DC Sec

Leibniz
Universität
Hannover

CaaS

Usable Security for the Cloud

Sascha Fahl, Marian Harbach and Matthew Smith

Overview - Defence

- Why Security Fails
  - Economic Factors
  - Technical Factors
  - Human Factors
  - Legal Factors
  - Usable Security
- Security as a Service
  - Facebook Example
  - Mock-ups
  - CaaS
  - User Study
- MindMesh
  - Human Centric Information Sharing
  - User Study

OVERVIEW - ATTACK

- Mobile and the Cloud
  - Appification
  - Android background
  - SSL Problems
  - Example Attacks
  - User Study

```
.class public Leu/nullbyte/android/urllib/EasySSLSocketFactory;
.super Ljava/lang/Object;
.source "EasySSLSocketFactory.java"
# interfaces
.implements Lorg/apache/http/conn/scheme/SocketFactory;
.implements Lorg/apache/http/conn/scheme/LayeredSocketFactory;
# instance fields
.field private sslcontext:Ljavax/net/ssl/SSLContext;
# direct methods
.method public constructor <init>()V
    .locals 1
    .prologue
    .line 47
    invoke-direct {p0}, Ljava/
    .line 50
```

Why Security Fails

Security Fails – a lot

Security Flaw Found in Windows
Worm Blasts Across the Net

Comodo Hack: 37,000 Legitimate
Certificates Issued by CAs for
Unqualified Names

Stuxnet Virus sets back Iran’s Nuclear
Program by 2 Years.
Physical damage to facilities

Sony Hack: Personal Information from
Approximately 24.6 Million Sony OE
Accounts may have been stolen

Security Fails – a lot

IT vs. Automotive Industry

"If General Motors had kept up with the technology like the computer industry has, we would all be driving $25 cars that got 1,000 miles to the gallon."[1]

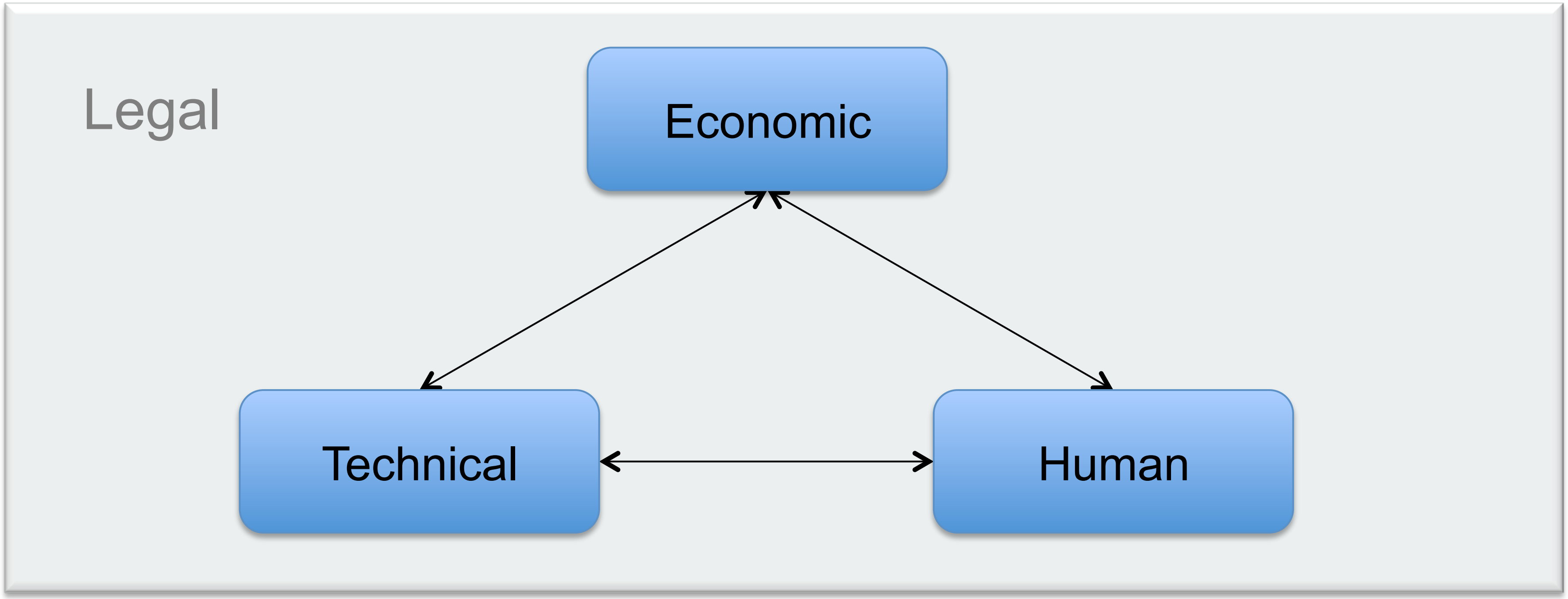
GM's Response:

- but they would crash unexpectedly every couple of days;
- we would just accept this, restart and drive on;
- the oil, water temperature, and alternator warning lights would all be replaced by a single "General Protection Fault" warning light;
- the airbag system would ask "are you sure" before deploying;
- every time GM introduced a new car, car buyers would have to learn to drive all over again because none of the controls would operate in the same manner as the old car.[2]

1) Reputedly said by Bill Gates
2) Summarized response by GM

Why Security Fails - Structure

We will look at the following factors



Legal Factors

Enforcing and Transferring Liabilities

- If it turns out that the tires of a car are faulty and may cause accidents,
- the manufacturer is obliged by law to recall them.
- This is facilitated by the fact that the manufacturer is liable for any problems that may arise from using faulty tires

- If a database software crashes and destroys the entire dataset costing millions
- or the network of a hospital goes down and costs the lives of patients
- the software vendors are not liable (due to clever licensing agreements)

Cloud Problems

- Countries often require providers to allow access to their users' data
  - e.g. U.S. Patriot Act gives law enforcement the right to access all data that is stored by U.S. service providers
  - Similar laws exist in other countries
- Social Networks (e.g Facebook)
  - Store all communications of their users
  - May have to hand it out or lose it in a breach
  - Have the right to use it for their own purposes
- Cloud Storage (e.g. Dropbox)
  - Stores all data of their users
  - May have to hand it out or lose it in a breach

Jurisdiction mandated by location of resources (probably)

Location transparency & Provider choice are issues

Economic Factors

Incentive Problems (Organizational)

- Customers would like productive, bug-free IT/software
- IT/SW Companies would like to maximise profits
- Security does not factor in either of these wishes directly

- Principle of adequate protection
  - Goal is not to maximize security, but to maximize utility while limiting risk to an acceptable level within reasonable cost

vs.

Security considerations

The first questions to ask when securing a system:

- Who do we think will attack us?
- What is their motivation?
- What resources and skills do they have?
- How would the attack affect us?
  - Direct damage:
    theft, destroyed work, recovery costs…
  - Indirect damage:
    reputation, future business, stock market value

Vulnerability vs. Profit

- The release of 146 vulnerabilities was analysed and it was shown that the stock price of a company drops on average by 0.63% compared to the NASDAQ15 on the day the flaw is announced [1]

- Microsoft stocks rise 7% after strong Q3 earnings (Windows 7 release Jul'09) [23rd Oct '09]
- Microsoft stocks fall 3% after reporting lagging OS sales [29th April '11]
- Sony stocks fall 3.7% due to "largest hack in corporate history" [6th May '11]
- Toyota stocks fall 7% after accelerator pedal recall [27th Jan '10]

IT failures common in all organizations

This leads to little incentive to invest in good security

[1] Why Computer Security Fails – An Economic View – Taba 2005

Technical Factors

Technical Factors (some examples)

Technical factors (administrative):

- Standard off-the-shelf but insecure systems
  - updated infrequently
- Changes in environment; bad feature interaction
- Outsourcing to the Cloud, Decentralised Systems
  - Administration no longer under local control
  - Fortress approach does not work anymore

Technical factors (user driven):

- The market pushes non-securable devices and services
  - iPhone, Dropbox, Facebook, etc
- Enterprises need to cope with these unsecured entities in their corporate environment.
  - Gadgets mostly don‘t include enterprise security features
  - Consumer security features (SSL for social network sites/blogs/etc) can work against enterprise security features

System vs Security Engineering

- System/software engineering - making systems behave in a clearly specified way - is a difficult activity.
- Security engineering - preventing systems misbehaving in many unspecified ways - is, in a sense, even more difficult.

This often leads to cumbersome and complex security mechanisms which frustrate users

Human Factors

Usable Security?

Human Factors (some examples)

- failure to follow procedure
  - turning off or skipping security checks, ignoring warnings
  - choosing weak passwords
  - putting confidential data on unencrypted thumb drives
- failure to understand security implications of actions
  - opening unexpected attachments, installing Apps
  - accepting certificate warnings
- dealing with exceptional circumstances improperly
  - preferring to believe everything is ok (contrary to evidence)
  - following on-screen instructions (of the attacker) without question
- falling prey to social engineering attacks
  - divulging information inadvertently, accidently
  - being corruptible
- insider attacks
  - payback for being sacked

Password Example

- Passwords are still a mainstay of modern security
  - and a very common cause of security problems
- Common password advice
  - make it long and random
  - use special characters
  - don't write it down
  - change it often
  - don't re-use across services

good technical advice

bad usability advice

- Password problems lead to
  - lost productivity
  - recovery cost
  - frustrated users who try and circumvent system

economic disincentive to use good passwords

Usable Security: An Emerging Research Field

Google Scholar “hits”

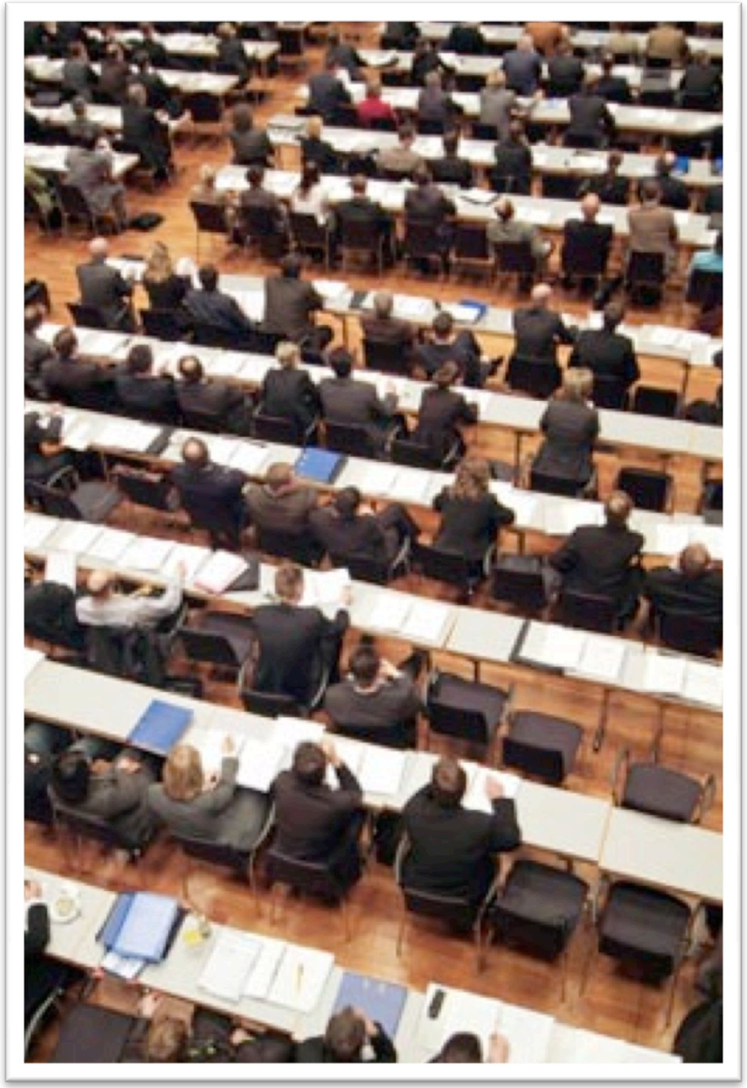
- security 233,000
- usability 25,140
- security and usability 433

IT books on Amazon.com

- security 13,739
- usability 1,647
- security and usability 1

Potential for growth

- publication of papers, books, lectures
- organisation of conferences
- development of centres of excellence



Areas of usable eSecurity Research

- Systems that "just work"
  - with minimal involvement of humans in security-critical functions
  - domain specific solutions
- Making secure systems intuitive and easy to use
  - human friendly systems
  - self explaining systems
  - context awareness
  - intelligent interaction & integration
- Approaches to teaching humans security-critical tasks
  - person to person
  - machine to person

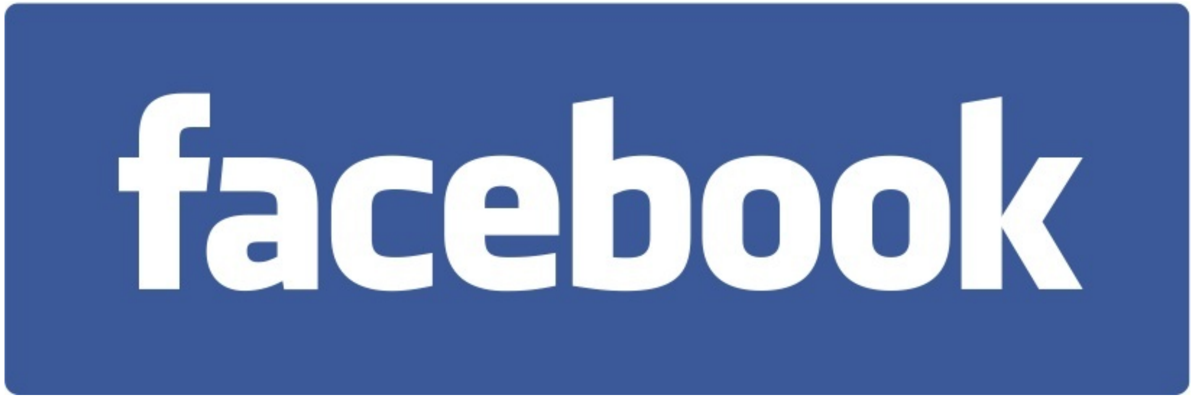
Confidentiality as a Service

PROTECT YOUR DATA IN THE CLOUD

Personal Information Sharing

- How can personal information be shared?
  - web servers, cloud storage, social networks etc.
- Confidentiality (crypto) is a key aspect for sensitive data
  - requires user expertise
  - cumbersome
  - error-prone
  - hard to fix
- How is information often shared?
  - e-Mail, DVDs, Skype, Print-Outs
- Why?
  - usability, security, usability of security

Facebook Conversations



- In 2010, **500 million Facebook** users sent **4 billion messages** per day
- Today, there are more than 900 million Facebook users

Are they aware of potential privacy threats?

Screening Study

Questions

- Do users realize the privacy threats for their conversations on Facebook?
- Are they concerned that Facebook is able to access their conversations?

Design

- Introduced as an online poll on Facebook privacy
- Invited 16,915 students
- Also: find participants for follow-up study

Screening Study II

Results

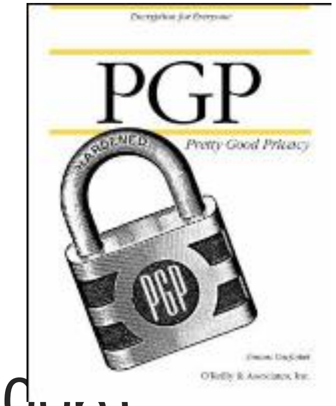
- 514 participants
- 413 (80 %) knew that Facebook was able to access their conversations
- 342 (67 %) were concerned about their conversations’ privacy
- 82 (16 %) did not care what Facebook does with their messages

➢ **So, why is nobody encrypting Facebook messages?**

Related User Studies

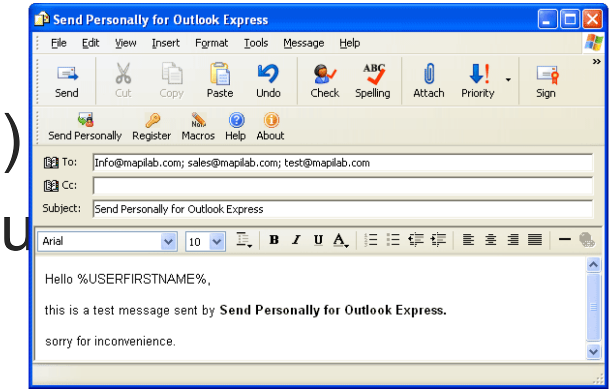
Email Security

- Why Johnny Can't Encrypt (Whitten and Tygar, 1999)
  - PGP 5 user study



- Why Johnny Still Can't Encrypt (Sheng et al., 2006)
  - PGP 9 user study

- Johnny 2 (Garfinkel and Miller, 2005)
  - S/MIME KCM for Outlook user stu



To: Info@mapilab.com; sales@mapilab.com; test@mapilab.com

Cc:

Subject: Send Personally for Outlook Express

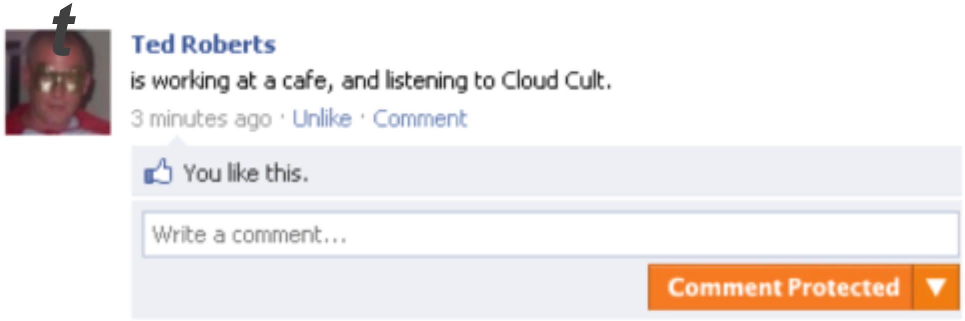
Hello %USERFIRSTNAME%,

this is a test message sent by **Send Personally for Outlook Express.**

sorry for inconvenience.

Available Solutions

uProtect.it



Ted Roberts

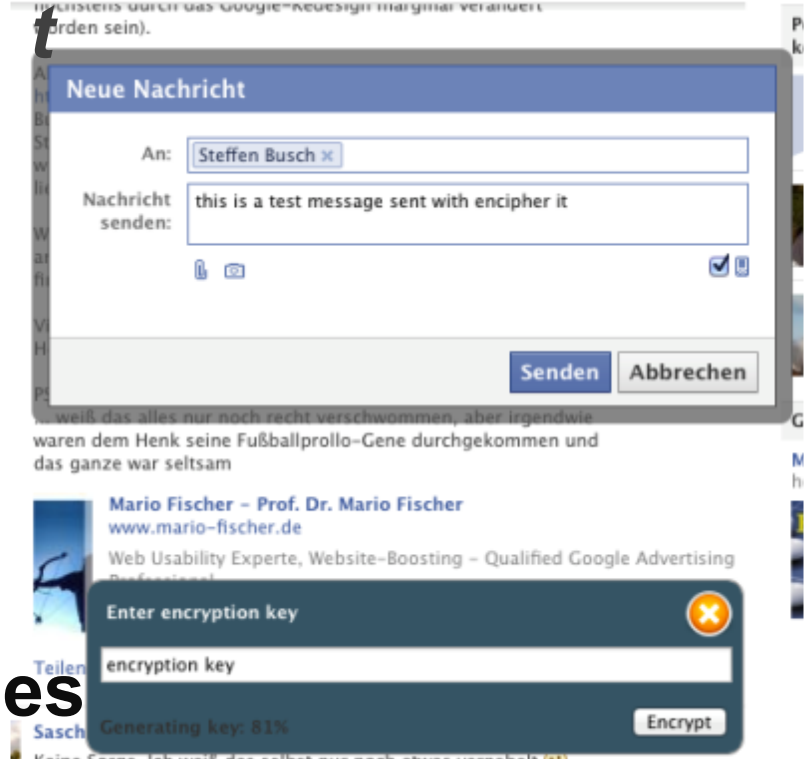
is working at a cafe, and listening to Cloud Cult.

3 minutes ago · Unlike · Comment

You like this.

Write a comment...

encipher.it



...rden sein).

Neue Nachricht

An: Steffen Busch

Nachricht senden: this is a test message sent with encipher it

...weiß das alles nur noch recht verschwommen, aber irgendwie waren dem Henk seine Fußballprollo-Gene durchgekommen und das ganze war seltsam

Mario Fischer – Prof. Dr. Mario Fischer

www.mario-fischer.de

Web Usability Experte, Website-Boosting – Qualified Google Advertising Professional

Enter encryption key

encryption key

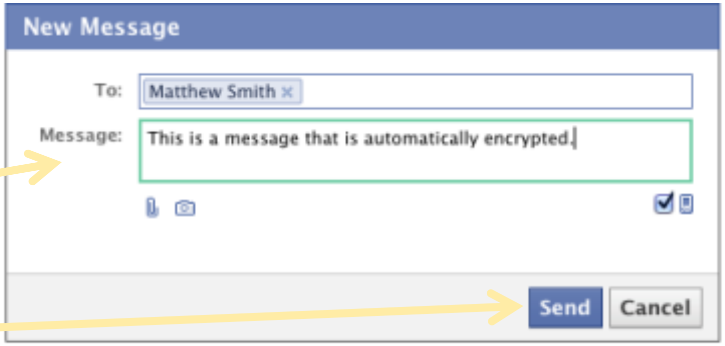
Generating key: 81%

Extracted functional variables

- Manual/automatic encryption
- Manual/automatic key management

Mockups

1



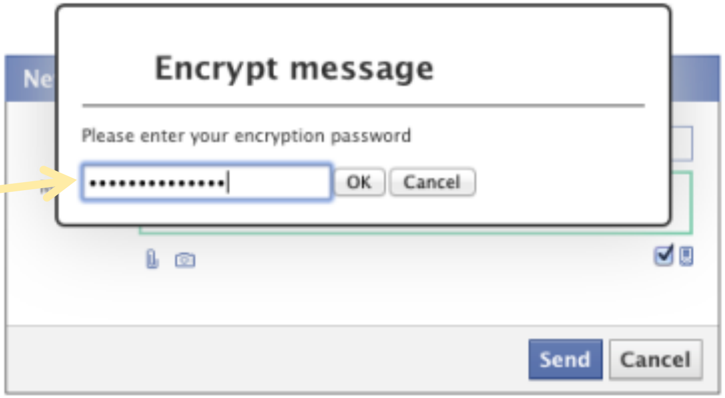
To: Matthew Smith ×

Message: This is a message that is automatically encrypted.

Enter recipients + message

Push the send button

2

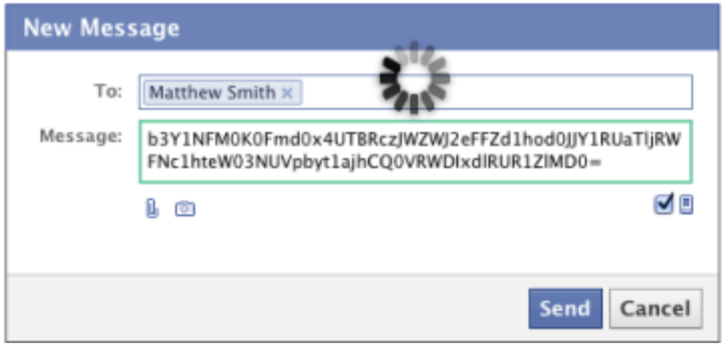


Please enter your encryption password

••••••••••••••

Enter encryption key
(once per session or once per message)

3



To: Matthew Smith ×

Message: b3Y1NFM0K0Fmd0x4UTBRczJWZWJ2eFFZd1hod0JJY1RUaTljRW FNc1hteW03NUVpbyt1ajhCQ0VRWDIxdlRUR1ZIMD0=

Encrypted message is sent

Lab Study

Goals

1. Which features enable most usable mechanism?
2. Do users want a key recovery mechanism?
3. Who are users afraid of?

Within-subjects Design with random latin squares setup

Participants

- Needed to be concerned about their privacy
  - Frequent Facebook users, non IT experts
- 96 participants
- No personal data was required during the study

Lab Study Results

- **Automatic** encryption and key management have *better usability* than manual

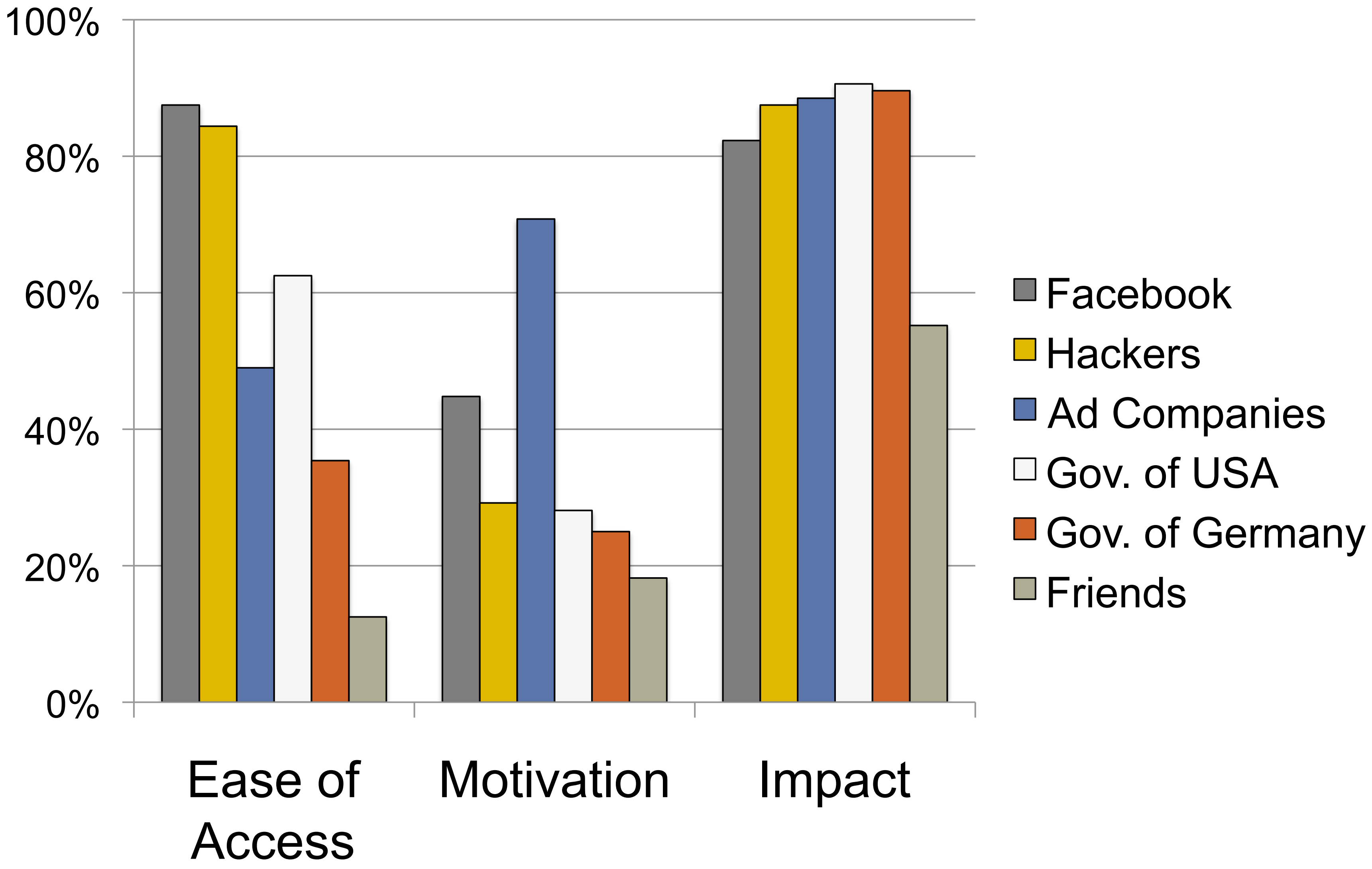
- **Automatic** key management has *higher acceptance*
  - No difference for automatic encryption

- **Key Recovery is necessary**
  - 72% of users afraid to loose password would not use mechanisms without key recovery

DC Sec

Who Are Users Afraid Of?

Leibniz Universität Hannover



DESIGNING A USABLE SOLUTION

Confidentiality as a Service

- How to protect data on popular Cloud services such as
  - Dropbox, Facebook, Amazon S3, web mail, etc.?

- Public Key / CA Infrastructures
  - requires user expertise
  - cumbersome
  - error-prone
  - hard to fix

An Ideal Solution Would Offer…

… perfect security without any effort.

Previous Johnny Studies showed that setup of encryption mechanisms is crucial

- Apply well-known paradigms from everyday applications

No complex cryptographic objects, but username/ password

- Users are familiar with this concept
- Email Based Identification and Authentication (EBIA)
  - Garfinkel, 2003

Key recovery possible

- Loosing decryption credentials ≠ encrypted data lost
- Desirable according to our study

Our Solution

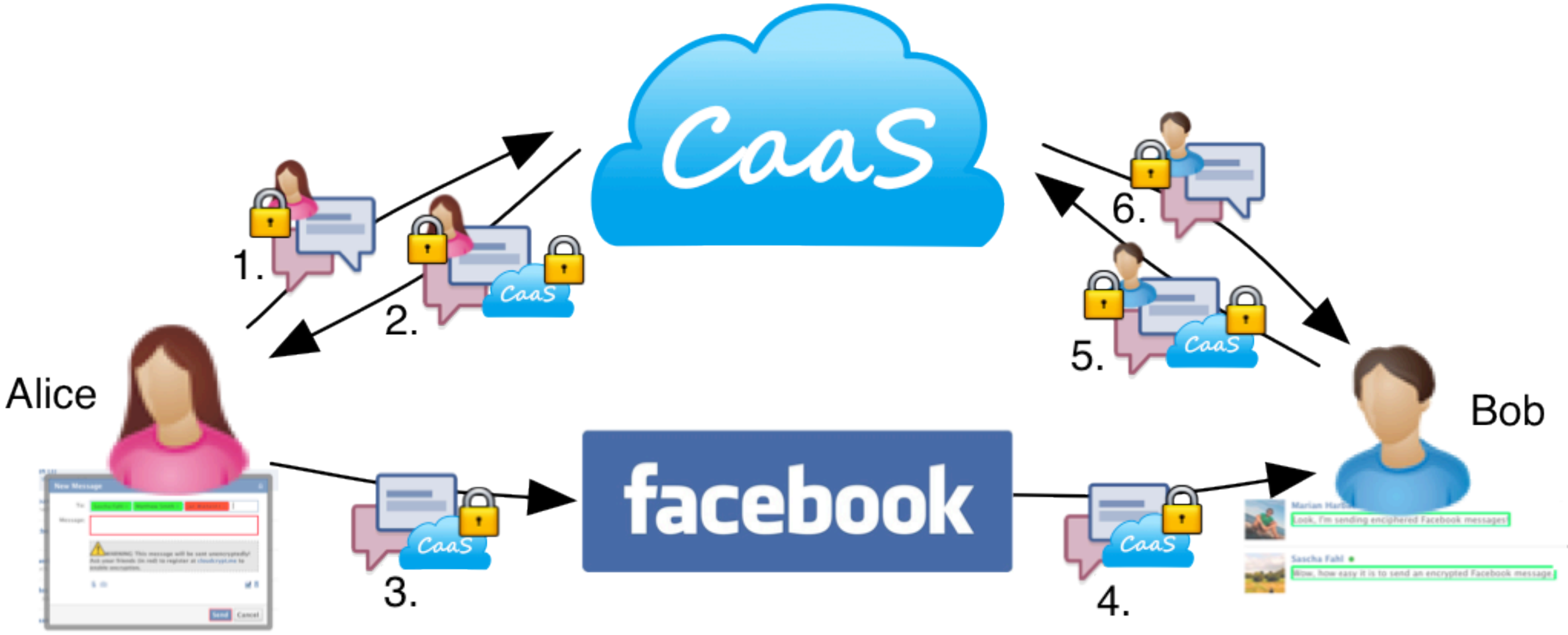
Based on the lab study results we extracted the following requirements

- Username/password authentication
- Automatic encryption
- Automatic key management
- Key recovery feature

Confidentiality as a Service

- How to protect data on popular Cloud services such as
  - Dropbox, Facebook, Amazon S3, web mail, etc.?
- Public Key / CA Infrastructures
  - requires user expertise
  - cumbersome
  - error-prone
  - hard to fix
- Confidentiality as a Service (CaaS)
  - separation of capabilities
  - less need to trust Cloud or CaaS provider
  - leverages existing infrastructure
  - zero key management for the user / known paradigms

Usability / Security Trade-off



New Message

To:

Message:

WARNING: This message will be sent unencrypted! Ask your friends (in red) to register at cloudcrypt.me to enable encryption.

Send Cancel

Marian Harb

Look, I'm sending enciphered Facebook messages!

Sascha Fahl

Wow, how easy it is to send an encrypted Facebook message.

Traditional approaches to confidentiality:

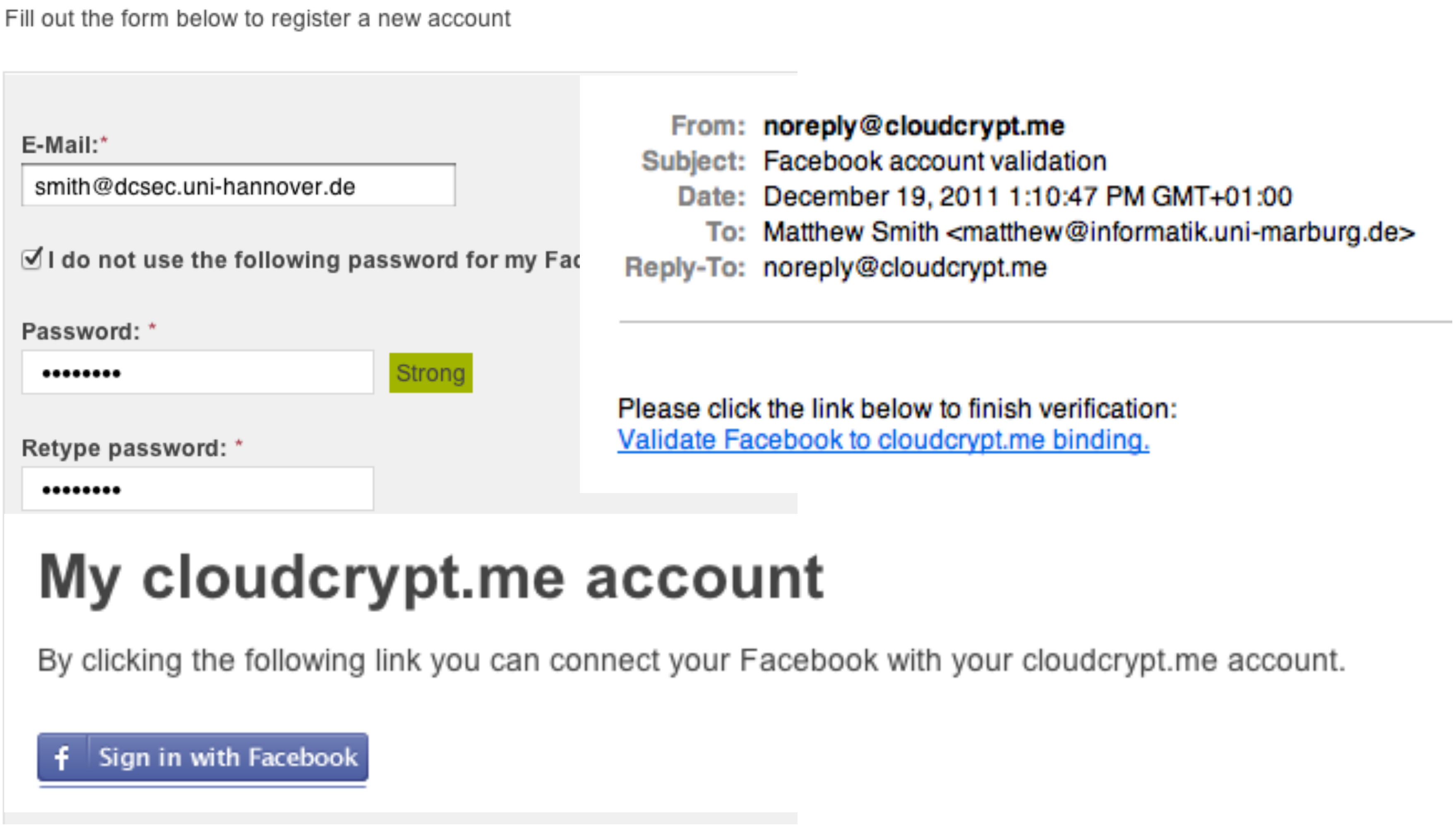
- encrypt data to
- protect it from everybody

Our approach to confidentiality:

- encrypt data to
- protect it from those who can but shouldn't access it

User Binding

Create and bind CaaS to a Facebook account using a known paradigm



Fill out the form below to register a new account

E-Mail:*

smith@dcsec.uni-hannover.de

☑ I do not use the following password for my Fac

Password: *

••••••••• Strong

Retype password: *

•••••••••

From: noreply@cloudcrypt.me
Subject: Facebook account validation
Date: December 19, 2011 1:10:47 PM GMT+01:00
To: Matthew Smith <matthew@informatik.uni-marburg.de>
Reply-To: noreply@cloudcrypt.me

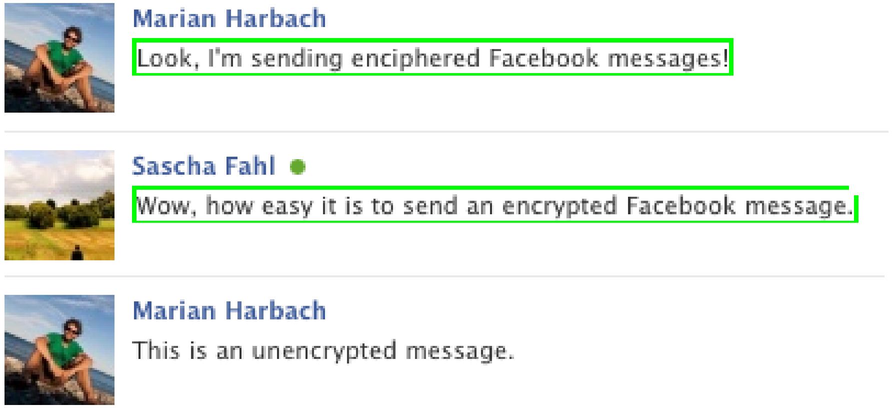
Please click the link below to finish verification:
Validate Facebook to cloudcrypt.me binding.

My cloudcrypt.me account

By clicking the following link you can connect your Facebook with your cloudcrypt.me account.

CaaS Facebook UI

- Minimally intrusive (workflow)
  - no key management
  - multiple device capable
- Highly visible (perception)
  - direct connection between data and security UI



Marian Harbach

Look, I'm sending enciphered Facebook messages!

Sascha Fahl

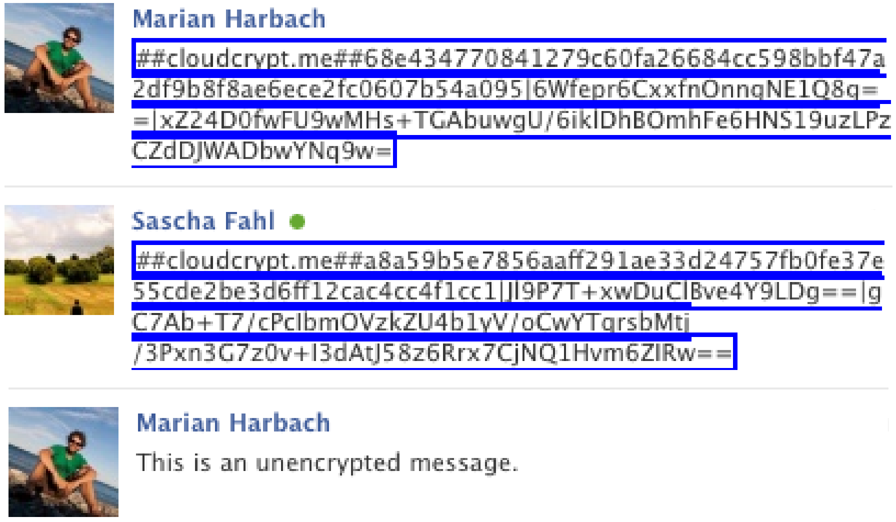
Wow, how easy it is to send an encrypted Facebook message.

Marian Harbach

This is an unencrypted message.

Cleartext after CaaS decryption

Ciphertext stored at Facebook



Marian Harbach

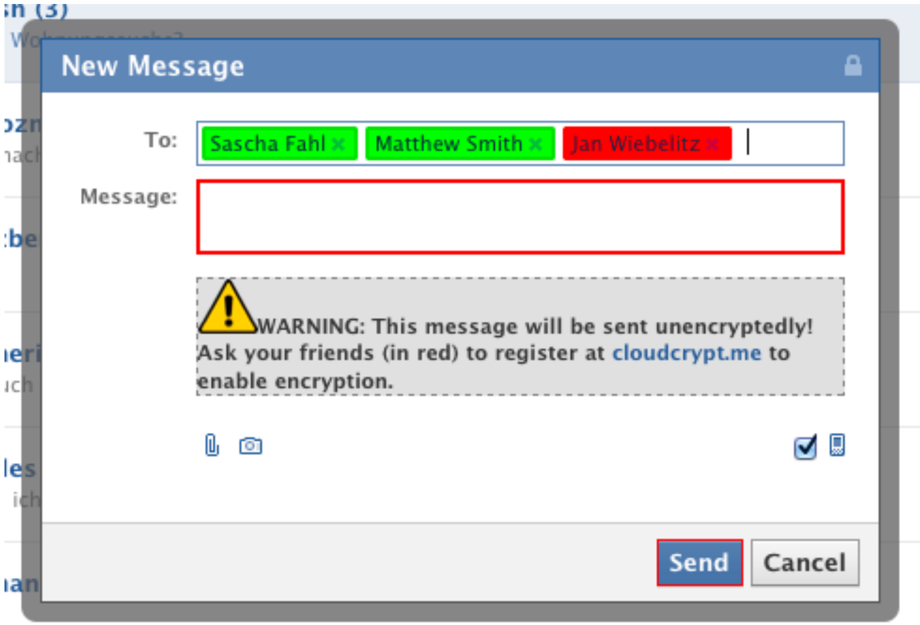
##cloudcrypt.me##68e434770841279c60fa26684cc598bbf47a2df9b8f8ae6ece2fc0607b54a095|6Wfepr6CxxfnOnngNE1Q8q==|xZ24D0fwFU9wMHs+TGAbuwgU/6iklDhBOmhFe6HNS19uzLPzCZdDJWADbwYNq9w=

Sascha Fahl

##cloudcrypt.me##a8a59b5e7856aaff291ae33d24757fb0fe37e55cde2be3d6ff12cac4cc4f1cc1|Jl9P7T+xwDuClBve4Y9LDg==|gC7Ab+T7/cPclbmOVzkZU4b1yV/oCwYTqrsbMtj/3Pxn3G7z0v+l3dAtJ58z6Rrx7CjNQ1Hvm6ZIRw==

Marian Harbach

This is an unencrypted message.



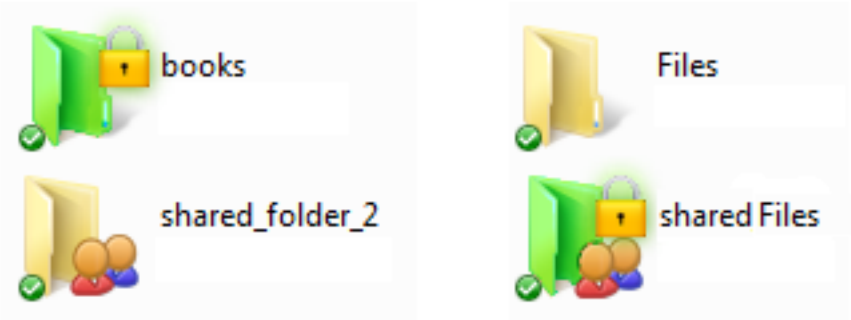
New Message

To: Sascha Fahl × Matthew Smith × Jan Wiebelitz ×

Message:

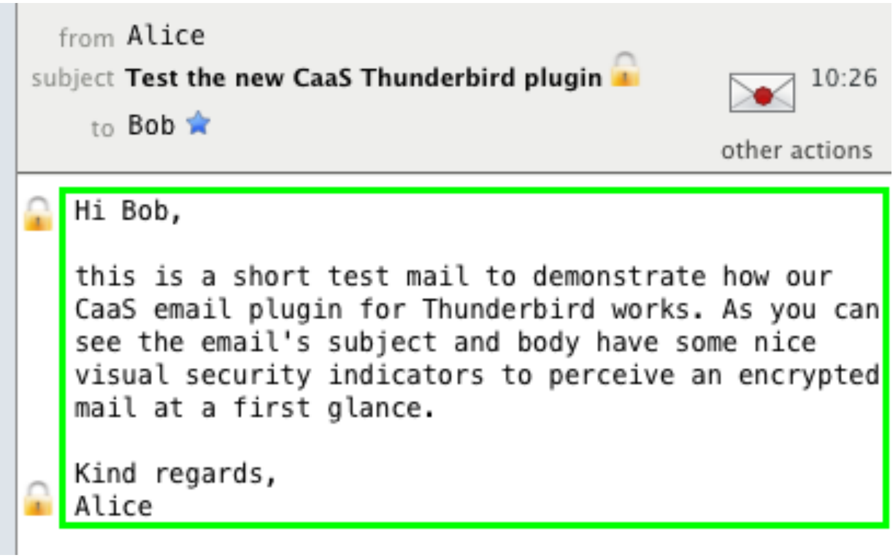
⚠ WARNING: This message will be sent unencrypted!
Ask your friends (in red) to register at cloudcrypt.me to enable encryption.

Dropbox and Thunderbird



- Dropbox
  - Protect both private and shared folders
  - data encrypted locally
- Thunderbird
  - eMail protection

- AC based on service identity
  - CaaS binds account to service identity
  - eMail verification
  - we use the Cloud AC to minimize the security usability overhead

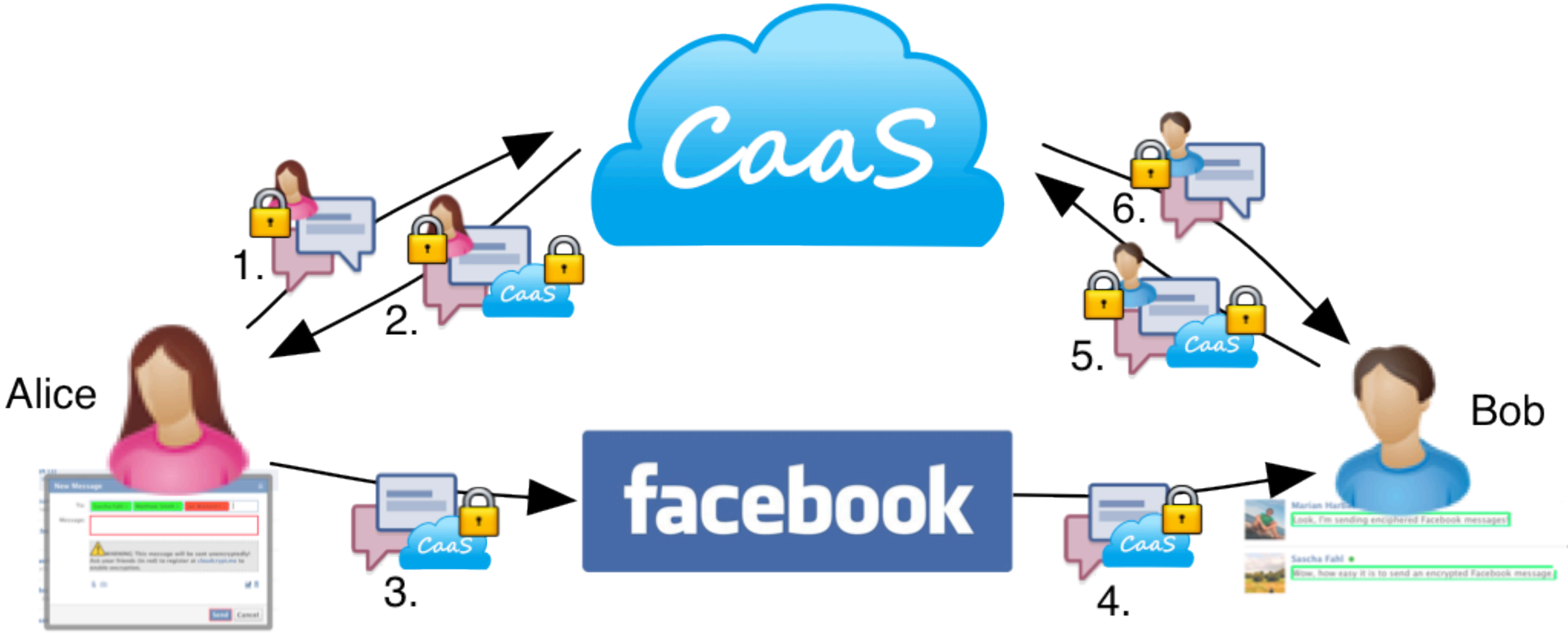


Hi Bob,

this is a short test mail to demonstrate how our CaaS email plugin for Thunderbird works. As you can see the email's subject and body have some nice visual security indicators to perceive an encrypted mail at a first glance.

Kind regards,
Alice

Commutative Encryption Layers



WARNING: This message will be sent unencryptedly! Ask your friends (in red) to register at cloudcrypt.me to enable encryption.

Marian Harb
Look, I'm sending enciphered Facebook messages!

Sascha Fahl
Wow, how easy it is to send an encrypted Facebook message.

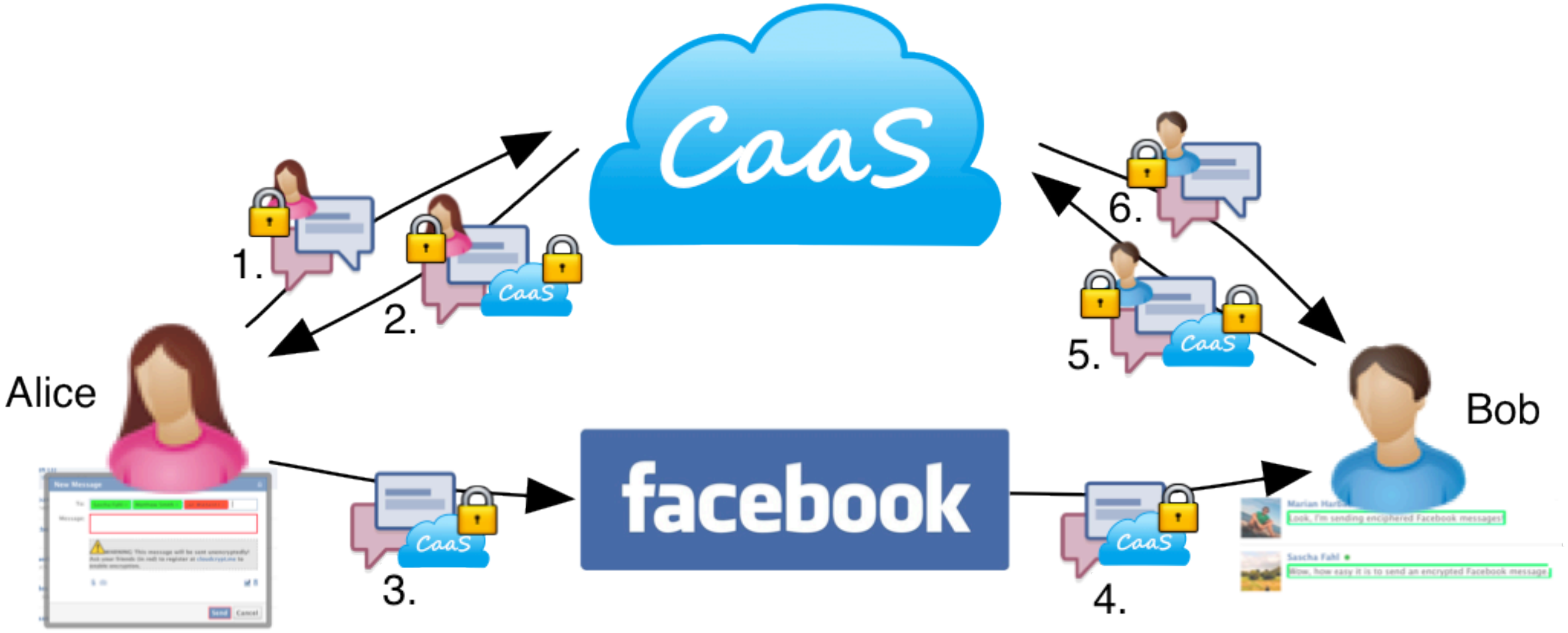
Bootstrapping of AC allows us to forgo asymmetric cryptography

- no key management
- device portability

Layered symmetric cryptography approach:

- XOR-based commutative cryptographic protection layers
- novel actor based ephemeral key generation

CaaS Workflow



WARNING: This message will be sent unencrypted!/ Ask your friends (in red) to register at cloudcrypt.me to enable encryption.

Marian Harb

Look, I'm sending enciphered Facebook messages!

Sascha Fahl

Wow, how easy it is to send an encrypted Facebook message.

① Alice adds local cLayer (+cLayerLocal$_A$)

② CaaS adds remote cLayer (+cLayerRemote)

③ Alice removes her local cLayer (-cLayerLocal$_A$)

④ Bob adds local cLayer (+cLayerLocal$_B$)

⑤ CaaS removes cLayer (-cLayerRemote)

⑥ Bob removes his local cLayer (-cLayerLocal$_B$)

Key Management

① Alice adds local cLayer (+cLayerLocal$_A$)
- random symmetric key K$_A$
- send encrypted data + ACL

② CaaS adds remote cLayer (+cLayerRemote)
- create symmetric key from ID$_{Alice}$+ ACL + master secret
- no need to store key

③ Alice removes her local cLayer (-cLayerLocal$_A$)
- discard key K$_A$

④ Bob adds local cLayer (+cLayerLocal$_B$)
- random symmetric key K$_B$
- send encrypted data + sender ID + ACL

⑤ CaaS removes cLayer (-cLayerRemote)
- check if ID$_{Bob}$ in ACL
- create symmetric key from ID$_{Alice}$ + ACL + master secret

⑥ Bob removes his local cLayer (-cLayerLocal$_B$)

CaaS Security

CaaS Provider

- data presented to the CaaS provider is protected by a local cLayer
- CaaS provider cannot retrieve remote cLayer protected data from Cloud service provider

Cloud Service Provider (CSP)

- data presented to the CSP is protected by the remote cLayer
- ACL injection attacks can be detected by the client

Only if CaaS and CSP collude confidentiality is broken
- use multiple CaaS provider to minimse threat

Evaluation Studies

Goals

- Usability evaluation of the process as a whole
- Are users willing to pay for such a service?
- More details on the needs for key recovery
- What is the level of perceived security?

Participants

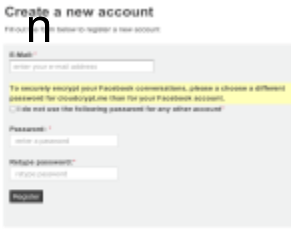
- 15 participants, randomly selected from the screening study
- Students, 6 male + 9 female, 22 years on average
- 233 Facebook friends on average
- At least 5 private Facebook messages/week

Evaluation (contd.)

Procedure

- Registration + message encryption/decryption questionnaire, semi-structured interview
- 1 interviewer + 1 assistant present
- Took 33 minutes on average overall
- 10 Euros compensation

Registration



Create a new account

Fill out the form below to register a new account

E-Mail: *

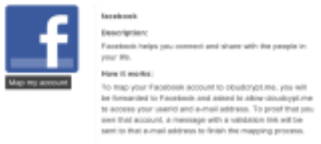
To securely encrypt your Facebook conversations, please a choose a different password for cloudcrypt.me than for your Facebook account.

☐ I do not use the following password for any other account *

Password: *

Retype password: *

+ Binding



facebook

Description:

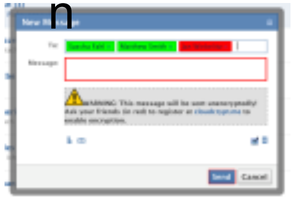
Facebook helps you connect and share with the people in your life.

How it works:

To map your Facebook account to cloudcrypt.me, you will be forwarded to Facebook and asked to allow cloudcrypt.me to access your userid and e-mail address. To proof that you own that account, a message with a validation link will be sent to that e-mail address to finish the mapping process.

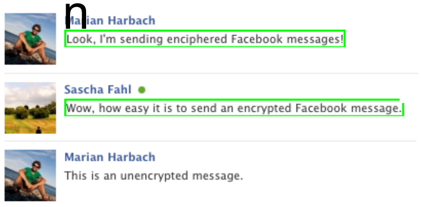
+ Installation

+ Encryption



WARNING: This message will be sent unencrypted! Ask your friends (in red) to register at cloudcrypt.me to enable encryption.

+ Decryption



Marian Harbach

Look, I'm sending enciphered Facebook messages!

Sascha Fahl ●

Wow, how easy it is to send an encrypted Facebook message.

Marian Harbach

This is an unencrypted message.

Evaluation – Results

Core questionnaire findings

- 5-point Likert-scale questions

(1= *I completely disagree*, 5 = *I completely agree*)

N = 15	**avg**	**sd**
I‘m sure I used the mechanism correctly	3.93	1.03
I would send sensitive messages with this service in the future	4.06	0.96
I would send all messages with this service in the future	3.46	1.06
I have the feeling that my messages are now well protected	3.53	1.06

Evaluation – Results (contd.)

Comments from the Interviews:

Registration process

- *P2: “I would describe the effort involved in setting up such an account as relatively small. I think it took me about 30 seconds – if it really helps to protect my messages this is definitely worthwhile.”*

Encryption and Decryption

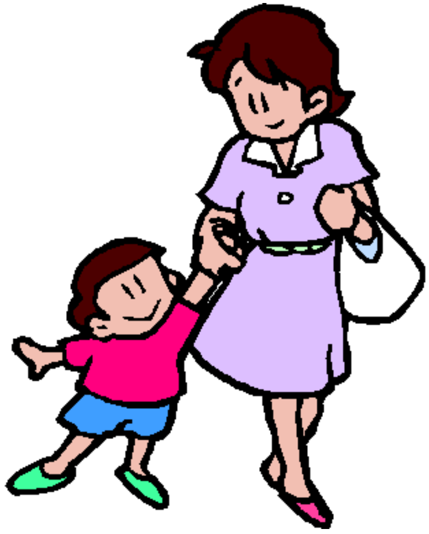
- *“uncomplicated, simple, secure”*
- *“I thought there would be annoying popups and I really liked that none appeared”*

Evaluation – Results (contd.)

Would users be willing to pay for such a service?

- 4 of the 15 participants answered they would not be willing to pay anything for encrypting their private FB messages
- Rest would pay a small amount for the servi

r an iPhone App" (5 participants)



A female participant said: *"I would not pay for the service for myself, but if I had children I would pay money to protect their privacy."*

Evaluation – Results (contd.)

Password Recovery

- 11 participants would not use the service without recovery
- 1 was concerned about security problems through recovery
- 3 did not care

“I would definitely need a recovery mechanism because losing access to my data would be disastrous.” (P15)

“This would be much less secure, because a hacker who has access to my email and Facebook account can then also decrypt my Facebook messages.” (P12)

“I never read old Facebook messages.” (P3)

Evaluation – Results (contd.)

Perceived Security and Trust

- Five participants stated they knew that the messages were encrypted *"because of the jumbled up text that was displayed" (P2)*.

- Yet, all participants stated that they needed to establish trust into the encryption software to send **more sensitive** messages
  - 4 participants do not trust computer software in general
  - 11 participants said that they needed to be convinced by friends or experts

"I really cannot say if the program does what it purports to do. I mean, any app could probably draw a green border around my message to simulate security. I would need some proof." (P2)

"On the Internet, you can download a program to crack everything." (P6)

CaaS Evaluation Summary

User study with 20 undergrad students for Facebook setup

- registering for the CaaS service
- binding to a Facebook account
- took 3:08 minutes on average
- no mistakes made

Lab study with 100 students for Facebook message encryption

- no mistakes made

User study with 15 students for entire process

- no mistakes made

Compared with PKI/CA based approaches, CaaS is child's play

- registration & binding in minutes instead of hours or days

CaaS Conclusion

No need to trust Cloud or CaaS provider individually

- trust splitting allows for security / usability trade-off

By choosing CaaS provider in country X

- user is able to chose legal jurisdiction for data protection indecently of Cloud providers location(s)
- different jurisdiction add security since all locations need to cooperate
- multiple CaaS providers can be chained to add further protection

Helping Johnny 2.0 to Encrypt His Facebook Conversations

- Symposium on Usable Privacy and Security (SOUPS) 2012

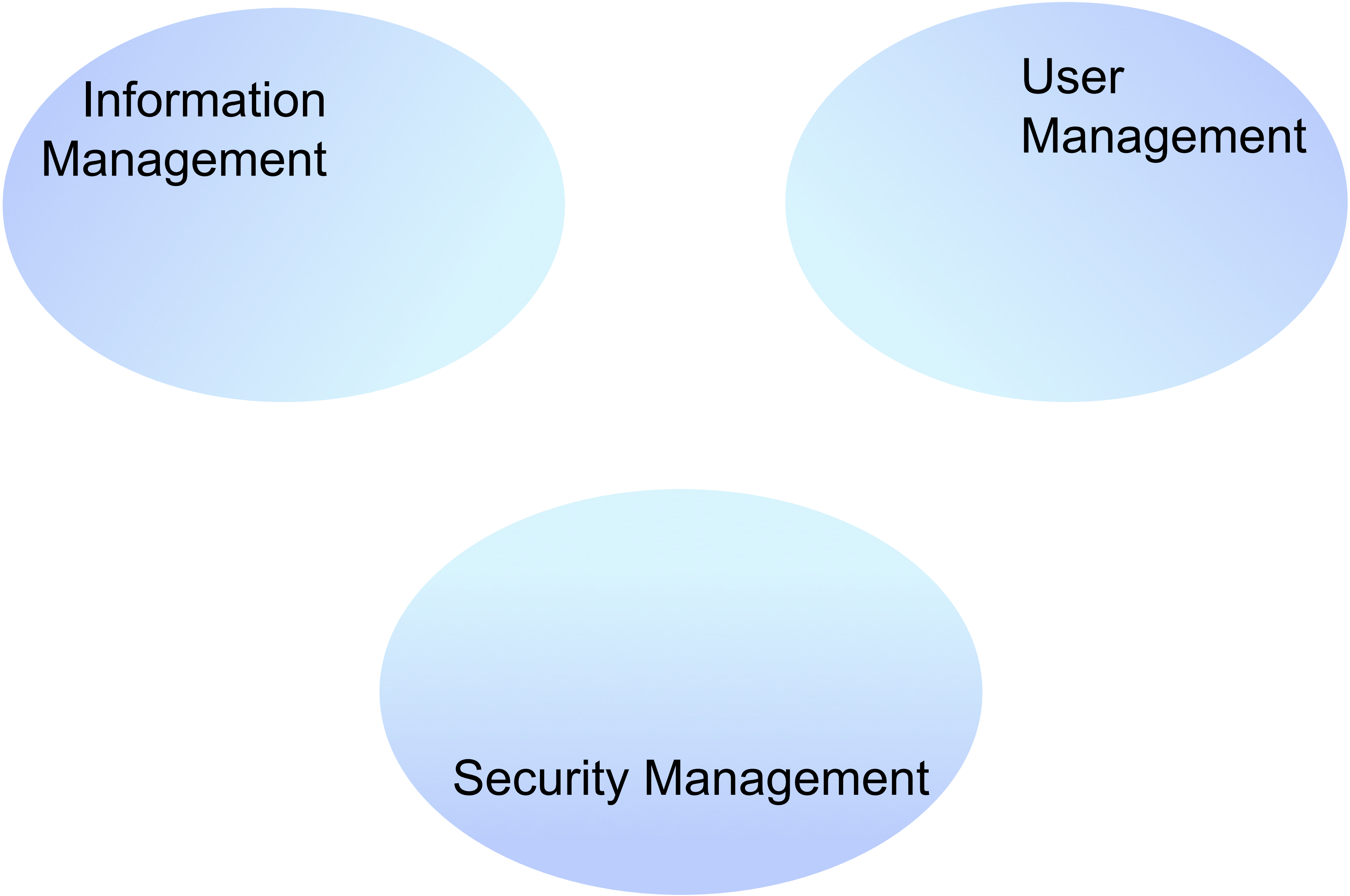
Mind Mesh

Funded by Wege in
die Forschung II
Search
Managing Information
the Human Way

Research Information Sharing

- How can research information be shared?
  - web servers, cloud storage, social networks etc.
- Confidentiality (crypto) is a key aspect for sensitive data
  - requires user expertise
  - cumbersome
  - error-prone
  - hard to fix
- How is information often shared?
  - e-Mail, DVDs, Skype, Print-Outs
- Why?
  - usability, security, usability of security

Problem



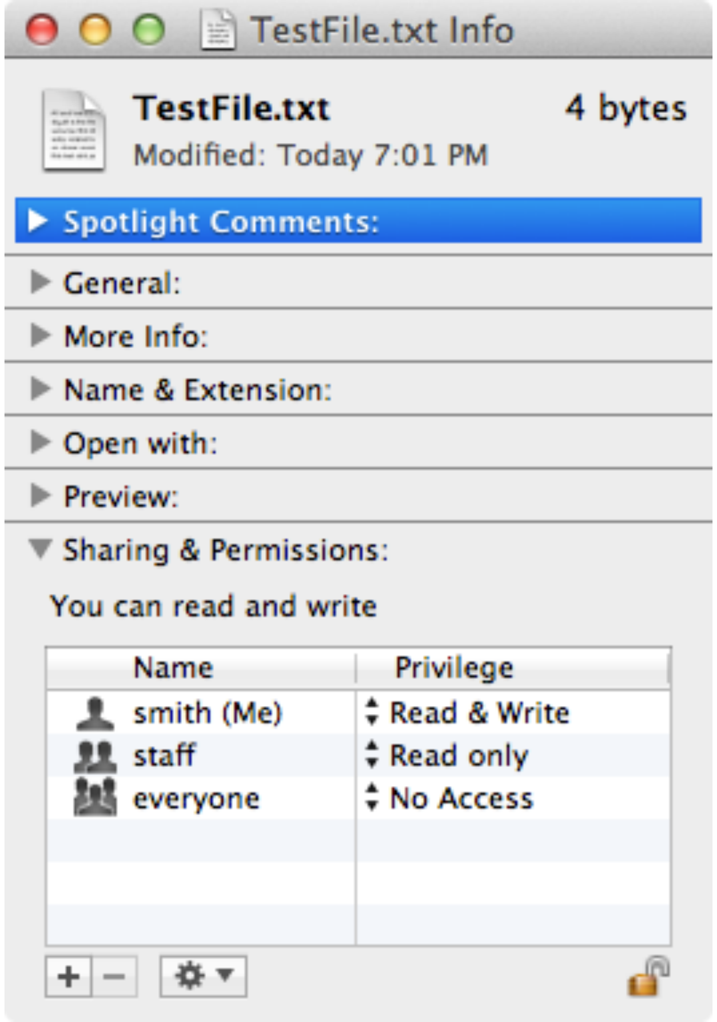
Information Management

User Management

Security Management

Access Question:

Who can access my file “TestFile.txt”?



TestFile.txt Info

TestFile.txt 4 bytes

Modified: Today 7:01 PM

▶ Spotlight Comments:

▶ General:

▶ More Info:

▶ Name & Extension:

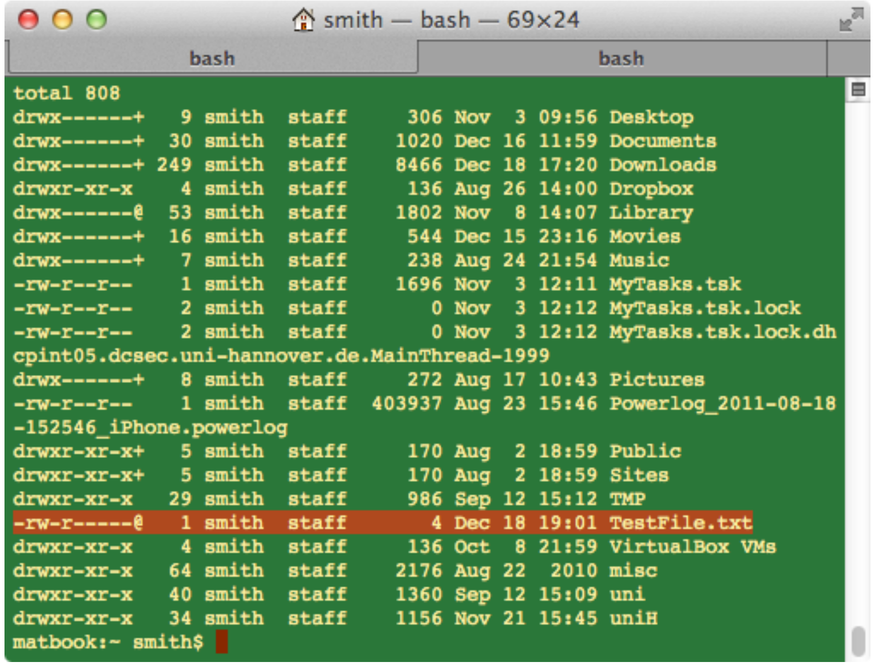
▶ Open with:

▶ Preview:

▼ Sharing & Permissions:

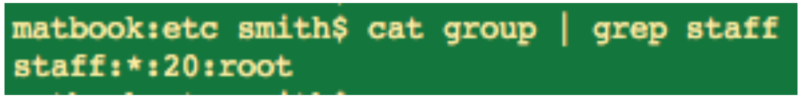
You can read and write

Name	Privilege
smith (Me)	Read & Write
staff	Read only
everyone	No Access



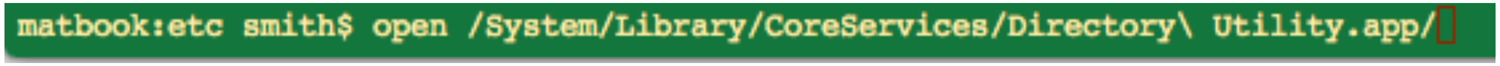
smith — bash — 69×24

```
total 808
drwx------+   9 smith  staff     306 Nov  3 09:56 Desktop
drwx------+  30 smith  staff    1020 Dec 16 11:59 Documents
drwx------+ 249 smith  staff    8466 Dec 18 17:20 Downloads
drwxr-xr-x    4 smith  staff     136 Aug 26 14:00 Dropbox
drwx------@  53 smith  staff    1802 Nov  8 14:07 Library
drwx------+  16 smith  staff     544 Dec 15 23:16 Movies
drwx------+   7 smith  staff     238 Aug 24 21:54 Music
-rw-r--r--    1 smith  staff    1696 Nov  3 12:11 MyTasks.tsk
-rw-r--r--    2 smith  staff       0 Nov  3 12:12 MyTasks.tsk.lock
-rw-r--r--    2 smith  staff       0 Nov  3 12:12 MyTasks.tsk.lock.dh
cpint05.dcsec.uni-hannover.de.MainThread-1999
drwx------+   8 smith  staff     272 Aug 17 10:43 Pictures
-rw-r--r--    1 smith  staff  403937 Aug 23 15:46 Powerlog_2011-08-18
-152546_iPhone.powerlog
drwxr-xr-x+   5 smith  staff     170 Aug  2 18:59 Public
drwxr-xr-x+   5 smith  staff     170 Aug  2 18:59 Sites
drwxr-xr-x   29 smith  staff     986 Sep 12 15:12 TMP
-rw-r-----@   1 smith  staff       4 Dec 18 19:01 TestFile.txt
drwxr-xr-x    4 smith  staff     136 Oct  8 21:59 VirtualBox VMs
drwxr-xr-x   64 smith  staff    2176 Aug 22  2010 misc
drwxr-xr-x   40 smith  staff    1360 Sep 12 15:09 uni
drwxr-xr-x   34 smith  staff    1156 Nov 21 15:45 uniH
matbook:~ smith$
```

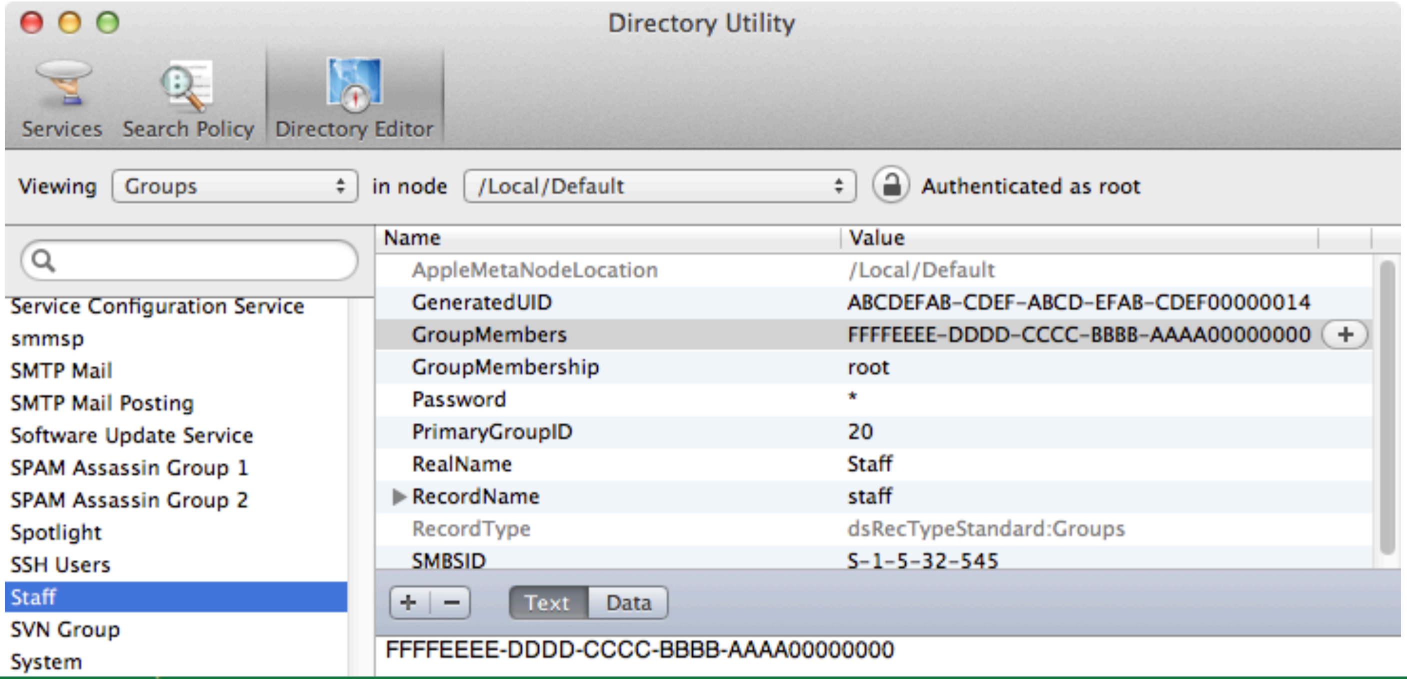


```
matbook:etc smith$ cat group | grep staff
staff:*:20:root
```

The quest for answers



```
matbook:etc smith$ open /System/Library/CoreServices/Directory\ Utility.app/
```

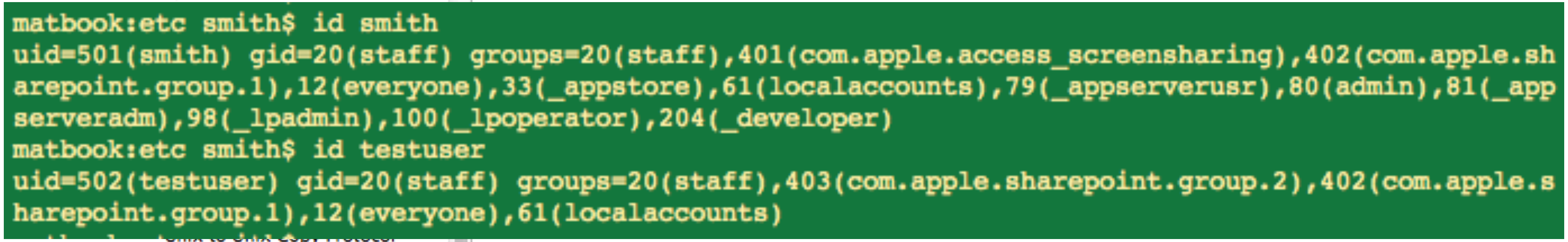


Viewing: Groups, in node: /Local/Default, Authenticated as root

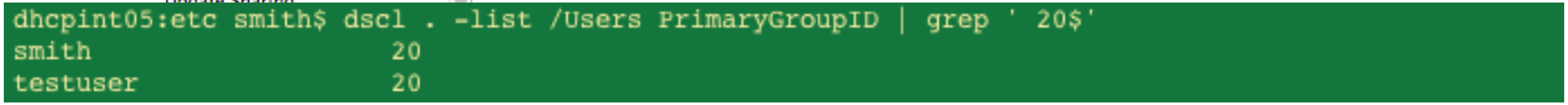
- Service Configuration Service
- smmsp
- SMTP Mail
- SMTP Mail Posting
- Software Update Service
- SPAM Assassin Group 1
- SPAM Assassin Group 2
- Spotlight
- SSH Users
- Staff
- SVN Group
- System
- Update Sharing

Name	Value
AppleMetaNodeLocation	/Local/Default
GeneratedUID	ABCDEFAB-CDEF-ABCD-EFAB-CDEF00000014
GroupMembers	FFFFEEEE-DDDD-CCCC-BBBB-AAAA00000000
GroupMembership	root
Password	*
PrimaryGroupID	20
RealName	Staff
▶ RecordName	staff
RecordType	dsRecTypeStandard:Groups
SMBSID	S-1-5-32-545

FFFFEEEE-DDDD-CCCC-BBBB-AAAA00000000



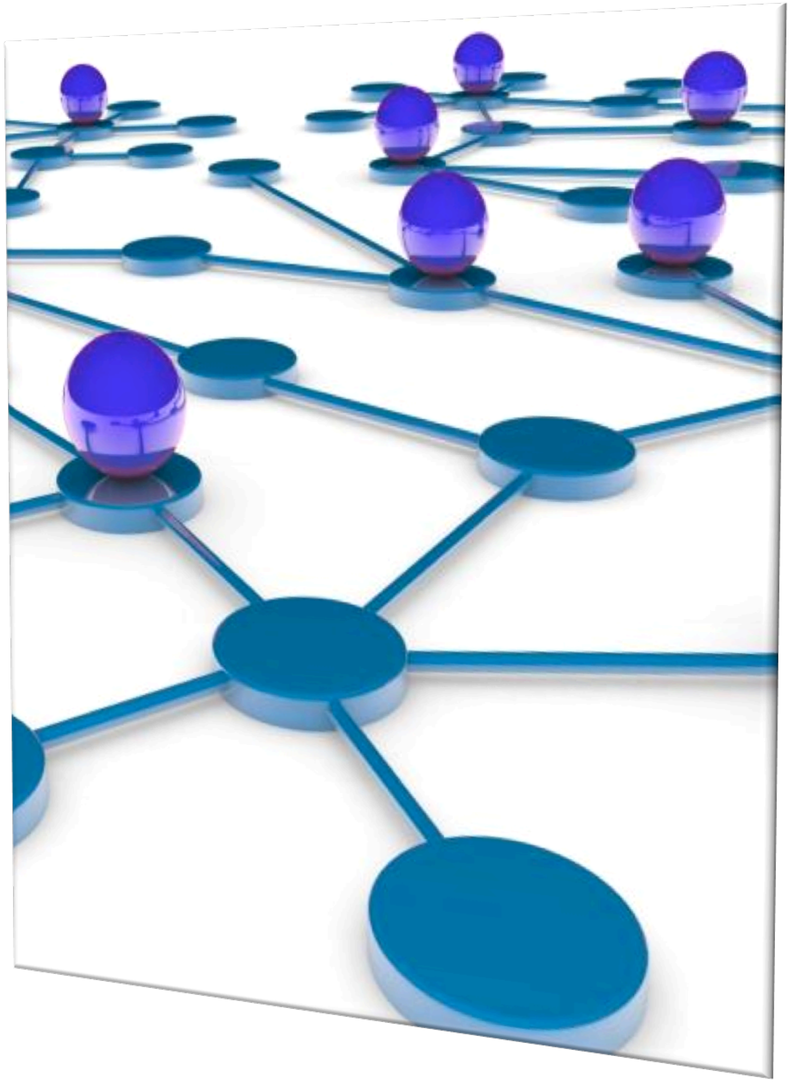
```
matbook:etc smith$ id smith
uid=501(smith) gid=20(staff) groups=20(staff),401(com.apple.access_screensharing),402(com.apple.sh
arepoint.group.1),12(everyone),33(_appstore),61(localaccounts),79(_appserverusr),80(admin),81(_app
serveradm),98(_lpadmin),100(_lpoperator),204(_developer)
matbook:etc smith$ id testuser
uid=502(testuser) gid=20(staff) groups=20(staff),403(com.apple.sharepoint.group.2),402(com.apple.s
harepoint.group.1),12(everyone),61(localaccounts)
```



```
dhcpint05:etc smith$ dscl . -list /Users PrimaryGroupID | grep ' 20$'
smith                   20
testuser                20
```

Add some distributed resources

- Each new resource can come with
  - new admin(s)
  - new users
  - new ways to access data
  - new security systems
  - new legal constraints
- Common approach:
  - call administrator



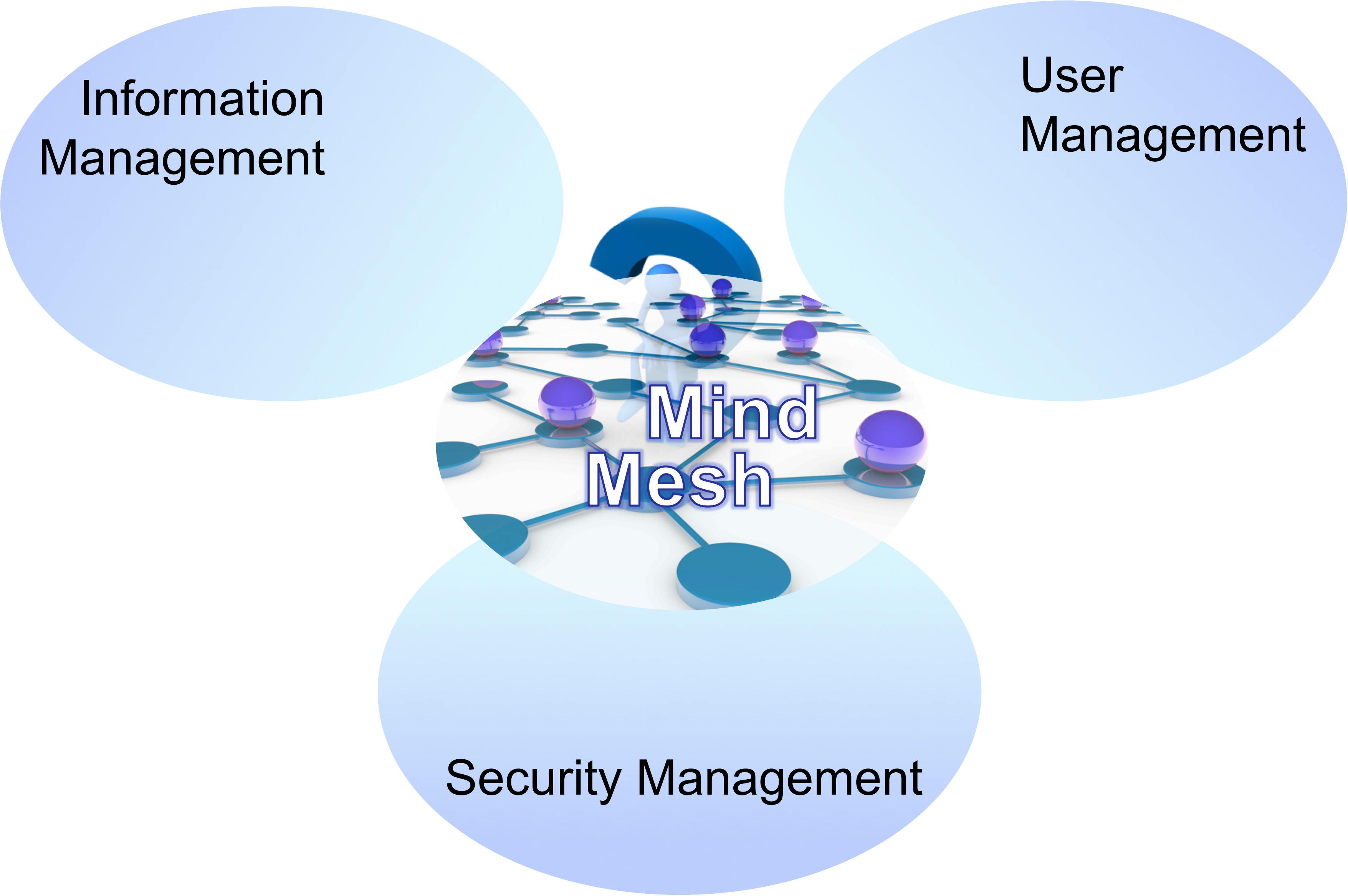
Distributed Systems example

- FlexLM based AC
  - config file:
    1673 lines of text
  - updated and tweaked over several years
  - by several administrators
  - exemplary logging of action in GIT repository

```
#
======================================================================
=====
# NAME xxx
# KONTAKT 1xxx
# E-Mail: xxx
# Vertr. Nr.: xxx
# EIGENTLICH 130.75.xxx.[xxx-xxx]
HOST_GROUP xxx\
130.75.65.xxx 130.75.65.xxx 130.75.65.xxx 130.75.65.xxx 130.75.65.xxx \
130.75.65.xxx 130.75.65.xxx 130.75.65.xxx 130.75.65.xxx 130.75.65.xxx \
130.75.65.xxx 130.75.65.xxx 130.75.65.xxx 130.75.65.xxx 130.75.65.xxx \
130.75.65.xxx 130.75.65.xxx 130.75.65.xxx 130.75.65.xxx 130.75.65.xxx \
130.75.65.xxx 130.75.65.xxx 130.75.65.xxx 130.75.65.xxx 130.75.65.xxx \
130.75.65.xxx 130.75.65.xxx 130.75.65.xxx
INCLUDE    MATLAB:asset_info=49487                HOST_GROUP  xxx
#MAX      6 MATLAB:asset_info=49487               HOST_GROUP  xxx
INCLUDE    Image_Toolbox:asset_info=49487         HOST_GROUP  xxx
#MAX      1 Image_Toolbox:asset_info=49487        HOST_GROUP  xxx
#
======================================================================
=====
# NAME xxx
# KONTAKT xxx
# E-Mail: xxx
# Vertr. Nr.: xxx
HOST_GROUP xxx 130.75.26.*
INCLUDE    Wavelet_Toolbox:asset_info=49487       HOST_GROUP  xxx
#MAX      1 Wavelet_Toolbox:asset_info=49487      HOST_GROUP  xxx
INCLUDE    Symbolic_Toolbox:asset_info=49487      HOST_GROUP  xxx
#MAX      1 Symbolic_Toolbox:asset_info=49487     HOST_GROUP  xxx
INCLUDE    PDE_Toolbox:asset_info=49487           HOST_GROUP  xxx
#MAX      1 PDE_Toolbox:asset_info=49487          HOST_GROUP  xxx
INCLUDE    MATLAB:asset_info=49487                HOST_GROUP  xxx
#MAX      2 MATLAB:asset_info=49487               HOST_GROUP  xxx
#
======================================================================
=====
```

```
User xxx fuer VPN bei xxx gesetzt
xxx Matlab wieder reduziert auf 35
Neuer Eintrag fuer xxx
VPN-Host bei xxx hinzugefuegt
Subnetz von xxx getrennt
Neuer Eintrag fuer xxx xxx
xxx-cip Schnipsel in license.dat
aktualisiert
```

Sec. Inf. Management - Problem



Information Management

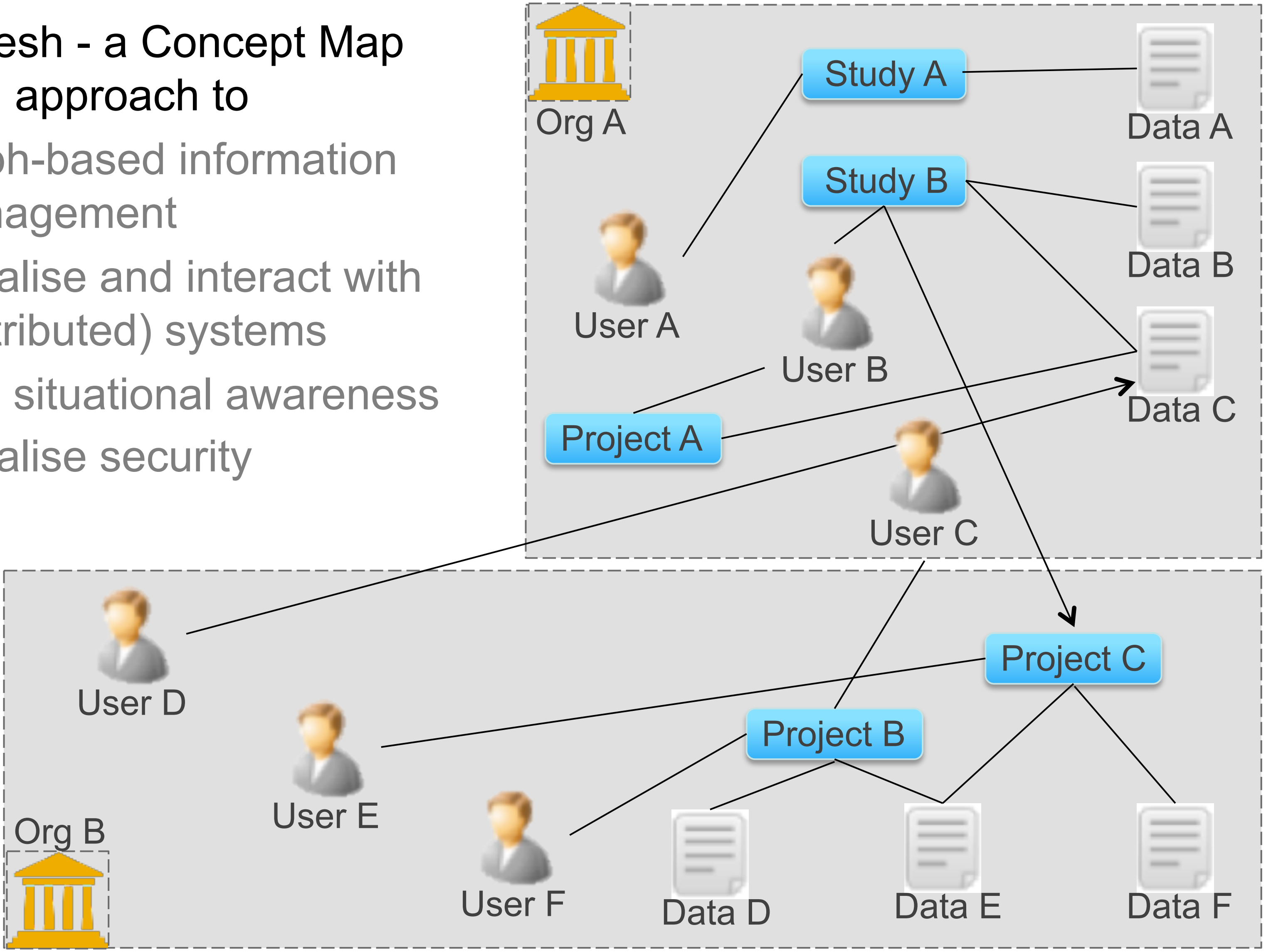
User Management

Mind Mesh

Security Management

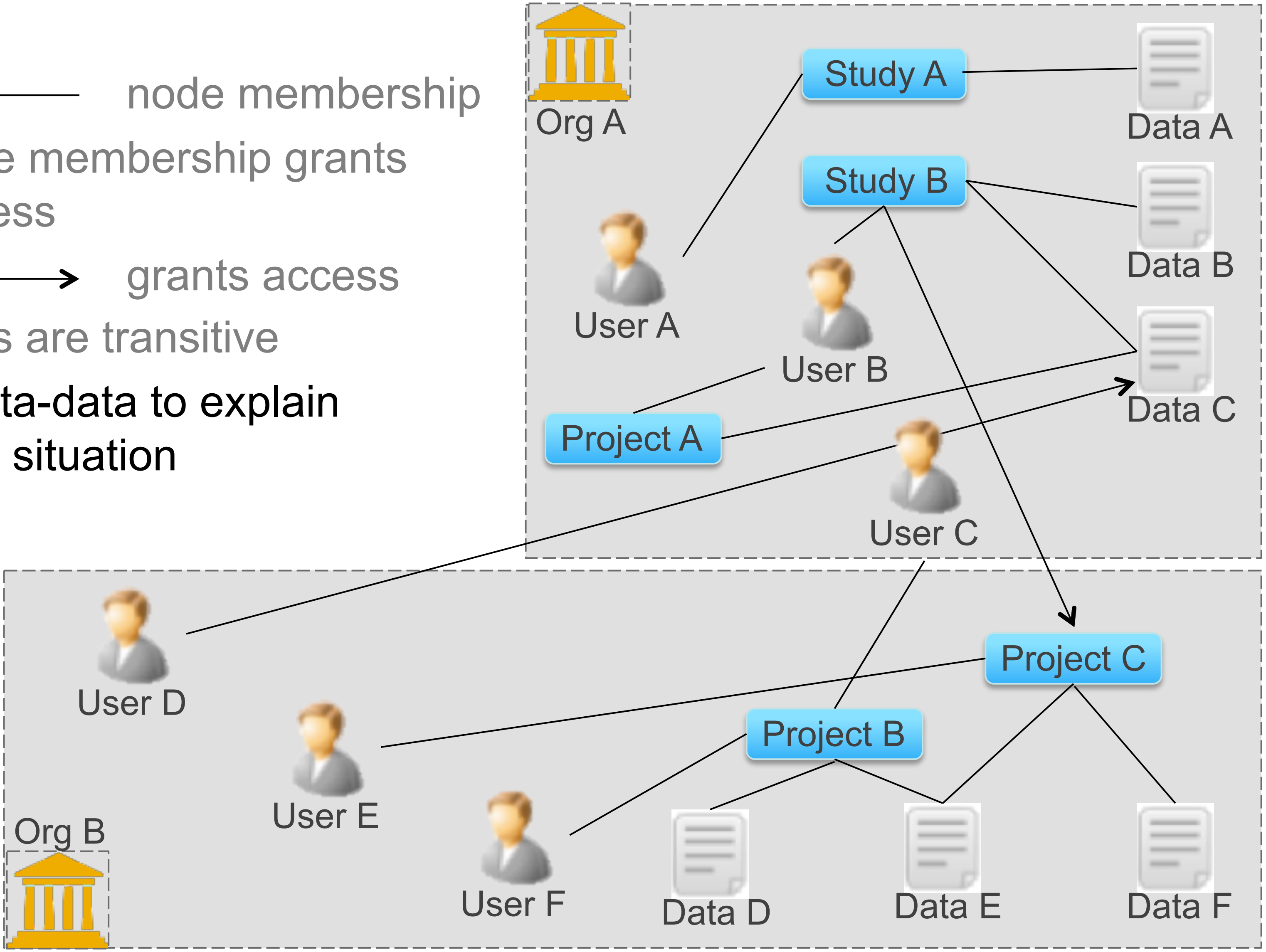
Mind Mesh

- Mind Mesh - a Concept Map inspired approach to
  - graph-based information management
  - visualise and interact with (distributed) systems
  - gain situational awareness
  - visualise security



Mind Mesh

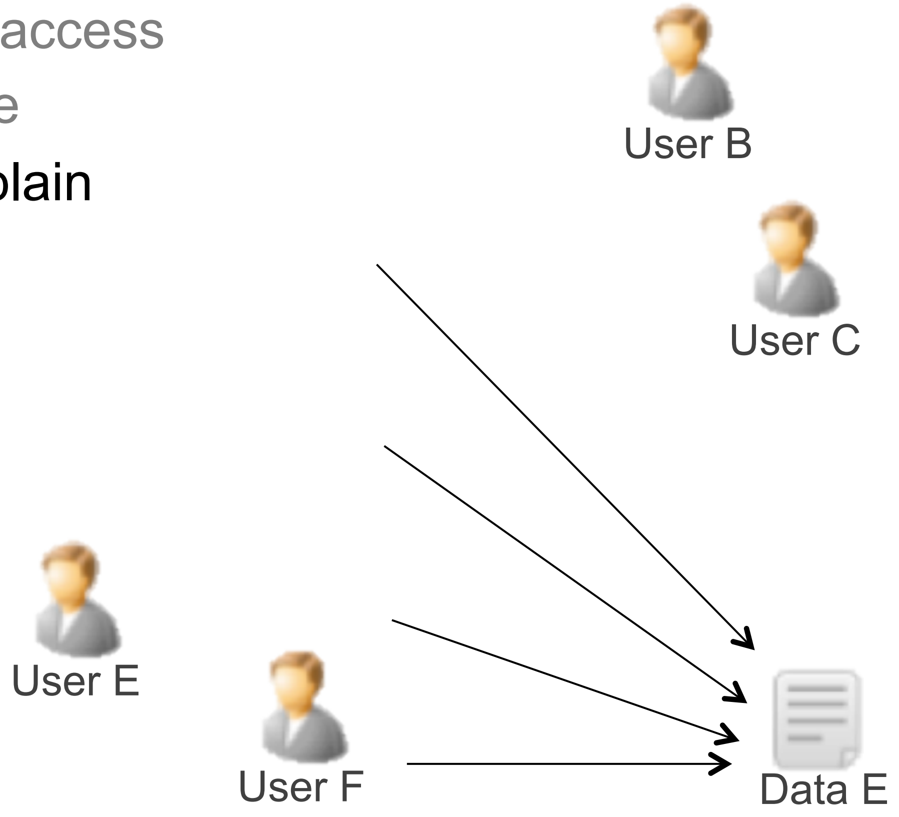
- Rules
  - ———— node membership
  - node membership grants access
  - ——→ grants access
  - rules are transitive
- Use meta-data to explain security situation



Mind Mesh

- Rules
  - ——— node membership
  - node membership grants access
  - ——→ grants access
  - rules are transitive
- Use meta-data to explain security situation

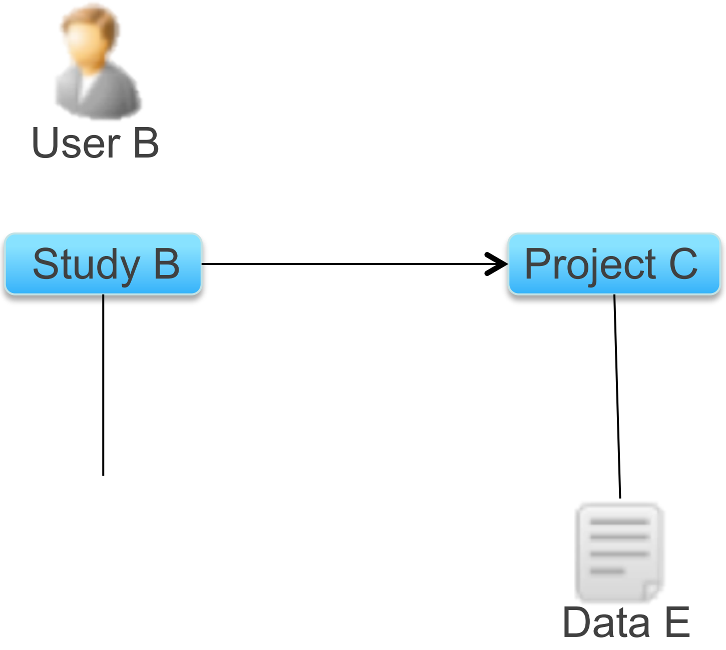
Who has access to Data E?



Mind Mesh

- Rules
  - ——— node membership
  - node membership grants access
  - ⟶ grants access
  - rules are transitive
- Use meta-data to explain security situation

Who has access to Data E?

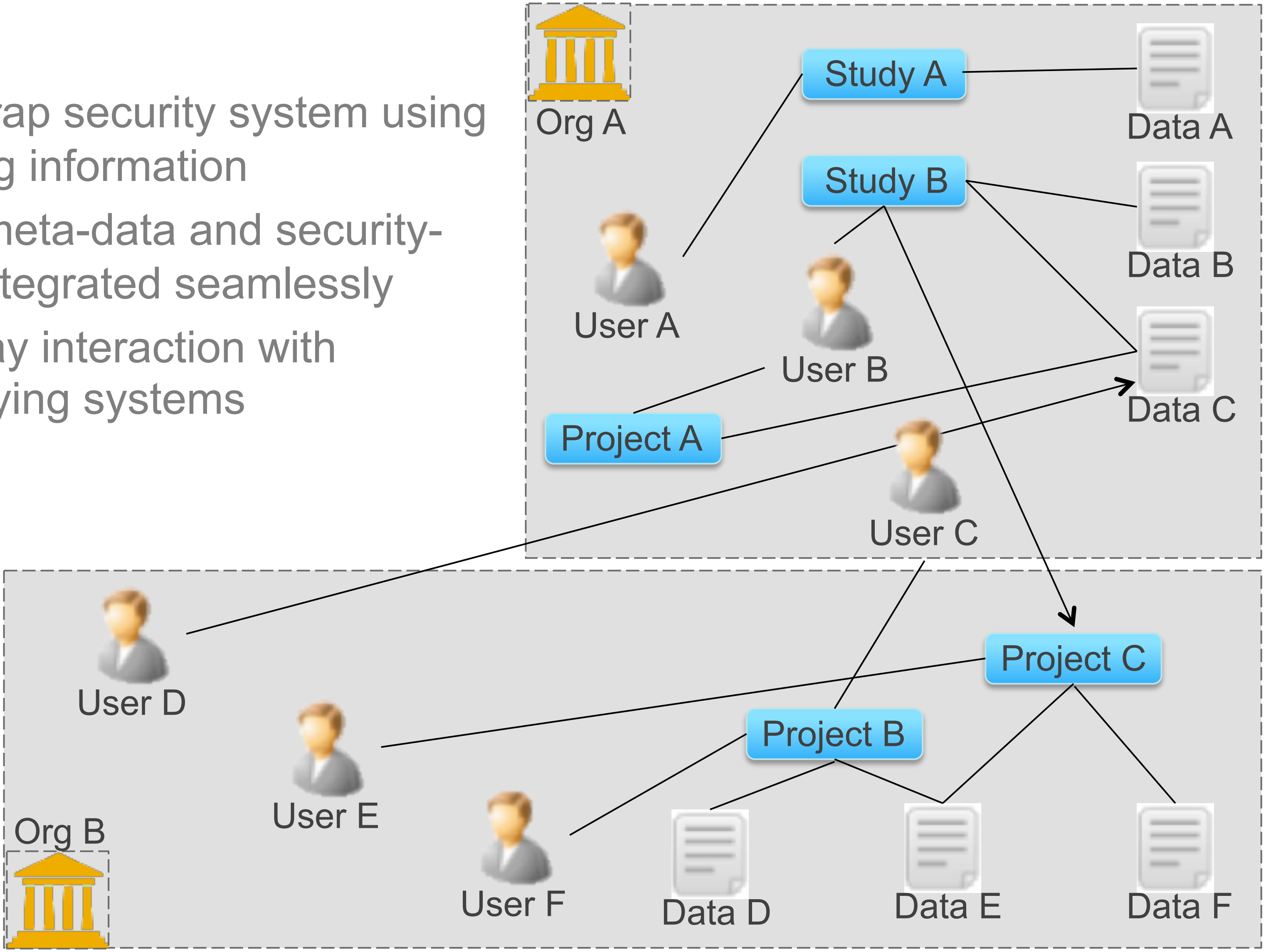


Why does User B have access to Data E?

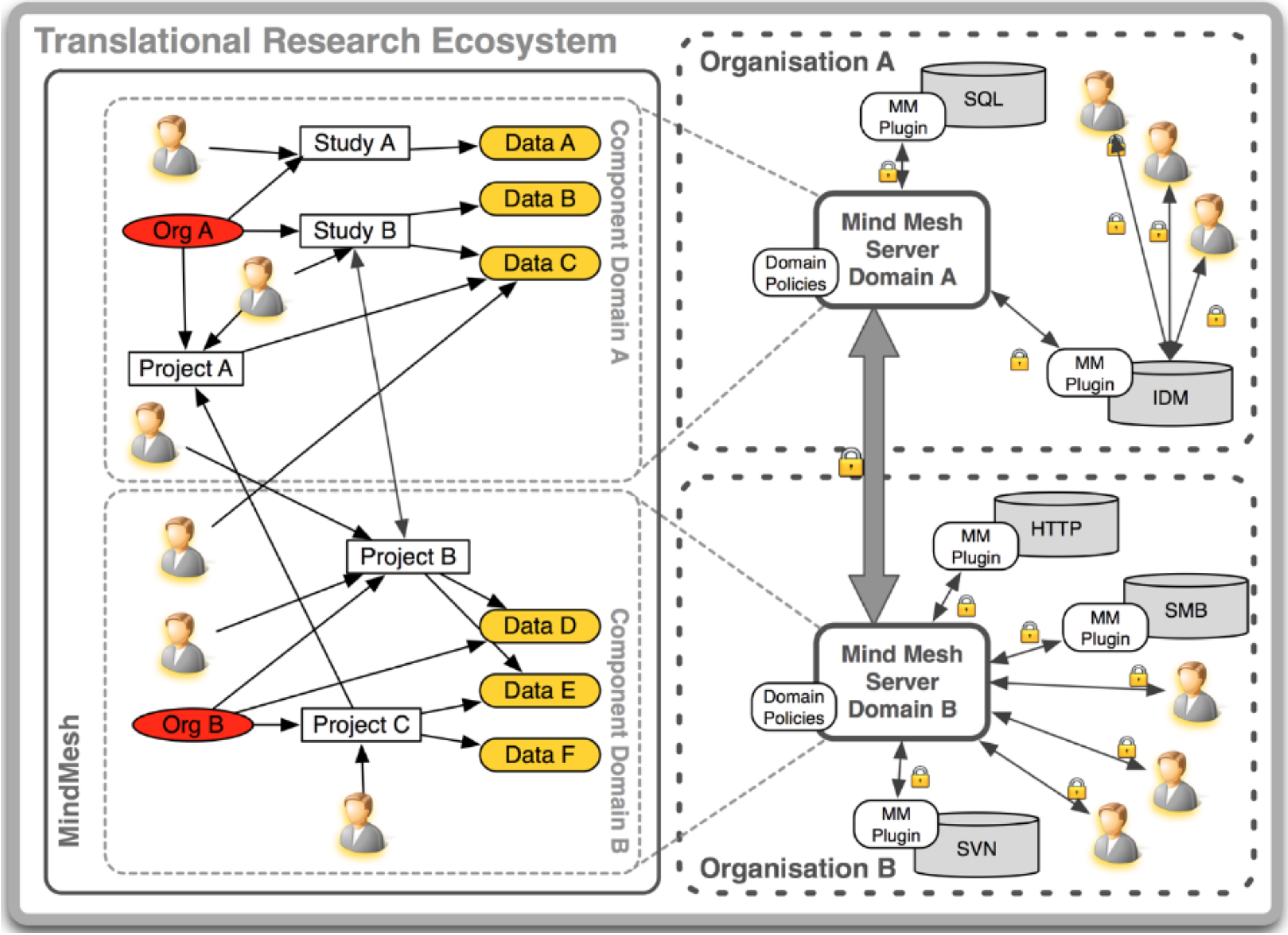
Mind Mesh

Features

- bootstrap security system using existing information
- data, meta-data and security-data integrated seamlessly
- two-way interaction with underlying systems



MindMesh Plugins

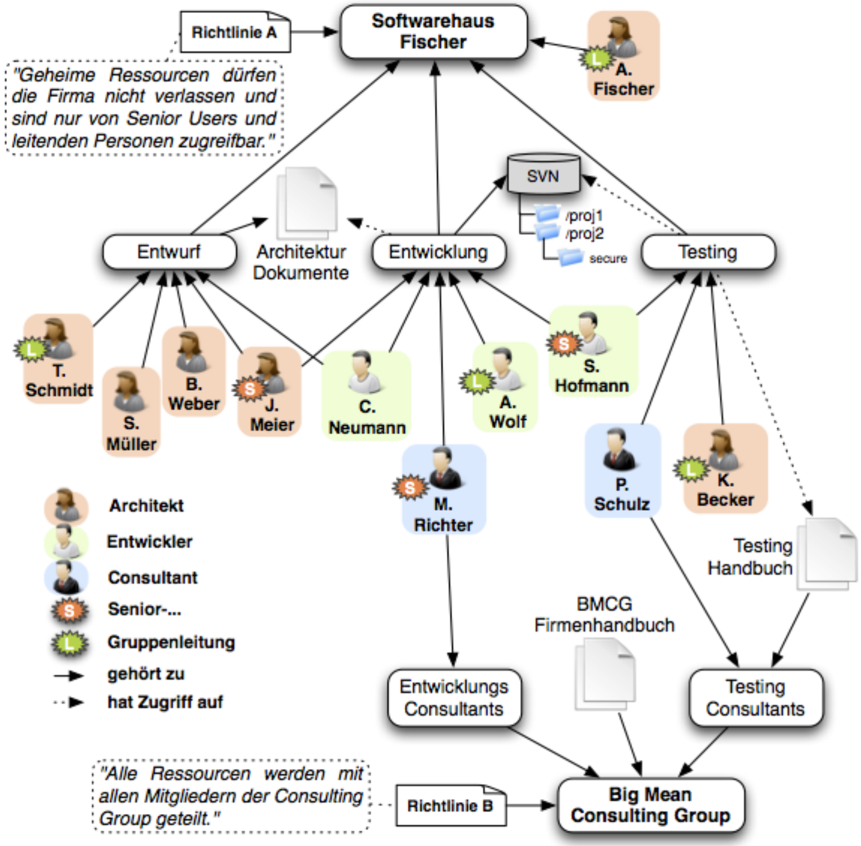


Questionnaire

Abteilungen des Softwarehaus Fischer = Entwurf, Entwicklung, Testing
Geschäftsführer des Softwarehaus Fischer = A. Fischer (A)
Entwurf = T. Schmidt (A), S. Müller (A), B. Weber (A), J. Meier (SA), C. Neumann (E)
Entwicklung = J. Meier (SA), C. Neumann (E), M. Richter (SE)(C), A. Wolf (E), S. Hofmann (SE)
Testing = S. Hofmann (SE), P. Schulz (C), K. Becker (A)

Big Mean Consulting Group = Entwicklungs Consultants, Testing Consultants
Development Consultants = M. Richter (C)
Testing Consultants = P. Schulz (C)

Person → Resource	Zugriff?	
	Ja	Nein
M. Richter → BMCG Firmenhandbuch	□	□
M. Richter → Testing Handbuch	□	□
K. Becker → Testing Handbuch	□	□
K. Becker → BMCG Firmenhandbuch	□	□
A. Wolf → BMCG Firmenhandbuch	□	□
A. Wolf → Testing Handbuch	□	□

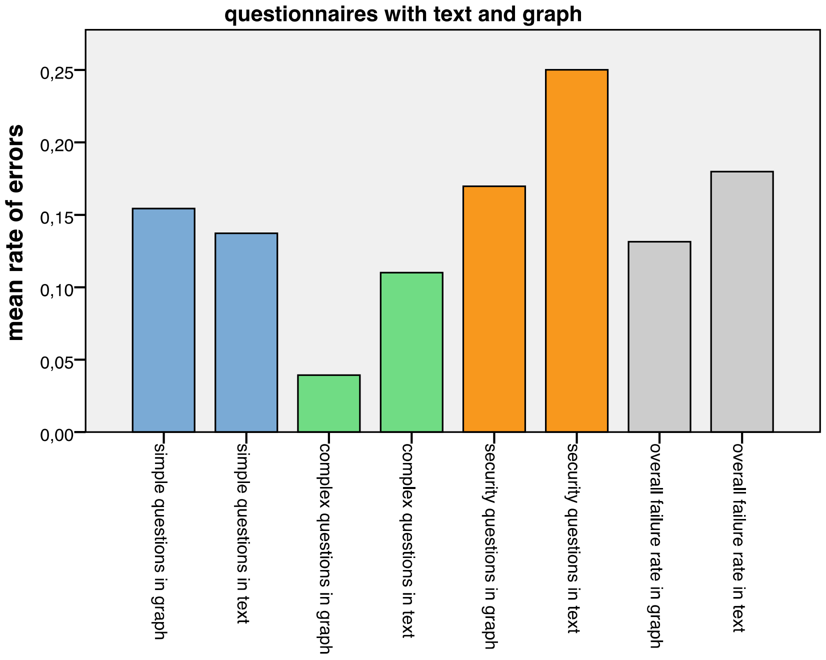


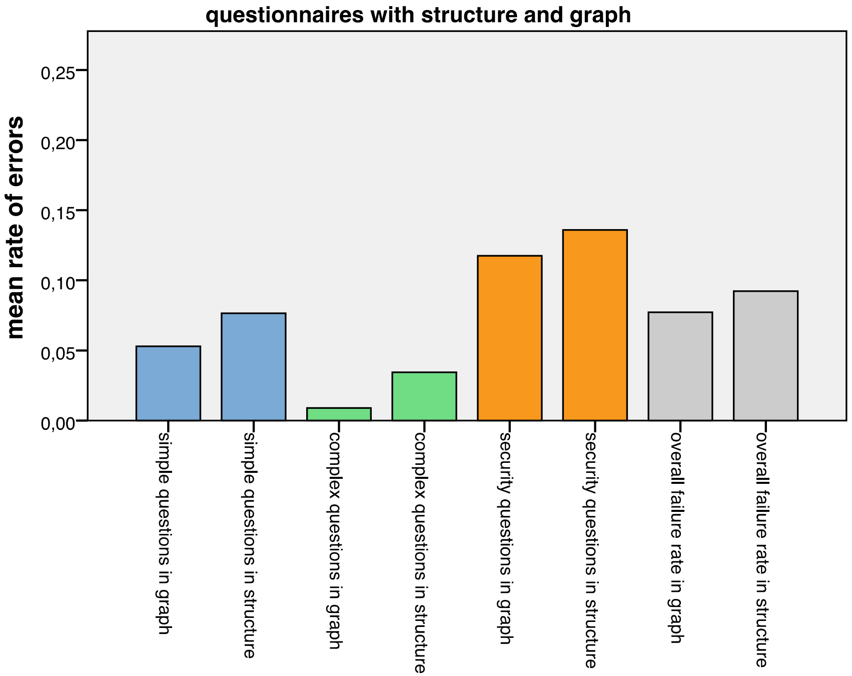
"Geheime Ressourcen dürfen die Firma nicht verlassen und sind nur von Senior Users und leitenden Personen zugreifbar."

"Alle Ressourcen werden mit allen Mitgliedern der Consulting Group geteilt."

Questionnaire Results

- Graphical representation leads to less security mistakes
- Students felt the graph was easier to understand and nicer to work with
- Students had a higher confidence that the answers based on the graph were correct compared to the text representation





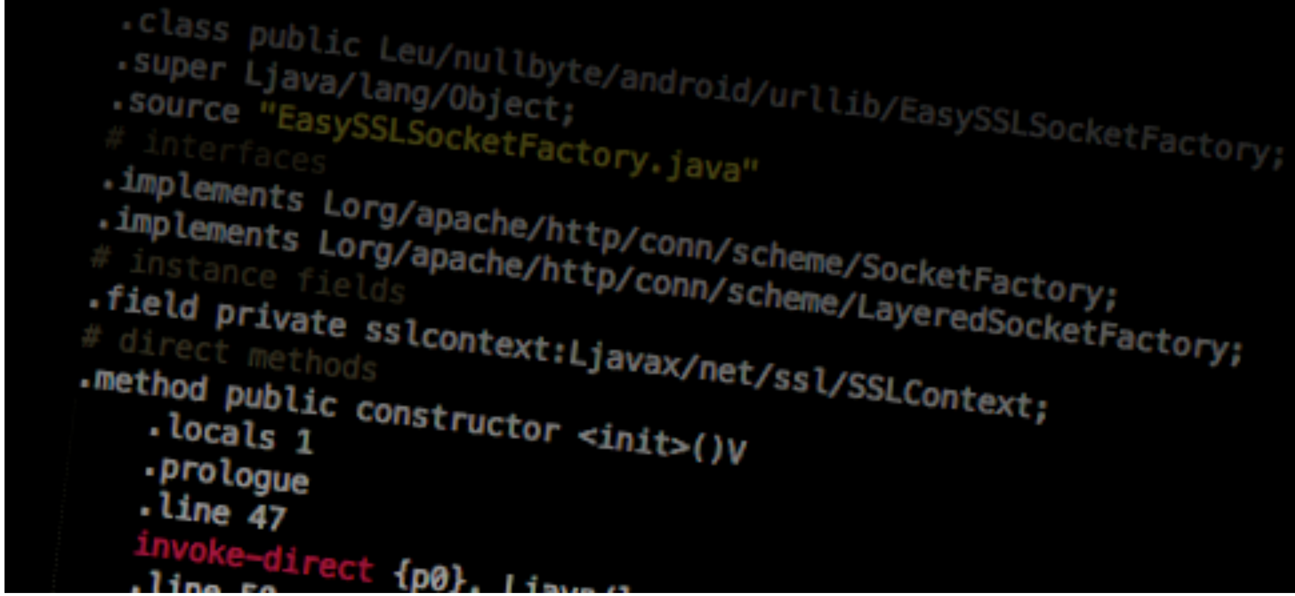
Half Time

This was the user-side of things…

Now let’s have a look at developer issues

Why Eve and Mallory Love Android An Analysis of Android SSL (In)Security

- and a call for Usable Security for Developers



```
.class public Leu/nullbyte/android/urllib/EasySSLSocketFactory;
.super Ljava/lang/Object;
.source "EasySSLSocketFactory.java"
# interfaces
.implements Lorg/apache/http/conn/scheme/SocketFactory;
.implements Lorg/apache/http/conn/scheme/LayeredSocketFactory;
# instance fields
.field private sslcontext:Ljavax/net/ssl/SSLContext;
# direct methods
.method public constructor <init>()V
    .locals 1
    .prologue
    .line 47
    invoke-direct {p0}, Ljava/
    .line 5
```

Sascha Fahl

Marian Harbach

Thomas Muders

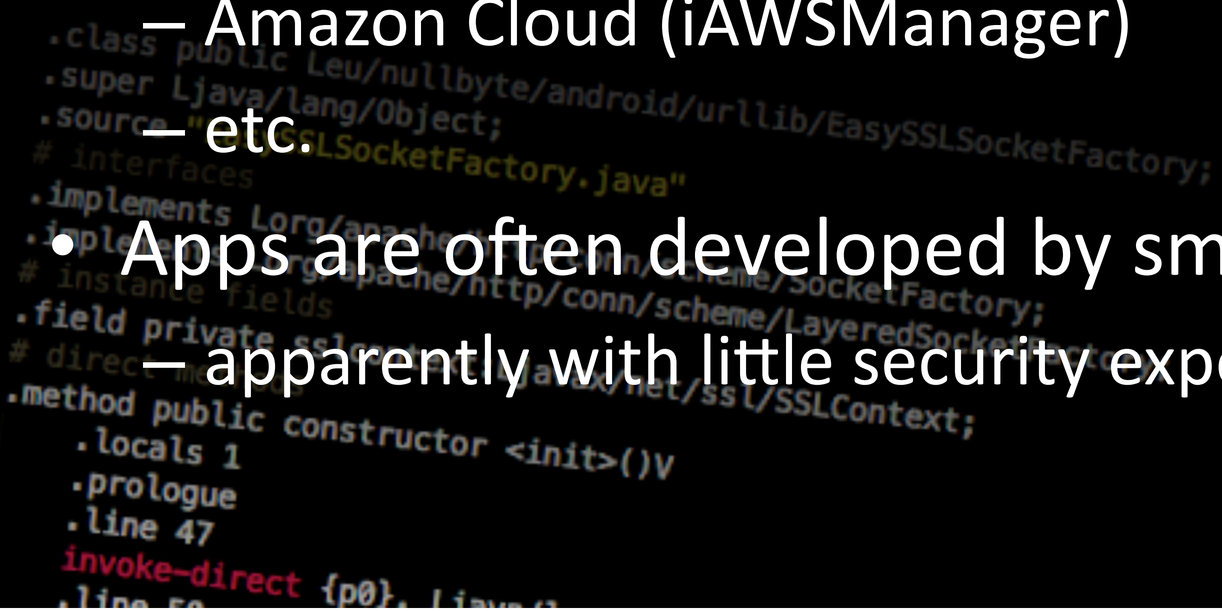
Lars Baumgärtner

Bernd Freisleben

Matthew Smith

Appification

- 50% of phones are smartphone
- Cloud services are often wrapped in Apps
  - Dropbox
  - Facebook
  - Amazon Cloud (iAWSManager)
  - etc.
- Apps are often developed by small teams
  - apparently with little security expertise ;)



```
.class public Leu/nullbyte/android/urllib/EasySSLSocketFactory;
.super Ljava/lang/Object;
.source "EasySSLSocketFactory.java"
# interfaces
.implements Lorg/apache/http/conn/scheme/SocketFactory;
.implements Lorg/apache/http/conn/scheme/LayeredSocketFactory;
# instance fields
.field private sslcontext:Ljavax/net/ssl/SSLContext;
# direct methods
.method public constructor <init>()V
    .locals 1
    .prologue
    .line 47
    invoke-direct {p0}, Ljava/
```

Some Android Facts

- 59% smartphone market share
- 331 million devices (as of Q1 2012)
- 934,000 activations per day (as of Q1 2012)
- 450,000 apps (as of June 2012)
- Also used on tablets, TVs and within cars
- It’s Open Source

```
.class public
.super Ljava/lang/Object;          e/android/urllib/EasySSLSocketFactory;
.source "EasySSLSocketFactory
# interfaces
.implements Lorg/apache/http/conn/scheme/SocketFactory;
.implements Lorg/apache/http/conn/scheme/LayeredSocketFactory;
# instance fields
.field private sslcontext:Ljavax/net/ssl/SSLContext;
# direct methods
.method public constructor <init>()V
    .locals 1
    .prologue
    .line 47
    invoke-direct {p0}, Ljava/l
```

What do Cloud apps have in common?

All share data over the Internet

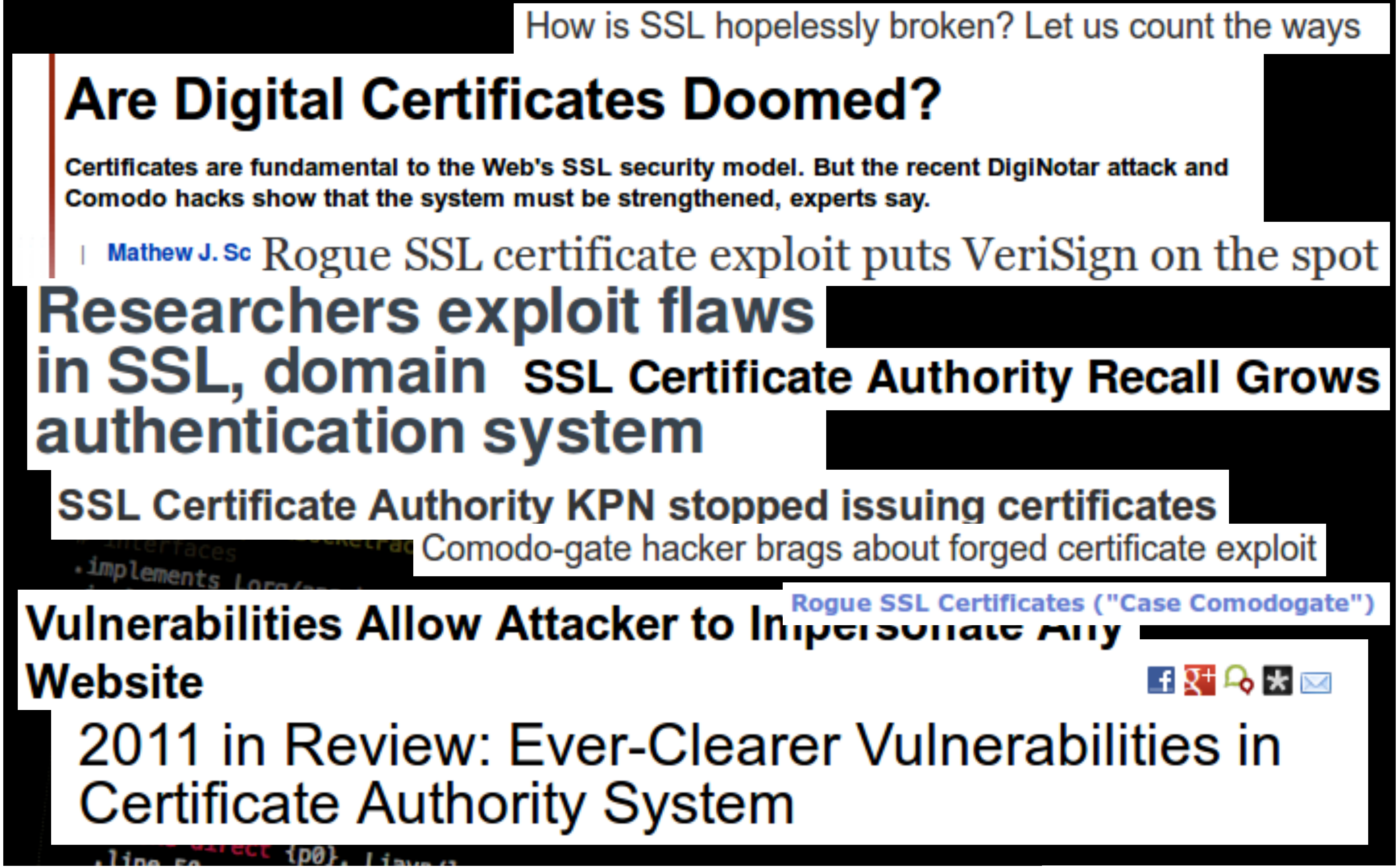
Some of them even „secure“ transfer using:

```
.class public Leu/nullbyte/android/urllib/EasySSLSocketFactory;
.super Ljava/lang/Object;
.source "EasySSLSocketFactory.java"
# interfaces
.implements Lorg/apache/http/conn/scheme/SocketF
.implements Lorg/apache/http/conn/scheme/LayeredSocketFactory;
# instance fields
.field private sslcontext:Ljavax/net/ssl/SSLContext;
# direct methods
.method public constructor <init>()V
    .locals 1
    .prologue
    .line 47
    invoke-direct {p0}, Ljava/l
    .line 5
```

SSL

(Secure Sockets Layer protocol)
(Transport Layer Security **(TLS)** protocol)

All quiet on the SSL front?



How is SSL hopelessly broken? Let us count the ways

Are Digital Certificates Doomed?

Certificates are fundamental to the Web's SSL security model. But the recent DigiNotar attack and Comodo hacks show that the system must be strengthened, experts say.

| Mathew J. Sc

Rogue SSL certificate exploit puts VeriSign on the spot

Researchers exploit flaws in SSL, domain authentication system

SSL Certificate Authority Recall Grows

SSL Certificate Authority KPN stopped issuing certificates

interfaces

.implements Lorg/

Comodo-gate hacker brags about forged certificate exploit

Vulnerabilities Allow Attacker to Impersonate Any Website

Rogue SSL Certificates ("Case Comodogate")

2011 in Review: Ever-Clearer Vulnerabilities in Certificate Authority System

direct {p0}, Ljava/

.line 50

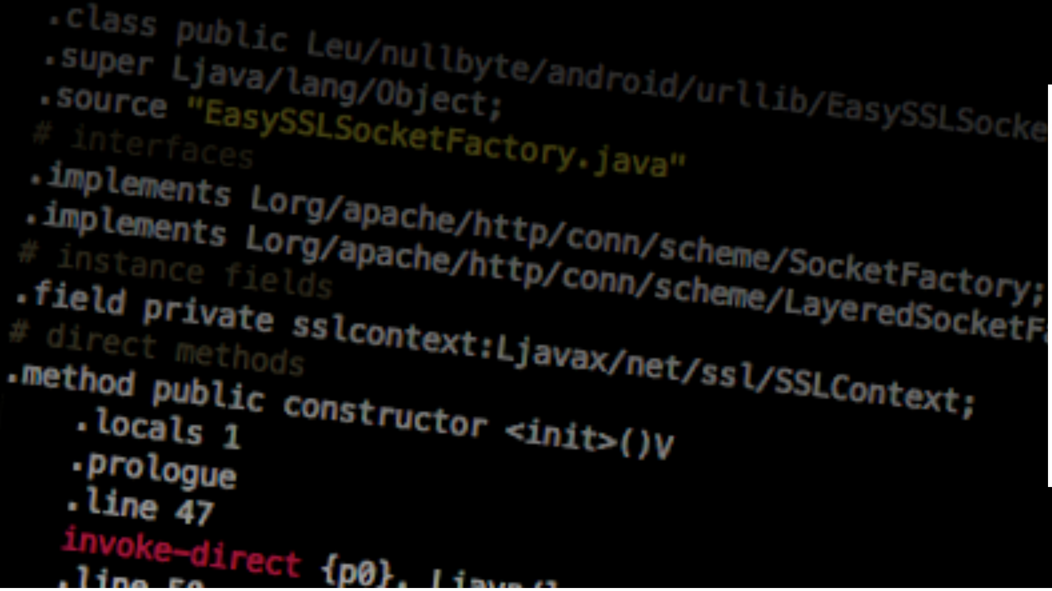
S**T HAPPENS

But sometimes you wish it happened to someone else.

```
  .class
  .super
  .source
 # inter
 .implem
 .impleme
# instan
.field p
# direct
.method p
    .locals 1                     <init>()V
    .prologue
    .line 47
    invoke-direct {p0}, Ljava/l
    .line 5
```

SSL misuse

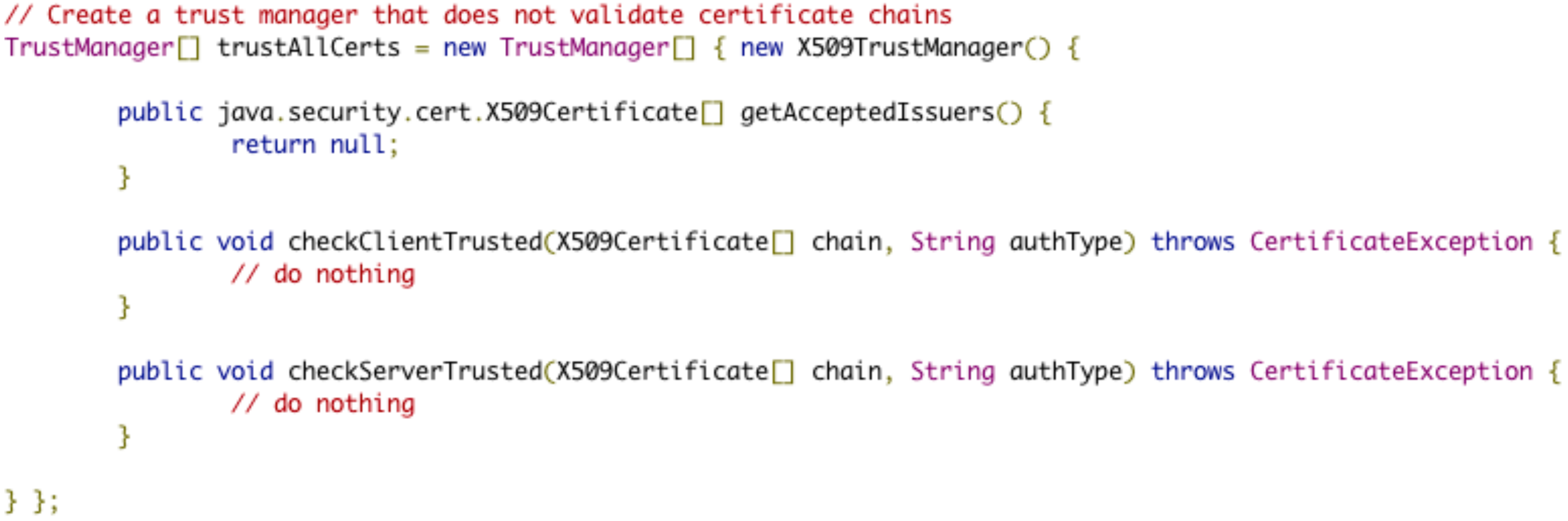
- Trusting all certificates
- Allowing all hostnames
- Trusting (too) many CAs
- Mixed mode/no SSL



```
.class public Leu/nullbyte/android/urllib/EasySSLSocke
.super Ljava/lang/Object;
.source "EasySSLSocketFactory.java"
# interfaces
.implements Lorg/apache/http/conn/scheme/SocketFactory;
.implements Lorg/apache/http/conn/scheme/LayeredSocketF
# instance fields
.field private sslcontext:Ljavax/net/ssl/SSLContext;
# direct methods
.method public constructor <init>()V
    .locals 1
    .prologue
    .line 47
    invoke-direct {p0}, Ljava/l
    .line 50
```

Trusting all Certificates

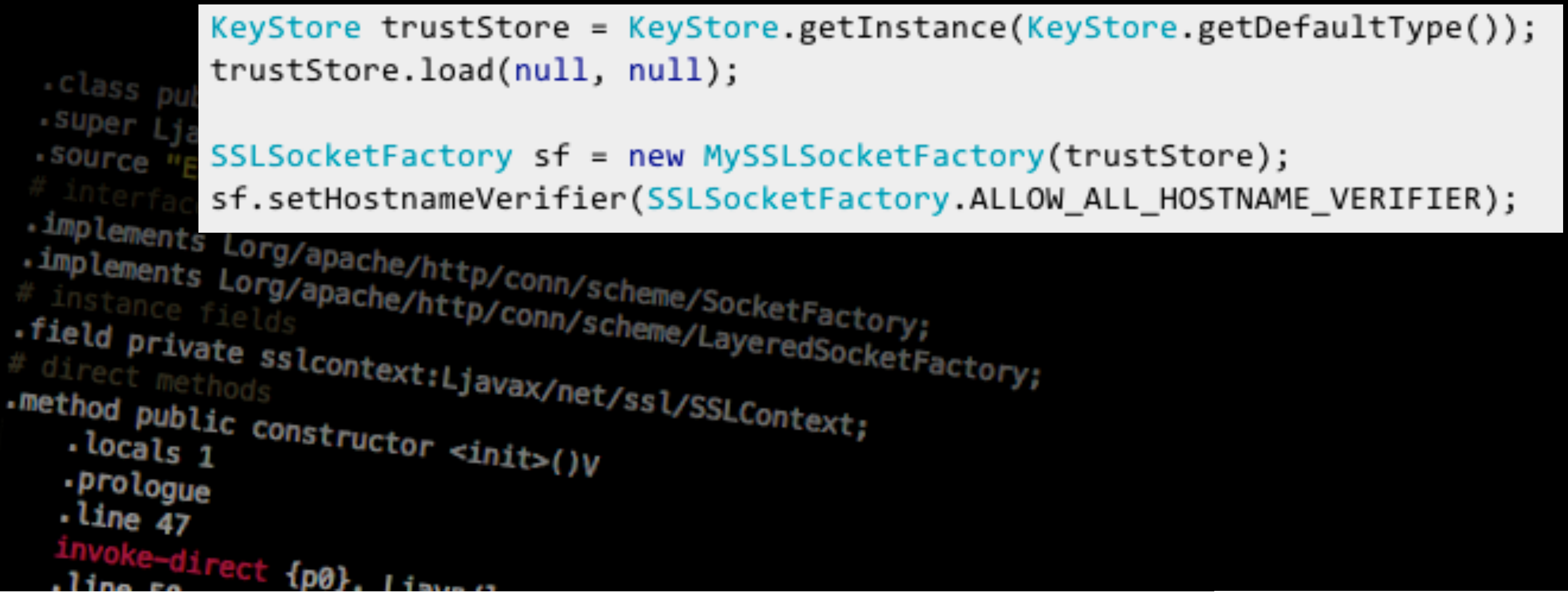
- Correct SSL certificate validation is so easy
  - Only a (commercial) trusted CA signed certificate required
- What some Apps do:



```
// Create a trust manager that does not validate certificate chains
TrustManager[] trustAllCerts = new TrustManager[] { new X509TrustManager() {

        public java.security.cert.X509Certificate[] getAcceptedIssuers() {
                return null;
        }

        public void checkClientTrusted(X509Certificate[] chain, String authType) throws CertificateException {
                // do nothing
        }

        public void checkServerTrusted(X509Certificate[] chain, String authType) throws CertificateException {
                // do nothing
        }

} };
```

Allowing all Hostnames

- What other Apps do:
  - Check CA signature, but allow mallory.com for google.com

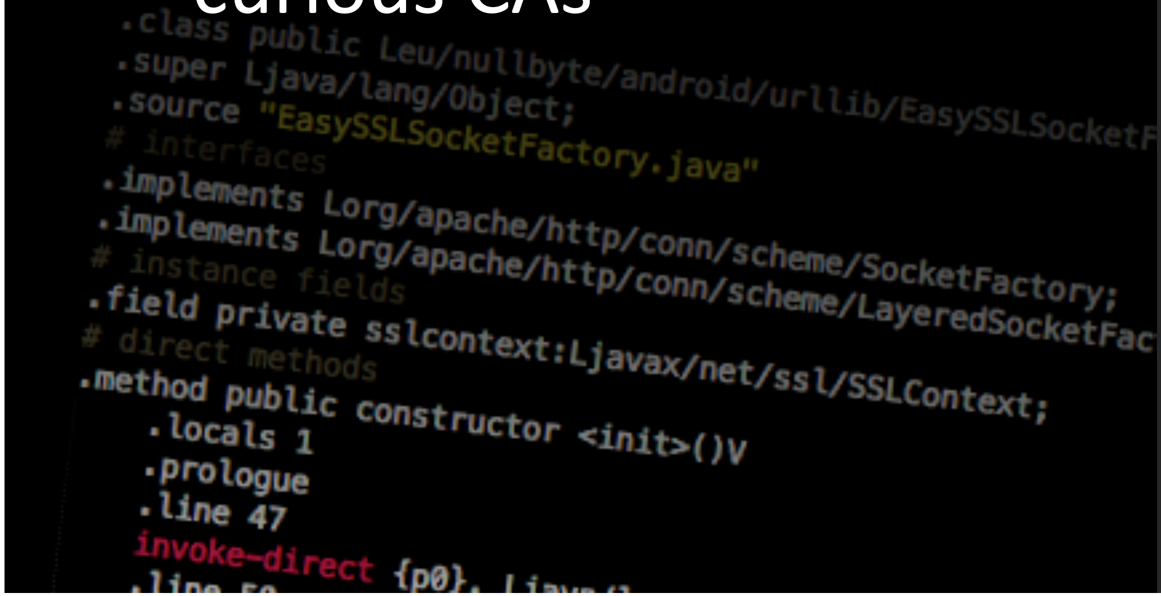


```
KeyStore trustStore = KeyStore.getInstance(KeyStore.getDefaultType());
trustStore.load(null, null);

SSLSocketFactory sf = new MySSLSocketFactory(trustStore);
sf.setHostnameVerifier(SSLSocketFactory.ALLOW_ALL_HOSTNAME_VERIFIER);
```

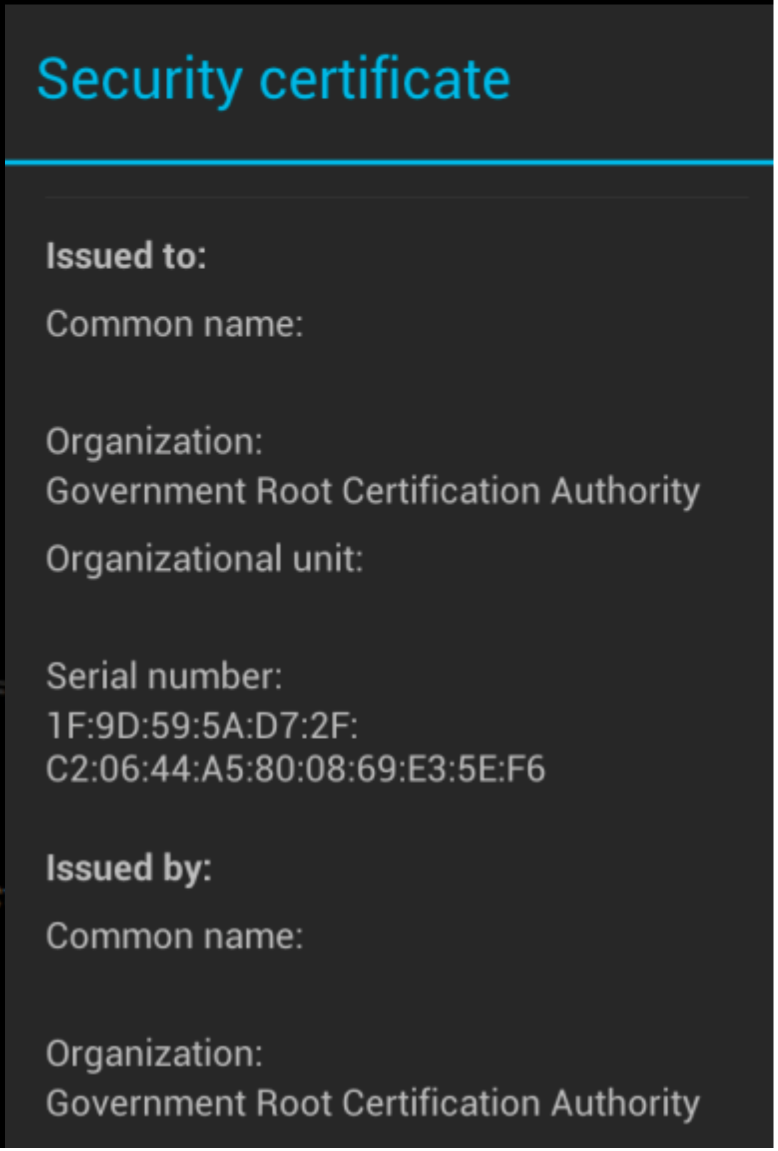
```
.class pu
.super Lja
.source "E
# interfac
.implements Lorg/apache/http/conn/scheme/SocketFactory;
.implements Lorg/apache/http/conn/scheme/LayeredSocketFactory;
# instance fields
.field private sslcontext:Ljavax/net/ssl/SSLContext;
# direct methods
.method public constructor <init>()V
    .locals 1
    .prologue
    .line 47
    invoke-direct {p0}, Ljava/l
```

Trusting many CAs

- By default Android trusts 164 different CAs
- Some are even really curious CAs



```
.class public Leu/nullbyte/android/urllib/EasySSLSocketF
.super Ljava/lang/Object;
.source "EasySSLSocketFactory.java"
# interfaces
.implements Lorg/apache/http/conn/scheme/SocketFactory;
.implements Lorg/apache/http/conn/scheme/LayeredSocketFac
# instance fields
.field private sslcontext:Ljavax/net/ssl/SSLContext;
# direct methods
.method public constructor <init>()V
    .locals 1
    .prologue
    .line 47
    invoke-direct {p0}, Ljava/l
```



Security certificate

Issued to:

Common name:

Organization:
Government Root Certification Authority

Organizational unit:

Serial number:
1F:9D:59:5A:D7:2F:
C2:06:44:A5:80:08:69:E3:5E:F6

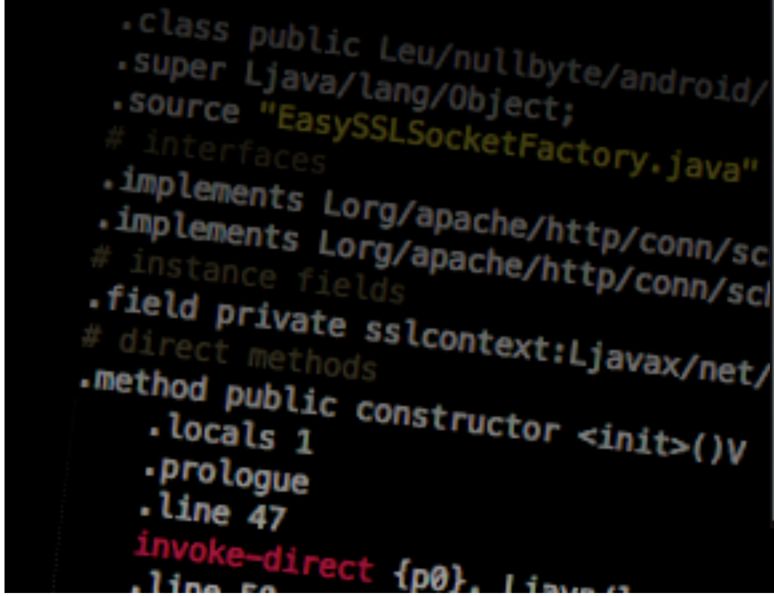
Issued by:

Common name:

Organization:
Government Root Certification Authority

Mixed Mode/No SSL

- The worst Apps even don‘t use SSL at all
- Mixed Mode:
  - Vulnerable to SSL stripping



```
.class public Leu/nullbyte/android/
.super Ljava/lang/Object;
.source "EasySSLSocketFactory.java"
# interfaces
.implements Lorg/apache/http/conn/sc
.implements Lorg/apache/http/conn/sc
# instance fields
.field private sslcontext:Ljavax/net/
# direct methods
.method public constructor <init>()V
    .locals 1
    .prologue
    .line 47
    invoke-direct {p0}, Ljava/l
```

New Tricks For Defeating SSL In Practice

Moxie Marlinspike
moxie@thoughtcrime.org

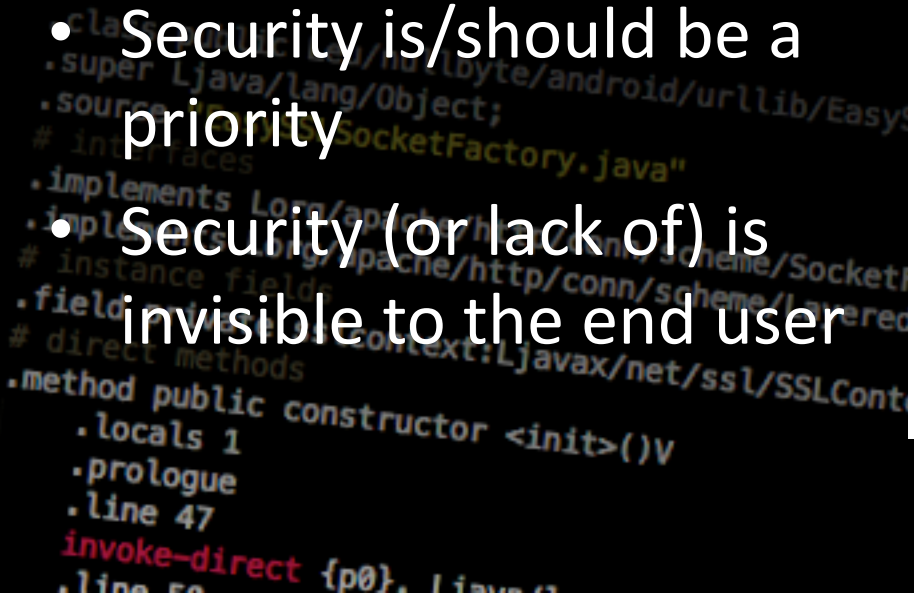
If we can do it, so can you… and Eve & Mallory

SHOW REEL

```
.class public Lde/nullbyte/android/urllib/EasySSLSocketFactory;
.super Ljava/lang/Object;
.source "EasySSLSocketFactory.java"
# interfaces
.implements Lorg/apache/http/conn/scheme/SocketFactory;
.implements Lorg/apache/http/conn/scheme/LayeredSocketFactory;
# instance fields
.field private sslcontext:Ljavax/net/ssl/SSLContext;
# direct methods
.method public constructor <init>()V
    .locals 1
    .prologue
    .line 47
    invoke-direct {p0}, Ljava/l
    .line 5
```

Banking Apps

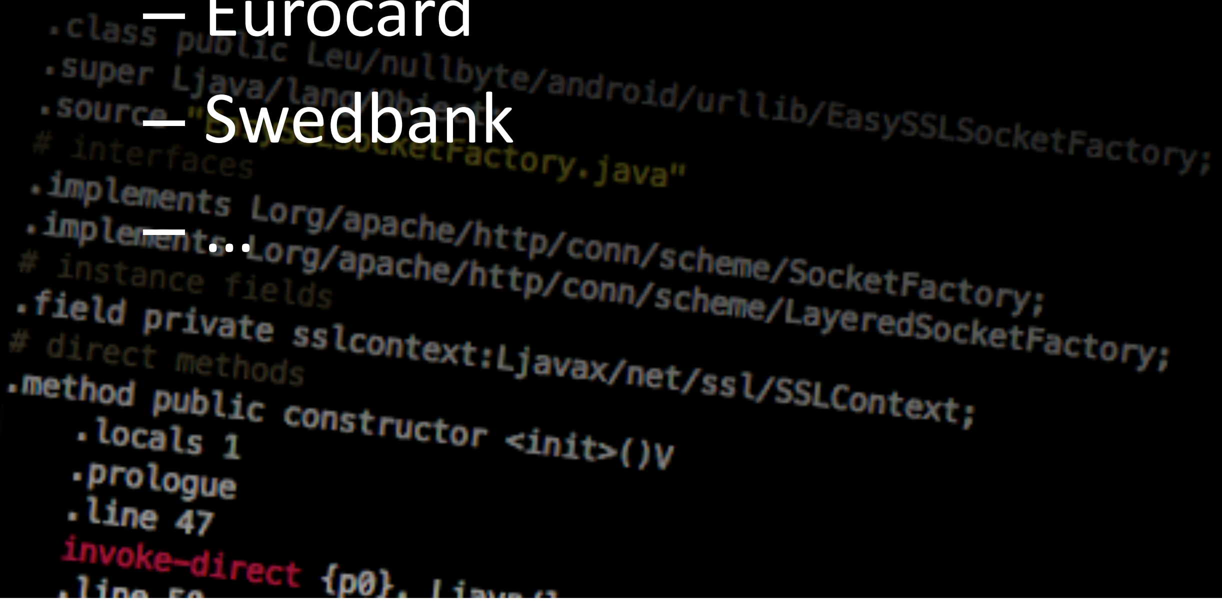
- Many banking apps exist to access online banking services
- Access to highly sensitive data
- Security is/should be a priority
- Security (or lack of) is invisible to the end user



```
.cla
.super Ljava/lang/Object;     byte/android/urllib/EasyS
.sour                SocketFactory.java"
# interfaces
.implements Lorg/apa
.impl                pache/http/conn/scheme/Socket
# instance fields                         scheme/Layered
.field               ontext:Ljavax/net/ssl/SSLCont
# direct methods
.method public constructor <init>()V
    .locals 1
    .prologue
    .line 47
    invoke-direct {p0}, Ljava/
```

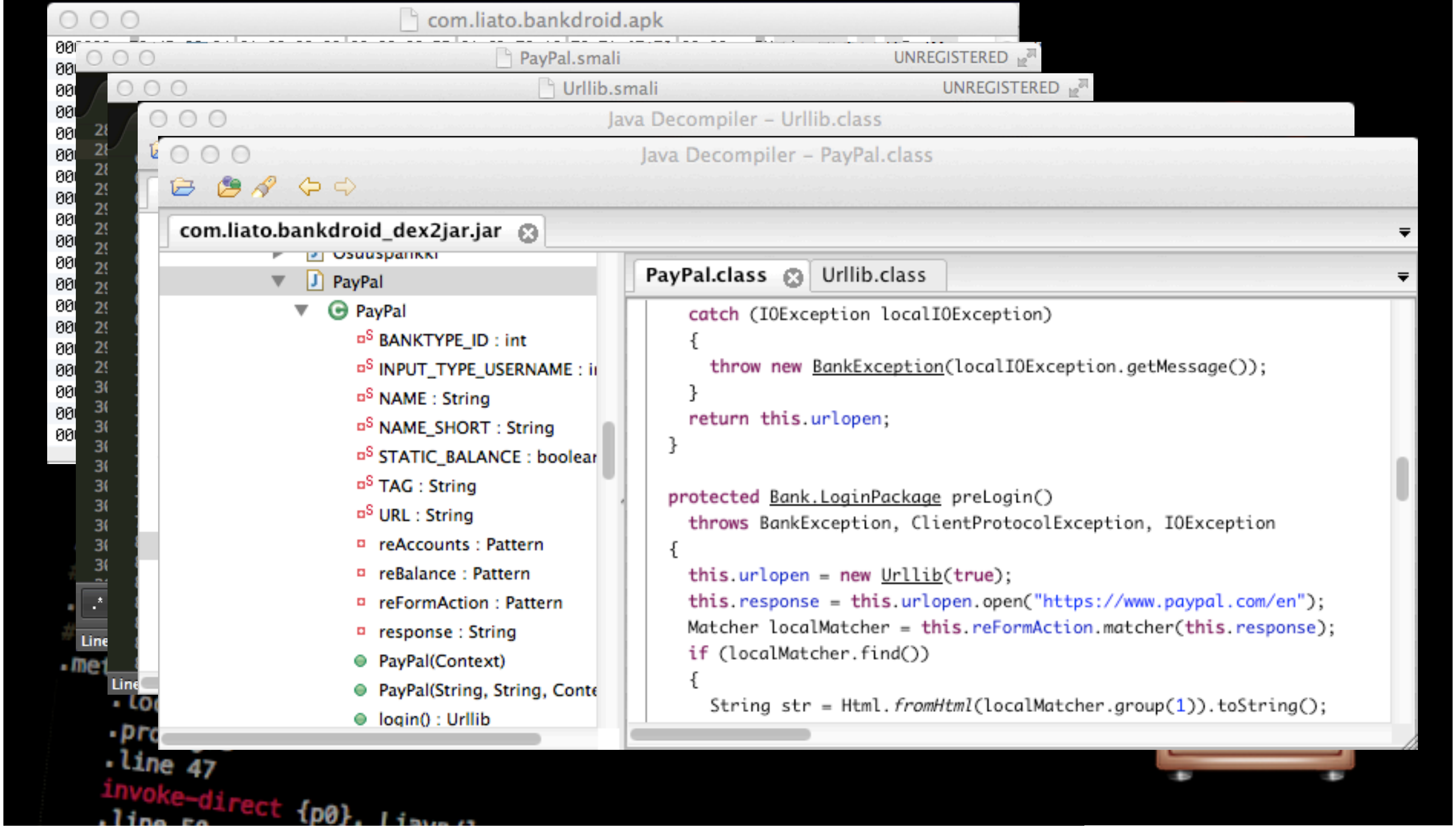
BankDroid

- Swedish banking app
- Support for ~60 banks/payment services
  - PayPal
  - Steam Wallet
  - Eurocard
  - Swedbank
  - …



```
.class public Leu/nullbyte/android/urllib/EasySSLSocketFactory;
.super Ljava/lang/Object;
.source "EasySSLSocketFactory.java"
# interfaces
.implements Lorg/apache/http/conn/scheme/SocketFactory;
.implements Lorg/apache/http/conn/scheme/LayeredSocketFactory;
# instance fields
.field private sslcontext:Ljavax/net/ssl/SSLContext;
# direct methods
.method public constructor <init>()V
    .locals 1
    .prologue
    .line 47
    invoke-direct {p0}, Ljava/l
    .line 5
```

From Binary to Source

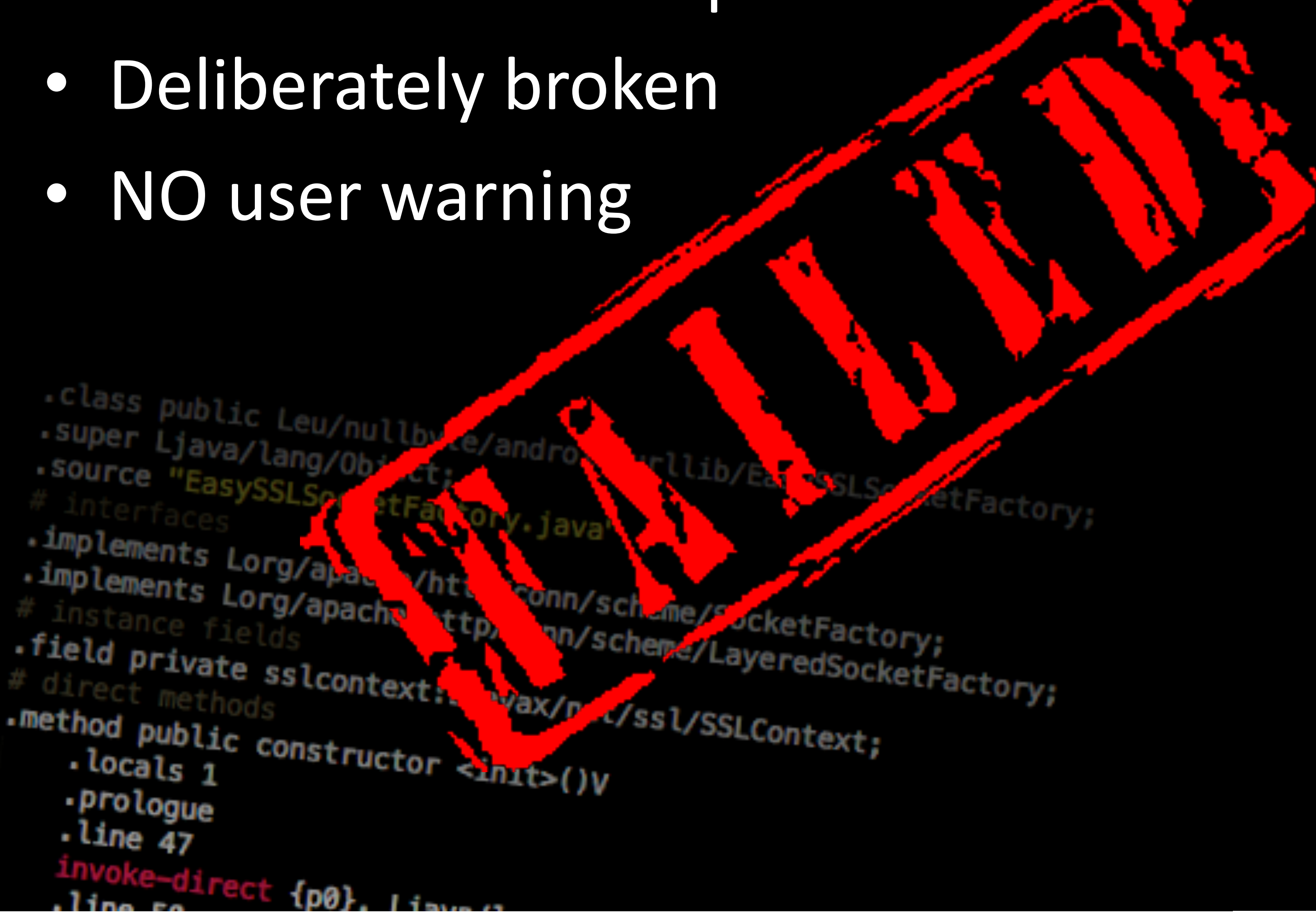


```
    catch (IOException localIOException)
    {
      throw new BankException(localIOException.getMessage());
    }
    return this.urlopen;
  }

  protected Bank.LoginPackage preLogin()
    throws BankException, ClientProtocolException, IOException
  {
    this.urlopen = new Urllib(true);
    this.response = this.urlopen.open("https://www.paypal.com/en");
    Matcher localMatcher = this.reFormAction.matcher(this.response);
    if (localMatcher.find())
    {
      String str = Html.fromHtml(localMatcher.group(1)).toString();
```

```
.met
.loc
.pro
.line 47
invoke-direct {p0}, Ljava/l
.line 50
```

BankDroid - Aftermath

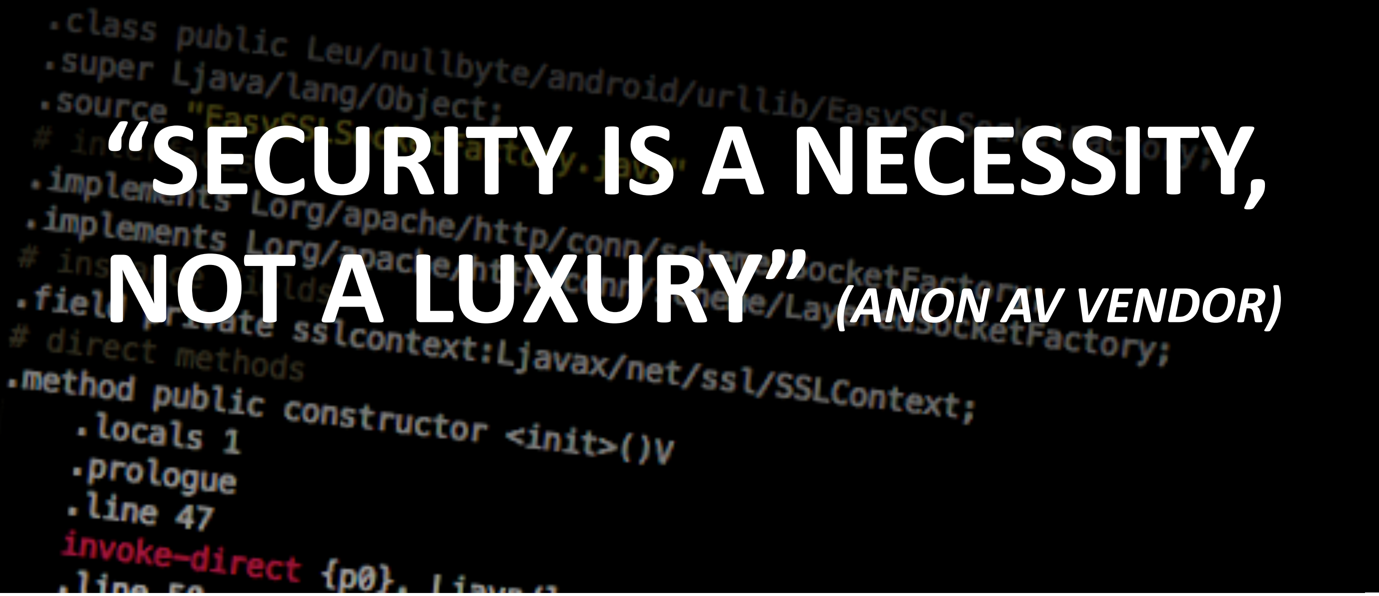
- 26 out of 41 SSL implementations broken
- Deliberately broken
- NO user warning



```
.class public Leu/nullbyte/android/urllib/EasySSLSocketFactory;
.super Ljava/lang/Object;
.source "EasySSLSocketFactory.java"
# interfaces
.implements Lorg/apache/http/conn/scheme/SocketFactory;
.implements Lorg/apache/http/conn/scheme/LayeredSocketFactory;
# instance fields
.field private sslcontext:Ljavax/net/ssl/SSLContext;
# direct methods
.method public constructor <init>()V
    .locals 1
    .prologue
    .line 47
    invoke-direct {p0}, Ljava/l
    .line 5
```

If you have a problem,
if no one else can help,
and if you can find them,
maybe you can hire...

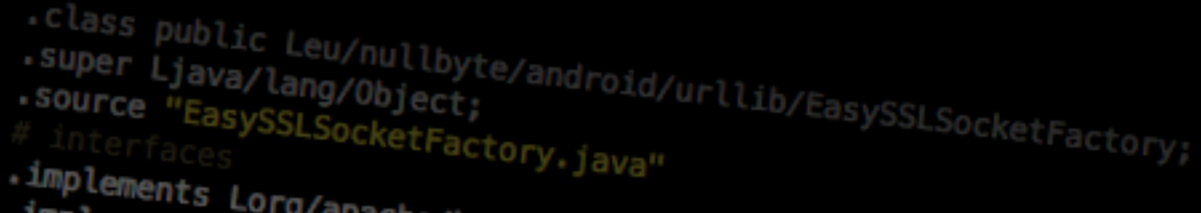
"SECURITY IS A NECESSITY, NOT A LUXURY" *(ANON AV VENDOR)*



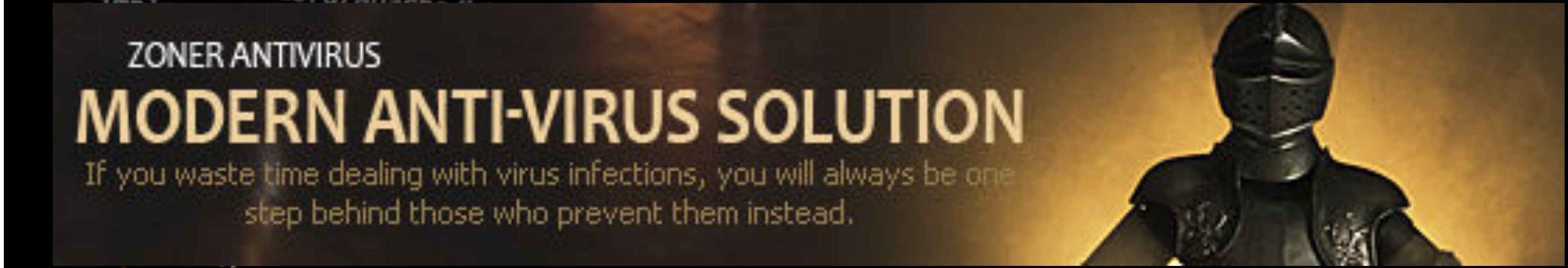
```
.class public Leu/nullbyte/android/urllib/EasySSLSocketFactory;
.super Ljava/lang/Object;
.source "EasySSLSocketFactory.java"
# interfaces
.implements Lorg/apache/http/conn/scheme/SocketFactory;
.implements Lorg/apache/http/conn/scheme/LayeredSocketFactory;
# instance fields
.field private sslcontext:Ljavax/net/ssl/SSLContext;
# direct methods
.method public constructor <init>()V
    .locals 1
    .prologue
    .line 47
    invoke-direct {p0}, Ljava/l
    .line 50
```

Best of the Best: Zoner AV

- Awarded best free Anti-Virus App
- More than just AV
- Up-to-date Signatures
- Developed in Europe

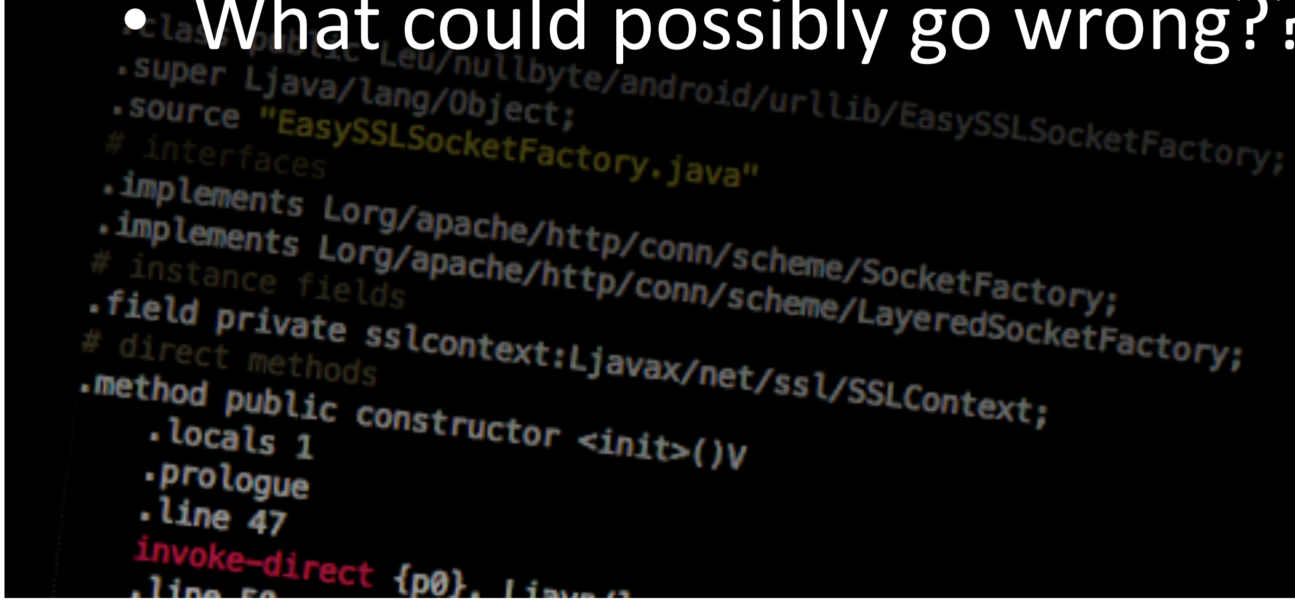


```
.class public Leu/nullbyte/android/urllib/EasySSLSocketFactory;
.super Ljava/lang/Object;
.source "EasySSLSocketFactory.java"
# interfaces
.implements Lorg/apache/
```


ZONER ANTIVIRUS
MODERN ANTI-VIRUS SOLUTION
If you waste time dealing with virus infections, you will always be one step behind those who prevent them instead.

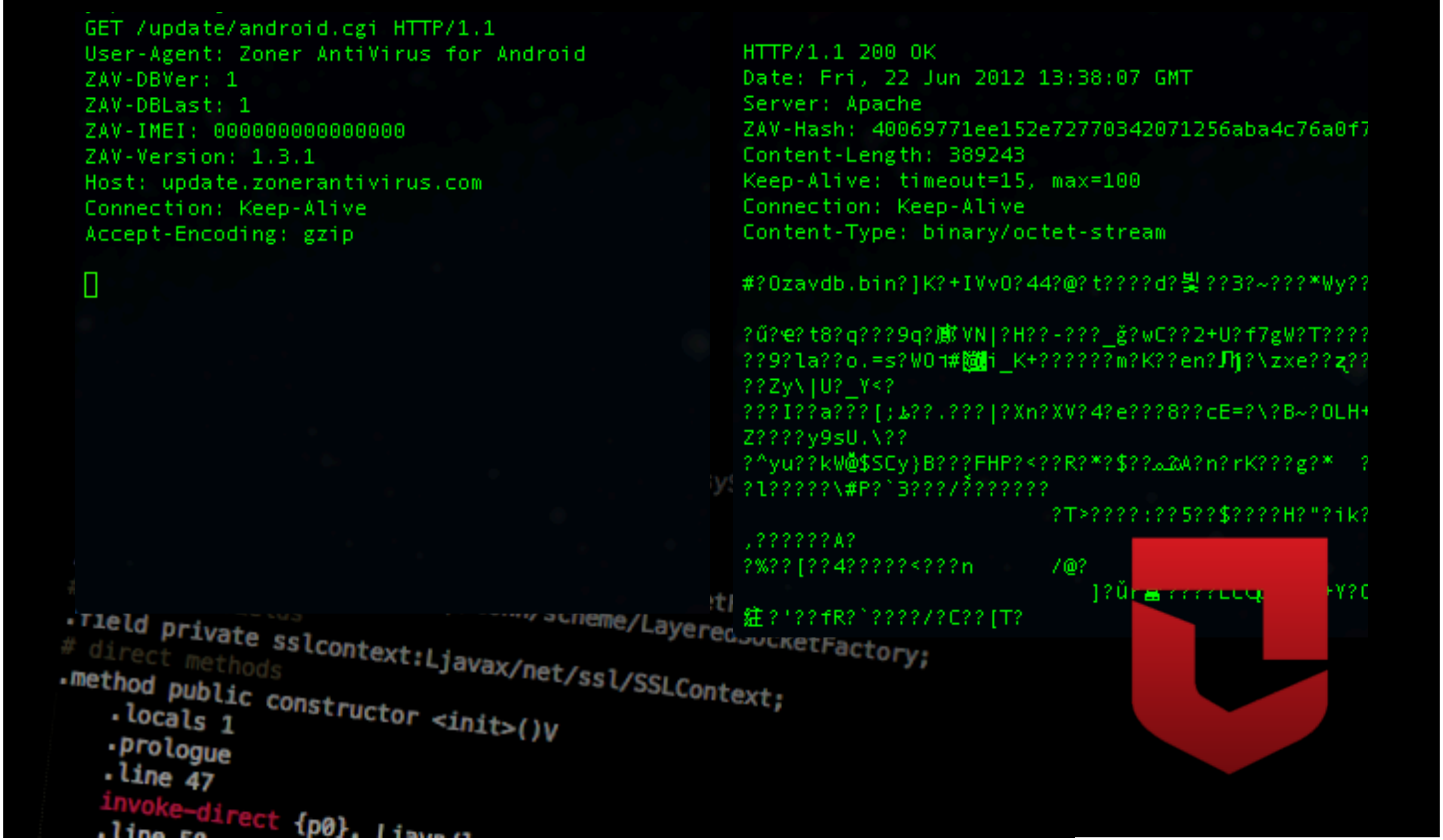
A quick peek behind the curtain…

- The good thing: It uses SSL
  - Unfortunately: The wrong way
  - Accepts all hostnames for signature update
- Virus signatures are public anyway
- What could possibly go wrong??



```
.class public Leu/nullbyte/android/urllib/EasySSLSocketFactory;
.super Ljava/lang/Object;
.source "EasySSLSocketFactory.java"
# interfaces
.implements Lorg/apache/http/conn/scheme/SocketFactory;
.implements Lorg/apache/http/conn/scheme/LayeredSocketFactory;
# instance fields
.field private sslcontext:Ljavax/net/ssl/SSLContext;
# direct methods
.method public constructor <init>()V
    .locals 1
    .prologue
    .line 47
    invoke-direct {p0}, Ljava/
    .line 50
```

Signature Update in Depth



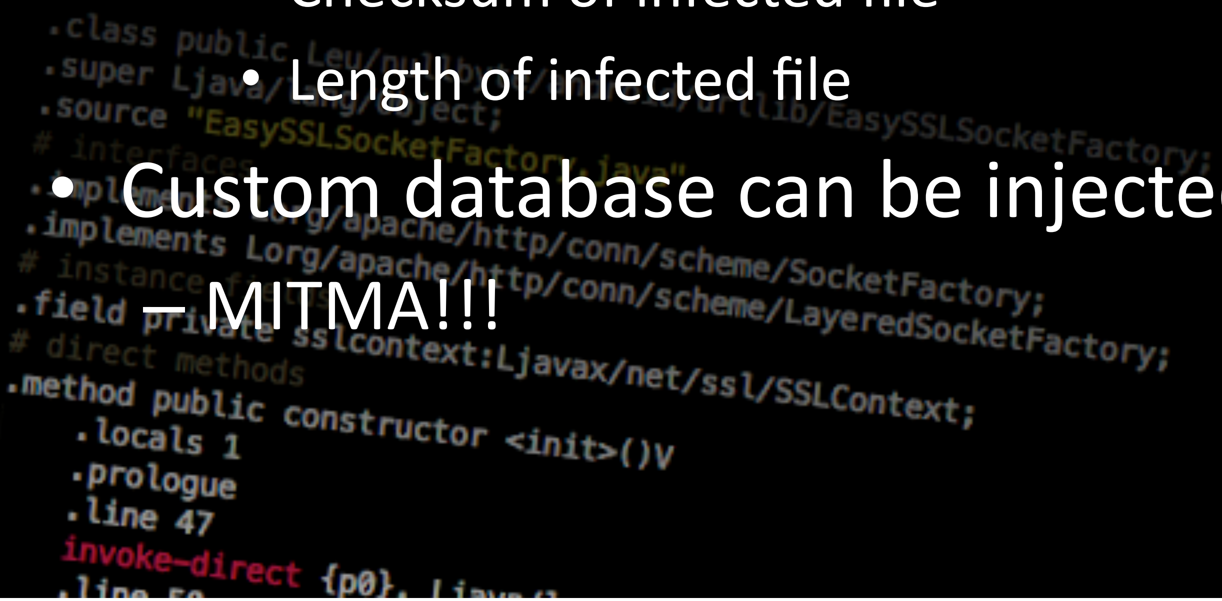
```
GET /update/android.cgi HTTP/1.1
User-Agent: Zoner AntiVirus for Android
ZAV-DBVer: 1
ZAV-DBLast: 1
ZAV-IMEI: 000000000000000
ZAV-Version: 1.3.1
Host: update.zonerantivirus.com
Connection: Keep-Alive
Accept-Encoding: gzip
```

```
HTTP/1.1 200 OK
Date: Fri, 22 Jun 2012 13:38:07 GMT
Server: Apache
ZAV-Hash: 40069771ee152e72770342071256aba4c76a0f7
Content-Length: 389243
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: binary/octet-stream

#?0zavdb.bin?]K?+IVvO?44?@?t????d?뵞???3?~???*Wy??

?ű?ɐ?t8?q????9q?瀌VN|?H??-???_ğ?wC??2+U?f7gW?T????
??9?la??o.=s?W0┤#▒i_K+???????m?K??en?Лj?\zxe??ʐ??
??Zy\|U?_Y<?
???I??a???[;ﻼ??.???|?Xn?XV?4?e???8??cE=?\?B~?0LH+
Z????y9sU.\??
?^yu??kW@$SCy}B???FHP?<??R?*?$??ﻣﭏA?n?rK???g?*  ?
?l??????\#P?`3????/?????????
                                ?T>?????:??5??$????H?"?ik?

,???????A?
?%??[??4??????<????n       /@?
                                     ]?ŭr▄????LLQ      +Y?C
絓?'???fR?`????/?C??[T?
```

```
.field private sslcontext:Ljavax/net/ssl/SSLContext;
# direct methods
.method public constructor <init>()V
    .locals 1
    .prologue
    .line 47
    invoke-direct {p0}, Ljava/
```

The Problem

- SHA-1 Checksum != Crypto Signature
- Database can be reverse engineered
  – Simple hashmap
    - Description
    - Checksum of infected file
    - Length of infected file
- Custom database can be injected
  – MITMA!!!



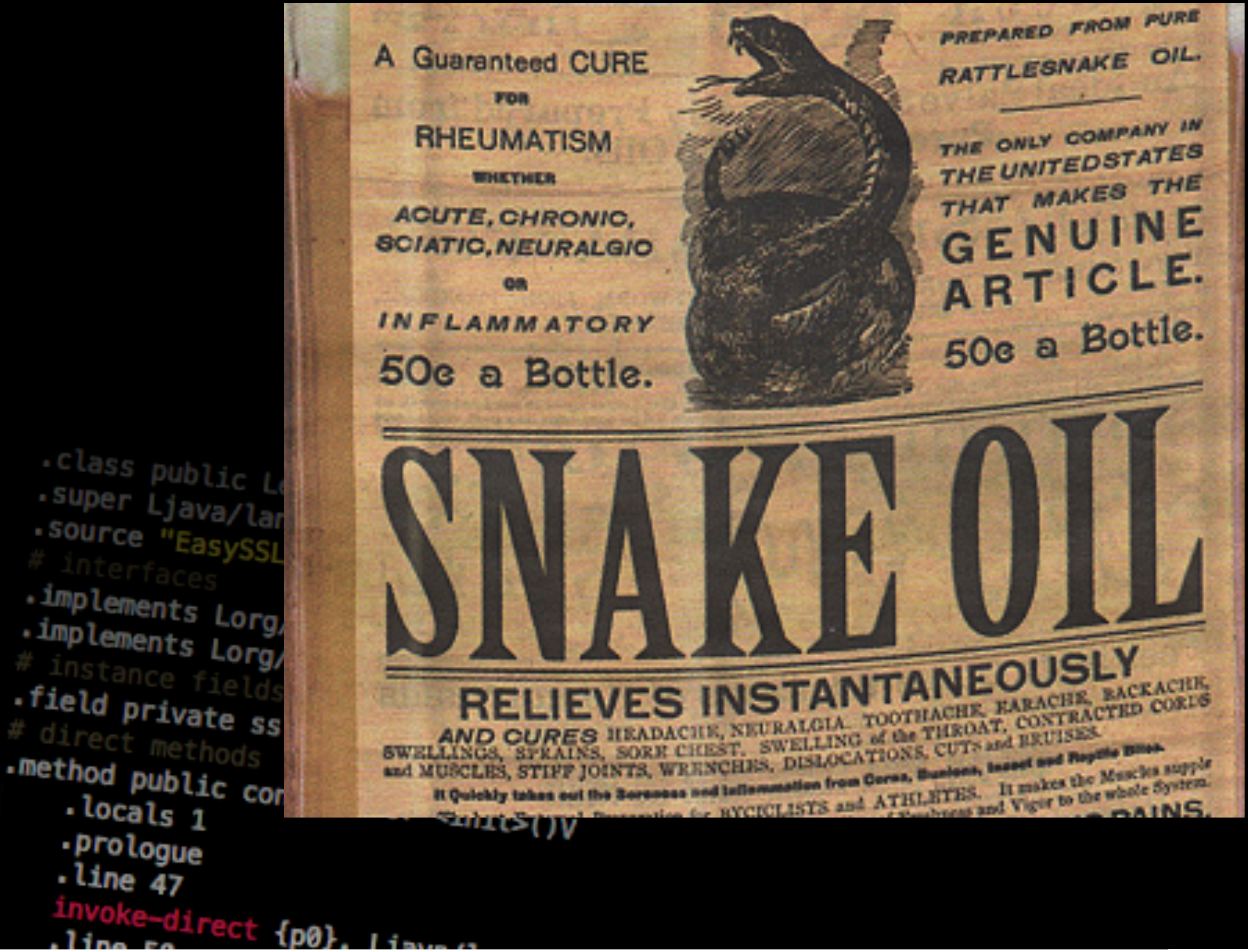
```
.class public L
.super Ljava/lang/Object;
.source "EasySSLSocketFactory.java"
# interfaces
.implements Lorg/apache/http/conn/scheme/SocketFactory;
.implements Lorg/apache/http/conn/scheme/LayeredSocketFactory;
# instance fields
.field private sslcontext:Ljavax/net/ssl/SSLContext;
# direct methods
.method public constructor <init>()V
    .locals 1
    .prologue
    .line 47
    invoke-direct {p0}, Ljava/
```

FAILED

Proof of Concept

```
.clas
.supe
.sour
# inte
.imple
.imple
# inst
.field
# direc
.method
    .lo
    .pr
    .line 47
    invoke-direct {p0}, Ljava/l
    .line 50
```

False sense of security



A Guaranteed CURE
FOR
RHEUMATISM
WHETHER
ACUTE, CHRONIC,
SCIATIC, NEURALGIC
OR
INFLAMMATORY
50c a Bottle.

PREPARED FROM PURE
RATTLESNAKE OIL.

THE ONLY COMPANY IN
THE UNITED STATES
THAT MAKES THE
GENUINE
ARTICLE.
50c a Bottle.

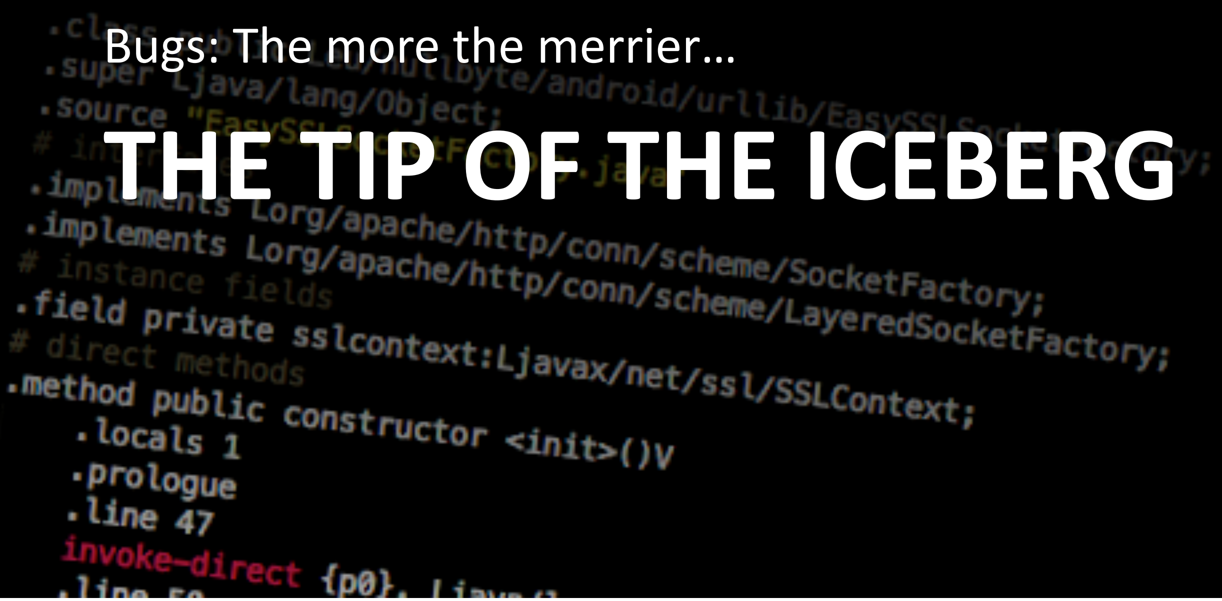
SNAKE OIL

RELIEVES INSTANTANEOUSLY
AND CURES HEADACHE, NEURALGIA. TOOTHACHE, EARACHE, BACKACHE,
SWELLINGS, SPRAINS, SORE CHEST. SWELLING of the THROAT, CONTRACTED CORDS
and MUSCLES, STIFF JOINTS, WRENCHES, DISLOCATIONS, CUTS and BRUISES.
It Quickly takes out the Soreness and Inflammation from Corns, Bunions, Insect and Reptile Bites.
It makes the Muscles supple
Preparation for BYCICLISTS and ATHLETES. and Vigor to the whole System.
PAINS.

```
.class public L
.super Ljava/lan
.source "EasySSL
# interfaces
.implements Lorg/
.implements Lorg/
# instance fields
.field private ss
# direct methods
.method public con
    .locals 1
    .prologue
    .line 47
    invoke-direct {p0}, Ljava/l
    .line 5
```

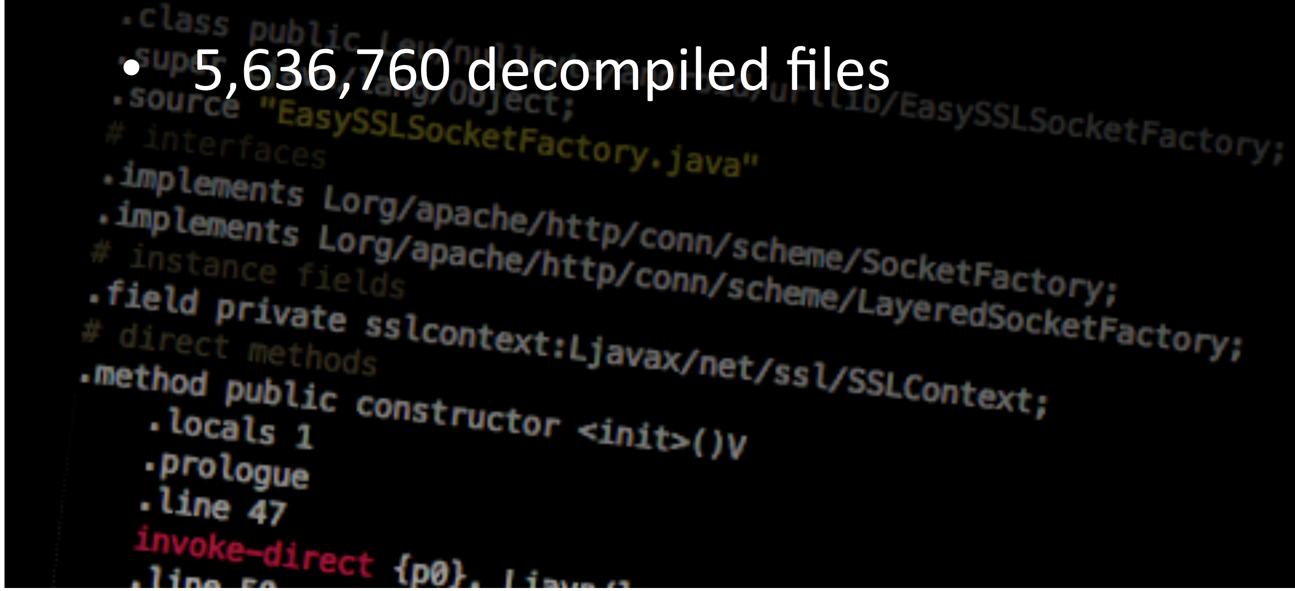
Bugs: The more the merrier…

THE TIP OF THE ICEBERG



```
.class [illegible]/nutlbyte/android/urllib/EasySSLSo[illegible]
.super Ljava/lang/Object;
.source "EasySSLSocketFactory.java"
# interfaces
.implements Lorg/apache/http/conn/scheme/SocketFactory;
.implements Lorg/apache/http/conn/scheme/LayeredSocketFactory;
# instance fields
.field private sslcontext:Ljavax/net/ssl/SSLContext;
# direct methods
.method public constructor <init>()V
    .locals 1
    .prologue
    .line 47
    invoke-direct {p0}, Ljava/l
    .line 50
```

- Of the 13,667 most popular apps
  - 12,135 apps use the network
- Android 4.0 only
- 169 GB total
- 5,636,760 decompiled files

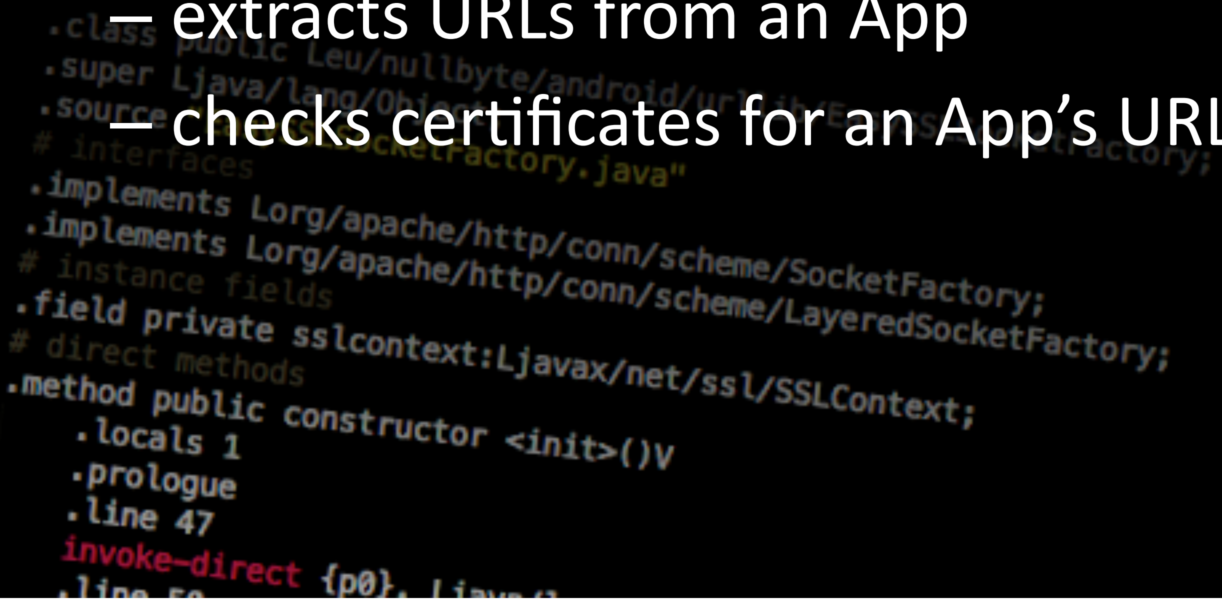


```
.class public Leu/nullbu[illegible]/urllib/EasySSLSocketFactory;
.super Ljava/lang/Object;
.source "EasySSLSocketFactory.java"
# interfaces
.implements Lorg/apache/http/conn/scheme/SocketFactory;
.implements Lorg/apache/http/conn/scheme/LayeredSocketFactory;
# instance fields
.field private sslcontext:Ljavax/net/ssl/SSLContext;
# direct methods
.method public constructor <init>()V
    .locals 1
    .prologue
    .line 47
    invoke-direct {p0}, Ljava/l
    .line 50
```

MalloDroid: Static Code Analysis

Androguard extension which:

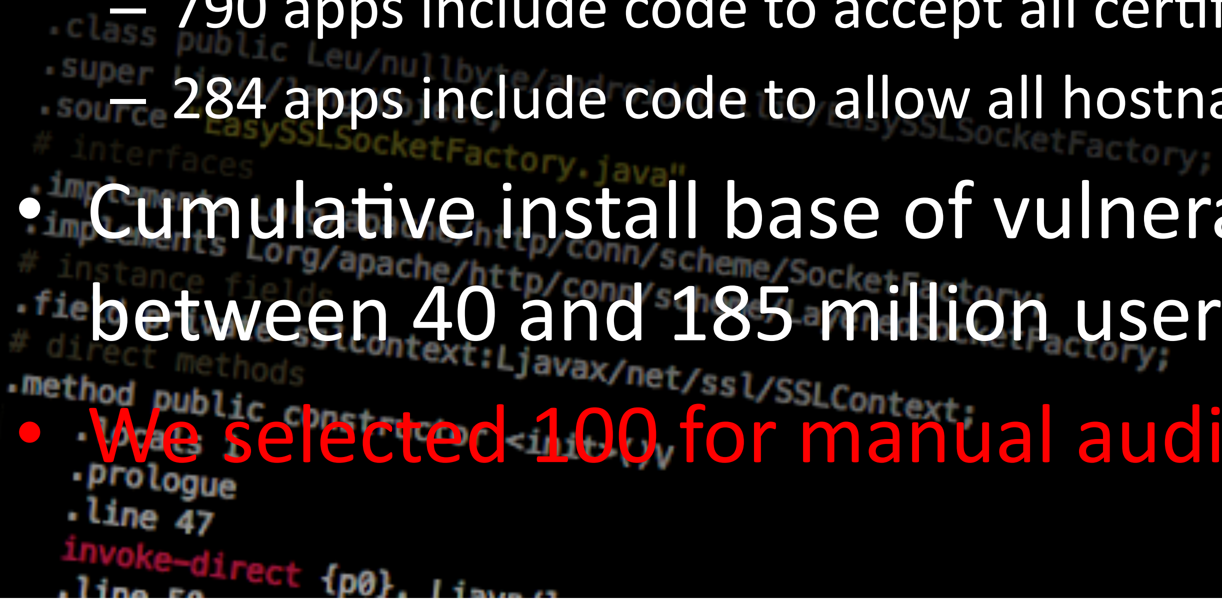
- finds broken TrustManagers like: EasySSLTrustManager, FakeTrustManager, NullTrustManager, …
  (48 different names for the same problem)
- finds Apps that use allow all hostname verifiers
- extracts URLs from an App
- checks certificates for an App's URLs



```
.class public Leu/nullbyte/android/ur
.super Ljava/lang/Object;
.source "              cketFactory.java"
# interfaces
.implements Lorg/apache/http/conn/scheme/SocketFactory;
.implements Lorg/apache/http/conn/scheme/LayeredSocketFactory;
# instance fields
.field private sslcontext:Ljavax/net/ssl/SSLContext;
# direct methods
.method public constructor <init>()V
    .locals 1
    .prologue
    .line 47
    invoke-direct {p0}, Ljava/l
```

SSL on Android

- Of the 12,135 apps
- 6,214 apps mix HTTPS and HTTP
- 5,810 apps use HTTP only
- 111 apps use HTTPS only
- 1,074 apps vulnerable to SSL MITMA!!
  - 790 apps include code to accept all certificates
  - 284 apps include code to allow all hostnames
- Cumulative install base of vulnerable apps lies between 40 and 185 million users
- We selected 100 for manual audit…



SSL on Android

- From 41 apps, we were able to capture credentials for
  - American Express, Diners Club, Paypal, bank accounts, Facebook, Twitter, Google, Yahoo, Microsoft Live ID, Box, WordPress, remote control servers, arbitrary email accounts, and IBM Sametime, among others.
- It was also possible to remotely inject and execute code in an app created by a vulnerable app-building frame- work.

```
.class public L
.super Ljava/lang/Object;                  d/urllib/EasySSLSocketFact
.source "EasySSLSocketFa
# i
.implements Lorg/apache/http/conn/scheme/SocketFactory;
.implements                http/conn/scheme/LayeredSocketFactory;
# instance fields
.field private sslcontext:Ljavax/net/ssl/SSLContext;
# direct methods
.method public constructor <init>()V
    .locals 1
    .prologue
    .line 47
    invoke-direct {p0}, Ljava/l
```

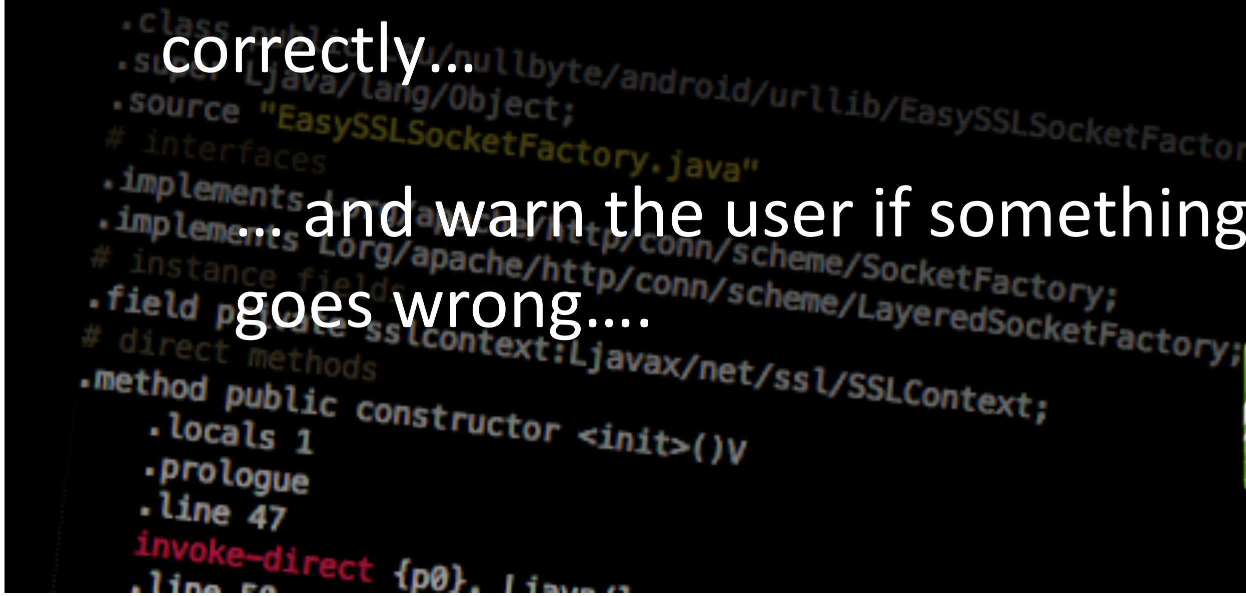
We’re down but not out…

- We know there are Apps that do it the wrong way

- Fortunately they are here to protect us:

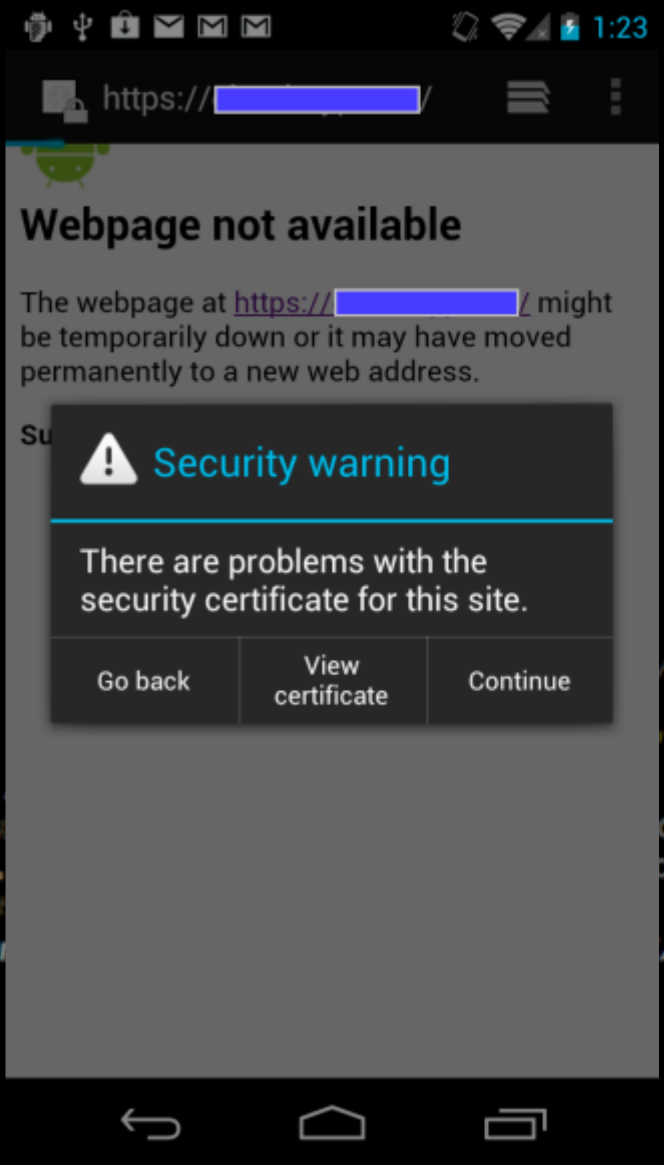
- All do SSL certificate validation correctly…

  … and warn the user if something goes wrong….



```
.class publi
.super Ljava/lang/Object;          ullbyte/android/urllib/EasySSLSocketFactory;
.source "EasySSLSocketFactory.java"
# interfaces
.implements
.implements Lorg/apache/http/conn/scheme/SocketFactory;
# instance field
.field private sslcontext:Ljavax/net/ssl/SSLContext;     apache/http/conn/scheme/LayeredSocketFactory;
# direct methods
.method public constructor <init>()V
    .locals 1
    .prologue
    .line 47
    invoke-direct {p0}, Ljava/l
    .line 5
```

The Last Line of Defense



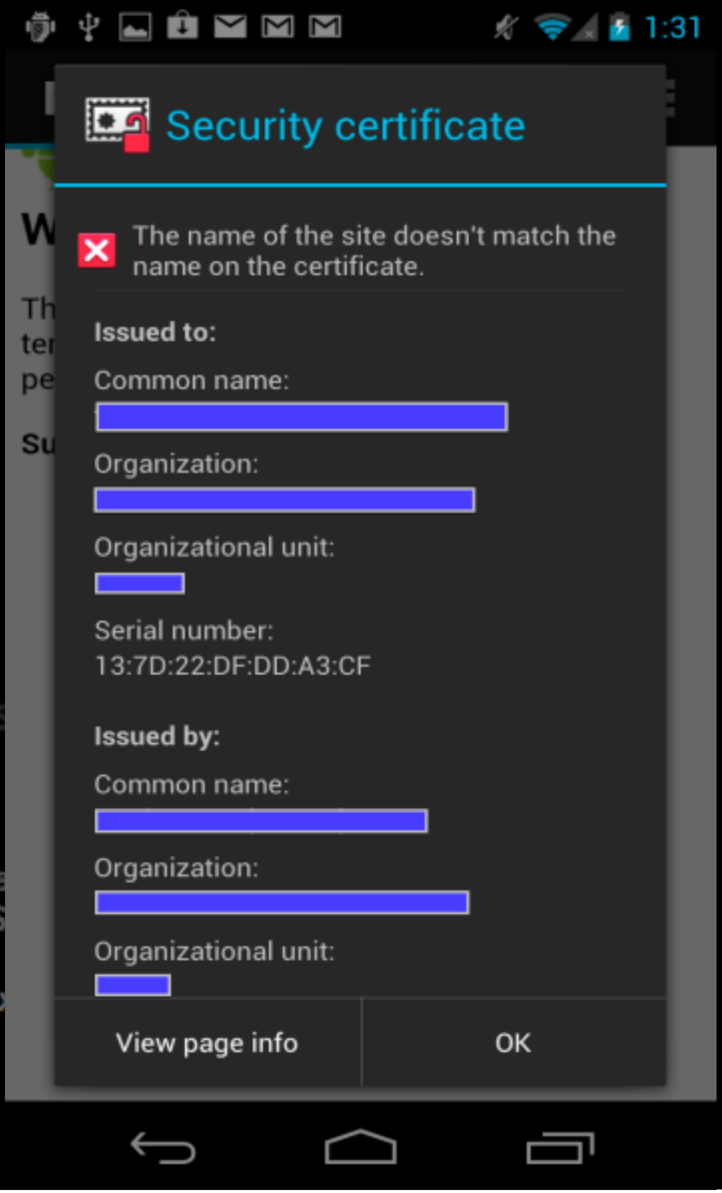
https://[redacted]/

Webpage not available

The webpage at https://[redacted]/ might be temporarily down or it may have moved permanently to a new web address.

Security warning

There are problems with the security certificate for this site.



Security certificate

☒ The name of the site doesn't match the name on the certificate.

Issued to:

Common name:
[redacted]

Organization:
[redacted]

Organizational unit:
[redacted]

Serial number:
13:7D:22:DF:DD:A3:CF

Issued by:

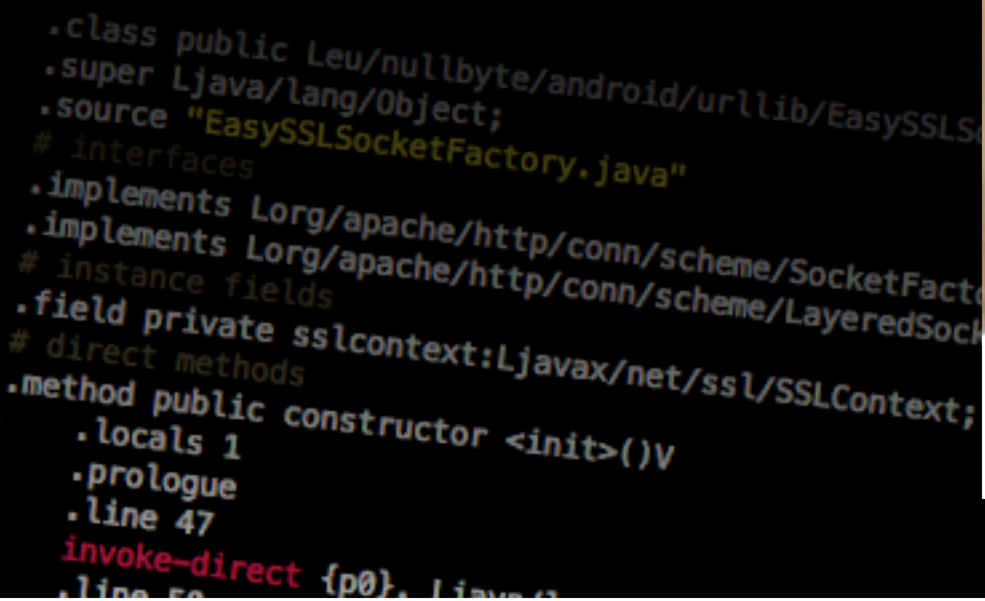
Common name:
[redacted]

Organization:
[redacted]

Organizational unit:
[redacted]

Stop! There’s Trouble in Paradise

- We conducted an online survey
  - To find out if the warning messages help the users
  - To see if users know when they are surfing on an SSL protected website

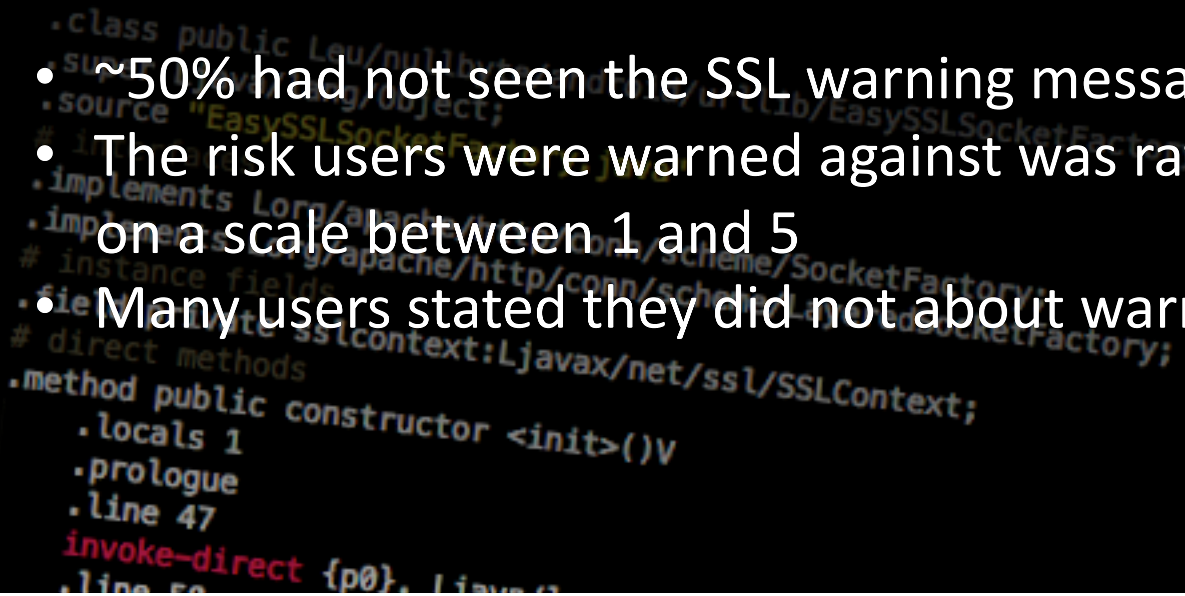


```
.class public Leu/nullbyte/android/urllib/EasySSLS
.super Ljava/lang/Object;
.source "EasySSLSocketFactory.java"
# interfaces
.implements Lorg/apache/http/conn/scheme/SocketFact
.implements Lorg/apache/http/conn/scheme/LayeredSoc
# instance fields
.field private sslcontext:Ljavax/net/ssl/SSLContext;
# direct methods
.method public constructor <init>()V
    .locals 1
    .prologue
    .line 47
    invoke-direct {p0}, Ljava/l
    .line 5
```

- 745 participants
  - avg. age 24 years
  - 88% university students

- 47.5% of non-IT experts believed they were using a secure Internet connection...although it was plain HTTP.

- ~50% had not seen the SSL warning message before.
- The risk users were warned against was rated with 2.86 (sd=.94) on a scale between 1 and 5
- Many users stated they did not about warning messages at all.



```
.class public Leu/null...
.super ...object;
.source "EasySSLSocketFactory.java"
.implements Lorg/apache/...
.implements Lorg/apache/http/conn/scheme/SocketFactory;
# instance fields
.field private sslcontext:Ljavax/net/ssl/SSLContext;
# direct methods
.method public constructor <init>()V
    .locals 1
    .prologue
    .line 47
    invoke-direct {p0}, Ljava/l...
```

```
.cl             ttbyte/android/urllib/EasySSLS
.super Ljava/lang/Object;
.source "EasySSL
# in
.implements Lorg/apache/http/conn/scheme/SocketFactory;
.implements Lorg/apache/http/conn/scheme/LayeredSocketFactory;
# instance fields
.field private sslcontext:Ljavax/net/ssl/SSLContext;
# direct methods
.method public constructor <init>()V
    .locals 1
    .prologue
    .line 47
    invoke-direct {p0}, Ljava/l
    .line 5
```

Step by step into the future

TAMING THE GHOSTS WE CALLED

Possible Solutions

- Enforce the use of the standard SSL API
- Improved usability of API/PKI/CaaS
- Android version of EFF's *HTTPS Everywhere*
- Visual Security Feedback
- Add *MalloDroid* to app installers/app market

```
.clas                byte/android/urllib/Easy
.super Ljava/lang/Object;
.source "EasySSLSocketFactory.java"
# interfa
.implements Lorg/apache/http/conn/scheme/Socket
.implements Lorg/apache/http/conn/scheme/Layere
# instance fields
.field private sslcontext:Ljavax/net/ssl/SSLCont
# direct methods
.method public constructor <init>()V
    .locals 1
    .prologue
    .line 47
    invoke-direct {p0}, Ljava/l
    .line 50
```

DCSec Conclusion

Design systems with the user in mind

- Conduct preliminary user studies *before* designing the system
- Test systems during development and before role-out

Cloud computing is particularly challenging

- many (non-tech) actors
- offer Security as a Service
- anything more complicated than user name / password creates problems unless it is made *very* usable

Conclusion

Usable Security is also important for developers

- Create API which are easy to use and difficult to abuse
- Only burden App/Cloud developers with absolutely necessary security code
- Educate developers about security technology

The merging of paradigms creates issues for traditional (and secure) services which did not exist before.

No droids were harmed during this research!