keeping javascript safe

keeping javascript safe

security & the npm registry

C J Silverio

CTO @ npm, @ceejbot

using node since 2011
node has grown up!

running npm's registry since 2014

npm has grown up too!

the story of the npm registry **mirrors** the story of node

npm is **infrastructure** for millions of developers

npm dependably serves node packages 24/7 around the world

Fortune 100 companies depend on npm & node

3 billion downloads/week

9 million users

156K package authors (1.7%)

npm has as many users as the New York City

it didn't start that way

in 2009, node & npm's users knew each other by name

the npm registry is now too large to depend on community policing

but you need that **policing**

you **rely** on the packages you install

questions you ask

1. Is the registry **secure**?
2. Does this package have **vulnerabilities**?
3. Is this package **malware**?
4. **Who** published this package?

1. Is the registry secure?

What does secure mean?

registry systems can't be broken into

data can't be tampered with

we don't try to do this alone ongoing contract with ^Lift

this guy, Adam Baldwin (he'll come up again) & his colleagues

periodic **pen testing**
ongoing **code reviews**

good security practices are on-going work

2. Does this package have **vulnerabilities**?

our friends at ^Lift again
as the **Node Security Platform**

NSP reviews popular packages,

reports vulnerabilities, & handles reports

https://nodesecurity.io

early access NSP data is integrated into npm enterprise

newsflash! npm is a company that sells services!

npm enterprise is a registry inside your firewall

NSP keeps us informed

we keep them informed in turn

3. Is this package malware?

malware doesn't advertise

malware comes in flavors:

spam & poison

spammers found the registry in 2016

two kinds of spam:

spam content & js spam support

npm + cdns built on top == trivial hosting for GA clickjacking

now using machine learning to catch spam thanks to the Smyte service

spam speedbumps:

validated email to publish

disallow throwaway addresses

we seem to have made a dent
but this war will never end

poison-flavored malware:

typosquatting

publishing packages with names that are very close to real names

Historically this was **competitive**: authors would try to steal traffic to pump their download numbers

somebody typosquatted moment.js
with another date-formatting package

also accidental

JSONStream vs jsonstream

recently it's been nefarious:

typosquat of cross-env

as crossenv with a env var stealer

typosquat of bluebird

wrapping bluebird with a cryptocoin miner

Adam Baldwin typosquatted coffee-script early on

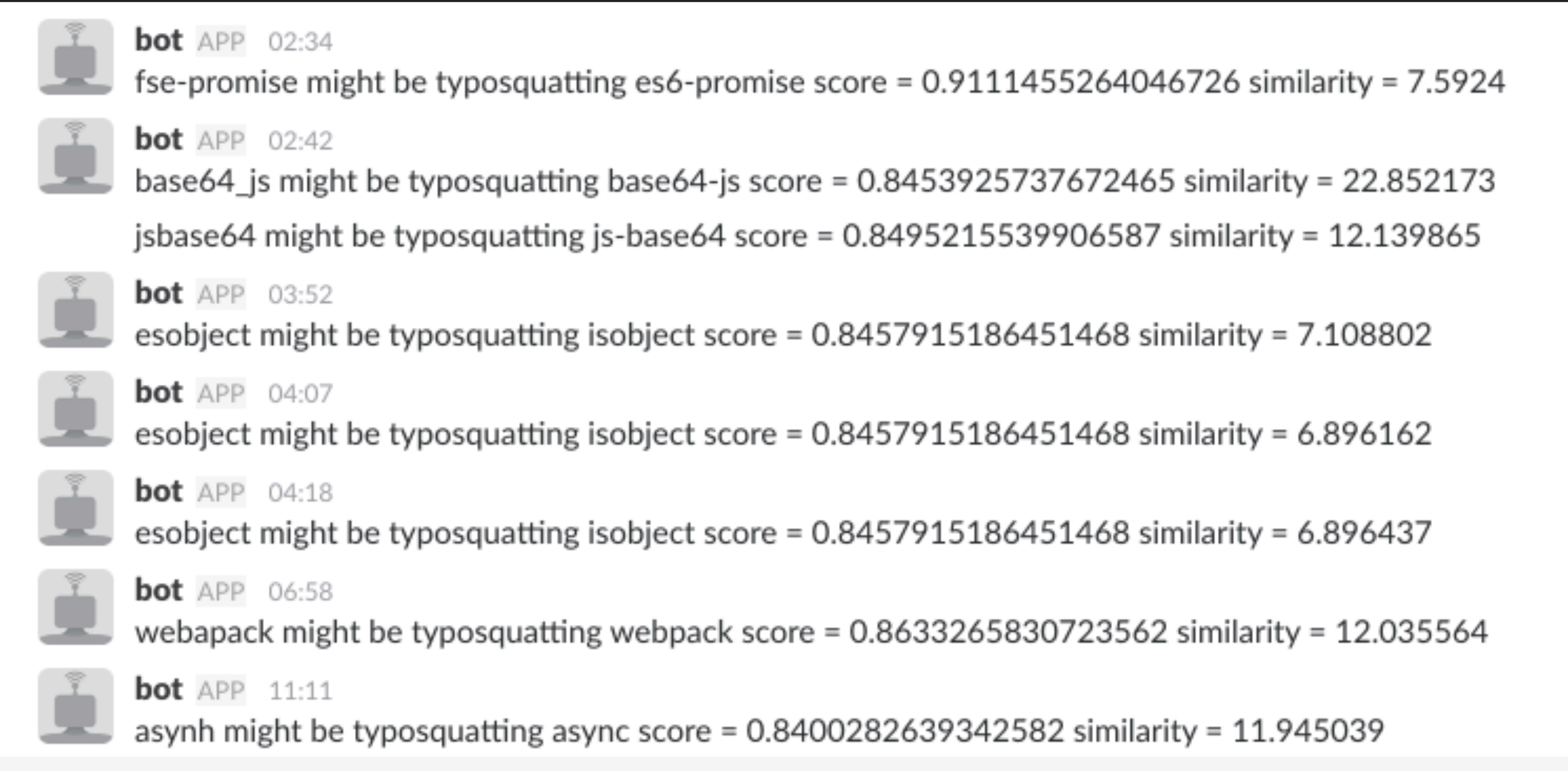
it took **days** for the community to notice

now it takes weeks

if the community notices at all

as spiderman said, with great popularity comes great annoyance

automated similarity checker



bot APP 02:34

fse-promise might be typosquatting es6-promise score = 0.9111455264046726 similarity = 7.5924

bot APP 02:42

base64_js might be typosquatting base64-js score = 0.8453925737672465 similarity = 22.852173

jsbase64 might be typosquatting js-base64 score = 0.8495215539906587 similarity = 12.139865

bot APP 03:52

esobject might be typosquatting isobject score = 0.8457915186451468 similarity = 7.108802

bot APP 04:07

esobject might be typosquatting isobject score = 0.8457915186451468 similarity = 6.896162

bot APP 04:18

esobject might be typosquatting isobject score = 0.8457915186451468 similarity = 6.896437

bot APP 06:58

webapack might be typosquatting webpack score = 0.8633265830723562 similarity = 12.035564

bot APP 11:11

asynh might be typosquatting async score = 0.8400282639342582 similarity = 11.945039 this war will never end

so long as there is $ to be made

4. Who published this package?

What happens if somebody steals JDD's auth token & posts malware as lodash?

Well, that's scary.

npm auth tokens are **sensitive**.

new! tools in the npm cli to help you control auth tokens

new command: npm token
control your auth tokens

```
npm token create --readonly
```

read-only auth tokens
the principle of least power

give your CI system a **read-only** token

```
npm token create --cidr=[10.0.0.1/32]
```

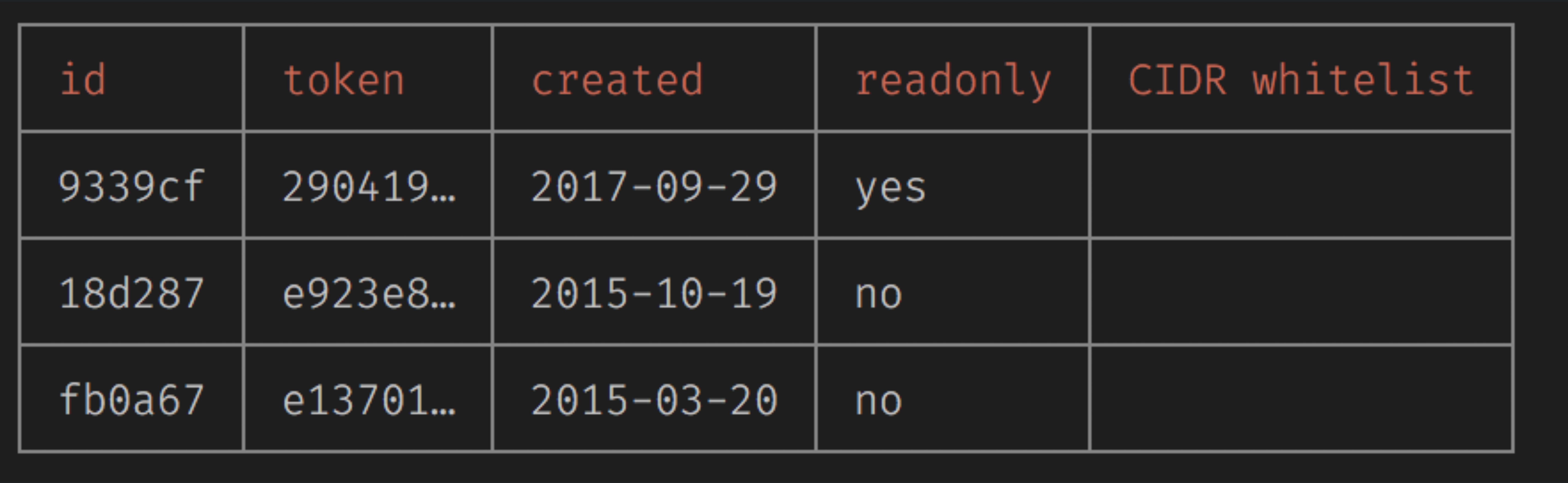
CIDR-bound tokens

bind tokens to IP ranges

further limit your tokens

by controlling where they can be used

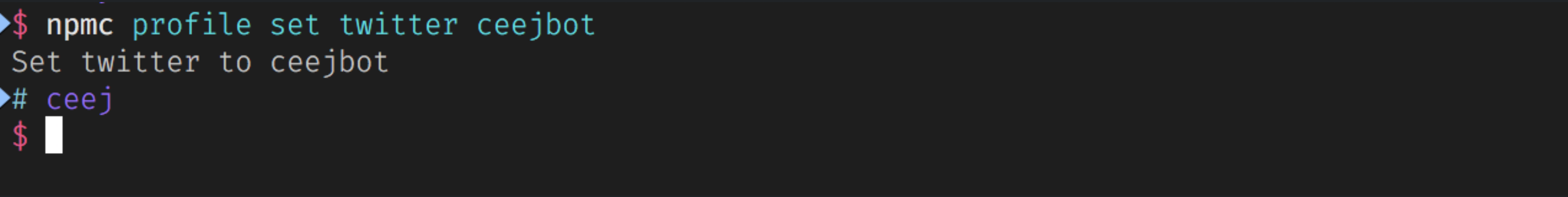
```
npm token list
npm token delete <tokenKey>
```



id	token	created	readonly	CIDR whitelist
9339cf	290419…	2017-09-29	yes	
18d287	e923e8…	2015-10-19	no	
fb0a67	e13701…	2015-03-20	no	

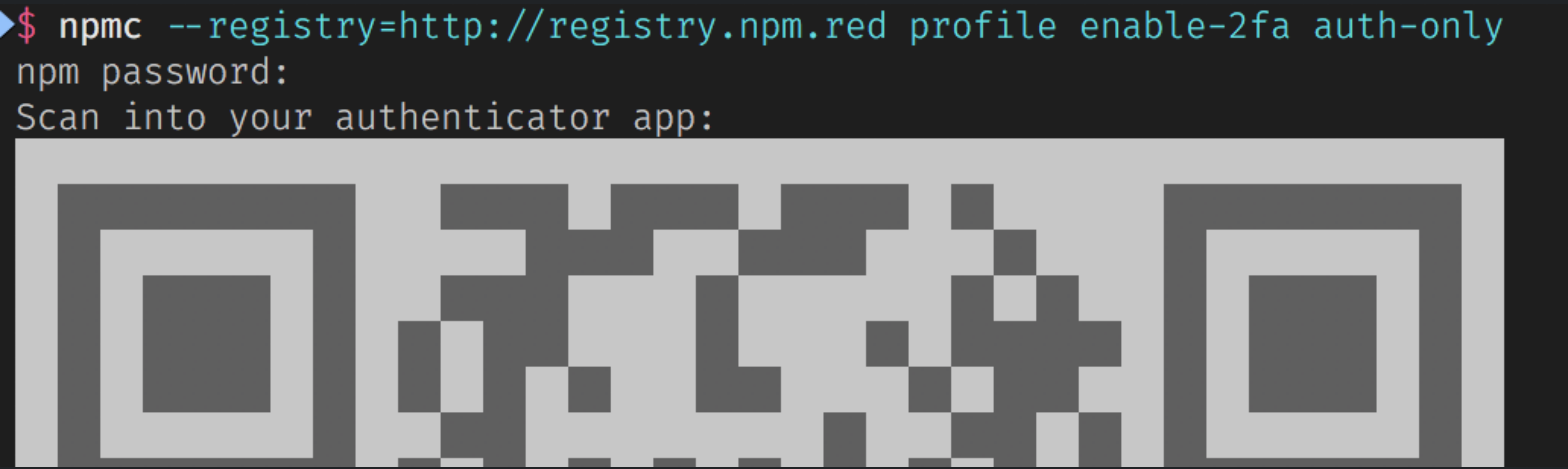
new command: npm profile

set your profile data like your email or ...



```
$ npmc profile set twitter ceejbot
Set twitter to ceejbot
# ceej
$ 
```

well that's boring



```
$ npmc --registry=http://registry.npm.red profile enable-2fa auth-only
npm password:
Scan into your authenticator app:
```

that's **not** boring

npm profile enable-2fa

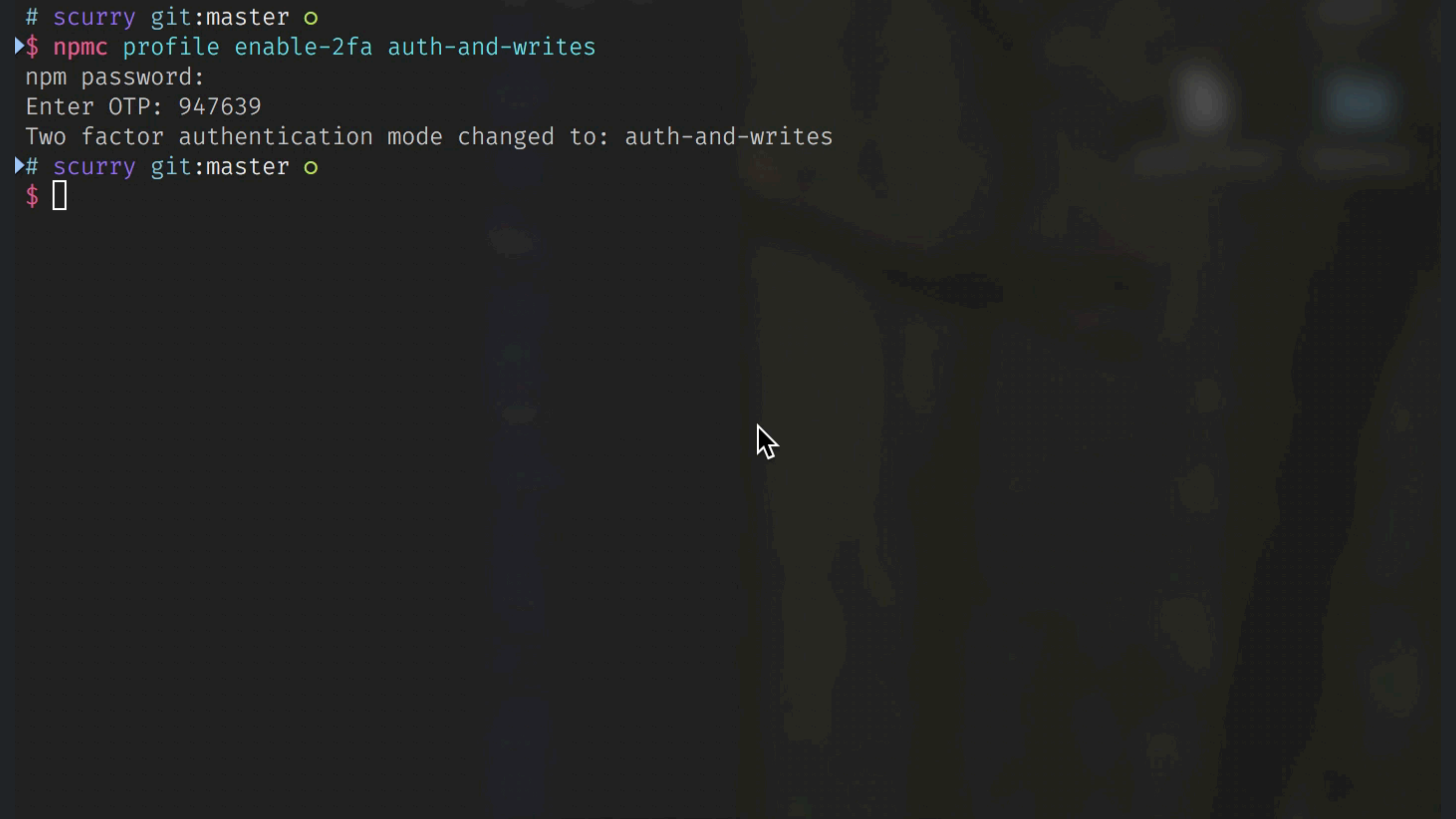
two-factor authentication is here

require regular password plus a one-time password

```
npm profile enable-2fa auth-only
```

auth-only: any time you log in or manipulate tokens

npm profile enable-2fa auth-and-writes



```
 # scurry git:master o
▶$ npmc profile enable-2fa auth-and-writes
 npm password:
 Enter OTP: 947639
 Two factor authentication mode changed to: auth-and-writes
▶# scurry git:master o
 $ ▯
```

writes: your package publications

pass the `--otp` flag

```
npm publish --otp=123456
```

pass it on the command line!

use a **TOTP code generation app**

Google Authenticator, Authy, etc

```
npm install -g npm@next
```

try it now!

code: github.com/npm/npm-profile
api docs: github.com/npm/registry

one more thing

coming attraction!

protect a package with 2FA

require an OTP any time

that package is published by anybody

protect packages with many maintainers
next cli minor release 5.6.0

coming soon! 2fa for your npm organization

coming soon! npm ci

3x speed for your CI installs

but what about **package signing**?
we think we've figured out how

coming soon! even more

questions? help setting this up?

come see me & puppies at the npm booth

npm wants you to develop in confidence

npm loves you