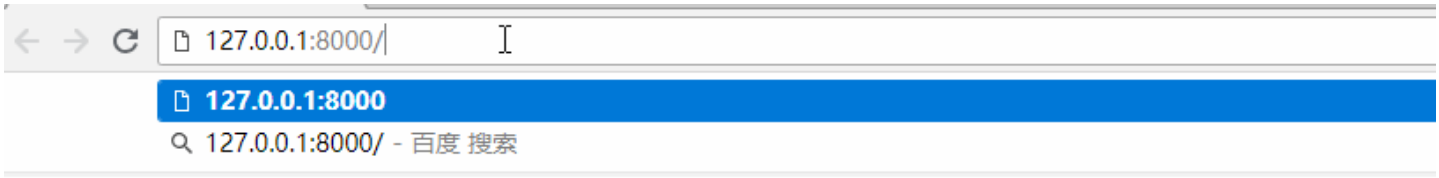
Django URL跳转漏洞分析（CVE-2018-14574 ）

xman21 / 2018-11-17 08:30:00 / 浏览数 4527 技术文章 技术文章 顶(0) 踩(0)

零、基本信息

CVE-ID：CVE-2018-14574
漏洞类型：URL跳转
影响范围：1.11.0 <= version < 1.11.15 和 2.0.0 <= version < 2.0.8

一、环境复现



二、漏洞分析

当setting中配置了django.middleware.common.CommonMiddleware且APPEND_SLASH为True时漏洞就会触发，而这两个配置时默认存在的，而且APPEND_SLASH不

```
- Forbid access to User-Agents in settings.DISALLOWED_USER_AGENTS

- URL rewriting: Based on the APPEND_SLASH and PREPEND_WWW settings,
append missing slashes and/or prepends missing "www."s.

 - If APPEND_SLASH is set and the initial URL doesn't end with a
   slash, and it is not found in urlpatterns, form a new URL by
   appending a slash at the end. If this new URL is found in
   urlpatterns, return an HTTP redirect to this new URL; otherwise
   process the initial URL as usual.

This behavior can be customized by subclassing CommonMiddleware and
overriding the response_redirect_class attribute.

- ETags: If the USE_ETAGS setting is set, ETags will be calculated from
the entire page content and Not Modified responses will be returned
appropriately. USE_ETAGS is deprecated in favor of
ConditionalGetMiddleware.
```

而漏洞就与URL
rewriting有关：如果设置了APPEND_SLASH=True并且初始URL没有以斜杠结尾，并且在urlpatterns中找不到它，则通过在末尾附加斜杠来形成新的URL。如果在urlpatte

但是当发起当发起类似这样的请求http://127.0.0.1:8000//baidu.com
程序就会进行设定的跳转，首先会执行process_request()函数，在61行进入get_full_path_with_slash()函数

```
59          # Check if a slash should be appended
60          if self.should_redirect_with_slash(request):
61              path = self.get_full_path_with_slash(request)
62          else:
63              path = request.get_full_path()
```

这个函数的作用就是get_full_path()函数给path末尾加上斜杠

```
83      def get_full_path_with_slash(self, request):
84          """
85          Return the full path of the request with a trailing slash appended.
86
87          Raise a RuntimeError if settings.DEBUG is True and request.method is
88          POST, PUT, or PATCH.
89          """
90          new_path = request.get_full_path(force_append_slash=True)
91          if settings.DEBUG and request.method in ('POST', 'PUT', 'PATCH'):
92              raise RuntimeError(
93                  "You called this URL via %(method)s, but the
94                  "in a slash and you have APPEND_SLASH set. Django
95                  "redirect to the slash URL while maintaining %(method)s data. "
96                  "Change your form to point to %(url)s (note the trailing "
97                  "slash), or set APPEND_SLASH=False in your Django settings." % {
98                      'method': request.method,
99                      'url': request.get_host() + new_path,
100                 }
101             )
102         return new_path
```

variable "method"

返回的new_path就是//baidu.com/ ，然后在68行进入HttpResponseRedirectBase这个类，它是HTTP跳转的一个基类

```
65            # Return a redirect if necessary
66            if redirect_url or path != request.get_full_path():
67                redirect_url += path
68                return self.response_redirect_class(redirect_url)
```

虽然类的初始化函数里（409行）有对协议的检查，但是scheme根本就不存在，所以会跳过这个判断。

```
class HttpResponseRedirectBase(HttpResponse):
    allowed_schemes = ['http', 'https', 'ftp']  allowed_schemes: <class 'list'>: ['http', 'https', 'ftp

    def __init__(self, redirect_to, *args, **kwargs):  self: Unable to get repr for <class 'django.http
        super().__init__(*args, **kwargs)
        self['Location'] = iri_to_uri(redirect_to)
        parsed = urlparse(str(redirect_to))
        if parsed.scheme and parsed.scheme not in self.allowed_schemes:
            raise DisallowedRedirect("Unsafe redirect to URL with protocol '%s'" % parsed.scheme)
```

在往后就是正常的301跳转

Raw | Params | Headers | Hex

```
GET //baidu.com HTTP/1.1
Host: 127.0.0.1:8000
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/67.0.3396.99 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie:
csrftoken=zUMOcMuQEuuFRE6UO2i5ITqqXDWOzUWYisZlNkKLJySkSOUN8riF6NVaQH9ekwlL;
_ga=GA1.1.1158728480.1531893501;
request_token=GjuZhCRjFHTmyBmbtOTXVXfqveLLzmbTqzfxlRKagYjezVuEltrjUZfgPkSxgOHY
Connection: close
```

Raw | Headers | Hex

```
HTTP/1.1 301 Moved Permanently
Date: Thu, 15 Nov 2018 13:09:54 GMT
Server: WSGIServer/0.2 CPython/3.6.3
Content-Type: text/html; charset=utf-8
Location: //baidu.com/
Content-Length: 0
```

双斜线是为了告诉浏览器这是绝对路径，否则就会跳转到http://127.0.0.1:8000/baidu.com/ 而不是baidu了。

三、补丁分析

修补方法就是加了一个编码函数，

```
438 def escape_leading_slashes(url):
439     """
440     If redirecting to an absolute path (two leading slashes), a slash must be
441     escaped to prevent browsers from handling the path as schemaless and
442     redirecting to another host.
443     """
444     if url.startswith('//'):
445         url = '/%2F{}'.format(url[2:])
446     return url
447
```

对第二个/编码，这样就构不成绝对路径了

Request

Raw Params Headers Hex

```
GET //baidu.com HTTP/1.1
Host: 127.0.0.1:8000
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/67.0.3396.99 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie:
csrftoken=zUMOcMuQEuuFRE6UO2i5ITqqXDWOzUWYisZlNkKLJySkSOUN8riF6NVaQH9ekwlL;
_ga=GA1.1.1158728480.1531893501;
request_token=GjuZhCRjFHTmyBmbtOTXVXfqveLLzmbTqzfxlRKagYjezVuEltrjUZfgPkSxgOHY
Connection: close
```

Response

Raw Headers Hex

```
HTTP/1.1 301 Moved Permanently
Date: Thu, 15 Nov 2018 14:16:23 GMT
Server: WSGIServer/0.2 CPython/3.6.3
Content-Type: text/html; charset=utf-8
Location: /%2Fbaidu.com/
Content-Length: 0
```

四、参考文献

https://nvd.nist.gov/vuln/detail?vulnId=2018-14574
https://github.com/django/django/commit/6fffc3c6d420e44f4029d5643f38d00a39b08525#diff-1f8be0eae49a1bf37d52829eaeda6a4eR14

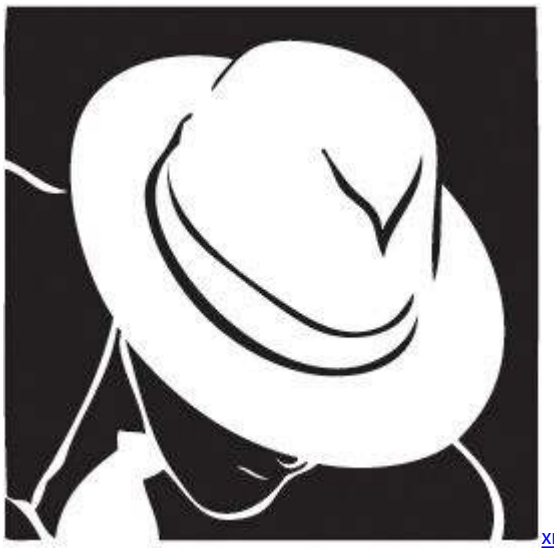
上一篇：Hook深度研究:监视WOW64程... 下一篇：Hook深度研究:监视WOW64程...

1. 3 条回复

Lilyan 2018-11-26 16:40:59

有个小疑问 为什么我使用 django 2.0.7 无法复现这个漏洞？
想问一下 使用的那个及 .py 文件里面的规则是怎么写的？？

0 回复Ta



xman21 2018-12-07 14:19:51

@Lilyan 具体是什么错误或什么响应呢

0 回复Ta


p****@qq.com 2019-01-24 14:35:16

零 亮了。。。。。

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