Hi!

I'm Dominik!

About me

Developer Evangelist at

twilio

Get in touch with me!

@dkundel
dkundel@twilio.com
github/dkundel

HACKERS!


HACKERMAN
HM

2FA, WTF?

Two-Factor Authentication


Hacking in
progress...

Two-Factor Authentication

Two different forms of identification from the user

Typically:

→ Something that you know

→ Something that you have

Why?

Passwords Alone Are Weak


Story Time!

Mark Zuckerberg

Users are bad with passwords!

Top 10 Passwords of 2015

1. 123456
2. password
3. 12345678
4. qwerty
5. 12345
6. 123456789
7. football
8. 1234
9. 1234567
10. baseball

Source: https://www.teamsid.com/worst-passwords-2015/


Other websites are bad
with passwords!

LastPass •••|


Mat Honan

Hacking Timeline

→ Hackers find his personal website and then his Gmail

→ Detect alternative email through Gmail password recovery

→ Get Honan's address through `whois` on his domain

→ Phone Amazon to add a new credit card to Honan's account

→ Call again to recover the Amazon account

→ Hacker log into Amazon to retrieve last 4 digits of his actual card

Hacking Timeline

→ **4:33pm** Call Apple to recover the iCloud access using the billing address and 4 digits of the credit card

→ **4:50pm** Permanently reset iCloud password

→ **4:52pm** Reset Gmail password

→ **5:00pm** Hacker delete his iPad and iPhone

→ **5:02pm** Reset Twitter password

→ **5:05pm** Wipe Macbook

→ **5:12pm** Hacker tweet to tack credit


@mat


Social engineering works!

Passwords Alone Are Weak

Physical protection layer for a digital world

27 — Dominik Kundel (@dkundel)

How?

Typical User Registration Flow

1. User visits registration page
2. Enters username and password
3. User is logged in

Typical User Log-in Flow

1. User visits log-in page
2. Enters username and password
3. System verifies details
4. User is logged in

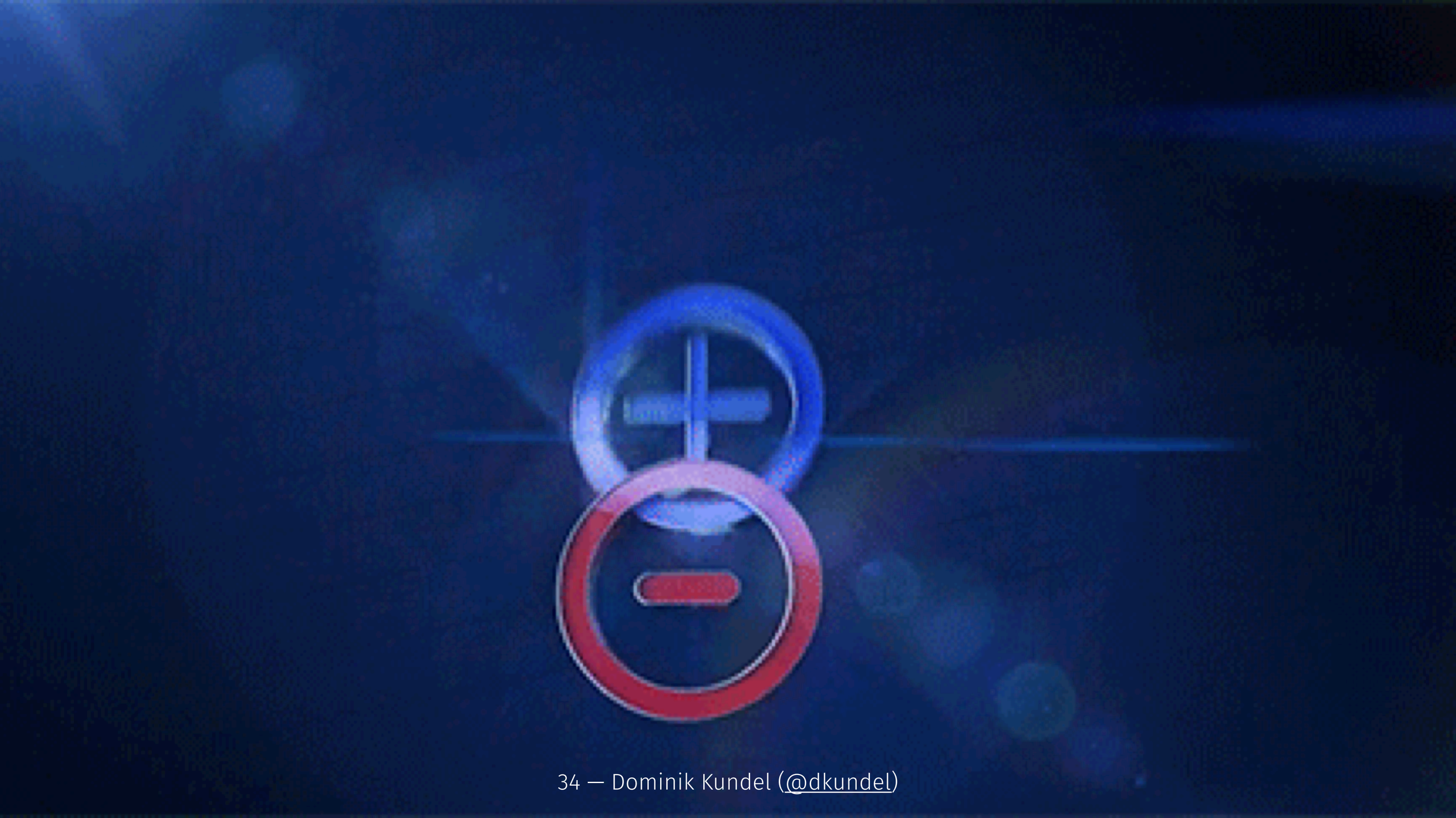

Phone 2FA
SMS / Voice

SMS-based User Registration Flow

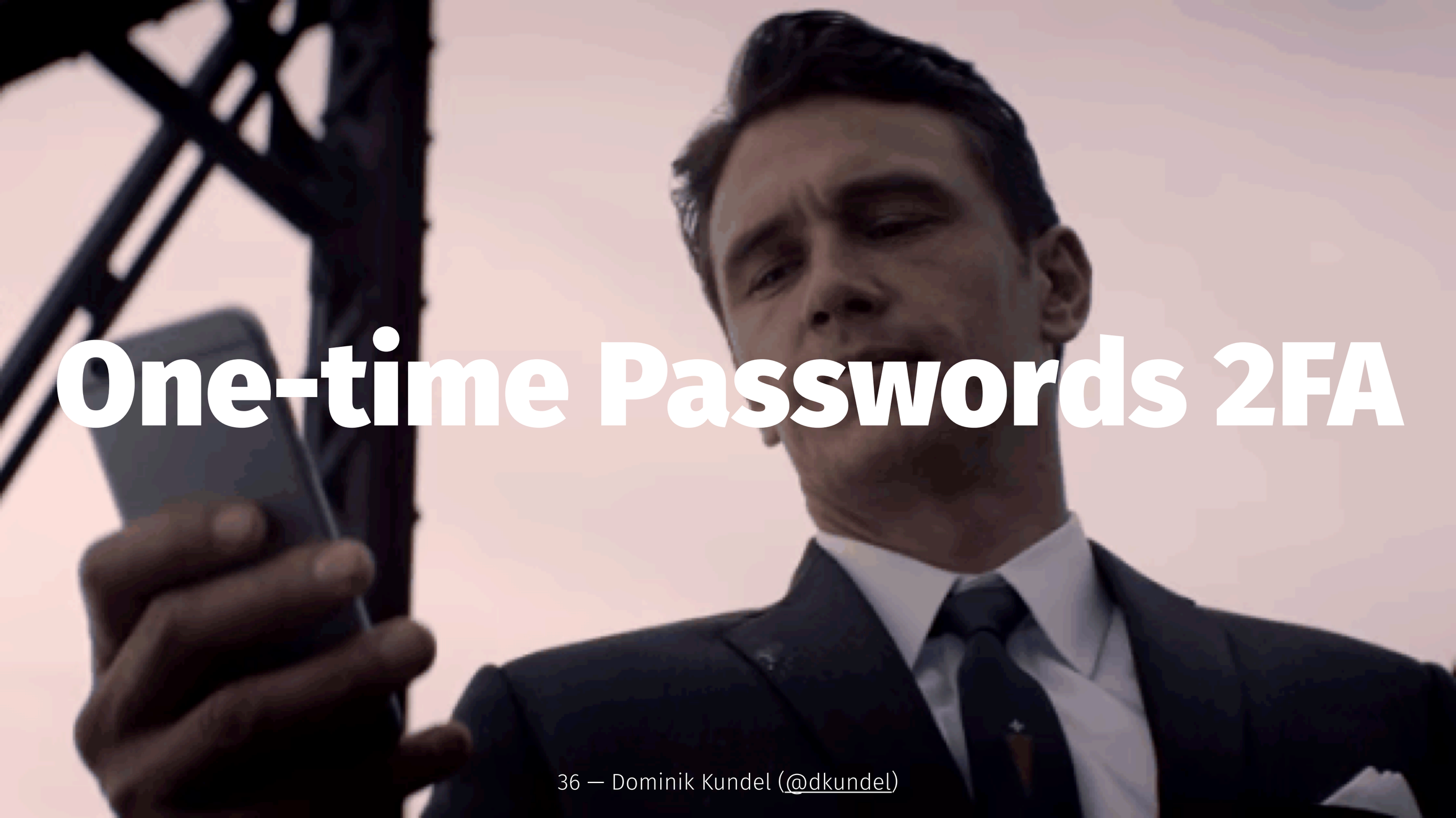
1. User visits registration page
2. Enters username, password and phone number
3. Verifies phone number
4. User is logged in

SMS-based User Log-in Flow

1. User visits log-in page
2. Enters username and password
3. System verifies details
4. System sends verification code to user by SMS
5. User enters verification code
6. System verifies code
7. User is logged in



DeRay Mckesson



One-time Passwords 2FA

OTP-based User Registration Flow

1. User visits registration page
2. Enters username and password
3. Generate secret for the user
4. Share secret with the user
5. User is logged in

OTP-based User Log-in Flow

1. User visits log-in page
2. Enters username and password
3. System verifies details
4. User opens auth app
5. Enters app verification code on site
6. System verifies code
7. User is logged in

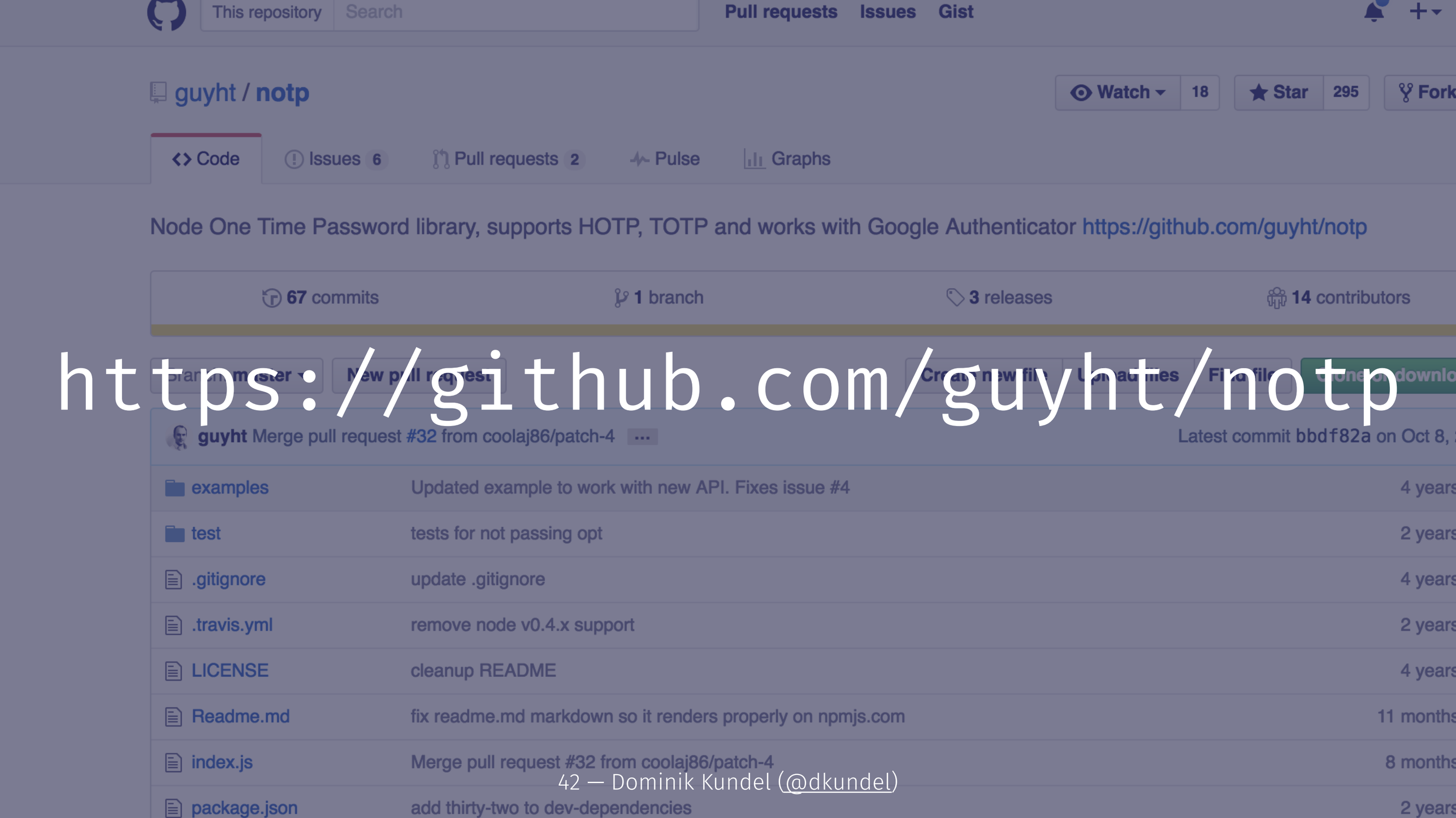
Secret based Codes

HOTP/TOTP

HOTP Formula

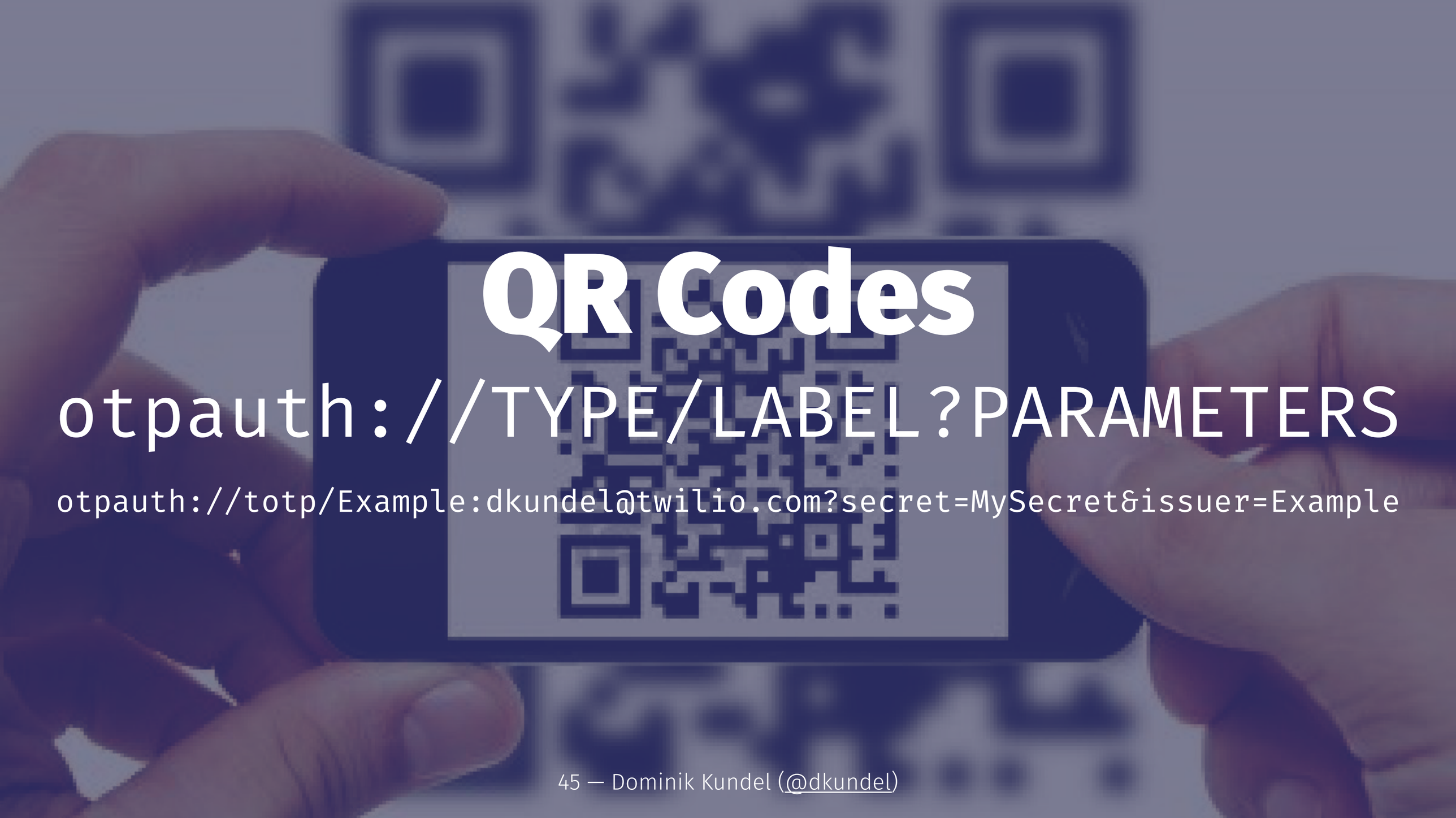
$$HOTP(K,C) = Truncate(HMAC(K,C)) \& 0x7FFFFFFF$$

$$HOTP\text{-}Value = HOTP(K,C) \bmod 10^{d}$$


This repository
Search
Pull requests
Issues
Gist
guyht / notp
Watch 18
Star 295
Code
Issues 6
Pull requests 2
Pulse
Graphs
Node One Time Password library, supports HOTP, TOTP and works with Google Authenticator https://github.com/guyht/notp
67 commits
1 branch
3 releases
14 contributors
https://github.com/guyht/notp
guyht Merge pull request #32 from coolaj86/patch-4
Latest commit bbdf82a on Oct 8,
examples
Updated example to work with new API. Fixes issue #4
4 years
test
tests for not passing opt
2 years
.gitignore
update .gitignore
4 years
.travis.yml
remove node v0.4.x support
2 years
LICENSE
cleanup README
4 years
Readme.md
fix readme.md markdown so it renders properly on npmjs.com
11 months
index.js
Merge pull request #32 from coolaj86/patch-4
8 months
package.json
add thirty-two to dev-dependencies
2 years

DEMO

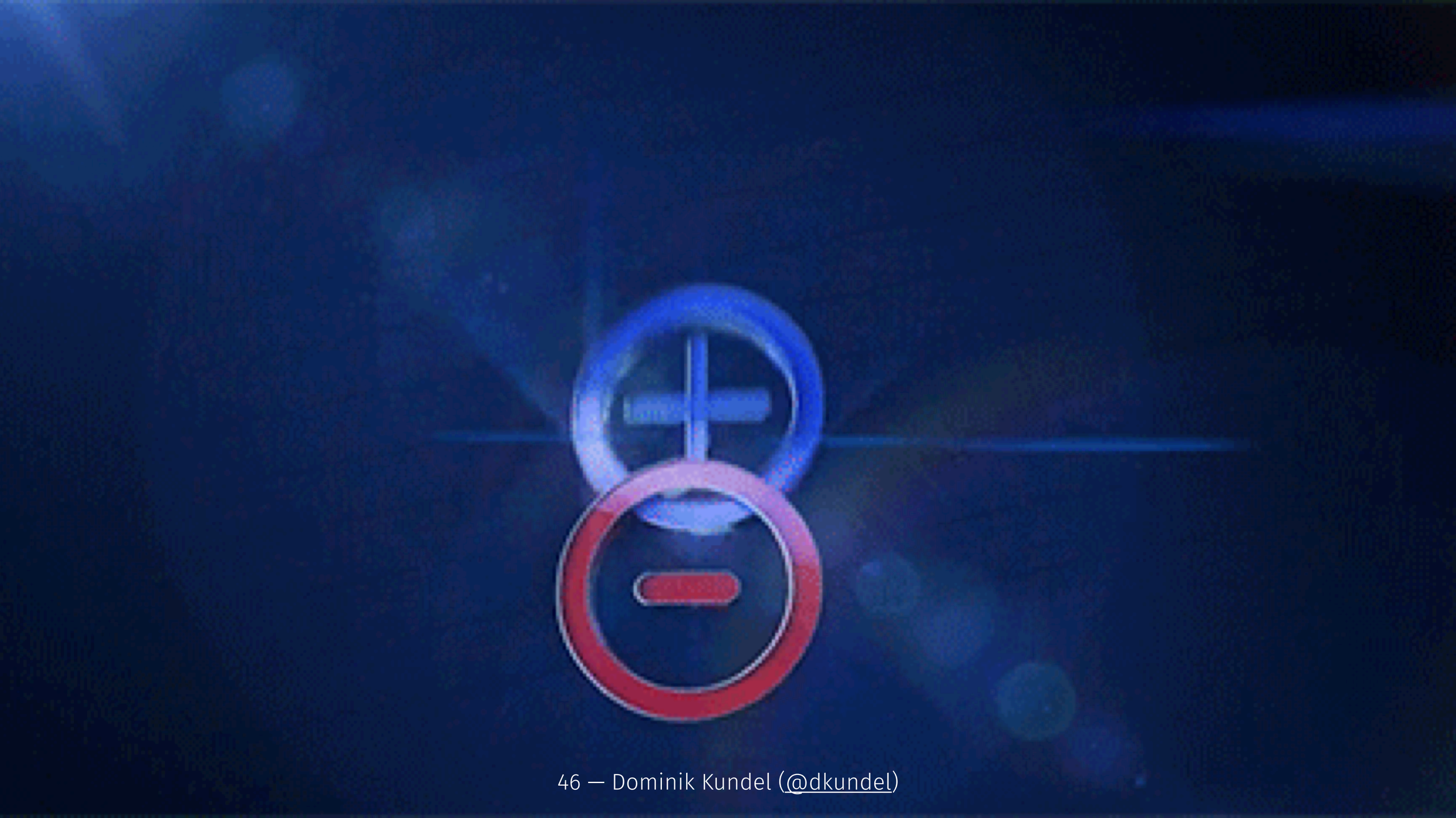
Sharing Secrets



QR Codes

otpauth://TYPE/LABEL?PARAMETERS

otpauth://totp/Example:dkundel@twilio.com?secret=MySecret&issuer=Example



Friends don't let friends write their own authentication frameworks!

Friends don't let friends write their own two-factor authentication frameworks!


AUTHY

Authy-based User Registration Flow

1. User visits registration page
2. Enters username, password and phone number
3. System registers user with Authy
4. User is logged in

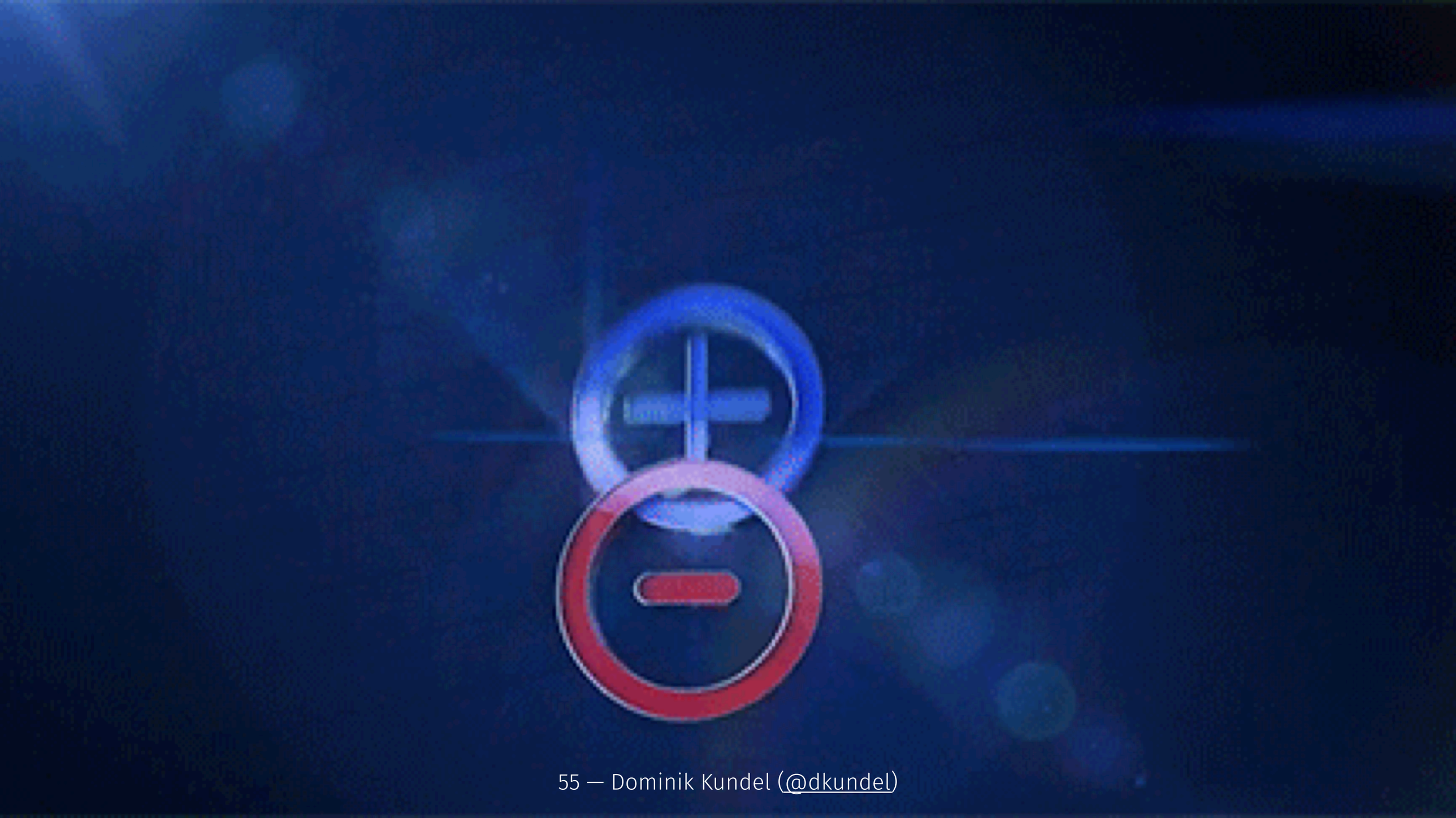
Authy-based User Log-in Flow

1. User visits log-in page
2. Enters username and password
3. System verifies details
4. Authy prompts user
5. User enters app verification code on site
6. System verifies success with Authy
7. User is logged in


UX
or
2FA

Push notifications (OneTouch)


Demo



Summary

Users are bad with passwords!

Other websites are bad with passwords!

Social engineering works!

2FA can be push, tokens or SMS

2FA is for your users!


PASSWORD

I need your help!

On a scale from *0* to *10* how likely is it that would you recommend this talk to a friend or colleague:

+49 1111 1111 1111

Thank You!

On a scale from *0* to *10* how likely is it that would you recommend this talk to a friend or colleague:

+49 1111 1111 1111

@dkundel
dkundel@twilio.com
github/dkundel

Credits:

http://www.hackercg.com/wp-content/uploads/2015/12/Hacker.jpg
http://www.v3.co.uk/IMG/494/302494/hacker-hacking-dark-hoodie.jpg
http://qruniversity.hipscan.net/sites/default/files/article-images/computer-hacker.jpg
http://www.wpdroids.com/wp-content/uploads/2014/12/How-to-scan-QR-code-in-your-Smartphone.jpg
https://img1.etsystatic.com/036/0/9343025/il_fullxfull.654477583_8ktu.jpg
http://cdn1.tnwcdn.com/wp-content/blogs.dir/1/files/2015/01/mark-zuckerberg-qa-colombia.png
https://lastpass.com/press-room/
http://66.media.tumblr.com/d19d0b84160d51e696aeaa939b84f4de/tumblr*ns7wyq9uVl1qhub34o10*r1_500.gif