Lost & Found

The Hidden Risks of Account Recovery in a Passwordless Future

Speakers: Sid Rao, Gabriela Sonkeri

Blackhat USA 2025

August 7, Thursday

Note: This handout version of the slide deck has slightly different (and more) content than the presentation version

Who are we?

Dr. Sid Rao

Senior Security Researcher

Nokia Bell Labs
Finland

Gabriela Sonkeri *

Security Engineer

Wolt
Finland

Amel Bourdoucen *

User and Impact Researcher

F-Secure, Aalto University
Finland

Prof. Janne Lindqvist

Associate Professor

Aalto University
Finland

*** Contributions while working at Nokia Bell Labs**

Special thanks: Prof. Tuomas Aura, Dr. Thanh Bui, and Dr. Markku Antikainen

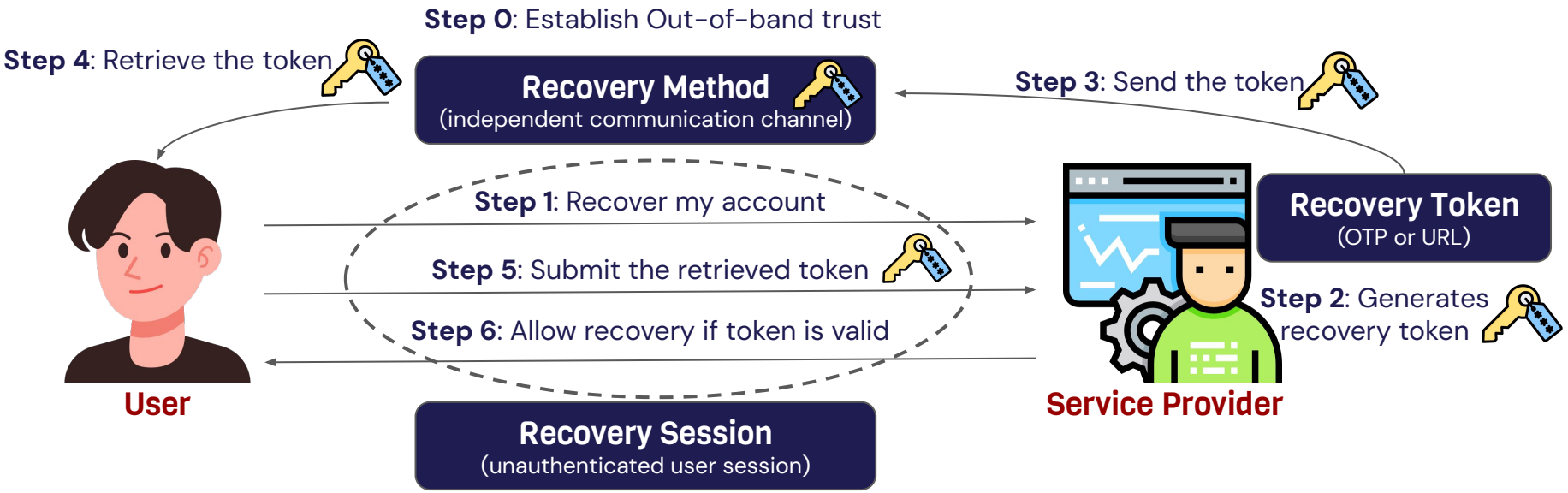
Background

User's authentication credentials become unavailable

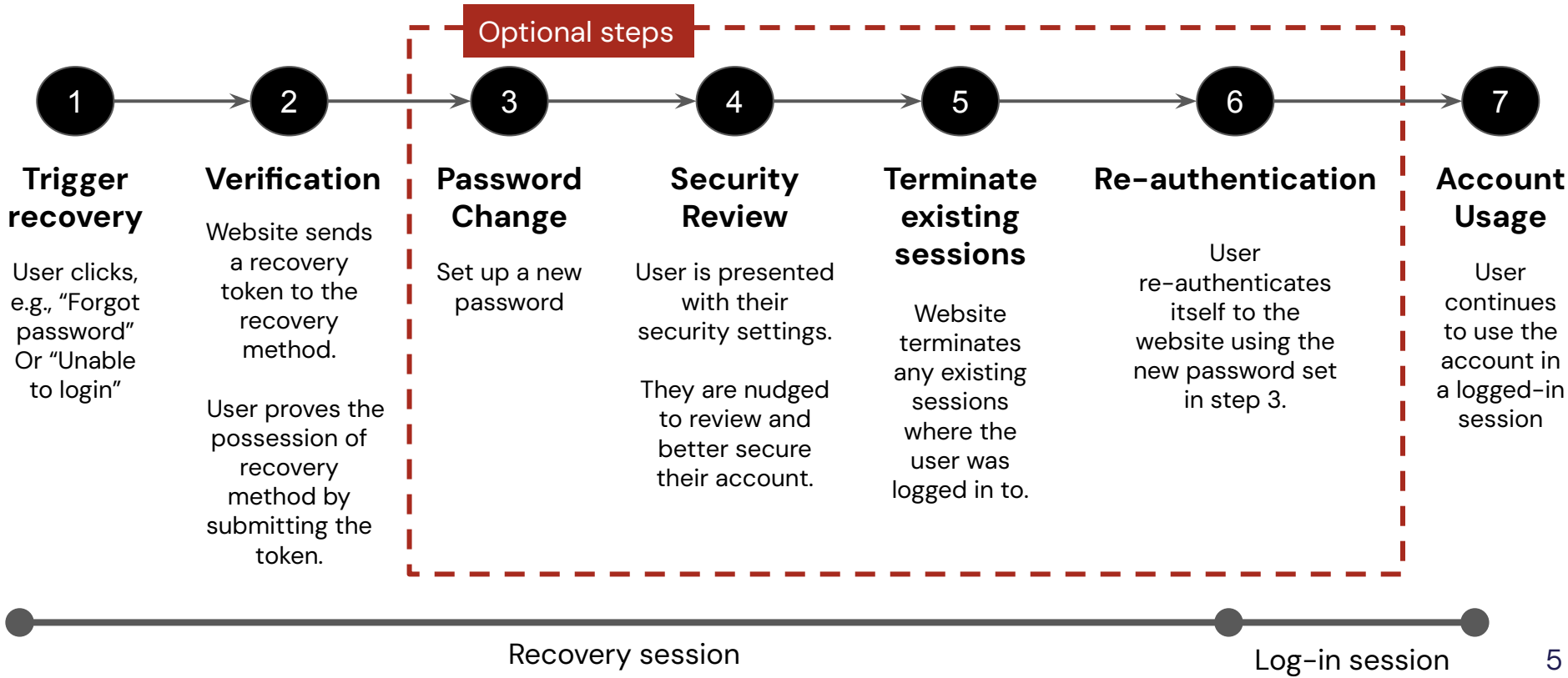
- ***# 1***: Authentication credentials are **forgotten** or **mislaid** by the user
- ***# 2***: Authentication credentials are **inaccessible** to the user
  - Personal device is lost
  - Logging in from a new device or location

Account Recovery Overview

An automated process provisioned by the service provider for benign users to reclaim access



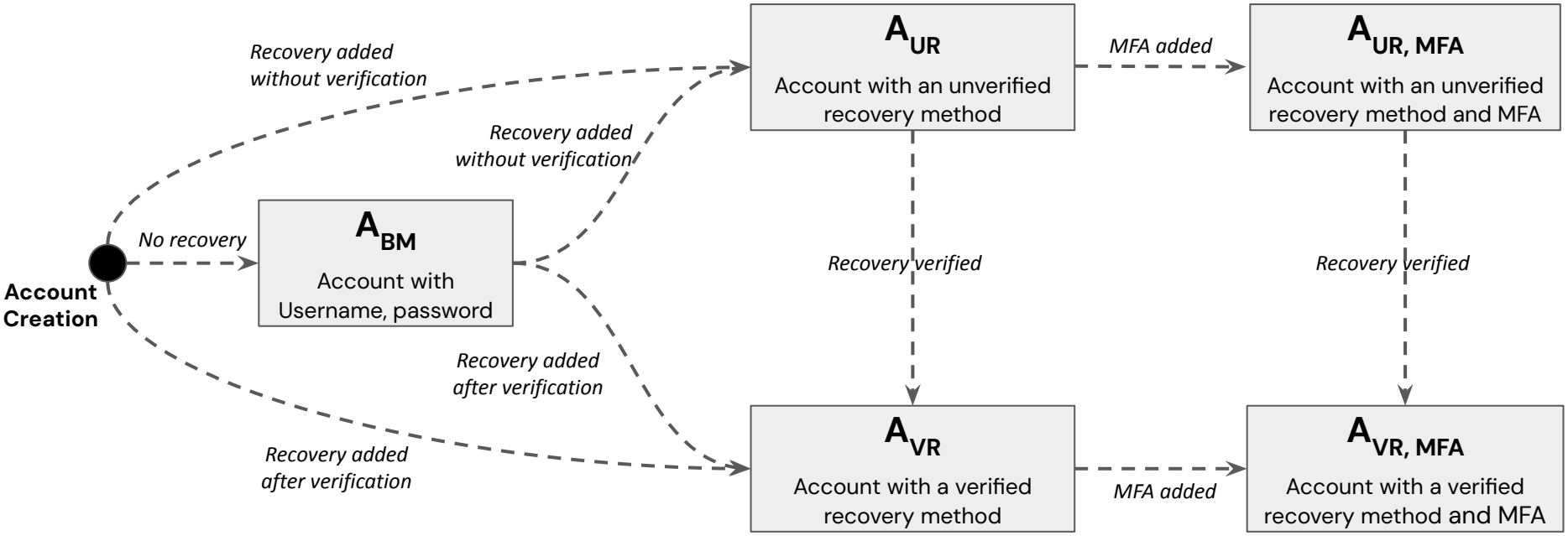
Account Recovery Lifecycle



Motivation

- **Account recovery is a very common user action**
  - 4 out of 5 users have forgotten at least one credential within the last 90 days
  - 25% experiencing the need for account recovery on a daily basis
- **Account recovery is insecure by design**
  - Recovery channels are not under direct control
    - Not possible to know whether the channels are compromised
  - Difficulties of distinguishing between benign users and adversaries
    - Cannot verify the authenticity of the recovery requests
- **Account recovery has not changed or won't change much**
  - Authentication methods have evolved
    - Passwords → Passphrases → Fingerprints → Face ID → Passkeys
  - Recovery relies on legacy methods of SMS and email-based channels
    - Adversaries can bypass strong authentication by exploiting weak recovery

Account states



Out of Scope

Account Hijacking

- Adversary compromises user accounts, e.g., via
  - Leaked credential dumps
  - Password brute force attacks
  - Phishing, Spear Phishing, Whaling
- Recovery channels remain intact during the compromise
  - But, the adversary may want to change them soon to kick out the user completely

We do not attempt account hijacking

Account Remediation

- A special case of account recovery
- Service provider assists a benign user recover its hijacked account
- Involves human intervention
- Requires verification of the affected user's real-life identity

We do not exploit account remediation

We perform **"Account takeover",** a lateral compromise where the adversary performs a successful account recovery

Adversary Model

Alice

Benign user

Eve

Controls: recovery method

Goal: persistent access

Mallory

Controls: recovery method

Goal: account takeover

Chad

Knows: recovery method, no access

Goal: spam or lock Alice out

Our contributions

1. Auditing Framework

How to conduct a systematic analysis of account recovery of any given web service?

2. Findings

Insights on what could go or has gone wrong in the wild?

3. Best Practice Recommendations

What needs to be taken into considerations for secure account recovery?

Terminologies

- **Account Recovery:** an automated process provisioned by the online service provider to their benign users for reclaiming access
- **Recovery method**: an independent communication channel agreed between the service provider and the user
- **Recovery token**: authentication material (e.g., one-time password or link) sent by the service provider to the user through the pre-agreed recovery method
  - The recovery token is submitted back to the service provider in the recovery session
  - The recovery token is used as an alternative to the unavailable credential and grant access
- **Recovery session**: A dedicated, unauthenticated session where an account recovery process takes place
  - ***Note:*** Transmission of recovery token from the service provider to the recovery method happens outside of the recovery session
- **Recovery window**: the duration for which the recovery token stays valid
- **Account Takeover**: Adversary gains control of the victim's recovery method and uses it to perform a *lateral compromise* of the target account associated with that method

Auditing Framework

Auditing framework

https://tinyurl.com/artha-framework

- **Test Setup**
  - **Test Environment:** to simulate real-life account recovery scenarios
  - **Test Process:** to guide the manual execution of the test cases
- **Test Cases**
  - Triggering recovery from different account states
  - Tinkering with recovery and MFA methods
  - Observing the recovery life cycle

Test Setup

Test Environment

Benign

"Trusted", previously logged in device/location

Semi-malicious

New device/location

Implementation
- VMs
- Browsers
- Proxies

Test Process

Agnostic to recovery flow variations

Covers applicable test cases and account states

Smooth transitions between account states

Avoids automatic lockouts and risk-based security controls

Test Case Summary

Test case #	Description
Test Case 1	Account creation tests
Test Case 2, 3, and 4	Account state specific tests
Test Case 5	Recovery when there are multiple recovery methods
Test Case 6	Session termination tests
Test Case 7	Use of MFA during recovery
Test Case 8	Interchangeability of the recovery and MFA factors/channels
Test Case 9	Settings review

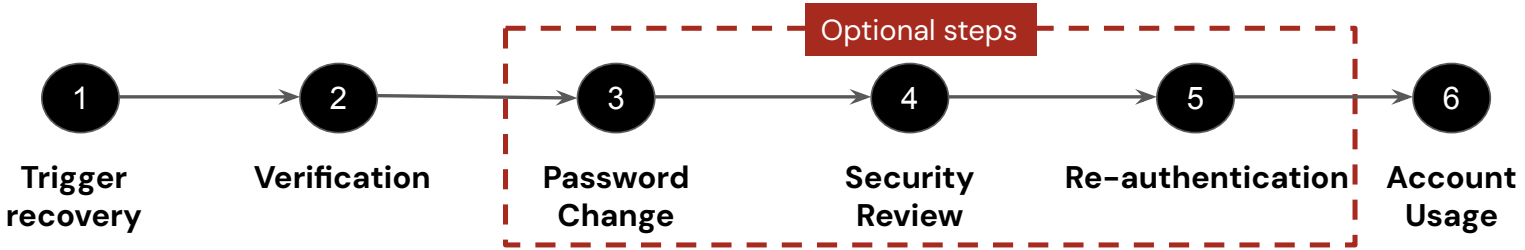
Account creation tests (Test case 1)

- Follow normal account creation and reach landing account state
- Check what information is collected during account creation:
  - Recovery methods
  - Whether MFA is enforced or not
  - Potential account functionality restricted after creation
- Only the mandatory fields of the forms are filled out
- The results of this test case indicate whether some attacks are invalid or not

Account state-specific tests (Test case 2, 3, and 4)

The goal of these test cases is to check how the recovery process works in these scenarios:

- Recovery when there is no recovery method
  - Is recovery even possible?
- Recovery from unverified recovery methods
  - Does the service provider inform that the method is unverified?
  - Are unverified methods use for recovery?
  - Are those methods marked as verified after a successful recovery?
- Recovery from verified recovery methods
  - Evaluate what happens during each of the stages of the account recovery lifecycle
  - Check the behavior when multiple recovery sessions are triggered simultaneously



Interplay between recovery and MFA methods

- Is it possible to have multiple recovery methods?
- Recovery when multiple methods available
- Recovery from a trusted vs untrusted device
- Behavior when there are changes to the recovery methods

Test case 5

- Leveraging MFA during recovery
- Is it possible to have multiple MFA factors?
- Recovery from a trusted vs untrusted device
- Behavior when there are changes to the MFA factors

Test case 7

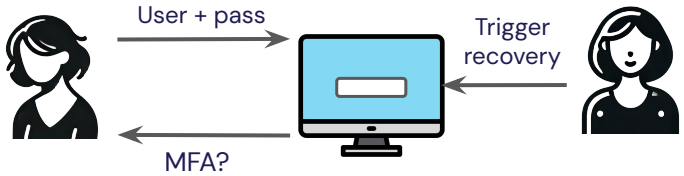
If there's a pool of recovery and MFA methods:

- Can recovery and MFA methods be used interchangeably during login and recovery?

Test case 8

Session termination (Test case 6)

Logged in session



Intermediate login session

Intermediate recovery session

- Impact of recovery on parallel sessions
- Termination of parallel sessions
- Differences between a benign or semi-malicious recovery

Settings Review (Test case 9)

Analyze account settings presented by the service providers:

- How are recovery methods and MFA factors presented to the user?
- Is there an activity log for the account?
- Is there additional authentication required for changing the security settings of the account?
  - Adding/removing recovery methods
  - Adding/removing MFA factors
  - Revoking existing sessions
  - Does recovery impact this?

Findings

Dataset for Empirical Analysis

- **Source: Tranco list** (https://tranco-list.eu/)
  - Research-oriented ranking of 1 Million websites
  - Standard for web security and Internet measurement empirical analysis
- **Shortlisted dataset**: Tranco 1M → 200 top websites →25 websites
  - Combination of top and random (excluding the top 13)
  - Matches the following selection criteria
    - ❑ Available in English
    - ❑ Non-explicit (safe at work content)
    - ❑ Fully accessible from desktop browsers
    - ❑ Allows free of charge account creation
    - ❑ Does not require real-life identities
    - ❑ Supports multi-factor authentication
    - ❑ Allows logging in with a website-based credential (not just SSO)
- Results presented in this talk are from the **22 websites**

1. Design Flaws

2. Security Policy Weaknesses

3. Missing Best Practices

1. **Design Flaws**

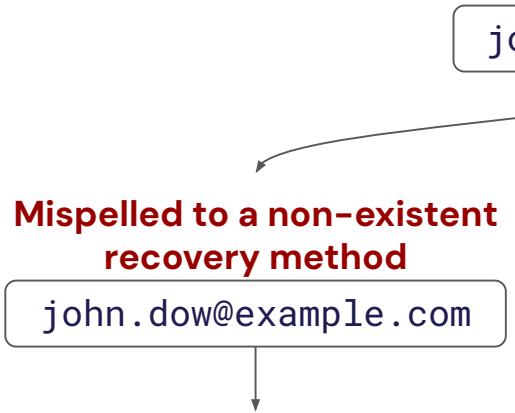
- Mistakes in system's architecture or logic
- UI design related or hampers UX
- Mismatches and inconsistencies

2. Security Policy Weaknesses

3. Missing Best Practices

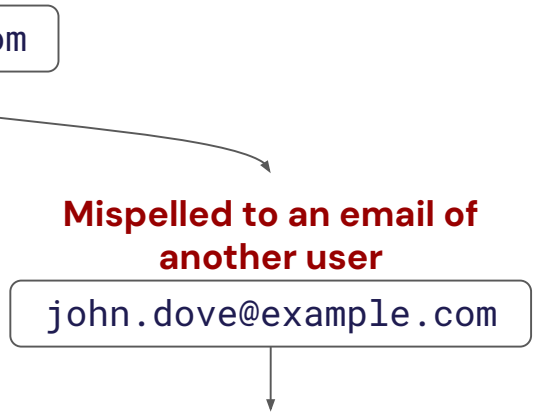
Design Flaws

#1 Use of unverified recovery methods



- User cannot recover their account
- May require human intervention similar to account remediation
- Leads to dummy or stale accounts

#2 Inconsistent verification



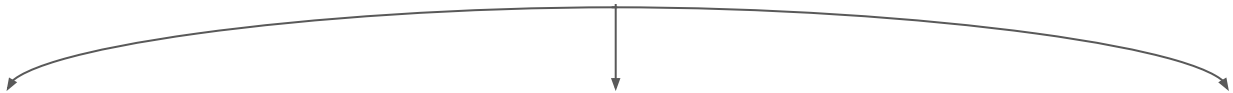
- John Dove may takeover John Doe's account
- Account remediation is required

#3 Restricting security functionalities until verification

Design Flaws

#4 Recovery flow doesn't match account states

- Email used as usernames becomes a default recovery method. But, what if
  - Account creation with email providers?
  - Only username and password required for account creation?
- What happens if there is no recovery method but the recovery is triggered?



Recovery not possible

- Unpleasant user experience
- Leads to dummy or stale accounts

Requires human intervention

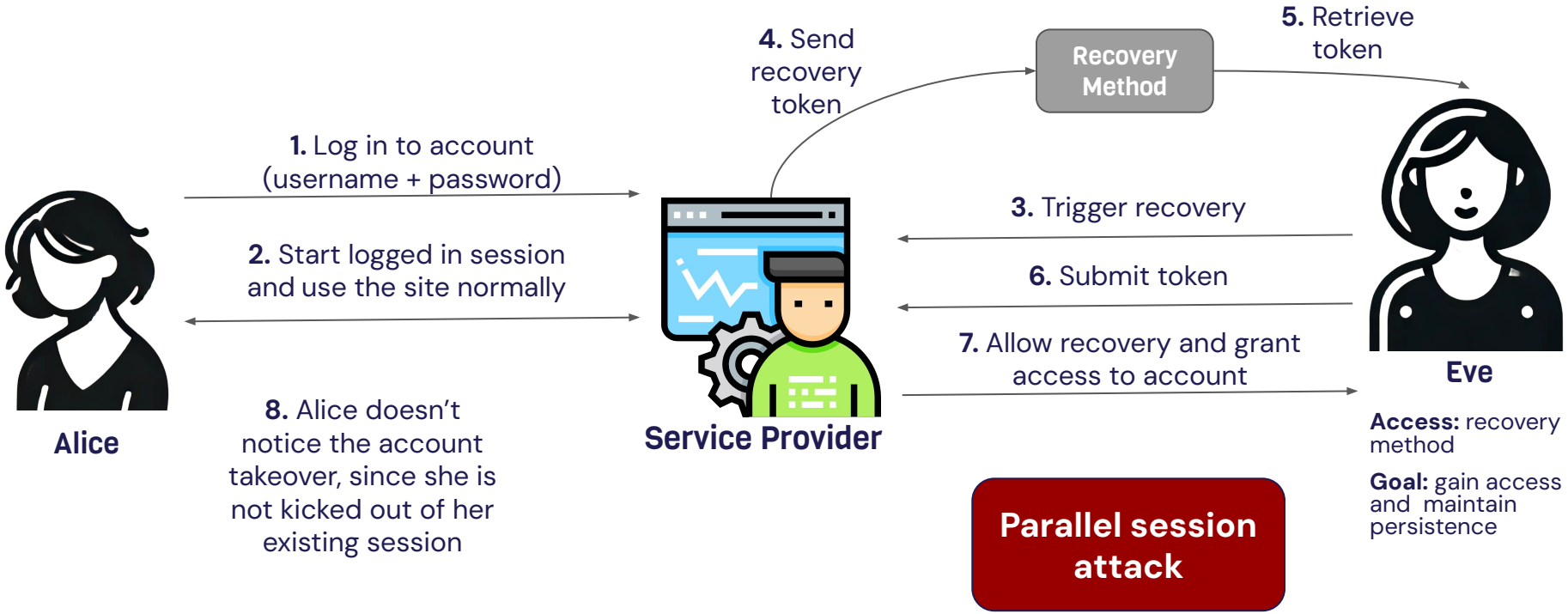
- Not scalable
- Expensive
- Unnecessary exposure of real-life identities

Recovery based on less secure heuristics

- Susceptible to evil maid attacks
- Falls back to case 1 or 2

Design Flaws

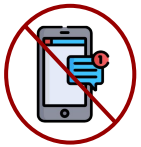
#5 Parallel sessions are allowed to continue after recovery



Design Flaws

#6 Inflexible rules

Restrictions on recovery methods



Lack of fallback options make recovery harder or unpleasant

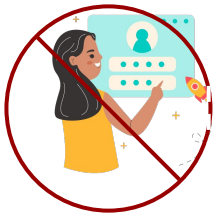
Restrictions on MFA methods

Limits MFA usability or hampers usability

#7 Missing or unprompted activity logs

Activity log does not exist

User forfeits the option to make informed security decisions



Exists, but user not nudged

Underutilized feature that could have helped to improve security

1. Design Flaws

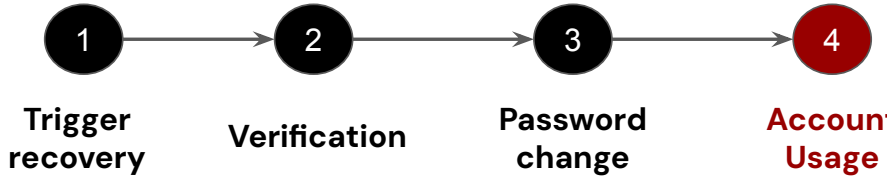
2. Security Policy Weaknesses

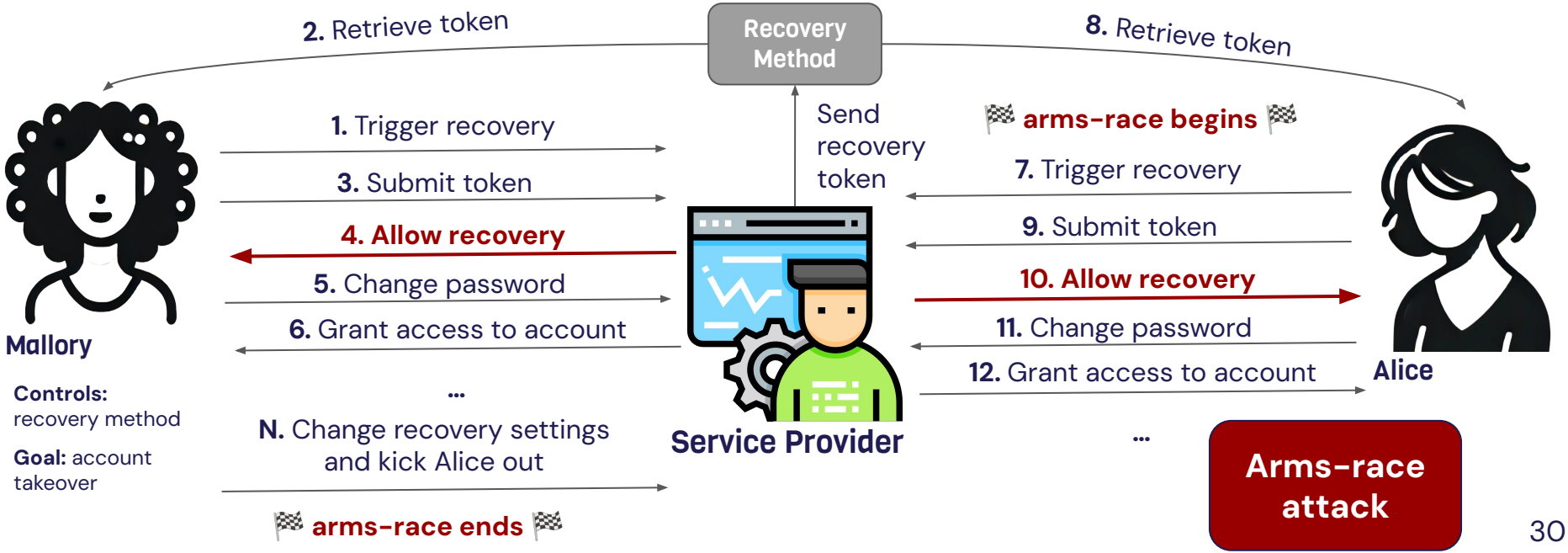
- Flaws in definition, scope, or enforcement of policies
- Too strict or too lenient rules
- Missing and insufficient policies

3. Missing Best Practices

Security Policy Weaknesses

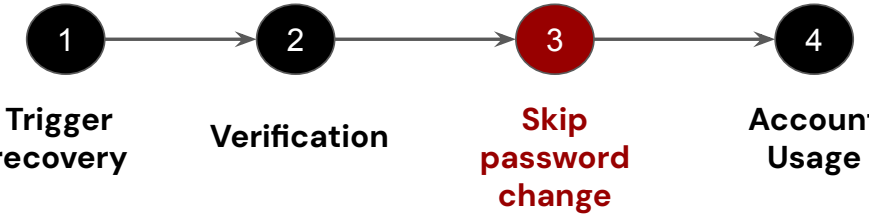
#1 MFA is not used during recovery

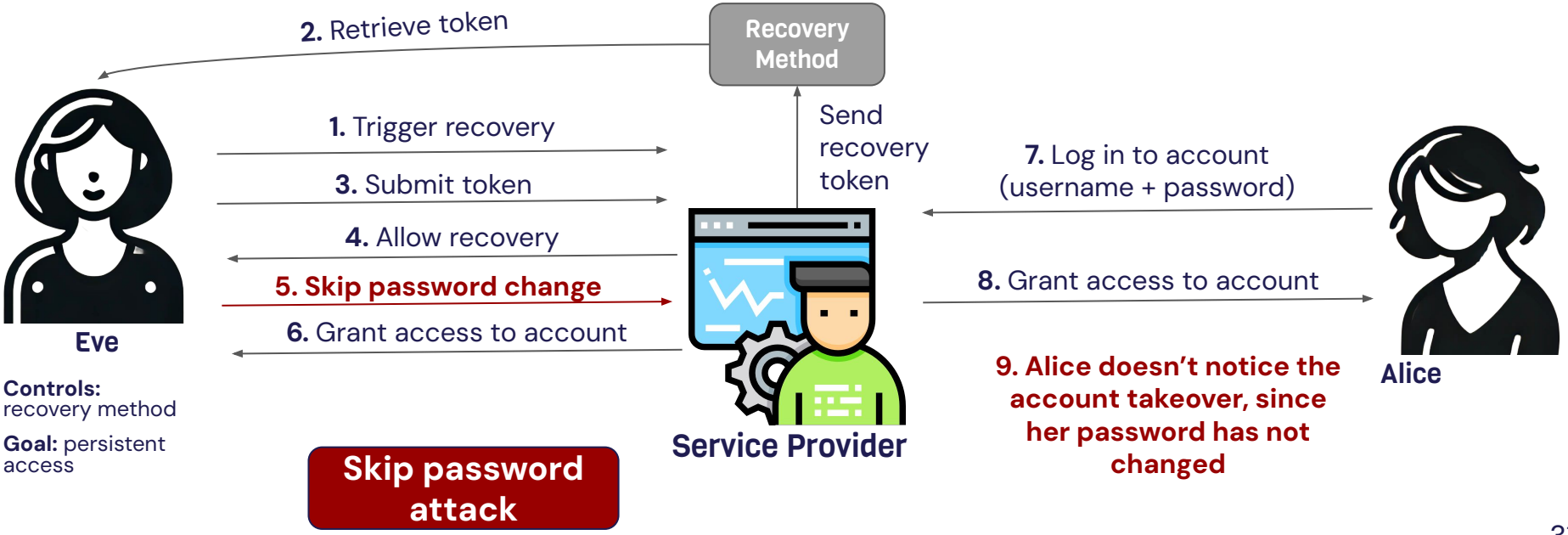




Security Policy Weaknesses

#2 Password change is not enforced after recovery





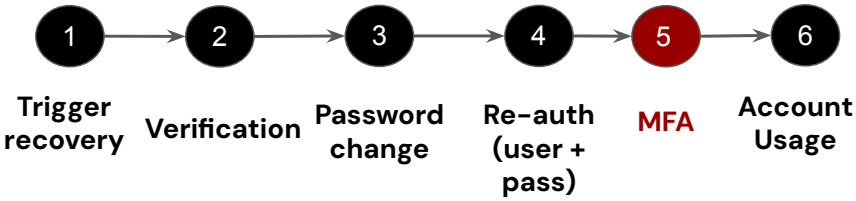
Isn't MFA always used?

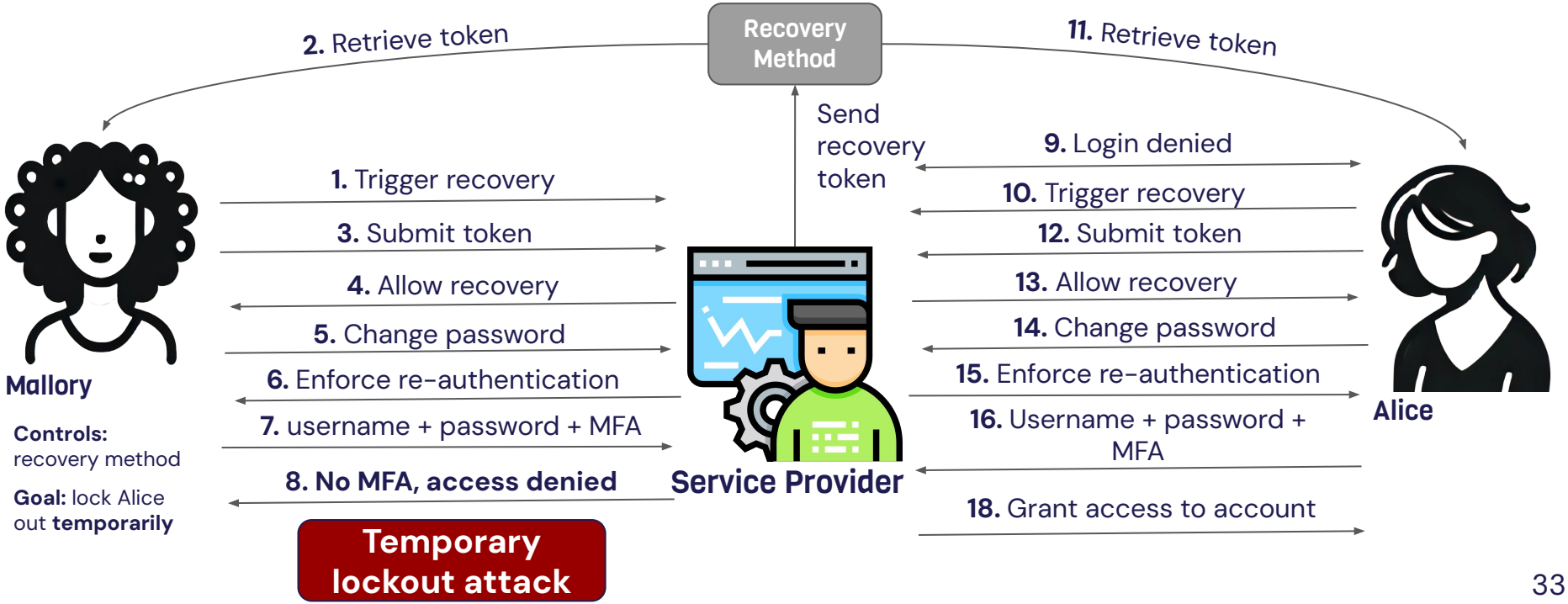
What if we add MFA to the mix?

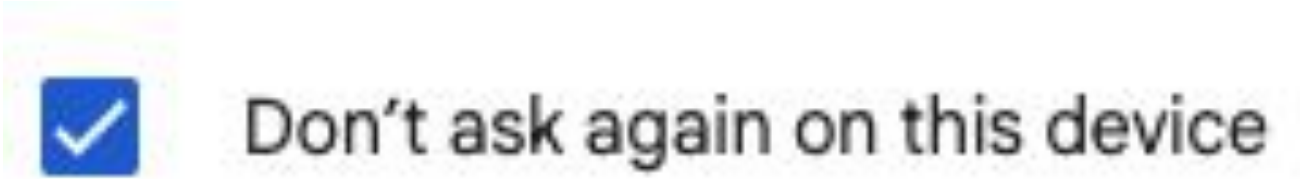
Isn't MFA the golden standard?

Security Policy Weaknesses

#3 MFA is only used after recovery

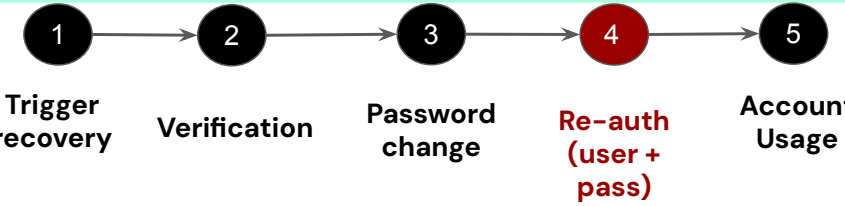


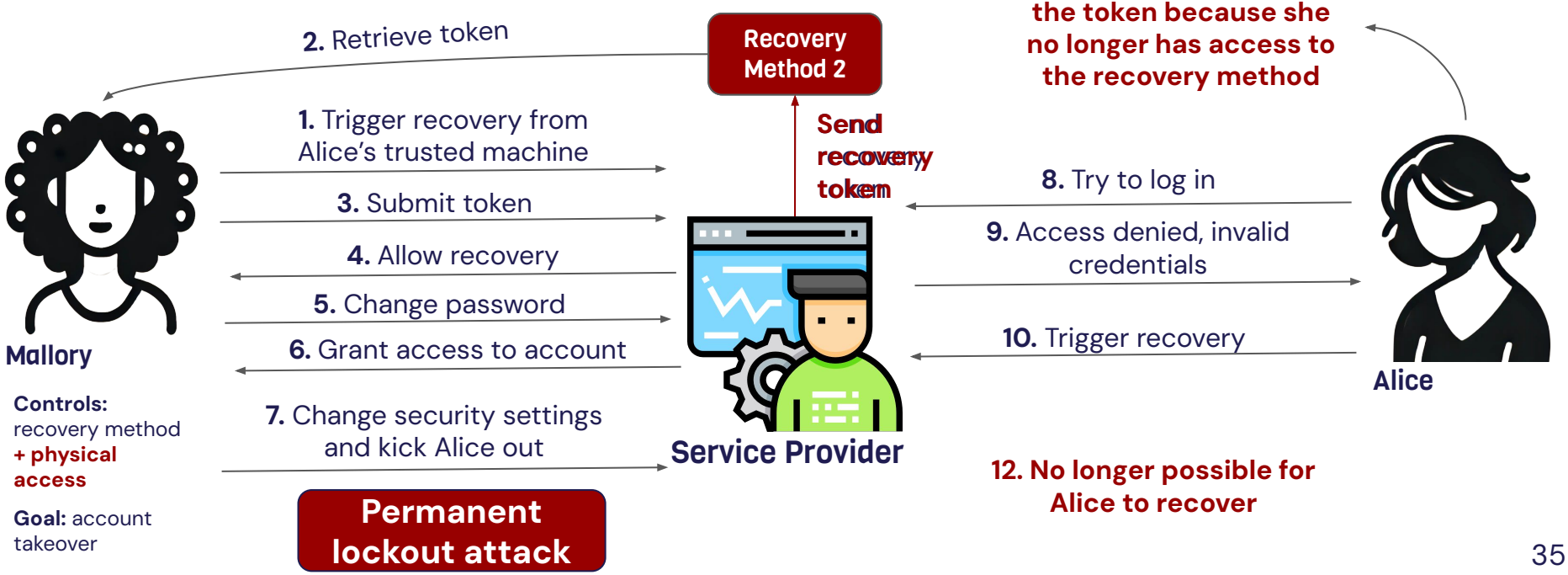



Don’t ask again on this device

Security Policy Weaknesses

#4 MFA is not used from a trusted device

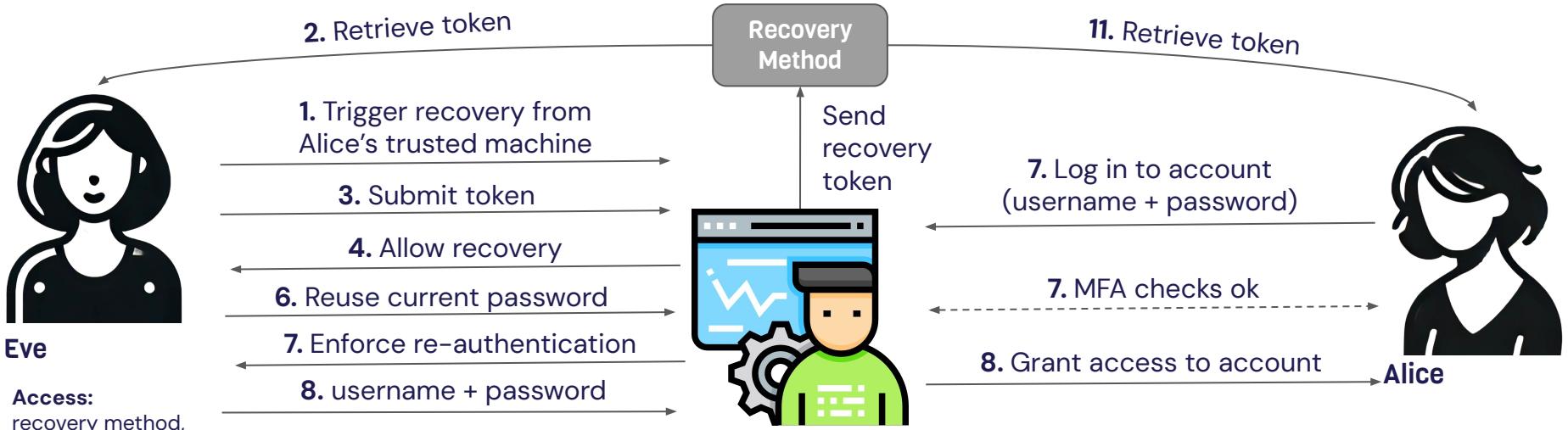




Security Policy Weaknesses

#5 Password policies are not applied to recovery lifecycle

Password reuse attack



Access: recovery method, physical access, knows password

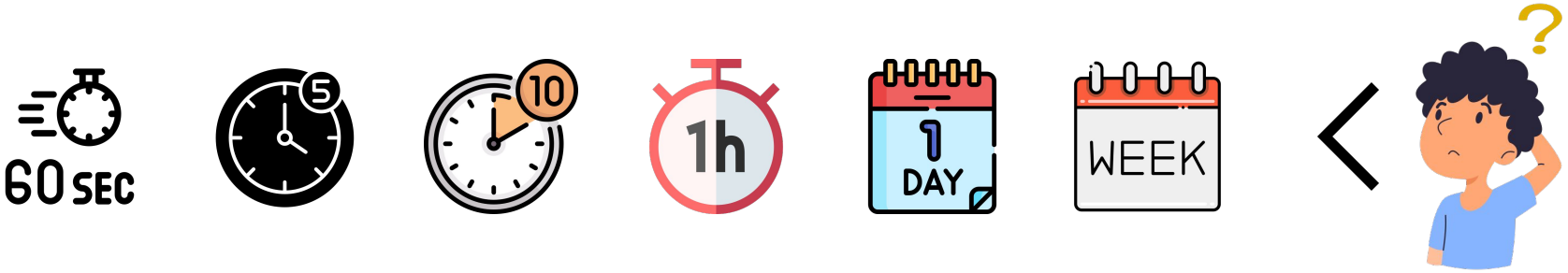
Goal: gain access and maintain persistence

9. Alice doesn't notice the account takeover, since her password has not changed

Security policy weakness

#6 Long recovery windows or flawed recovery token expiration policies

How long should be the recovery window?



Insecure or bad examples of token termination policies

- Expires after 1 week
- Expires upon use **but no auto expiration**

} **Increased exposure of attack window**

- **Does not expire on use**, but auto expires after XX duration

} **Token reuse**

1. Design Flaws

2. Security Policy Weaknesses

3. Missing Best Practices

- Best practice not followed
- Generic best practice is not applicable or insufficient
- Best practice not available

Missing Best Practices

#1 Inconsistency in communicating alerts

Alerts are not sent for all changes to security settings

Emails are prioritized as communication channels for alerts

Account takeover attempts may go unnoticed to Alice

🏁 **arms-race attack** 🏁

Mallory

Goal: account takeover

Change recovery settings and kick Alice out

Service Provider

No alerts sent to Alice about the changes

Alice

Missing Best Practices

#2 Account creation allows unsafe states

Unverified recovery methods

MFA not required

Weaker recovery process

#3 Inconsistent treatment between recovery and MFA methods

Guidance on how to secure MFA and its purpose

Changes to MFA required re-authentication

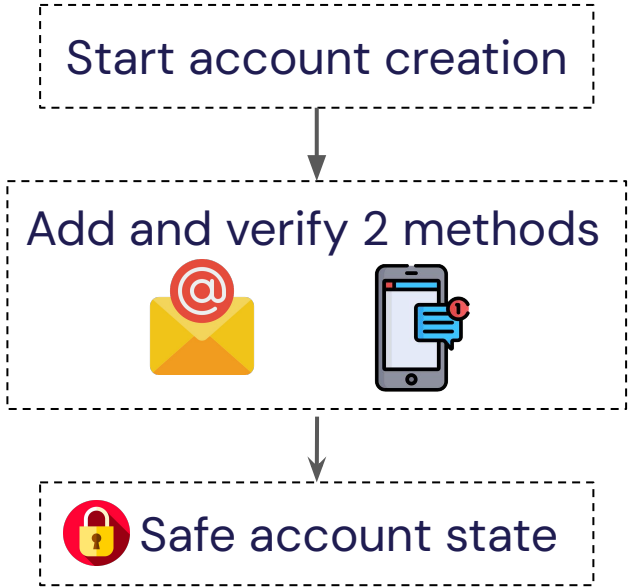
Does not apply to recovery methods

Lost opportunity to leverage user's secure habits

Best Practice Recommendations

For Account Creation

Assume user may need recovery right after signup



Must have

- ❏ Two or more **verified** authentication methods
- ❏ Of **different types**

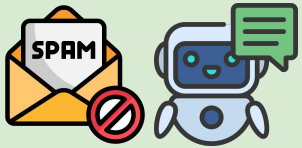
For short flow

- ❏ Use implicit recovery methods to avoid A_{BM}
- ❏ Nudge users with alert ribbons and restricted use

For Recovery Triggering

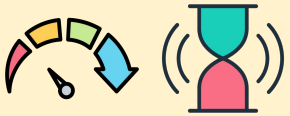
Recovery triggering is an unauthenticated action for which imposing access control is unfeasible

Defend



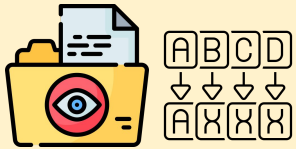
- ❑ Anti-bot protection
- ❑ Human verification

Decelerate



- ❑ Manual typing in the UI fields
- ❑ Avoid copy-pasting or auto-filling

Avoid data leaks



- ❑ Don't leak unnecessary PII
- ❑ Partially mask recovery hints

Free user choice

- ❑ No restrictions in recovery options
- ❑ Default can be most or recently used

For Recovery Processing (1)

Recovery flows should not assume by default that the recovery method is intact

Secure and indivisible process

- ❏ Always do a two-factor recovery
- ❏ Batch process the two factors (i.e., tokens) to avoid TOCTOU

Interchangeability of factors

- ❏ Recovery and MFA methods should be interchangeable
- ❏ Available from the same pool → free user choice

Noticeably intrusive to ignore

- ❏ Logged-in session terminates with an alert
- ❏ New credentials must be set such that old one is obsolete
- ❏ If “skip password” is inevitable, alert the security risks

For Recovery Processing (2)

Parallel recovery flows make it hard to assess the benign intent

Session Policy

- ❏ Recovery triggering should not terminate any ongoing sessions
- ❏ Successful recovery should terminate all types of parallel sessions

Sessions and tokens should be bound

Service providers and users should be able to review and revoke individually

Token Policy

- ❏ Recovery token validity should be short and not to be extended
- ❏ Communicate the validity to the users and nudge to complete in time

For Alert Notifications

Notifying at the right moment with the right content can save account takeover and remediation

Venues

- ❑ Website or app UI
- ❑ Push notifications
- ❑ Browser notifications
- ❑ Pop-ups and alert ribbons
- ❑ Via authentication methods

Occasions

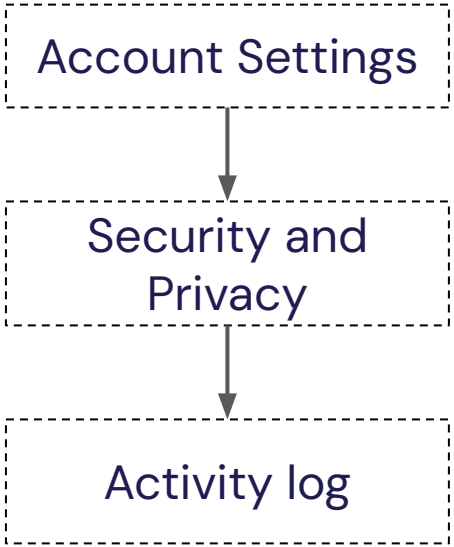
- ❑ Account state changes
  - ❑ Alterations to authentication methods
  - ❑ Verification of unverified methods
- ❑ During recovery
  - ❑ *Trigger* → recovery method + active session
  - ❑ *Successful recovery* → all channels
- ❑ During suspicious activities

Contents

- ❑ Incident details
  - ❑ Incident type
  - ❑ Metadata
- ❑ Next steps
  - ❑ Contents of the alert
  - ❑ Additional info needed
- ❑ Security concerns
  - ❑ Anomalies
  - ❑ Associated risks
  - ❑ Reporting

For Reviewing Recovery Events

Reviews should help users analyze, revoke and report suspicious activities that the service provider alone cannot verify



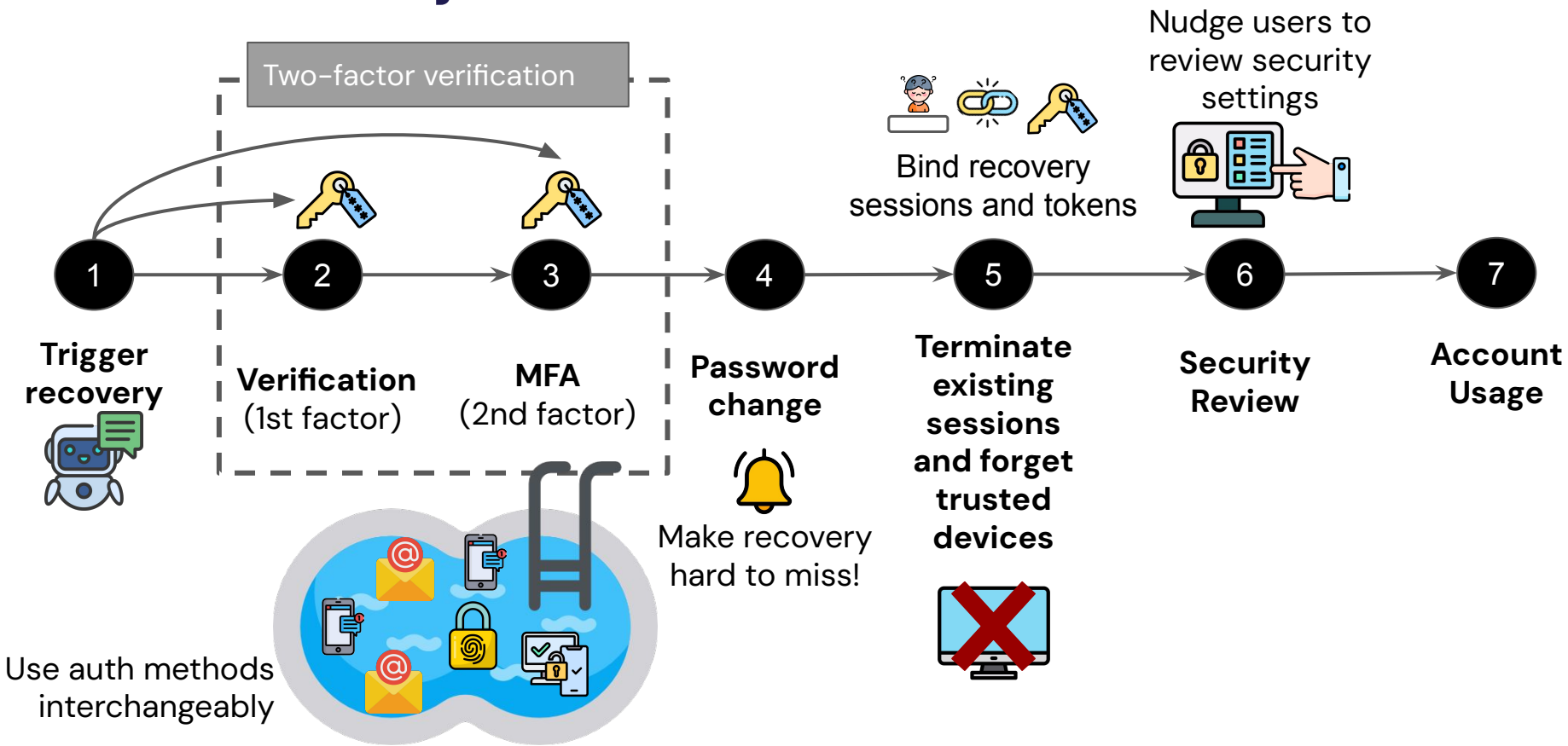
Activity log properties

- ❏ Tamper-proof
- ❏ Detailed
- ❏ Highlights anomalies
- ❏ Mention security risks
- ❏ Option to report

User nudging

- ❏ What?
  - ❏ To remove or update obsolete methods
  - ❏ To Identify and report suspicious entries
- ❏ When?
  - ❏ After account creation
  - ❏ After recovery
  - ❏ After remediation
  - ❏ Upon reporting suspicious activities

Ideal recovery flow



Closing Remarks

Recap

3	8	15
Adversaries	Attacks	Weaknesses

- Tested **22** most popular websites
  - all of them had **at least 1 security issue**
- **There could be more vulnerable websites and more security issues!!**
  - Contribute and use our auditing framework – ARTHA

Attacks Summary

Adversary	Potential Attacks	Description
Eve *Controls*: recovery method *Goal*: persistent access	Skip password persistence	Exploits the "skip password" option and no MFA in the recovery lifecycle to gain stealth access
	Password reuse persistence	Exploits the password reuse during recovery and no MFA needed on trusted device to gain stealth access
	Parallel session attack	Exploits active parallel sessions not terminating upon successful recovery to gain stealth access
Mallory *Controls*: recovery method *Goal*: account takeover	Arms race attack	Exploits no MFA in the recovery lifecycle to take part in an arms race and potential win to lockout the victim
	Temporary lockout	Exploits MFA needed only for login after recovery to lockout temporarily until victim can do recovery + login
	Permanent lockout	Exploits no MFA needed on trusted device to permanently lockout the victim
Chad *Knows*: recovery method *Goal*: spam or lockout	Recovery spam	Exploits lack of anti-spamming and control on recovery triggering to spam the victim
	Recovery lockout	Exploits lack of anti-spamming and control on recovery triggering to activate automatic lockout feature

Key Takeaways

- **Security vs usability trade-offs could lead to risky gaps**
  - Ease of account recovery over security
    - Low-friction but high-risk recovery mechanism
  - When OoB channels are not under control or cannot be monitored
    - Make no trust assumption
    - Utilize every heuristics and channels available
    - Prioritize security over usability

- **Non-typical security weaknesses could be harmful**
  - Equally harmful as any traditional software or hardware vulnerabilities
  - Low-tech adversaries can exploit
    - No scripting, coding
    - No tools required
    - No sophisticated bugs
    - No internal access or knowledge

- **Bridge the research-practice gaps**
  - Security audits and certification focuses on **internal evaluation** of policies and processes
    - However, the weaknesses discussed in this work mostly are out of scope of conventional vulnerability scanning or pen testing
  - Security research focuses on **external validation** of overlooked or missing best practices
  - Our work bridges this gap as an auditing carried out by an adversary outside of the system, effectively performing ***Attack surface mapping of account recovery***

Points to Remember Moving Forward

- **Users are NOT the weakest link in authentication, but account recovery is!**
  - Security weaknesses are not stemmed from user actions or knowledge, but mostly due to the oversight of service providers where users do not have a say
- **Weaknesses in account recovery goes beyond security and hacking**
  - Exploitable in cases of intimate partner violence and stalking where recovery weaknesses become *tools of controls and power*!
  - Real-world adversaries are mostly insiders who exploit the weaknesses to reset access, monitor every activity or lock out victims
- **Authentication will evolve, but account recovery may remain stagnant**
  - Insights and lessons from this work will stay relevant to future systems
  - Our work lays the foundation for standards, red team tooling, and compliance checks, influencing how to design, test, and monitor recovery processes

Thank You!

Contact

Sid Rao

Email: sid.rao@nokia-bell-labs.com

Linkedin: https://www.linkedin.com/in/siddharthprao/

Gabriela Sonkeri

Email: gabriela.sonkeri@gmail.com

Linkedin: https://www.linkedin.com/in/gabrielalimonta