登录

An Accidental SSRF Honeypot in Google Calendar

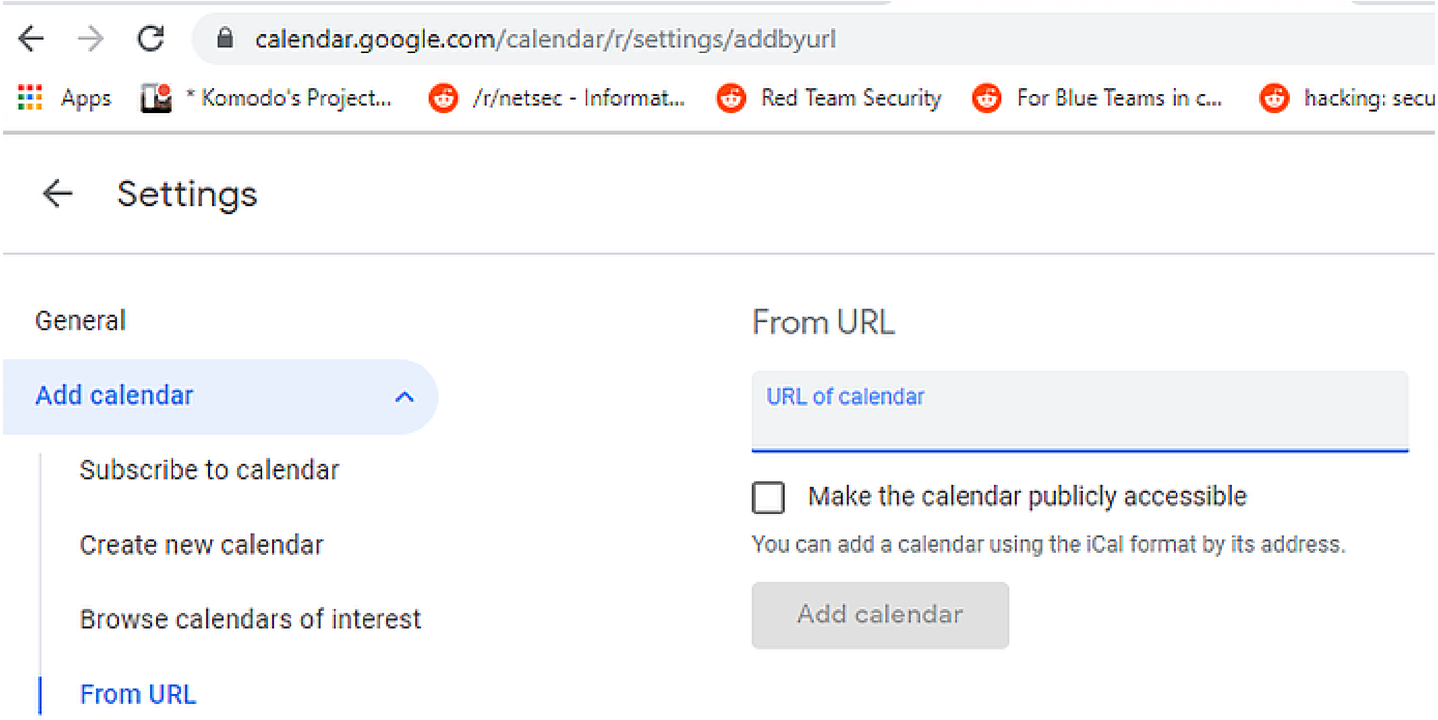
backcover7 / 2019-09-17 09:10:36 / 浏览数 3384 渗透测试 渗透测试 顶(0) 踩(0)

本文是翻译文章，原文链接为：https://www.komodosec.com/post/an-accidental-ssrf-honeypot-in-google-calendar

这是一个我和Google工程师都误以为是Google日历中的SSRF漏洞的故事，最后实际上是因为一些缓存机制的缘故。虽然结果不太好，但是我和Google安全团队的交流很好

从URL导入日历

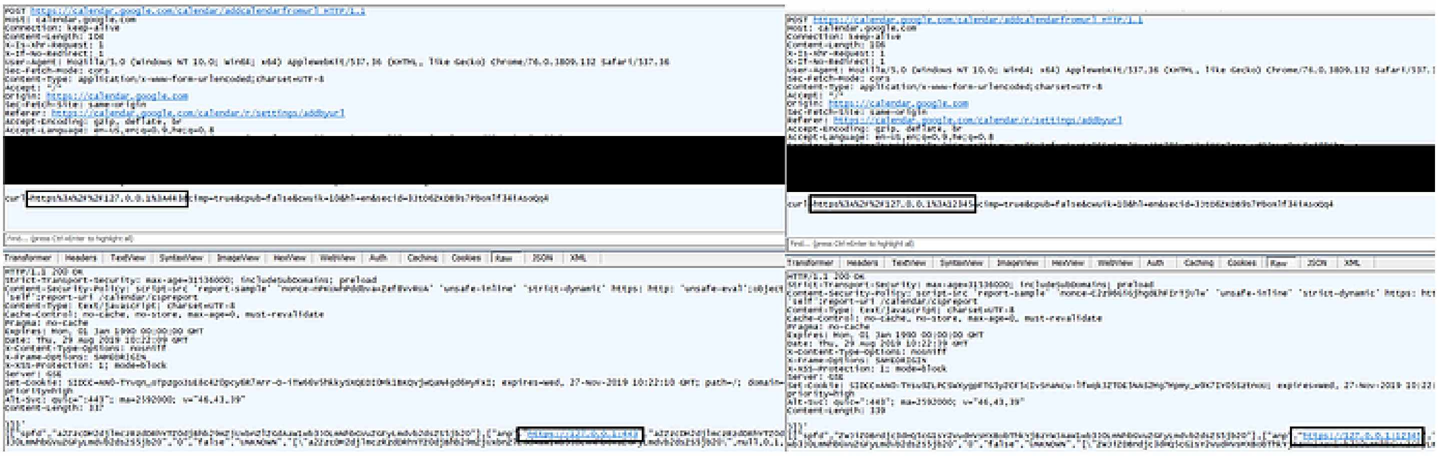
Google日历有很多不错的功能，其中有一个正如它名字一样，可以通过URL添加一个远程的日历。



Google服务器可以从远程日历上把事件添加到你自己的日历中。或者也可以说是我们使用了Google服务器的HTTP请求。所以访问外部URL是Google服务器的内部操作。但

第一次尝试——貌似安全

第一次测试并没有得到有趣的记过，因为当我们尝试去访问外部URL时，服务器报错了（不可达或错误格式等错误），当我们输入一个内部地址我们得到了同样的错误。AD



```
POST https://calendar.google.com/calendar/addcalendarfromurl HTTP/1.1
Host: calendar.google.com
Connection: keep-alive
Content-Length: 108
X-Is-Xhr-Request: 1
X-If-No-Redirect: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.132 Safari/537.36
Sec-Fetch-Mode: cors
Content-Type: application/x-www-form-urlencoded;charset=UTF-8
Accept: */*
Origin: https://calendar.google.com
Sec-Fetch-Site: same-origin
Referer: https://calendar.google.com/calendar/r/settings/addbyurl
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,he;q=0.8
[redacted]
curl=https%3A%2F%2F127.0.0.1%3A443&cimp=true&cpub=false&cwuik=10&hl=en&secid=[illegible]
```

Find... (press Ctrl+Enter to highlight all)
Transformer | Headers | TextView | SyntaxView | ImageView | HexView | WebView | Auth | Caching | Cookies | Raw | JSON | XML

```
HTTP/1.1 200 OK
Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
Content-Security-Policy: script-src 'report-sample' 'nonce-[illegible]' 'unsafe-inline' 'strict-dynamic' https: http: 'unsafe-eval';object
'self';report-uri /calendar/cspreport
Content-Type: text/javascript; charset=UTF-8
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: Mon, 01 Jan 1990 00:00:00 GMT
Date: Thu, 29 Aug 2019 10:22:09 GMT
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
Server: GSE
Set-Cookie: SIDCC=[illegible]; expires=Wed, 27-Nov-2019 10:22:10 GMT; path=/; domain=
priority=high
Alt-Svc: quic=":443"; ma=2592000; v="46,43,39"
Content-Length: 337

)]}'
[["spfd","[illegible]"],["er","https://127.0.0.1:443","[illegible]
[illegible]","0","false","UNKNOWN","[\"[illegible]",null,0,1,
```

```
POST https://calendar.google.com/calendar/addcalendarfromurl HTTP/1.1
Host: calendar.google.com
Connection: keep-alive
Content-Length: 106
X-Is-Xhr-Request: 1
X-If-No-Redirect: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.132 Safari/537.3
Sec-Fetch-Mode: cors
Content-Type: application/x-www-form-urlencoded;charset=UTF-8
Accept: */*
Origin: https://calendar.google.com
Sec-Fetch-Site: same-origin
Referer: https://calendar.google.com/calendar/r/settings/addbyurl
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,he;q=0.8
[redacted]
curl=https%3A%2F%2F127.0.0.1%3A12345&cimp=true&cpub=false&cwuik=10&hl=en&secid=[illegible]
```

Find... (press Ctrl+Enter to highlight all)
Transformer | Headers | TextView | SyntaxView | ImageView | HexView | WebView | Auth | Caching | Cookies | Raw | JSON | XML

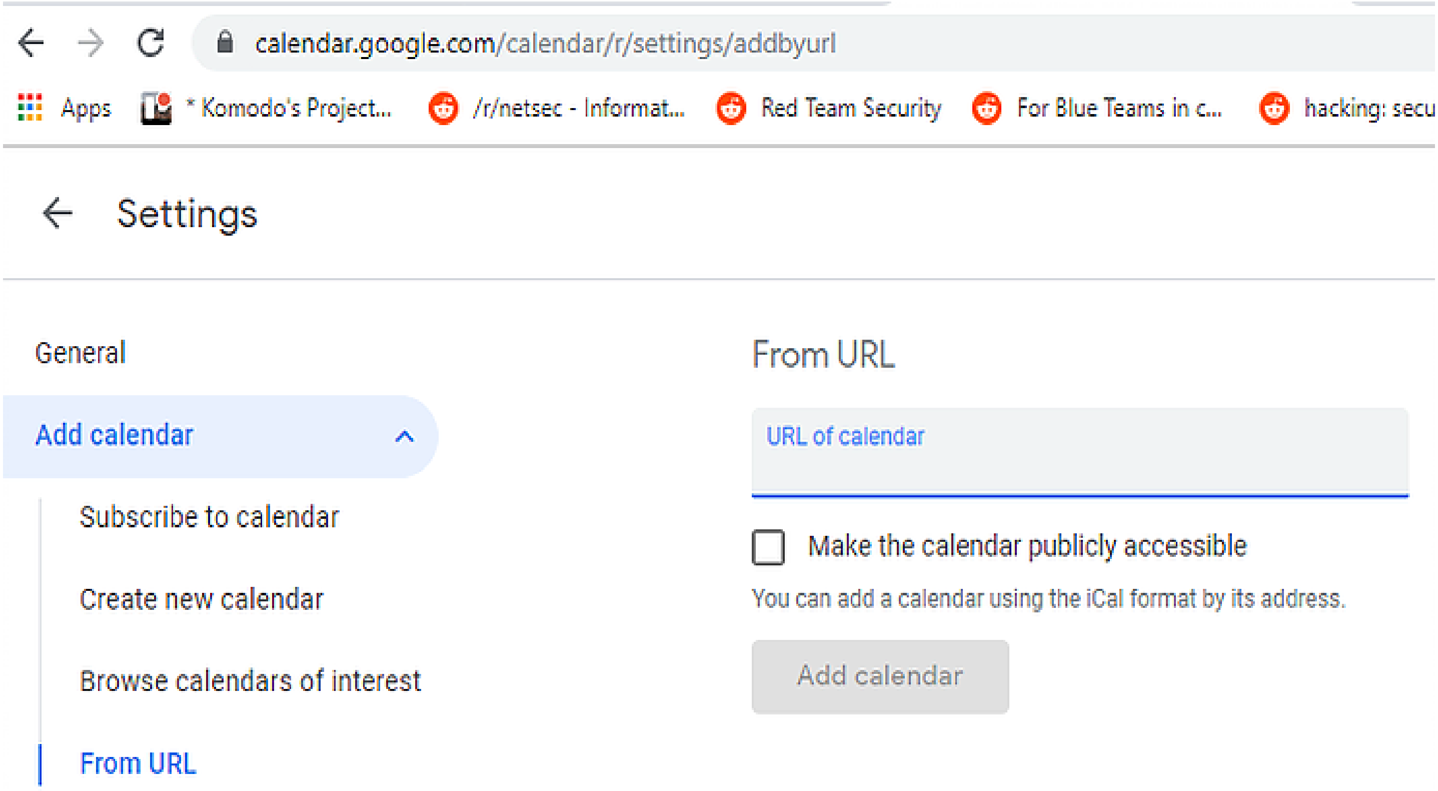
```
HTTP/1.1 200 OK
Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
Content-Security-Policy: script-src 'report-sample' 'nonce-[illegible]' 'unsafe-inline' 'strict-dynamic' https: htt
'self';report-uri /calendar/cspreport
Content-Type: text/javascript; charset=UTF-8
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: Mon, 01 Jan 1990 00:00:00 GMT
Date: Thu, 29 Aug 2019 10:22:39 GMT
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
Server: GSE
Set-Cookie: SIDCC=[illegible]; expires=Wed, 27-Nov-2019 10:22:
priority=high
Alt-Svc: quic=":443"; ma=2592000; v="46,43,39"
Content-Length: 339

)]}'
[["spfd","[illegible]"],["er","https://127.0.0.1:12345",
[illegible]","0","false","UNKNOWN","[\"[illegible]
```

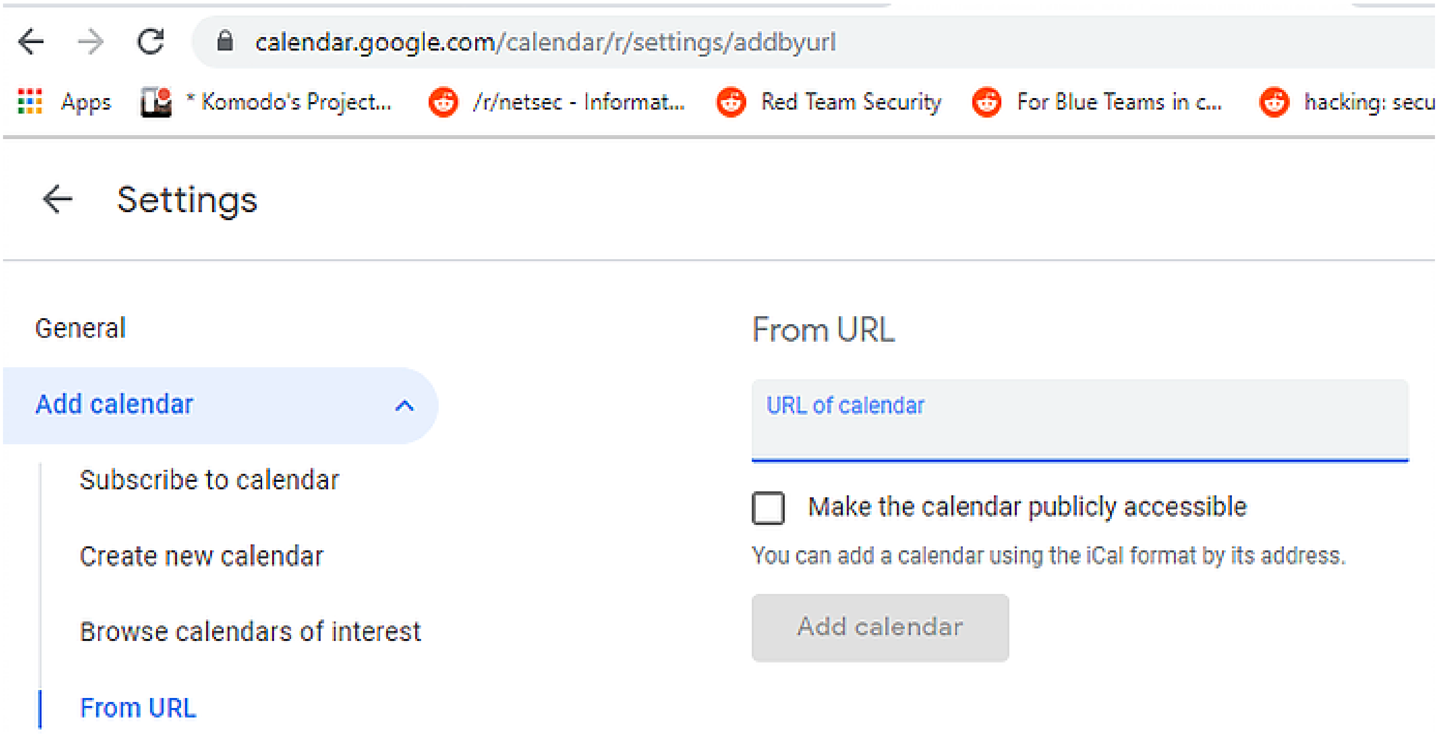
注意到这里curl就是POST参数（可能是calendar url的缩写），而不是一个curl工具。

犹豫不决，那就扫描！

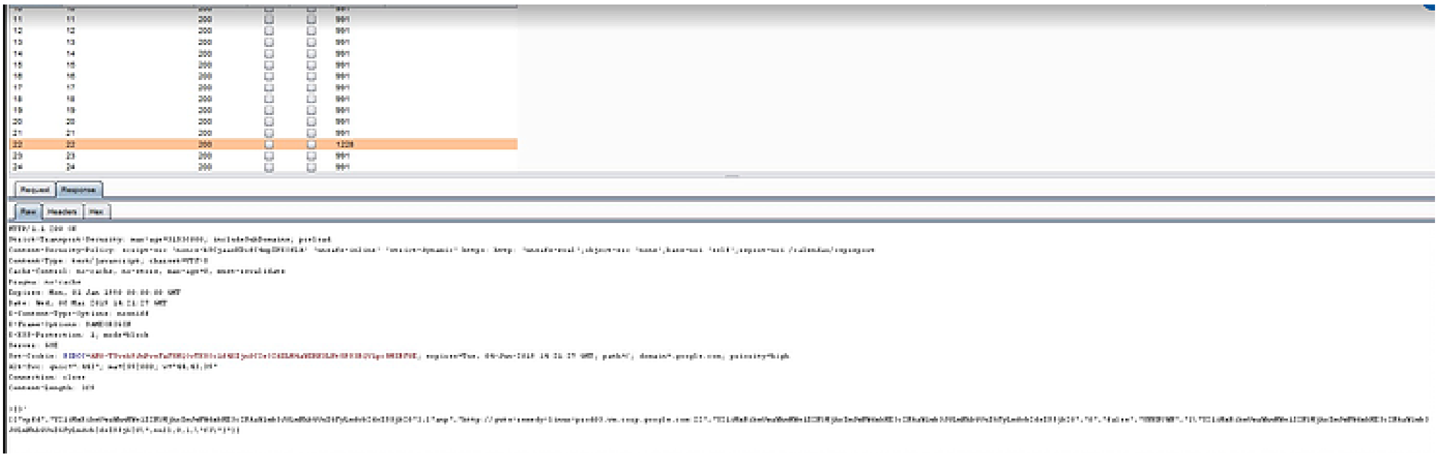
只是为了确信我没有错过什么（可能没有127.0.0.1:443这个端点）。我决定针对localhost上的端口做一个快速的自动化扫描（使用Fiddler的组件发送序列包）。令我惊讶的



这个结果告诉我们有些端口可能开着，有些端口可能关着。我调整了我的代理工具然后再Burp Intruder里跑扫描（payload是端口，不同的结果长度意味着端口可能开着或关着）。这个结果和我想要的一样。



在图中可以看到很明显，80,443,22都开着（从内部服务器访问），其他的测试端口都关着。我快速对内网服务器（guts-remedy-linux-prod03.vm.corp.google.com）进



```
11    11    200    □    □    991
12    12    200    □    □    991
13    13    200    □    □    991
14    14    200    □    □    991
15    15    200    □    □    991
16    16    200    □    □    991
17    17    200    □    □    991
18    18    200    □    □    991
19    19    200    □    □    991
20    20    200    □    □    991
21    21    200    □    □    991
22    22    200    □    □    1228
23    23    200    □    □    991
24    24    200    □    □    991
```

```
HTTP/1.1 200 OK
Strict-Transport-Security: [illegible]
Content-Security-Policy: [illegible]
Content-Type: text/javascript; charset=UTF-8
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: Mon, 01 Jan 1990 00:00:00 GMT
Date: [illegible] GMT
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
Server: GSE
Set-Cookie: [illegible]
Alt-Svc: [illegible]
Connection: close
Content-Length: [illegible]

)]}'
[illegible]
```

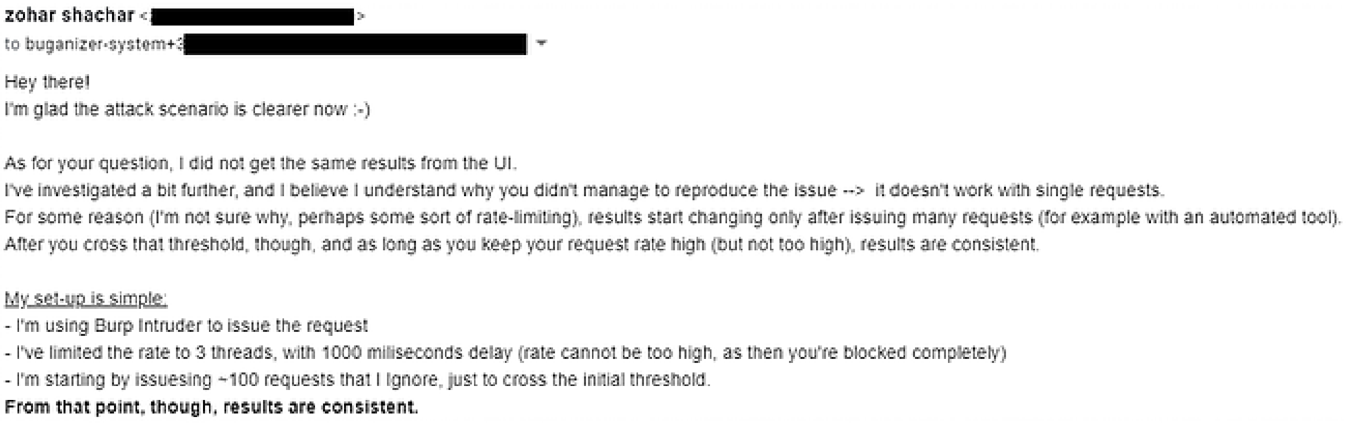
通过这个点我可以确信：

1. 我通过Google服务器发送了内部的HTTP的GET请求，并且我可以做一个自动化脚本；
2. 我可以接收到一些有用的信息（端口是否开着）。

或者换个说法，我在Google日历中发现了一个“盲SSRF”，是时候报告它。

Google工程师复现发现一个bug

我和Google的团队的交流非常迅速。在验证这些问题时他们发现有点复杂，生成结果的时候所做的自动化扫描的时候不能太快（有限制因素在其中）。但是尽管如此，问题



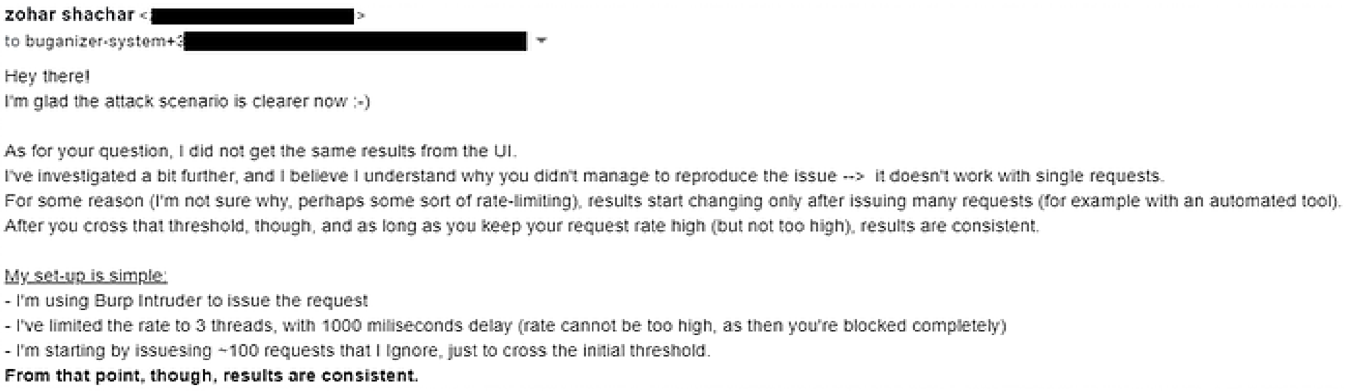
zohar shachar <[redacted]>
to buganizer-system+3[redacted]

Hey there!
I'm glad the attack scenario is clearer now :-)

As for your question, I did not get the same results from the UI.
I've investigated a bit further, and I believe I understand why you didn't manage to reproduce the issue --> it doesn't work with single requests.
For some reason (I'm not sure why, perhaps some sort of rate-limiting), results start changing only after issuing many requests (for example with an automated tool).
After you cross that threshold, though, and as long as you keep your request rate high (but not too high), results are consistent.

<u>My set-up is simple:</u>
- I'm using Burp Intruder to issue the request
- I've limited the rate to 3 threads, with 1000 miliseconds delay (rate cannot be too high, as then you're blocked completely)
- I'm starting by issuesing ~100 requests that I Ignore, just to cross the initial threshold.

From that point, though, results are consistent.



zohar shachar <[redacted]>
to buganizer-system+3[redacted]

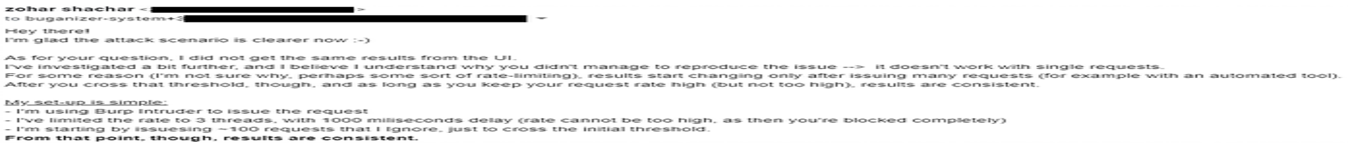
Hey there!
I'm glad the attack scenario is clearer now :-)

As for your question, I did not get the same results from the UI.
I've investigated a bit further, and I believe I understand why you didn't manage to reproduce the issue --> it doesn't work with single requests.
For some reason (I'm not sure why, perhaps some sort of rate-limiting), results start changing only after issuing many requests (for example with an automated tool).
After you cross that threshold, though, and as long as you keep your request rate high (but not too high), results are consistent.

<u>My set-up is simple:</u>
- I'm using Burp Intruder to issue the request
- I've limited the rate to 3 threads, with 1000 miliseconds delay (rate cannot be too high, as then you're blocked completely)
- I'm starting by issuesing ~100 requests that I Ignore, just to cross the initial threshold.

From that point, though, results are consistent.

所以Google的团队可以复现这个问题，但是他们因为UI的问题导致收到了不一样的结果。为了验证Google的理论，我创建了一个新的没有使用痕迹的Google日历账号并再

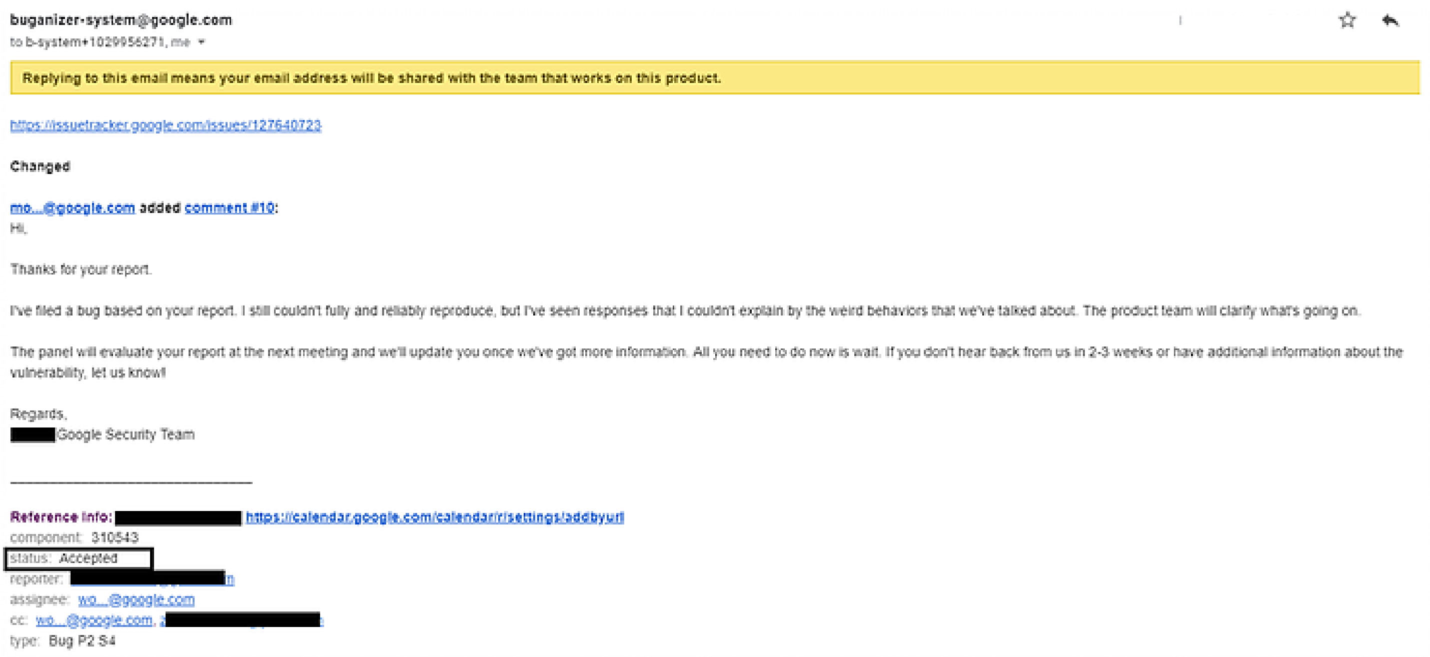


zohar shachar <[redacted]>
to buganizer-system+3[redacted]
Hey there!
I'm glad the attack scenario is clearer now :-)

As for your question, I did not get the same results from the UI.
I've investigated a bit further, and I believe I understand why you didn't manage to reproduce the issue --> it doesn't work with single requests.
For some reason (I'm not sure why, perhaps some sort of rate-limiting), results start changing only after issuing many requests (for example with an automated tool).
After you cross that threshold, though, and as long as you keep your request rate high (but not too high), results are consistent.

<u>My set-up is simple:</u>
- I'm using Burp Intruder to issue the request
- I've limited the rate to 3 threads, with 1000 miliseconds delay (rate cannot be too high, as then you're blocked completely)
- I'm starting by issuesing ~100 requests that I Ignore, just to cross the initial threshold.
From that point, though, results are consistent.

这个点上Google的团队和我认为的一样并且开了一个新的bug。



buganizer-system@google.com
to b-system+1029956271, me

Replying to this email means your email address will be shared with the team that works on this product.

https://issuetracker.google.com/issues/127640723

Changed

mo...@google.com added comment #10:
Hi,

Thanks for your report.

I've filed a bug based on your report. I still couldn't fully and reliably reproduce, but I've seen responses that I couldn't explain by the weird behaviors that we've talked about. The product team will clarify what's going on.

The panel will evaluate your report at the next meeting and we'll update you once we've got more information. All you need to do now is wait. If you don't hear back from us in 2-3 weeks or have additional information about the vulnerability, let us know!

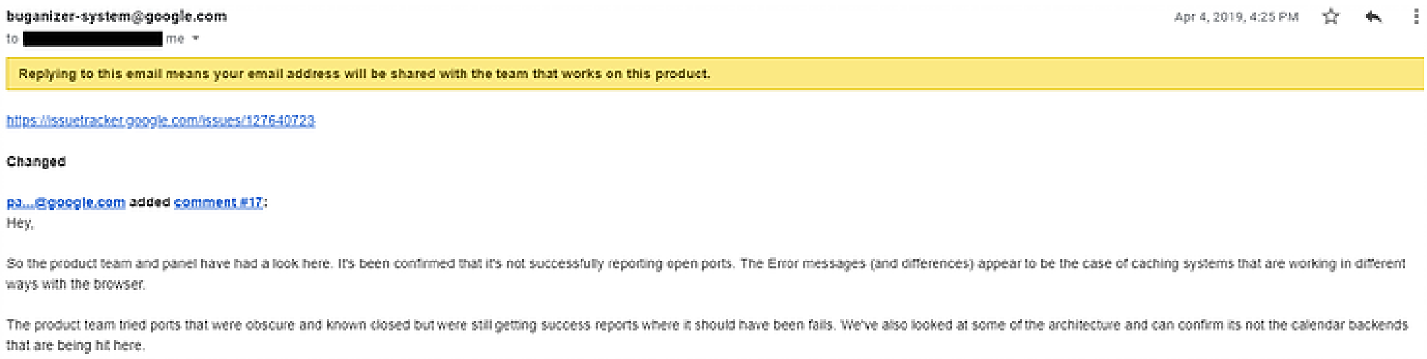
Regards,
[redacted] Google Security Team

____

Reference Info: [redacted] https://calendar.google.com/calendar/r/settings/addbyurl
component: 310543
status: Accepted
reporter: [redacted]n
assignee: wo...@google.com
cc: wo...@google.com, [redacted]
type: Bug P2 S4

产品团队发现问题和安全无关

做完这些研究后，Google的产品团队发现这个问题和安全无关。我所找到的打开的端口实际上是因为一个缓存机制没有按预想一样工作导致的。



buganizer-system@google.com
to [redacted] me
Apr 4, 2019, 4:25 PM

Replying to this email means your email address will be shared with the team that works on this product.

https://issuetracker.google.com/issues/127640723

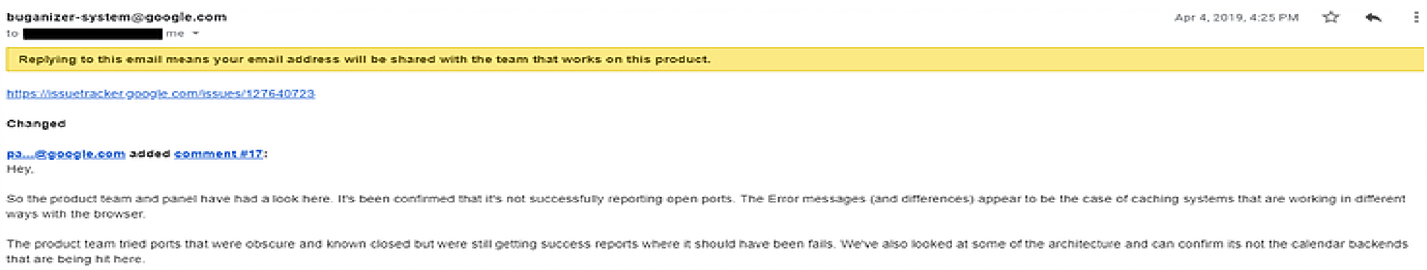
Changed

pa...@google.com added comment #17:
Hey,

So the product team and panel have had a look here. It's been confirmed that it's not successfully reporting open ports. The Error messages (and differences) appear to be the case of caching systems that are working in different ways with the browser.

The product team tried ports that were obscure and known closed but were still getting success reports where it should have been fails. We've also looked at some of the architecture and can confirm its not the calendar backends that are being hit here.

报告被关闭了我收到了回复。



buganizer-system@google.com
to [redacted] me
Apr 4, 2019, 4:25 PM

Replying to this email means your email address will be shared with the team that works on this product.

https://issuetracker.google.com/issues/127640723

Changed

pa...@google.com added comment #17:
Hey,

So the product team and panel have had a look here. It's been confirmed that it's not successfully reporting open ports. The Error messages (and differences) appear to be the case of caching systems that are working in different ways with the browser.

The product team tried ports that were obscure and known closed but were still getting success reports where it should have been fails. We've also looked at some of the architecture and can confirm its not the calendar backends that are being hit here.

最后的思考

我非常尊重Google的团队，因为他们没有放弃这个，直到他们整理出来（缓存问题BTW已修复 - 同样的错误不再返回）。
不能说我没有失望地发现在旅程结束时，我的SSRF只不过是一个幽灵，但我当然很享受骑行。
我认为对我来说最重要的一点就是不要因最初失败的结果而气馁，因为有时候更深层次的潜水会带你进入复杂而迷人的道路。
这次它导致了一个行为不端的缓存，也许下次会导致一个RCE :)

点击收藏 | 0 关注 | 1

上一篇：code-breaking全部题解... 下一篇：pwn堆入门系列教程4

1. 0 条回复
   - 动动手指，沙发就是你的了！

登录 后跟帖

先知社区