The Bank in the Browser

Defending Web Infrastructures from Malware Attacks

Giorgio Fedon
Owasp Antimalware Project
Founder

giorgio.fedon@mindedsecurity.com

OWASP
EU09 Poland

The OWASP Foundation

http://www.owasp.org

About Anti-malware Project

- Antimalware is not a product, but a free and open Owasp project:
  - Embrace the philosophy of protecting the banking customer: *The Bank in the Browser*
  - Document Banking Malware Attacks
  - Model and Evaluate exposure of Banking provided security Measures to Malware Attacks
  - Define the best practices and how to fight Banking Malware
  - Rise Awareness
- Join us at: owasp-anti-malware@lists.owasp.org

Owasp Antimalware Goals

- Create a strong knowledge base about what malware do against Banking Portals
  - Build an updated reference focusing on malware features used to attack Web security measures
- Define security requirements to counter-attack malware
  - Tell to the industry what works against malware and what's not
  - Often victims of malware have not been compensated on suspicion of policy infringement
- Open Awareness program
  - Teach users about risks connected to malware

About Myself

- Research
  - OWASP Antimalware project leader
  - Testing Guide Contributor
  - Analysis and discovery of important security vulnerabilities

- Work at Minded Security
  - Chief Operation Officer
  - Leading hundreds of Penetration Testing activities and Code Reviews; many of them for the Bank Industry
  - Blog: http://blog.mindedsecurity.com

Agenda

- Introduction
- Banking Attack Process
- Banking Malware Families
- Threat Modeling for Banking Malware Attacks
- Security Rating
- Best Practices Against Banking Malware

Introduction

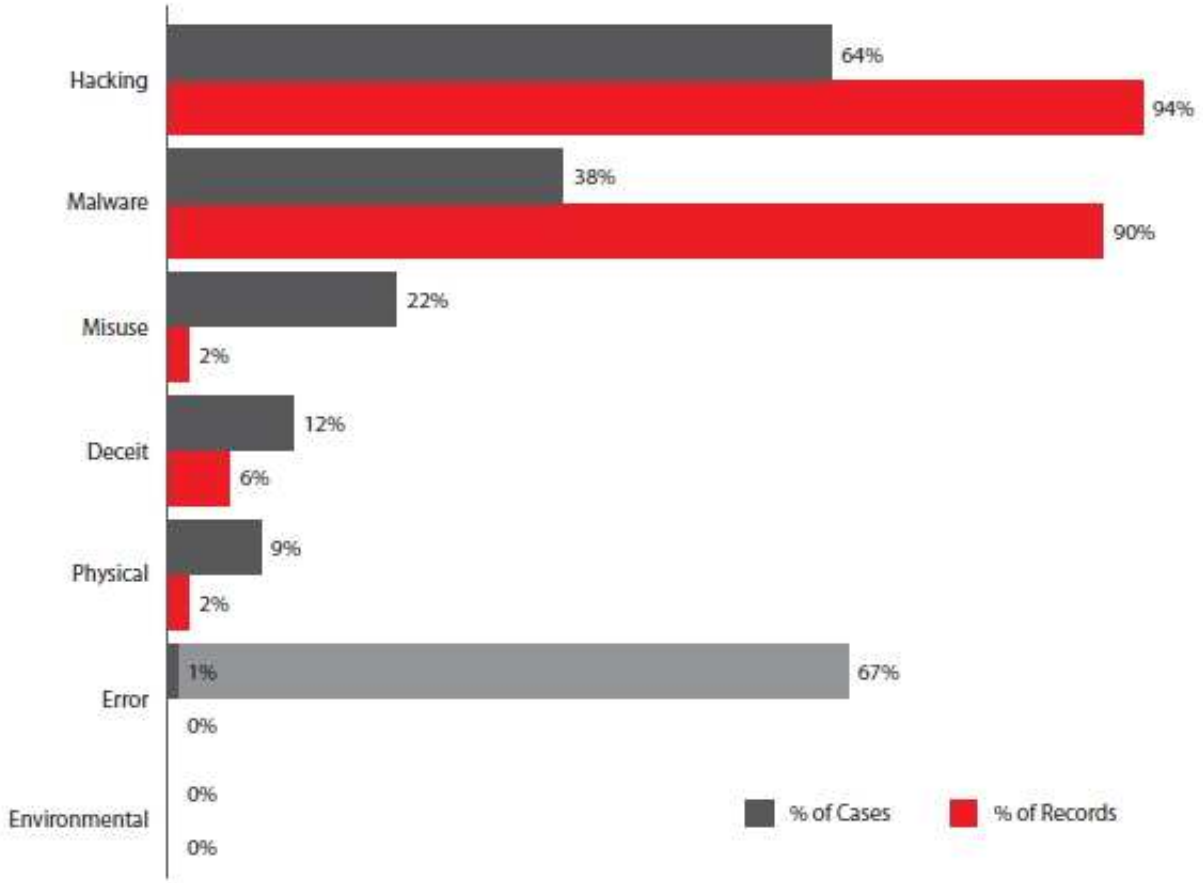
Recent items in the news

- "*Swedish bank* has informed the press that it has been stung for between seven and eight million *Swedish* krona — up to £580000" by a single Malware attack
- "*Silent Banker* Trojan Targets 400 Banks, Circumvents Two-Factor Authentication, just for starters"
- "Banking Spyware use stealth Techniques to hide and some of them are very advanced, e.g. Mebroot"
- A security breach hit CardSystems Solutions resulting in the compromise of 40 million credit card account numbers.
- Custom Keyloggers at Sumitomo provided IDs and passwords to intruders in an attempt to wire $423 Million out of the bank.

What are you up against?

- Malware threats are often made up by professional tools developed by specialized software factories
  - Unethical companies trade this type of tools across the Black market
- Companies are the main target
  - Organized crime wants the big money
  - Vast majority of transaction frauds
  - Downgrade trend (XP vs. Vista, Static Passwords vs. Dynamic Tokens)
- Remember that Malware targets anyone

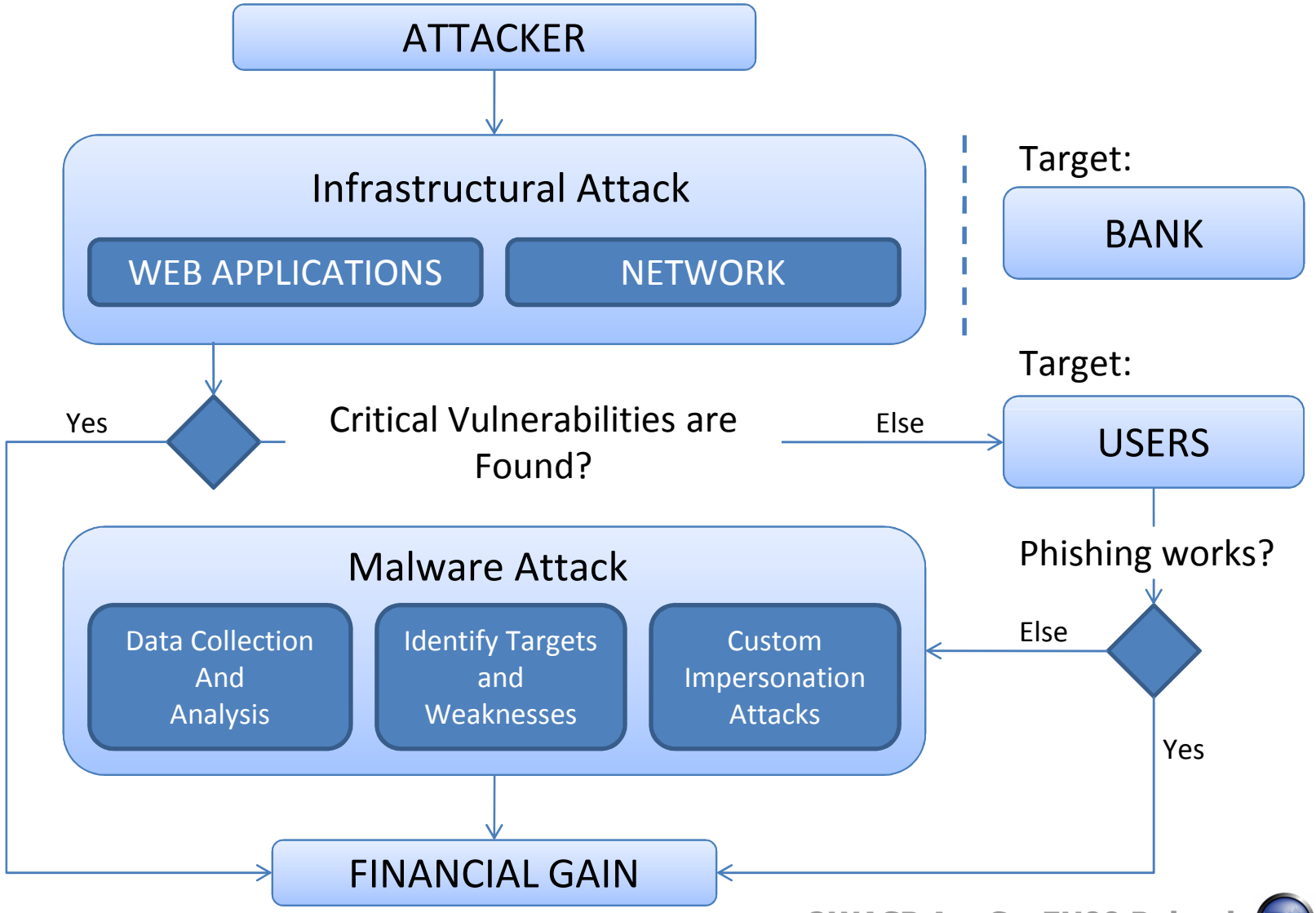
Attack Statistics



Source: Verizon Data Breach Report 2009

Banking Attack Process

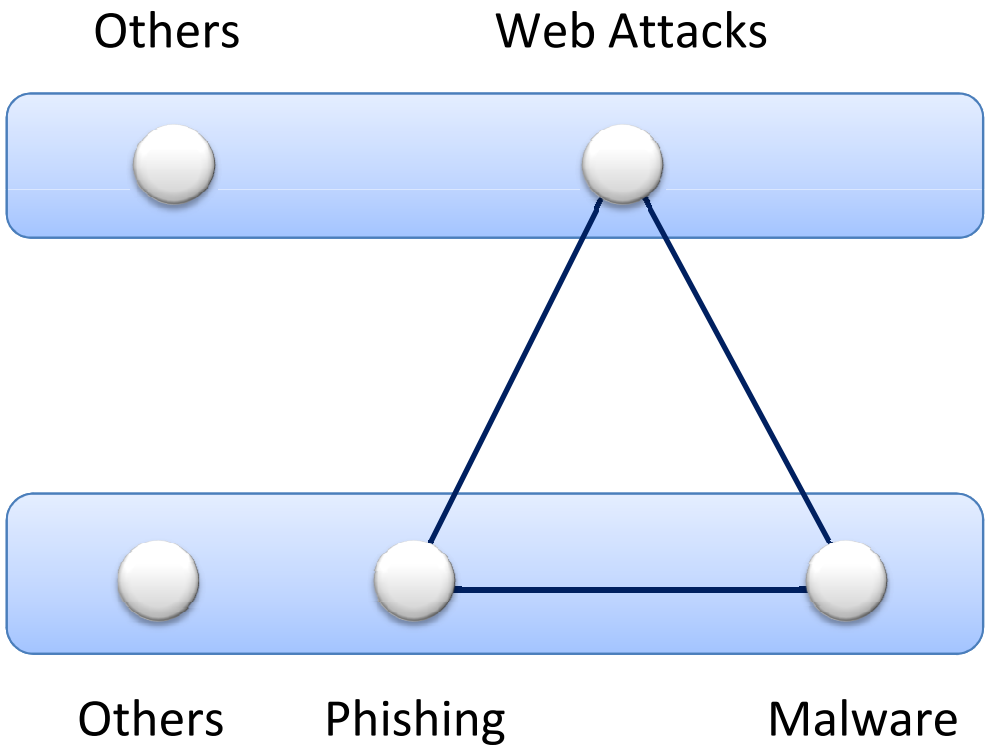
Beginning of Banking Attack



Attack Interactions

Mutual Empowerment

- Direct infrastructural attacks increase the strength of user attacks and vice-versa
- Web Application security design, should involve the definition of security requirements also to contain user attacks



Attacks against infrastructure

+

Attacks against the users

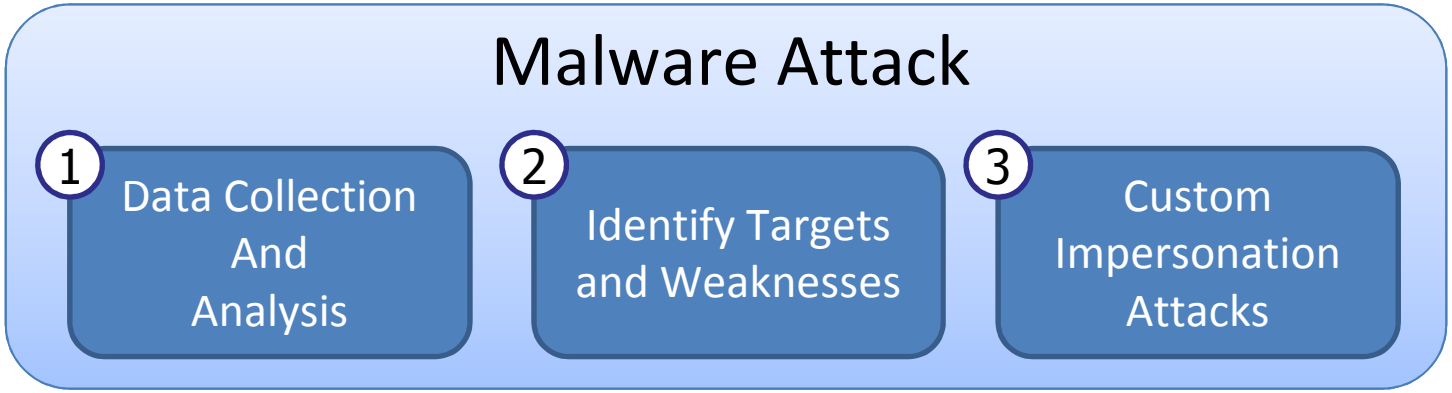
Attack Interactions (2)

- Bank infrastructure
  - Web Attacks: direct attacks against the web infrastructure
  - Others: Network Attacks
- User devices
  - Phishing Attacks: luring the user into doing something wrong
  - Malware Attacks: execute malicious code on a remote client, in order to control or spy the victim
  - Others: DNS Rebinding, Router Hacks, etc.

Attack Interactions (3)

- Web Attacks add points to Malware Attacks
  - Challenge Code Predictability permits to phish the next token code (e.g. next grid-card value)
- Malware Attacks add points to Web Attacks
  - Attacker steals session using Malware, then exploits an internal SQL Injection

Details of Malware Attack process



1. Dropzones are the places where data is collected; preliminary attacks just log any HTTP traffic from the banking session
2. From the obtained info, the attacker studies the bank security measures and what the bank offers (transition graphs and security boundaries)
3. The attacker creates a custom configuration entry and updates the malware remotely

Data collection and analysis

- Analysis of information harvested (Silent Banker)
  - The attacker tries to harvest all information about user browsing session
  - Following configuration tells to log all HTML coming from the website (use of wildcards is important):

```
ghjfe87=0
hgknc87=*secure.newbank.com
hgknn87 = <html>
```

  - HTML pages harvested are in order of millions. This help to familiarize with unknown portal structures
  - Recent analysis of Torpig, shows the same approach

Identify the target

- Choose the target
  - From our analysis we can tell for sure that targets are chosen from usage statistics
  - Usage statistics are influenced by the behavior of the infected population
- Malware author monitors URL visited
  - from analysis of security measures, they decide if a customize impersonation attack is needed

Custom Impersonation Attacks

- Attack Strategy
  1. Intercept user credentials in clear text and reuse them
  2. Trick the user into authorizing the wrong transaction
- Most effective way to reach these goals
  - Rewrite the user interface (Local MITM aka MITB Man in The Browser aka HTTP injection)
  - Monitor Mouse Clicks (screen grab feature)
- Attacks need to be customized
  - Bank pages to monitor
  - HTML code to be injected

Custom Impersonation Attack (2)

- Custom HTML injection (Silent Banker)

```
[jhw144]
pok=insert
qas=secureportal.bank.cm/index.do
dfr=16
req=100
xzq=9
rek=<input type="hidden" name="username_phish" value="">
<input type="hidden" name="password_phish" value="">
njd=name="login_Form"
xzn=value="">
```

This configuration will make the malware searching for the "*login_Form*" string as an anchor point, and then inserting the fields in defined in "rek" after next *value=""*> string

Return on Investment

Zeus and Nethell Dropzones

Information Category	Number	Percentage
Credit Cards	5682	3,44
Paypal	5000	3,02
Bank Accounts	5200	3,15
Email Passwords	149458	90,39

Rif: Holz, Engelberth, Freiling - Learning more About the Underground Economy

Silent Banker Dropzone

Information Category	Number	Percentage
Credit Cards	1120	6,35
Bank Accounts	865	4,91
Paypal	220	1,25
Email Passwords	15430	87,5

Rif: Owasp Antimalware

Torpig Dropzone

Information Category	Number	Percentage
Paypal	1170	1,84
Bank Accounts	6600	10,39
Credit Cards	1160	1,83
Email Passwords	54590	85,94

Rif: Stone, Cavallari, Vigna and others
Your Botnet is My Botnet: Analysis of Botnet takeover

The Rise of Javascript Banking Malware

- Crime-ware injects locally HTML and Javascript into the pages surfed by the user
- This attack is called Local Man in the middle or Man in the Browser

- Local Man in the Middle can be performed without compromising **either the user host or the banking website**?

The Rise of Javascript Banking Malware (2)

- Many pages include and not validate third parties content
  - Tracking Javascript code
  - Callcenter help buttons
  - News, Market Trends etc.
- Partner websites are constantly checked? Answer: **NO**
- "Modifying the Javascript Code, the attacker gets full control on the browser", like with a local MITM malware attack*
- Potential backdoor in "https//www.bank.com/login.do"

```
<!-- BEGIN Marketing Tag. PLACE IT BEFORE THE /BODY TAG -->
<script language='javascript' src='https//www.unsafeagency.com/bank.com.js' >
<!– END Marketing Tag. -->
```

* "Subverting Ajax Paper" – Prototype hijacking
Active MITM Attacks paper – Saltzman, Sharabani

Banking Malware Families

Banking Malware Evolution

- In 2003 very few malicious codes were able to bypass javascript keyboards
- In 2008 Banking Malware starts using amazing rootkit technologies. Mebroot (New Version of Trojan Anserin) is able to infect the MBR (Windows XP and Vista) and to patch the kernel in real time to hide his presence.
- In 2009 more and more custom attacks are emerging, as ATM Machine rootkits and Malware able to render visual Captchas*

*Ref. http://www.pcworld.com/businesscenter/article/161854/german_police_twofactor_authentication_failing.html

Banking Malware Evolution (2)

- We assisted to the born of different banking malware samples:
  - Silent Banker
  - Haxdoor
  - Banker.C (aka Zeus/Zbot/NTOS)
  - Banker.D (aka Limbo/NetHell)
  - Torpig/Sinowal/Anserin-MebRoot

- Banking Malware aka Crimeware are modified versions of common threats known as password stealing trojan. However they have additional features to attack bank authentication systems, such as *multiple factor authentications*

Features of Banking Malware

- Following features are the ones used to attack the Bank security measures
  - **Browser API Hooking:** Ability to intercept submitted text in forms or HTTP traffic
  - **Local Man in The Middle:** Ability to manipulate the HTTP traffic from the local machine
  - **Remote Man in The Middle:** Ability to redirect HTTP requests to remote sites
  - **Screencapture:** Ability to defeat JS keyboards or sim.
- Banking Malware has many features
  - Rootkit technology, Control Center, Covert Channels, etc.

Silent Banker

- Found in the wild targeting more than 400 banks
  - The “engine” is separated from the configuration files
  - Settings vary from region to region
  - From our analysis less than ¼ of all banks have fine customized rule-sets

Feature	Need Specific Configuration Entry
Browser API Hooking	No (generic patters are definied)
Local MITM	Yes
Remote MITM	Yes
Screencapture	Yes (needs URL to target)
Remote Update	Yes (upgrades and additional features)

Haxdoor → Adrenaline

- Responsible of Nordea attack in 2005
  - Discontinued since 2006
  - Found to target not more than 20 different banks
  - Available in the black market for 1500 euros

Feature	Need Specific Configuration Entry
Browser API Hooking	Yes
Redirection to pharming sites	Yes
Remote Update	Yes (upgrades and additional features)

- Features added in Adrenaline

Feature	Need Specific Configuration Entry
Local MITM	Yes
Screencapture	Yes

Zeus

- One of the most spreaded
  - Trojan Horse, some versions are packaged with a custom Mp3 player
  - Similar to Nethell
  - Crime-ware authors copy from each other

Feature	Need Specific Configuration Entry
Browser API Hooking	Yes
Local MITM	Yes
Remote MITM	Yes
Screencapture	Yes (Needs URL to target)
Remote Update	Yes (upgrades and additional features)

Net Hell

- Flexible configuration
  - Very similar to Silent Banker and Zeus
  - Samples as late 2008 has a powerful html injection and remote control system

Feature	Need Specific Configuration Entry
Browser API Hooking	No (generic patters are definied)
Local MITM	Yes
Remote MITM	Yes
Screencapture	Yes (Needs URL tu target)
Remote Update	Yes (upgrades and additional features)

- http://www.virusbtn.com/pdf/conference_slides/2007/LuisCorronsVB2007.pdf

Torpig/Sinowal/MebRoot

- Crimeware with the most powerful rootkit
  - MBR infection
  - Engine is updated once a month to remain undetectable

Feature	Need Specific Configuration Entry
Browser API Hooking	No (generic patters are definied)
Local MITM	Yes
Remote MITM	Yes
Screencapture	Yes (Needs URL to target)
Remote Update	Yes (upgrades and additional features)

- Your computer is now Stoned (Kasslin, Florio) - http://www.f-secure.com/weblog/archives/Kasslin-Florio-VB2008.pdf
- Taking over the Torpig Botnet - http://www.cs.ucsb.edu/~seclab/projects/torpig/index.html

Considerations about Malware

- In 2003 we were used to talk about "Common Malware"
  - We can no-more discriminate since most of the capabilities are the same for all the different banking malware families analyzed
- Most of Banking malware need customized settings to work properly
  - However If your bank institution is not in the list doesn't mean to be safe
    - Configuration are easy to make
    - Banking Malware can be installed as a component

Threat Model for Banking Malware Attacks

Banking Malware Attacks

- Malware Attack takes place when malicious code is executed on user client
  - Banking Malware Attacks resemble Phishing Attacks, but they can manipulate data in *real time*, in *both directions*
- To evaluate the exposure of an infrastructure to malware attack, we need to consider
  - The strength of authentication/authorization security measures adopted
  - Probability of Malware diffusion among the users
    - Difficult to know without sampling

Evaluate Exposure To Malware Attacks

- Threat Modeling Process
  1) Enumerate the interesting targets
  2) Define the path to the targets (Transition graphs)
  3) Apply trust boundaries (security measures)
  4) Define the weaknesses of the security measures adopted

- Risk Rating
  - Rate the effort to trespass the security measures by attacks performed with different kind of Malware

Tipical Banking Attacker Targets

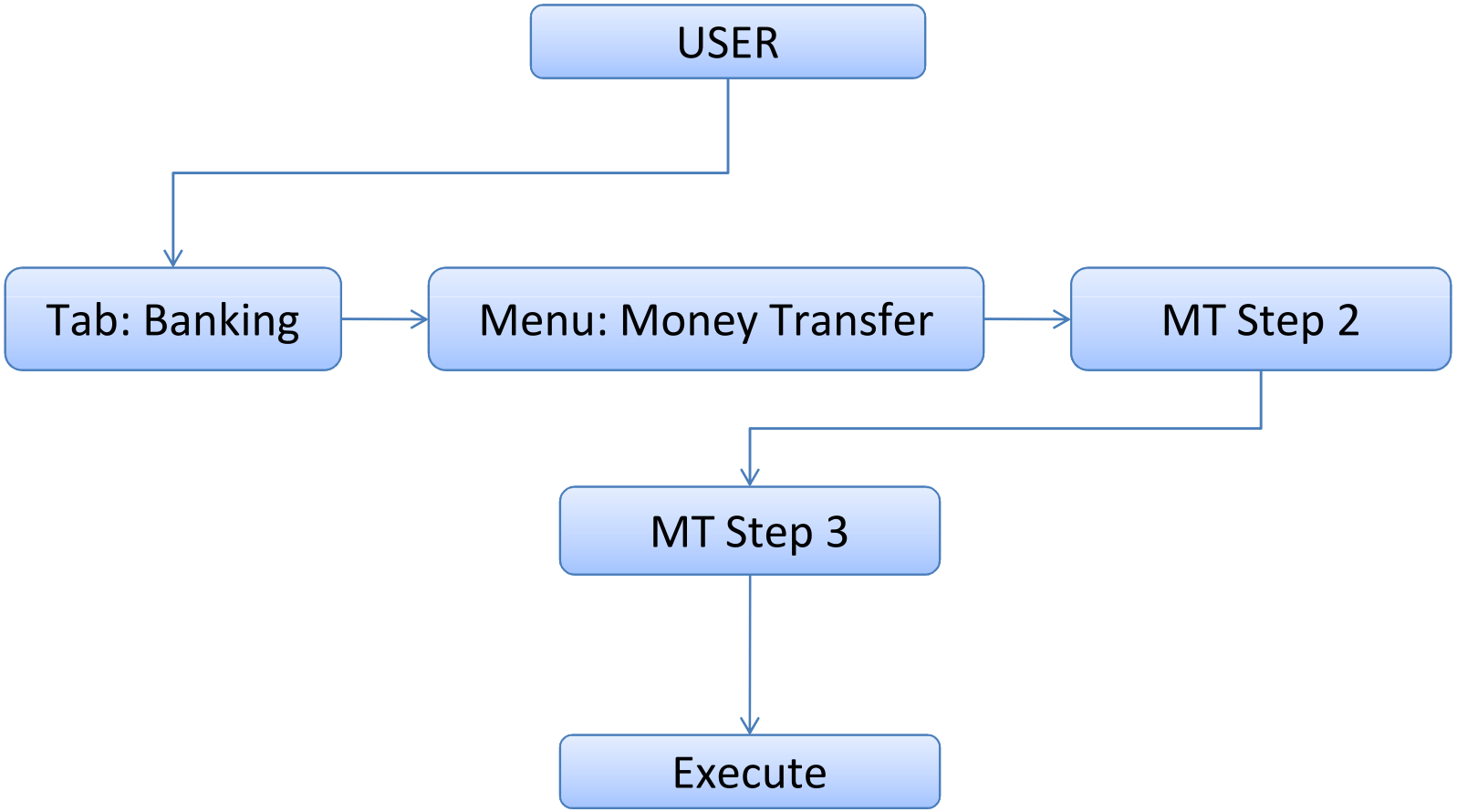
- Get Important Information
  - Credit card information
  - User Credentials and Transaction Tokens
  - User Details
- Abuse Banking Functionalities
  - Transfer Money
  - Modifying user details for receiving goods (e. Checks)
- Abuse Trading Functionalities
  - Buy, Sell (Pump and Dump)
- Covering Tracks
  - Disable Notification Alerts

Transition Graphs

- Show all the known paths in the application to reach a target
- Visual representation of authentication/authorization checks
- Separate attacker's goals from attack trees
- Portals with similar subsets of functionalities can have very different transition graphs
- Important to define the effect of layered security measures
- This approach follows the logic of the attacker

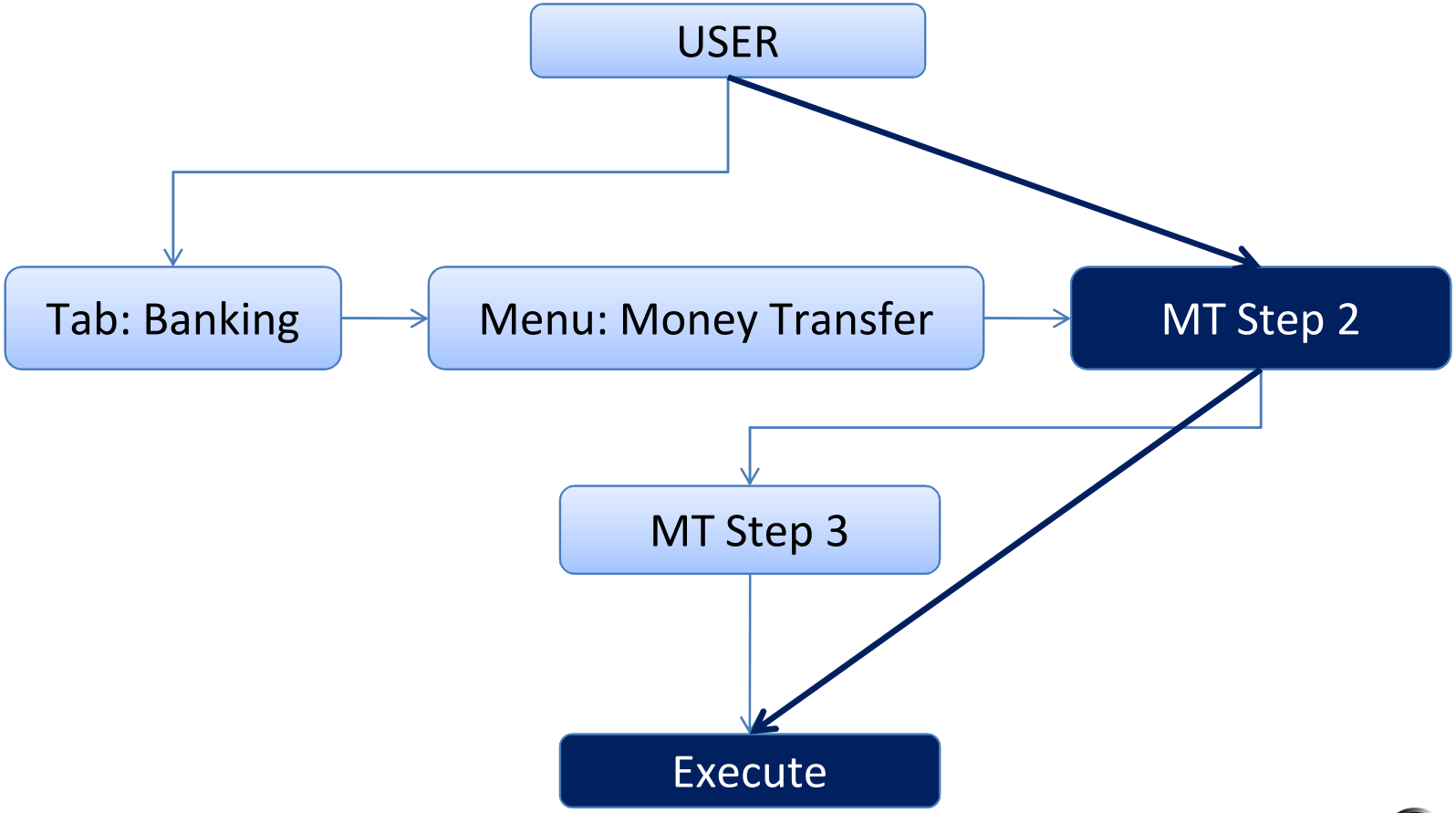
Transition Graphs (2)

- Example of Money Transfer



Transition Graphs (3)

- Define Primary Nodes
  - Represents the least number of steps to complete the process



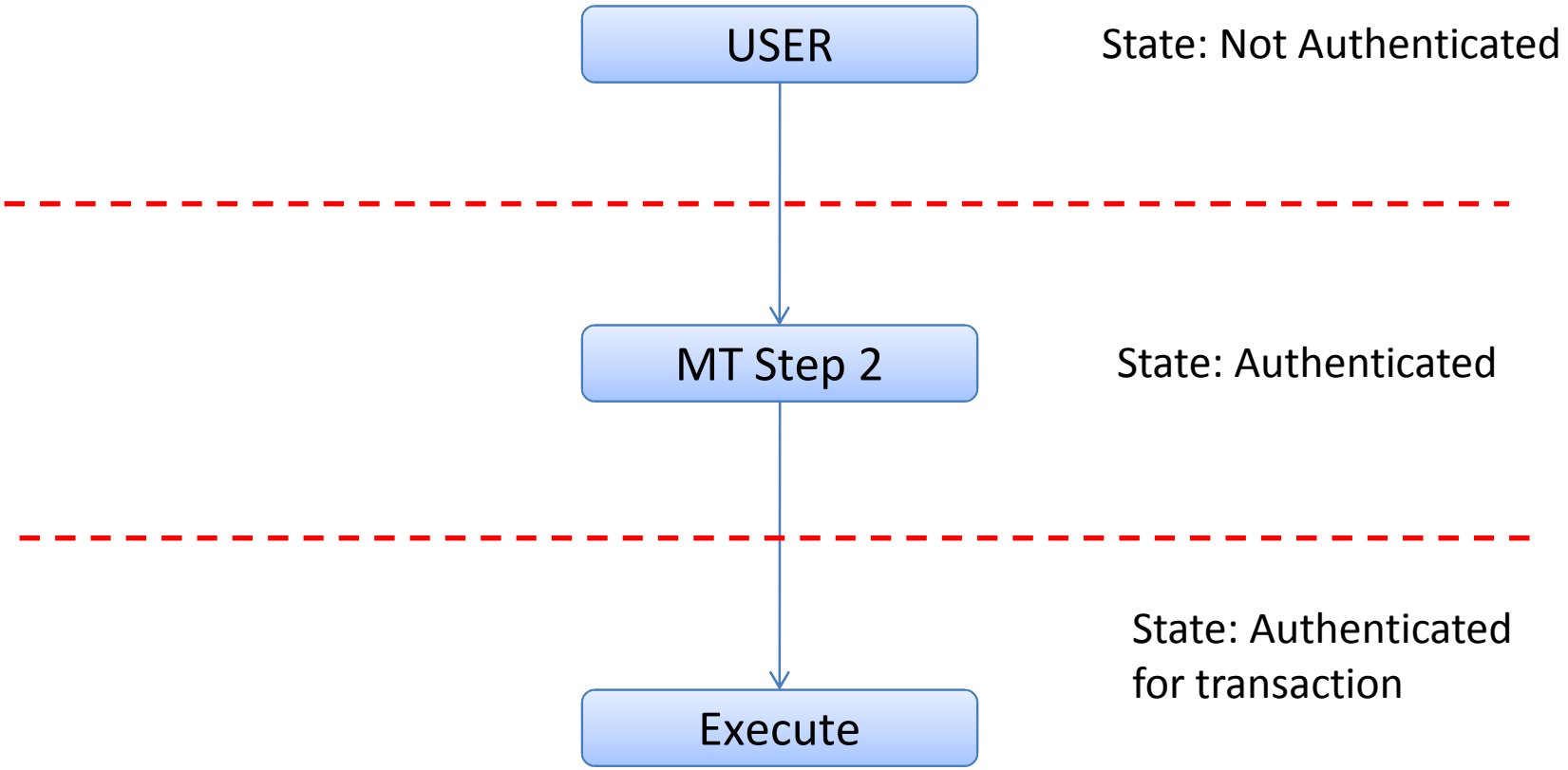
CodeReview and Transition Graphs

- Global view and representation of all functionalities, even the hidden ones from the user interface
- Comprehensive check of Cross Site Request Forgeries vulnerabilities
- Control all nodes to have the appropriate Authentication/Authorization set
- Check for old functionalities that are still active and their duplicates

Apply the Trust Boundaries

Applies to:
corporate.bank.cm
retail.bank.cm

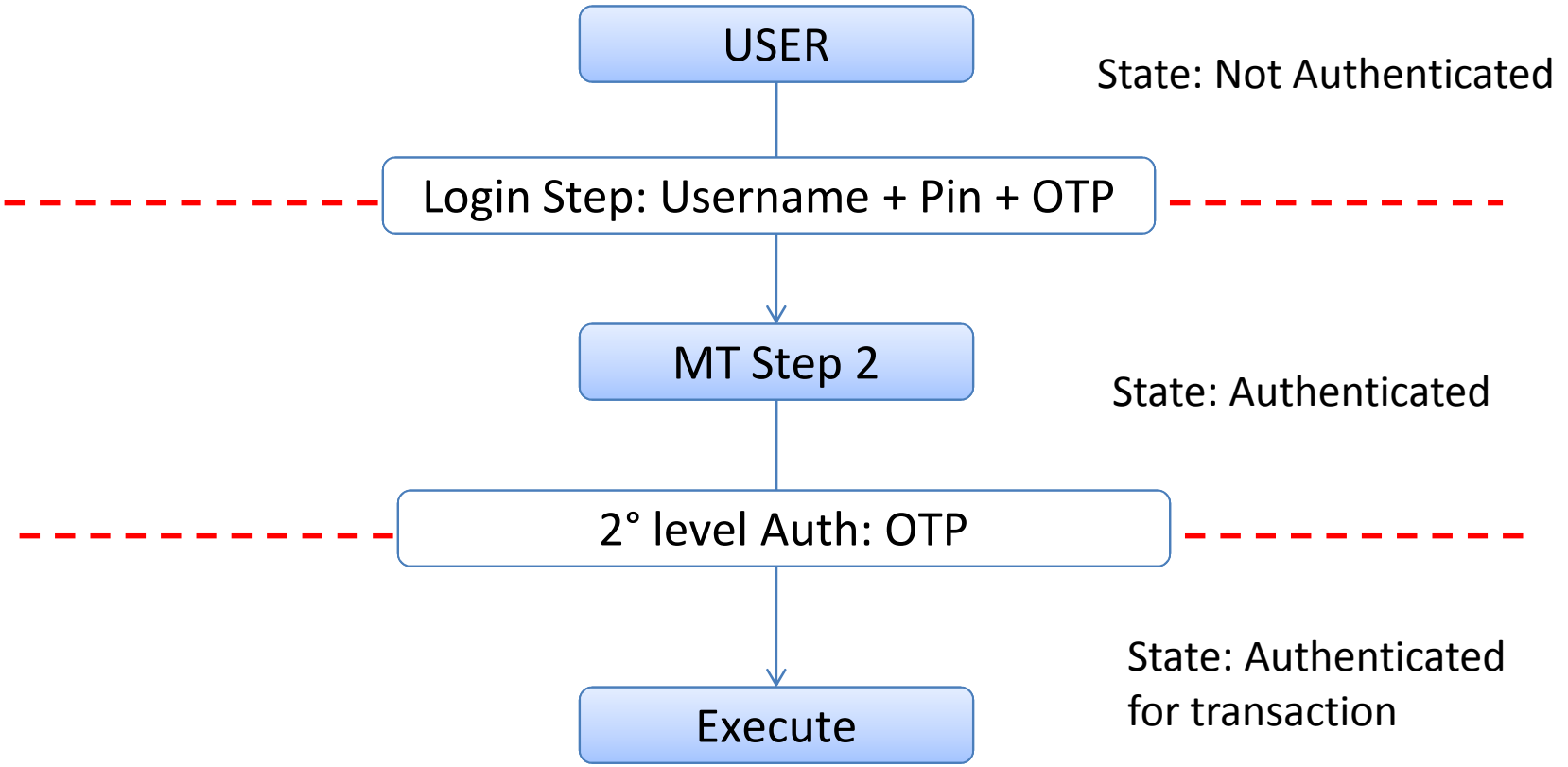
- Example of Money Transfer



Apply the Trust Boundaries (2)

Applies to:
retail.bank.cm

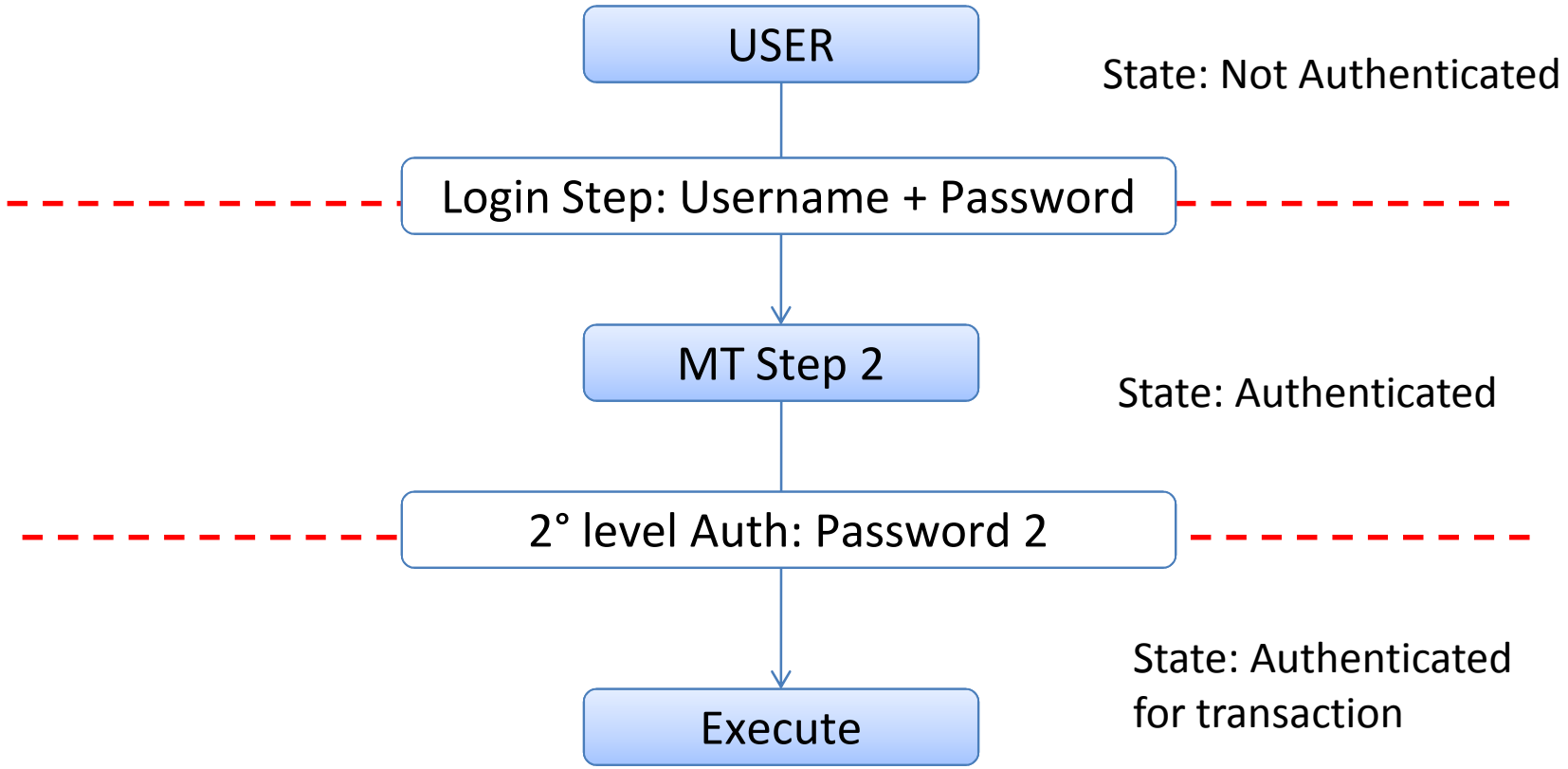
- Trust Boundaries are defined by security checks



Apply the Trust Boundaries (3)

Applies to:
corporate.bank.cm

- Different profiles may have different security measures applied



List Nodes and their associated Security

- Following table is very important from the point of a security assessor
  - Understand Autentication and Authorization steps
  - Report anomalies from the defined policies
    - Es. Security Measure Downgrade for corporate users

Functionality	ID	Primary	Level
Transfer Money	Menu_TF	no	Authenticated
Transfer Money	step1_TF	no	Authenticated
Transfer Money	step2_TF	Yes	Authenticated
Transfer Money	Execute_TF	Yes	Authenticated For Trasaction

Level	Profile	Security Solution
Authenticated	Retail	username + PIN + OTP
Authenticated	Corporate	Username + Password
Authenticated For Trasaction	Retail	OTP
Authenticated For Trasaction	Corporate	Password2

Web Application Security is a requirement

- Lack of server side application security has significant effects on our analysis

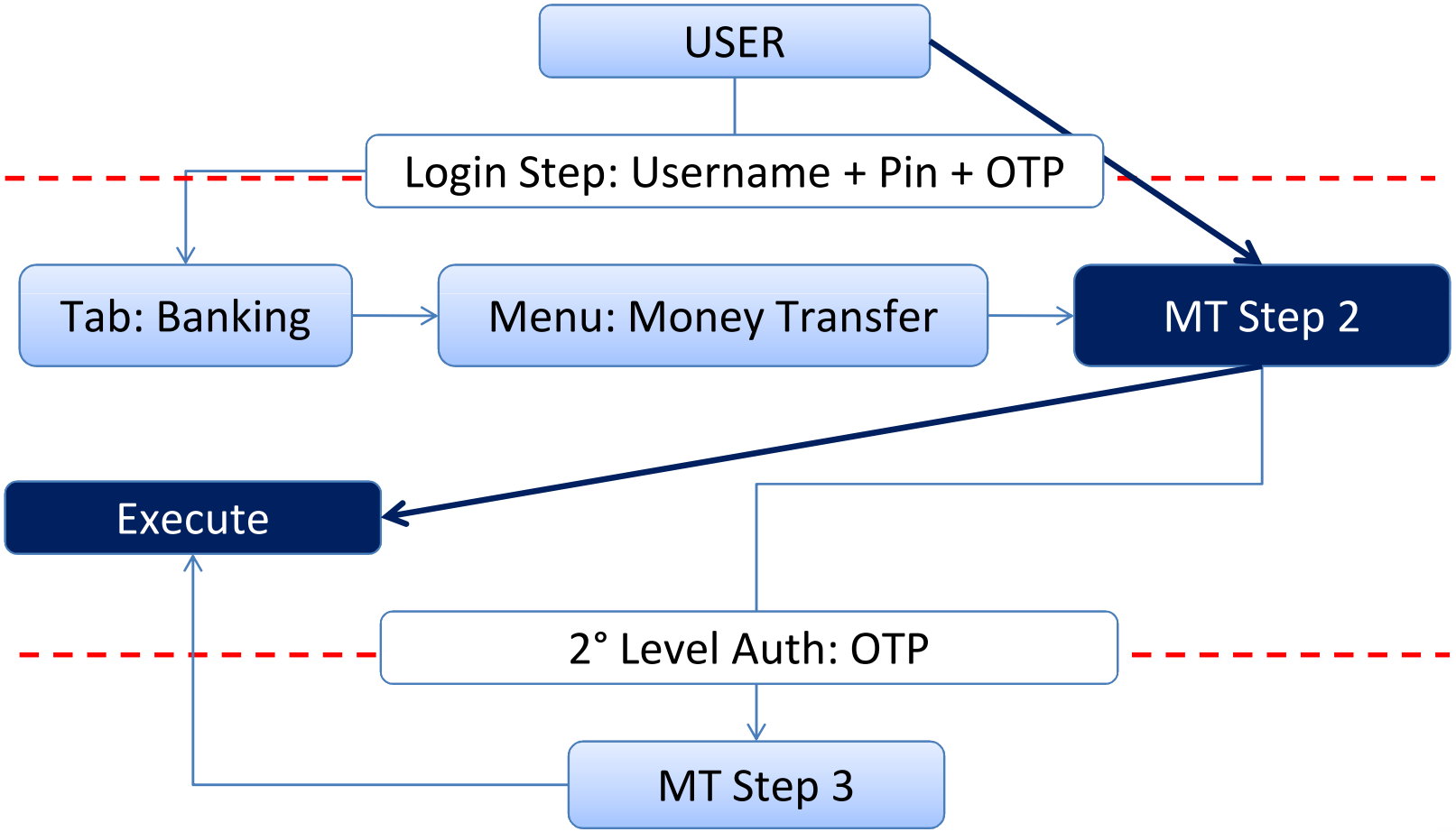
Effects of Web Vulnerabilities on Analysis

- Highly critical Vulnerabilities on any web page can lead to system compromise or to bypass the Authentication / Authorization controls
- Unknown Web Vulnerabilities, if discovered, change consistently the transition graphs or create new path for attacks

- On the other hand client-side (eg. XSS) attacks are equivalent Malware Attacks
  - The attacker gets control over the victim browser

Es. Broken Access Control and CSRF

- 2nd Level Auth effectiveness is lowered to 0

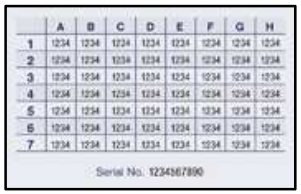


Banking Provided Measures

Banking provided Security measures

- Password

- TAN (Gridcard, Scratch Card)
  - Transaction Authorization Numbers



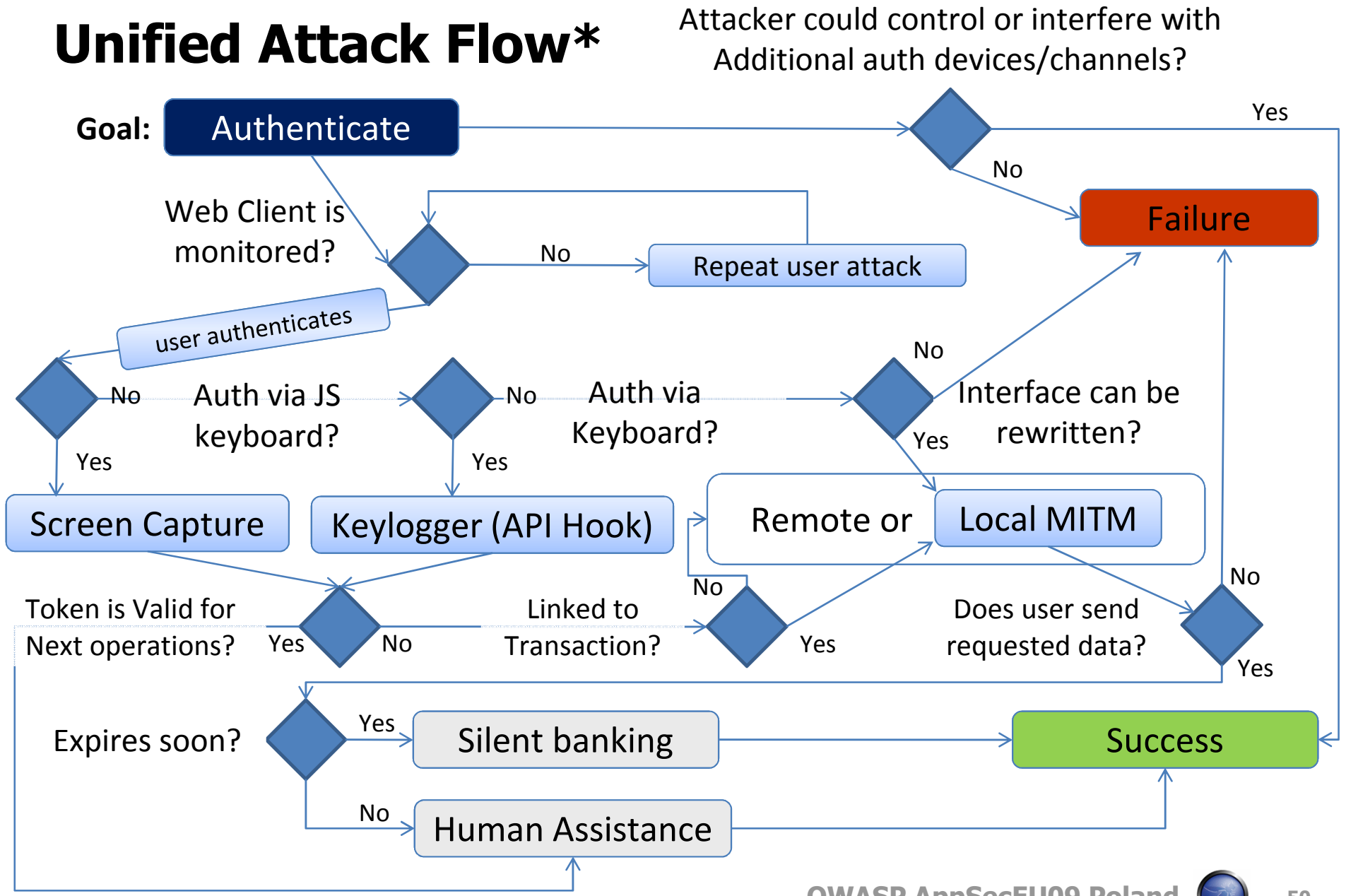
- OTP (Time Based, Click Based)
  - One Time password

- CAP (Random Nonce, Challenge Response)
  - Card Authentication Protocol; Random Nonce is like OTP

- SMS Challenges

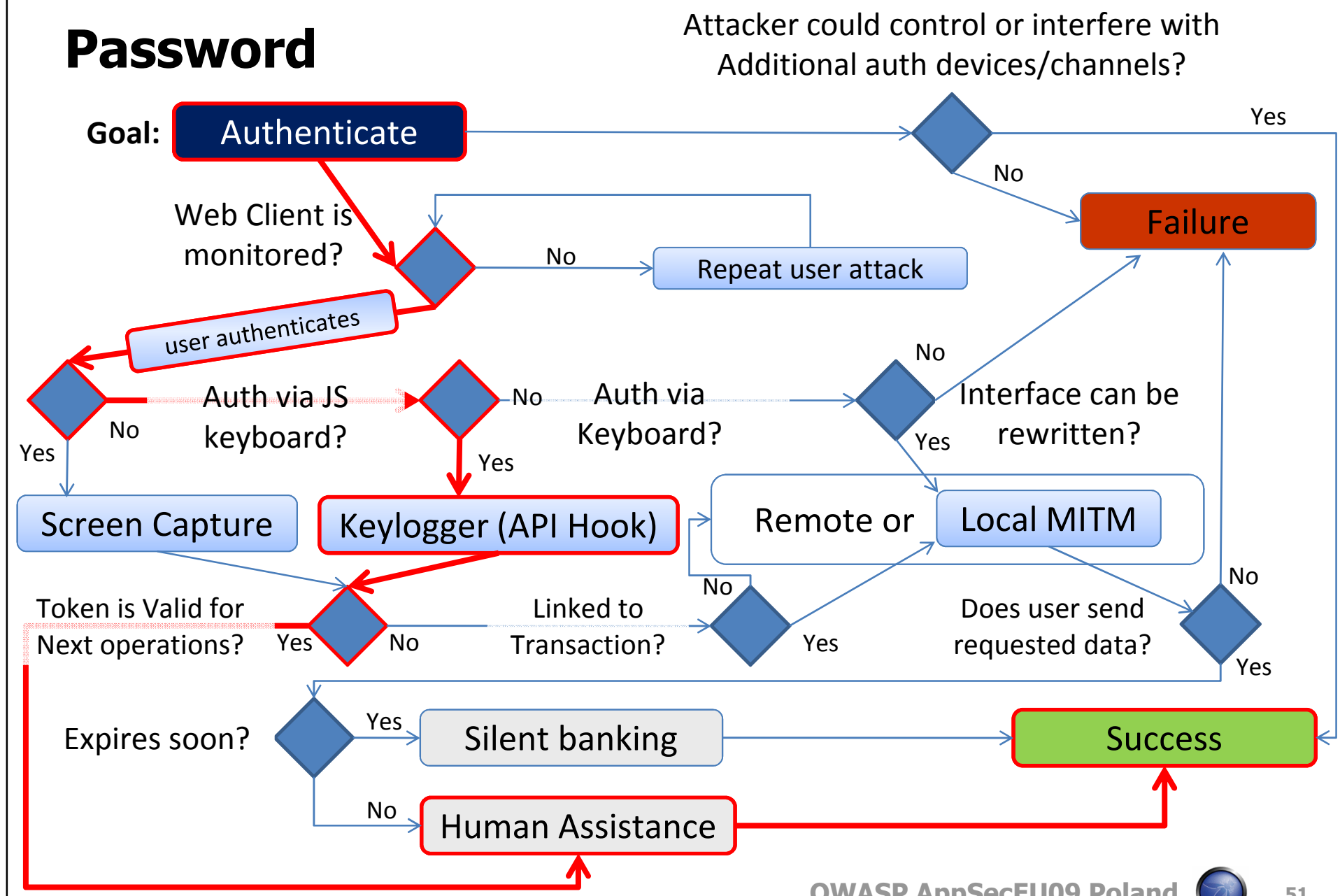
- Cellphone Caller-ID

Unified Attack Flow*

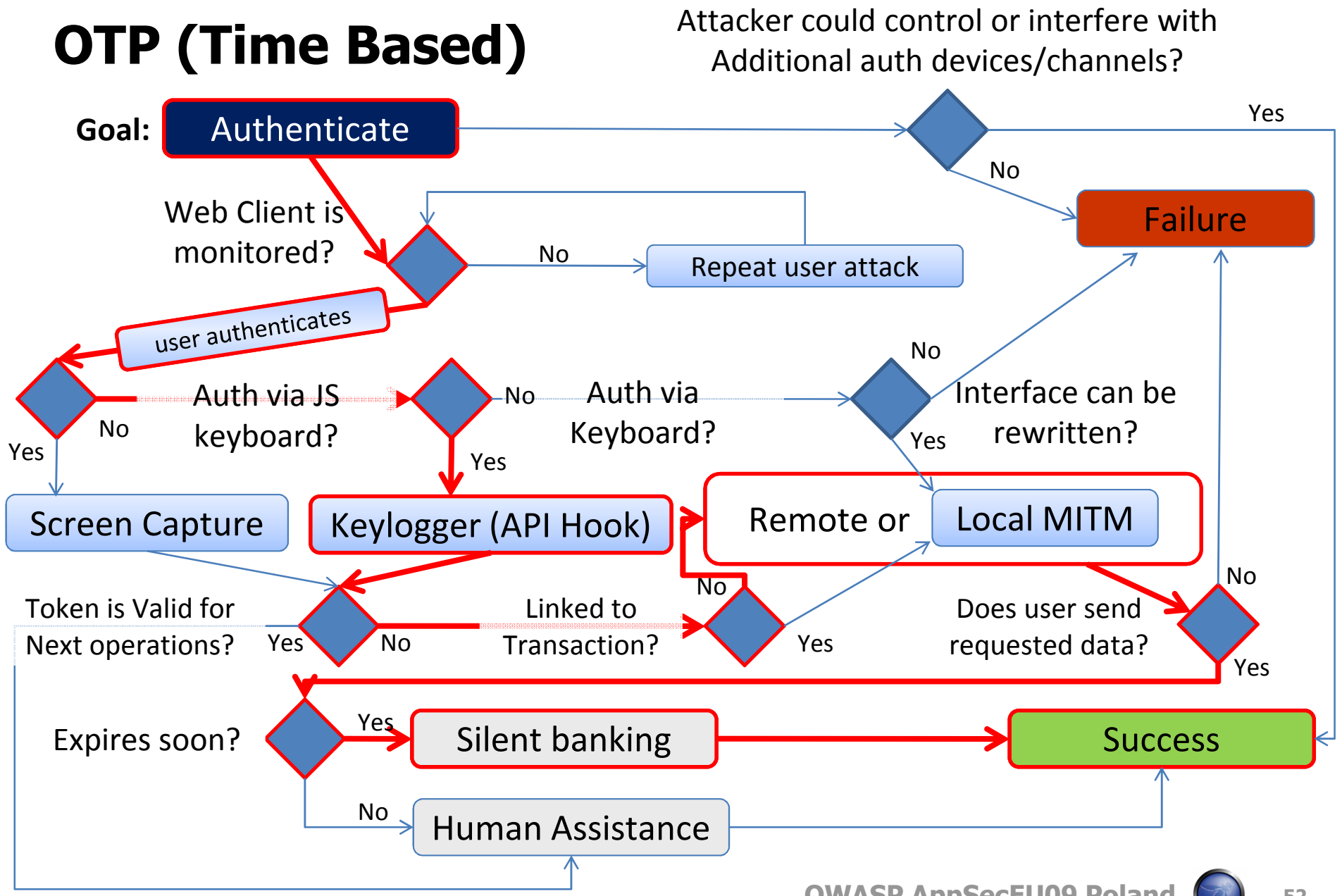


* it's supposed that user is banking from an infected PC or from any other equivalent device

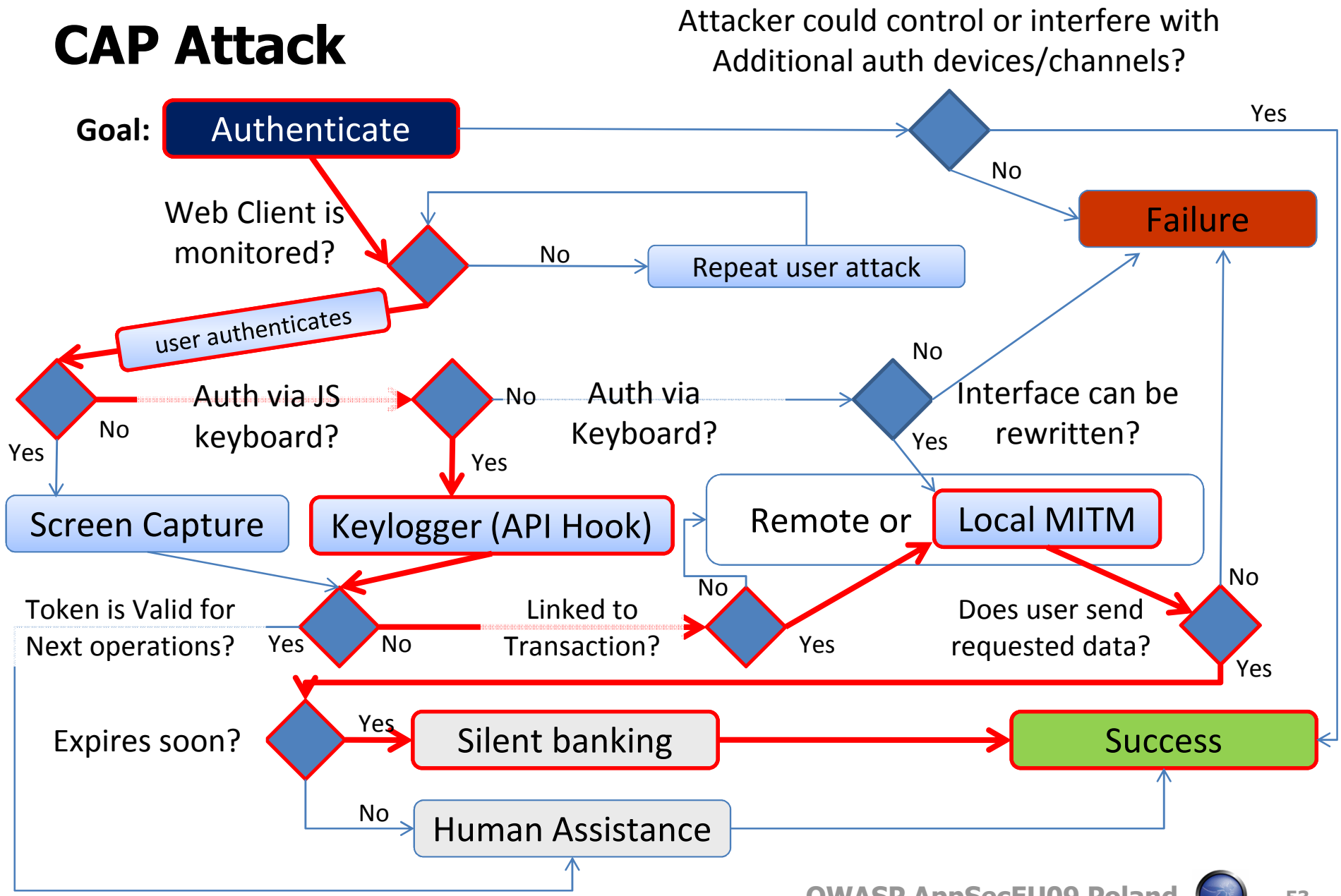
Password


Attacker could control or interfere with
Additional auth devices/channels?
Goal:
Authenticate
Yes
No
Failure
Web Client is
monitored?
No
Repeat user attack
user authenticates
No
Auth via JS
keyboard?
No
Auth via
Keyboard?
No
Interface can be
rewritten?
Yes
Yes
Yes
Screen Capture
Keylogger (API Hook)
Remote or
Local MITM
No
Token is Valid for
Next operations?
Yes
No
Linked to
Transaction?
No
Yes
Does user send
requested data?
No
Yes
Expires soon?
Yes
Silent banking
Success
No
Human Assistance

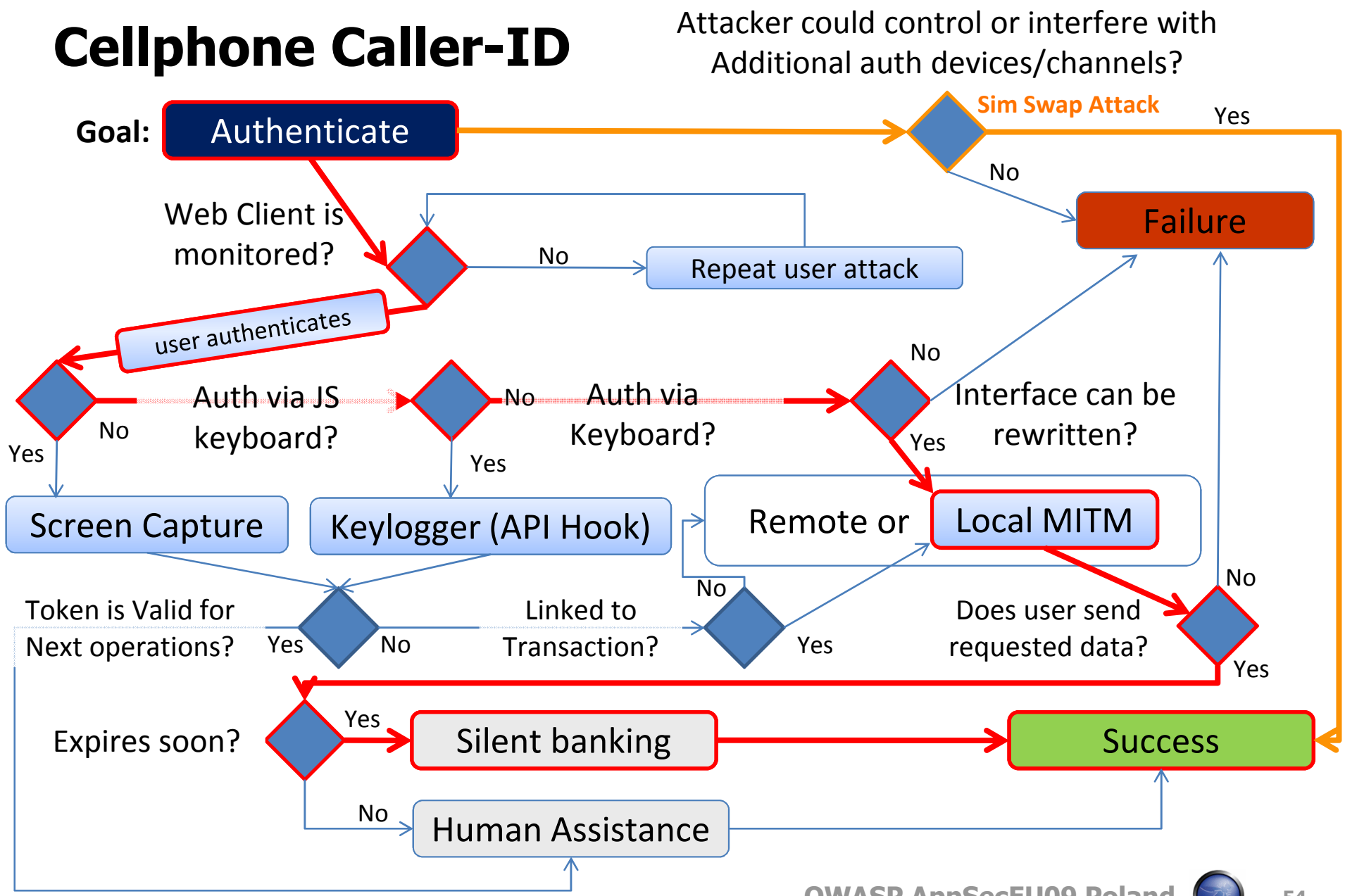
OTP (Time Based)



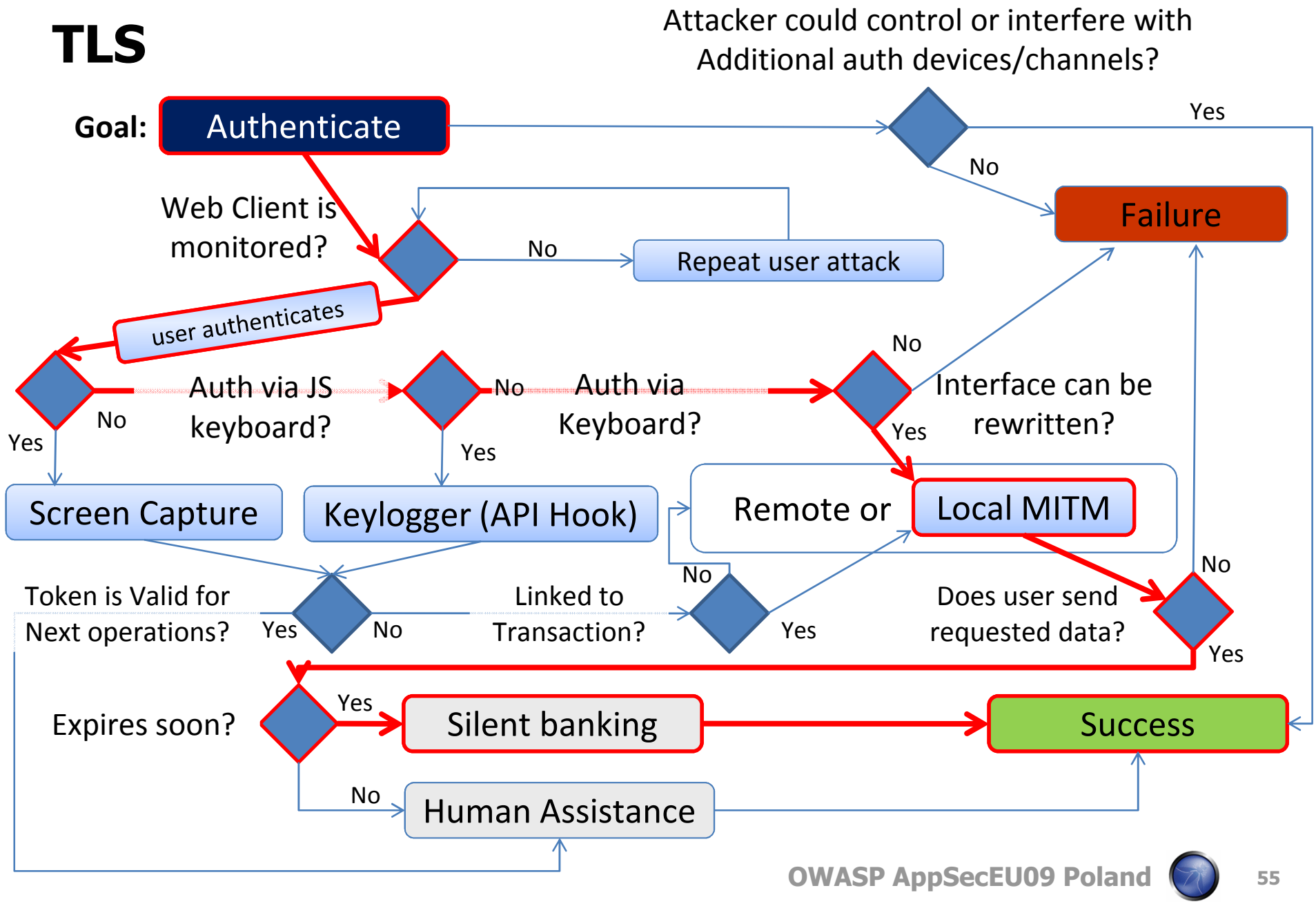
CAP Attack


Attacker could control or interfere with Additional auth devices/channels?
Goal:
Authenticate
Yes
No
Failure
Web Client is monitored?
No
Repeat user attack
user authenticates
No
Auth via JS keyboard?
No
Auth via Keyboard?
No
Interface can be rewritten?
Yes
Yes
Yes
Screen Capture
Keylogger (API Hook)
Remote or
Local MITM
No
Token is Valid for Next operations?
Yes
No
Linked to Transaction?
No
Yes
Does user send requested data?
No
Yes
Expires soon?
Yes
Silent banking
Success
No
Human Assistance

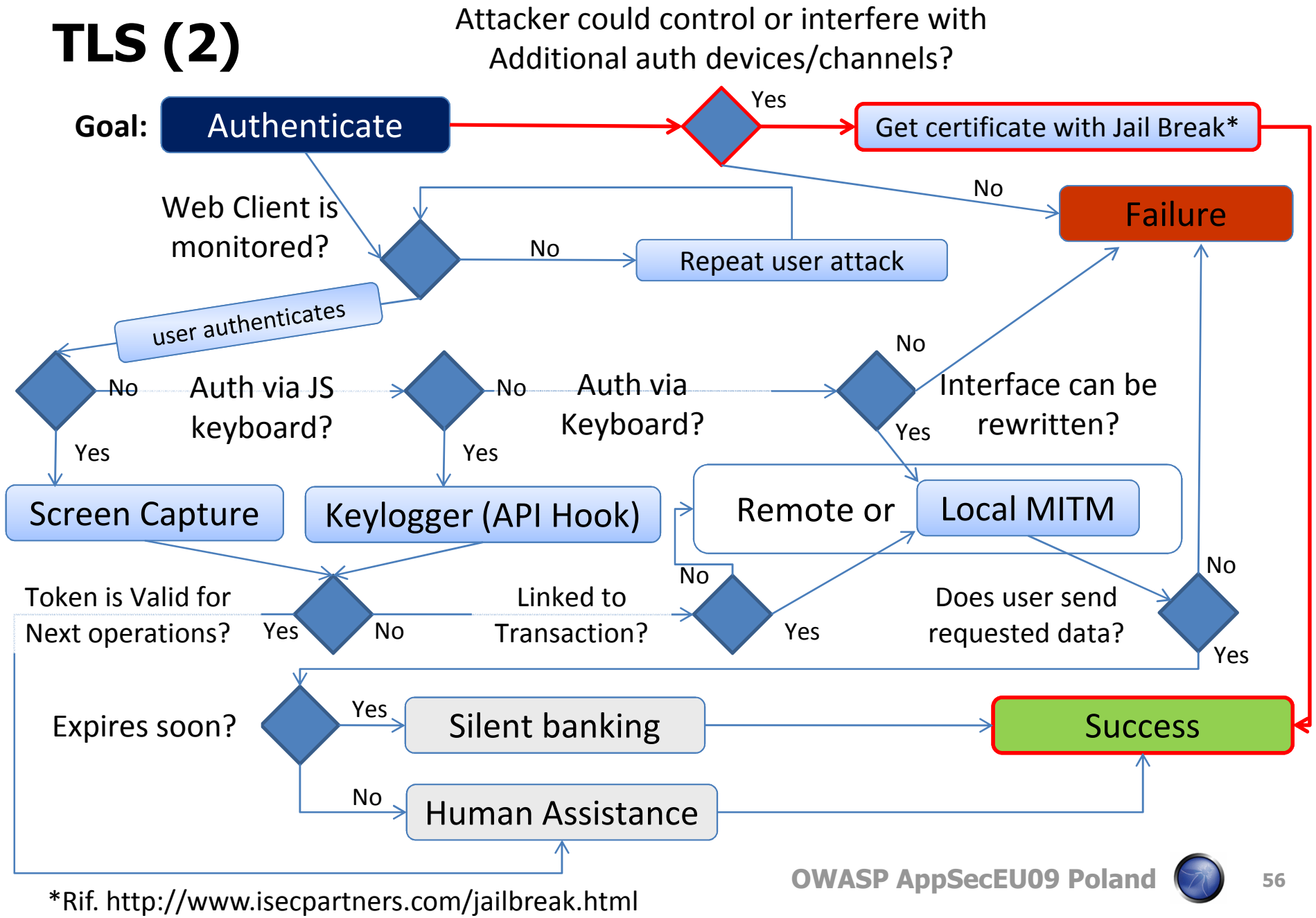
Cellphone Caller-ID



TLS


Attacker could control or interfere with
Additional auth devices/channels?
Goal:
Authenticate
Yes
No
Failure
Web Client is monitored?
No
Repeat user attack
user authenticates
No
Auth via JS keyboard?
No
Auth via Keyboard?
No
Interface can be rewritten?
Yes
Yes
Yes
Screen Capture
Keylogger (API Hook)
Remote or
Local MITM
No
Token is Valid for Next operations?
Yes
No
Linked to Transaction?
No
Yes
Does user send requested data?
No
Yes
Expires soon?
Yes
Silent banking
Success
No
Human Assistance

TLS (2)



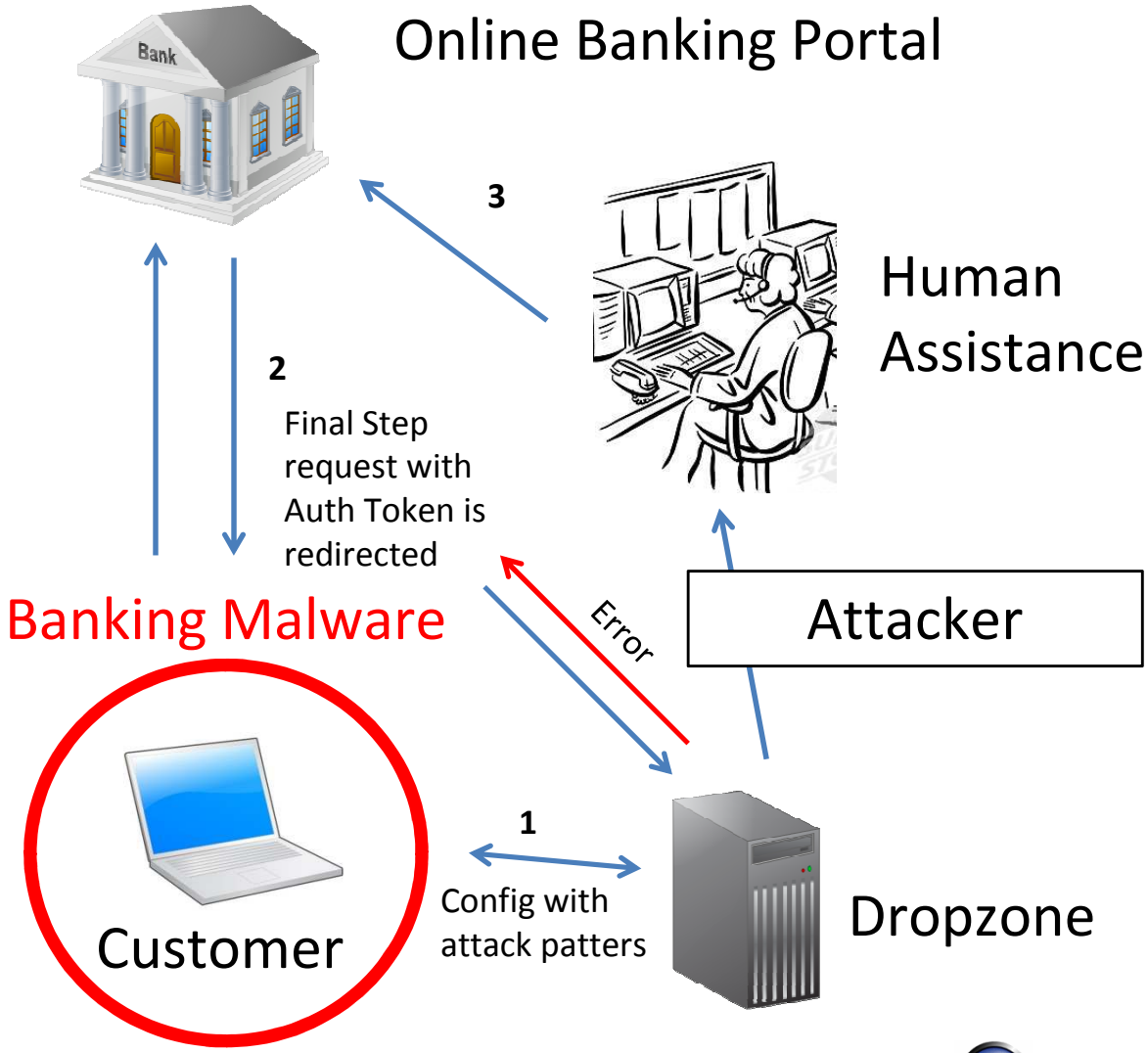
*Rif. http://www.isecpartners.com/jailbreak.html

Remote MITM + Human Assistance

Herein is assumed that the customer is banking from an infected machine.

"*Human Assistance*" is provided by people working for the attacker.

1) Attacker updates the Malware through the Dropzone
2) When Customer performs a transaction the malware re-routes the Token information to the Attackers
3) User is impersonated and transaction is performed

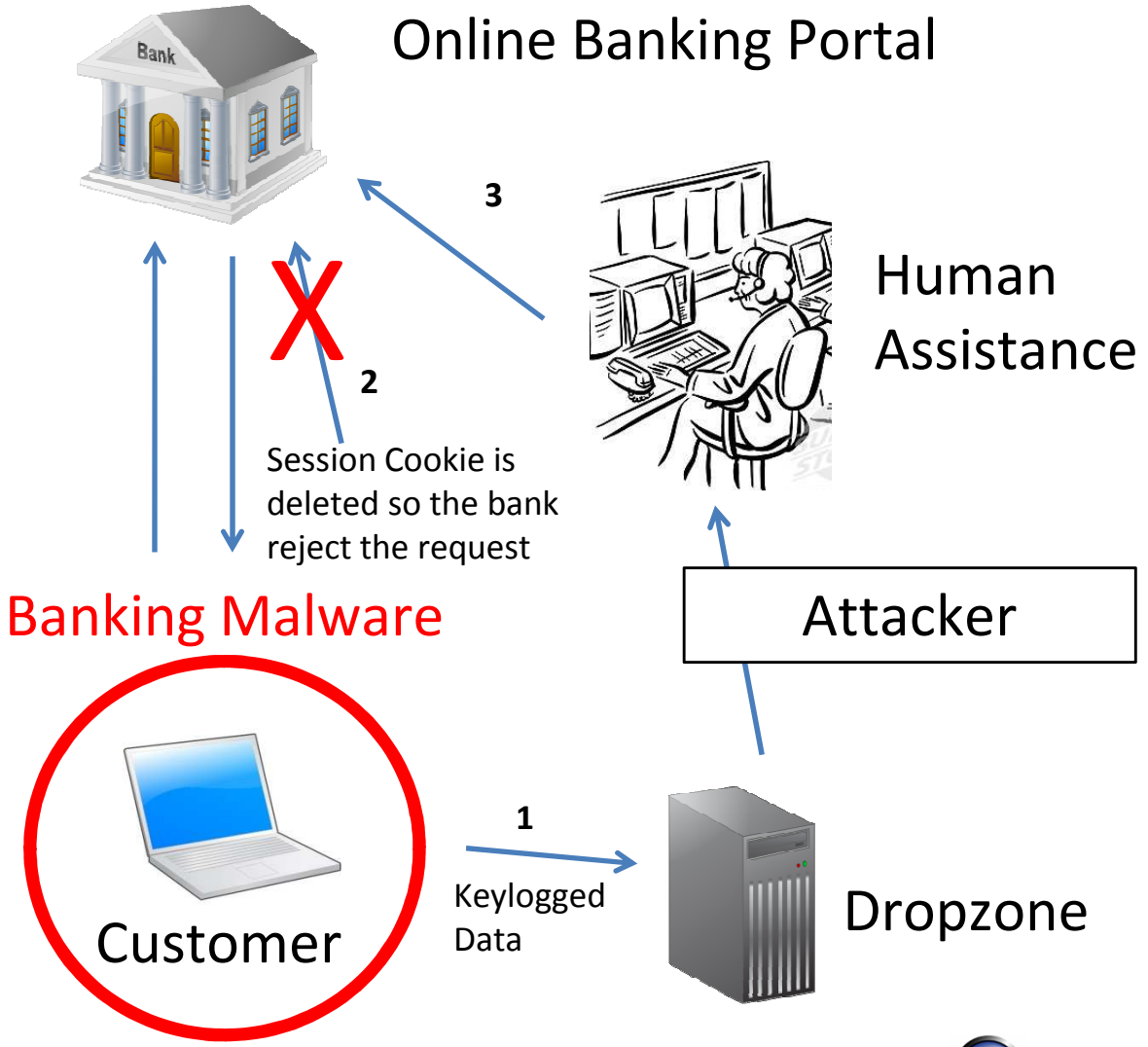


Local MITM + Human Assistance

Herein is assumed that the customer is banking from an infected machine.

"*Human Assistance*" is provided by people working for the attacker.

1) Keylogger data is constantly sent to the dropzone
2) When Customer performs a transaction the malware deletes the cookie
3) User is impersonated using the stolen token stored in the dropzone



Local MITM + Silent Banking

Herein is assumed that the customer is banking from an infected machine.

"*Banking in silence*"* is the ability to perform autonomous transactions.

1) Attacker updates the Malware through the Dropzone, including their bank account number
2) When Customer performs a transaction the malware silently substitutes the details
3) The user authorizes a different transaction from the one desired

Online Banking Portal

3

2

MITM traffic Manipulation lures the user into authorizing a different transaction

Attacker

Banking Malware

Customer

1

Config and updates

Dropzone

*http://www.symantec.com/enterprise/security_response/weblog/2008/01/banking_in_silence.html

Banking provided Security measures

- Password

- TAN (Gridcard, Scratch Card)
  - Transaction Authorization Numbers

- OTP (Time Based, Click
  - One Time password

- CAP (Ran ... allenge Response)
  - Card ... ndom Nonce is like OTP

- SI ... ges

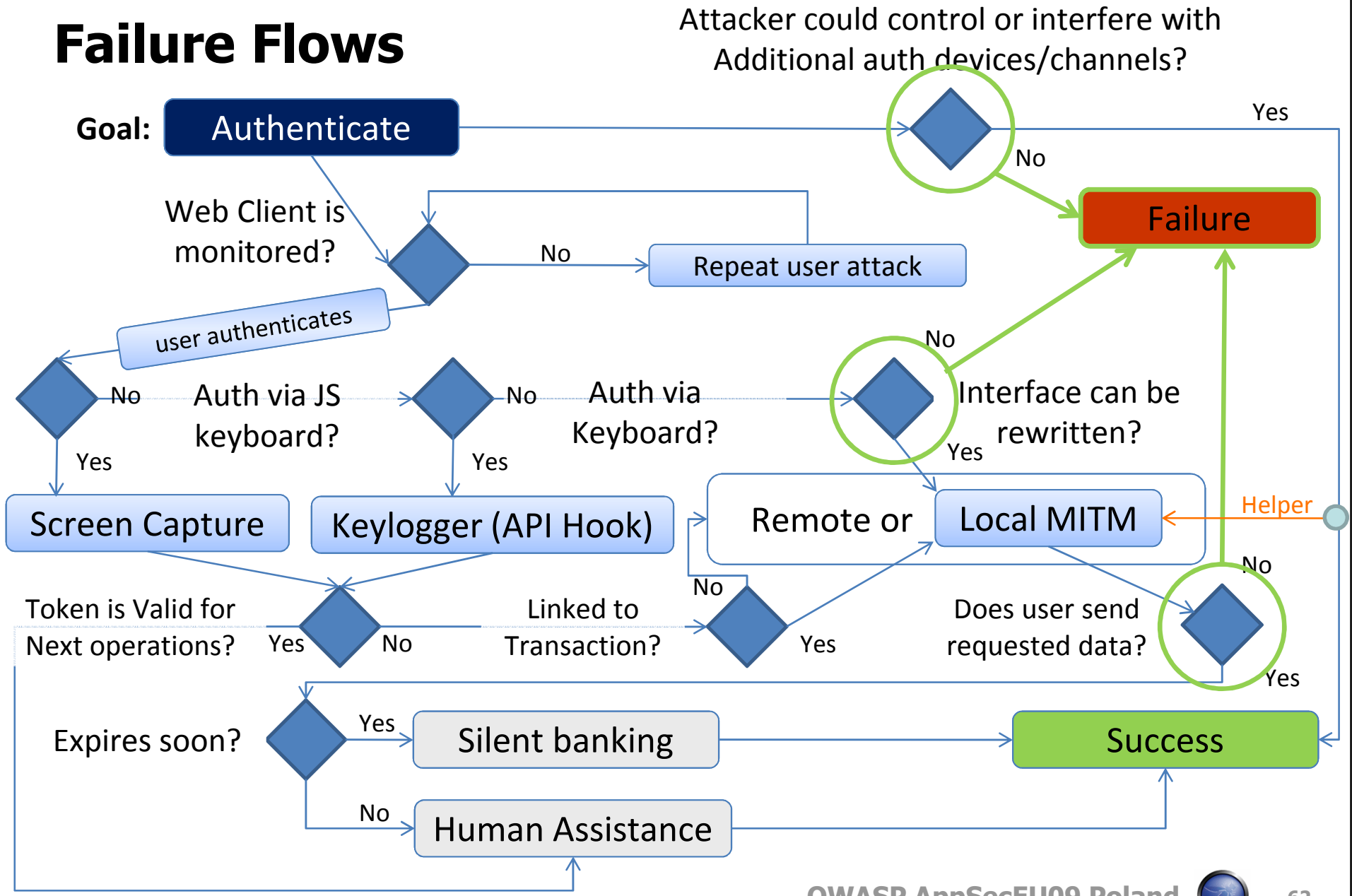
- Cellphone Caller ID

VULNERABLE TO MALWARE ATTACKS

Why Attacker succeeds?

- User can't understand where the money go if the user interface is rewritten
  - Will the user confirm the right transaction?
    - Local MITM can defeat even security solutions based upon Caller-ID confirmation, if the user confirms!
- Most measures are vulnerable to race attacks
  - Who is getting the authentication token first? The Bank or the Attacker?
    - Tokens can be used only once, but needs to reach the bank before the attacker.

Failure Flows



HAT Model for Malware Attacks

- HACK IT
  - The device can be attacked? To which degree?
- ASK HIM
  - Is it possible to ask the user for information? Which information may ask?
- TELL YOU
  - To which degree the user will tell the information required? Is there any barrier?

Antimalware Design Requirements

Priority

- Attacker **should not** control or interfere with Additional auth devices/channels
  - Additional devices must be hard to attack

Very High

- User **should not** tell
  - Authenticate transactions to the user

Very High

- Attacker **should not** ask
  - UI Protection: could imply client-side protections, build completely independent channels or policy restrictions

Medium

Effort estimated as high here

Security Rating

Security Rating

- No-more common malware, instead there is:
  1. Banking Malware with "custom rulesets"
  2. Banking Malware with "no custom rulesets"

1. In the first case all security measures are failing!

2. In the second case:
   - Passwords are very exposed
   - TAN - Gridcards are exposed if tokens are rotated

Rising the bar

- Solution Designed to be malware-resistant
  1. Proprietary solutions
  2. SMS-Challenges with transaction details
  3. CAP with transaction details
  4. Banking Dongle Prototype
- SMS-Challenges with Transaction Details

> Transfer to UK: cc **1293 – Mark Fr**** eur 200 – Token: 3398393883

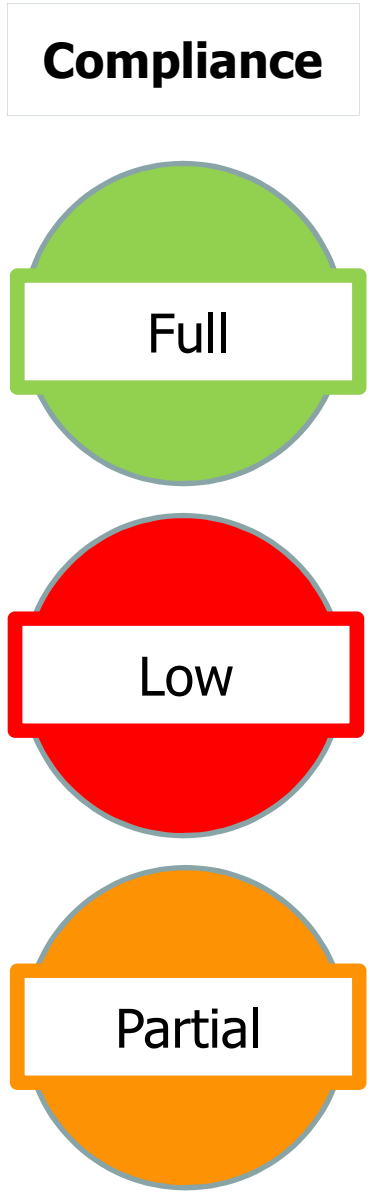
- Visual Banking Dongle

*Rif. A New Approach to Internet Banking – Matthew Johnson

SMS-C. with transaction details

Compliance to our design guidelines

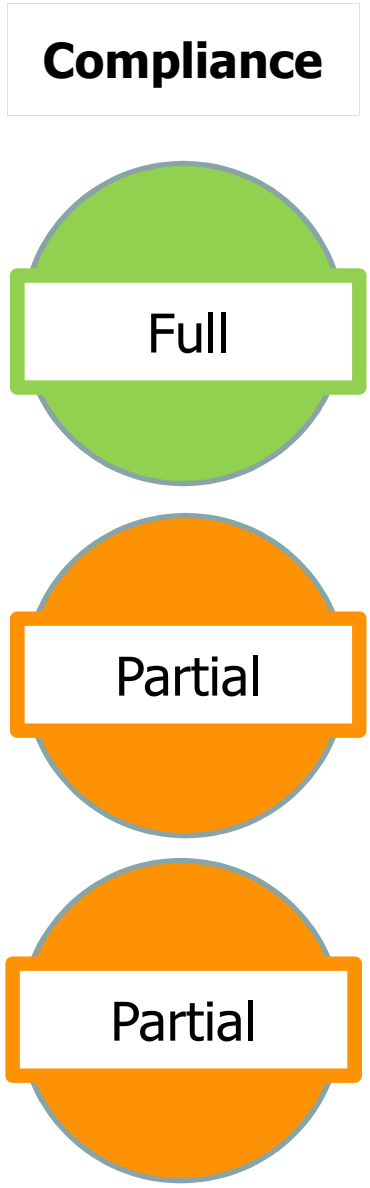
- Authenticate transactions to the user
  - Yes, transaction details are displayed on a separate channel
- Hard to attack
  - Partially: Sim Swap Attacks, OTA messages, Mobile Viruses are a risk
- User Interface Protection
  - Interface is full rewritable. All steps are performed via infected Browser



CAP with transaction details

Compliance to our design guidelines

- Authenticate transactions to the user
  - Yes, transaction details are displayed in a secure manner
- Hard to attack
  - Yes: external device no connection, some vulnerabilities are already known*
- User Interface Protection
  - Partial: communication is not bidirectional, but HTML interface can still be rewritten

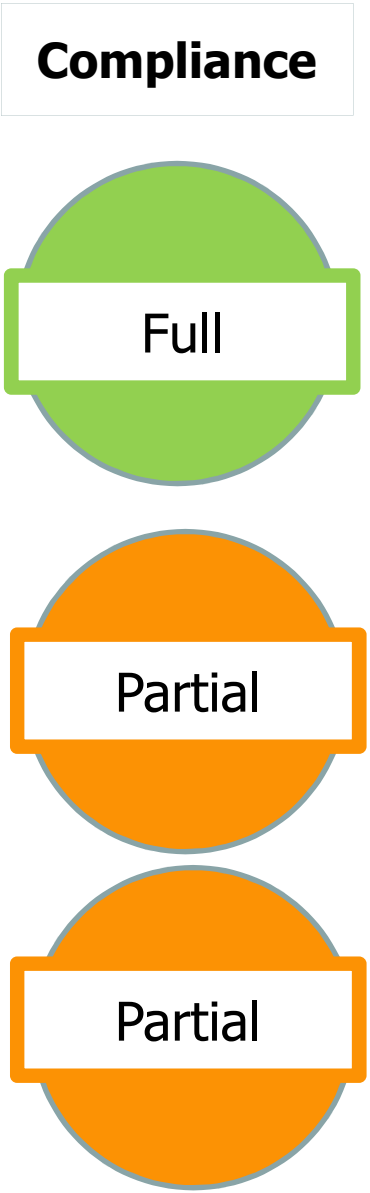


*http://www.cl.cam.ac.uk/~sjm217/papers/fc09optimised.pdf

Visual Banking Dongle

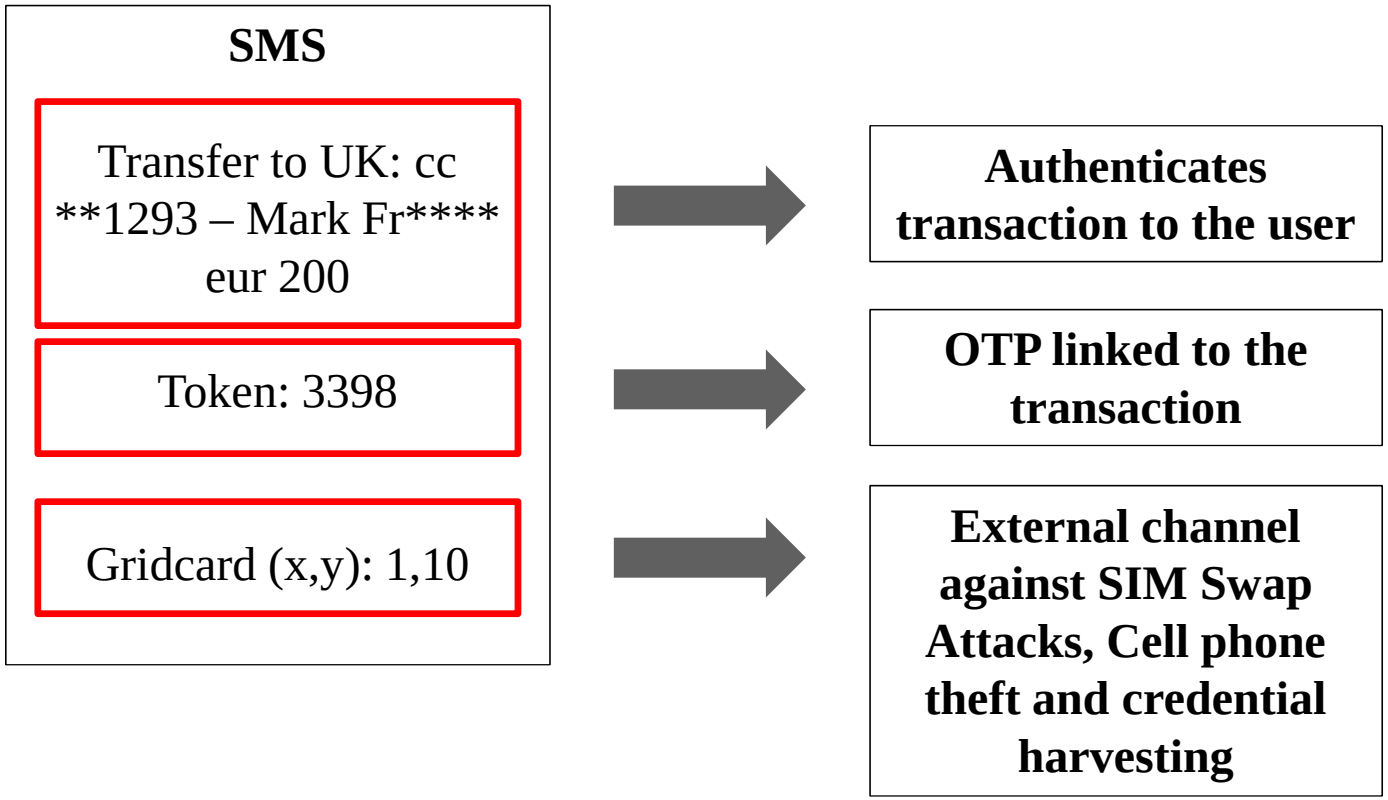
Compliance to our design guidelines

- Authenticate transactions to the user
  - Yes, transaction details are displayed in a secure manner
- Hard to attack
  - Partial (but still a prototype): external device, strong cryptography, open protocol, but exposed to DOS attacks
- User Interface Protection
  - Partial: bidirectional communication on the last step, but HTML interface can still be rewritten



SMS challenge + TD on Gridcard

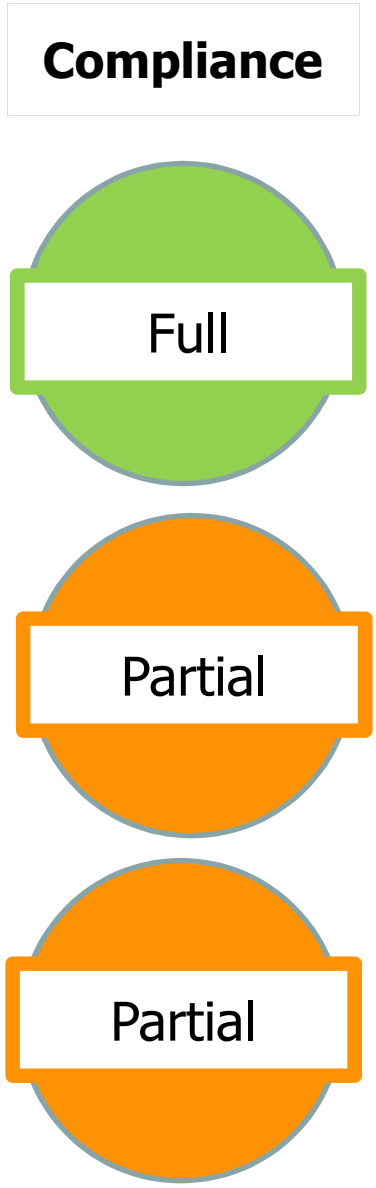
- Add user authentication and rise the security of grid-cards
- Cheaper than OTP and CAP



SMS challenge + TD on Gridcard

Compliance to our design guidelines

- Authenticate transactions to the user
  - Yes, transaction details are displayed in a secure manner
- Hard to attack
  - Achieved control over mobile phone still requires time to gather grid-card values, but not ideal solution for phone banking
- User Interface Protection
  - Partial: interface (browser HTML interface) can still be rewritten



Phishing Protection Dilemma

- Attacks leveraged by the fact that interface can be rewritten could be contained?
  - Answer: Yes. But not only with technology
- Process: Unify Security Measures
  - Old access functionalities downgrade to password
    - Password complexity is not a constrain for keyloggers
  - Downgrade to static secrets is always possible (PCI)
    - So far (May 09) "Secured-by Visa" code prevents only CVV2 from bruteforcing attacks
- Train the user
  - The user will tell his secrets if the bank asks at the right moment

Guerrilla Awareness

- Train the user, with simulated test cases
  - Use the techniques developed by attackers
  - Have a program with different type of attacks
  - Tell the user if he did something wrong
- Users will authenticate to bank honeynet
  - Detailed risk profiling on customer population
    - exposure to basic or advanced user attacks (ex. Flash Codec)
- Note: Users must agree with the program
  - Anyway advertisements and spam are divided by the thin line of consent

Best Practices
Against Banking Malware

Best Practices

- Build on solid bases
  - No Web Security = no need of malware attacks
  - Partial web security = more exposure to malware
- Include partners in the SDLC process
  - You have security, they do not. You do not have security (Ex. Js Malware via Included Tracking Scripts)
- Remove Weakest Links
  - Unify the security measures
    - Exactly know were security measures downgrade
      - Ex. Voice Banking: PIN (static password) and Sister surname.
  - Be sure that possible targets are well protected
    - Ex. 1 Credit Card PAN available at level one is obfuscated
    - Ex. 2 User alerts can be disabled only after 2nd level auth

Best Practices (2)

- Transactions should authenticate to the user
  - The user should be able to discriminate
    - Transaction details announced over a clean channel
    - Geolocation helps (Ex. You are connecting from Rome area)
- Additional channels should be Hard to Attack
    - Ex. Mobile Phones alone are not
      - Sim Swap Attacks (Jpg of Id card can be obtained via Malware)
- Contain impact of User Attacks
    - Train the user by means of attacks and real test cases
      - Lower the likelihood of attacks
    - Enforce Authorization Policies for advanced users
      - Ex. PIN 1: Full Control; PIN 2: Do transfer only to friend list.

Questions

OWASP
EU09 Poland