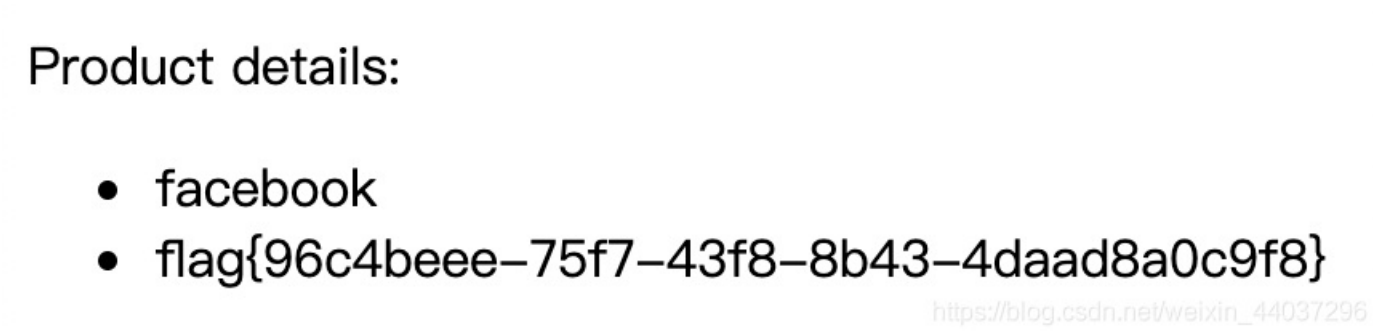
```
Name:facebook                                                    11
Secret:qweASDzxc123
Description:123
```



添加成功，此时再查询刚刚添加的产品详情：

```
Name:facebook
Secret:qweASDzxc123
// 因为在存入数据库时，添加的Name属性长度超过限制被截断
```

查询成功，得到flag：



https://blog.csdn.net/weixin_44037296