Web Security

实验主题：Web Security

助教：崔吉星，李思帆

主要内容

实验环境搭建

实验内容介绍

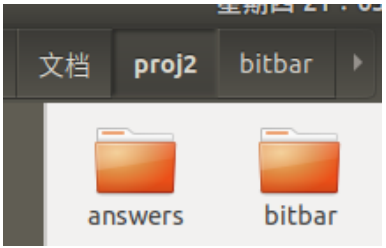
实验提交要求

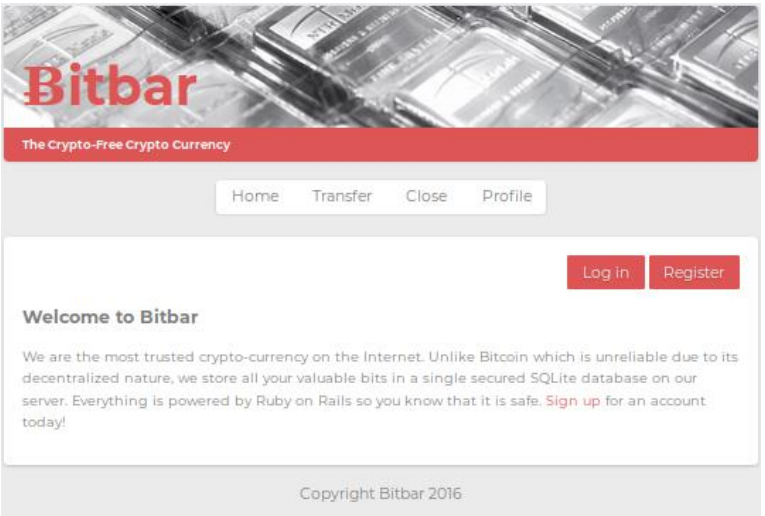
实验环境搭建

Project 2 主要探究对web的攻击，本次试验共有6个部分。

Project 2中攻击的是一个电子货币服务网站--bitbar（使用ruby 2.4和rails 5.0.2实现）。

接下来介绍bitbar网站的搭建。





实验环境搭建

安装Ruby2.5.9和rails 5.0.7

- http://gorails.com/setup/ubuntu/16.04
- 按照上面网址的布置安装Ruby 2.5.9和rail 5.0.7（安装正确的版本非常重要）。可以跳过MySQL和PostgreSQL部分。

下载实验提供的project 2源码

重定位到/bitbar目录下，执行bundle install

开启服务器 （rails server)

以上的步骤执行完后，可以在http://localhost:3000上访问bitbar。如果要关闭服务器，可以在终端上执行Ctrl+C。在攻击的过程中，不允许对网站的源码进行修改

实验内容介绍

Attack 1： Warn-up exercise: Cookie Theft

开始网址

- http://localhost:3000/profile?username=

将提前以user1的身份登录bitbar，然后打开以上的开始网址（user1密码：one)

你的目标是偷取user1的会话cookie并且将cookie发送到

- http://localhost:3000/steal_cookie?cookie=...cookie_data_here...

你可以在以下网址上查看最近被偷取的cookie

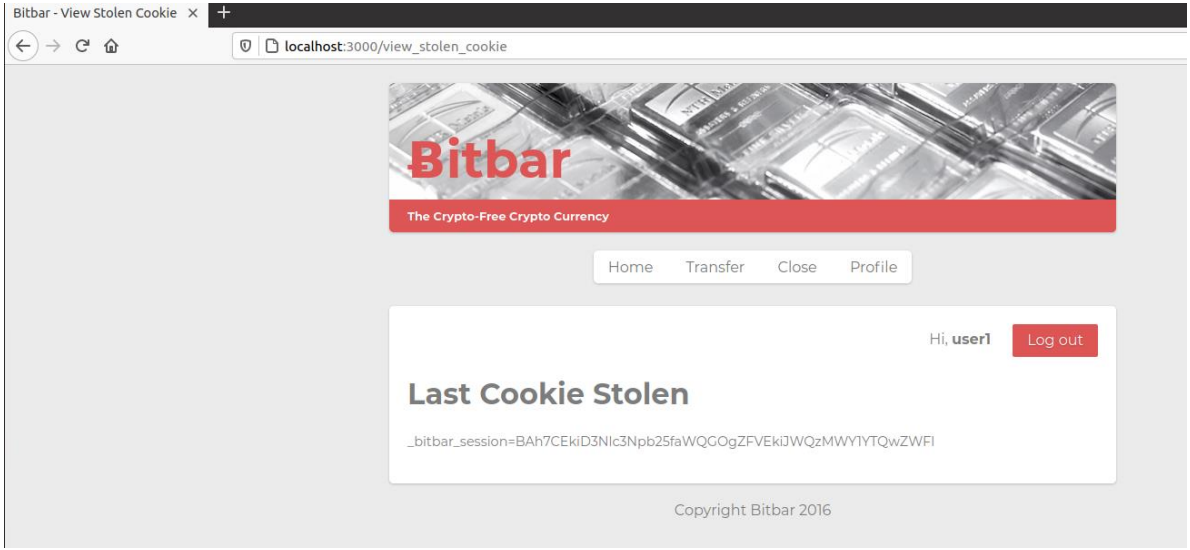
- http://localhost:3000/view_stolen_cookie

请将你的答案写在warmup.txt中

- 提示：XSS漏洞 <script>alert(/xss/)</script>

实验内容介绍

Attack 1： Warn-up exercise: Cookie Theft



实验内容介绍

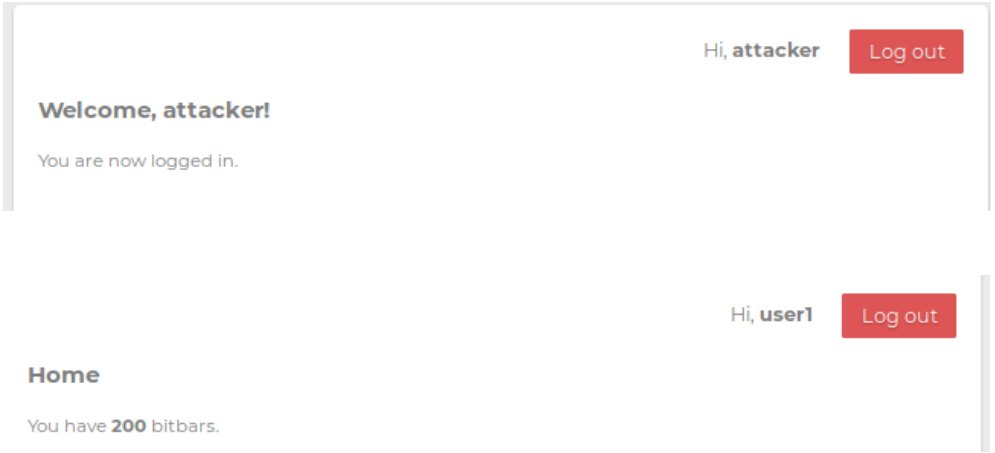
Attack 2: Session hijacking with Cookies

在本次试验中，你将会获得attacker的身份：用户名attacker，密码attacker。你的目的是伪装成用户user1登录系统。

你的答案应该是一个脚本。当这个脚本在JavaScript控制台中执行时，bitbar将误认为你是以user1。请将这个脚本写到a.sh中。

提示：网站是如何保存会话的？网站是如何验证用户当前是否登录？网站是如何验证cookie的真实性的？

实验内容介绍
Attack 2: Session hijacking with Cookies



实验内容介绍

Attack 3: Cross-site Request Forgery

你的答案是一个名字为b.html的html文件。

将提前使用user1的身份登录到bitbar，然后打开b.html。

打开b.html后，10个bitbar将从user1的账户转到attacker的账户，当转账结束时，页面重定向到www.baidu.com。

你可以在http://localhost:3000/view_users 查看用户列表以及每个用户拥有的bitbar

在攻击的过程中，浏览器的网址中不能出现localhost:3000

提示：构造一个自动执行的html文件，通过XMLHttpRequest，构造发送的数据包，在登陆过User1的浏览器中执行。

实验内容介绍

Attack 3: Cross-site Request Forgery

实验内容介绍
Attack 4: Cross-site request forgery with user assistance

你的答案是一个或者两个html页面，命名为bp.html，bp2.html(可选)

在打开bp.html前，系统中已经登录了user1

接下来将在bp.html页面进行交互，因此bp.html的设置要合理。也就是说，如果在页面上有一个表格或者有一个按钮，并且在页面上有一些提示要求用户进行一些操作，引导用户依照这些提示执行。

在用户与bp.html页面进行交互后，10 bitbars将会从user1账户转到attacker的账户。当这个转账操作执行完成后，页面将重定向到www.baidu.com

你的攻击必须要在与用户互动的前提下执行（与Attack 3不同）。特别的要注意的是，你的攻击要针对的网址是http://localhost:3000/super_secure_transfer或者http://localhost:3000/super_secure_post_transfer。这两个网址做了一些CSRF攻击的防护。在攻击的过程中，你不能直接与http://localhost:3000/transfer或者http://localhost:3000/post_transfer进行交互。

在你的攻击过程中，需要隐藏你的页面正从http://localhost:3000上下载内容的事实。

实验内容介绍

Attack 4: Cross-site request forgery with user assistance

Active user accounts

username	bitbars
user1	180
user2	200
user3	100000
attacker	20

实验内容介绍

Attack 5: Little Bobby Tables (aka SQL Injection)

你的答案是一个恶意的用户名。这个恶意的用户名允许你删除一个你不具有访问权限账户。

评分员将使用你提供的恶意用户名新建一个账户。并在“close”页面上确认删除该账户

作为结果，新建的账户以及user3的账户将会被删除。其他的账户不变

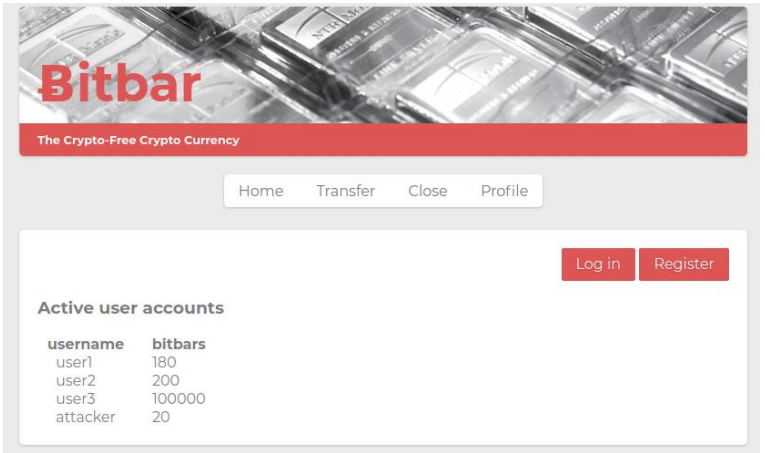
你可以在http://localhost:3000/view_users页面上查看所用的用户

如果数据库在测试攻击的过程中被破坏了，你可以停止Rails然后使用rake db:reset命令使数据库复原。

将你的最终答案写在c.txt中

- 提示：查看服务器删除用户时的操作。SQL注入；WHERE子句

实验内容介绍
Attack 5: Little Bobby Tables (aka SQL Injection)



Active user accounts

username	bitbars
user1	180
user2	200
attacker	20

实验内容介绍
Attack 6: Profile Worm

你的答案是一个用户的profile（简况）。当其他用户阅读这个profile时，1个bitbar将会从当前账户转到attacker的账户，并且将当前用户的profile修改成该profile。

因此，如果attacker将他的profile修改成你的答案，以下情况会发生：

- 如果user1浏览了attacker的profile，那么1 bitbar将从user1的账户转到attacker的账户，user1的profile修改成你答案中的profile
- 之后，如果user2浏览了user1的profile，那么1 bitbar将从user2的账户转到attacker的账户，user2的profile也被替换成你答案中profile

因此，你的profile worm将会很快扩散到全部的用户账户中

将你的恶意的profile写在d.txt中

实验内容介绍
Attack 6: Profile Worm

评分过程：将你提供的恶意profile复制到attacker的profile上。然后，使用多个账户浏览attacker的profile。检查是否正常进行转账以及profile的复制

转账和profile复制的过程应该控制在15s之内。

在转账和profile的复制过程中，浏览器的地址栏需要始终停留在http://localhost:3000/profile?username=x ，其中x是profile被浏览的用户名。

提示：MySpace vulnerability，存储型XSS

实验提交要求

提交时间：15周周日，5月29号，24：00 PM前

提交方式：

- **命名：学号-姓名-project2.rar**
- **邮箱：jixingstephen@163.com@qq.com**

提交作业包括：

- **攻击答案（answers文件夹）**
- **实验报告（学号-姓名-project2.docx）**

实验报告要求：

- **Attack 1-6漏洞分析**
- **Attack 1-6攻击原理**