



vuln03-CSRF漏洞

课程大纲

- 1、什么是CSRF漏洞?
- 2、CSRF案例分析
- 3、CSRF漏洞挖掘
- 4、CSRF漏洞防御



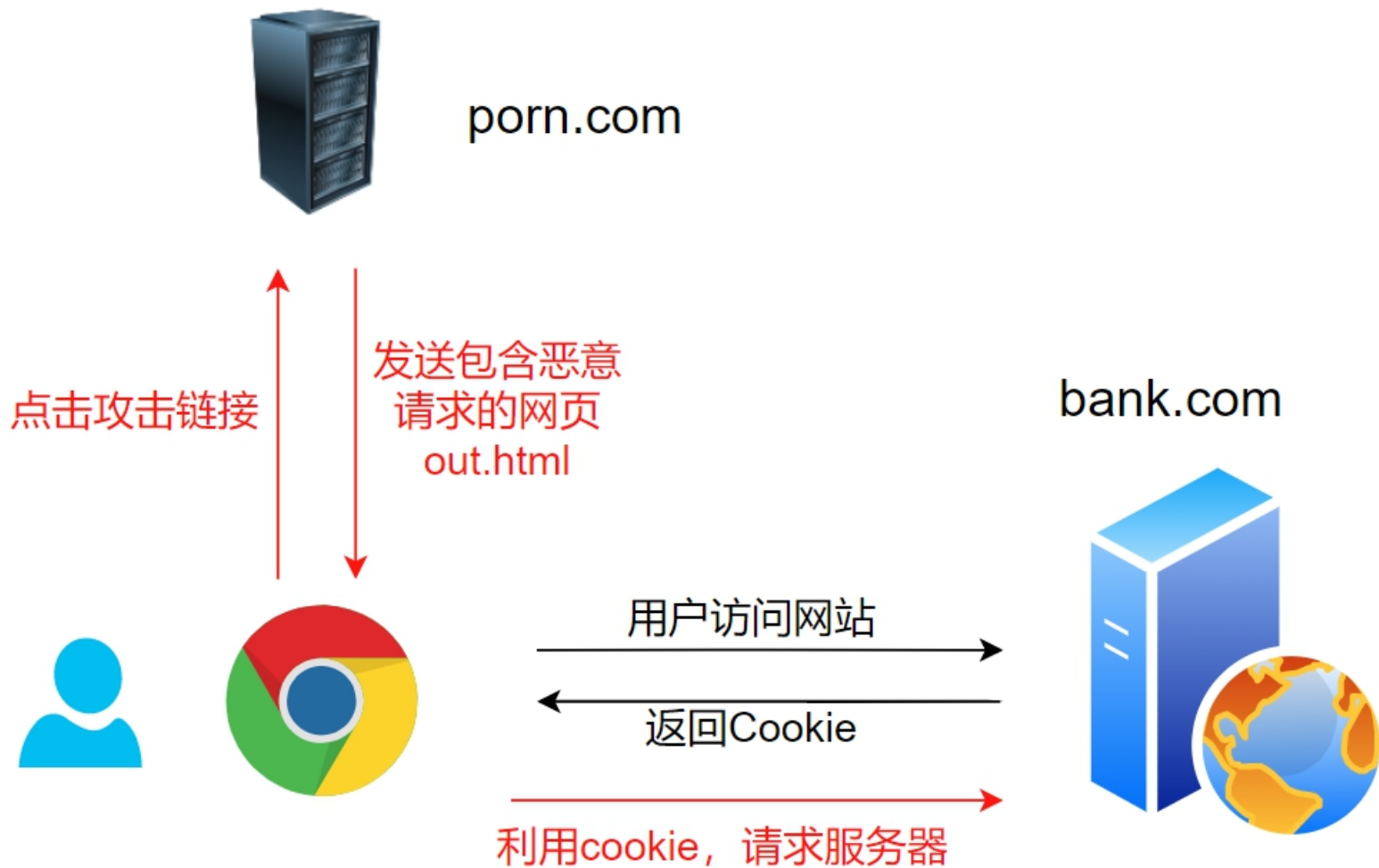
01

什么是CSRF漏洞

⋮ CSRF(XSRF)

Cross-Site Request Forgery
跨站请求伪造

CSRF实现流程





02

CSRF案例分析

CSRF典型案例

-  Gmail CSRF漏洞 (设置邮件转发)
-  微博 Weibo CSRF漏洞 (自动关注账号)

CSRF漏洞危害

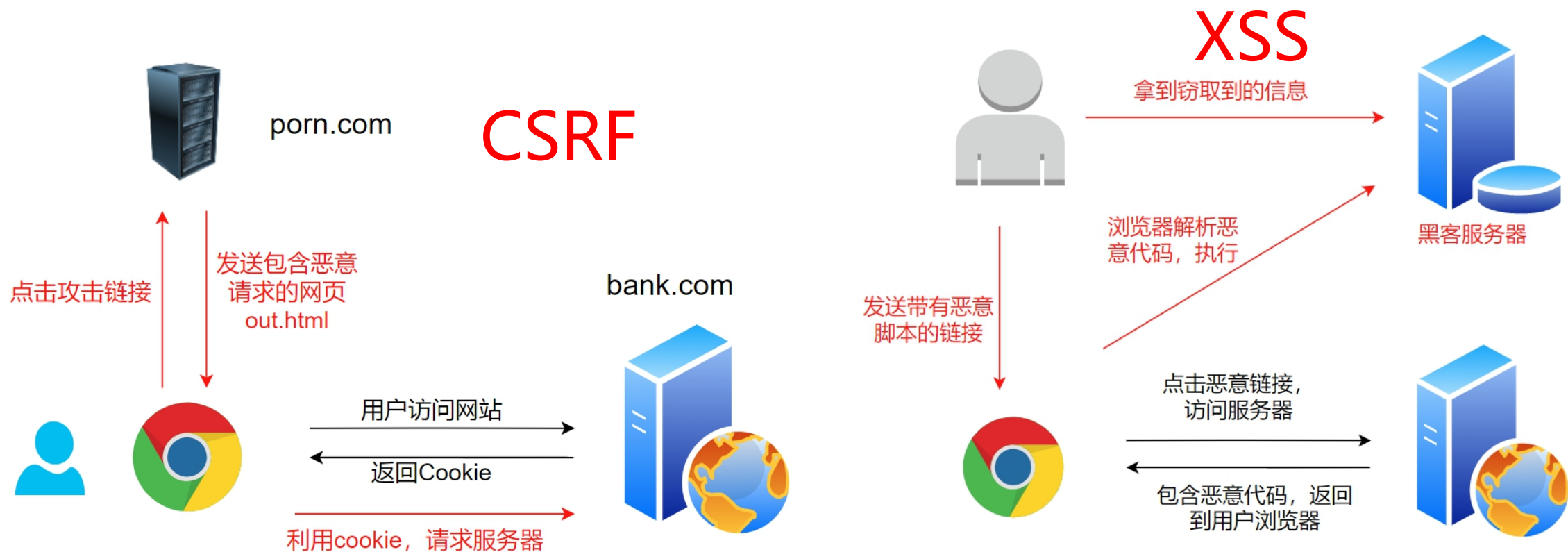
例如：

- 修改账户信息
- 利用管理员账号，上传木马文件
- 传播蠕虫病毒（点击、扩散、点击.....）
- 和其他攻击手段配合，实现攻击，比如XSS、SQL注入

⋮ XSS漏洞危害

- 获取cookie，实现冒充身份的后续操作
- 刷点击
- 弹广告
- 传播蠕虫病毒

CSRF与XSS区别



⋮ CSRF payload1

1) 通过图片的img src属性, 自动加载, 发起GET请求

```

```

⋮ CSRF payload2

2) 构建一个超链接, 用户点击以后, 发起GET请求

```
<a  
href="http://superbank.com/transfer.php?amount=1000&  
to=jiangang" target="_blank">  
  小姐姐在线视频聊天! !  
</a>
```

⋮ CSRF payload3

3) 构建一个隐藏表单，用户访问，自动提交，发起POST请求

```
<form action="http://superbank.com/withdraw" method=POST>  
  <input type="hidden" name="account" value="xiaoming" />  
  <input type="hidden" name="amount" value="1000" />  
  <input type="hidden" name="to" value="jiangang" />  
</form>  
<script> document.forms[0].submit(); </script>
```



03

CSRF漏洞挖掘

CSRF漏洞挖掘

- 1、道：怎么确定一个接口地址有没有CSRF漏洞呢？
- 2、术：具体怎么操作？
- 3、器：有没有工具可以使用？

检测工具

Burp Suite

CSRF Tester

<https://github.com/s0md3v/Bolt>

各种云产品

⋮ Burp 代理抓包





04

CSRF漏洞防御

： 防御思路

a、我们能不能区分一个请求是来自于自己的前端页面，还是第三方的网站？

b、怎么让自己的前端页面和伪造的请求变得不一样呢？

HTTP Request Header

Referer: `https://www.baidu.com/s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=5a7&rsv_t=89efDHGeD8cmcPWIdmwOyCTnQ0%2F1LS7ydq1hJfuG6FaKwU%2BZOS7gu0vX3rE&rsv_sug3=4&rsv_sug1=4&rsv_sug7=101&rsv_sug2=0&rsv_btype=i&prefixsug=666&rsp=5&in`

Referer: 引用页; 引荐; 来源页面

作用: 跟踪来源, 比如访问统计、广告效果

⋮ DVAW-Medium

检查REFERER

(referer里面是否包含了主机名 (IP或域名))

```
if( strpos( $_SERVER[ 'HTTP_REFERER' ] ,$_SERVER[ 'SERVER_NAME' ]) !== false ){
```

⋮ referer的不足

- 1、可以任意修改
- 2、可以为空

： 第二种方案

在请求中加入一些随机字段（第三方不知道也猜不出来），让第三方网站无法伪造请求

⋮ CSRF Token step1/5

1、用户使用用户名密码登录，服务端下发一个随机的 token 字段给客户端，并且服务端把这个字段保存在 session 中。

```
session_start();
if (empty($_SESSION['token'])) {
    $_SESSION['token'] = bin2hex(random_bytes(32));
}
$token = $_SESSION['token'];
```


⋮ CSRF Token step2/5

- 2、客户端把这个token保存起来，放到隐藏字段。
- 3、用户在登陆状态下，在之后访问的时候，都要携带这个token字段。

⋮ CSRF Token step4/5

4、服务端从session中拿出token值进行对比，如果一致，说明请求合法。

```
if (!empty($_POST['token'])) {  
    if (hash_equals($_SESSION['token'], $_POST['token'])) {  
        // 执行业务逻辑  
    } else {  
        // ...  
    }  
}
```

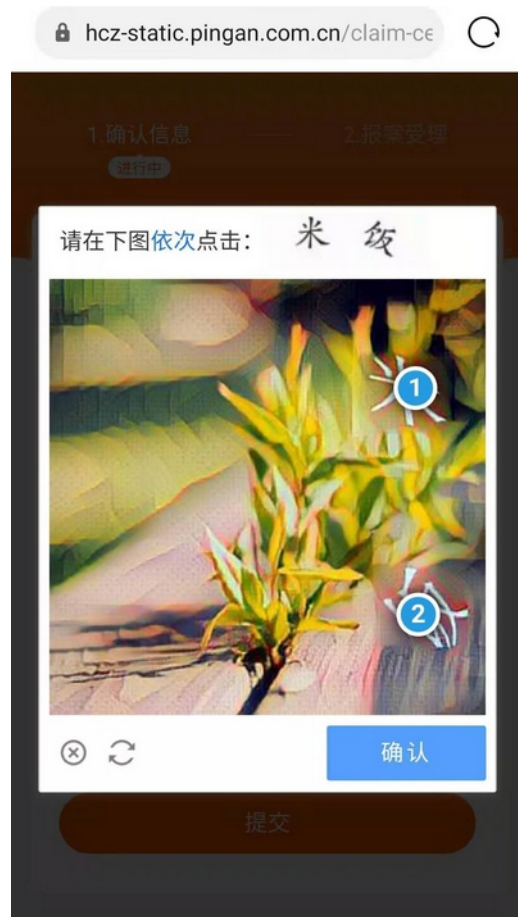
⋮ CSRF Token step5/5

5、用户退出，session销毁，token失效。

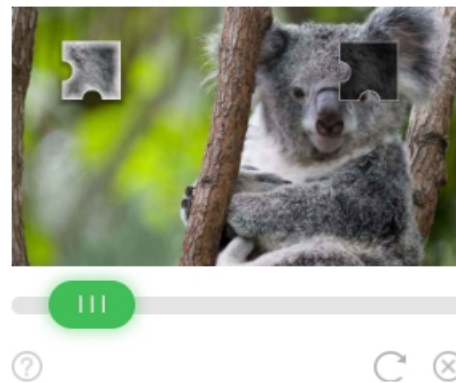
问题

有没有更加安全的办法？

验证码



拖动下方滑块完成拼图



二次验证

✅ 验证码发送成功

手机150*****07

请输入验证码

发送验证码

确定

二维码登录

⚠️ 你的帐号存在安全隐患，为保证帐号安全，请在常用设备上使用 豆瓣App扫码登录



请打开豆瓣 App 扫一扫 或 短信登录验证

人脸识别认证

0

非法攻击

请根据提示完成人脸识别

退出 重新检测



请平视屏幕
距离30-60厘米

版本号v2.0.9

reCAPTCHA (IP验证)

Confirm Humanity

Before we subscribe you, we need to confirm you are a human.



进行人机身份验证



Captcha failed. Please try again.

Select all images with
a fire hydrant
Click verify once there are none left.

🔄 🎧 ⓘ

VERIFY

个人用户建议

- 1、不要访问不安全的网站
- 2、不要随意点开别人发给你的链接





Thank you for watching