




APPSEC
EUROPE

Internet banking safeguards vulnerabilities

Wojtek Dworakowski, @wojtwo
SecuRing

#login

Wojtek Dworakowski


SecuRing (since 2003) securing

OWASP Poland Chapter Leader

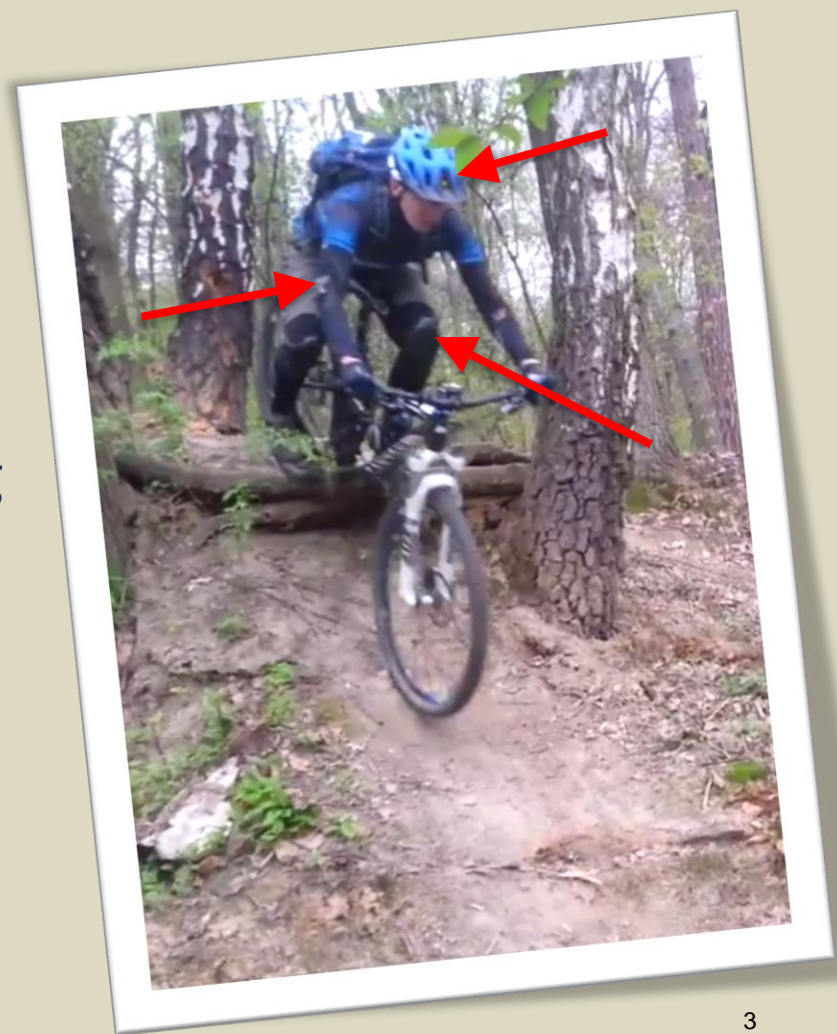


#login

Wojtek Dworakowski

SecuRing (since 2003)  securing

OWASP Poland Chapter Leader



Agenda

- Intro
 - Banks vs attackers – current state, common security features
- Vulnerabilities and best practices
 - transaction authorization vulnerabilities, trusted recipients feature abuses, transaction limit bypass, user auth mistakes...
- PSD2 – future changes to internet banking security
- Should OWASP publish guidelines for specific application domains (e.g. internet banking)?





APPSEC
EUROPE

BANKS VS ORGANIZED (?) CRIME

Common attack patterns

- Malware
 - Web inject
 - Keylogger + remote desktop
 - Clipboard manipulation
- Vulnerability exploitation
 - Infrastructure
 - Application
 - Libraries / frameworks !

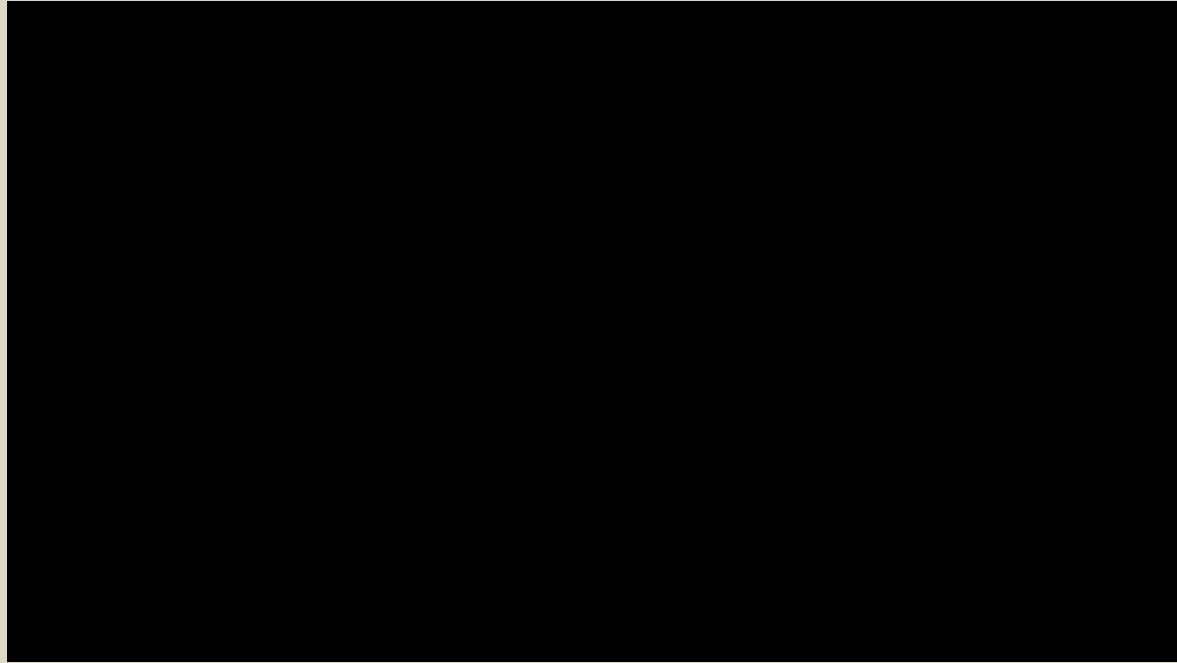


Clipboard (or memory) manipulation



Source: CERT Poland
http://www.cert.pl/news/8999/langswitch_lang/en

Clipboard (or memory) manipulation



Source: CERT Poland
http://www.cert.pl/news/8999/langswitch_lang/en

Server vulnerability exploitation

- In late 2015 one of Polish banks was pwned **(outdated components)**
- Intruder was able to modify transactions

Potwierdzenie transakcji

Typ transakcji	Przelew przychodzący
Data wykonania	2015-02-16
Data księgowania	2015-02-16
Rachunek	12 [redacted]
Posiadacz rachunku	[redacted]
Adres	[redacted] DĄBROWA
Rachunek nadawcy	82 [redacted]
Nadawca	[redacted] DEALE
Adres nadawcy	[redacted] WARSZAWA
Tytułem	wpłata własna
Kwota	179 600,00 PLN

Potwierdzenie przelewu złodzieja *Obraz 3 z 4*

~ 40 000 eur



Server vulnerability exploitation

- In late 2015 one of Polish banks was pwned (outdated components)
- Intruder was able to modify transactions

Historia transakcji

za okres 2015-03-01 do 2015-03-02

Saldo początkowe:

672 558,53

Saldo końcowe:

262 393,53

Data wykonania	Data księgowania	Opis transakcji		Saldo
2015-03-02	2015-03-02	Justyna FV 79/2015		262 393,53
2015-03-02	2015-03-02	 Korekta fv 49/2015	-29 000,00	312 408,53
2015-03-02	2015-03-02	Anna Korekta faktury 45/2015	-31 000,00	341 423,53
2015-03-02	2015-03-02	Urszula FV 77/2015	-38 000,00	372 438,53
2015-03-02	2015-03-02	Urszula FV 78/2015	-32 000,00	410 453,53
2015-03-02	2015-03-02	Bartłomiej Faktura 76/2015 cz 2	-28 000,00	442 468,53

~ 92 000 eur



Server vulnerability



Pwnie for Most Epic FAIL

Sometimes giving 110% just makes your FAIL that much more epic. And what use would the Internet be if it wasn't there to document this FAIL for all time? This award is to honor a person or company's spectacularly epic FAIL.

- Oh, Please... Man!
Credit: U.S. Office of Personnel Management

Remember when you applied for that security clearance and you told a federal employee all the vile things you've ever done? Good news, now suspended. Well, that might not be good news. Regardless, the OPM is up and everyone else down. So much so, that the USA government might actually be pulling covert agents out of foreign countries. USA #1 (in awful federal data breaches).

- We're Not Quite Sure
Credit: Plus Bank

All this shit is in Polish so we can't begin to understand the story or be troubled with using Google translate, but apparently a bank in Poland got popped and then pulled a 40 year old mid-life crisis move and denied everything regardless of the evidence against them. We almost have to tip our hats to anyone that can live a lie of that magnitude. Kudos Plus Bank!

- Peepin' on the Creepin'
Credit: AshleyMadison.com

As a group of people who have been cheating on their operating system for years (Dino really loves



English writeup:

<https://www.linkedin.com/pulse/online-banking-owned-single-attacker-wojciech-dworakowski>



Wojciech Dworakowski

IT Security Expert, Owner at SecuRing, OWASP Poland Chapter Leader

Obserwuj

Online banking owned by single attacker

12 cze 2015 | 869 wyświetleń | 35 poleceń | 13 komentarzy

This week Polish internet is buzzing about break-in to online banking website. This case seems to be extremely interesting and quite different from

How banks mitigate these risks?

- Multi-factor authentication
- Transaction authorization
 - Trusted recipients
- Authorization schemes
- Transaction limits
- Notifications (SMS, e-mail, ...)
- Channel activation
 - Mobile device authorization



img src: <http://nuvali.ph/see-and-do/sports/mountain-bike-trails/>







APPSEC
EUROPE



Image: <https://www.britishcycling.org.uk>

Details matter

SECURITY FEATURES VULNERABILITIES BEST PRACTICES

Transaction authorization

E-banking transaction authorization

Common Vulnerabilities, Security Verification and Best Practices for Implementation

*Wojtek Dworakowski, @wojdwo
SecuRing*



OWASP AppSecEU 15
Amsterdam, The Netherlands



Domestic transfer

Account debited

Account balance 1 316 320,15 PLN

Available balance 1 316 320,15 PLN

Recipient

Choose the payment template > Search

Company name/Name and surname

Address

Recipient account number

Bank name Alior Bank SA

Transfer details

Amount PLN

Title

Payment date

Save new template Save as trusted

Please send me the confirmation

e-mail

Cancel

Confirm later

Confirm the transfer

Source: aliorbank.pl demo



APPSEC
EUROPE

Account debited

09 2490 0005 0000 4000 4183 7513

Recipient

Company name/Name and surname Jan Kowalski



Address

Piękna 12

Recipient account number

22 2222 2222 2222 2222 2222

Bank name

Alior Bank SA

Domestic transfer: Recipient
account 22XXXX222 amount
77.34 EUR authorization
code: 36032651

Transfer details

Amount

77,34PLN

Title

Opłata za mieszkanie

Payment date

22-01-2008

Transfer type

common

SMS code number: 2



Vuln examples (functional)



Domestic transfer from account
99XXXX890 amount 1.00 EUR
authorization code: 78537845



Vuln examples (non functional)

Step 1: User enters transaction data

POST /domesticTransfer HTTP/1.1

task=APPROVE_TRN

trnData.acc_id=910458

trnData.bnf_name=TELECOM+OPERATOR+Ltd

trnData.bnf_acc_no=PL99111100000000001234567890

trnData.amount=1.00

trnData.currency=EUR

trnData.title=invoice+123456



Vuln examples (non functional)

Step 2: User enters authorization code

POST /domesticTransfer HTTP/1.1

task=SEND_RESPONSE

trnData.response=87567340



Vuln examples (non functional)

Overwrite transaction data in step 2

```
POST /domesticTransfer HTTP/1.1
```

```
task=SEND_RESPONSE
```

```
trnData.response=8756734
```

```
trnData.bnf_acc_no=PL66222200000000006666666666
```

```
trnData.amount=1000.00
```

```
trnData.currency=EUR
```

Confirmation



Transfer executed



Transaction authorization – best practices

Page [Discussion](#) [Read](#) [View source](#) [View history](#)

Transaction Authorization Cheat Sheet



Last revision (mm/dd/yy): **12/23/2015**

[hide]

- 1 Purpose and audience
- 2 Introduction
- 3 1.0 Functional Guidelines
 - 3.1 1.1 Transaction authorization method has to allow a user to identify and acknowledge significant transaction data
 - 3.2 1.2 Change of authorization token should be authorized using the current authorization token
 - 3.3 1.3 Change of authorization method should be authorized using the current authorization method
 - 3.4 1.4 Users should be able to easily distinguish the authentication process from the transaction authorization process
 - 3.5 1.5 Each transaction should be authorized using unique authorization credentials
- 4 2. Non-functional guidelines
 - 4.1 2.1 Authorization should be performed and enforced server-side
 - 4.2 2.2 Authorization method should be enforced server side
 - 4.3 2.3 Transaction verification data should be generated server-side
 - 4.4 2.4 Application should prevent authorization credentials brute-forcing

Thx !

- Steven Wierckx
- Adam Zachara
- Adam Lange
- Sławomir Jasek
- Andrzej Kleśnicki
- Sven Thomassin
- James Holland
- Francois-Eric Guyomarch
- Milan Khan





Trusted recipients

All predefined account list (3)

Name

Czyn

Gaz



Account number ▾

Authorization required ▾

20 1240 1112 1111 0000 0168 6771

Yes

- > Execute
- > Modify
- > Delete

77 1240 1112 1787 0000 0168
6768

No

- > Execute
- > Modify
- > Delete

Trusted recipients

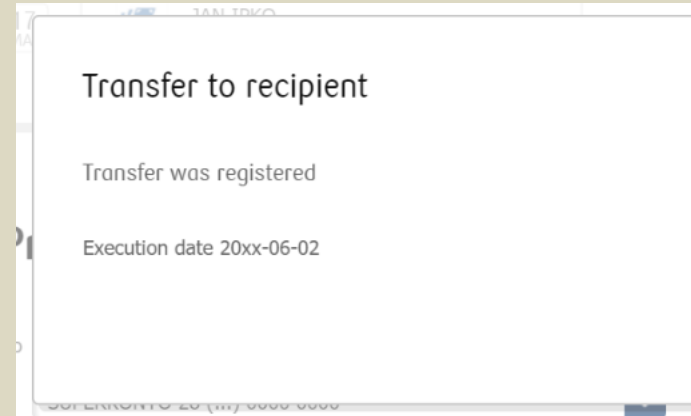
- The process is **very error prone**
- Developers are using same forms and are slightly modifying logic to do normal and “trusted” transfers



Vuln example #1: make it trusted

1. Send unauthorized transfer
2. Add some magic

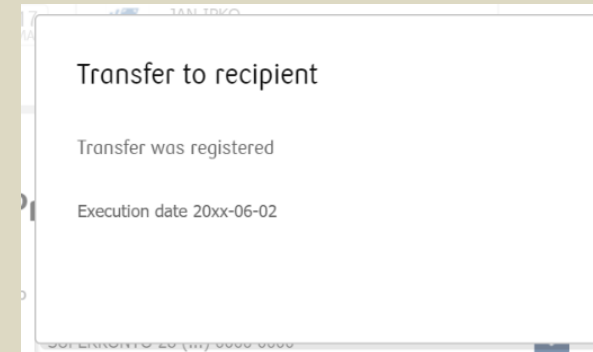
`transferData.trusted=Y`



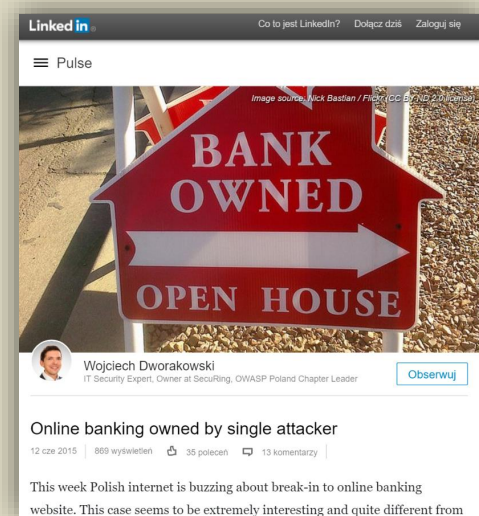
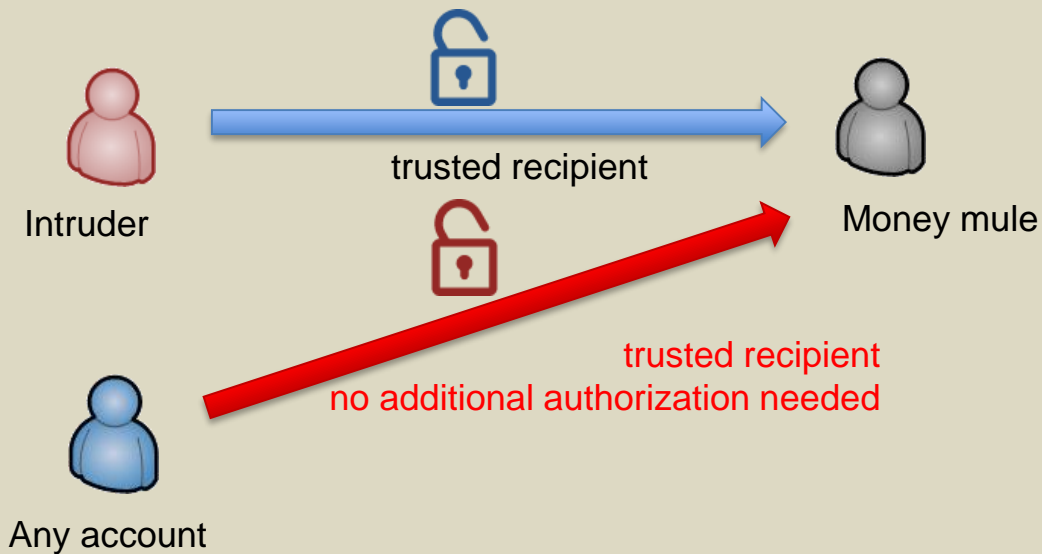
Vuln example #2: overwrite data

1. Create the transfer from trusted recipient template
2. Add (or overwrite) transfer parameters during sending

```
transferData.beneficiary_acc_no=66222200000000006666666666  
transferData.beneficiary_adr=Changed+Address+8%2F5
```



Vuln example #3: business logic error



Yes – it's totally twisted... but it's real

Read more: <https://www.linkedin.com/pulse/online-banking-owned-single-attacker-wojciech-dworakowski>



Trusted recipients

Recommendations

- Decision should be taken server-side
- Carefully control transfer state
- Do not allow additional params
- Control gate (go | not go) at the end of the process





APPSEC
EUROPE

TRANSACTION LIMITS

Transaction limits

Limit type	Amount	Available limit	Awaiting payments [?]	
Daily limit [?]	70.000,00 PLN	63.949,35 PLN	0,00 PLN	> Change limit
Monthly limit [?]	70.000,00 PLN	63.899,35 PLN	0,00 PLN	> Change limit

Limit examples

- transfers / card operations
 - cash / online
 - one-time amount
 - daily / weekly sum
 - daily transactions number
 - business parameters
(e.g. max/min deposit amount)
 - ...
- If the limit is exceeded:
 - forbid operation
 - ask for additional credentials
 - call customer to verify the transaction



Vuln example #1

- Simply – change limits
- Sometimes it doesn't require additional authorization ;)



Vuln example #2: overwrite at confirmation

Enter transaction data below limits

Send form → **Limits are validated** → Confirmation form

Details

Amount:

 PLN

ⓘ Your remaining balance: 1925,00 PLN

Title:

Mode of processing:

 ▼

Next

Back

Details

Amount: 100,00 PLN
Transaction date: 15.06.2016
Title: Test

Details ▼


Confirm the transaction

Transaction does not require authorization with text message code.

Confirm

```
POST /retail/transfer
task=APPROVE_TRANSFER
trn_amount=1000000
```

Confirmation

 Transfer executed



Transaction limits - requirements

- Transaction limits change should require additional authorization (SMS code, one time token, challenge-response, ...)
- Do not allow additional params
- Control gate (check limits) also at the end of the process



APPSEC
EUROPE

NOTIFICATIONS

Transfer

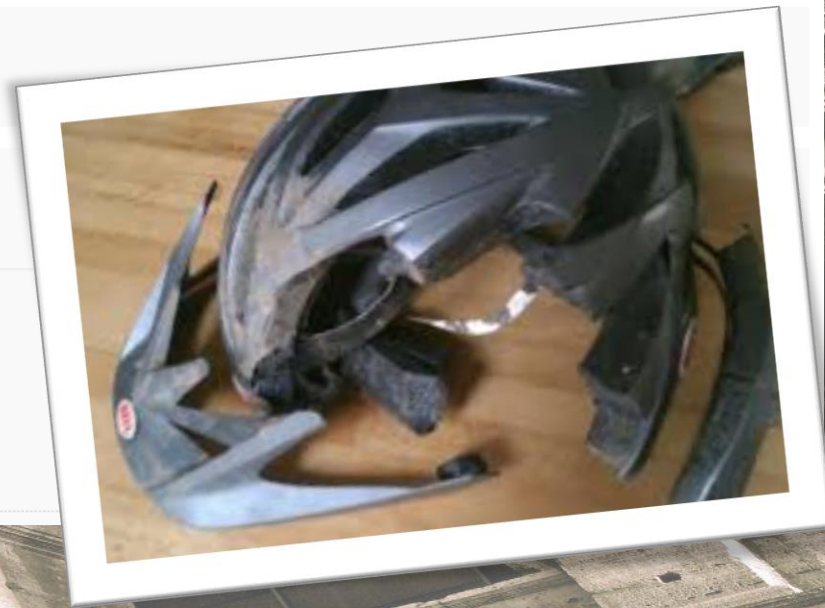
Incoming

Outgoing

Konto walutowe w CHF - 0197723584 [?](#)

Notify from amount

CHF



Vuln examples

- Simply change phone number (or email)
- ...or disable / reconfigure notifications
- Sometimes it doesn't require additional authorization

Notifications - requirements

- Notification change should require additional authorization (SMS code, one time token, challenge-response, ...)
- User should be notified about:
 - wrong authentication attempt
 - (even better) about positive authentication
 - transaction (SUM) above defined limit
 - activation of new access channel (mobile, IVR) or new device pairing
 - password or phone number change





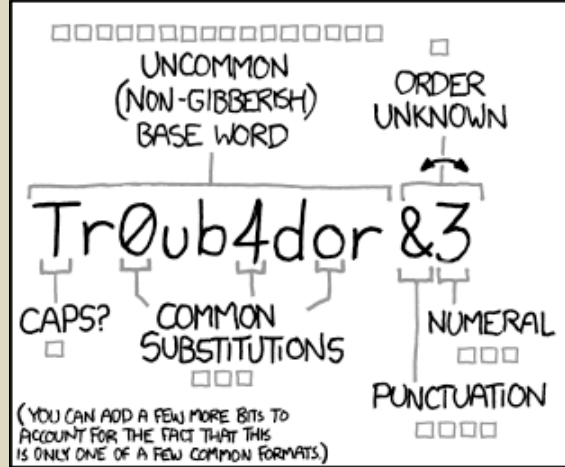
APPSEC
EUROPE

USER AUTHENTICATION



Image: <http://www.cyclingweekly.co.uk/>

We are teaching users to have strong passwords



~28 BITS OF ENTROPY

$2^{28} = 3 \text{ DAYS AT } 1000 \text{ GUESSES/SEC}$

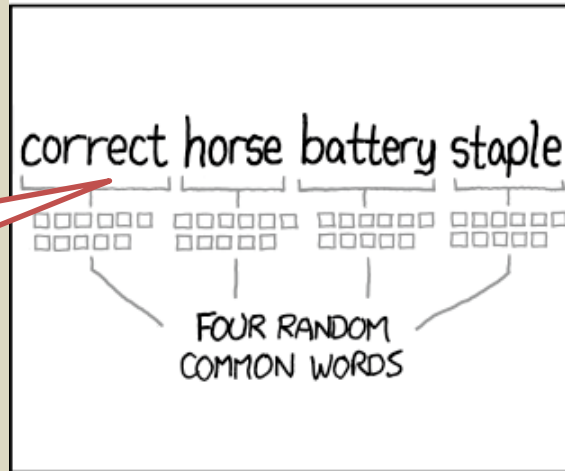
(PLAUSIBLE ATTACK ON A WEAK REMOTE WEB SERVICE. YES, CRACKING A STOLEN HASH IS FASTER, BUT IT'S NOT WHAT THE AVERAGE USER SHOULD WORRY ABOUT.)

DIFFICULTY TO GUESS: **EASY**

WAS IT TROMBONE? NO, TROUBADOR. AND ONE OF THE 0s WAS A ZERO?

AND THERE WAS SOME SYMBOL...

DIFFICULTY TO REMEMBER: **HARD**



~44 BITS OF ENTROPY

$2^{44} = 550 \text{ YEARS AT } 1000 \text{ GUESSES/SEC}$

DIFFICULTY TO GUESS: **HARD**

THAT'S A BATTERY STAPLE.

CORRECT!

DIFFICULTY TO REMEMBER: YOU'VE ALREADY MEMORIZED IT

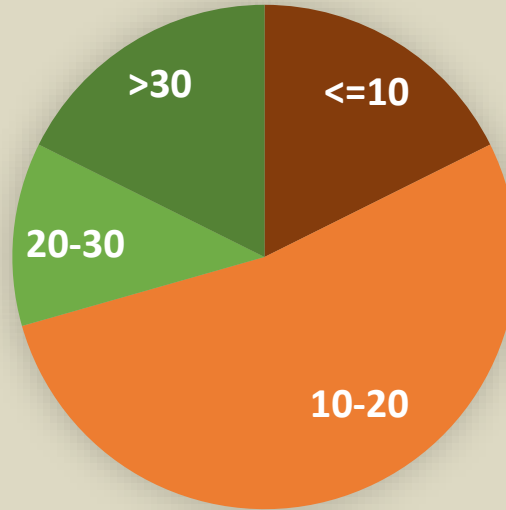
28
characters

THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.



Passphrases and password managers obstacles

Max password length



- Limited length
- Limited chars
 - Extreme case: 8 digits
- Masked passwords

28
characters

Source: Data collected by SecuRing during research on security features used by online banking in Poland (17 banks)





APPSEC
EUROPE

FUTURE – CRASH COURSE IN PSD2



Fot. STEFANO RELLANDINI REUTERS

PSD2

Payment Services Directive (revised)

Major topics:

- SCA – Strong Customer Authentication
- PIS – Payment Initiation Service
- AIS – Account Information Service

Scope:

- Mandatory for all „Payment Service Providers”



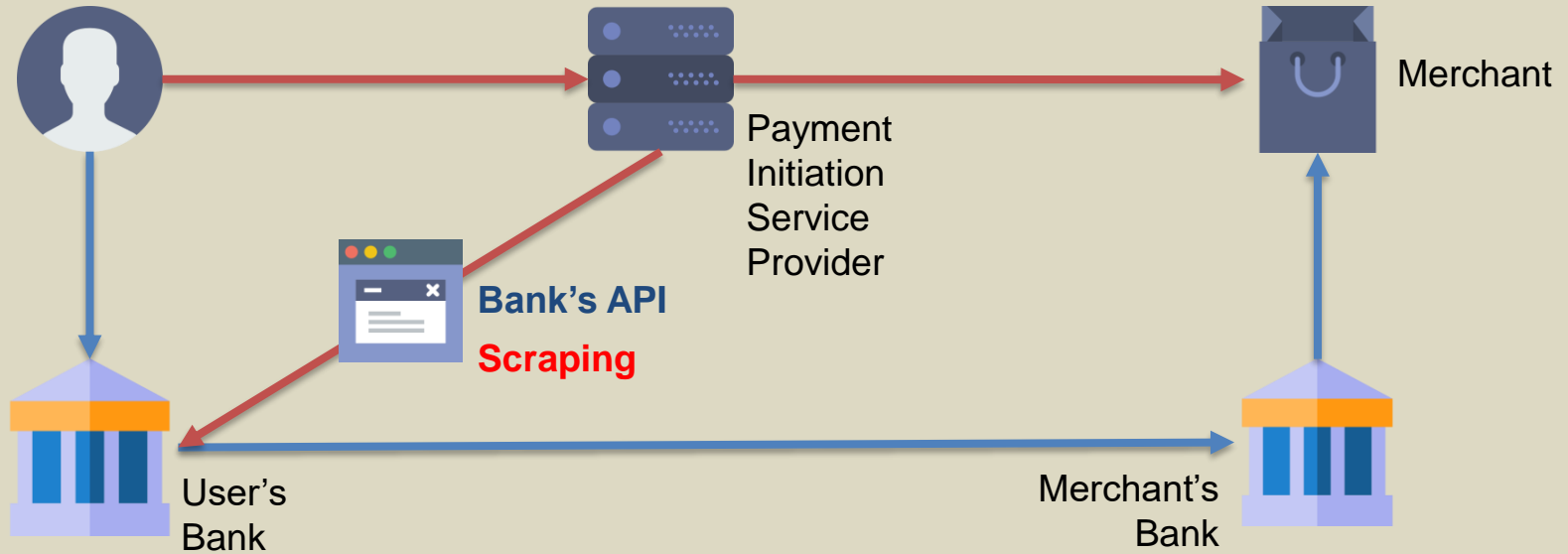
Strong Customer Authentication (SCA)

‘strong customer authentication’ means an authentication based on the use of

- **two or more elements** categorised as
 - **knowledge** (something only the user knows),
 - **possession** (something only the user possesses)
 - **inherence** (something the user is)
- that are **independent**, in that the breach of one does not compromise the reliability of the others,
- and is designed in such a way as to protect the confidentiality of the authentication data;



Payment Initiation Service



PIS – scraping example

Błyskawiczne doładowanie z Twojego konta bankowego

IFKO ▾

Kwota

PLN

Kontynuuj

Trustly 11 PLN PayPal

Wprowadź swój identyfikator użytkownika i hasło do swojego banku internetowego w celu zarejestrowania płatności.

Identyfikator użytkownika: 

Hasło:

Wprowadź swój identyfikator użytkownika

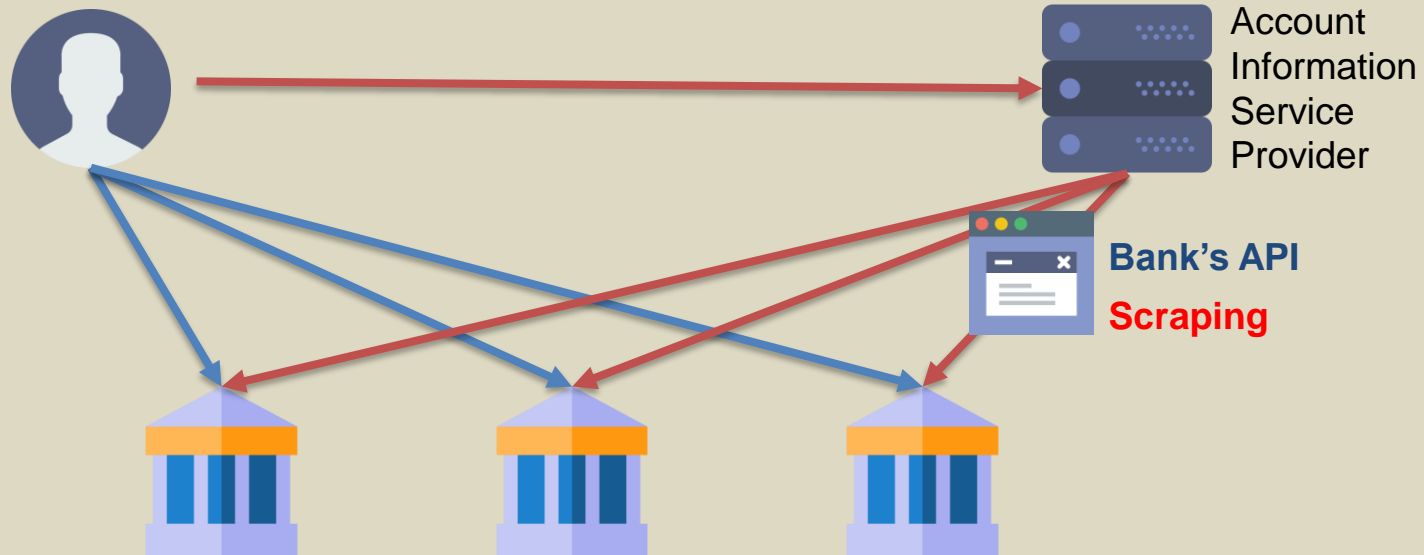
Kontynuuj >

2573086465

Usługę dostarcza Trustly Group AB, nie bank klienta. Dane logowania zostaną przesłane przez platformę płatniczą Trustly, zapewniającą bezpieczne oraz szyfrowane połączenie z bankiem internetowym klienta. W ten sposób przelew bankowy zostanie zrealizowany z konta bankowego klienta w jego imieniu i za potwierdzeniem jego tożsamości. Korzystając z tej usługi, zgadzasz się na [Warunki korzystania z usług](#). W celu uzyskania dalszych informacji odwiedź [trustly.com](#).



Account Information Service



SCA / PIS / AIS

- Possible implementation errors consequences
 - Payment data change
 - Unauthorized access to user's data (mass scale?)
 - Authentication bypass
 - ...



PSD2 - current state

- EBA will issue Regulatory Technical Standard (Jan 2017)
 - Until then, current status quo should be maintained
- EBA released [Discussion Paper](#), call for comments was closed 8 Feb 2016
 - My AppSec related comments:
<https://www.linkedin.com/pulse/strong-customer-authentication-secure-communication-psd2-dworakowski>





APPSEC
EUROPE

SUMMARY & WHAT NEXT?



image:: <http://www.pinkbike.com/>

Implementation errors = vulnerabilities

- If security controls are implemented with vulnerabilities then they are useless
- Vulnerable safeguards cause false sense of security



Precise requirements



OWASP to the rescue !

- Common problems:
 - SQLi
 - XSS
 - ..or features
 - Authentication, 2FA
 - Transaction authorization
- Cheat Sheet Series
 - ASVS
 - Dev Guide
 - SKF
 - ...



OWASP to the rescue ?

Common application business domains (and features)

- online banking / mobile banking
- PIS / AIS (PSD2)
- e-commerce
- SCADA
- social networking
- company webpage

?

Shouldn't we have common requirements for common app domains?

Internet banking - proposal

- Online Banking Cheat Sheet
- ASVS „module”
- Dev Guide chapter





APPSEC
EUROPE

Q & A

@wojtwo

wojtekd@securing.pl

<http://www.securing.pl/en>

