Cross Device Flows

Pieter Kasselman Daniel Fett Filip Skokan

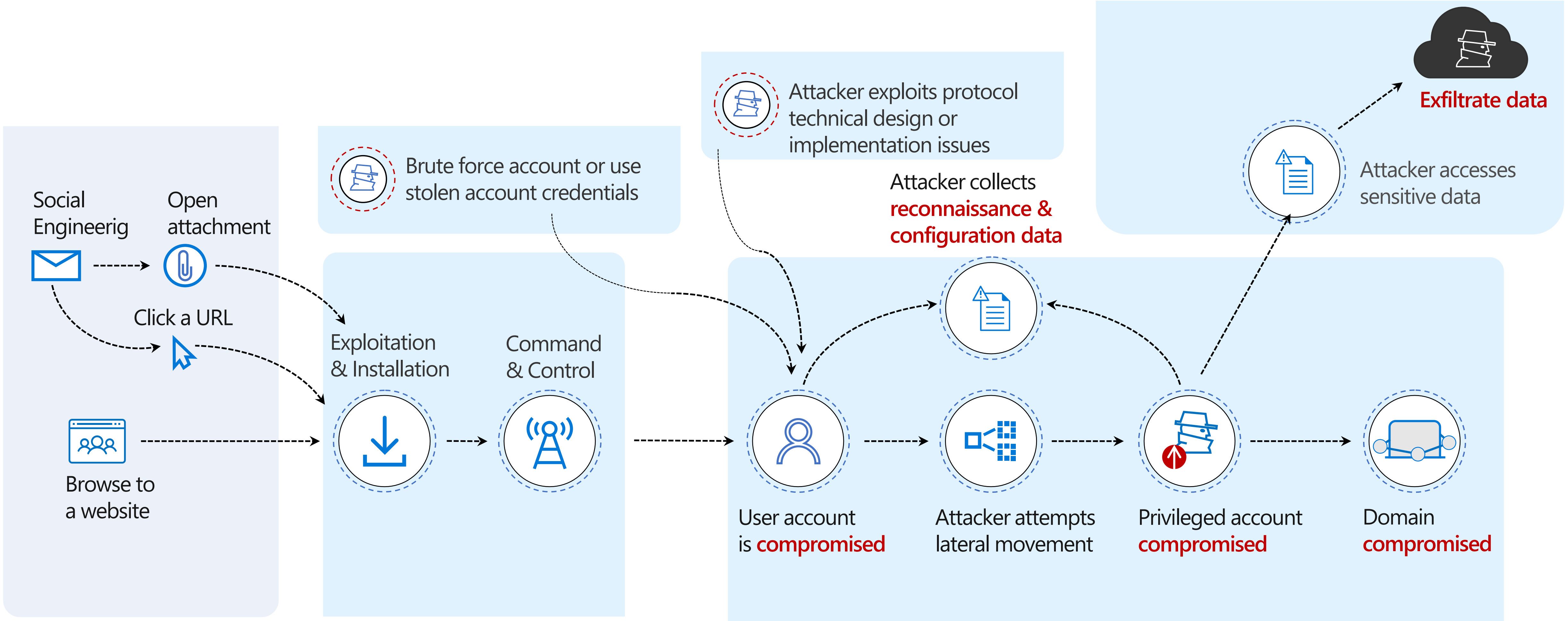
IETF 118 Prague
Date: 8 November 2023

Agenda

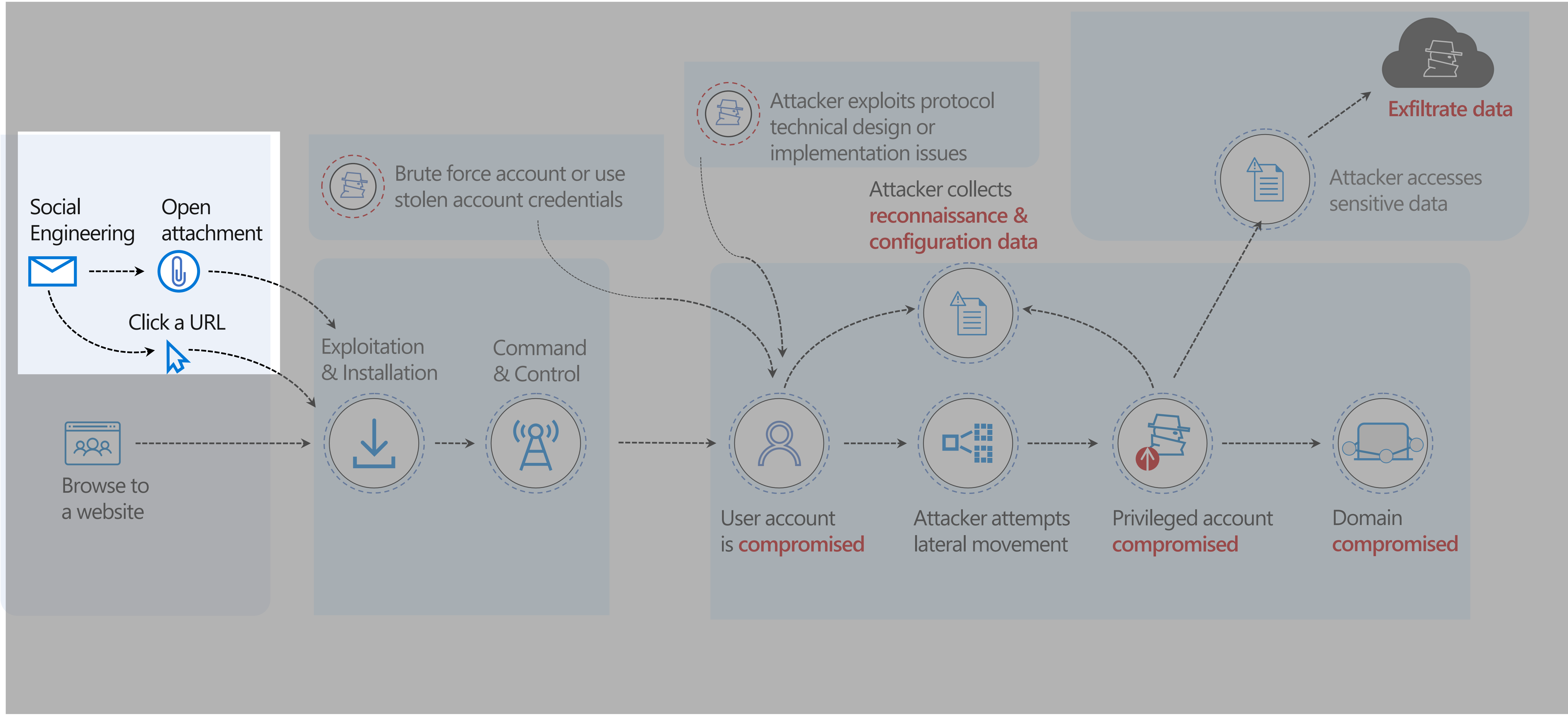
- Why are we here?
- Where are we?
- Where do we go next?

Why are we here?

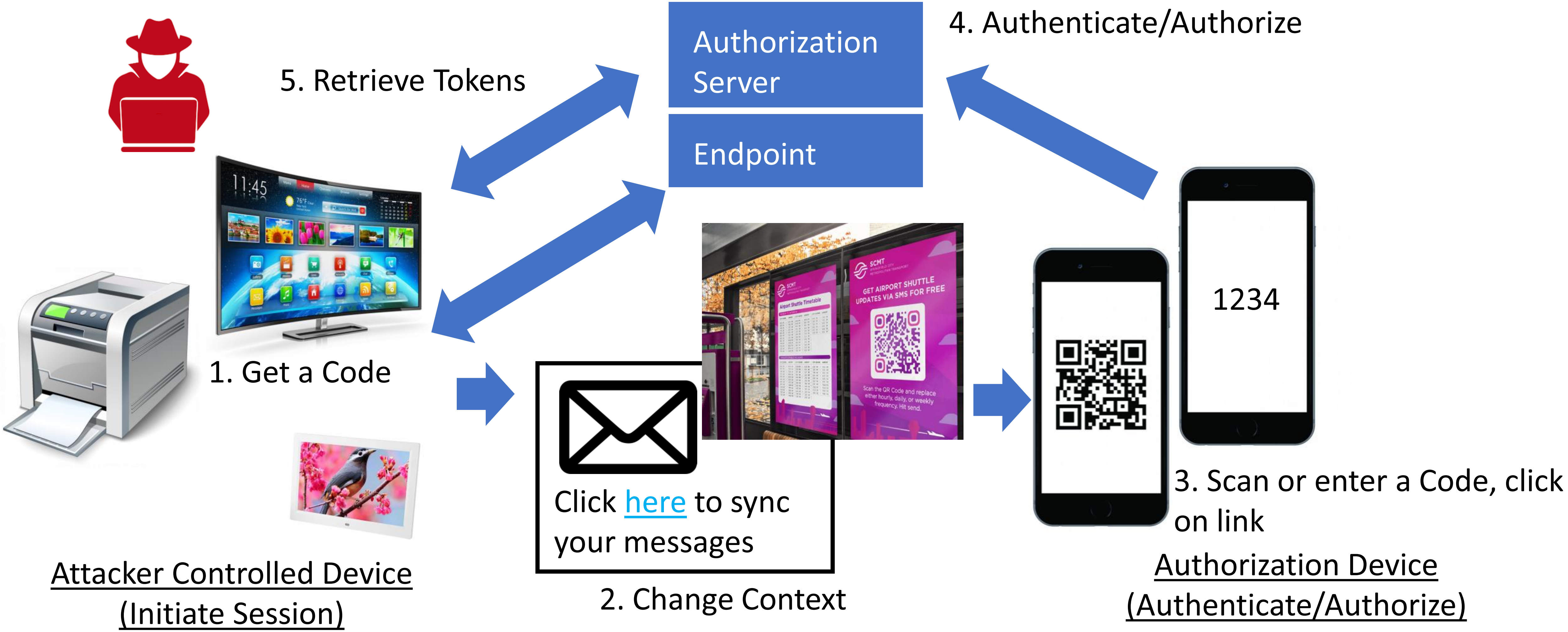
Anatomy of an attack


Social Engineerig
Open attachment
Click a URL
Browse to a website
Brute force account or use stolen account credentials
Exploitation & Installation
Command & Control
Attacker exploits protocol technical design or implementation issues
Attacker collects reconnaissance & configuration data
User account is compromised
Attacker attempts lateral movement
Privileged account compromised
Domain compromised
Attacker accesses sensitive data
Exfiltrate data

Mind the Gap – Where Attackers (often) Enter


Social Engineering
Open attachment
Click a URL
Browse to a website
Brute force account or use stolen account credentials
Exploitation & Installation
Command & Control
Attacker exploits protocol technical design or implementation issues
Attacker collects reconnaissance & configuration data
User account is compromised
Attacker attempts lateral movement
Privileged account compromised
Domain compromised
Attacker accesses sensitive data
Exfiltrate data

Cross-Device Flow Social Engineering Exploit



11:45
SCMT
GET AIRPORT SHUTTLE UPDATES VIA SMS FOR FREE
Airport Shuttle Timetable
Scan the QR Code and replace either hourly, daily, or weekly frequency. Hit send.
1234

Authorization Server

Endpoint

1. Get a Code

2. Change Context

Click here to sync your messages

3. Scan or enter a Code, click on link

4. Authenticate/Authorize

5. Retrieve Tokens

Attacker Controlled Device (Initiate Session)

Authorization Device (Authenticate/Authorize)

Attack Pattern Summary: Exploit the Unauthenticated Channel

1. Initiate the session, retrieve code (QR code, user code)
2. Use social engineering to change context and persuade user to authorize session (illicit consent grant)
3. Bypasses multi-factor authentication (don’t need to harvest credentials)

Mitigation Framework

Pragmatic Mitigations

- Other.....
- User Experience
- Secure QR
- Trusted Devices
- Sender Constrained Toke
- User Code Meta Data
- Content Filtering
- Proximity

Explore Alternatives

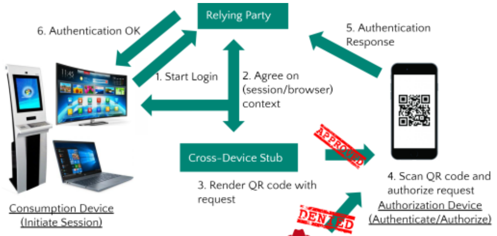
Authenticated Channel

- Authorization Code Grant (WebAuthn fido)

Unauthenticated Channel

- Client Initiated Back Channel Authentication (OpenID)
- Device Authorization Grant

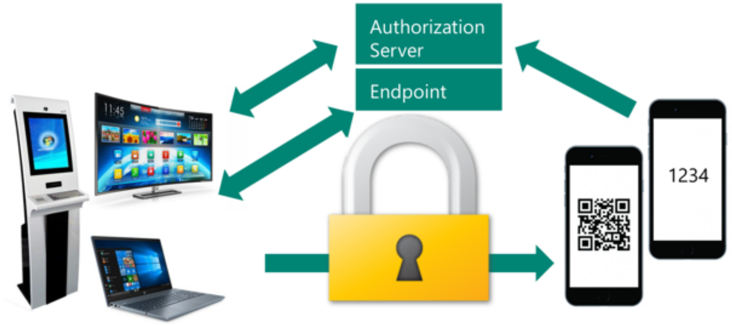
Foundational Underpinnings



Relying Party
6. Authentication OK
1. Start Login
2. Agree on (session/browser) context
5. Authentication Response
Cross-Device Stub
3. Render QR code with request
APPROVED
DENIED
4. Scan QR code and authorize request
Authorization Device (Authenticate/Authorize)
Consumption Device (Initiate Session)

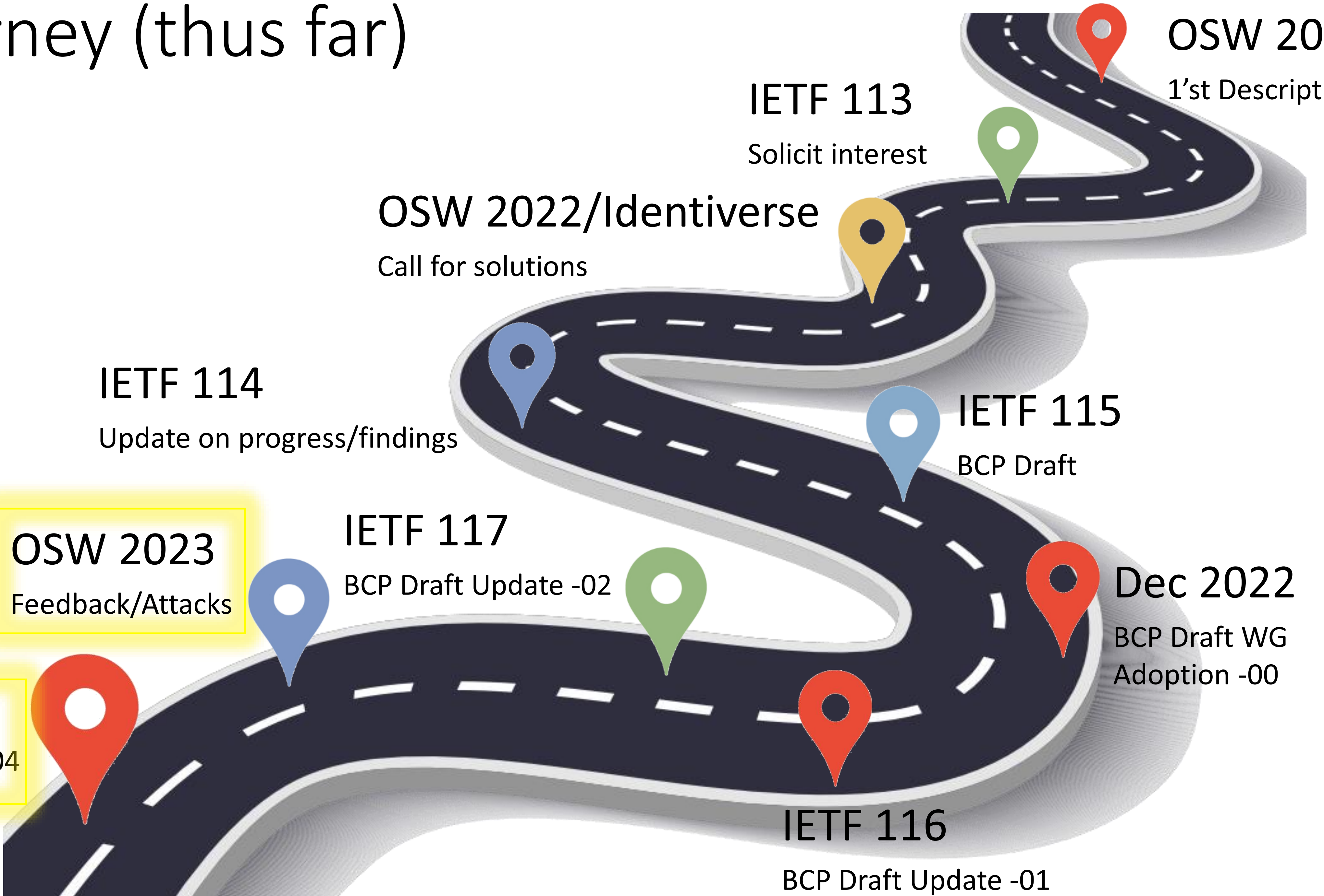
Summary

1. Move part of authorization device to the web
2. Perform checks on request using this web service
3. Pairing of stub web service and authorization device application



Where are we?

The Journey (thus far)



Attacks

OSW 2021

1’st Description

IETF 113

Solicit interest

OSW 2022/Identiverse

Call for solutions

IETF 114

Update on progress/findings

IETF 115

BCP Draft

Dec 2022

BCP Draft WG
Adoption -00

IETF 116

BCP Draft Update -01

IETF 117

BCP Draft Update -02

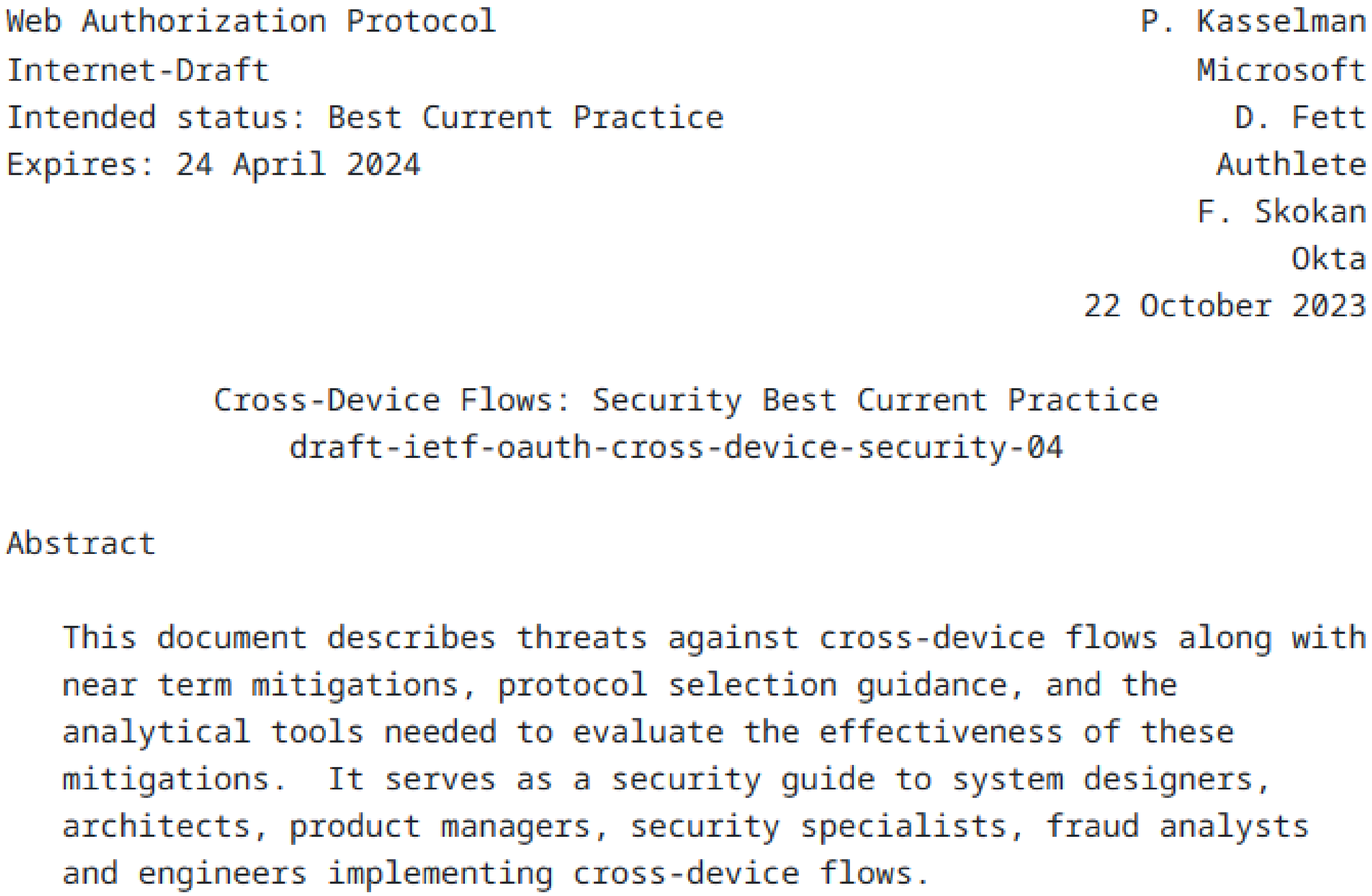
OSW 2023

Feedback/Attacks

IETF 117

BCP Draft Update -04

Cross-Device Flows: Security Best Current Practice



Web Authorization Protocol
Internet-Draft
Intended status: Best Current Practice
Expires: 24 April 2024

P. Kasselman
Microsoft
D. Fett
Authlete
F. Skokan
Okta

22 October 2023

Cross-Device Flows: Security Best Current Practice
draft-ietf-oauth-cross-device-security-04

Abstract

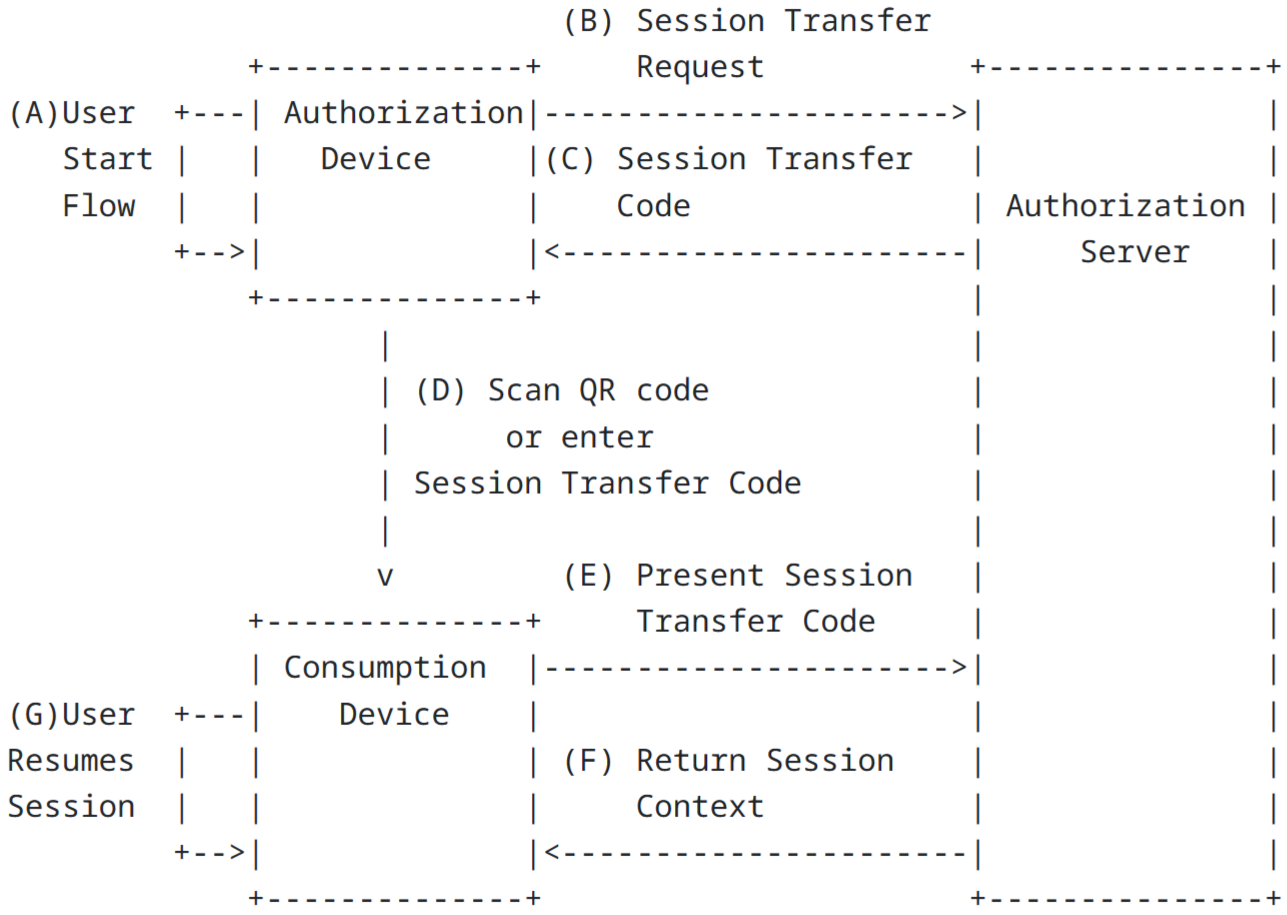
This document describes threats against cross-device flows along with near term mitigations, protocol selection guidance, and the analytical tools needed to evaluate the effectiveness of these mitigations. It serves as a security guide to system designers, architects, product managers, security specialists, fraud analysts and engineers implementing cross-device flows.

https://datatracker.ietf.org/doc/draft-ietf-oauth-cross-device-security/

What’s New: Cross-Device Session Transfer Pattern

Feedback at OSW

- Two of the examples did not cleanly map to the general patterns described.
  - Example A5/B5 and A7/B7
  - User starts flow on authorization device, not consumption device.
  - QR code is scanned to transfer a session not request authorization
  - Cross-Device Session Phishing
  - Example: OpenID4VCI pre-auth code
- Thanks to Marco Pernpruner and Giada Sciarretta



```
                              (B) Session Transfer
             +--------------+     Request           +----------------+
(A)User  +---| Authorization|---------------------->|                |
   Start |   |   Device     |(C) Session Transfer   |                |
   Flow  |   |              |    Code               | Authorization  |
         +-->|              |<----------------------|     Server     |
             +--------------+                       |                |
                    |                               |                |
                    | (D) Scan QR code              |                |
                    |      or enter                 |                |
                    | Session Transfer Code         |                |
                    |                               |                |
                    v         (E) Present Session   |                |
             +--------------+     Transfer Code     |                |
             | Consumption  |---------------------->|                |
(G)User  +---|    Device    |                       |                |
Resumes  |   |              | (F) Return Session    |                |
Session  |   |              |     Context           |                |
         +-->|              |<----------------------|                |
             +--------------+                       +----------------+
```

Figure 4: Cross-Device Flows: Session Transfer Pattern

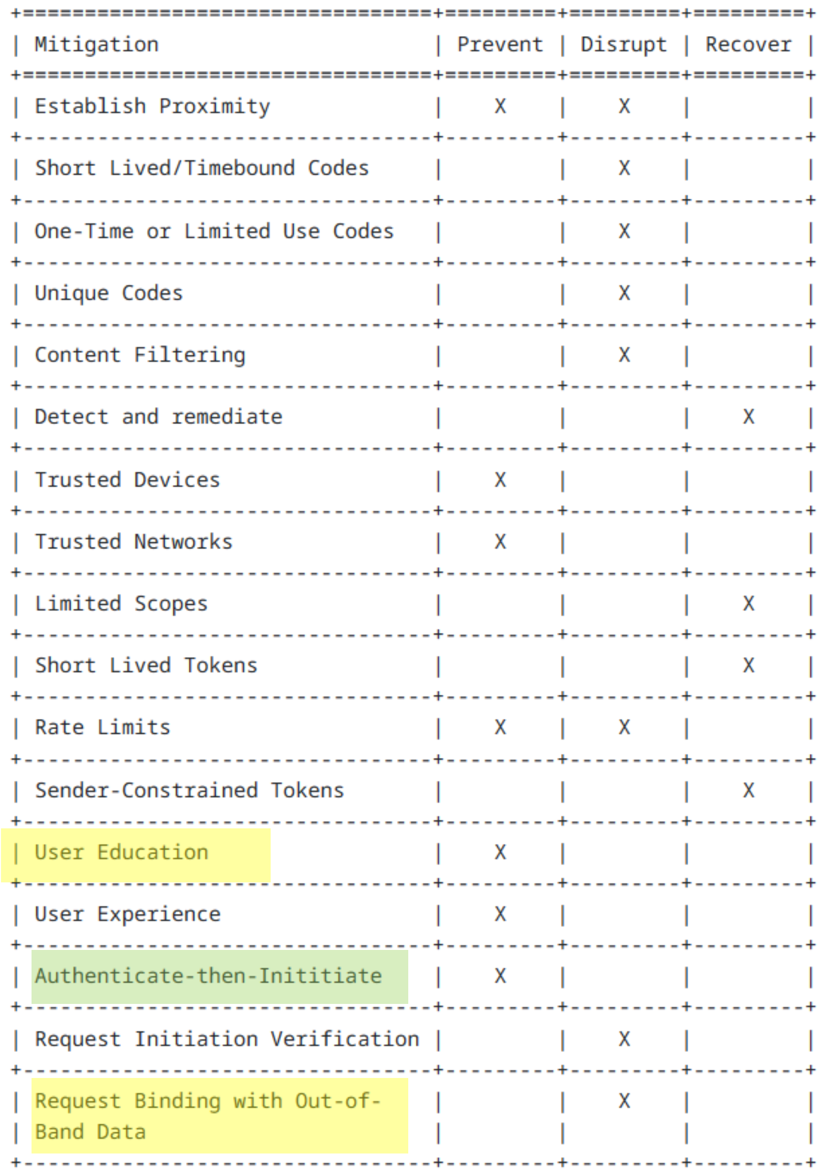
What’s New: 2 new mitigations, 1 Rename

New mitigations

- User Education
- Request Binding with Out-of-Band Data

New name

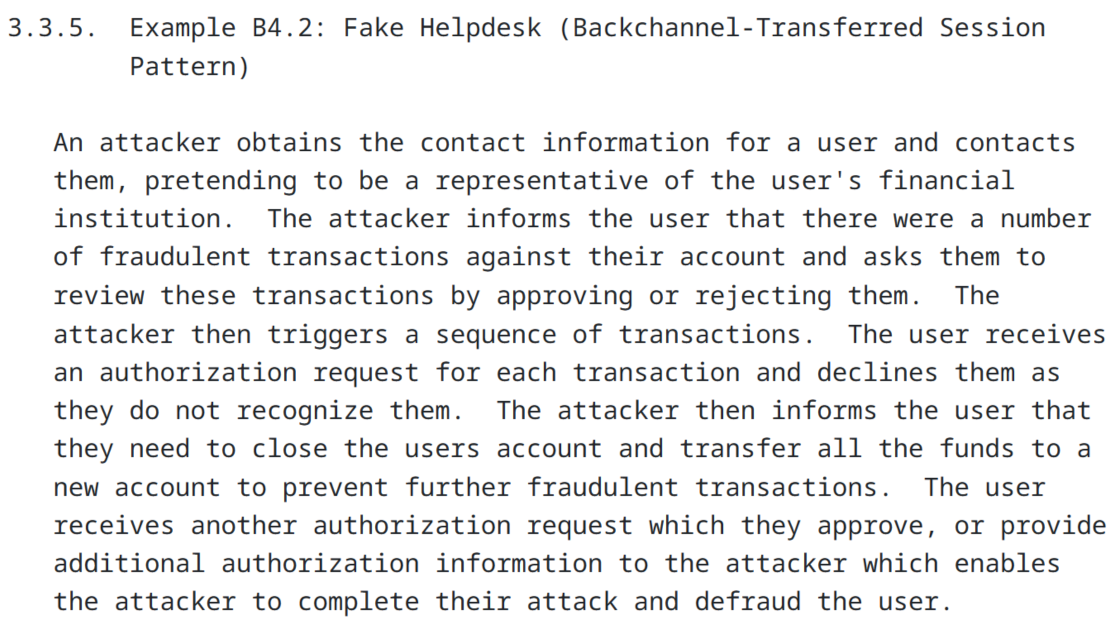
- Authenticate-then-Initiate



Mitigation	Prevent	Disrupt	Recover
Establish Proximity	X	X	
Short Lived/Timebound Codes		X	
One-Time or Limited Use Codes		X	
Unique Codes		X	
Content Filtering		X	
Detect and remediate			X
Trusted Devices	X		
Trusted Networks	X		
Limited Scopes			X
Short Lived Tokens			X
Rate Limits	X	X	
Sender-Constrained Tokens			X
User Education	X		
User Experience	X		
Authenticate-then-Inititiate	X		
Request Initiation Verification		X	
Request Binding with Out-of-Band Data		X	

What’s New: 2 new exploits observed in the wild

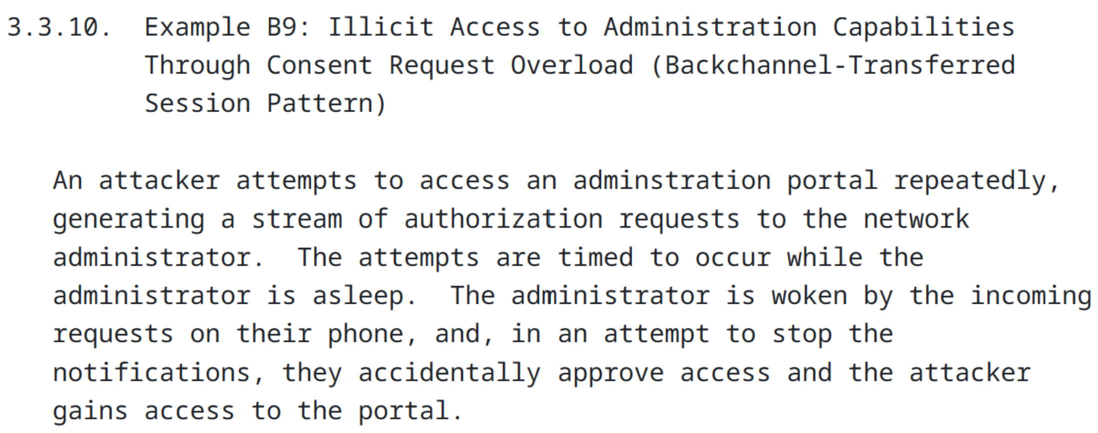
Fake Helpdesk



3.3.5. Example B4.2: Fake Helpdesk (Backchannel-Transferred Session Pattern)

An attacker obtains the contact information for a user and contacts them, pretending to be a representative of the user's financial institution. The attacker informs the user that there were a number of fraudulent transactions against their account and asks them to review these transactions by approving or rejecting them. The attacker then triggers a sequence of transactions. The user receives an authorization request for each transaction and declines them as they do not recognize them. The attacker then informs the user that they need to close the users account and transfer all the funds to a new account to prevent further fraudulent transactions. The user receives another authorization request which they approve, or provide additional authorization information to the attacker which enables the attacker to complete their attack and defraud the user.

Consent Request Overload



3.3.10. Example B9: Illicit Access to Administration Capabilities Through Consent Request Overload (Backchannel-Transferred Session Pattern)

An attacker attempts to access an adminstration portal repeatedly, generating a stream of authorization requests to the network administrator. The attempts are timed to occur while the administrator is asleep. The administrator is woken by the incoming requests on their phone, and, in an attempt to stop the notifications, they accidentally approve access and the attacker gains access to the portal.

What’s New: SHOULD, RECOMMENDED and MAY

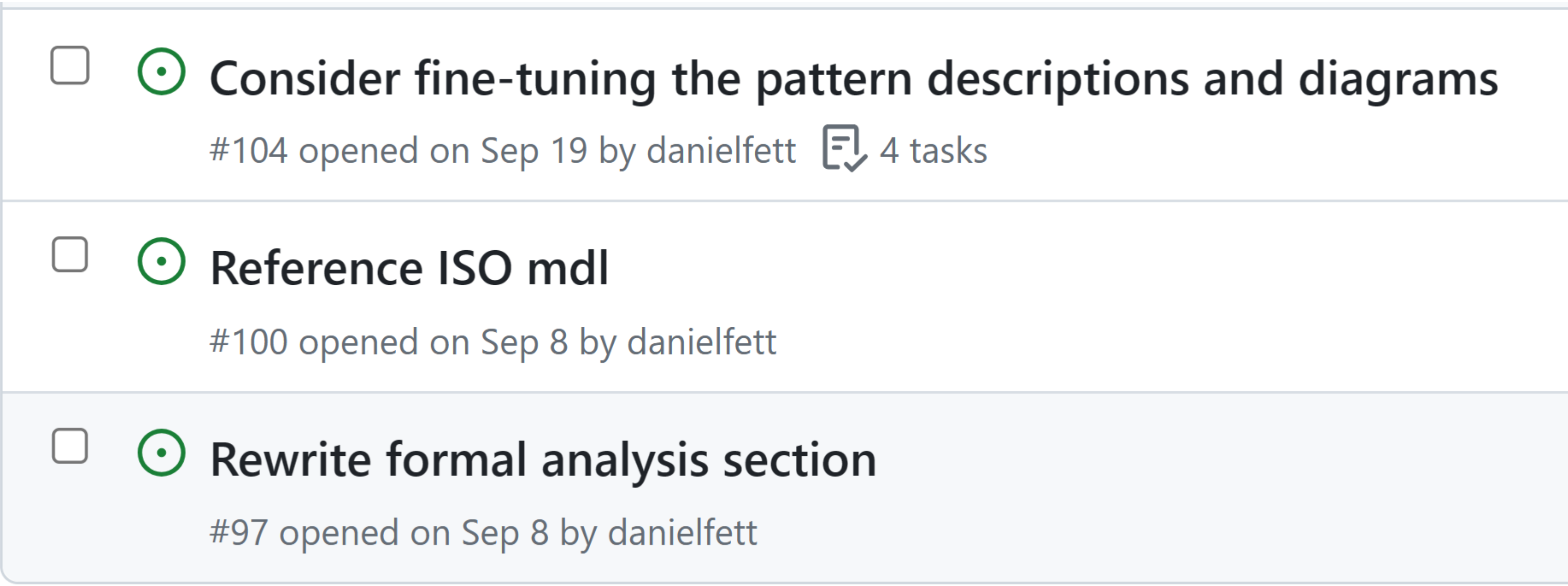
- Discussed at IETF 117
- Several “should, may, recommended”, no “SHOULD, MAY or RECOMMENDED”
    - Applies to the Authorization Server, Resource Server or Client
- Why change
    - Provide clear guidance to implementors
    - Emphasise importance of mitigations
    - Make conformance\adoption meaningful

What’s New: Editorial updates

- Editorial scrub
- Adopted the OpenID Foundation terminology from CIBA
- Acknowledgements
  - Marco Pernpruner
  - Giada Sciarretta
  - Maryam Mehrnezhad

Where do we go Next?

Open Issues



□ ⊙ **Consider fine-tuning the pattern descriptions and diagrams**

#104 opened on Sep 19 by danielfett 4 tasks

□ ⊙ **Reference ISO mdl**

#100 opened on Sep 8 by danielfett

□ ⊙ **Rewrite formal analysis section**

#97 opened on Sep 8 by danielfett

Next Steps

- Update Formal Analysis section (December)
- WG Last Call before IETF 119?

Questions?