



Northeastern  
University



UNIVERSITÀ  
DI TRENTO

# Look at the source: refine standard to harden OAuth security

Tommaso Innocenti

innocenti.t@northeastern.edu





Northeastern  
University



Tommaso Innocenti

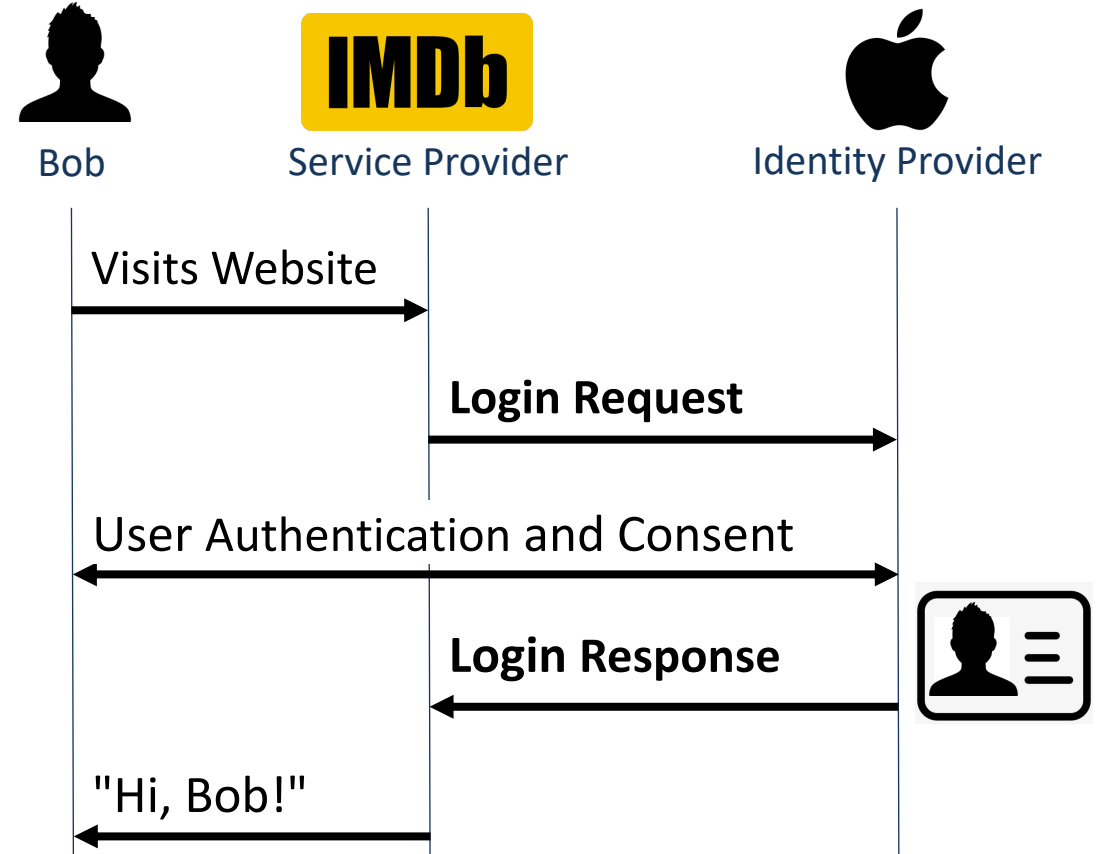
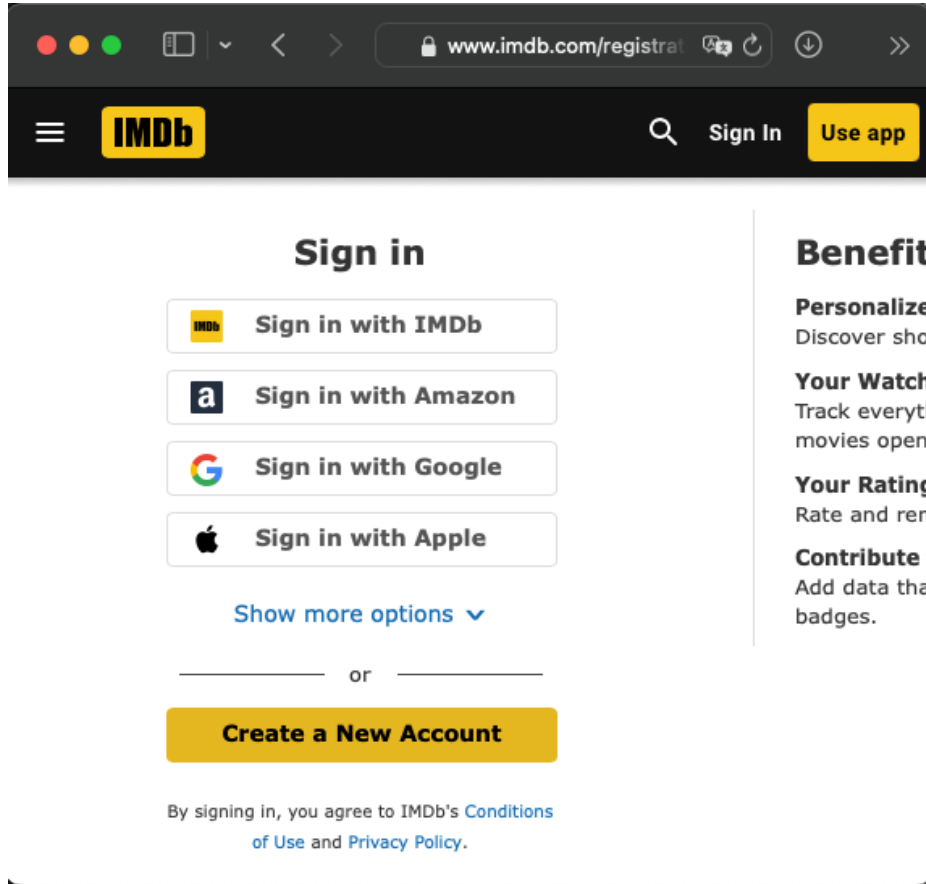
PhD candidate

@Northeastern University  
(Boston)

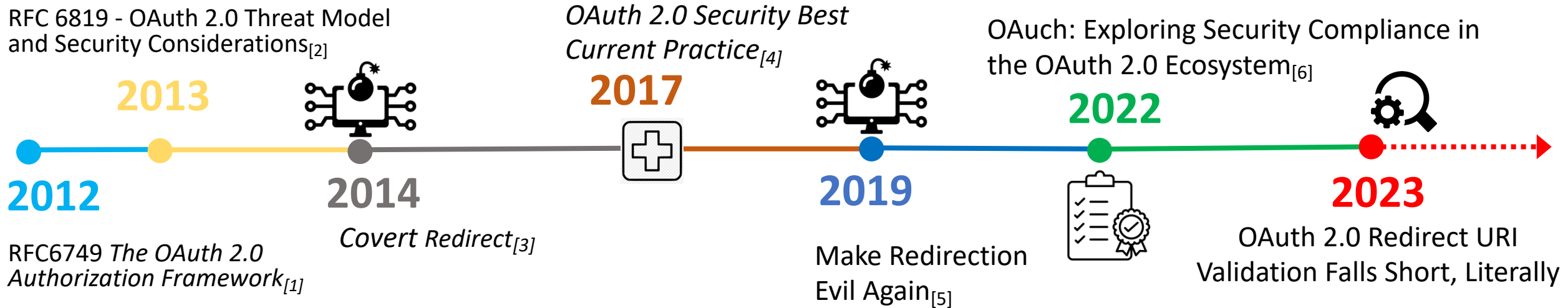
[innotommy.com](http://innotommy.com)



# What the heck is OAuth 2.0?



# ”redirect\_uri” attacks timeline



[1] D. Hardt. The oauth 2.0 authorization framework, October 2012. URL <https://tools.ietf.org/html/rfc6749>.

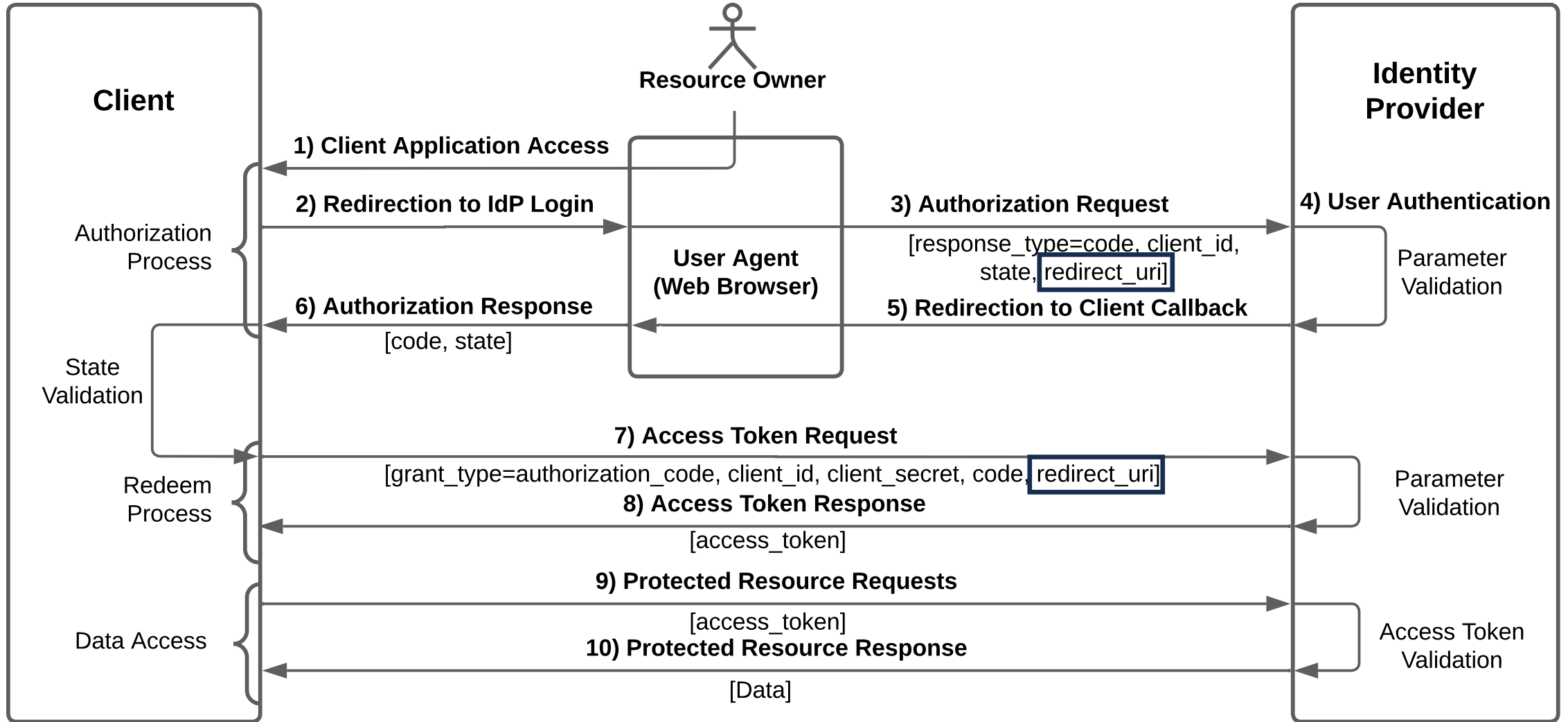
[2] T. Lodderstedt, M. McGloin, and P. Hunt. OAuth 2.0 Threat Model and Security Considerations. RFC 6819, IETF, January 2013. URL <http://tools.ietf.org/rfc/rfc6819.txt>.

[3] J. Wang, "Covert Redirect Vulnerability," 2014. [Online]. Available: [http://tetrapp.com/covert\\_redirect/](http://tetrapp.com/covert_redirect/).

[4] Torsten Lodderstedt, John Bradley, Andrey Labunets, and Daniel Fett. OAuth 2.0 security best current practice. Internet-Draft draft-ietf-oauth-security-topics-15,

[5] Wang, Xianbo, et al. "Make redirection evil again: Url parser issues in oauth." *BlackHat Asia 2019* (2019).

[6] Pieter Philippaerts, Davy Preuveneers, and Wouter Joosen. 2022. OAuch: Exploring Security Compliance in the OAuth 2.0 Ecosystem. In International Symposium on Research in Attacks, Intrusions and Defenses.



- Current research trend analysis

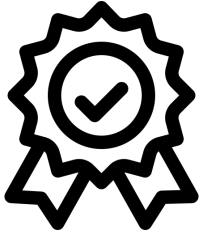
- Covert redirect



- Make Redirection Evil Again



External validation

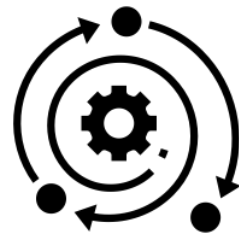


- Look at the "source" concept

- Expert validation



- impact



## *redirect\_uri* validation in RFC:

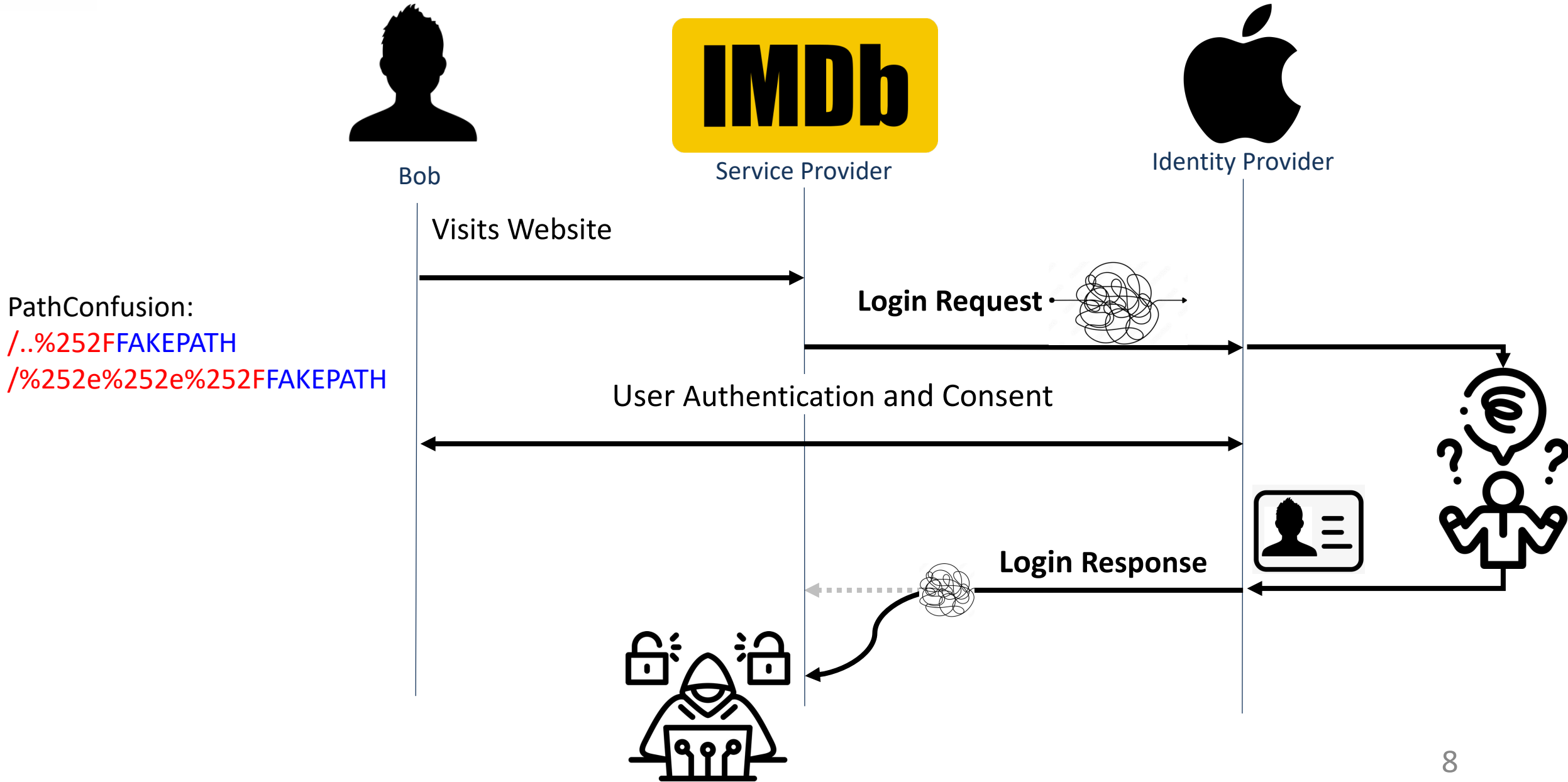
- **RFC 6749 Section 3.1.2.3**

The authorization server **MUST compare the two URIs using simple string comparison as defined in RFC 3986 Section 6.2.1.**

- **RFC 3986 Section 6.2.1**

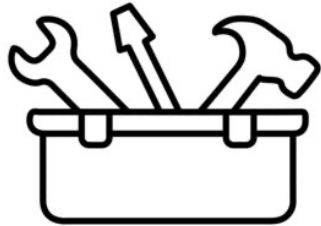
Testing strings for equality is normally based on **pair comparison of the characters that make up the strings, starting from the first and proceeding until both strings are exhausted, and all characters are found to be equal, until a pair of characters compares unequal, or until one of the strings is exhausted before the other.**

# What is Path Confusion?





- Find empirical validation



DISTINCT (CCS'22)<sub>[1]</sub>

Cerberus (CCS'22)<sub>[2]</sub>

SAAT (S&P'22)<sub>[3]</sub>

MoScan (ISSTA'21)<sub>[4]</sub>

OAuthScope (WPES'21)<sub>[5]</sub>

Manual login + In browser communications ❌

Static analysis of libraries ❌

Limited to only Facebook + session management ❌

State machine analysis ❌

Limit to 4 providers + user's data access ❌



**Modular, scalable and capable**

[1] Browser Communications in Dual-Window Single Sign-On." *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. 2022.

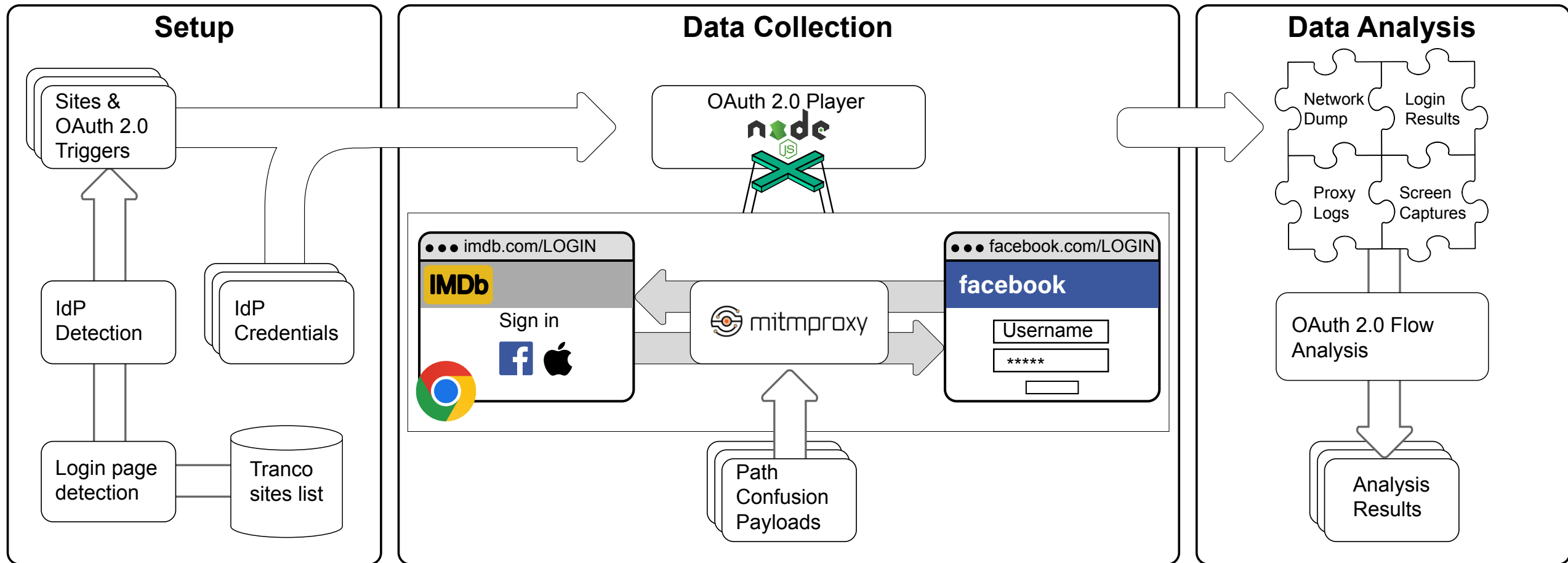
[2] Tamjid Al Rahat, Yu Feng, and Yuan Tian. Cerberus: Query-driven Scalable Security Checking for Oauth Service Provider Implementations, 2022. URL <http://arxiv.org/abs/2110.01005>.

[3] M. Ghasemisharif, C. Kanich, and J. Polakis. Towards Automated Auditing for Account and Session Management Flaws in Single Sign-On Deployments. In 2022 IEEE Symposium

[4] Hanlin Wei, Behnaz Hassanshahi, Guangdong Bai, Padmanabhan Krishnan, and Kostyantyn Vorobyov. MoScan: A Model-Based Vulnerability Scanner for Web Single Sign-On Services.

In Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis

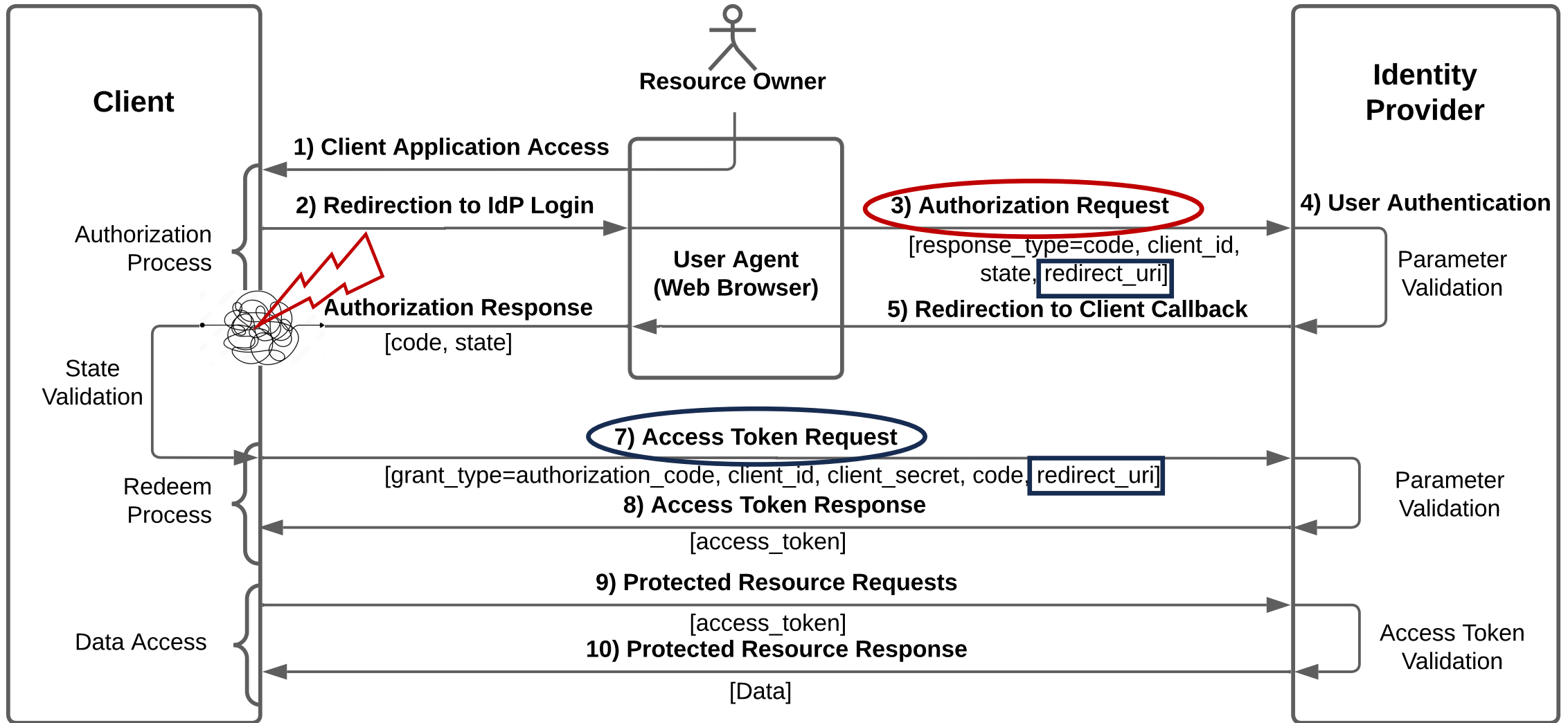
[5] Chaoshun Zuo, Qingchuan Zhao, and Zhiqiang Lin. AUTHSCOPE: Towards Automatic Discovery of Vulnerable Authorizations in Online Services. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security



**6/16 IdPs vulnerable to Path Confusion**  
(Facebook, Microsoft, GitHub, Atlassian, NAVER, and VK)



# Are we doomed?



## *redirect\_uri* validation in redeem step

- **RFC 6749 Section 4.1.3**

The Client makes a request to the token endpoint by sending the following parameters[...] **redirect\_uri REQUIRED, if the "redirect\_uri" parameter was included in the authorization request as described in Section 4.1.1, and their values MUST be identical.**

Will IdPs use the same vulnerable validation procedure?

2/16 IdPs are vulnerable (Naver, GitHub)



Full Victim's account takeover is possible!!!

- Path Confusion

Attack checklist:

- 1) Vulnerable *redirect\_uri* parsing in Authorization step → 6/16 IdPs ✓
- 2) Vulnerable Client → openbugbounty.com ✓
- 3) Vulnerable *redirect\_uri* check in redeem step → 2/16 IdPs ✓

Attack URL:

```
https://nid.naver.com/oauth2.0/authorize?client_id=<REDACTED>&response_type=c
ode&redirect_uri=https%3A%2F%2F<REDACTED>%2Fopenapi%2Fsocial%2Flogin.php
/%252e%252e/%252e%252e/%252e%252e/redirect.php%3Ftarget%3Dhttps%3a%2F
%2F<attacker-domain>%2F&state=random-state
```

Full Victim's account takeover is possible!!!

All IdPs involved in the study which has been found vulnerable has been contacted.

- Microsoft acknowledge our report and fixed their validation procedure.
- GitHub is tracking internally the problem and is actively working on a fix
- We are actively working with Naver to help fixing the issue

Reported our findings to the OAuth working group, which included our recommendation in the BCP.

OpenID foundation modified the conformance test suite to include our attack

Current “best practice” is not **good enough**

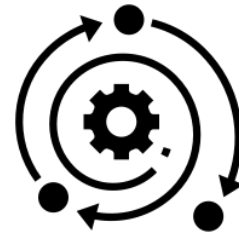
**Recommendations:**

- 1) ***redirect\_uri*** validation should use strict string equality check
- 2) IdPs server should never sanitize ***redirect\_uri*** to avoid introducing any discrepancy, instead should validate them



# Our concept takes time, effort and doesn't bring money

- February 2023 first contact with OAuth WG → August 2023 OSW → November 2023 BCP Acknowledged ✓  
 ↙  
 OpenID conformance test suite updated ✓



Follow the current trend or go back at the source?

Questions?