Security and human behavior

Some material from Lorrie Cranor, Mike Reiter, Rob Reeder, Blase Ur
In these lectures ...

• Overview

• Minimizing effort

• Case studies
  – Grey, password expiration, security images, password meters, implantable devices

• Making things better
Humans

“Humans are incapable of securely storing high-quality cryptographic keys, and they have unacceptable speed and accuracy when performing cryptographic operations… But they are sufficiently pervasive that we must design our protocols around their limitations.”

More on humans

“Not long ago, [I] received an e-mail purporting to be from [my] bank. It looked perfectly legitimate, and asked [me] to verify some information. [I] started to follow the instructions, but then realized this might not be such a good idea … [I] definitely should have known better.”

-- former FBI Director Robert Mueller
And one more …

“I think privacy is actually overvalued … If someone drained my cell phone, they would find a picture of my cat, some phone numbers, some email addresses, some email text. What’s the big deal?”

-- Judge Richard Posner

U.S. Court of Appeals, 7th circuit
Better together

Examining security/privacy and usability together is often critical for achieving either
The human threat

• Malicious humans
• Humans who don’t know what to do
• Unmotivated humans
• Humans with human limitations
Key challenges

• Security is a **secondary task**
  – Users are trying to get something else done
• Security concepts are **hard**
  – Viruses, certificates, SSL, encryption, phishing
• Human capabilities are **limited**
Are you capable of remembering a unique strong password for every account you have?
Key challenges

• Security is a secondary task
• Security concepts are hard
• Human capabilities are limited
• Misaligned priorities
Keep the bad guys out!

Don’t lock me out!

Security Expert

User
Key challenges

• Security is a secondary task
• Security concepts are hard
• Human capabilities are limited
• Misaligned priorities
• Active adversaries
  – Unlike ordinary UX
What is Twitter?

Twitter is a service for friends, family, and co-workers to communicate and stay connected through the exchange of quick, frequent answers to one simple question: What are you doing?
Key challenges

- Security is a **secondary task**
- Security concepts are **hard**
- Human capabilities are **limited**
- Misaligned **priorities**
- Active adversaries
  - Unlike ordinary UX
- Habituation
  - The “crying wolf” problem
KEY CHALLENGE EXAMPLE:

HABITUATION
Exercise: Draw a penny

• Draw a circle

• Sketch the layout of the four basic items on the front of a US penny
  – What are the items, and how are they positioned?

• Hint:
  – Someone’s portrait (who?)
  – Two patriotic phrases
  – Another item
  – Extra credit: an item that some pennies have and some don’t
Score your sketch

• Score:
  – 1 for Abraham Lincoln
  – +1 for Abraham Lincoln facing right
  – +1 for “Liberty”
  – +1 for “Liberty” to Abe’s left
  – +1 for “In God We Trust”
  – +1 for “In God We Trust” over Abe’s head
  – +1 for the year
  – +1 for the year to Abe’s right
  – Extra credit: +1 for the mint letter under the year
  – -1 for every other item
Lessons from Abe

• You’ve probably seen hundreds of pennies
  – And yet, this is hard

• Memory limitations
  – Remembering a penny isn’t important, unless you take this quiz!

• Habituation
  – You see it so often, you don’t remember it anymore
Habituation to warnings

- Microsoft Internet Explorer:
  - You are about to open:
    - Name: ..\dule - 2009 05 04.pptx
    - From: windows
  - How would you like to open this file?
    - Read Only
    - Edit

- Windows Security Center:
  - You are about to run a program that did not come with Windows. Do you trust this program?
  - Publisher: Microsoft Corporation
  - Program: Microsoft Forefront System
  - C:\Program Files\Microsoft Forefront\Forefront System\Client\UXF\Sys\ClientUtil.exe

- Explorer:
  - This page is accessing information that is not under its control. This poses a security risk. Do you want to continue?
  - Yes
  - No
Something happened and you need to click OK to get on with doing things.

Certificate mismatch security identification administrator communication intercept liliputian snotweasel foxtrot omegaforce.
If it’s important, make it stand out

SSL warning; risk low; yellow background

Malware warning; risk very high; red background
MINIMIZING EFFORT
People are economical

• Given two paths to a goal, they’ll take the shorter path

• More steps = less likely they’ll be completed

• Can they figure out what to do?
  – Too hard = give up and take easiest path
You should only open attachments from a trustworthy source.

Attachment: TUX Scope Framing and Ownership 091211b.pptx from Inbox - Microsoft Outlook

Would you like to open the file or save it to your computer?

[Open] [Save] [Cancel]

☑ Always ask before opening this type of file
You should only open attachments from a trustworthy source.

Attachment: TUX Scope Framing and Ownership 091211b.pptx from Inbox - Microsoft Outlook

Would you like to open the file or save it to your computer?

- Open
- Save
- Cancel

Always ask before opening this type of file
What’s the source of this attachment?
What’s the source of this attachment?

What makes a source trustworthy or not trustworthy?
You should only open attachments from a trustworthy source.

Attachment: TUX Scope Framing and Ownership 091211b.pptx from Inbox - Microsoft Outlook

Would you like to open the file or save it to your computer?

Open  Save  Cancel

Always ask before opening this type of file

What’s the source of this attachment?

What makes a source trustworthy or not trustworthy?

What will happen if I don’t follow this advice?
What's the source of this attachment?

What makes a source trustworthy or not trustworthy?

What will happen if I don't follow this advice?

Does this mean that opening is dangerous but saving is safe?
What’s the source of this attachment?

What makes a source trustworthy or not trustworthy?

What will happen if I don’t follow this advice?

What steps can I take to decide what to do?

Does this mean that opening is dangerous but saving is safe?
Good security practices that people don’t do

• Install anti-virus software
• Keep your OS and applications up-to-date
• Change your passwords frequently *
• Read a website’s privacy policy before using it
• Regularly check accounts for unusual activity
• Pay attention to the URL of a website
• Research software’s reputation before installing
• Enable your software firewall
• Make regular backups of your data
• Read EULAs before installing software
What can go wrong when you don’t consider human factors

CASE STUDIES
GREY AND USER BUY-IN
Is Grey too slow?

• Grey: Smartphone-based access control
  – (Covered earlier in the semester)
  – Strong security benefits vs. keys
• Users complained about speed
  – Videotaped a door to measure vs. keys

[Bauer et. al, SOUPS 2007]
Average access times

Grey is not noticeably slower than keys!

Getting keys → Stop in front of door → Door opened → Door closed
3.6 sec $\sigma = 3.1$
5.4 sec $\sigma = 3.1$
5.7 sec $\sigma = 3.6$
Total 14.7 sec $\sigma = 5.6$

Getting phone → Stop in front of door → Door opened → Door closed
8.4 sec $\sigma = 2.8$
2.9 sec $\sigma = 1.5$
3.8 sec $\sigma = 1.1$
Total 15.1 sec $\sigma = 3.9$
“I find myself standing outside and everybody inside is looking at me standing outside while I am trying to futz with my phone and open the stupid door.”

Takeaway: Misaligned priorities
PASSWORD EXPIRATION AND USER BEHAVIOR
Does password expiration improve security in practice?

• Observation
  – Users often respond to password expiration by transforming their previous passwords in small ways [Adams & Sasse 99]

• Conjecture
  – Attackers can exploit the similarity of passwords in the same account to predict the future password based on the old ones [Zhang et. al, CCS 2010]
Empirical analysis

• UNC “Onyen” logins
  – Broadly used by campus and hospital personnel
  – Password change required every 3 months
  – No repetition within 1 year

• 51141 unsalted hashes, 10374 defunct accounts
  – 4 to 15 hashes per account in temporal order

• Cracked ~8k accounts, 8 months, standard tools

• Experimental set: 7752 accounts
  – At least one cracked password, NOT the last one
Transform Trees

- Approximation algorithm for optimal tree searching
Location Independent Transforms

<table>
<thead>
<tr>
<th>CATEGORY</th>
<th>EXAMPLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capitalization</td>
<td>tarheels#1 → tArheels#1</td>
</tr>
<tr>
<td>Deletion</td>
<td>tarheels#1 → tarheels1</td>
</tr>
<tr>
<td>Duplication</td>
<td>tarheels#1 → tarheels#11</td>
</tr>
<tr>
<td>Substitution</td>
<td>tarheels#1 → tarheels#2</td>
</tr>
<tr>
<td>Insertion</td>
<td>tarheels#1 → tarheels#12</td>
</tr>
<tr>
<td>Leet Transform</td>
<td>tarheels#1 → t@rheels#1</td>
</tr>
<tr>
<td>Block Move</td>
<td>tarheels#1 → #tarheels1</td>
</tr>
<tr>
<td>Keyboard Transform</td>
<td>tarheels#1 → tarheels#!</td>
</tr>
</tbody>
</table>
Evaluation

• Pick a known plaintext, non-last password (OLD)
• Pick any later password (NEW)
• Attempt to crack NEW with transform tree rooted at OLD
Results: Offline Attack

Success rate

Edit Dist 39% 41% 41% 26%
Edit w/ Mov 28% 25% 24% 28%
Loc Ind 25% 37% 17% 30%
Pruned 28% 30% 37% 24%

Within 3 Seconds!!

Takeaway: Memory limitations, convenience
SECURITY IMAGES AND THE ADVERSARY PROBLEM
Complete Sign On

Verify Identity

Please verify that your Personal Security Image and Caption are correct

Step 1: Verify Your Personal Security Image and Caption

Is this your Personal Security Image?

Is this Your Caption? Nice House

If you do not recognize your Personal Security Image & Caption then DO NOT enter your password and email us immediately at cmu-banking-research-study@ece.cmu.edu.

Step 2: Enter Password

User ID: tyutyu
Password:  
Forgot Password?

Sign On Questions

What is the Personal Security Image and Caption?
What should I do if I forgot my Personal Security Image and/or Caption?
What should I do if the wrong Personal Security Image and/or Caption is showing?
Goal: Prevent phishing

If you do not recognize your Personal Security Image & Caption then DO NOT enter your password!
Study design

• Participants recruited via MTurk
• Each day, receive an email with a small $ amount. Log in and “report” the deposit.
• At the end of the study, receive the amount “deposited.”
• On last day, security image is absent: “Under maintenance.”
• Will participants log in?
Varieties of security images

• Control
• Large, blinking
• Interactive (click, type a word)
• Custom image
• No caption
• Also: security priming, less habituation
Results

• 80-100% claimed they looked at the image, but:
  • 73% entered passwords despite no image
  • No significant differences by image type
  • Users with stronger passwords logged in less often (65% to 80%)

Takeaway: Attention failure, misaligned priorities, misunderstanding security concepts
PASSWORD METERS AND MOTIVATING YOUR USERS
Password Meters …

• … come in all shapes and sizes

[Ur et. al, USENIX Sec 2012]
Experimental setup

• No meter
• Baseline (boring) meter
• Visual differences
  – Size, text only
• Dancing bunnies (wait and see)
• Scoring differences
  – Same password scores differently
# Conditions with Visual Differences

<table>
<thead>
<tr>
<th>Type</th>
<th>Feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type new password:</td>
<td>8-character minimum; case sensitive</td>
</tr>
<tr>
<td>Baseline meter</td>
<td>Bad. Consider adding a digit or making your password longer.</td>
</tr>
<tr>
<td>Three-segment</td>
<td>Bad. Consider adding a digit or making your password longer.</td>
</tr>
<tr>
<td>Green</td>
<td>Bad. Consider adding a digit or making your password longer.</td>
</tr>
<tr>
<td>Tiny</td>
<td>Bad. Consider adding a digit or making your password longer.</td>
</tr>
<tr>
<td>Huge</td>
<td>Bad. Consider adding a digit or making your password longer.</td>
</tr>
<tr>
<td>No suggestions</td>
<td>Bad.</td>
</tr>
<tr>
<td>Text-only</td>
<td>Bad. Consider adding a digit or making your password longer.</td>
</tr>
</tbody>
</table>
## Conditions with Visual Differences

<table>
<thead>
<tr>
<th>Condition</th>
<th>Feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type new password:</td>
<td>[Input Field] 8-character minimum; case sensitive</td>
</tr>
<tr>
<td>Baseline meter</td>
<td>Bad. Consider adding a digit or making your password longer.</td>
</tr>
<tr>
<td>Three-segment</td>
<td>Bad. Consider adding a digit or making your password longer.</td>
</tr>
<tr>
<td>Green</td>
<td>Bad. Consider adding a digit or making your password longer.</td>
</tr>
<tr>
<td>Tiny</td>
<td>Bad. Consider adding a digit or making your password longer.</td>
</tr>
<tr>
<td>Huge</td>
<td>Bad. Consider adding a digit or making your password longer.</td>
</tr>
<tr>
<td>No suggestions</td>
<td>Bad.</td>
</tr>
<tr>
<td>Text-only</td>
<td>Bad. Consider adding a digit or making your password longer.</td>
</tr>
</tbody>
</table>
## Conditions with Visual Differences

<table>
<thead>
<tr>
<th>Condition</th>
<th>Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type new password:</td>
<td>usenIX</td>
</tr>
<tr>
<td>Baseline meter</td>
<td>Fair. Consider adding a digit or making your password longer.</td>
</tr>
<tr>
<td>Three-segment</td>
<td>Fair. Consider adding a digit or making your password longer.</td>
</tr>
<tr>
<td>Green</td>
<td>Fair. Consider adding a digit or making your password longer.</td>
</tr>
<tr>
<td>Tiny</td>
<td>Fair. Consider adding a digit or making your password longer.</td>
</tr>
<tr>
<td>Huge</td>
<td>Fair. Consider adding a digit or making your password longer.</td>
</tr>
<tr>
<td>No suggestions</td>
<td>Fair.</td>
</tr>
<tr>
<td>Text-only</td>
<td>Fair. Consider adding a digit or making your password longer.</td>
</tr>
</tbody>
</table>
Conditions with Visual Differences

Type new password: useniX$  
8-character minimum; case sensitive

Baseline meter  
Good. Consider adding a digit or making your password longer.

Three-segment  
Good. Consider adding a digit or making your password longer.

Green  
Good. Consider adding a digit or making your password longer.

Tiny  
Good. Consider adding a digit or making your password longer.

Huge  
Good. Consider adding a digit or making your password longer.

No suggestions  
Good.

Text-only  
Good. Consider adding a digit or making your password longer.
Conditions with Visual Differences

Type new password: useniX$e5
8-character minimum; case sensitive

Baseline meter: Excellent!
Three-segment: Excellent!
Green: Excellent!
Tiny: Excellent!
Huge: Excellent!
No suggestions: Excellent!
Text-only: Excellent!
Conditions with Visual Differences

- **Baseline meter**: Excellent!
- **Three-segment**: Excellent!
- **Green**: Excellent!
- **Tiny**: Excellent!
- **Huge**: Excellent!
- **No suggestions**: Excellent!
- **Text-only**: Excellent!

Type new password: useniX$e5

8-character minimum; case sensitive
Bunny Condition

A strong password helps prevent unauthorized access to your email account.
The stronger your password, the faster Bugs Bunny dances!

Type new password: 

8-character minimum; case sensitive

Password strength: Please enter a password in the box above.

Retype new password: 

Make my password expire every 72 days.

Save
Bunny Condition

A strong password helps prevent unauthorized access to your email account. The stronger your password, the faster Bugs Bunny dances!

Type new password: [ ]

8-character minimum; case sensitive

Password strength: Please enter a password in the box above.

Retype new password: [ ]

☐ Make my password expire every 72 days.

Save
Conditions with Scoring Differences

Type new password: `userIX`

8-character minimum; case sensitive

**Baseline meter**

Fair. Consider adding a digit or making your password longer.

**Half-score**

Bad. Consider adding a digit or making your password longer.

**One-third-score**

Bad. Consider adding a digit or making your password longer.

**Nudge-B16**

Bad. Consider making your password longer.

**Nudge-Comp8**

Fair. Consider adding a digit or making your password longer.
Conditions with Scoring Differences

Type new password: useniX$e5
8-character minimum; case sensitive

Baseline meter
Excellent!

Half-score
Poor. Consider adding a different symbol or making your password longer.

One-third-score
Bad. Consider adding a different symbol or making your password longer.

Nudge-B16
Poor. Consider making your password longer.

Nudge-Comp8
Excellent!
Conditions with Scoring Differences

Type new password: usenIX$e5WHYis
8-character minimum; case sensitive

Baseline meter: Excellent!

Half-score: Fair. Consider adding a different symbol or making your password longer.

One-third-score: Poor. Consider adding a different symbol or making your password longer.

Nudge-B16: Good. Consider making your password longer.

Nudge-Comp8: Excellent!
Conditions with Scoring Differences

Type new password: usenlX$e5WHYismyP4$$
8-character minimum; case sensitive

Baseline meter: Excellent!

Half-score: Good. Consider adding a different symbol or making your password longer.

One-third-score: Poor. Consider adding a different symbol or making your password longer.

Nudge-B16: Excellent.

Nudge-Comp8: Excellent!
Conditions with Scoring Differences

Type new password: usen1X$e5WHYismyP4$$word99

8-character minimum; case sensitive

Baseline meter: Excellent!

Half-score: Excellent!

One-third-score: Fair. Consider adding a different symbol or making your password longer.

Nudge-B16: Excellent.

Nudge-Comp8: Excellent!
Conditions with Scoring Differences

Type new password: usen!X$e5WHYismyP4$$word99notGOOD

8-character minimum; case sensitive

Baseline meter: Excellent!

Half-score: Excellent!

One-third-score: Fair. Consider making your password longer.

Nudge-B16: Excellent.

Nudge-Comp8: Excellent!
Conditions with Scoring Differences

Type new password: userI\$e5WHYismyP4$$word99notGOODenough?

8-character minimum; case sensitive

Baseline meter

Excellent!

Half-score

Excellent!

One-third-score

Excellent!

Nudge-B16

Excellent.

Nudge-Comp8

Excellent!
Password Meters (Scoring)

<table>
<thead>
<tr>
<th>Number of Guesses</th>
<th>Percentage of Passwords Cracked</th>
</tr>
</thead>
<tbody>
<tr>
<td>No meter</td>
<td>0%</td>
</tr>
<tr>
<td>Baseline meter</td>
<td>10%</td>
</tr>
<tr>
<td>Nudge-comp8</td>
<td>20%</td>
</tr>
<tr>
<td>Bold text-only</td>
<td>30%</td>
</tr>
<tr>
<td>Text-only half</td>
<td>40%</td>
</tr>
<tr>
<td>Nudge-16</td>
<td>50%</td>
</tr>
<tr>
<td>One-third-score</td>
<td>60%</td>
</tr>
<tr>
<td>Half-score</td>
<td>70%</td>
</tr>
<tr>
<td>Weak</td>
<td>80%</td>
</tr>
<tr>
<td>Medium</td>
<td>90%</td>
</tr>
<tr>
<td>Strong</td>
<td>100%</td>
</tr>
</tbody>
</table>

- Weak: $5 \times 10^8$
- Medium: $5 \times 10^{10}$
- Strong: $5 \times 10^{12}$

Number of Guesses vs. Percentage of Passwords Cracked
Visual changes don’t significantly increase resistance to guessing.

Stringent meters with visual bars increase resistance to guessing, without affecting memorability.

Too stringent can deplete user buy-in and backfire.
Today

• Finish up usability, end early
  – Case study: implantable devices
  – Some tips for making things better
  – Recent work my collaborators and I did with crypto libraries

• Thursday: Exam review!
IMPLANTABLE DEVICES: BALANCING SECURITY AND OTHER VALUES
Implantable medical devices

• E.g., pacemakers, implantable defibrillators

• Increasingly, wireless comms:
  – Configure non-invasively
  – Report status and alerts automatically

• 2008: One model can be hacked wirelessly
  – Modify settings, steal private info, send large shock
A security paradox

• Authorized clinical access: ALWAYS
• Unauthorized access: NEVER

• … EXCEPT:
  – Emergency access for EMTs, unknown docs/hospitals

• Non-goal: Protection given long physical access
Brainstorm: Potential solutions?
Some potential solutions

• Passwords
  – Available via some broad medical database
  – Carried in wallet
  – Carried on medical alert bracelet
  – Visible or UV tattoo

[Denning et. al, CHI 2010]
More potential solutions

• Proximity device
  – “Master key” kept in doctor’s offices, hospitals
  – Locked when wearing bracelet/wearable
  – Unlocked when wearing bracelet/wearable

• Automated detection of emergency condition
### Interview study: Result highlights

<table>
<thead>
<tr>
<th>Feature</th>
<th>Liked (%)</th>
<th>Disliked (%)</th>
<th>Would Choose (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password on bracelet</td>
<td>0</td>
<td>27</td>
<td>0</td>
</tr>
<tr>
<td>Visible tattoo</td>
<td>9</td>
<td>55</td>
<td>9</td>
</tr>
<tr>
<td>UV tattoo</td>
<td>18</td>
<td>27</td>
<td>18</td>
</tr>
<tr>
<td>Unlock if bracelet absent</td>
<td>0/45</td>
<td>36/27</td>
<td>0/27</td>
</tr>
<tr>
<td>Proximity master key</td>
<td>27</td>
<td>0</td>
<td>27</td>
</tr>
<tr>
<td>Emergency detection</td>
<td>27</td>
<td>18</td>
<td>27</td>
</tr>
</tbody>
</table>

\[ N = 11 \]
MAKING THINGS BETTER
Use psychology in your favor

• Limit memory requirements
• Grab attention when you need it
• Make critical information stand out / avoid habituation
• Minimize effort:
  – To get users to take action, make it easy
  – To get users to avoid danger, make it difficult
Limit the user’s cognitive load

<table>
<thead>
<tr>
<th>High probability of danger</th>
<th>Might be dangerous</th>
<th>Very low probability of danger</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block</td>
<td>User must decide</td>
<td>Don’t bother user</td>
</tr>
</tbody>
</table>

Improve warnings

Help user decide by asking a question user is qualified to answer
Bad question

Your web browser thinks this is a phishing web site. Do you want to go there anyway?

Don’t go there  Go there anyway

I don’t know what a phishing site is.

I really want to go to this site.

Of course I will go there anyway!
Better question

You are trying to go to evilsite.com. Do you really want to go there or would you rather go to yourbank.com?

Go to yourbank.com  Go to evilsite.com

Of course I want to go to yourbank.com!
Hierarchy of solutions

• Make it “just work”
  – Invisible security

• Make security/privacy understandable
  – Make it visible
  – Make it intuitive
  – Use metaphors that users can relate to

• Train the user
In submission

COMPARING CRYPTOGRAPHIC APIS
Getting crypto right is hard

• Developers must pick:
  – algorithm
  – mode of operation
  – key size, etc.

• Challenging, error prone (ICSE’16)

• Alternatives claim to be more usable
  – libsodium, keyczar, cryptography.io

• Is this really true?
Online developer study

• Short python tasks, secure/insecure solutions
  – Symmetric: key gen./storage, encrypt/decrypt
  – Asymmetric: also certification validation

• One of 5 libraries:
  – PyCrypto, M2Crypto, cryptography.io, keyczar, PyNacl

• Exit survey
Not all libs support all tasks well

<table>
<thead>
<tr>
<th>Library</th>
<th>Current Version</th>
<th>Designed for Usability</th>
<th>Symmetric Key Generation</th>
<th>Secure Symmetric Key Storage</th>
<th>Asymmetric Encryption/Decryption</th>
<th>Secure Asymmetric Key Storage</th>
<th>Certificate Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PyCrypto</td>
<td>2.6.1</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>M2Crypto</td>
<td>0.25.1</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>cryptography.io</td>
<td>1.4</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Keyczar</td>
<td>0.716</td>
<td>●</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>PyNaCl</td>
<td>1.0.1</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>○</td>
</tr>
</tbody>
</table>

● = fully applies; ○ = partly applies; ○ = does not apply
Certificate validation

**Goal:** Verify that the SSL certificate from the central Citizen Measure server was issued by the Let’s Encrypt Certificate Authority to ensure that citizen reports are not being intercepted. You have to validate the certificate’s digital signature and common name. For your convenience, the SSL certificate from the Citizen Measure server is stored in ./citizenMeasureCertificate.pem and the Let’s Encrypt Certificate Authority certificate in ./leca.pem. You can take also a look at the Let’s Encrypt X3 Root CA and the server certificate.

```python
In [0]:
import nacl

def validate(certificate, root_certificate, hostname="citizen-measure.tk"):
    
    Purpose:
    Validate the given certificate's digital signature and common name.

    Arguments:
    certificate: The certificate to validate.
    hostname: The server's hostname.

    Return value:
    validation result: True if validating the certificate is correct, False otherwise.

    Notes:
    - The Citizen Measure server certificate can be found at ./citizenMeasureCertificate.pem
    - The Let's Encrypt Certificate Authority certificate can be found at ./leca.pem
    - If you used any other information source to solve this task than the linked documentation (e.g. a post on Stack Overflow, a blog post or a discussion in a forum), please provide the link right below:
    - additional information sources go here (e.g. https://stackoverflow.com/questions/415511/how-to-get-current-time-in-python)

    # This is where your code goes
    return False

# This is to test the code for this task.
certificate = open("./citizenMeasureCertificate.pem").read()
root_certificate = open("./leca.pem").read()
assert validate(certificate, root_certificate, "citizen-measure.tk"), "Certificate validation failed."
print "Task completed! Please continue."
```

Skeleton code, online code editor
Evaluation

- Correctness: Runs without errors, “works”
- Security: Manually coded
  - Predefined categories, 2 independent coders
- Self-report
  - Security thinking
  - System Usability Scale (SUS)
  - New API scale we designed
- Primarily analyzed w/ multiple regression
Recruitment via Github

• Scraped committers to 100k Python repos
• Invited random 50k of these
• Final, “valid” sample: 256
  – 208 professionals
  – 198 w/ no security background
  – 1571 who consented; many dropped out
Many similarities; Participants slightly more active
Functionality by library

Keyczar, m2crypto worst; c&p helps (significant)
Security (among functional)

“simplified” libs are most secure; asymmetric more secure than symmetric
Self-reported data

• Believed secure but weren’t: 20% of tasks!
  – Not different by library

• SUS: Nothing better than mediocre
  – Most disliked: keyczar, m2crypto, asymm
  – Very similar to functionality results

• From our scale: Documentation is key!
  – Keyczar: “Your documentation is bad and you should feel bad.”
Participant background

• Experience level:
  – High if python is your job, or programming in python > 5 years
  – Did not matter on any metric

• Security background:
  – Almost mattered to security results
  – Helps with usability reports

• Library experience: maybe helps usability
Takeaways

• Implementing crypto is (still) hard
• Simplified APIs do promote security  
  – Sort of!
• Documentation, full-featured-ness are key!

• … Lots more data in the paper!