usenix
THE ADVANCED
COMPUTING SYSTEMS
ASSOCIATION

Demystifying the (In)Security of QR Code-based Login in Real-world Deployments

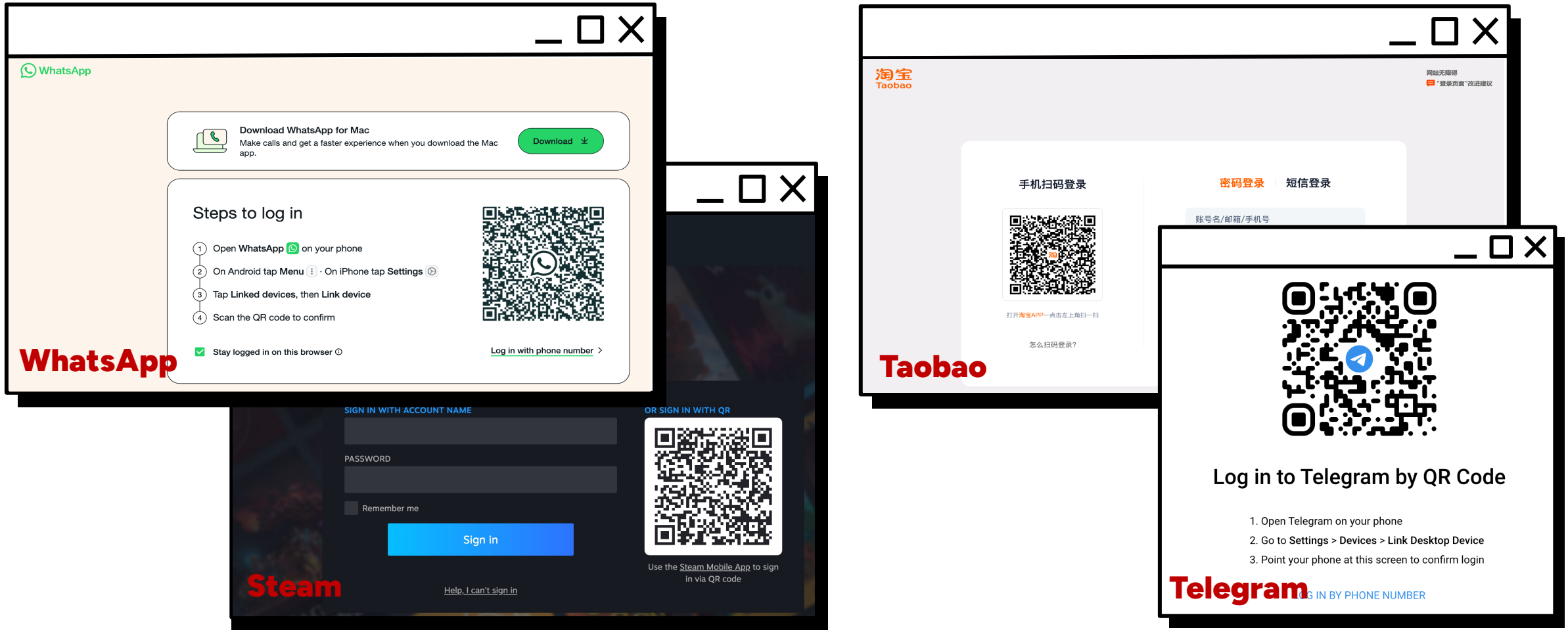
Xin Zhang[1], Xiaohan Zhang[1], Bo Zhao[1], Yuhong Nan[2], Zhichen Liu[1], Jianzhou Chen[1], Huijun Zhou[1], Min Yang[1]

[1]*Fudan University* [2]*Sun Yat-sen University*

復旦大學
FUDAN UNIVERSITY

中山大學
SUN YAT-SEN UNIVERSITY

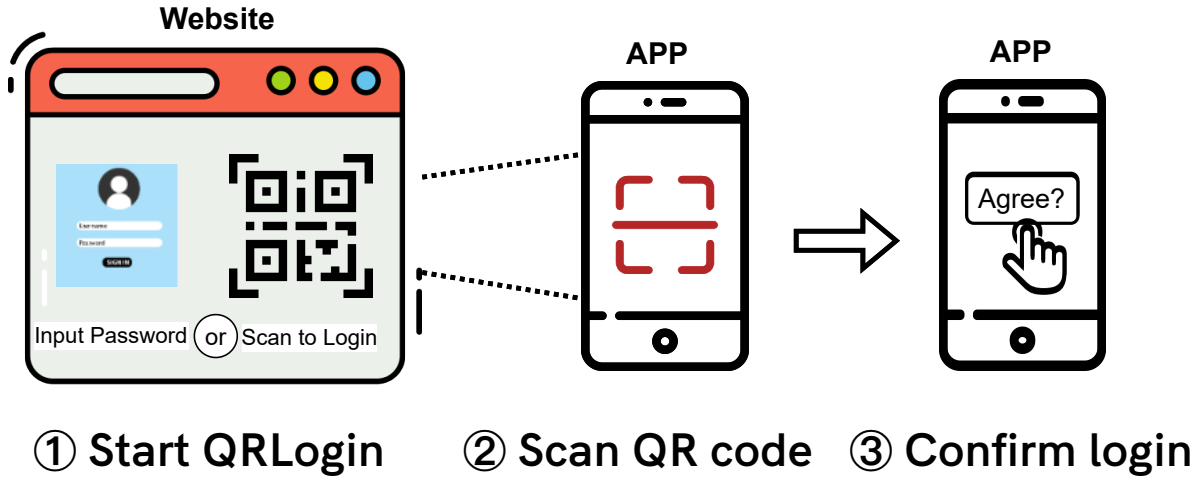
QR Code-based Login (QRLogin)



QRLogin is an emerging authentication scheme

What is QRLogin?

- **QRLogin is a cross-device authentication mechanism**
  - Allows users to log in to a website by scanning a QR code using a trusted mobile app



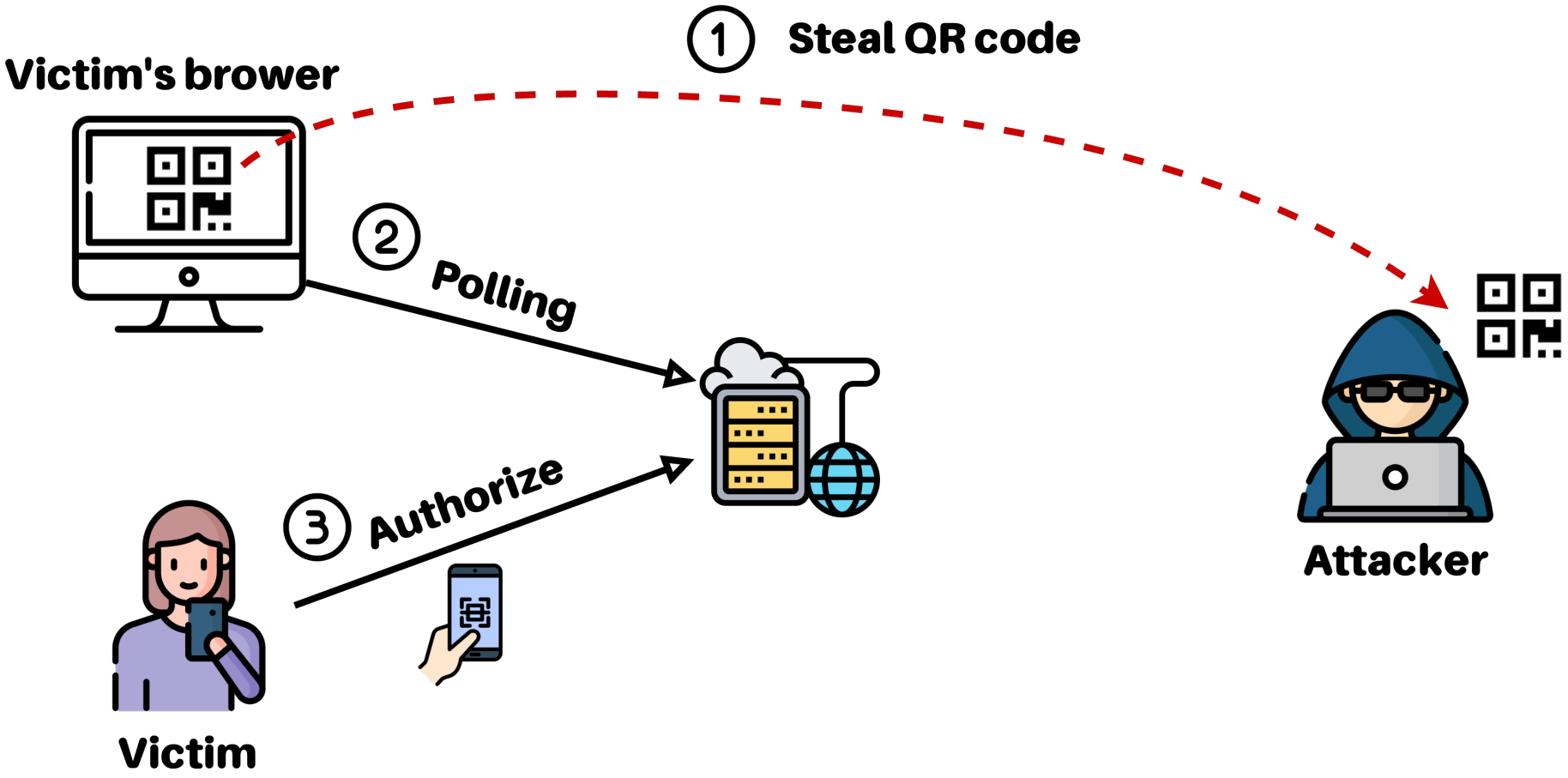
Looks secure, but …

No implementation standard

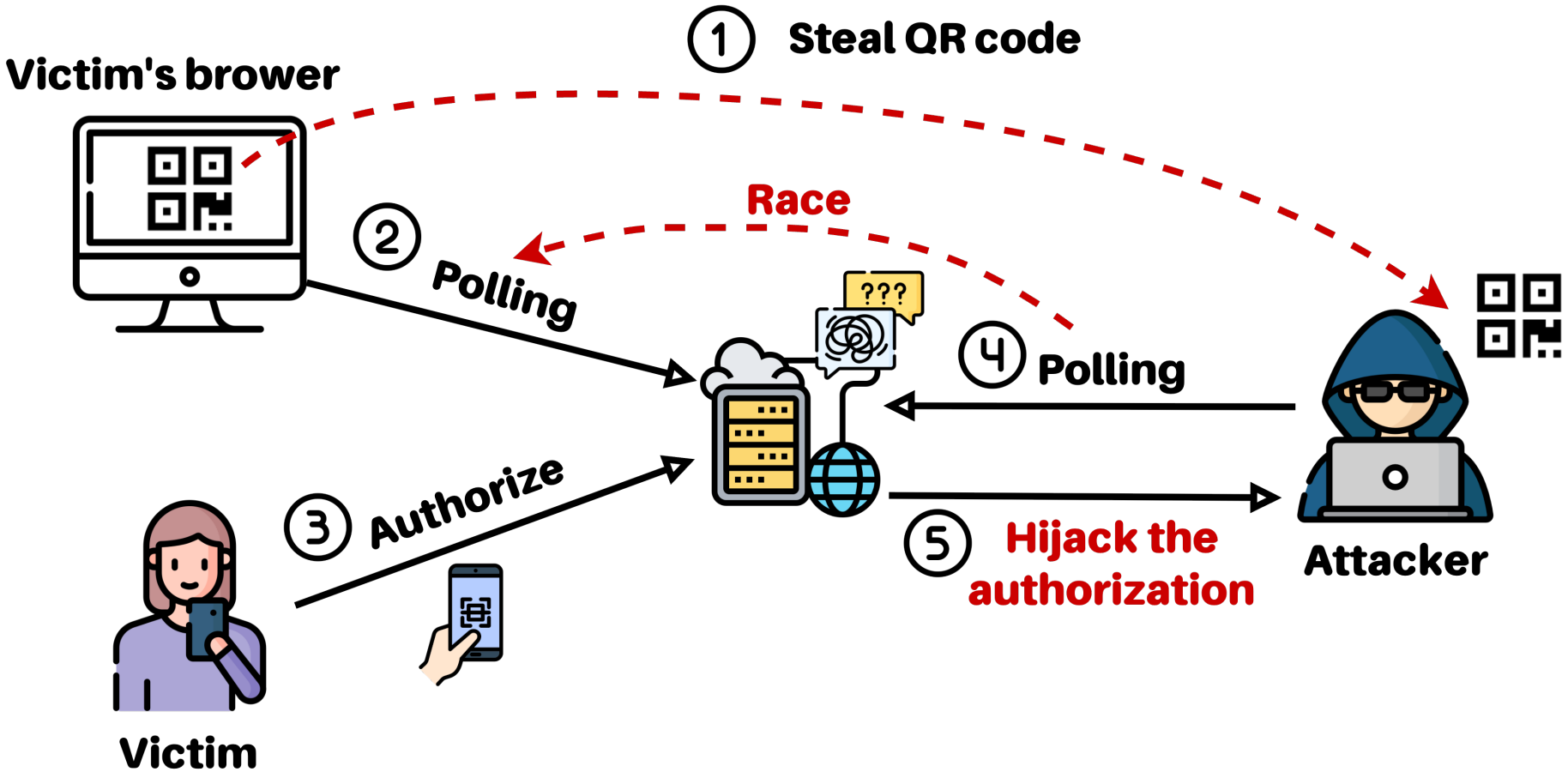
Diverse custom deployments

Varying security levels

Motivating Attack



Motivating Attack



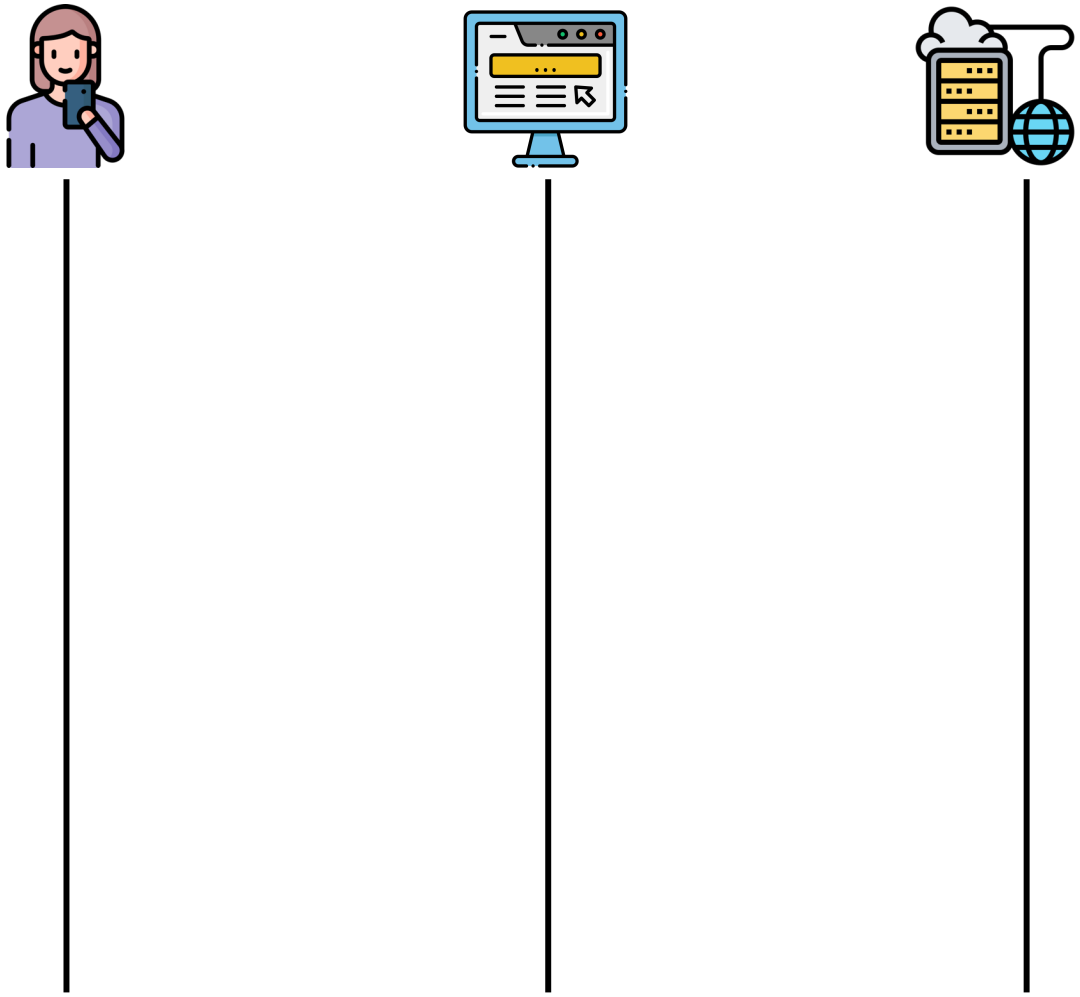
Our Paper

- **The first systematically study on the security of real-world QRLogin**
  - Demystify the workflow of QRLogin
  - Uncover **five new types of attacks** in QRLogin
  - Measure real-world QRLogin deployments
  - Enhance QRLogin security by offering audit tools and suggestions

- **Note: focus on flaws of QRLogin design and implementation**
  - Not considering malicious QR code[1]
  - Not considering attackers control victim's devices

[1] "Qrljacking, an attack introduced on owasp." https:// owasp.org/www-community/attacks/Qrljacking, 2024.

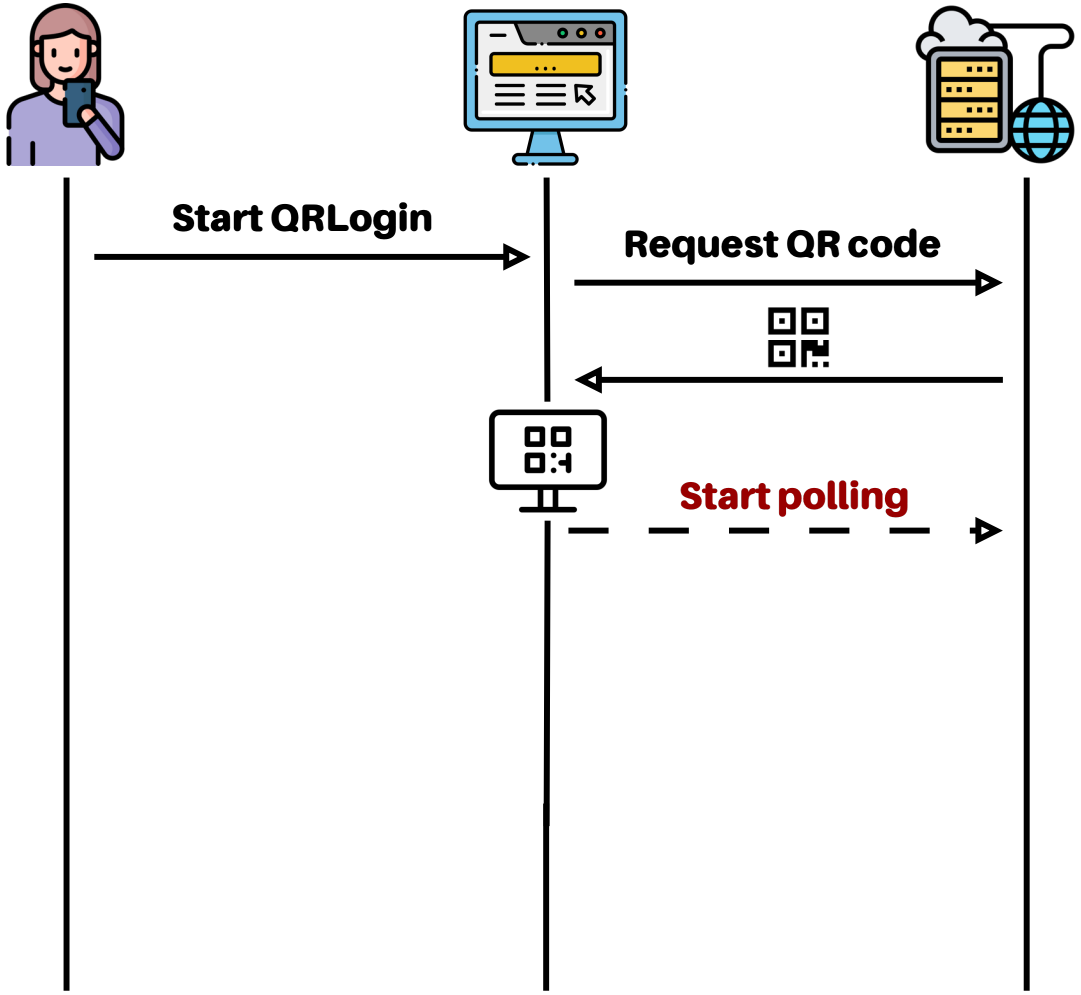
Exploring QRLogin Workflow



Simplified workflow

- QR code generation
- QR code scanning
- Login confirmation

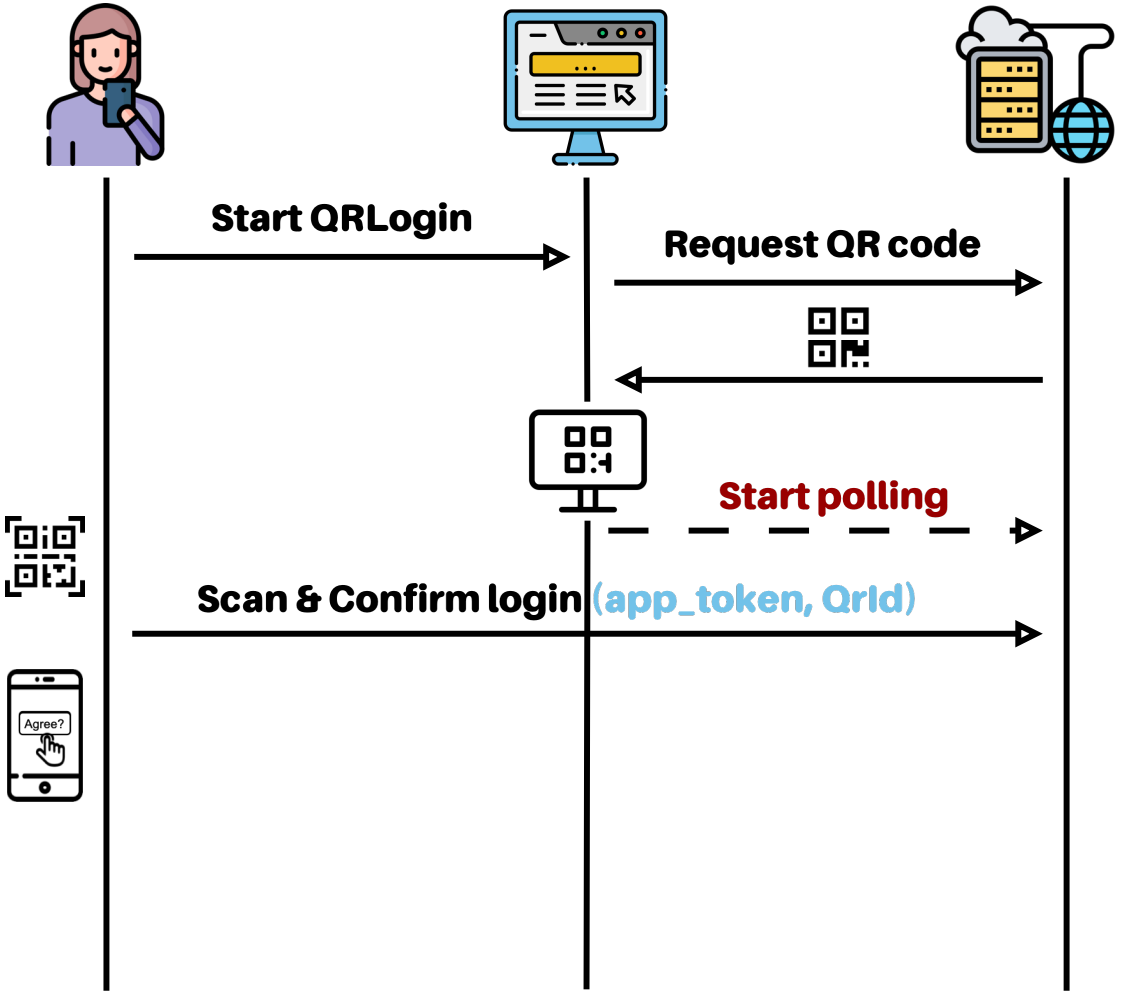
Exploring QRLogin Workflow



Simplified workflow

- QR code generation
- QR code scanning
- Login confirmation

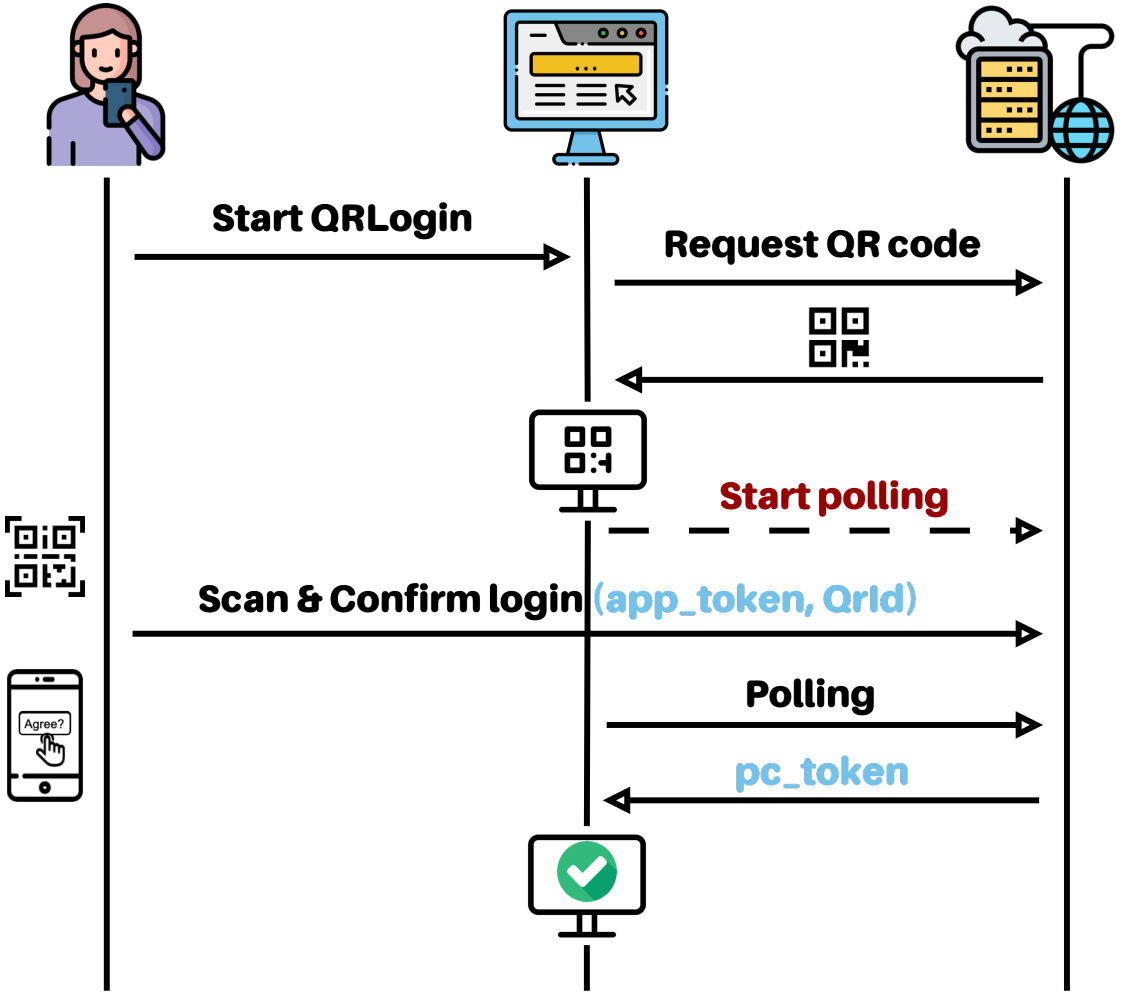
Exploring QRLogin Workflow



Simplified workflow

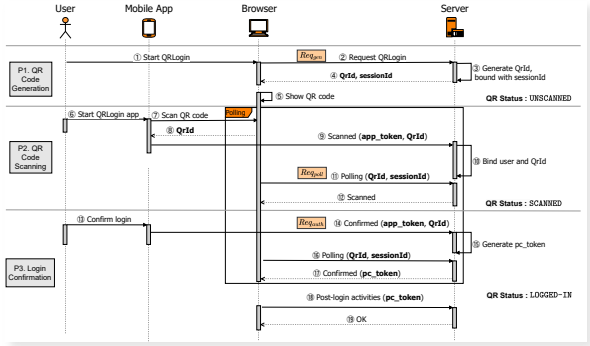
- QR code generation
- QR code scanning
- Login confirmation

Exploring QRLogin Workflow



Simplified workflow

- QR code generation
- QR code scanning
- Login confirmation



More details in paper

Threat Model

- **Attackers only can obtain one of the information:**
  -  The victim’s QR code
  -  The victim’s QR code id
  -  The victim’s identifiers for account login, such as phone number

Threat Model

- **Attackers only can obtain one of the information:**
  - The victim's ***QR code***
  - The victim's QR code id
  - The victim's identifiers for account login

Questionnaire

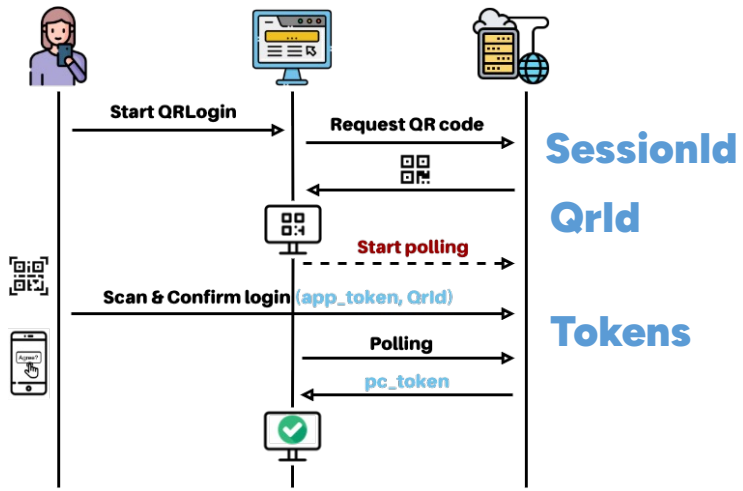
- **Our user study found that:**
  - Many users face the risk of QR code leakage
  - Most unaware of the risks

Principled Security Analysis

Two-step analysis

First, identify **security-critical variables**
Then, check if they meet core **security principles**

- **Security-critical variables**
  - SessionId, QrId, Tokens



- **Security principles**

Confidentiality

Can it be leaked or guessed?

Integrity

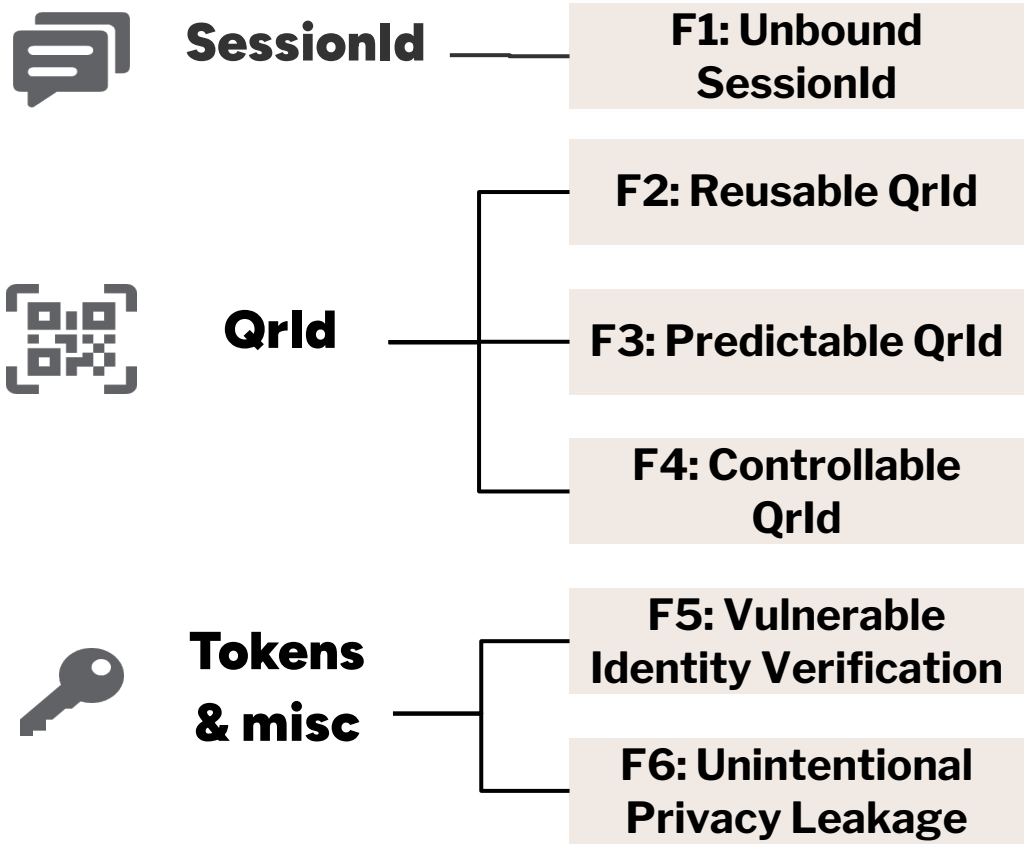
Can it be tampered with?

Consistency

Is it maintained consistently across the workflow?

Discovered Security Flaws & Attacks

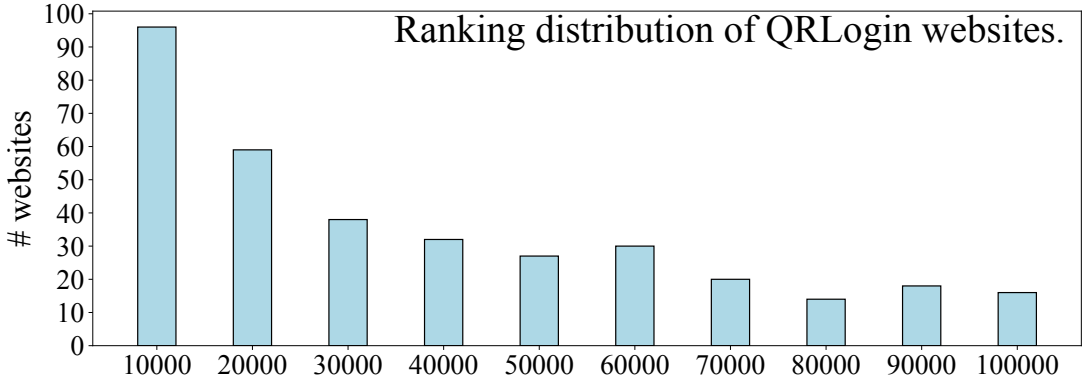
- **This work discovered six flaws in QRLogin leading to five new attacks**

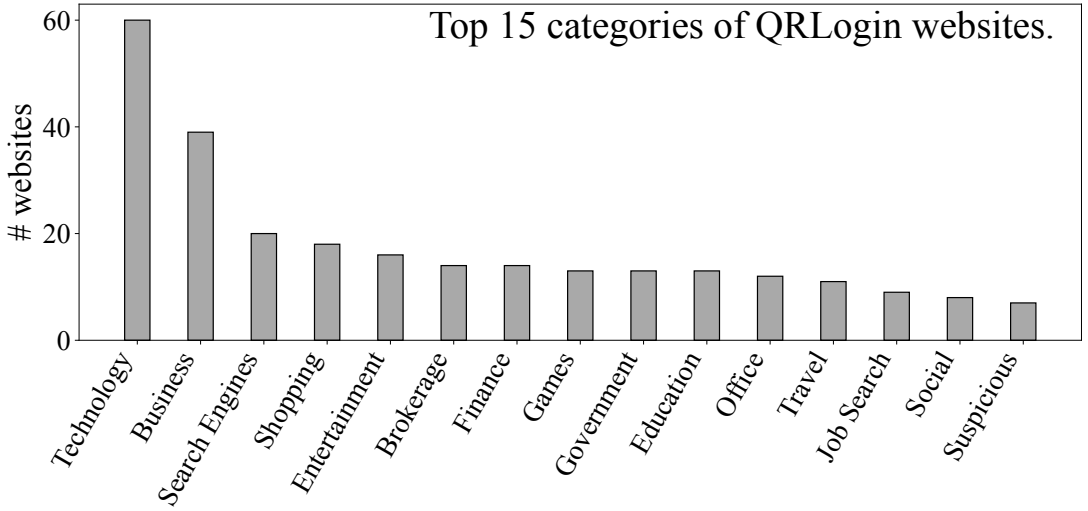


	Attack	Concerned Flaws
01	Authorization Hijacking	F1 | F4
02	Double Login	F1 & F2
03	Brute-force Login	F1 & F3
04	Universal Account Takeover	F5
05	Privacy Abuse	F6*

Measuring Real-world QRLogin Deployments

- **RQ1 – Adoption: How is QRLogin used in the real world?**
  - Collect 350 QRLogin websites from Tranco Top 100K



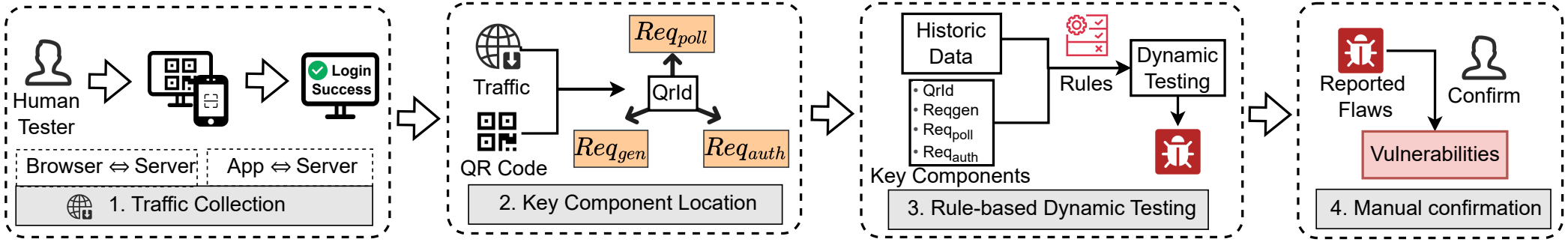


- 55% of them are in the Top 30K
- Top categories are most security-sensitive

QRLogin tends to appear in top-ranked and security-sensitive websites

Measuring Real-world QRLogin Deployments

- **RQ2 – Security: Are deployments actually secure?**
  - Implement a semi-automated pipeline to analyze QRLogin sites



  - Design rules for detecting flaws

Flaw	Premise	Rule Condition
F1. Unbound *sessionId*	$sid_1 \neq sid_2, qrid_1 \neq qrid_2, Req(sid_1, qrid_1), Req(sid_2, qrid_2), \neg Authed(qrid_1), \neg Authed(qrid_2)$	$Resp(sid_2, qrid_1) = \texttt{UNSCANNED}$
F2. Reusable *QR code*	$sid_1 \neq sid_2, qrid_1 \neq qrid_2, Req(sid_1, qrid_1), Req(sid_2, qrid_2), Authed(qrid_1), \neg Authed(qrid_2)$	$Resp(sid_2, qrid_1) = \texttt{LOGGED-IN}$
F3. Predictable *QrId*	$Req(sid, qrid)$	$Len(qrid) \leq 6 \wedge IsDigits(qrid)$
F4. Controllable *QrId*	$Req(sid, qrid)$	$qrid \in GetFields(Req_{gen})$
F5. Vulnerable Identity Verification	$A = GetFields(Req_{auth})$	$\exists a \in A, a \in PII$
F6. Unintentional Privacy Leakage	$\exists resp, A = GetFields(resp)$	$\exists a \in A, a \in PII$

Measuring Real-world QRLogin Deployments

- **RQ2 – Security: Are deployments actually secure?**

Dataset

- Analyzed 350 QRLogin sites (Top 100K)
- Found 181 unique implementations
- 109 are testable

Attack	# Websites
Authorization Hijacking	37
Double Login	17
Brute-force Login	1
Universal Account Takeover	2
Privacy Abuse	7

43% of websites have flaws!

- **Responsible disclosure**

Disclosure actions

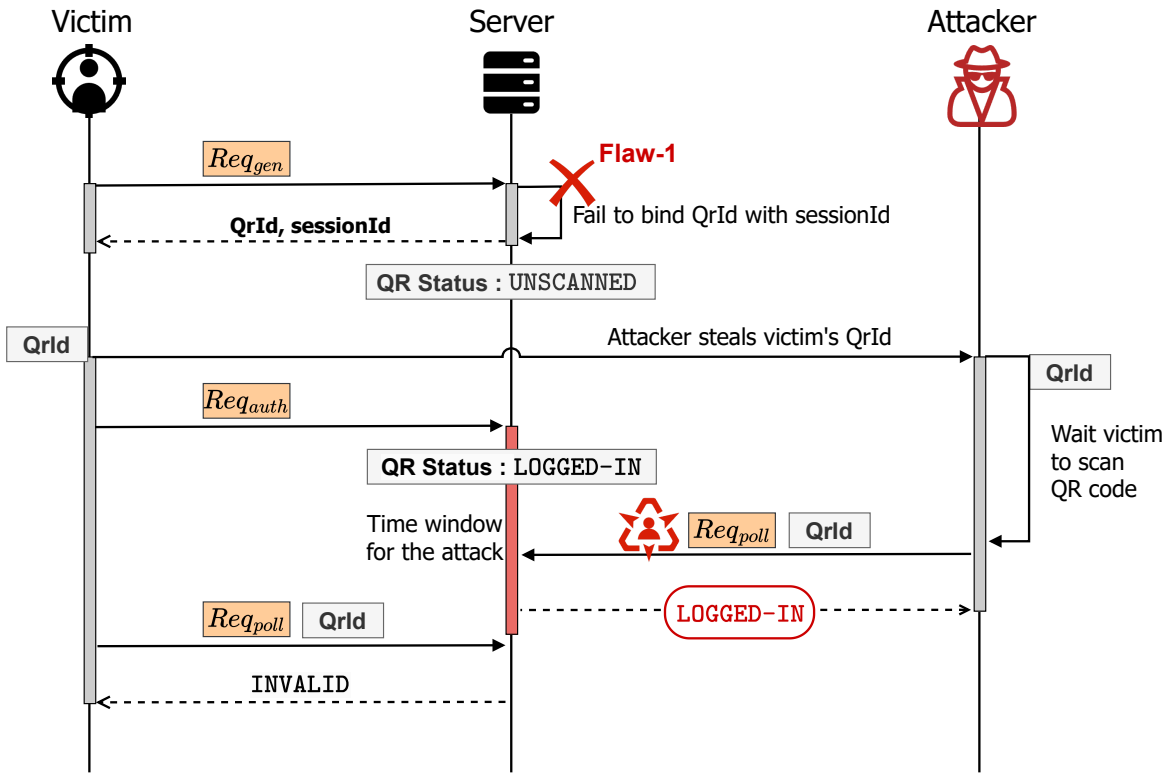
- Report all vulnerabilities to vendors & vulnerability platform (CNVD/CAPPVD)
- Provide mitigation suggestions

Impact

- 17 CNVD IDs
- 25 NVDB IDs

Typical Attacks

Attack-1: Authorization Hijacking



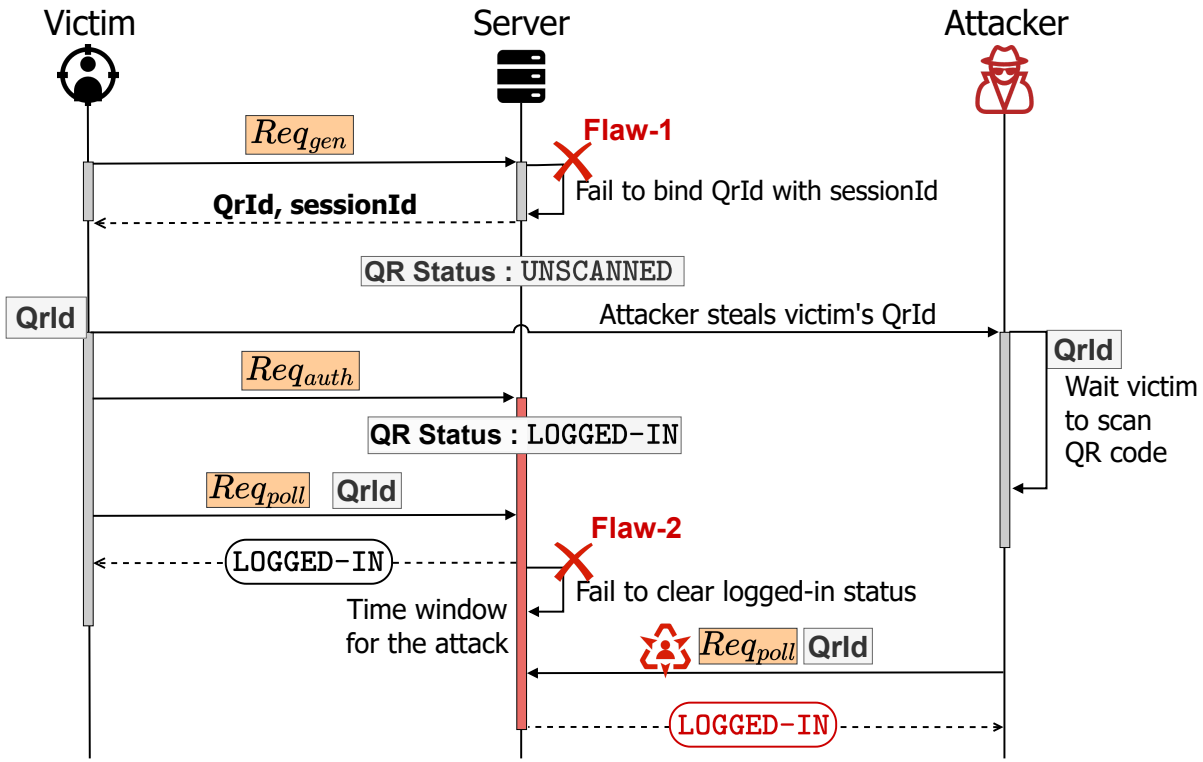
Flaw-1: unbound sessionId

- **Attack Process**
  - Attacker steals victim's QrId
  - Victim scan & confirm login
  - Attacker races poll → hijacks authorization

- **Case-1: A leading e-commerce platform**
  - Nearly 500 million users
  - Account takeover
    - Purchase & browsing history
    - Phone number & shipping address
    - …
  - NVDB-CAPPVD-2024143978

Typical Attacks

Attack-2: Double Login

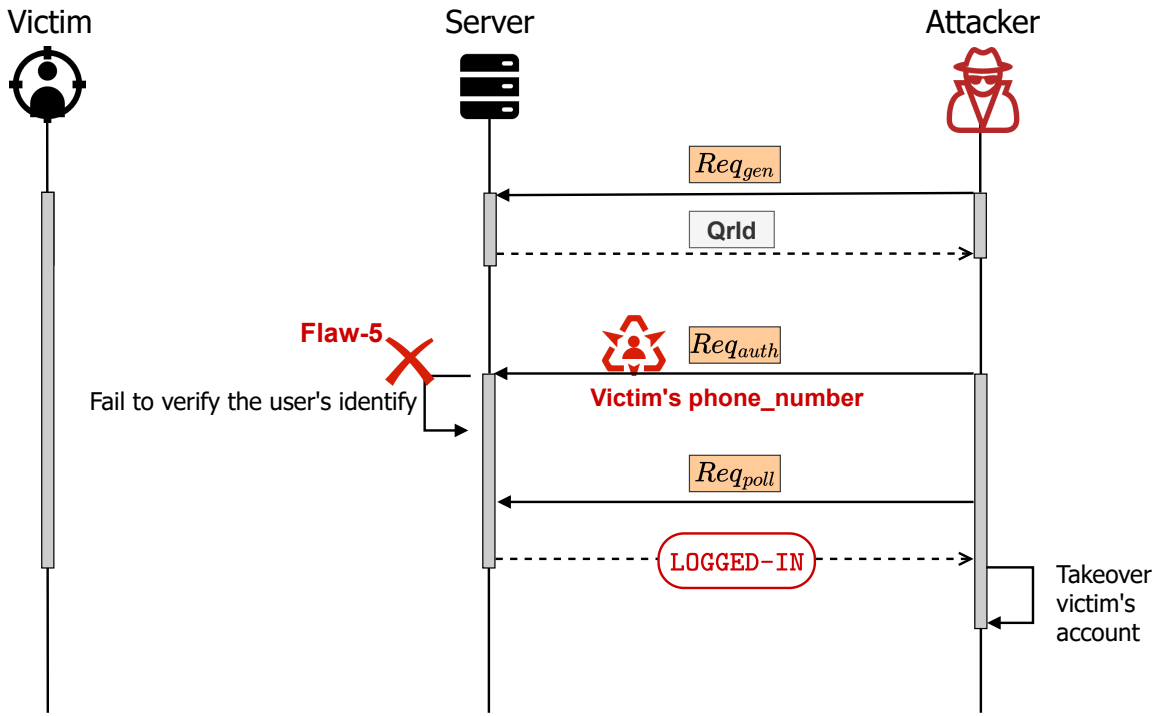


Flaw-1: unbound sessionId

Flaw-2: reusable QR code

- **Attack Process**
  - Attacker steals victim's QrId
  - Victim logs in
  - Attacker reuses victim's QrId (still valid)
  - Attacker silently logs in alongside victim
- **Case-2: A popular social media website**
  - Within top 500 websites
  - Account takeover
    - Monitor victim
    - Perform malicious actions
    - …

Attack-4: Universal Account Takeover



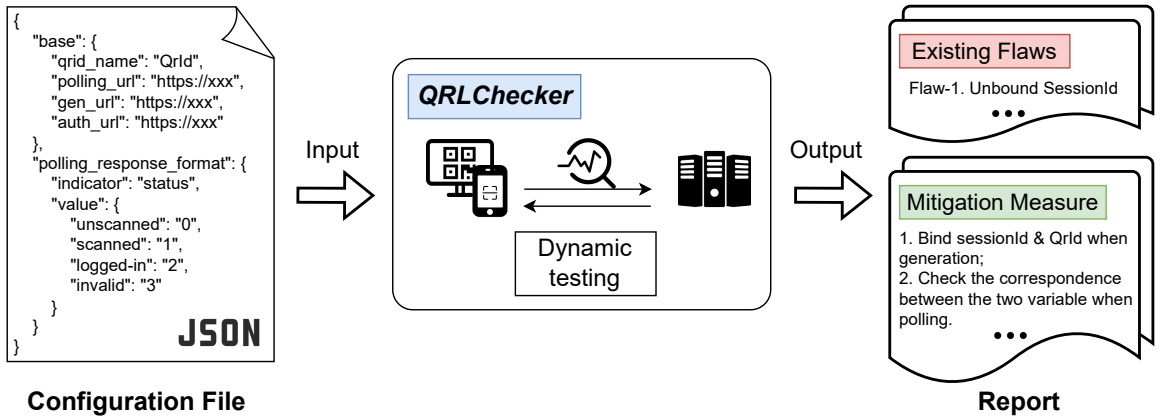
- **Attack Process**
  - Attacker knows victim's account id like phone number
  - Attacker impersonates victim to authorize
  - Server fails to check
  - Attacker is logged in as the victim
- **Case-4: A popular cloud storage service**
  - Over 100 millions users
  - Account takeover
    - Illegally access victim's private files
    - …

Flaw-5: vulnerable identity verification

Measuring Real-world QRLogin Deployments

- **RQ3 – Mitigation: How can we make it safer?**

- **Provide auditing tools for developers**
  - QRLChecker



Scan to request tool access

https://zenodo.org/records/14676842

- **Provide security suggestions**

For developers

- Server-side QrId generation
- Bind QrId + session
- Token verification
- Avoid privacy exposure

For users

- Protecting QR codes actively
- Raising security awareness

Thanks!

復旦大學
FUDAN UNIVERSITY

中山大學
SUN YAT-SEN UNIVERSITY

Xin Zhang

Fudan University

zhangx22@m.fudan.edu.cn