Safe Enough

Balancing Security and Usability

Natasha Eibich

M.Sc. Human Computer Interaction and Design | University of Twente, NL

M.Sc. (tech) Human Computer Interaction and Design (Advanced User Modelling) | Aalto University, Finland

UX Analyst and Designer at Clario Medical | Intelerad Medical Systems

Not a dev! :O

We can balance **Security** and **Usability** by understanding users, their intentions, and their interaction patterns.

Understanding our Users

Understanding Users

Equifax, US Personnel Management,
DNC - But it could never happen to me

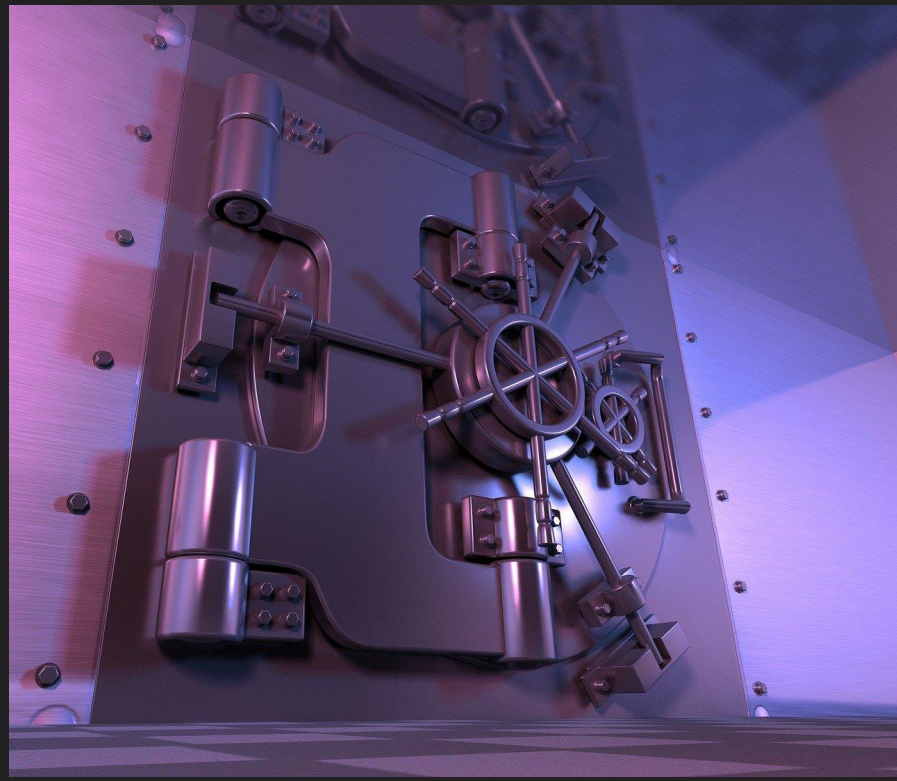
Implications:
Keys > Password
Lock a door > Lock a computer
Passport > Online Banking

Thomas/flickr

Understanding Users

Physical Security is Embodied

Implications:
Real world Security is Zuhanden
Digital Security is Vorhanden



8385/pixabay

Understanding Users

Security and Privacy is Social

Implications:
Security = Paranoia
Public/Private
Share vs. Keep to myself

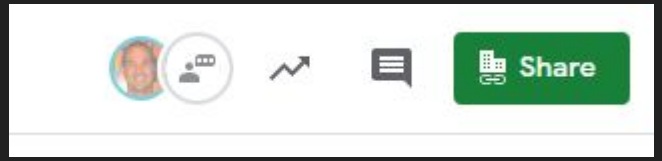
8385/pixabay

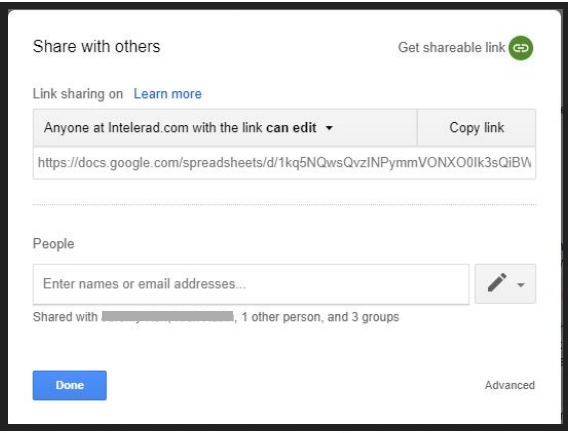
Understanding Users

Not understanding how or why something works makes it harder to use

Implications:
Strict password requirements, expirations, MFA
Why? What makes this more secure?

Visibility AND Transparency
Clear, Succinct answers outline the problem and the solution





Many policies require a minimum password length. Eight characters is typical but may not be appropriate.[2][3][4] Longer passwords are generally more secure, but some systems impose a maximum length for compatibility with legacy systems.

Some policies suggest or impose requirements on what type of password a user can choose, such as:

- the use of both upper-case and lower-case letters (case sensitivity)
- inclusion of one or more numerical digits
- inclusion of special characters, such as @, #, $
- prohibition of words found in a password blacklist
- prohibition of words found in the user's personal information
- prohibition of use of company name or an abbreviation
- prohibition of passwords that match the format of calendar dates, license plate numbers, telephone numbers, or other common numbers

Other systems create the password for the users or let the user select one of a limited number of displayed choices.

https://en.wikipedia.org/wiki/Password_policy

Understanding Users

Users are task oriented

Implications:
Every interaction has a goal

Security should:
Support the goal
Facilitate the goal
Get out of the way
Be Contextual

Stocksnap/pixabay

Shoring up the weakest link

I’m making the assumption that you are going to use best practices for securing user information on the backend, making the user and their log in the weakest link.

Don’t collect data

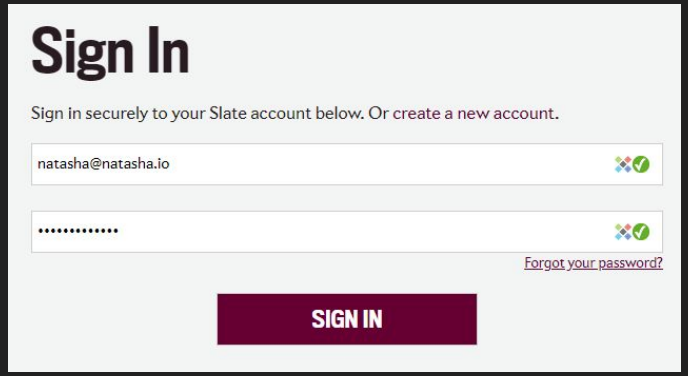
Don’t need it, don’t worry bout it

Pros

- Users don’t need to log in
- don’t need to think about their data
- Just fluid interactions

Cons

- No user profiles to run with



The Classic

Log in, Password, simple right?

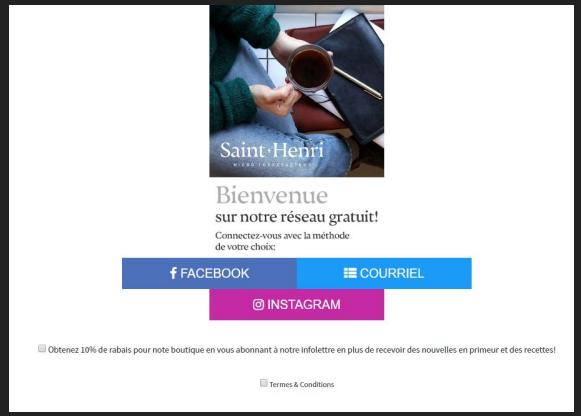
Pros

- Users know the drill
- It's two pieces of known info
- Uses technology (a keyboard) users know

Cons

- Crazy requirements that change every time
- Needing to update passwords every 3 months
- Need to remember dozens

Slate.com



The Social

Log me in with… [Google, Facebook, Microsoft]

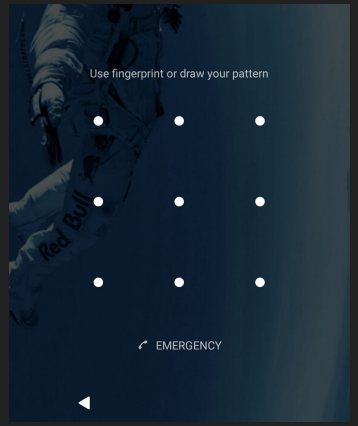
Pros

- Fast log in
- No remembering passwords
- User is probably already set up for this

Cons

- You’re using someone else’s authentication
- Lose user privacy

Café St. Henri, Montreal



The Artist

The pattern you’re drawing on your phone rn

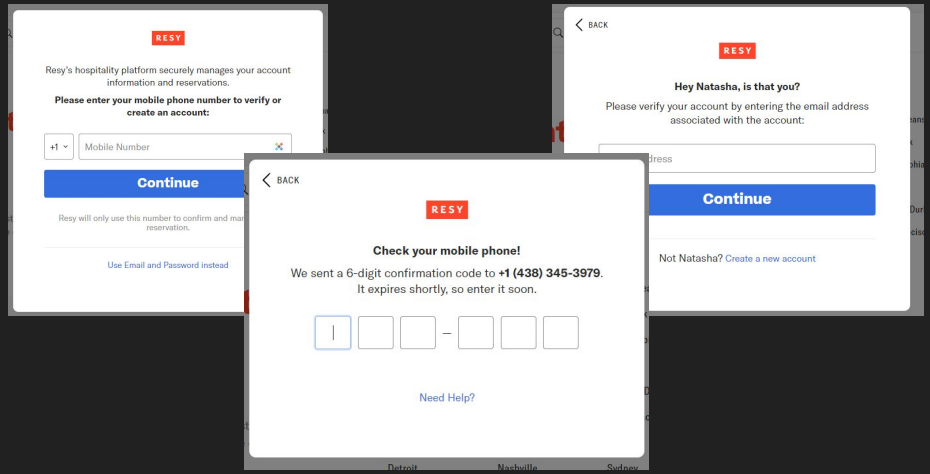
Pros

- Quick, easy to remember

Cons

- Can be too simple
- Smudges can betray the pattern

Café St. Henri, Montreal



S-M-S

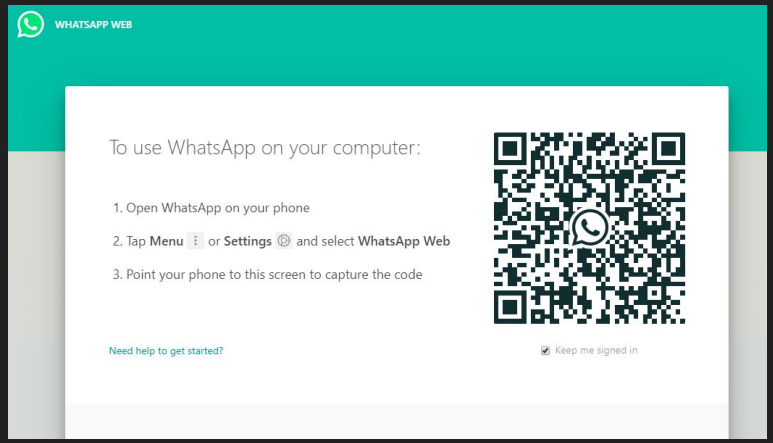
Ask for my phone number and text me a code to log in

Pros

- No password or log in to remember

Cons

- Can be slow
- Only as secure as the phone number



Scan It

App logged in, scan QR code to log in the website

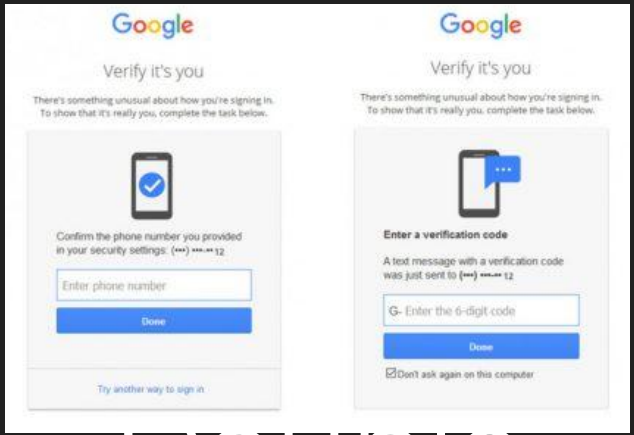
Pros

- Only need to remember app log in
- Quick authentication

Cons

- Need to be able to scan the QR code
- It’s a backwards kind of 2FA
- WhatsApp, Abn Amro

Web.whatsapp.com



Double Stuffed

Any variation of 2 factor authentication

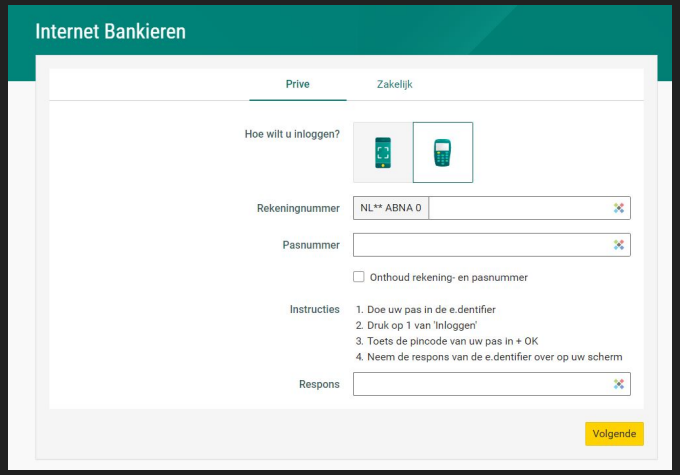
Pros

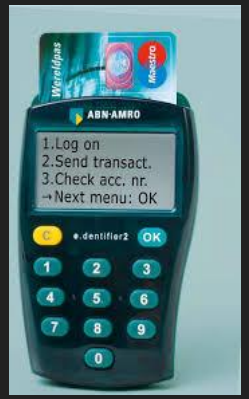
- More secure
- Effective protection against bad passwords

Cons

- Easy to forget one of the factors
- Takes more time
- Users might pick less secure passwords cause they trust 2nd
- Frustration building
- Users ‘trust’ a lot of devices so don’t get asked
- Really need more than one option

Google.com





Triple Stuffed

Anything more than 2FA

ABN Amro, Netherlands

Pros

- More secure
- Effective protection against bad passwords

Cons

- So much time, effort
- Easy to forget one of the pieces of the puzzle

Hijacking Habits

Habits are Powerful

Common tendencies that are hard to break

The way you get up in the morning
How and when you head home after work/school
The way you prep/take your coffee

They are subtle

The way you type (full home row, partial home row, other?)
Where you look on your computer for the toolbar (Mac/PC/Linux)
Where you expect your save/cancel buttons to be

Mastersenaiper/pixabay

Hijack all of the Habits

Make use of the habits you find with your users

- Pull from real life and existing UI

  Expectations influence perception, and matching them speeds users up

- Keep your interface consistent with best practices

  Prototypicality - cuts down on frustration and speeds up perception

- Keep your interactions consistent with others

  Cognitive Chunking – limit the number of interactions on one screen
  Low Visual Complexity – helps users determine information hierarchy and speeds up perception

HartmutStein/pixabay

Building Usable Security

01 Security from day one

Involve security in your system design as well as your interface design from the first draft - more than just “we need a log in page”

Questions to Ask yourself:

1. Do users need to log in to your system?
2. How are you going to protect your user’s data from outside attackers?
3. How will you protect your users from their own bad security habits?
4. What are your options?

02 Define data requirements and assess risk

How much personal data do you really need? Less data = less headaches. The aim is to protect user privacy

Questions to Ask yourself:

1. See the “Privacy by Design” talk by Marcus Bointon
2. How much personal and protected information do you need to accomplish the goals of your users?
3. Is the required data protected under some kind of regulation or legislation that will dictate security procedures?
4. How much of the system security relies on the users?

03 Research

Research your target users, research your competitors, research security options, more research = more options

Questions to Ask yourself:

1. Get to know the context in which you expect your users to interact with your system

2. What are the tasks that your solution is helping users accomplish and when/where/how are they accomplishing those tasks now

3. Who are your users? Who are they really- not just some persona a UX consultant slapped together for your or you found online

4. What are your users attitudes toward security? In this context? In any context? Do they care? Should they care? Just ask a bunch of questions!

5. What are your regulatory requirements? Privacy? Consent?

04 Design

Before you commit to any UI, design and iterate through various interactions

Questions to Ask yourself:

1. Design the dev stuff- I don't know what kind of system design you want to have, not my area

2. Figure out the sign in options available to you

3. Review the habits of your users, can you hijack any of them? Do their tasks lend themselves to one log in solution over another?

4. Pick one and go…. Test it.

Test everything with your users

Questions to Ask yourself:

1. Test early and test often

2. Test on whatever kinds of prototypes you might have- you can even test with other products that people might already have

3. Test with real scenarios- or better- test in real contexts with real users.

4. Make changes based on the feedback: change the text, change the order of inputs, change how much information you display. And test again

Build your solution

06
Build

Knowing your users will understand:

Why they need your security procedures,

How your security options keep them safe,

What security they are responsible for

Thank you.

Get more Technical:

Passwords are so 1990 (Feb 27 @13:00)

The Future of Authentication (Feb 27 @ 14:00)

Privacy in the Age of Analytica (Feb 27 @ 16:00)

Continuous Security (Feb 28 @ 10:00)

Supercharge Appsec (Feb 28 @13:00)

Also Check out: “Privacy by Design” talk from Marcus Bointon

Sources:

Informal Observations of clients, co-workers, and friends
Research for some course work (MHCID)
Paul Dourish works
Heidegger and Merleau-Ponty
Usability and Security - Lorrie Faith Cranor, Simson Garfinkel - 2005
Usable Security - Coursera Course, University of Maryland
http://techgenix.com/security-vs-usability/
https://whatis.techtarget.com/definition/security-by-design
https://www.ncsc.gov.uk/blog-post/security-and-usability--you-can-have-it-all-
https://discovery.ucl.ac.uk/id/eprint/20247/2/CACM%20FINAL.pdf
https://dl.acm.org/doi/book/10.5555/1098730
https://duo.com/blog/part-1-usability-is-security
https://security.googleblog.com/2017/11/new-research-understanding-root-cause.html
https://blogs.dropbox.com/tech/2012/04/zxcvbn-realistic-password-strength-estimation/
https://www.javelinstrategy.com/press-release/identity-fraud-hits-all-time-high-167-million-us-victims-2017-according-new-javelin
https://www.nngroup.com/articles/security-and-human-factors/
https://www.telegraph.co.uk/connect/better-business/security-versus-usability-ux-debate/
https://medium.com/mycrypto/the-impossible-balance-between-usability-security-55000c9fc46d
https://uxdesign.cc/three-surprising-lessons-ux-can-learn-from-phenomenology-d07882e354d0
https://www.cc.gatech.edu/~keith/pubs/ieee-intro-usable-security.pdf
https://link.springer.com/chapter/10.1007/978-3-319-07620-1_7
https://discovery.ucl.ac.uk/id/eprint/20345/2/cransimpsonbook.pdf
https://www.cs.virginia.edu/~evans/cs551/saltzer/
https://www.csoonline.com/article/3213568/john-mcafee-ranks-the-10-biggest-hacks-ever.html