

New issue

[Jump to bottom](#)

SSRF #115

✓ Closed

keworr opened this issue on May 25 · 17 comments · Fixed by [#117](#)

keworr commented on May 25 • edited ▼

Describe

There is a way to bypass your regex to validate private & local networks.

If we use <http://127.0.0.1/> or <http://localhost/> to link preview, we don't see it (Error: link-preview-js did not receive a valid a url or text), but if we use a domain that resolved to 127.0.0.1, we can. For example: localtest.me resolved to 127.0.0.1 (localhost), i.e. If you 'curl localtest.me', you'll see your localhost.

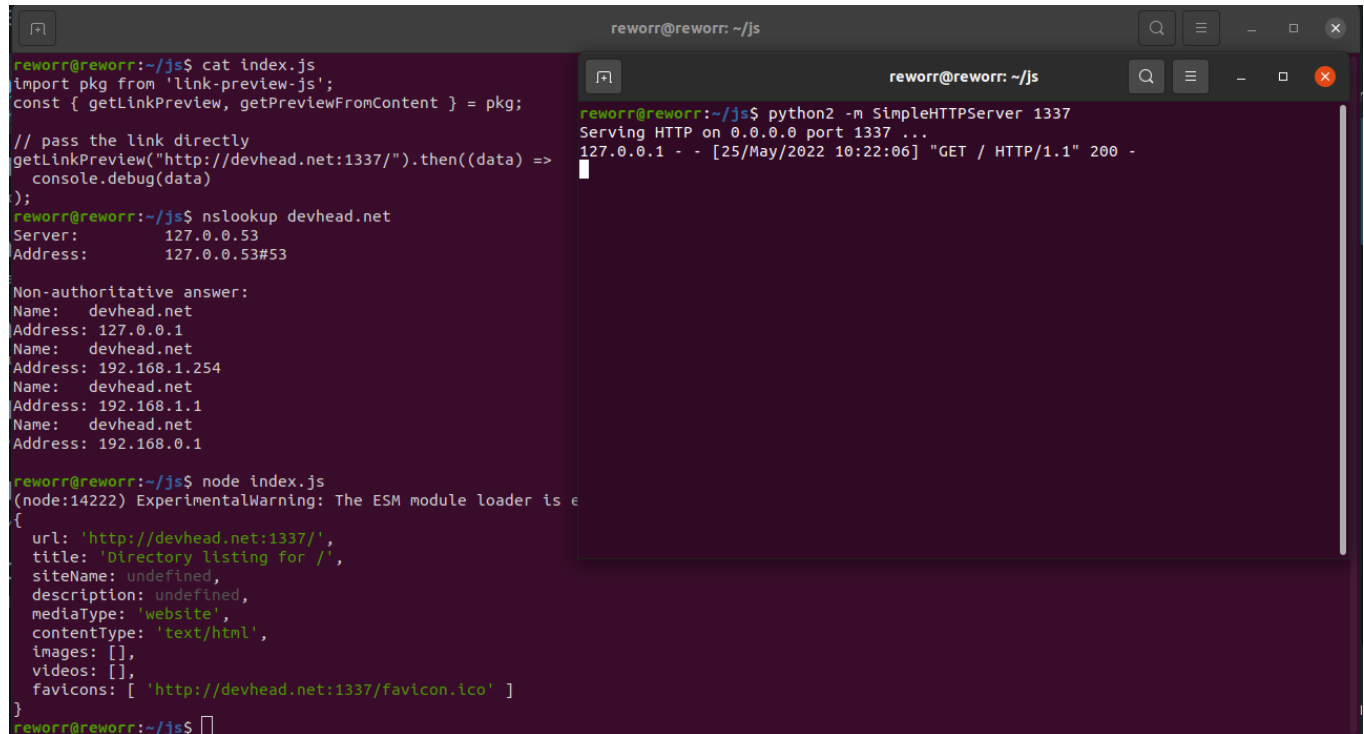
Similarly we can read any other private & local address, any port.

To Reproduce

Steps to reproduce:

1. Find domain that resolved to private address with reverse ip lookup or use localtest.me (127.0.0.1) or devhead.net (127.0.0.1 + 192.168.1.1 + 192.168.0.1).
2. Write it to getLinkPreview.
3. Done. You see your local domain.

Screenshots



The image shows two terminal windows. The left window shows a JavaScript script being executed with `node index.js`. The script uses the `link-preview-js` package to fetch a link preview for `http://devhead.net:1337/`. It also shows the output of `nslookup devhead.net`, which lists several IP addresses for `devhead.net`, including `127.0.0.1`. The right window shows a Python HTTP server running on port 1337, receiving a GET request from `127.0.0.1` at `25/May/2022 10:22:06`.

```
reworr@reworr: ~/js$ cat index.js
import pkg from 'link-preview-js';
const { getLinkPreview, getPreviewFromContent } = pkg;

// pass the link directly
getLinkPreview("http://devhead.net:1337/").then((data) =>
  console.debug(data)
);

reworr@reworr:~/js$ nslookup devhead.net
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   devhead.net
Address: 127.0.0.1
Name:   devhead.net
Address: 192.168.1.254
Name:   devhead.net
Address: 192.168.1.1
Name:   devhead.net
Address: 192.168.0.1

reworr@reworr:~/js$ node index.js
(node:14222) ExperimentalWarning: The ESM module loader is experimental
{
  url: 'http://devhead.net:1337/',
  title: 'Directory listing for /',
  siteName: undefined,
  description: undefined,
  mediaType: 'website',
  contentType: 'text/html',
  images: [],
  videos: [],
  favicons: [ 'http://devhead.net:1337/favicon.ico' ]
}
reworr@reworr:~/js$
```

```
reworr@reworr: ~/js$ python2 -m SimpleHTTPServer 1337
Serving HTTP on 0.0.0.0 port 1337 ...
127.0.0.1 - - [25/May/2022 10:22:06] "GET / HTTP/1.1" 200 -
```

ospfranco commented on May 25

Owner

Which version are you using? At some point, someone submitted a PR to patch redirections, by default it should not follow any.

ospfranco commented on May 25

Owner

I don't quite get what is the issue though... these sites seem to register the loopback address on DNS level. Even curl returns the data of the directory, then it also means that curl has an SSRF vulnerability?

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help
.erj(abi/mobile (-zsh) Metro (node) ~/Developer (-zsh) python3 (Python)
$ curl localhost.me:1337
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Directory listing for /</title>
</head>
<body>
<h1>Directory listing for /</h1>
<hr>
<ul>
<li><a href=".DS_Store">.DS_Store</a></li>
<li><a href="abl/">abl/</a></li>
<li><a href="access/">access/</a></li>
<li><a href="bodyfast/">bodyfast/</a></li>
<li><a href="capmo/">capmo/</a></li>
<li><a href="cidemon/">cidemon/</a></li>
<li><a href="cidemon_site/">cidemon_site/</a></li>
<li><a href="flatlisttest/">flatlisttest/</a></li>
<li><a href="jsi-rn-wallet-core/">jsi-rn-wallet-core/</a></li>
<li><a href="kipu/">kipu/</a></li>
<li><a href="messenger/">messenger/</a></li>
<li><a href="messenger_rn/">messenger_rn/</a></li>
<li><a href="messenger_site/">messenger_site/</a></li>
<li><a href="ospfranco.github.io/">ospfranco.github.io/</a></li>
<li><a href="productlane/">productlane/</a></li>
<li><a href="quick-sqlite/">quick-sqlite/</a></li>
<li><a href="site_sol/">site_sol/</a></li>
<li><a href="sol/">sol/</a></li>
<li><a href="sol.app/">sol.app/</a></li>
<li><a href="Sol.app.zip">Sol.app.zip</a></li>
<li><a href="Sol.zip">Sol.zip</a></li>
<li><a href="StorageBenchmark/">StorageBenchmark/</a></li>
<li><a href="turbo-secure-storage/">turbo-secure-storage/</a></li>
<li><a href="wallet-core/">wallet-core/</a></li>
<li><a href="WerkerApp/">WerkerApp/</a></li>
</ul>
<hr>
</body>
</html>
~/Developer
$

$ python3 -m http.server 1337
Serving HTTP on :: port 1337 (http://[::]:1337/) ...
::ffff:127.0.0.1 - - [25/May/2022 10:01:44] "GET / HTTP/1.1" 200 -
```

ospfranco commented on May 25

Owner

The regex is not a security feature, it's only a way to detect links in the text. As stated, the library does not follow redirections by default.

#105



ospfranco closed this as completed on May 25

keworrr commented on May 25

Author

But you accepted same SSRF here - #105?

ospfranco commented on May 25 • edited ▾

Owner

I really don't understand what you mean. In that issue, the author submitted a PR not to follow redirects, which is now the default behavior. The same author complained that it is possible to pass the regex with a localhost url, like I said, the regex is not meant to be a security feature, it is only there for link detection. Once the url is passed to the fetch library, there is no way to re-validate the url (even if the regex was meant to do that, which it is not) . On the other hand, I already showed you that curl also has this same behavior, because the domains you send use the localhost address on a DNS level, so that means (as far as I can see) that there is no SSRF vulnerability, this is just the expected behavior.

keworr commented on May 25 • edited ▼

Author

Redirects were disabled in that issue because it allows to read local hosts, right?
And my way also allows to read local hosts?
idk what is different

Yes, curl by default allow same, also curl allow requests as `curl file:///etc/passwd` by default, SMB requests, FTP, etc. Also e.g. any browsers allow redirects by default, but you disabled it.

I just saw that issue and thought that SSRF is sensitive here, if not - ok.

ospfranco commented on May 25

Owner

redirects were disallowed not because localhost, but because redirections are by default insecure. Your way allows to reach localhost, not because the library is doing something wrong, but because the domains you provided loopback to localhost **on the DNS level**, they are not intercepting the request and redirecting, they are straight up returning 127.0.0.1 when their DNS address is resolved. There is nothing I can do about that.

  **keworr** mentioned this issue on May 25

Invite to issue [aaydin-tr/svn-scanner#1](#)

✓ Closed

aaydin-tr commented on May 25 • edited ▼

Contributor

@**ospfranco** first of all i install lastest version of package for testing my local machine and cant see `followRedirects` option in source could there be a problem with npm? can you check please

```

scripts: {
  "test": "echo \"Error: no test specified\" && exit 1"
},
"author": "",
"license": "ISC",
"dependencies": {
  "link-preview-js": "2.1.13"
}

```

node_modules > link-preview-js > build > index.d.ts > ILinkPreviewOptions

```

1 interface ILinkPreviewOptions {
2   headers?: Record<string, string>;
3   imagesPropertyType?: string;
4   proxyUrl?: string;
5   timeout?: number;
6 }
7 interface IPreFetchedResource {
8   headers: Record<string, string>;
9   status?: number;
10  imagesPropertyType?: string;
11  proxyUrl?: string;
12  url: string;
13  data: string;
14 }

```

node_modules > link-preview-js > build > index.js > getLinkPreview > _awaiter() callback > _generator() callback

```

320 var _c;
321 return __generator(this, function (_d) {
322   switch (_d.label) {
323     case 0:
324       if (!text || typeof text !== "string") {
325         throw new Error("link-preview-js did not receive a valid url or text");
326       }
327       detectedUrl = text
328         .replace(/\n/g, " ")
329         .split(" ")
330         .find(function (token) { return constants_1.CONSTANTS.REGEX_VALID_URL.test(token); });
331       if (!detectedUrl) {
332         throw new Error("link-preview-js did not receive a valid a url or text");
333       }
334       timeout = (_a = options === null || options === void 0 ? void 0 : options.timeout) !== null && _a !== void 0 ? _a : 3000;
335       controller = new abort_controller_1.default();
336       timeoutCounter = setTimeout(function () { return controller.abort(); }, timeout);
337       fetchOptions = {
338         headers: (_b = options === null || options === void 0 ? void 0 : options.headers) !== null && _b !== void 0 ? _b : {},
339         redirect: "error",
340         signal: controller.signal,
341       };
342       fetchUrl = (options === null || options === void 0 ? void 0 : options.proxyUrl)
343         ? options.proxyUrl.concat(detectedUrl)
344         : detectedUrl;
345       return [4 /*yield*/, cross_fetch_1.fetch(fetchUrl, fetchOptions).catch(function (e) {
346         if (e.name === "AbortError") {
347           throw new Error("Request timeout");
348         }
349         throw e;
350       })];

```

Second of all i agree with @keworr this vulnerability is called DNS pinning (or DNS rebinding) and it allow to bypass all control mechanism on DNS level but library has proxy option and the purpose of proxy is to fix such DNS level problems and vulnerabilities. But if we want the library to prevent such DNS level attacks as well we can fix this by dns lookup before make request

ospfranco commented on May 26

Owner

hi @AbdurrahmanA, you are right I published a new version of the package, probably some cache was left on my machine when I published one of the newer versions. Thanks a lot!

Thanks a lot for the explanation, that was what I needed. [This StackOverflow answer](#) seems to suggest there is no way to resolve the DNS address from JavaScript, seems like the solution is to self-host a proxy?

aaydin-tr commented on May 26

Contributor

Actually if we added that kind of option to library it would be great i would love have that option. We can easily do `nslookup` for server side, for client side we can use `https://dns.google/resolve?name=localtest.me` or that option wont be available for client side. By the way this library should not be in client side it wil get lots of CORS error 😊

ospfranco commented on May 27 • edited ▼

Owner

If you mean hardcoding a DNS service, not a fan (and especially google's), but maybe an option to pass a DNS resolver if one chooses to do so.

The library will face CORS errors on websites, on React Native, Cordova, etc it works just fine

aaydin-tr commented on May 29

Contributor

We can easily set DNS resolver for Node js side, but if we do this using native `dns` library, will it be a problem for client-side users? otherwise we should make an online dns resolver option right?

ospfranco commented on May 30

Owner

I'm still not sure if the DNS resolver should always be there, just from a latency point of view, it might affect people who have already deployed this. But just passing a function that resolves the host might be enough for us to check against common attacks?

```
getLinkPreview('some text with a link', {  
  // Must resolve the url host, internally checks for loopbacks and common SSRF attacks  
  resolveDNSHost: (url: string) => Promise<string>  
})
```



ospfranco reopened this on May 30

ospfranco commented on May 30

Owner

On the other hand, I don't think this would be an issue on mobiles. Almost nobody has a server running on the phone for the loopback address to be considered dangerous. Maybe conditionally importing/requiring the DNS resolver on node is enough...

aaydin-tr commented on Jun 10

Contributor

Sorry for late response, Yes we just need the DNS resolver on node can we do that and also, we need to warn people about common SSRF vulnerabilities and make sure people understand what this library does.

ospfranco commented on Jun 13

Owner

Most people won't care 🙄 they barely read the CORS warning. But I'll try to find some time in the coming weeks to add the option, otherwise PRs are welcome

aaydin-tr commented on Jun 13

Contributor

Great to hear that, i could also do it if i have time, i will inform you 🙌

  ospfranco mentioned this issue on Jun 25

Add dns resolver #117

 Merged

 ospfranco closed this as completed in [#117](#) on Jun 27

Assignees

No one assigned

Labels

None yet

Projects


None yet

Milestone

No milestone

Development

Successfully merging a pull request may close this issue.

 **Add dns resolver**
ospfranco/link-preview-js

3 participants

