
SBA
Research

COMET

Competence Centers for Excellent Technologies

Protect Your User Accounts Like It's 2019

Thomas Konrad, SBA Research

sec4dev, Feb 27th, 2019

Bundesministerium Verkehr, Innovation und Technologie

Bundesministerium Digitalisierung und Wirtschaftsstandort

FFG Forschung wirkt.

wirtschaftsagentur wien Ein Fonds der Stadt Wien

FWF Der Wissenschaftsfonds.

netidee OPEN INNOVATIONS

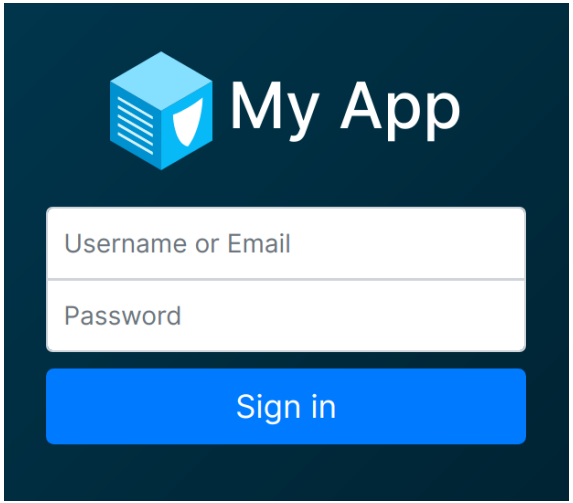
```
$ whoami
Thomas Konrad
$ id
uid=123(tom)
gid=0(SBA Research)
gid=1(Software Security)
gid=2(Penetration Testing)
gid=3(Software Development)
gid=4(Security Training)
```

Why Are We Here?

The starting point.

Some Assumptions

1. We're developing an **application** (web or mobile)
2. The application has **user accounts**
3. The processed data is at least somewhat **sensitive**
4. We have a **hard** time coming up with proper **account security solutions**



What Do We Want To Protect Against?

Motivations for account security measures

What do we strive for?

Image source: https://i2.wp.com/www.thesecurityawarenesscompany.com/wp-content/uploads/2015/05/CIAtriad-copy2.png?zoom=2.625&fit=2702%2C2448&ssl=1

A Basic Threat Model

Threat	Severity[1]	C/I/A	Countermeasures
Password guessing	**High**	C/I/-	?
Account lockout	**Medium**	-/-/A	?
Misuse of known passwords (public lists, other apps, ...)	**Medium**	C/I/-	?
Someone dumps the DB on the Internet	**Medium**	C/I/-	?
Enumerating valid user names	**Low**	C/-/-	?

[1] *The severity really depends on the classification of your data. Don't see them as absolute and unchangeable values.*

The Game Is On: Account Security Hammer Head

Image source: https://i.ytimg.com/vi/X4pSjBqbE0Y/maxresdefault.jpg

The C/I And The A

- In (account) security, these two are often **contradictory**
- Often, we have to find a **balance**
- **Password guessing** (C/I) vs. **account lockout** (A)

Thou Shalt Not Pass!

Protecting against credentials guessing attacks, a.k.a. “achieving the **C** and the **I**”

What Can We Do For The C And The A?

Gimme some ideas!

Image source: https://giphy.com/gifs/WilsonTennis-karen-khachanov-1o1iwQngXCF5guav2G

Password Policy

- NIST 800-63-3 Digital Identity Guidelines
- Major overhaul in June 2017

Password Policy

- **Current recommendations in a nutshell**
  - Nuke periodic changes 

  - Nuke complexity rules 

    - Only require a minimum length
  - **Screen passwords against compromised passwords**
  - Maximum length must be at least 64 cars 

  - Require MFA for high-privilege accounts

Image source: http://37wz5x2r8vbh3om46wmfhy71-wpengine.netdna-ssl.com/wp-content/uploads/2016/10/yeah-600px.png

Password Policy: Tips By NIST

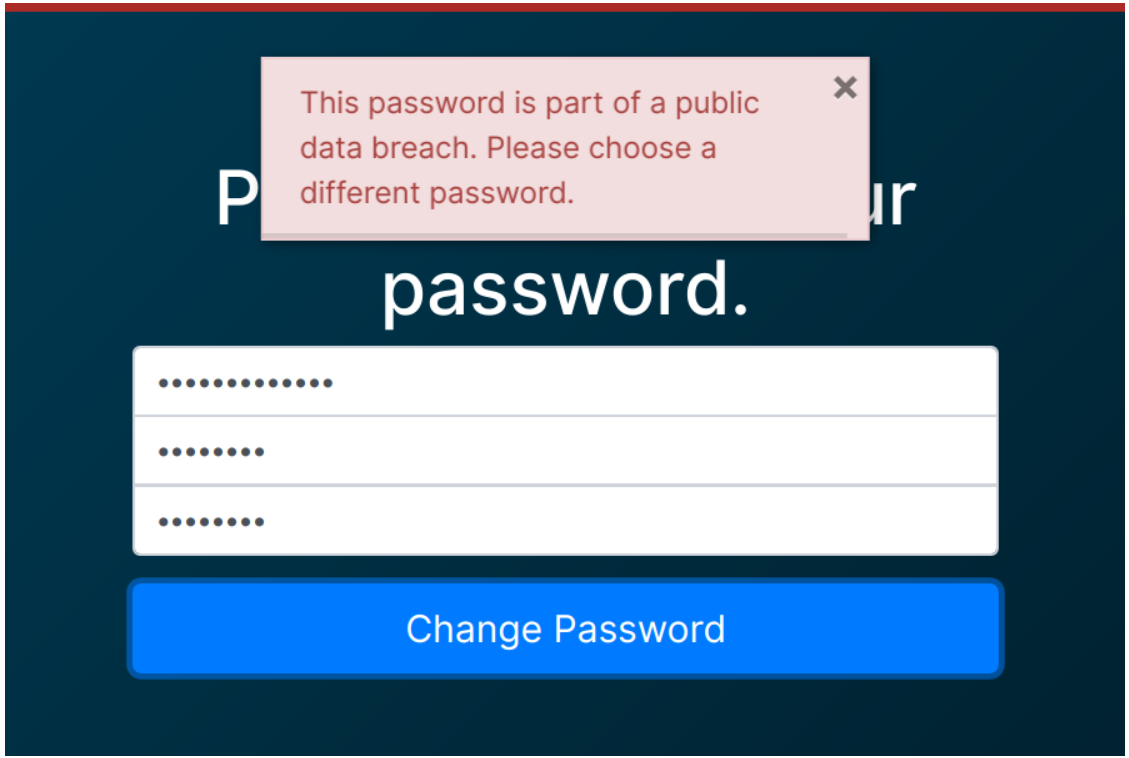
Password Policy: Screen Passwords

- Troy Hunt's Pwned Passwords API: https://haveibeenpwned.com/API/v2

```
GET https://api.pwnedpasswords.com/range/{first 5 hash chars}
```

```
0018A45C4D1DEF81644B54AB7F969B88D65:1
00D4F6E8FA6EECAD2A3AA415EEC418D38EC:2
011053FD0102E94D6AE2F8B83D76FAF94F6:1
012A7CA357541F0AC487871FEEC1891C49C:2
0136E006E24E7D152139815FB0FC6A50B15:2 ...
```

Password Policy: Screen Passwords



Use Proper Hashes

- Hash algorithms are designed to be **fast**
- If our DB gets breached, we wish they'd be **slow**
- **So how should we persist passwords?**
  - Use an algorithm that intentionally makes brute-force attacks slow
  - If you have the choice, use **Argon2**
  - **bcrypt** is ok, but has some pitfalls
  - **PBKDF2** is also ok

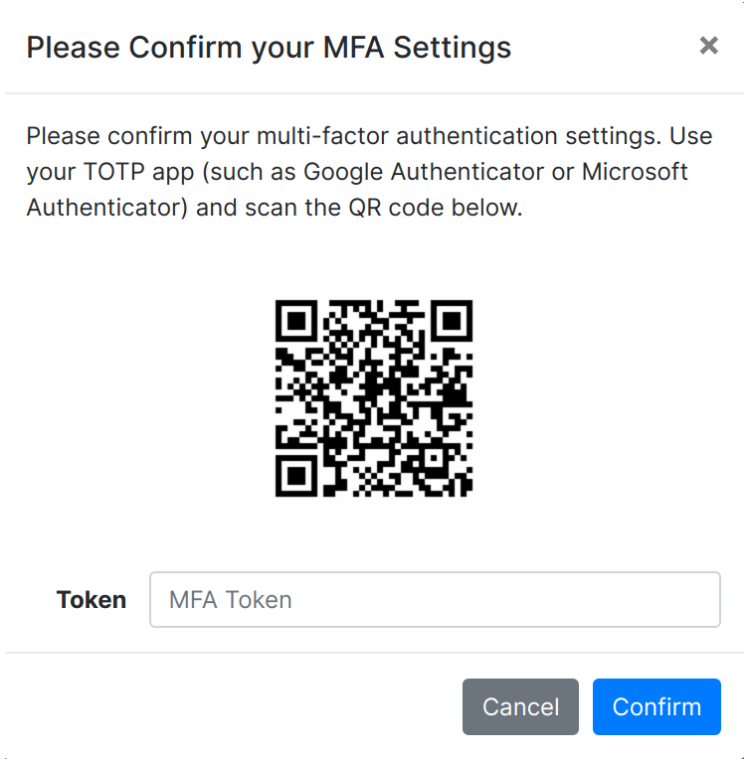
Lock Users After Too Many Failed Attempts

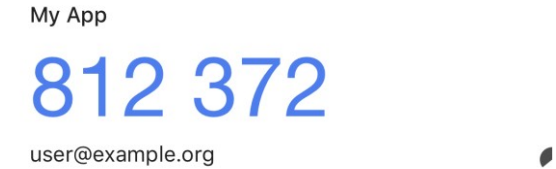
- **Soft lock** vs. **hard lock**
  - Soft == temporarily
  - Hard == permanently
  - **Have both implemented!**
- I'd generally recommend a **soft lock**
  - E.g., lock for 5 minutes after 5 wrong attempts
- But it really **depends on the C/I requirements**

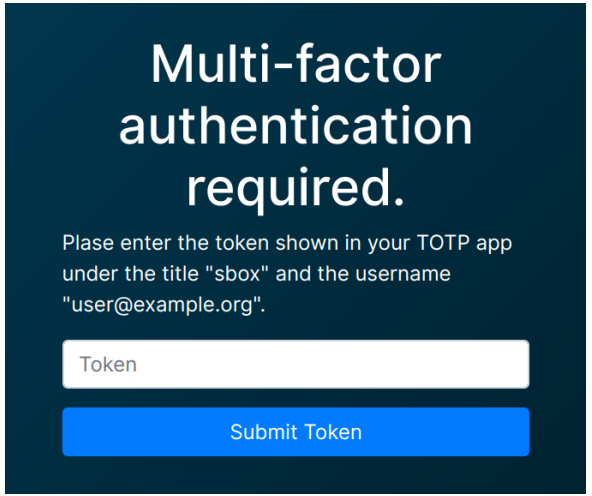
Multi-Factor Authentication

- Require not just user name and password, but also something else (optional, but mandatory for admins)
- Even if the credentials are breached, the attackers cannot log in
- **Some examples**
  - A one-time password sent via SMS
  - A TOTP app (e.g., Google Authenticator)
  - A hardware token
  - U2F

Multi-Factor Authentication: TOTP



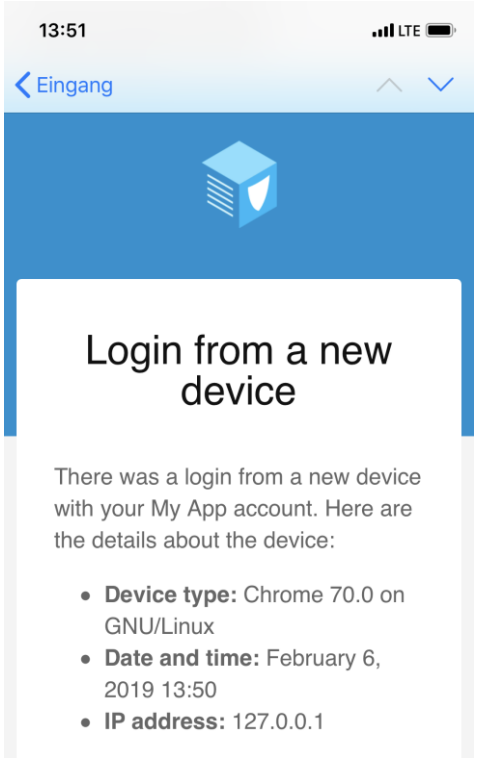




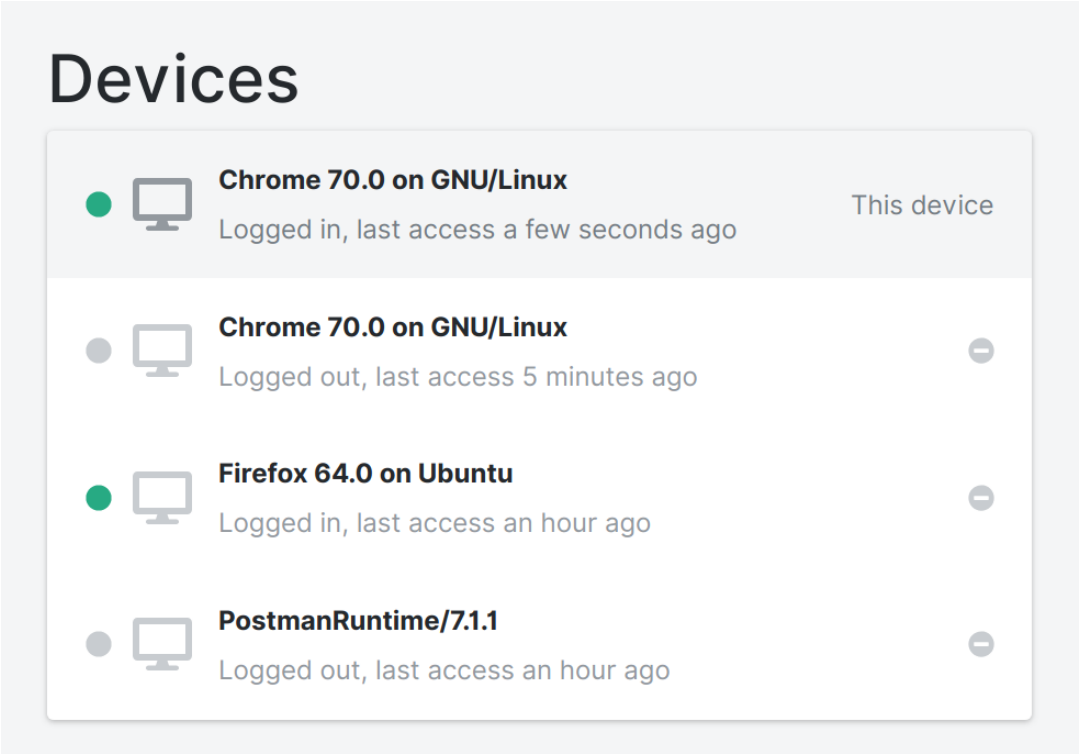
Transparency

- Even if we do all we can, there might still be malicious activity
- **If fishy things happen, we at least want …**
  - … the user to know it.
  - … have the ability to react on it.

Transparency: Notifications



Transparency: Device List



Transparency: How Do We Track Devices?

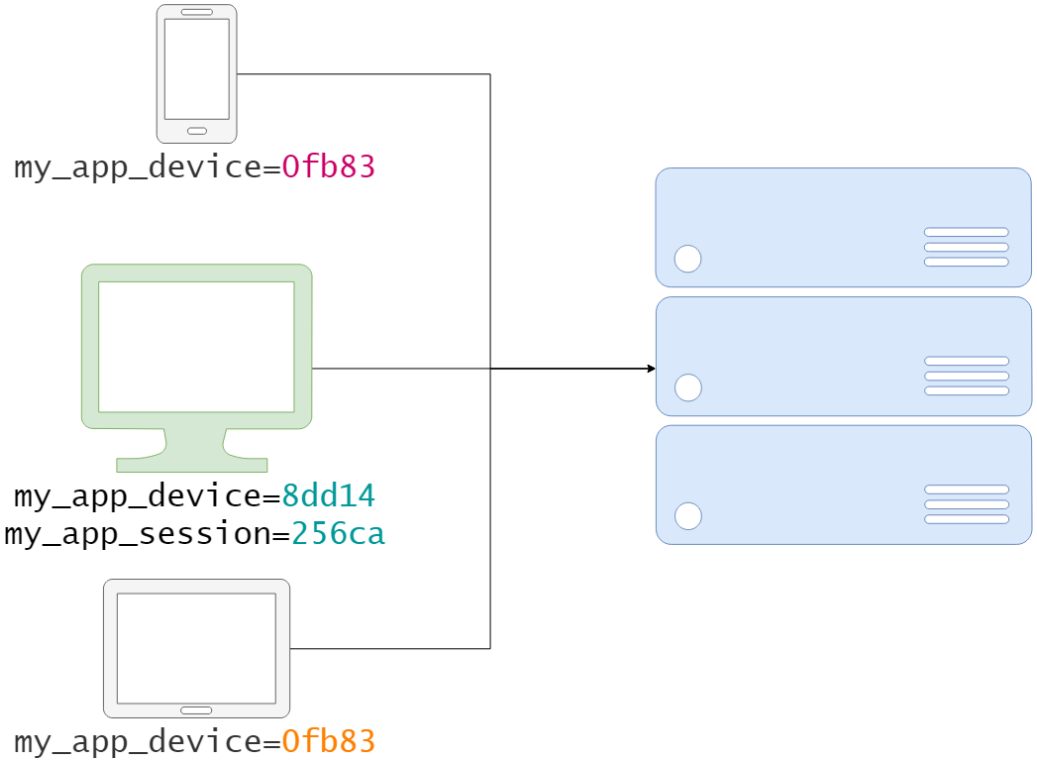
- With **Device Tokens** (Device Cookies)!

Name	Value
my_app_device	9d5f235ee236ab90dcb884d001…
my_app_session	pcippce933n26mrs2bqvp5fdr0

Detour: Device Tokens

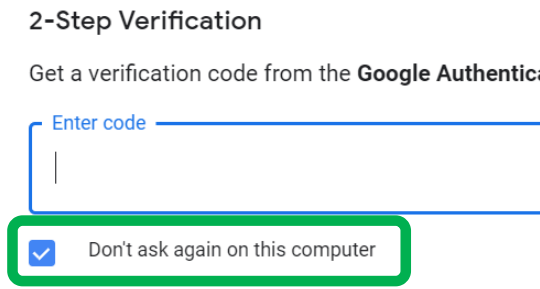
- **Device Tokens in a nutshell**
  - Catch *successful* login events
  - If this is a new device
    - Issue a Device Token
    - Send a notification (as you saw before)
  - The cookie (token) must be long-running
  - Connect the new session to it
  - Store source IP, user agent, first access, last access

Detour: Device Tokens

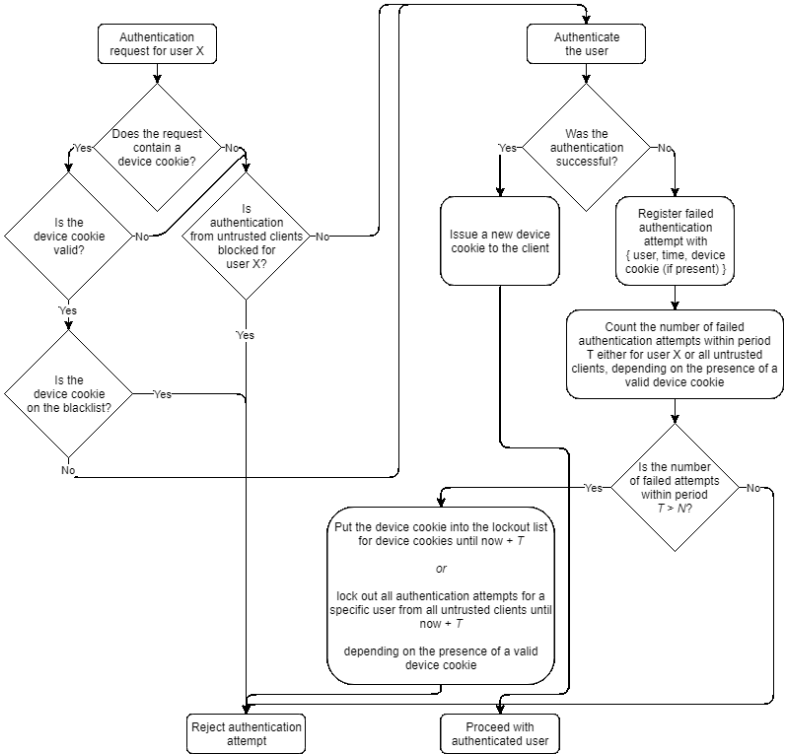


Detour: Device Tokens

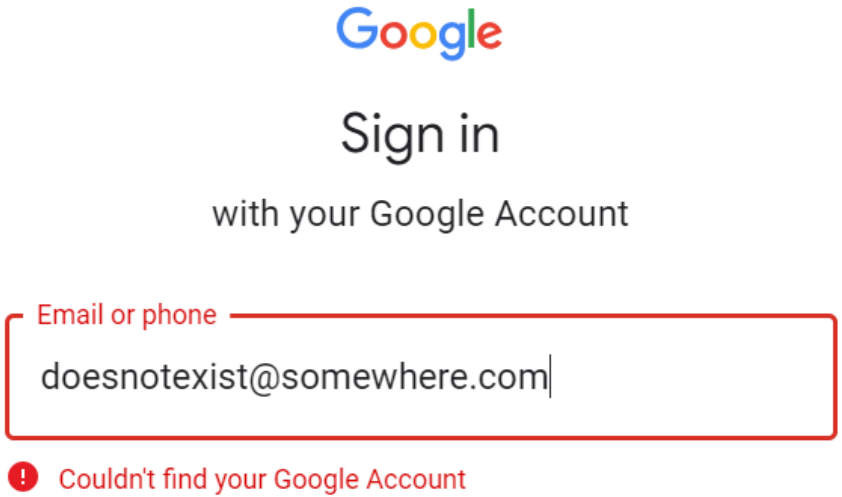
- **Device Tokens enable us to do tons of good things**
  - List devices (transparency, remember?)
  - Notifications upon a login from a new device (transparency, remember?)
  - Remember MFA for specific devices
  - Remember previously logged-in users
  - Slow down password guessing (you'll see later)
  - …
- **They are very helpful for good account security!**



Detour: Device Tokens



User Enumeration



User Enumeration

Google

Create your Google Account

First name: Somebody

Last name: Somewhere

Username: thomas.konrad@gmail.com

That username is taken. Try another.

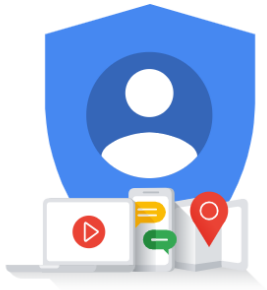
Available:

somewheres758 somewheresomebody677
somebodysomewhere05

Use my current email address instead

Password: ••••••••••••••••••

Confirm: ••••••••••••••••••



One account. All of Google working for you.

User Enumeration

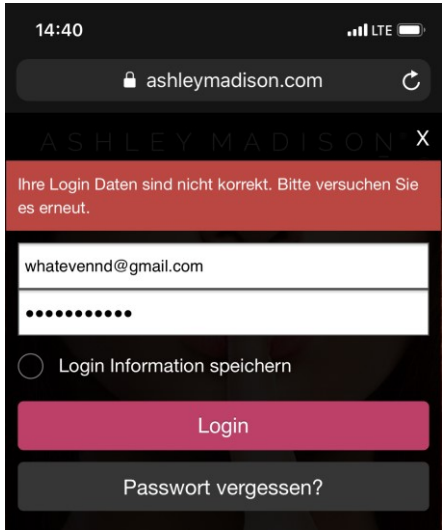
- ***Actually* protecting against user enumeration is *really hard***
  - Login form?
  - Login form after a user lockout?
  - Registration form?
  - Password reset?
  - Timing differences?
  - Other services that use the same user DB?
  - …

User Enumeration

- **It's mostly about error messages with really bad usability**
  - "Username and/or password wrong."
  - "Invalid credentials. Note that you might get locked out after too many failed login attempts."
  - "Password reset request received. If that account exists, you should have gotten an email with further instructions."

User Enumeration

- Again, this depends on the **sensitivity of the fact that somebody is registered**.
- Think "Ashley Madison vs. The Recipe Collection"
- **Sometimes, it is ok to accept the risk of user enumeration**



What Can We Do For The C And The I?

1. Use a good password policy
2. User proper hashes
3. Lock out users (hard lock vs. soft lock)
4. Multi-factor authentication (MFA)
5. Transparency (device lists, notifications)
6. Protect against user enumeration (?)

Thou Shalt Not Lock!

Keeping attackers from systematically locking out users, a.k.a. “achieving the **A**”

Preventing User Lockout

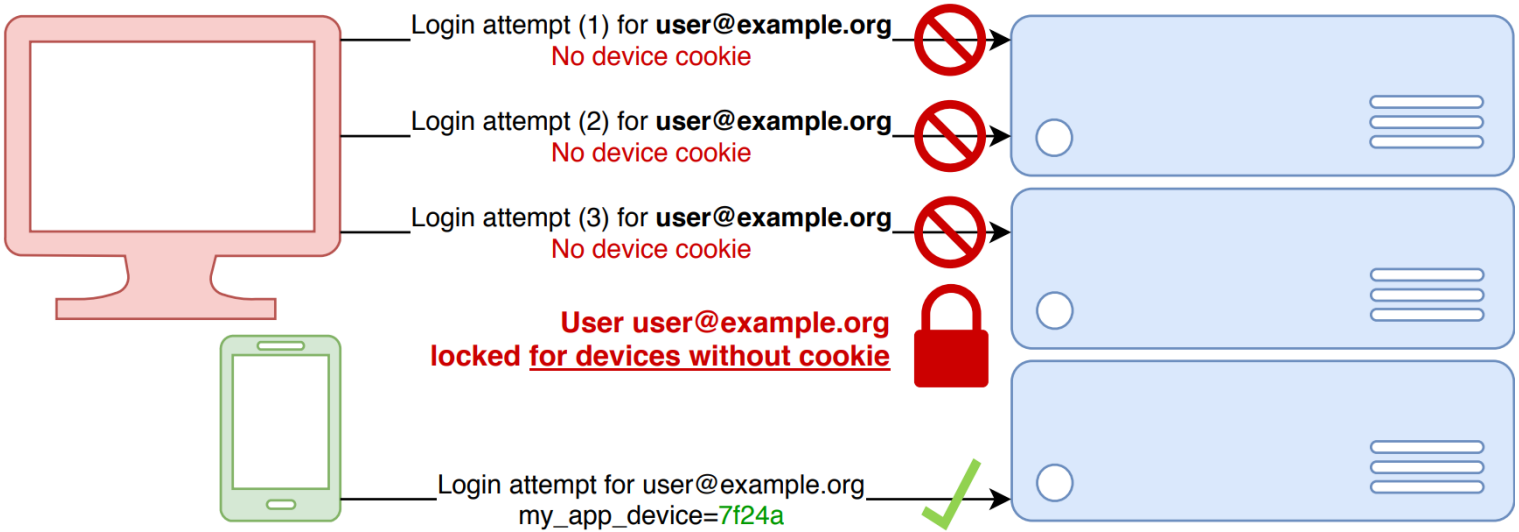
- **This is the harder part!**
- Remember the Hammer Head?

https://giphy.com/gifs/cuteness-Hnv3oVMOkmHiE

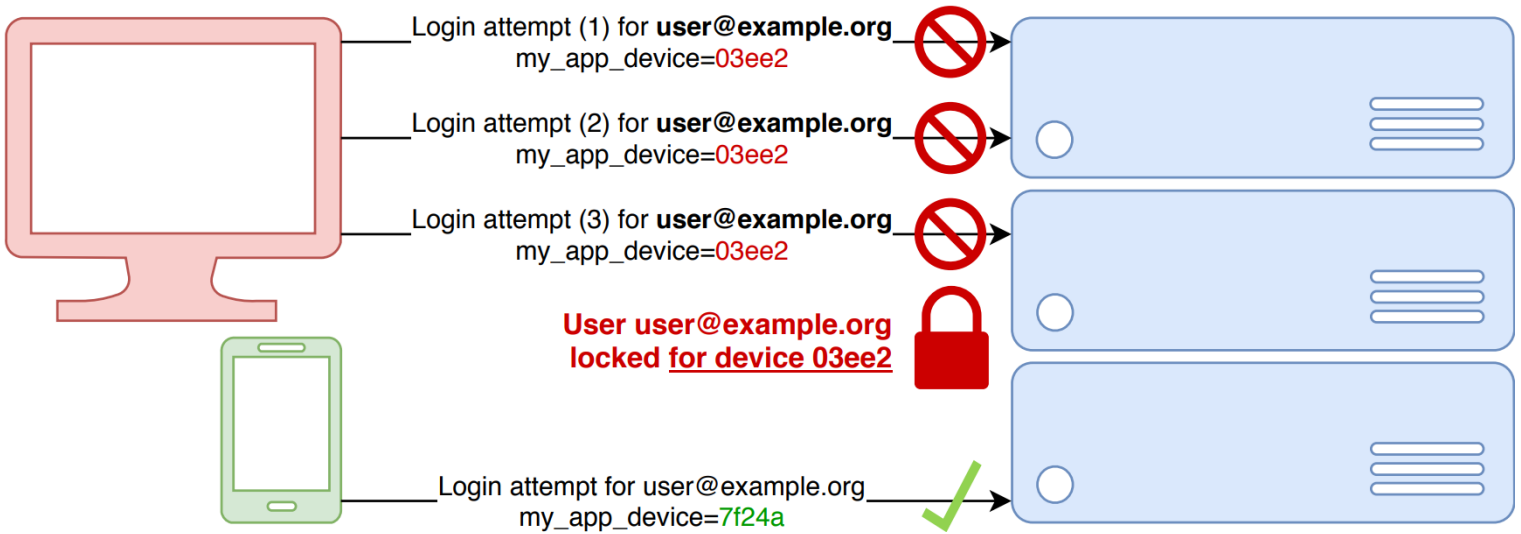
Preventing User Lockout: A Question Of Trust

Image source: https://www.supermarketguru.com/site/assets/files/6521/bakerycounter.jpg

Preventing User Lockout: A Question Of Trust



Preventing User Lockout: A Question Of Trust



Preventing User Lockout: The Pareto Principle

- You can **save most users** from being **locked out**
- But not 100 %!
- A note for apps with **public registration forms**
  - An attacker could register and issue themselves new device token via a script
  - Therefore: Count failed login attempts also for users and hard-lock them in case they’re attacking

Let's Update the Threat Model

What do all the countermeasures mean to our model?

A Basic Threat Model

Threat	Severity[1]	C/I/A	Countermeasures
Password guessing	**High**	C/I/-	(Temporary) user lockout, password policy, MFA, transparency (device lists and notifications, with Device Tokens)
Account lockout	**Medium**	-/-/A	Selective lockout (with Device Tokens)
Misuse of known passwords (public lists, other apps, ...)	**Medium**	C/I/-	MFA
Someone dumps the DB on the Internet	**Medium**	C/I/-	Proper hashes (Argon2)
Enumerating valid user names	**Low**	C/-/-	(Generic error messages, constant timing on all requests containing the user name)

[1] *The severity really depends on the classification of your data. Don't see them as absolute and unchangeable values.*

Advanced Countermeasures

Where to go from here

Advanced Countermeasures

- Geo IP blocking
- Heuristics
- Conditional CAPTCHA
- Conditional MFA
- …
- **A lot more can be cone, but think about taking this as a starting point!**

Account Security For End Users

How to minimize the risk of a data breach for yourself

Account Security For End Users

1. Use a password manager (not the browsers')
2. Don't re-use passwords
3. Turn on MFA where possible
4. Register on https://haveibeenpwned.com

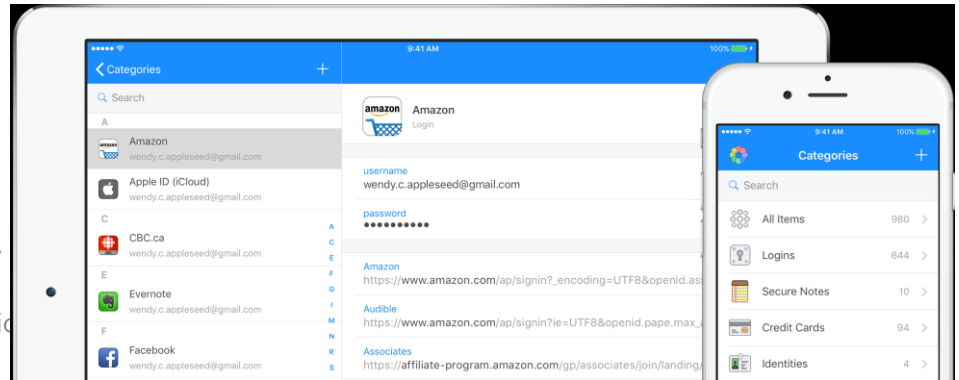


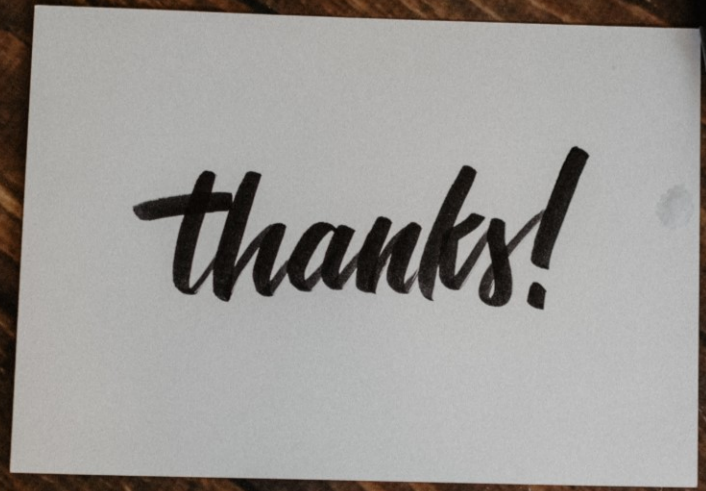
Image source: https://i.1password.com/media/ios-hero.png

Summing Up

Things to do for better account security

Summary

1. Define the **CIA requirements** for your data!
2. Do *your* **Threat Model**
3. Implement **Device Tokens**
4. Protect the **C/I** (password policy, proper hashes, lock users, (optional) MFA, transparency)
5. Protect the **A** (selective lock-out with Device Trokens)



Thomas Konrad

SBA Research gGmbH
Favoritenstraße 16, 1040 Vienna
tkonrad@sba-research.org
@_thomaskonrad

Photo by Kelly Sikkema on Unsplash

sec4dev

- Thank you all for being here!
- **Save The Date: 24 to 27 Feb 2020**
- **Let's build a community!**
  - Security Meetup by SBA Research (meetup.com)
  - Twitter: @sec4dev
- Please spread the world!


Thanks!
sec4dev