WEB2-100 详细解题思路

Author:phithon <root@leavesongs.com>

代码审计正式开始。
首先代码其实是完整的，如果想本地运行需要先 composer 安装所有 php 依赖，并且需要 php5.5.0 版本及以上+linux 环境。Web 目录设置为./front 即可。
源代码中没有 SQL 结构，可访问 http://xdsec-cms-12023458.xdctf.win/xdsec_cms.sql 下载 SQL 初始化文件。（在前台可以找到这个地址）

遍观代码可见是一个基于 Codeigniter 框架的 cms，模板库使用的是 twig，数据库使用 mysql，session 使用文件。
多的不说，直接说我留的漏洞。首先看前台（因为不知道后台地址）：
/xdsec_app/front_app/controllers/Auth.php 110 行 handle_resetpwd 函数，

```
public function handle_resetpwd()
    {
        if(empty($_GET["email"]) || empty($_GET["verify"])) {
            $this->error("Bad request", site_url("auth/forgetpwd"));
        }
        $user = $this->user->get_user(I("get.email"), "email");
        if(I('get.verify') != $user['verify']) {
            $this->error("Your verify code is error", site_url('auth/forgetpwd'));
        }
...
```

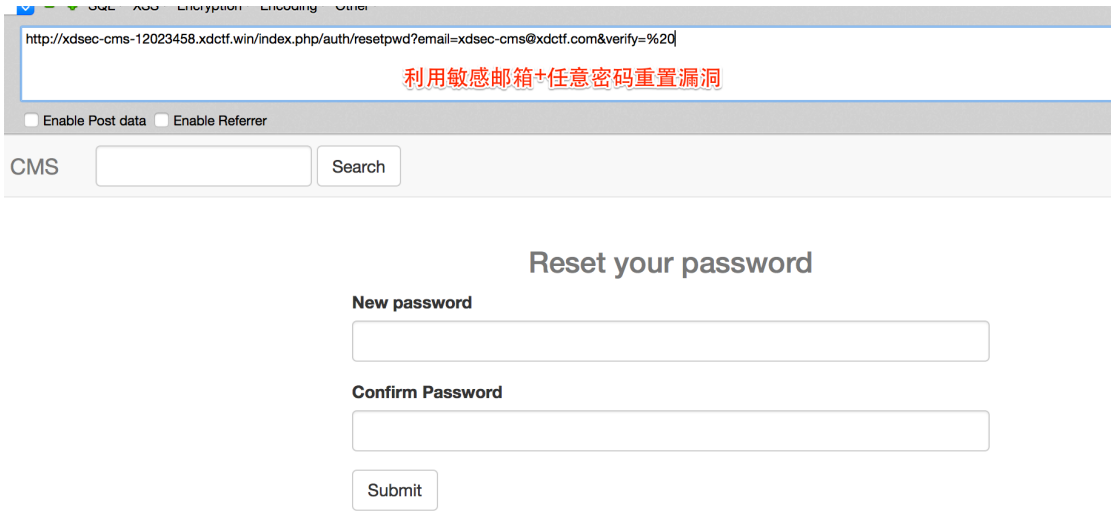
主要是判断传入的$_GET[‘verify’]是否等于数据库中的$user[‘verify’]。而数据库结构中可以看到，verify 默认为 null。
由 Php 弱类型比较（双等号）可以得知，当我们传入$_GET[‘verify’]为空字符串”” 时，””==null，即可绕过这里的判断。

但第一行代码使用 empty($_GET['verify'])检测了是否为空，所以仍然需要绕过。

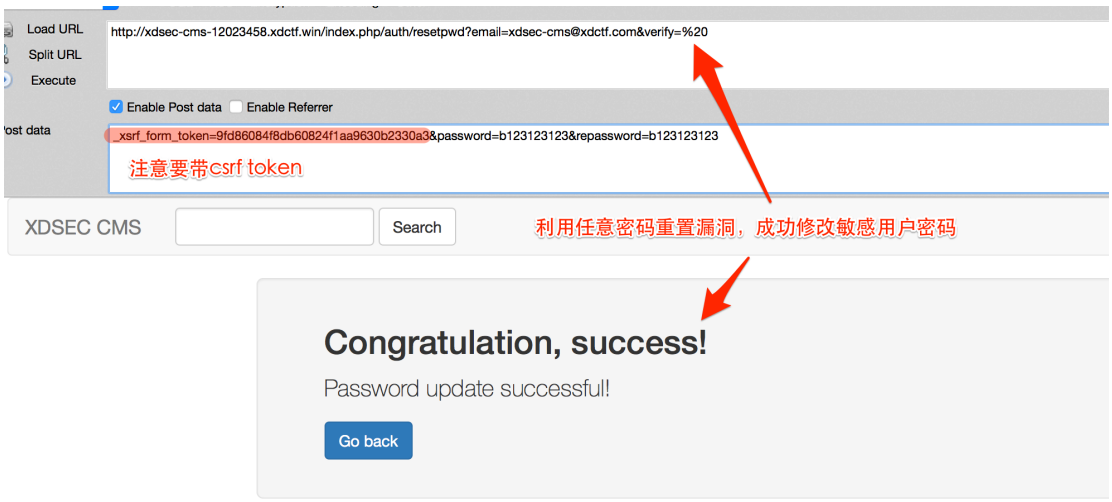
看到获取 GET 变量的 I 函数。I 函数的原型是 ThinkPHP 中的 I 函数，熟悉 ThinkPHP 的人应该知道，I 函数默认是会调用 trim 进行处理的。
查看源码得知，Xdsec-cms 中的 I 函数也会一样处理。所以我们可以通过传入%20 来绕过 empty()的判断，再经过 I 函数处理后得到空字符串，与 null 比较返回 true。即可重置任意用户密码。

那么挖掘到重置漏洞，下一步怎么办？
查看页面 HTML 源文件，可见 meta 处的版权声明，包含一个敏感邮箱：xdsec-cms@xdctf.com

我们直接重置这个邮箱代表的用户：

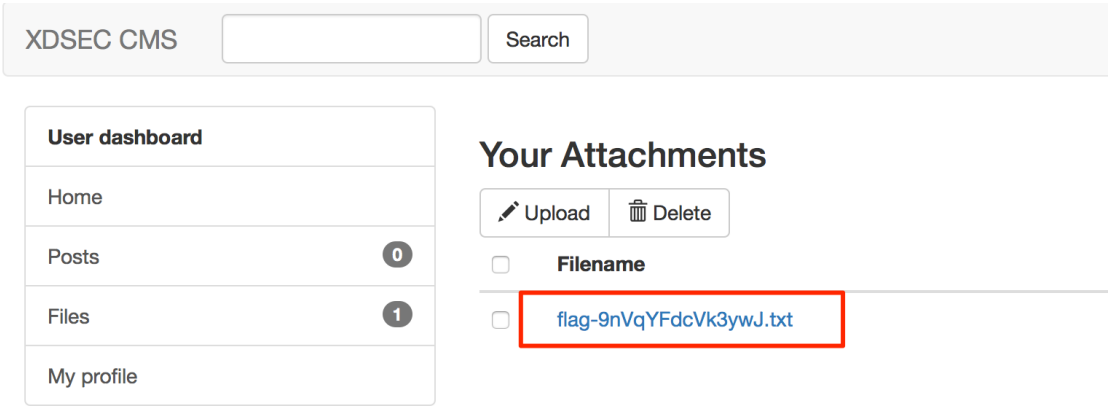


如下图提交数据包，重置成功。（前台开启了 csrf 防御，所以需要带上 token。CI 的 token 是保存在 cookie 中的，所以去看一下就知道了）



利用重置后的账号密码登录 xdsec-cms@xdctf.com。

在用户附件处，发现第 2 枚 flag：



打开：

可见除了 flag 以外告诉了后台地址为/th3r315adm1n.php 。
但没有后台账号密码，所以要进行下一步审计。

这里有同学说不知道管理员邮箱，我想说你即使把我社工个遍、再把网站翻个遍，也就 6、7 个邮箱顶多了，你一个个试，也就试出来了。
渗透时候的信息搜集也很重要，如果连管理员/开发者邮箱都找不着，后续的渗透可能就比较难办了。
相比于这篇文章里提到的类似漏洞，本次的漏洞要简单的多：https://www.leavesongs.com/PENETRATION/findpwd-funny-logic-vul.html，而本文的漏洞是实战中发现的。
所以，偏向实战是我出题的第一考虑因素。