Northeastern University

UNIVERSITÀ DI TRENTO

OAuth 2.0 Redirect URI Validation Falls Short, Literally

Tommaso Innocenti, Matteo Golinelli, Kaan Onarlioglu,
Ali Mirheidari, Bruno Crispo, and Engin Kirda

innocenti.t@northeastern.edu

Northeastern University

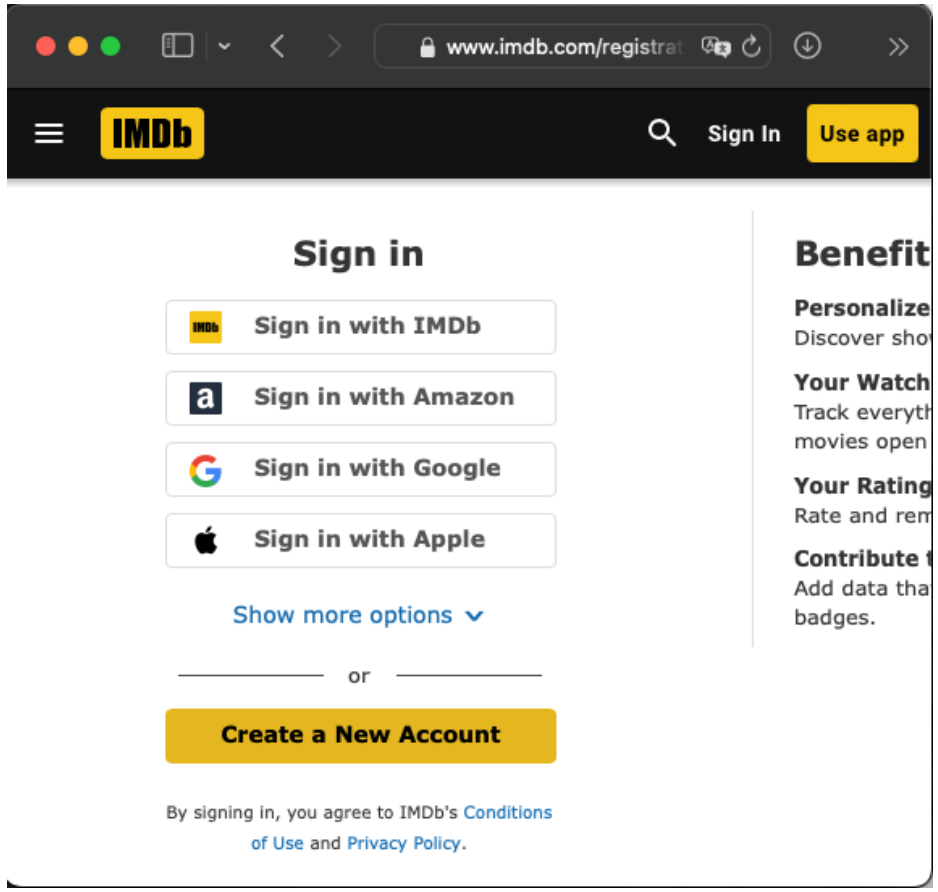
Tommaso Innocenti

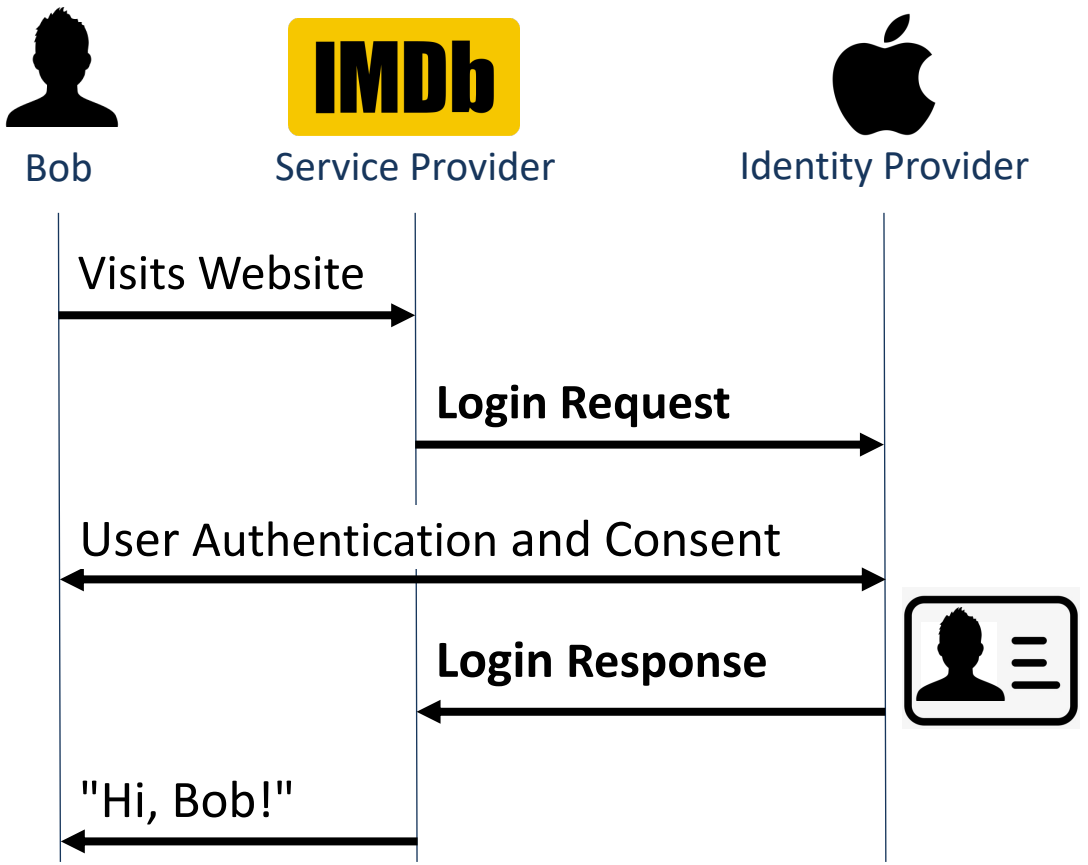
PhD candidate

@Northeastern University (Boston)

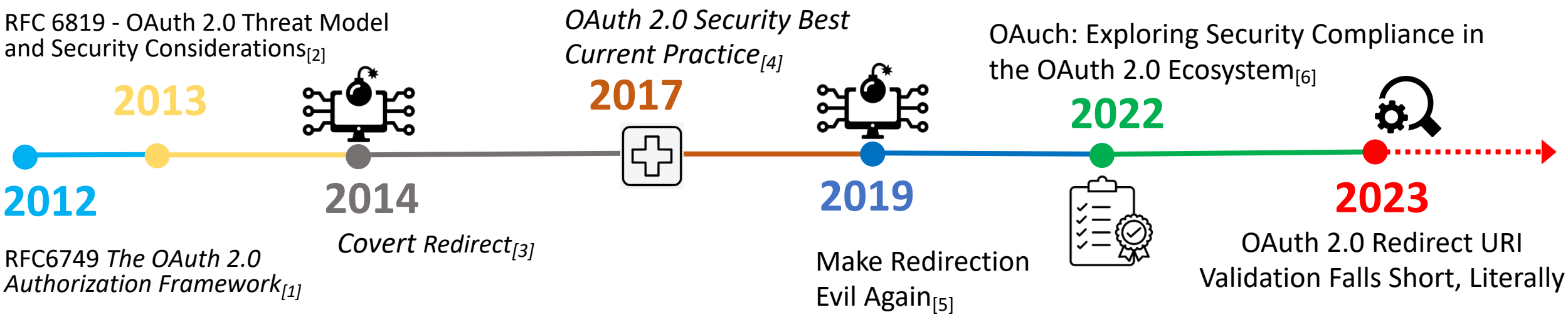
innotommy.com

What the heck is OAuth 2.0?



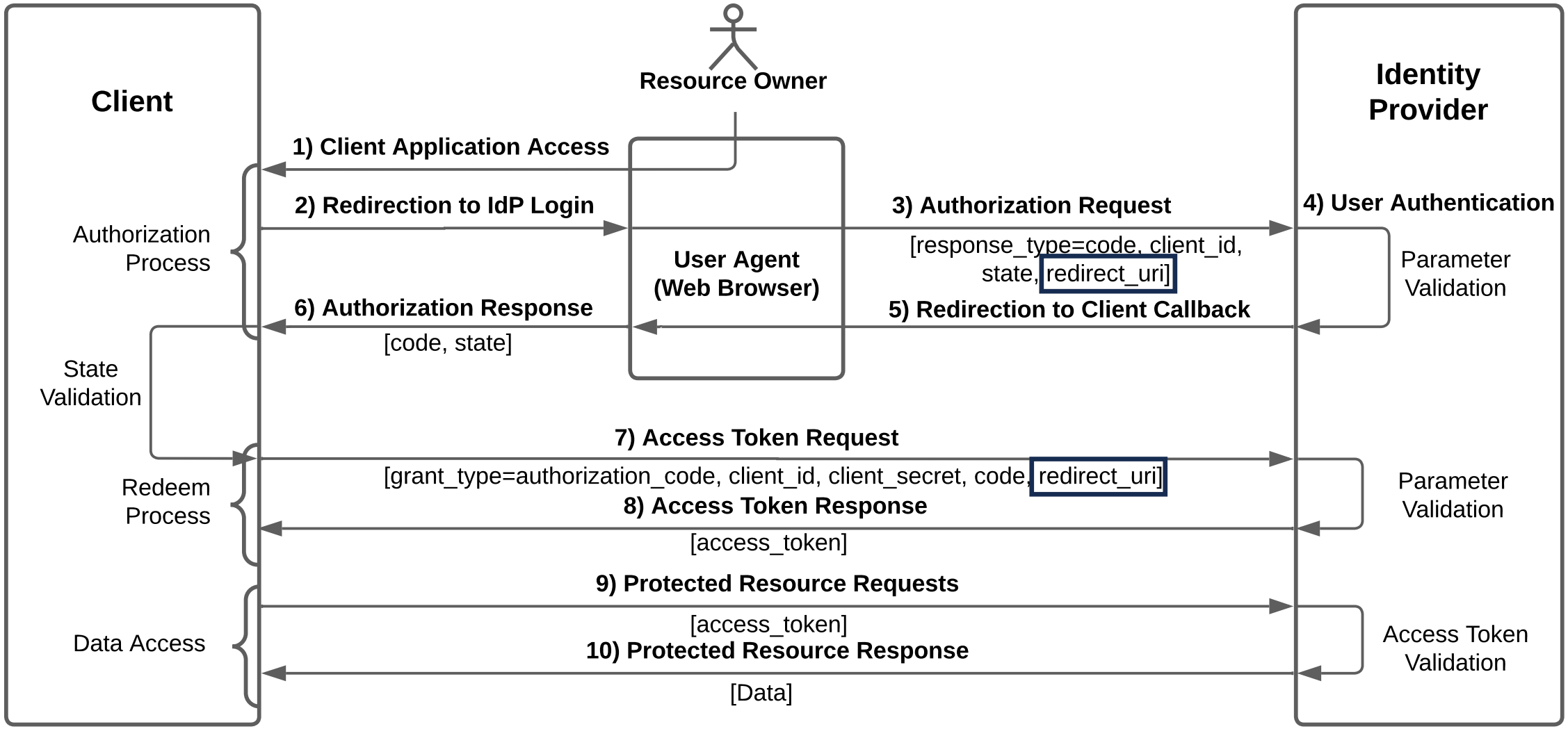


”redirect_uri” attacks timeline



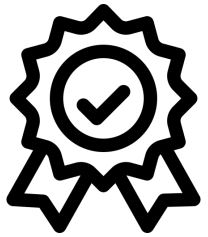
[1] D. Hardt. The oauth 2.0 authorization framework,October 2012. URL https://tools.ietf.org/html/ rfc6749.
[2] T. Lodderstedt, M. McGloin, and P. Hunt. Oauth 2.0 Threat Model and Security Considerations. RFC 6819, IETF, January 2013. URL http://tools.ietf.org/rfc/rfc6819.txt.
[3] J. Wang, “Covert Redirect Vulnerability,” 2014. [Online]. Available: http://tetraph.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: Explor- ing Security Compliance in the OAuth 2.0 Ecosystem. In International Symposium on Research in Attacks,Intrusions and Defenses.

Code grant flow



- Current research trend analysis
  - Covert redirect 

  - Make Redirection Evil Again 

External validation 

- Look at the ’’source’’ concept
  - Expert validation 
  - impact 

RFC ***redirect_uri***
validation issue

-XSS style
-HTML injection
-Open redirect
-OAuth token Leakage

Full victim’s account takeover

redirect_uri validation in RFC:

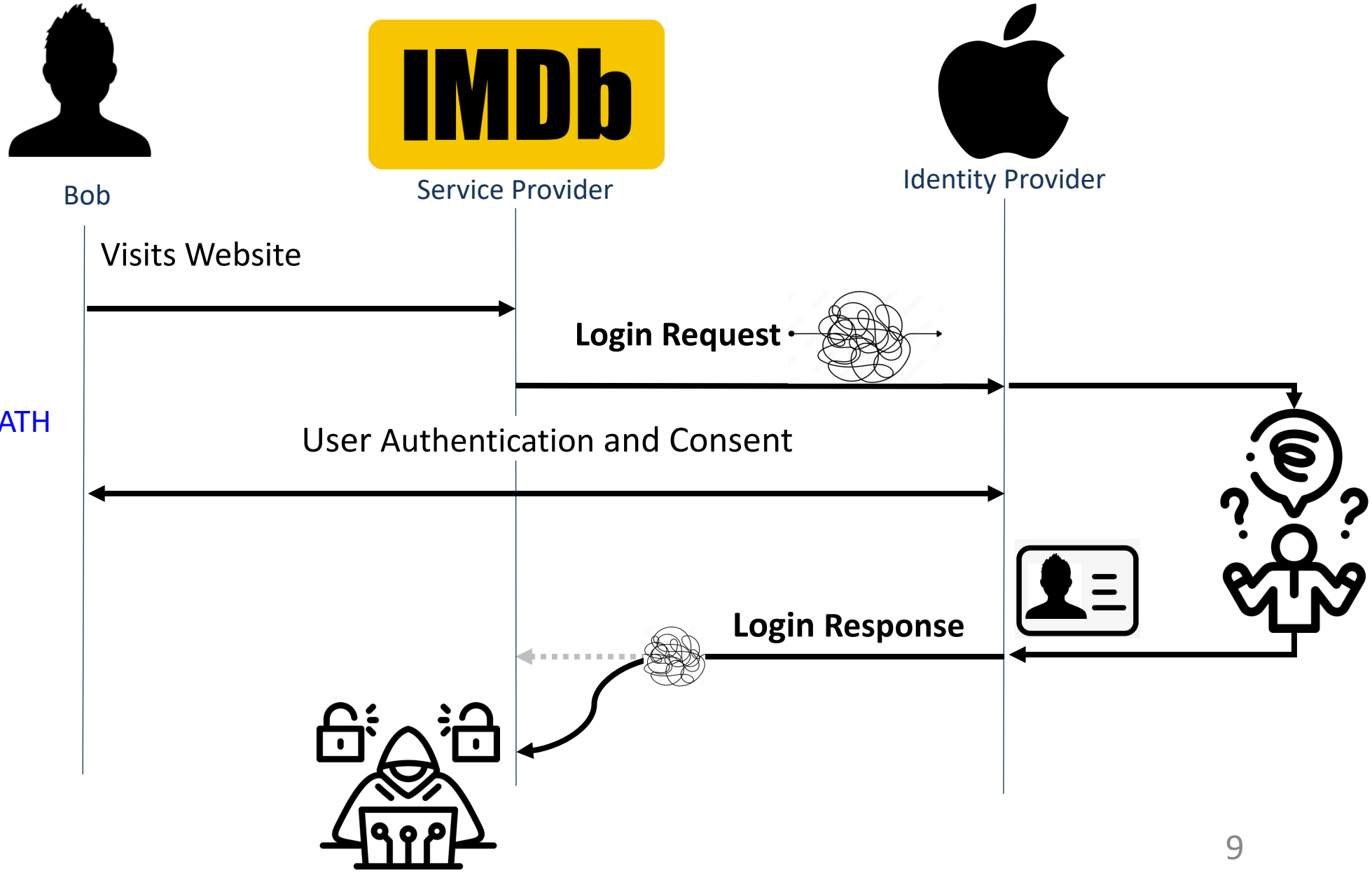
- **RFC 6749 Section 3.1.2.3**

  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**

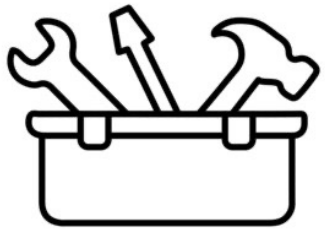
  Testing strings for equality is normally based on pair comparison of the characters that make up the strings, starting from the first and proceeding until both strings are exhausted, and all characters are found to be equal, until a pair of characters compares unequal, or until one of the strings is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

What is Path Confusion?



PathConfusion:
/..%252FFAKEPATH
/%252e%252e%252FFAKEPATH

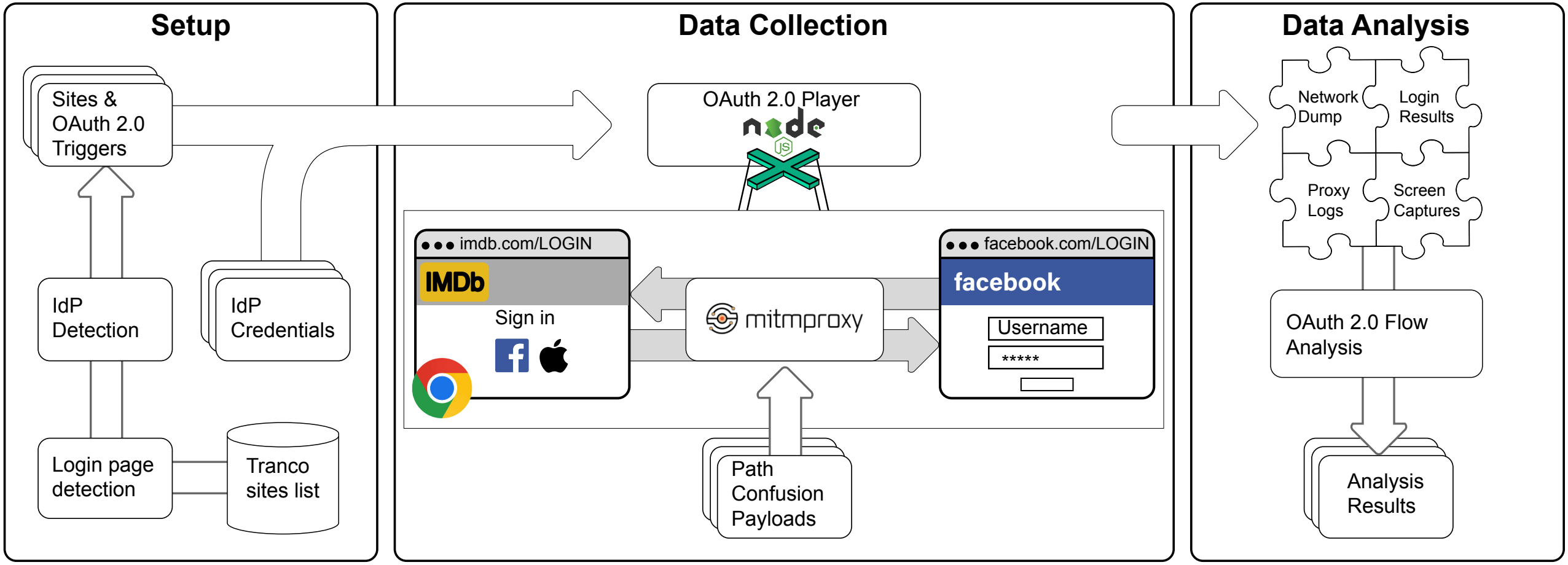
- Find empirical validation



DISTINCT (CCS'22)[1]	Manual login + In browser communications ✗
Cerberus (CCS'22)[2]	Static analysis of libraries ✗
SAAT (S&P'22)[3]	Limited to only Facebook + session management ✗
MoScan (ISSTA'21)[4]	State machine analysis ✗
OAuthScope (WPES'21)[5]	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



https://github.com/innotommy/OAuthpaper-code

6/16 IdPs vulnerable to Path Confusion

(Facebook, Microsoft, GitHub, Atlassian, NAVER, and VK)

redirect_uri parameter in RFC:

- **RFC 6749 Section 3.1**

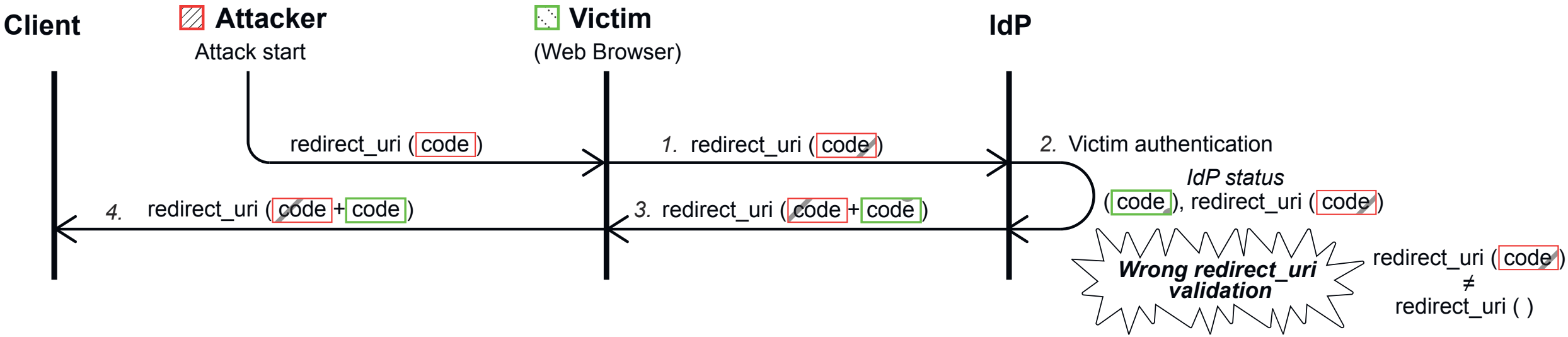
  The endpoint URI MAY include an "application/x-www-form-urlencoded" formatted (per Appendix B) query component (RFC 3986 Section 3.4), which MUST be retained when adding additional query parameters.
- **RFC 6749 Section 10.14**

  A code injection attack occurs when an input or otherwise external variable is used by an application unsanitized and causes modification to the application logic. This may allow an attacker to access the application device or its data, cause a denial of service, or introduce a wide range of malicious side effects. The authorization server and Client MUST sanitize (and validate when possible) any value received–in particular, the value of the "state" and "redirect_uri" parameters.

Lack on input validation directive or attack prevention

- Attack URL:

  https://idp.example.com/oauth/authorize?response_type=code&client_id=<validID>&state=<value>&redirect_uri=https://Client.example.com/oauth/callback%3Fcode%3D<value>

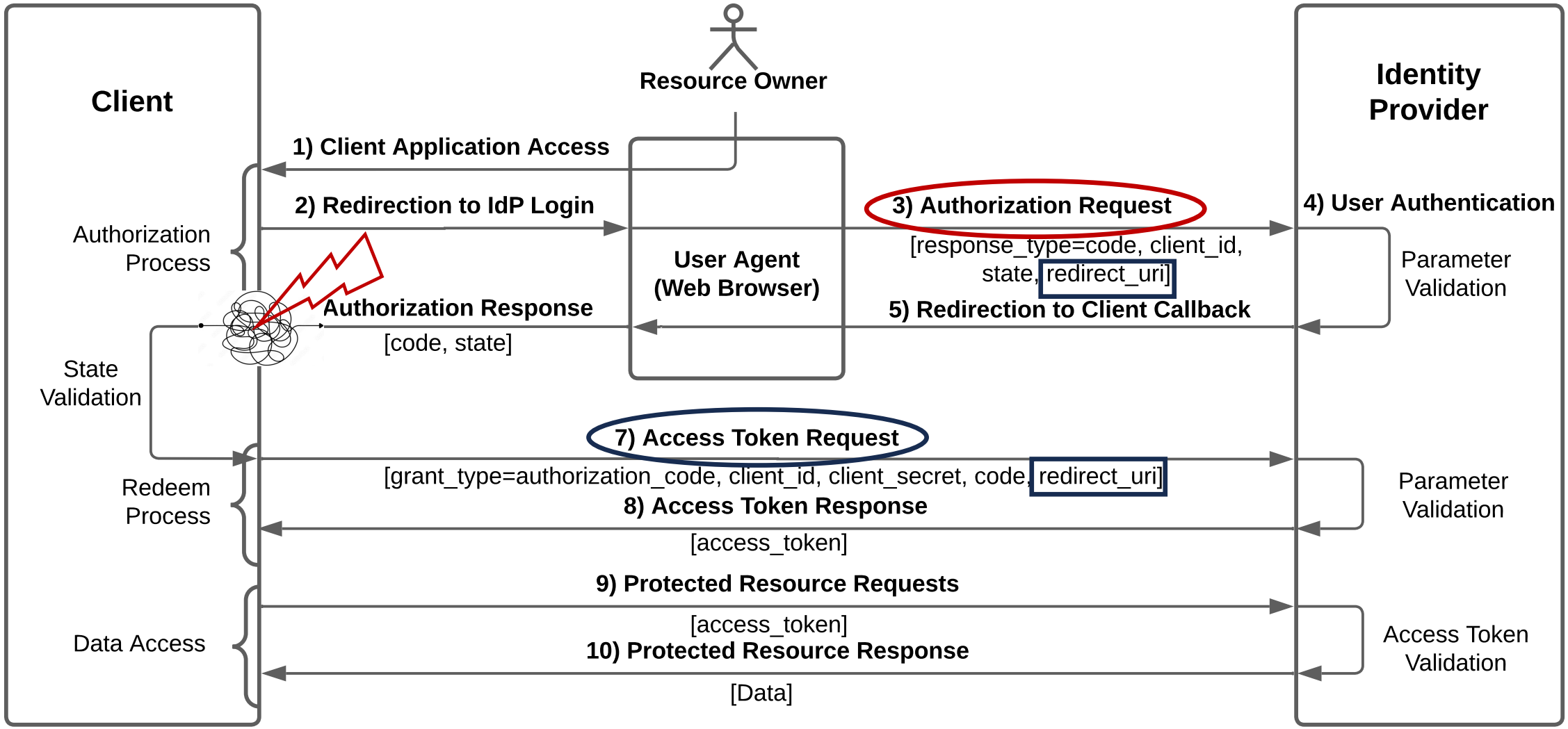


Victim's authenticated as the attacker!!

10/16 IdPs vulnerable to OAuth Parameter Pollution

(LinkedIn, Microsoft, GitHub, Slack, VK, OK, NAVER, Yahoo, ORCID and LINE)

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
- Oauth Parameter Pollution
- Redirect_uri validation in redeem step

-XSS style
-HTML injection
-Open redirect
-OAuth token Leakage

Full victim’s account takeover

- 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=code&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

3) IdPs should validate **redirect_uri** and block Authorization request where ***Code*** or ***state*** parameters are included in the ***redirect_uri*** as parameter.

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 ✓

Questions?