

# WPSE: FORTIFYING WEB PROTOCOLS VIA BROWSER-SIDE SECURITY MONITORING

Stefano Calzavara

Riccardo Focardi

Mauro Tempesta

Marco Squarcina

Matteo Maffei

Clara Schneidewind

August 17, 2018 - 27<sup>th</sup> Usenix Security Symposium

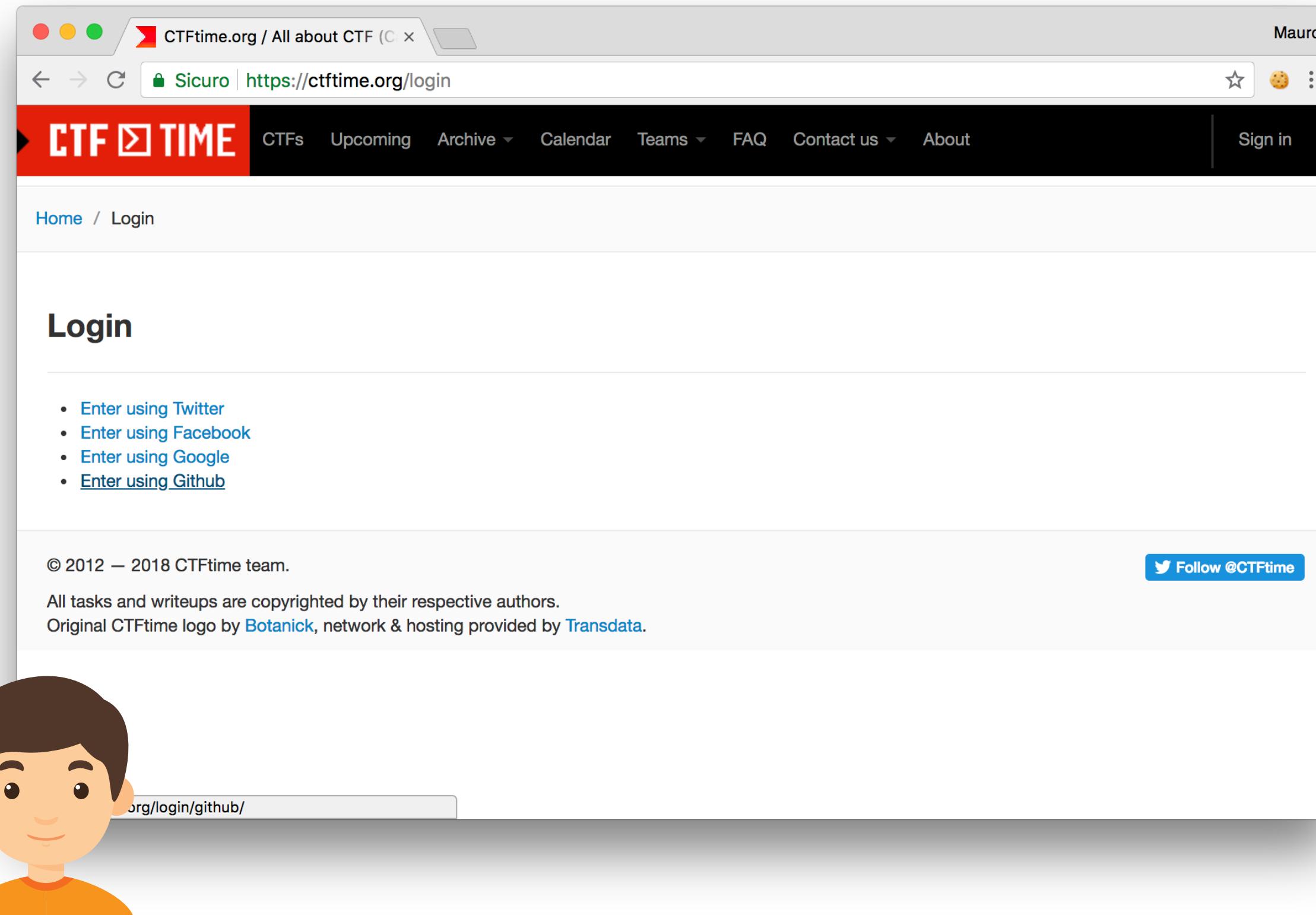


Università  
Ca' Foscari  
Venezia



TECHNISCHE  
UNIVERSITÄT  
WIEN

# OVERVIEW OF A WEB PROTOCOL



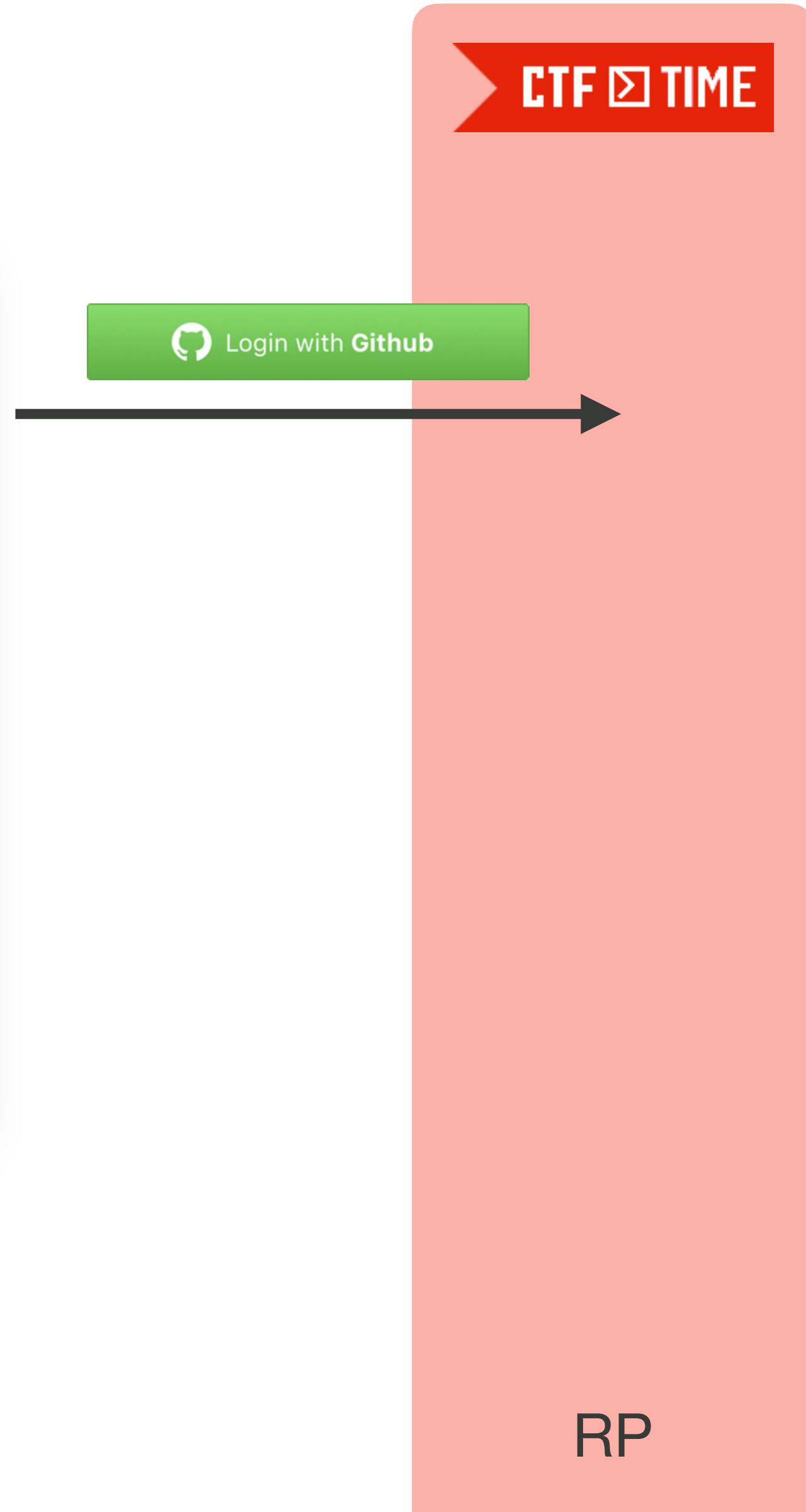
A screenshot of a web browser displaying the CTFtime.org login page. The page has a red header with the logo "CTF TIME". Below the header, there is a navigation bar with links for "Home", "CTFs", "Upcoming", "Archive", "Calendar", "Teams", "FAQ", "Contact us", and "About". On the right side of the header, there is a "Sign in" button. The main content area is titled "Login" and contains a list of social media login options: "Enter using Twitter", "Enter using Facebook", "Enter using Google", and "Enter using Github". At the bottom of the page, there is a copyright notice: "© 2012 – 2018 CTFtime team. All tasks and writeups are copyrighted by their respective authors. Original CTFtime logo by Botanick, network & hosting provided by Transdata." There is also a "Follow @CTFtime" button and a small profile picture of a person.



IdP

# OVERVIEW OF A WEB PROTOCOL

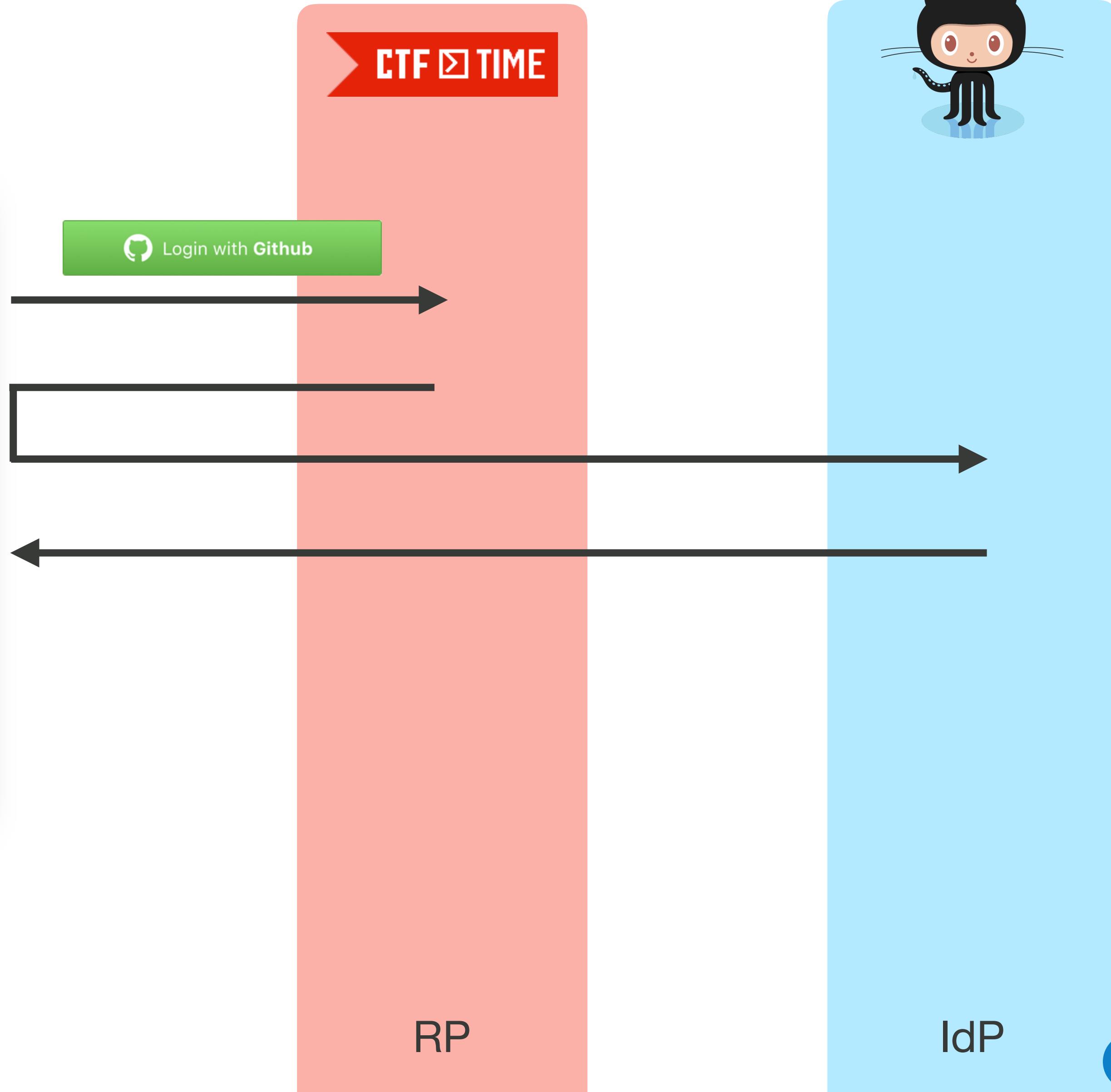
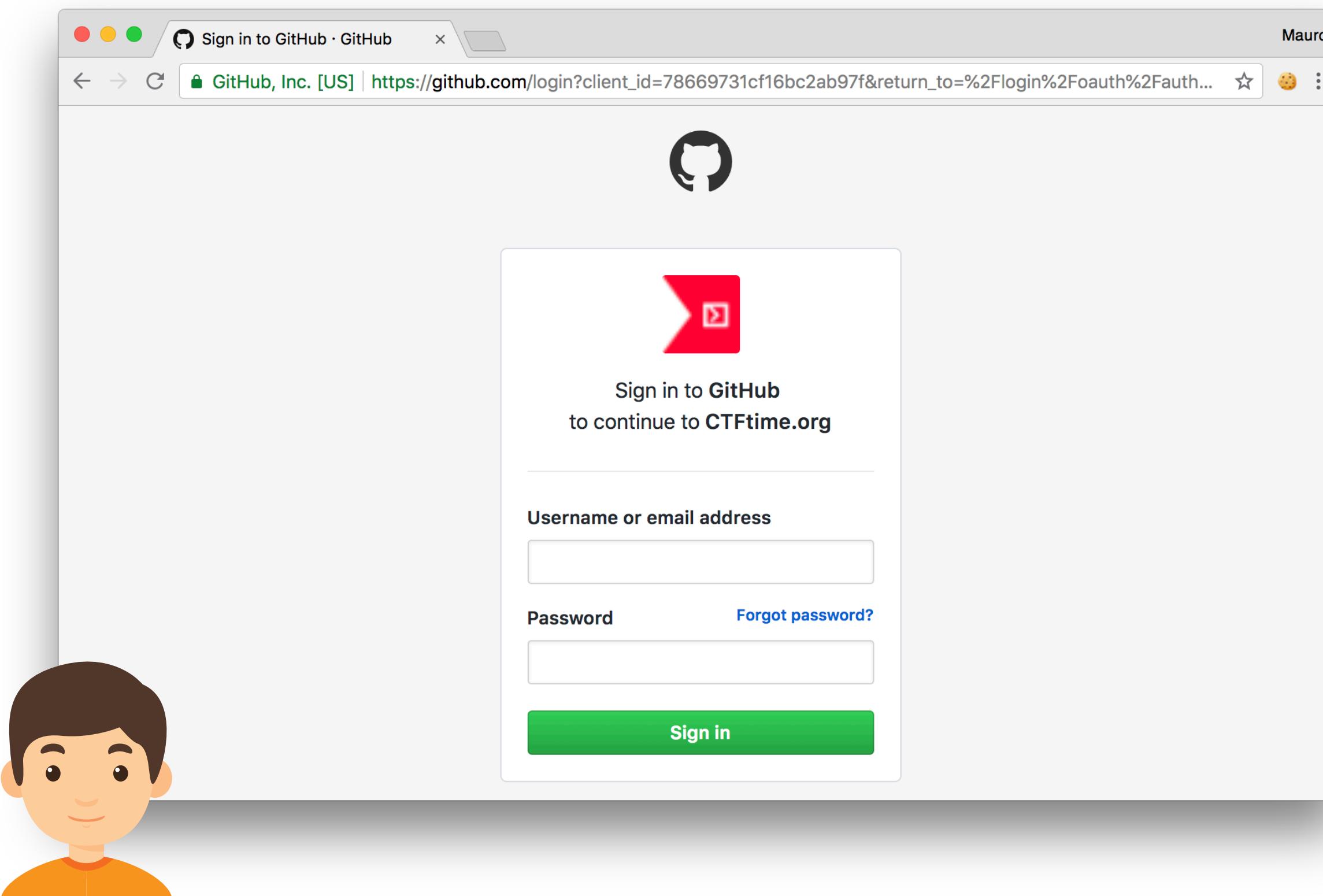
A screenshot of a web browser displaying the CTFtime.org login page. The page has a red header with the logo and navigation links: CTF TIME, CTFs, Upcoming, Archive, Calendar, Teams, FAQ, Contact us, and About. On the right, there is a "Sign in" button. Below the header, the URL https://ctftime.org/login is shown in the address bar. The main content area is titled "Login" and contains a list of social login options: Enter using Twitter, Enter using Facebook, Enter using Google, and Enter using Github. At the bottom, there is a copyright notice for 2012–2018 CTFtime team, a link to follow @CTFtime on Twitter, and credits for original logo by Botanick and network & hosting by Transdata. A placeholder image of a person's head is visible at the bottom left.



IdP

RP

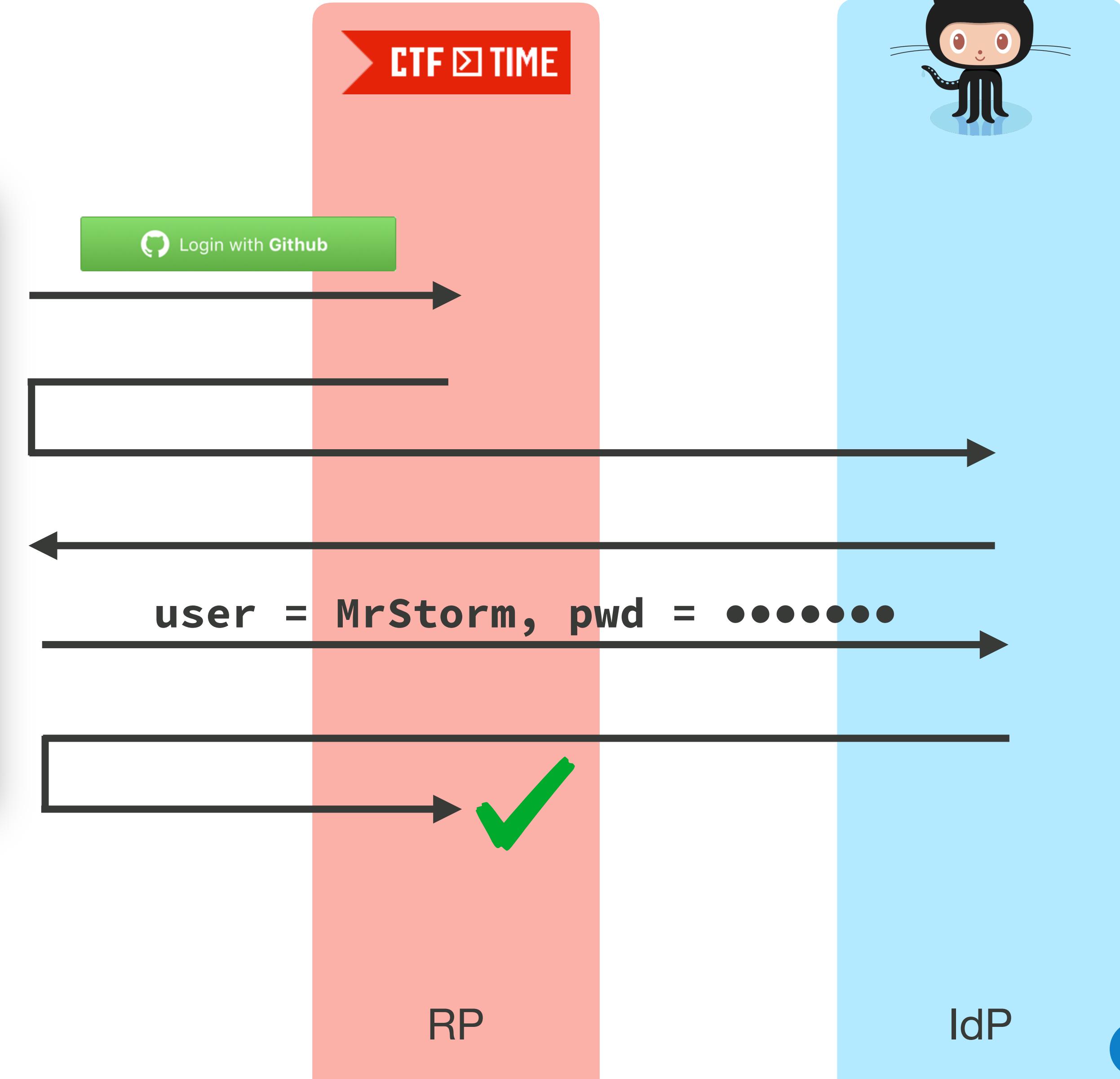
# OVERVIEW OF A WEB PROTOCOL



# OVERVIEW OF A WEB PROTOCOL

The screenshot shows the homepage of CTFtime.org. At the top, there's a navigation bar with links for CTFs, Upcoming, Archive, Calendar, Teams, FAQ, Contact us, About, Timezone: UTC (set to UTC), and MrStorm (logged in). Below the navigation is a section titled "Team rating" with a table showing the top teams from 2018. The table includes columns for Place, Team, Country, and Rating. The top teams listed are Dragon Sector (704,084), Plaid Parliament of Pwning (583,211), dcua (510,212), TokyoWesterns (485,811), p4 (461,567), LCBC (435,130), CyKOR (431,042), and Bushwhackers (411,255). To the right, there are sections for "Now running" (TJCTF 2018) and "Past events" (Real World CTF 2018 Quals). A sidebar on the left features a cartoon illustration of a person.

Place	Team	Country	Rating
1	Dragon Sector	PL	704,084
2	Plaid Parliament of Pwning	US	583,211
3	dcua	UA	510,212
4	TokyoWesterns	JP	485,811
5	p4	PL	461,567
6	LCBC	RU	435,130
	CyKOR	KR	431,042
	Bushwhackers	RU	411,255



# MOTIVATIONS

Designing and implementing web protocols is **HARD!**

- *Bansal et al.* - Discovering Concrete Attacks on Website Authorization by Formal Analysis (**S&P '12**)
- *Wang et al.* - Signing Me onto Your Accounts through Facebook and Google: A Traffic-Guided Security Study of Commercially Deployed Single-Sign-On Web Services (**S&P '12**)
- *Sun and Beznosov* - The Devil is in the (Implementation) Details: An Empirical Analysis of OAuth SSO Systems (**CCS '12**)
- *Fett et al.* - A Comprehensive Formal Security Analysis of OAuth 2.0 (**CCS '16**)
- ...

# MOTIVATIONS

Designing and implementing web protocols is **HARD!**

- *Bansal et al.* - Discovering Concrete Attacks on Website Authorization by Formal Analysis (**S&P '12**)
- *Wang et al.* - Signing Me onto Your Accounts through Facebook and Google: A Traffic-Guided Security Study of Commercially Deployed Single-Sign-On Web Services (**S&P '12**)
- *Sun and Beznosov* - The Devil is in the (Implementation) Details: An Empirical Analysis of OAuth SSO Systems (**CCS '12**)
- *Fett et al.* - A Comprehensive Formal Security Analysis of OAuth 2.0 (**CCS '16**)
- ...

## WHY?

# MOTIVATIONS

Designing and implementing web protocols is **HARD!**

- *Bansal et al.* - Discovering Concrete Attacks on Website Authorization by Formal Analysis (**S&P '12**)
- *Wang et al.* - Signing Me onto Your Accounts through Facebook and Google: A Traffic-Guided Security Study of Commercially Deployed Single-Sign-On Web Services (**S&P '12**)
- *Sun and Beznosov* - The Devil is in the (Implementation) Details: An Empirical Analysis of OAuth SSO Systems (**CCS '12**)
- *Fett et al.* - A Comprehensive Formal Security Analysis of OAuth 2.0 (**CCS '16**)
- ...

The browser is not aware of the existence of web protocols and of their semantics!

# OUR PROPOSAL - WPSE

Extend the browser with a **lightweight security monitor** that enforces the compliance of the browser behaviors with respect to the web protocol specifications

# OUR PROPOSAL - WPSE

Extend the browser with a **lightweight security module** that enforces the compliance of the browser behaviors with respect to the web protocol specifications



Implemented as a  
Google Chrome **extension**

# OUR PROPOSAL - WPSE

Extend the browser with a **lightweight security module** ...  
that enforces the compliance of the browser behaviors  
with respect to the web protocol specifications

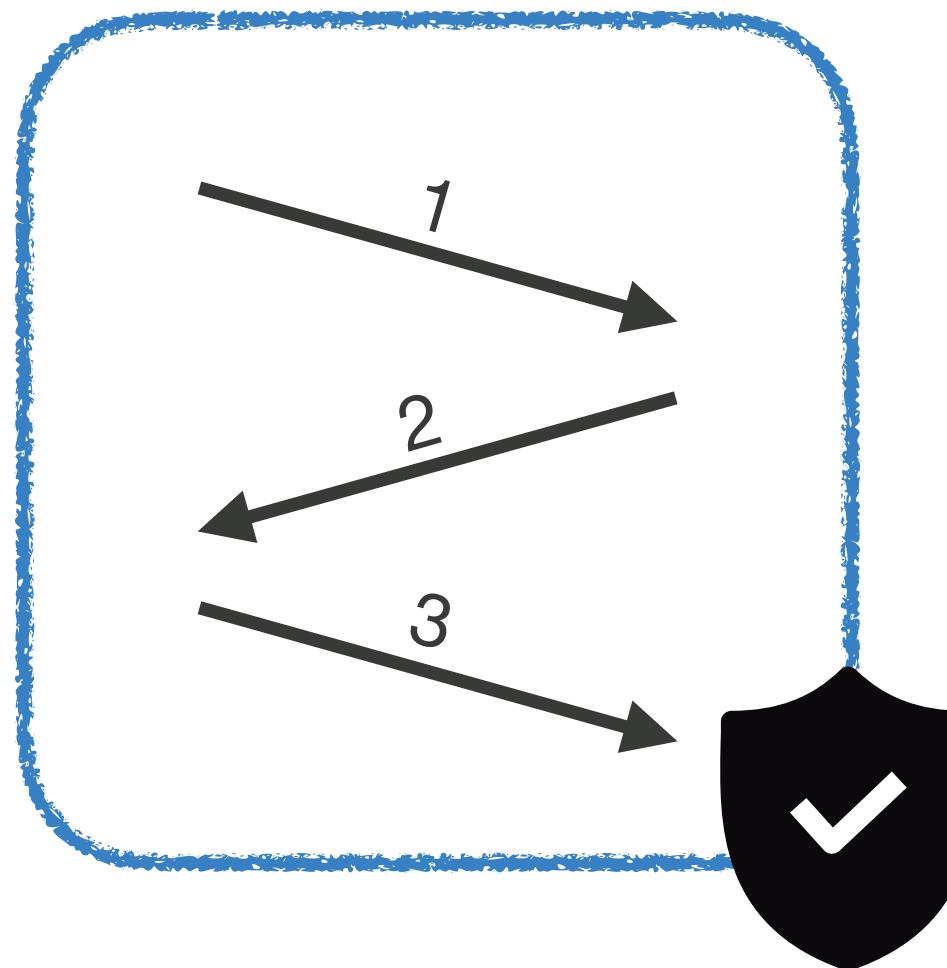


Implemented as a  
Google Chrome **extension**

## Advantages:

1. users of vulnerable websites are **automatically protected** against a large class of attacks
2. specifications can be written **once** and enforced on **several sites**

# SECURITY CHALLENGES IN WEB PROTOCOLS



## Compliance with the protocol flow

- Preserve the intended sequence of messages exchanged by honest participants
- Perform integrity checks on the contents of protocol messages



## Secrecy of message components

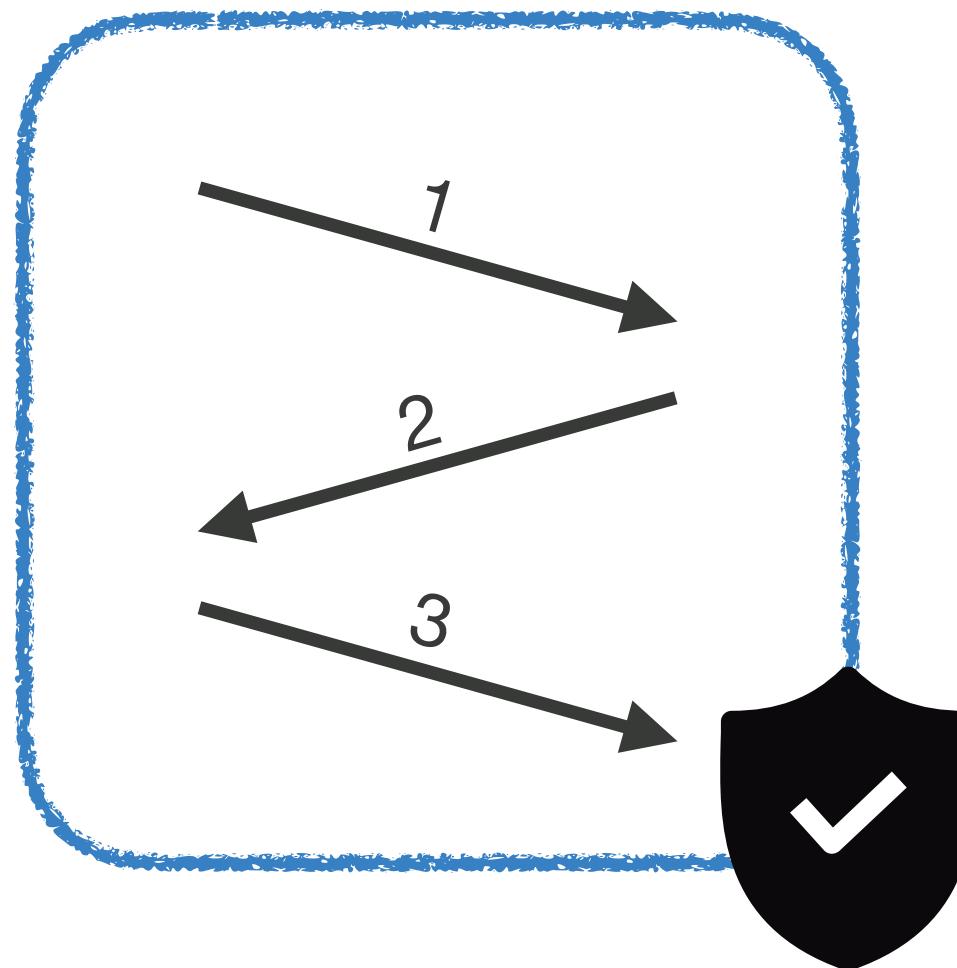
- Enforce the confidentiality of protocol secrets like tokens and credentials

# TACKLING THE CHALLENGES IN WPSE

## **WPSE protocol specification:**

- Structure and order of messages
- Desired security policies (confidentiality and integrity)

# TACKLING THE CHALLENGES IN WPSE

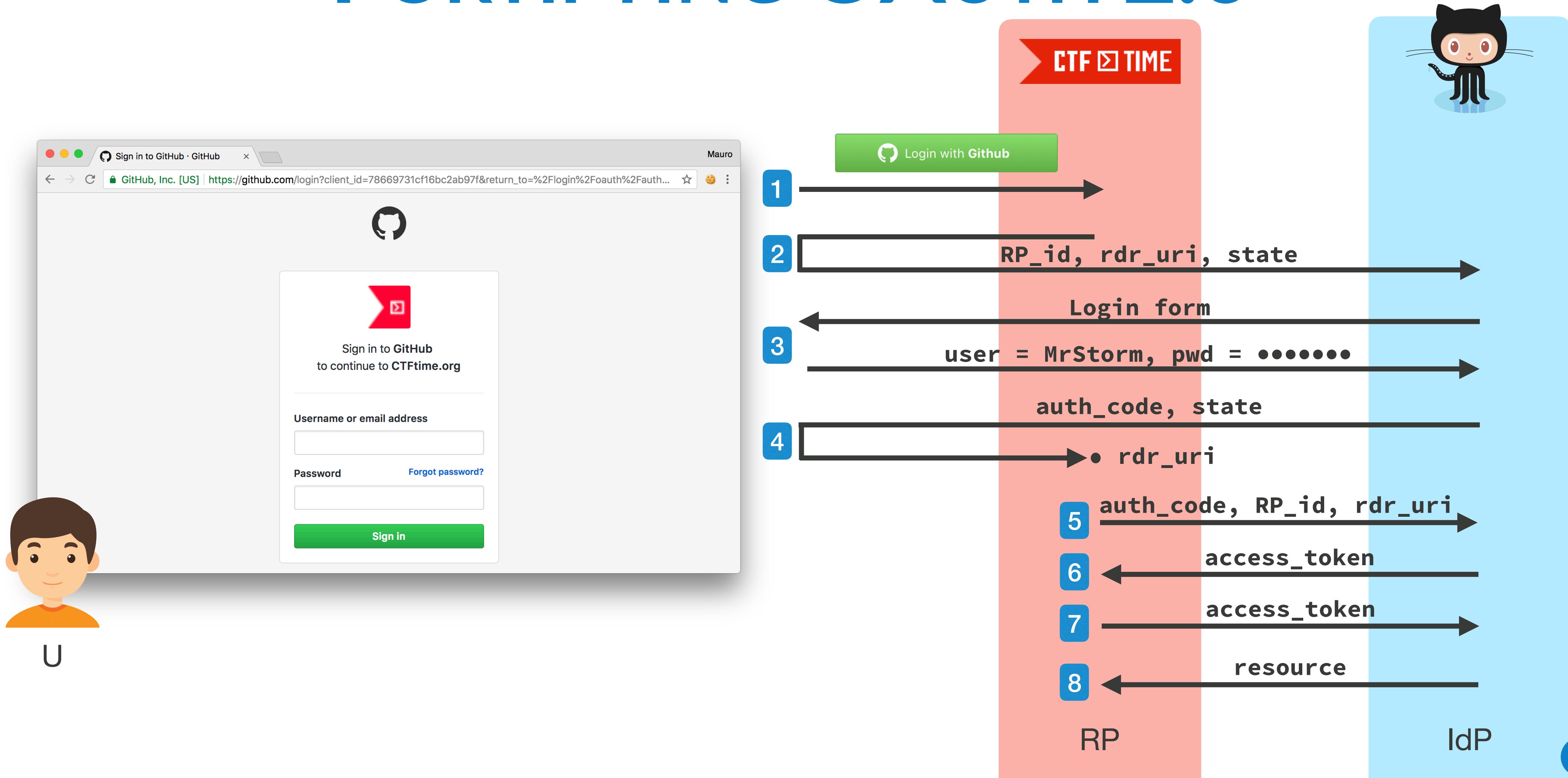


- Protocol messages are **blocked** if
  - not in the correct order
  - integrity constraints on messages are not satisfied
- Always **allow** protocol unrelated messages

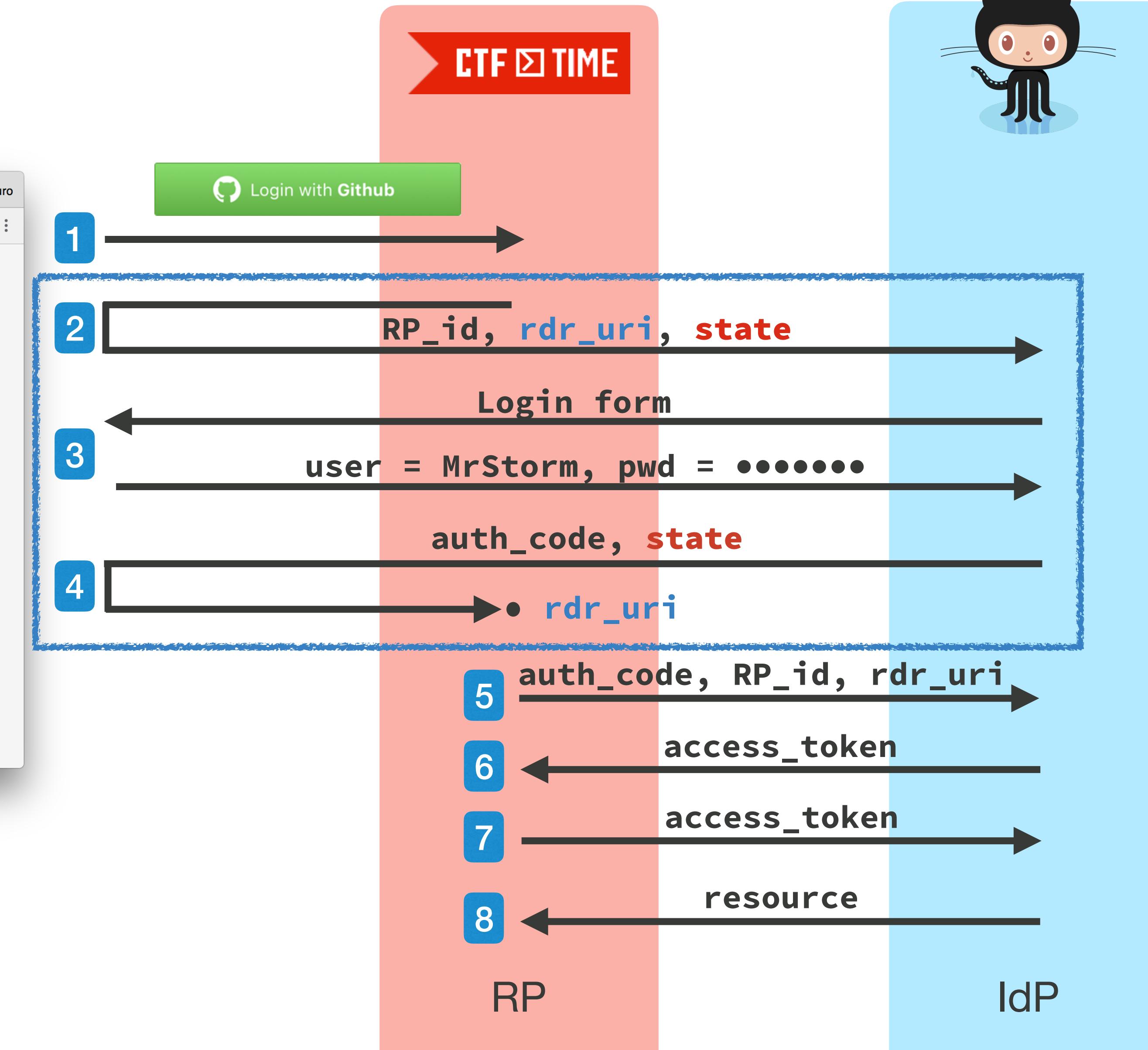
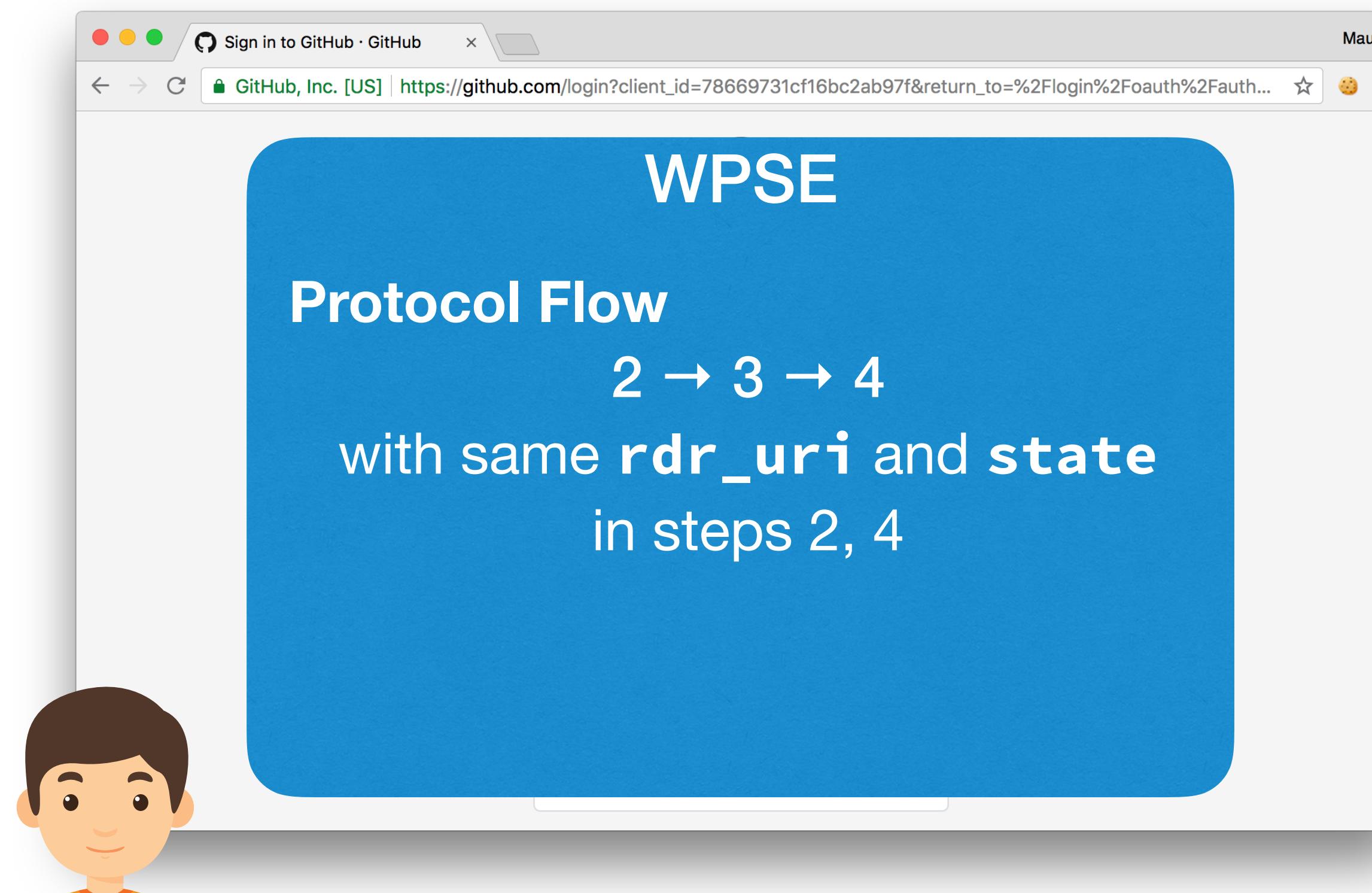


- Secrets in incoming messages are substituted with **random placeholders** before they enter the DOM
- Placeholders in outgoing requests are replaced with secrets **only** if sent to origins entitled to learn them

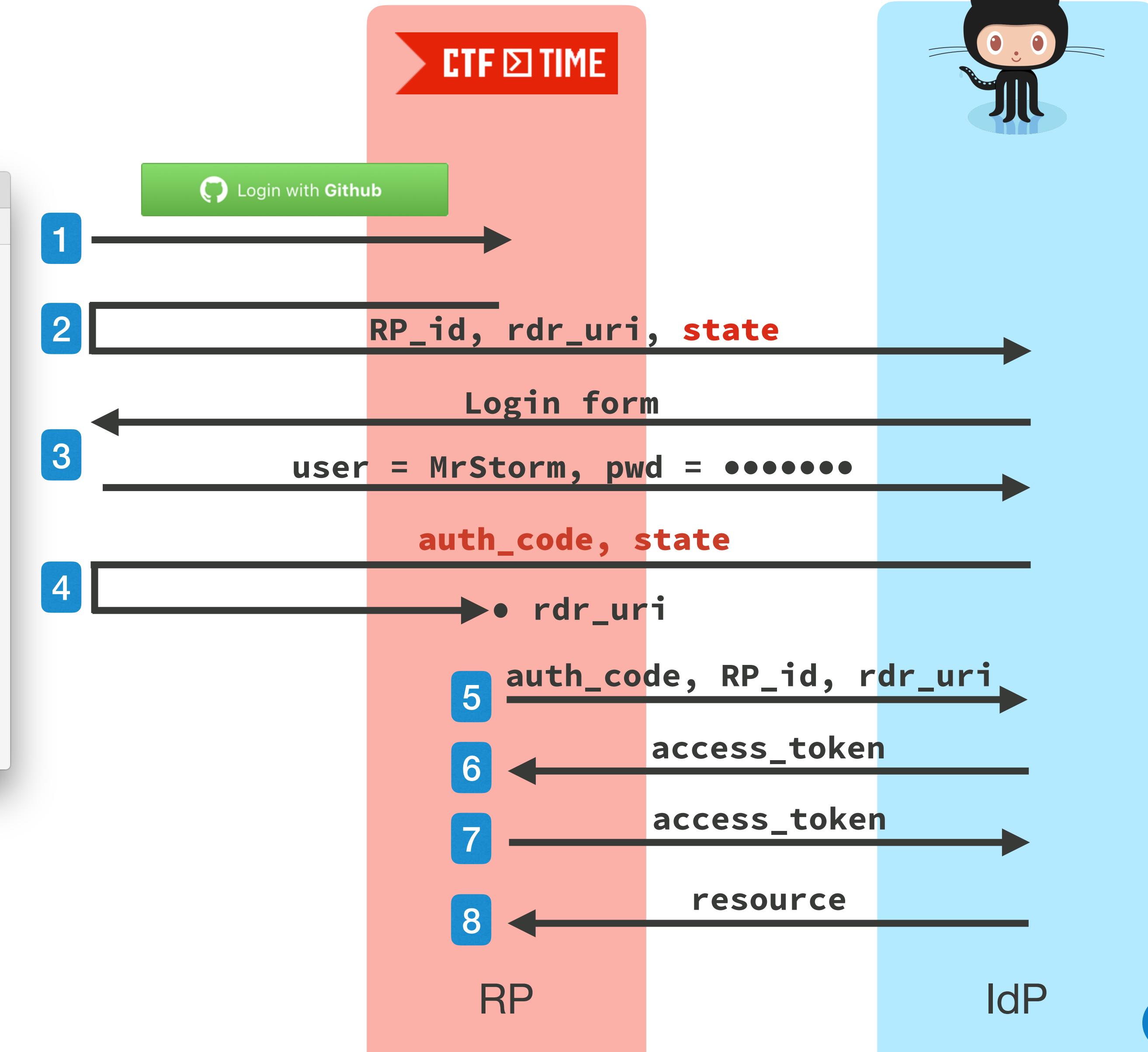
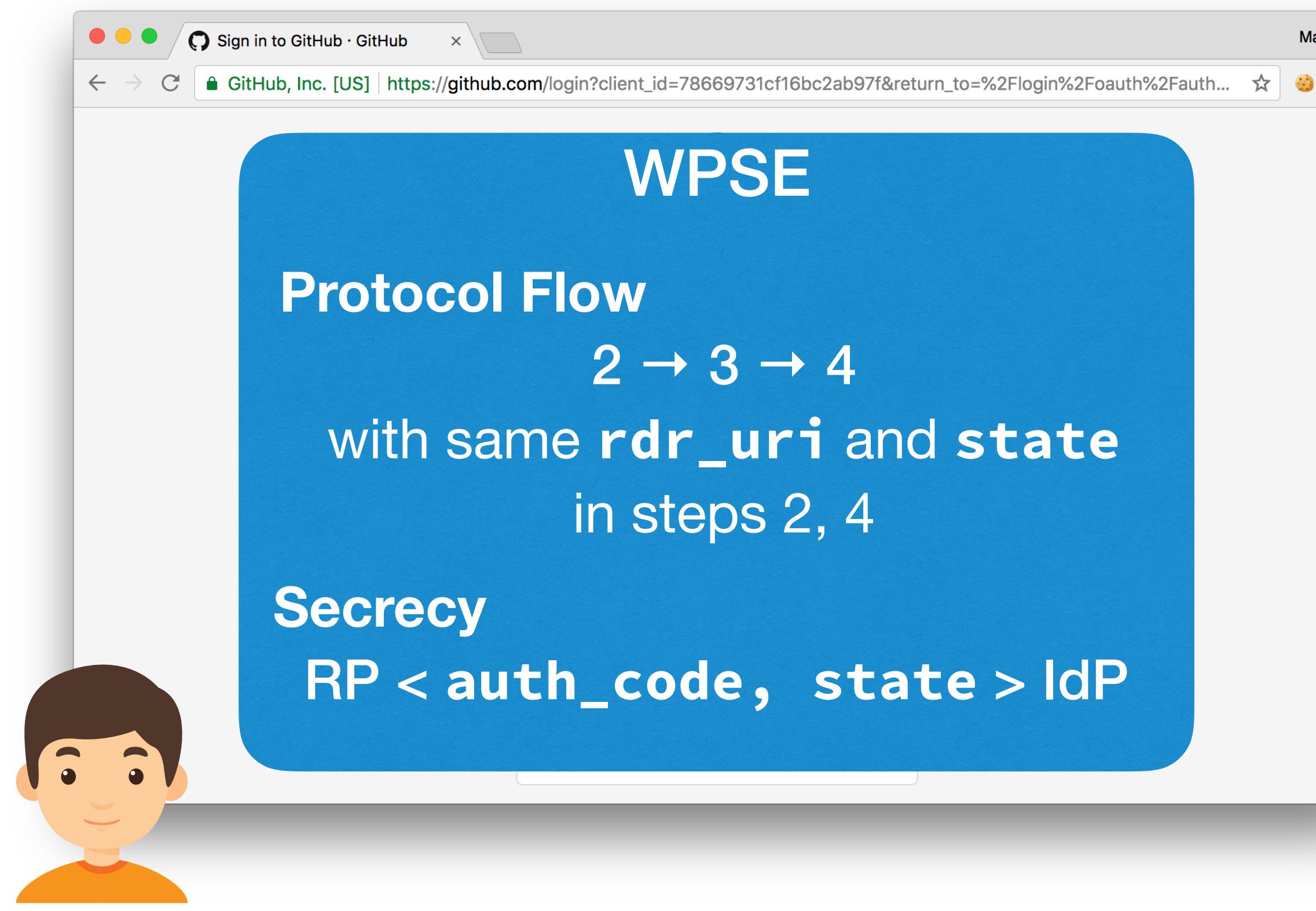
# FORTIFYING OAuth 2.0



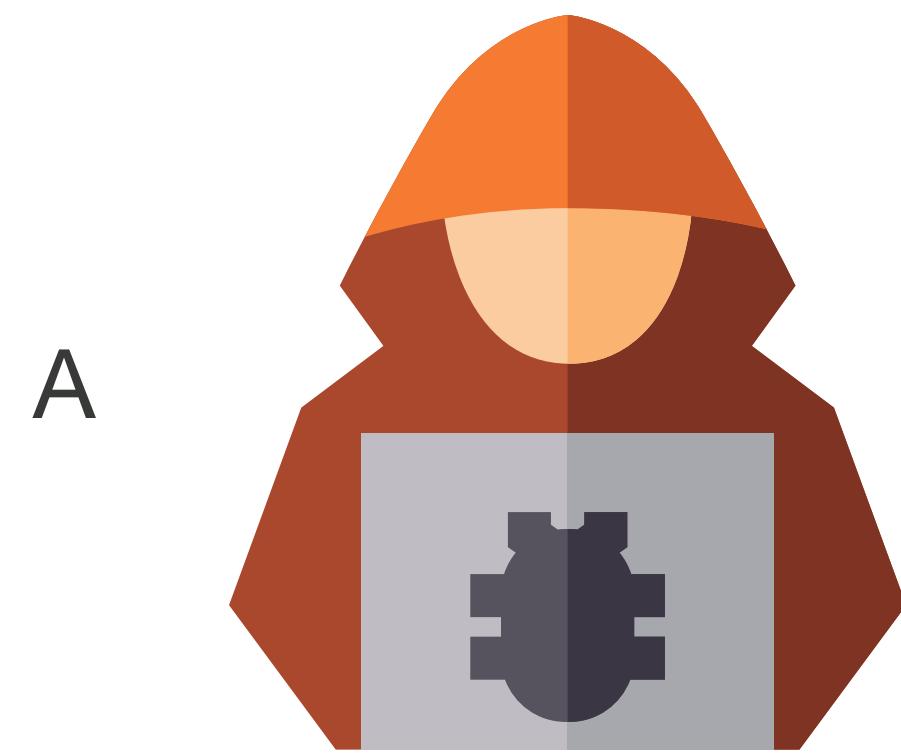
# FORTIFYING OAuth 2.0



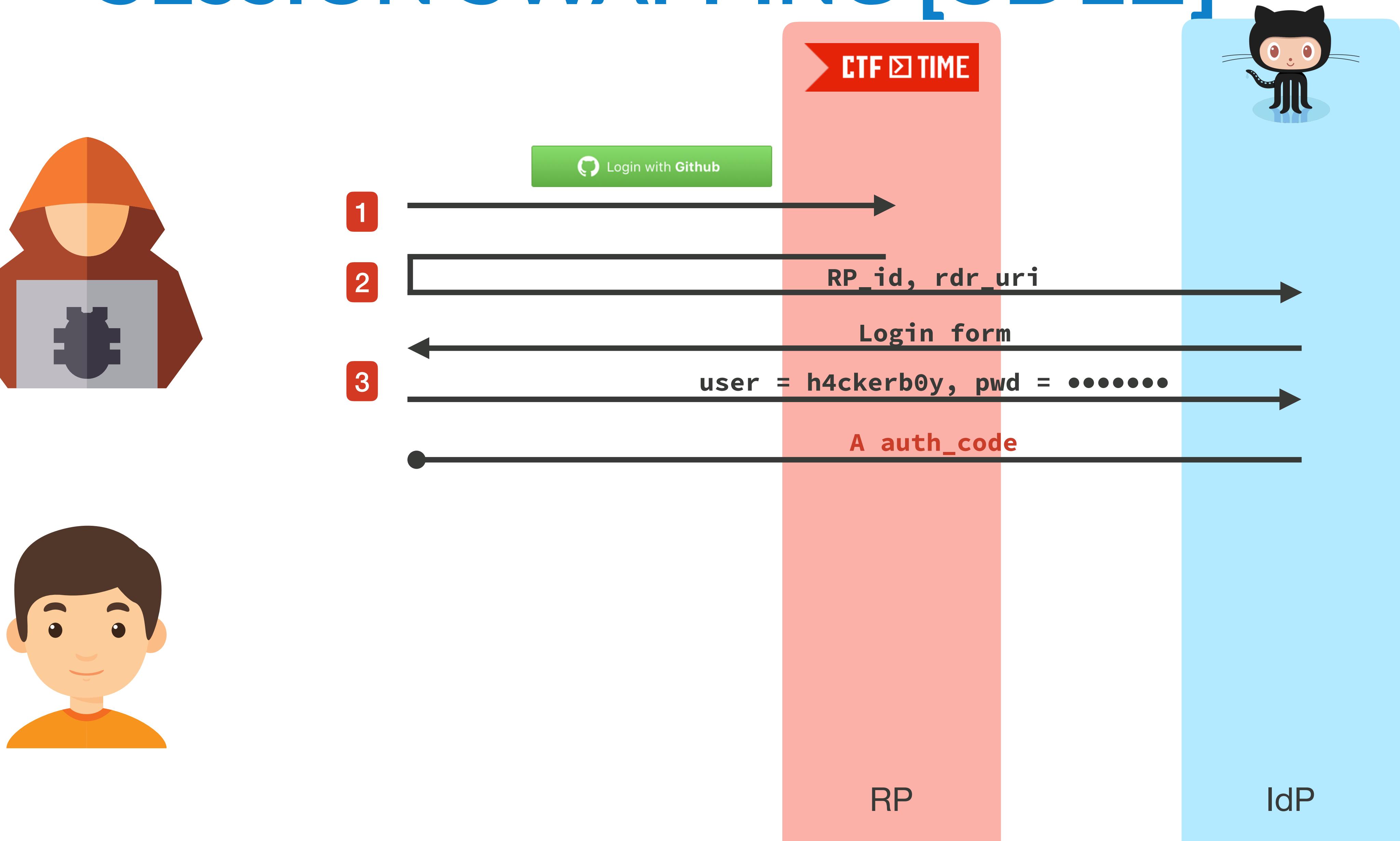
# FORTIFYING OAuth 2.0



# SESSION SWAPPING [SB12]



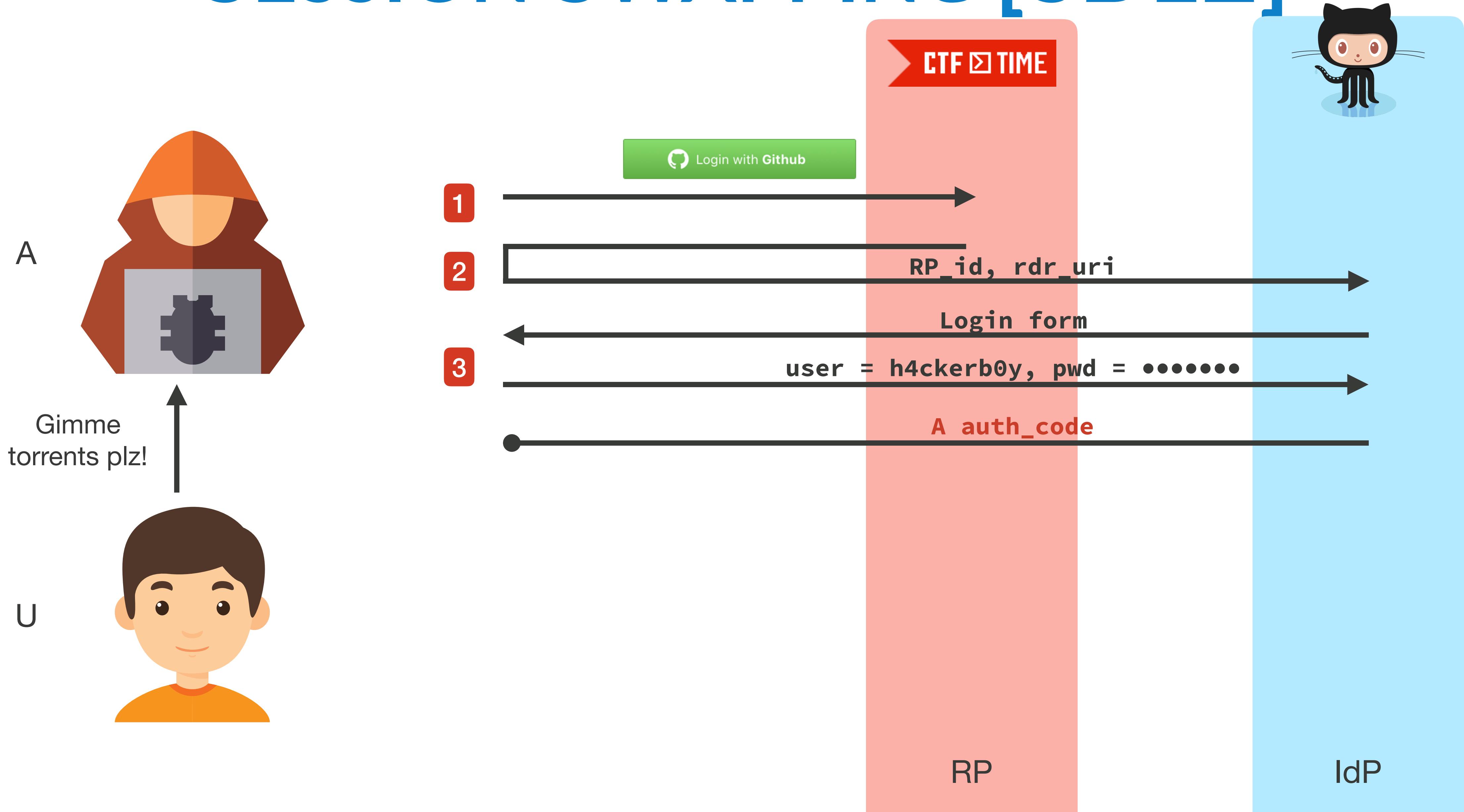
A



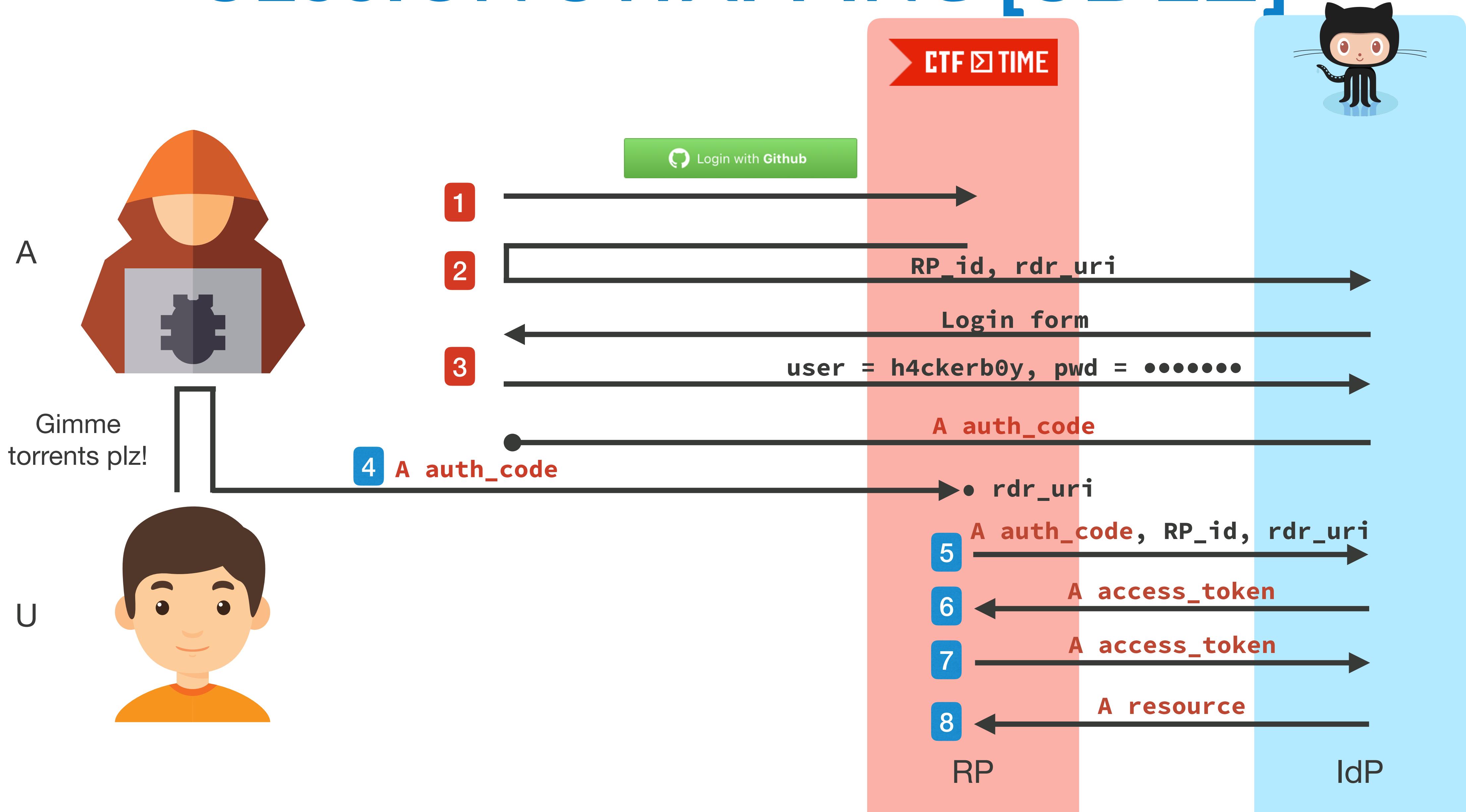
RP

IdP

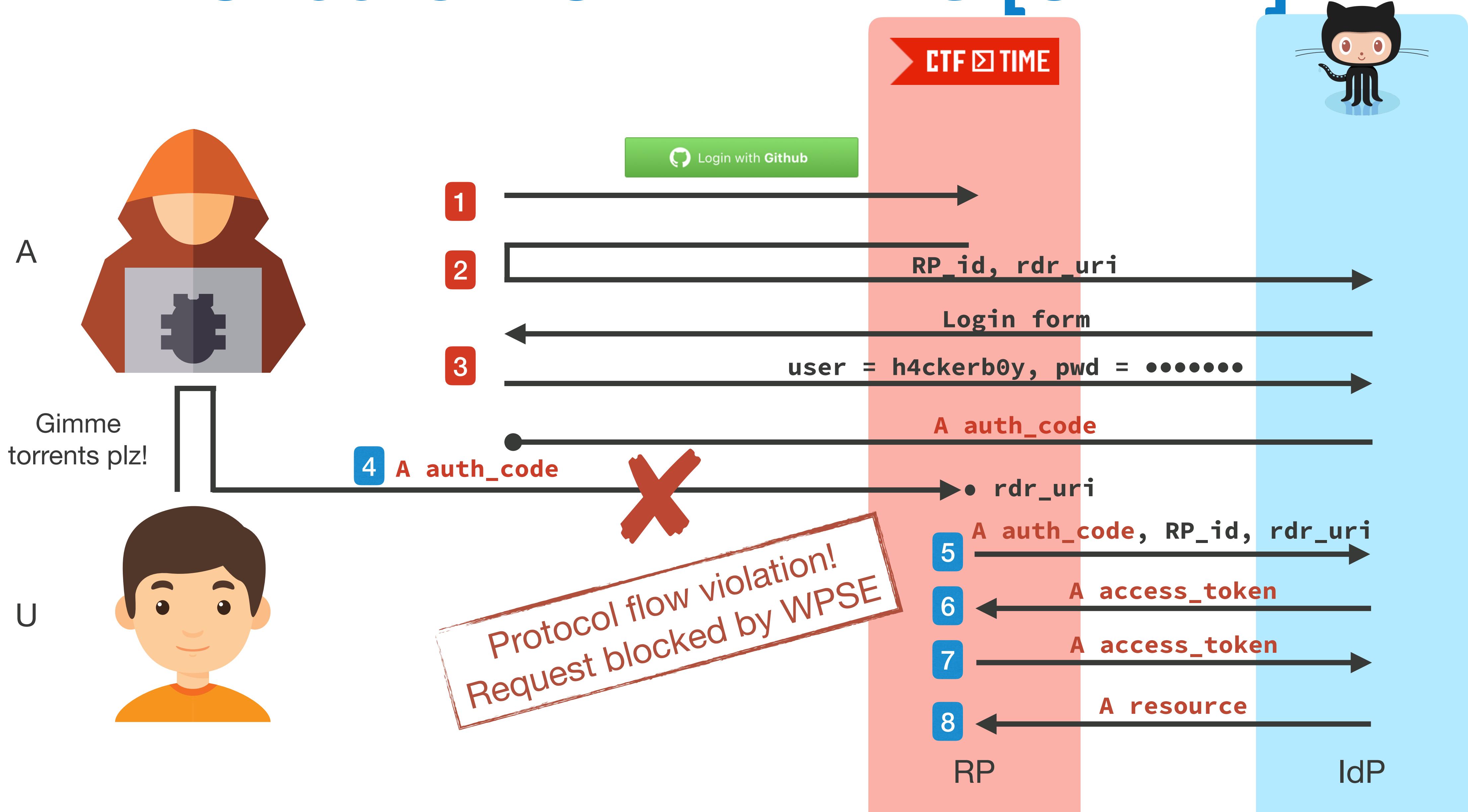
# SESSION SWAPPING [SB12]



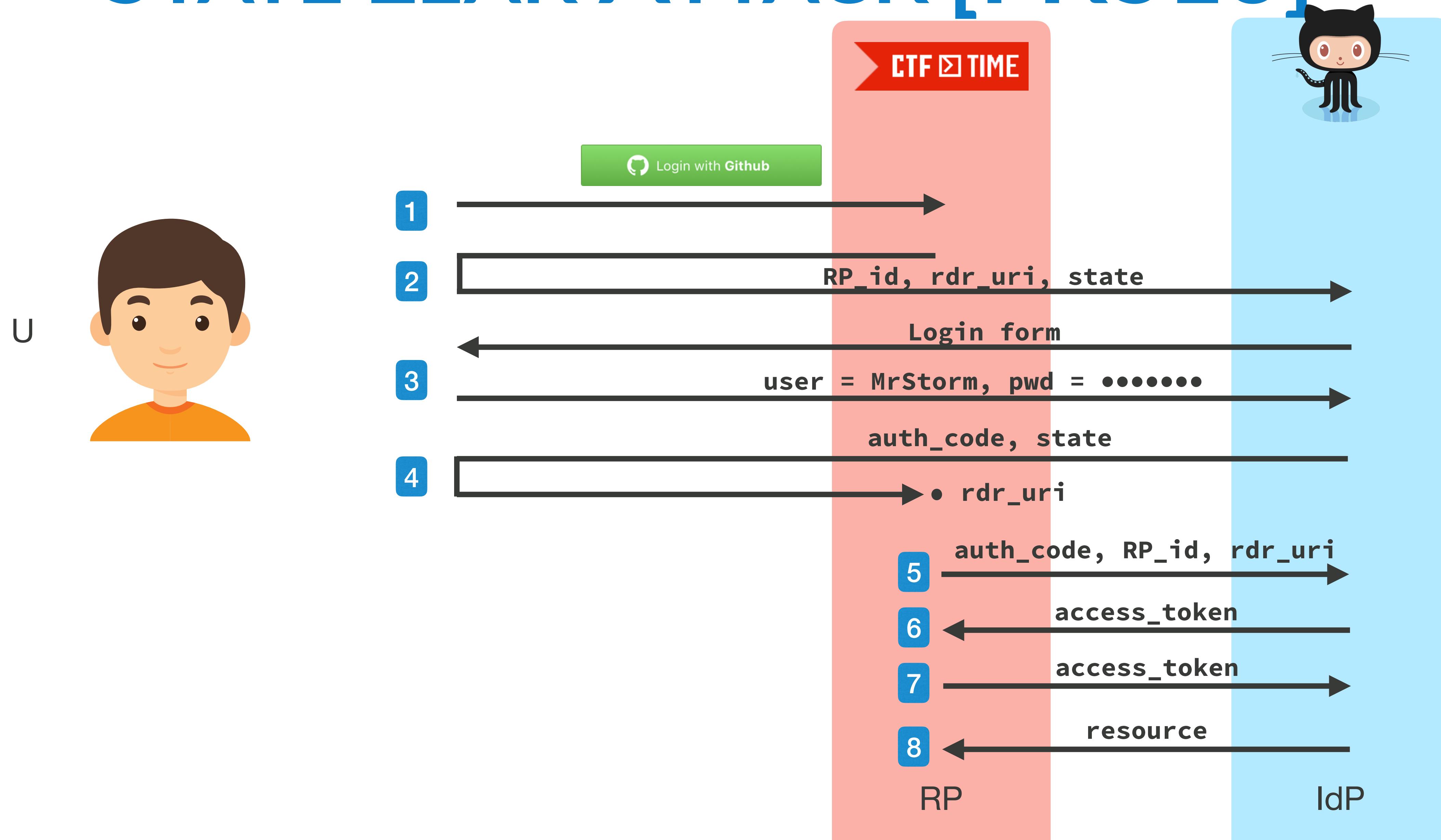
# SESSION SWAPPING [SB12]



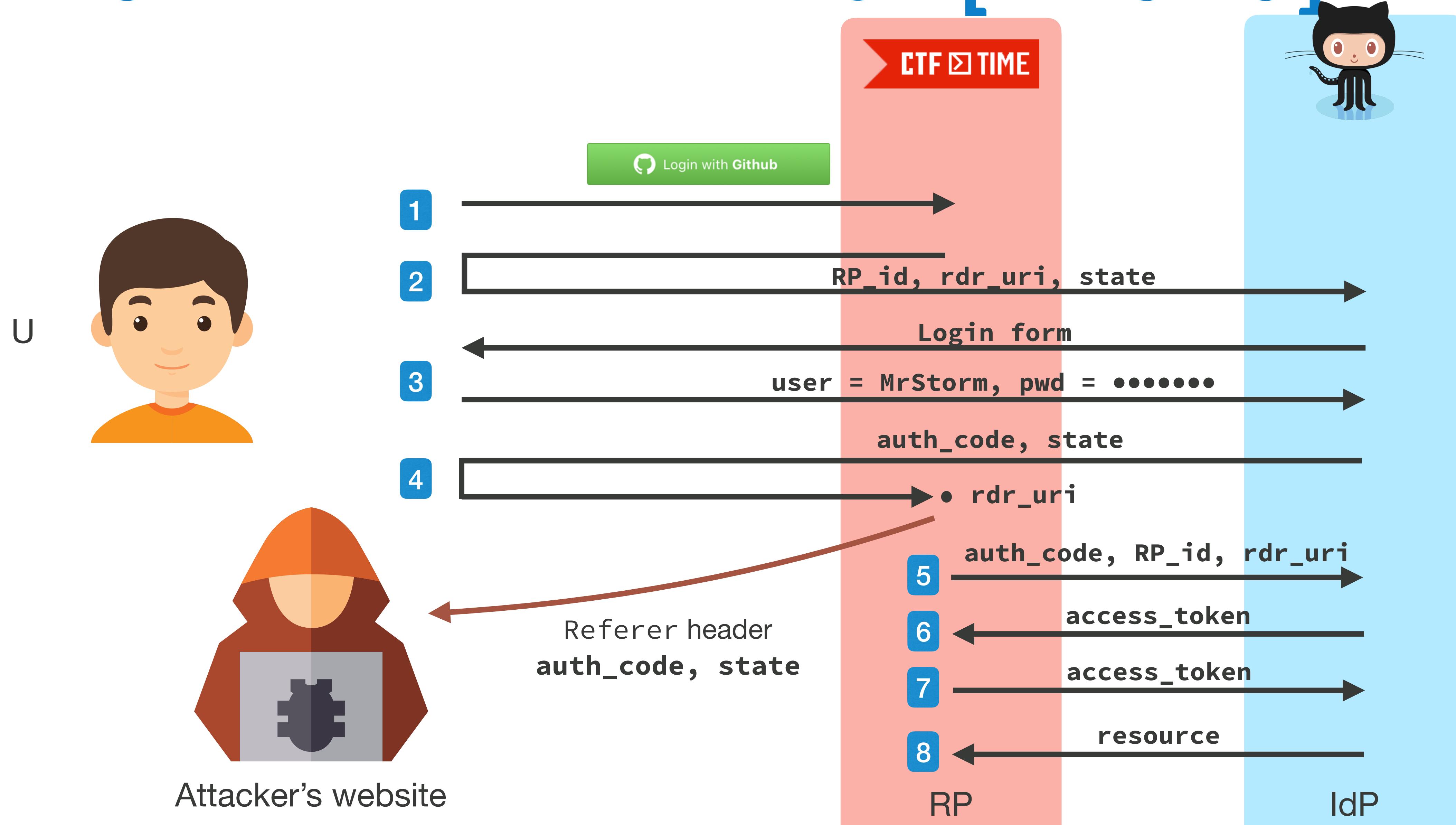
# SESSION SWAPPING [SB12]



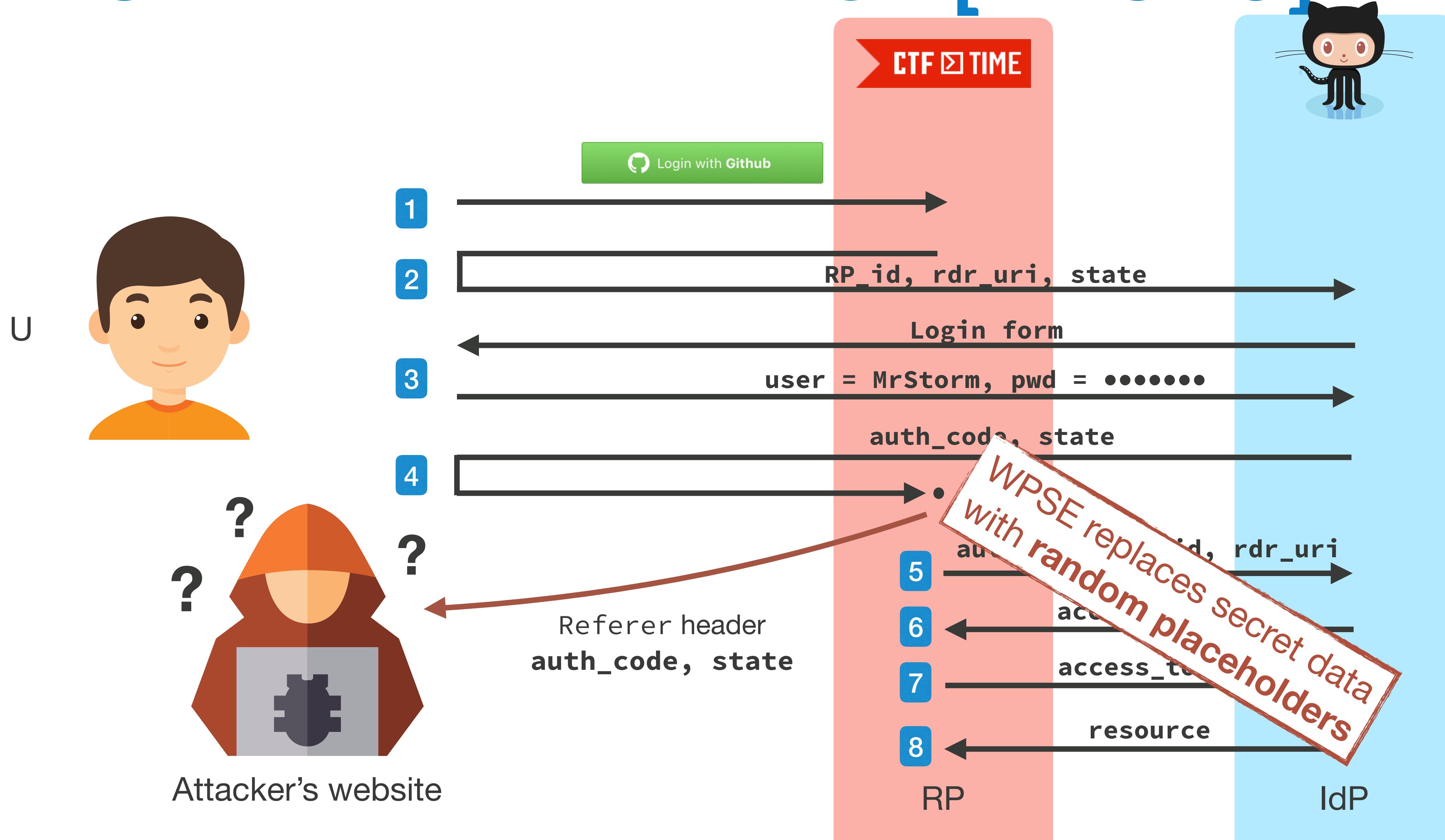
# STATE LEAK ATTACK [FKS16]



# STATE LEAK ATTACK [FKS16]



# STATE LEAK ATTACK [FKS16]

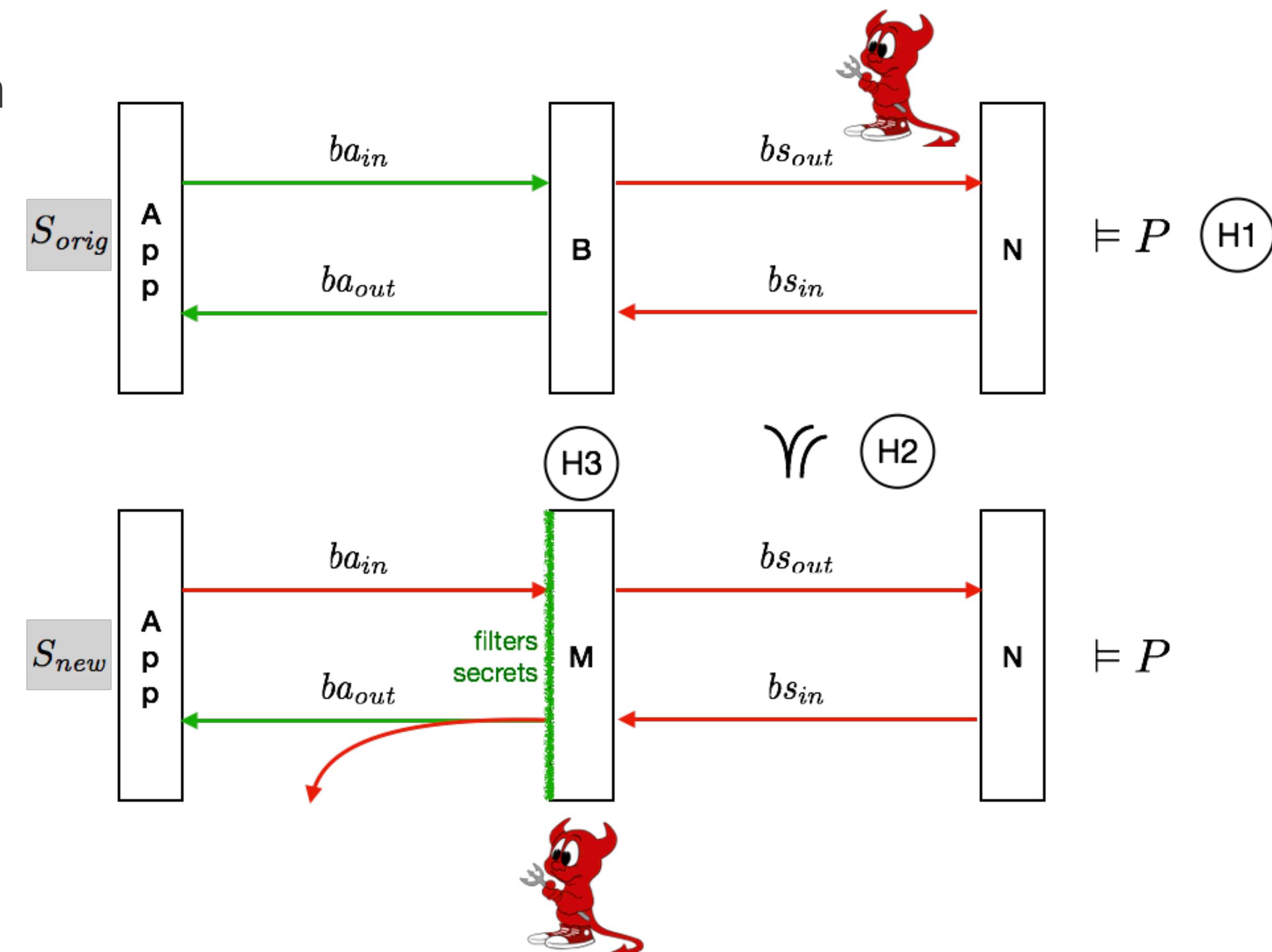


# FORMAL RESULTS

**(H1)** The protocol fulfills safety property  $P$  with a benign webpage

**(H2)** WPSE allows only a subset of the I/O sequences performed by the browser in a honest protocol run

**(H3)** Secrets are not leaked and securely stored by the browser



# FORMAL RESULTS

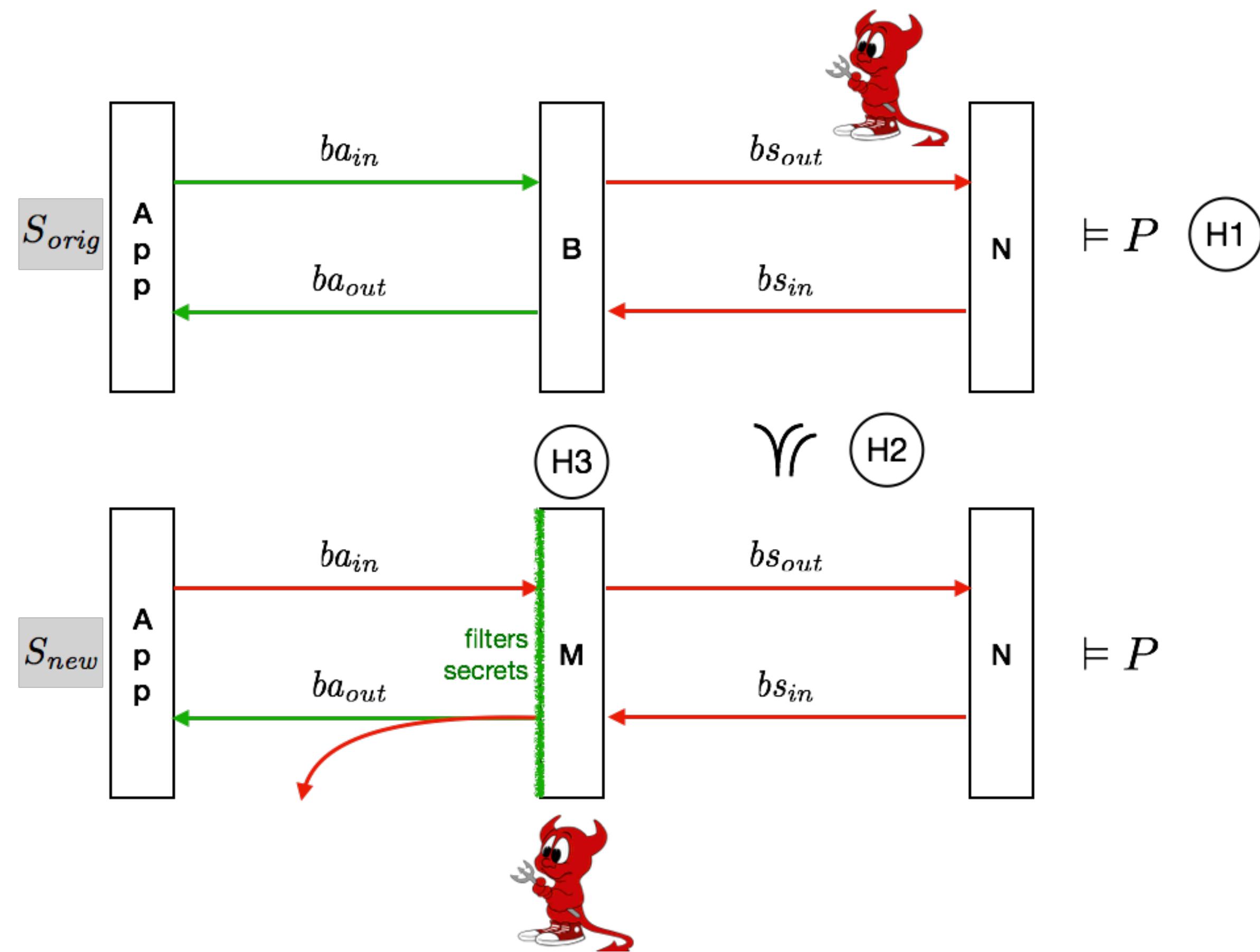
(H1) The protocol fulfills safety property  $P$  with a benign webpage

(H2) WPSE allows only a subset of the I/O sequences performed by the browser in a honest protocol run

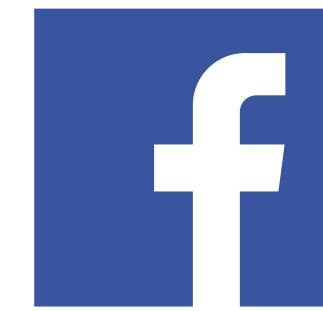
(H3) Secrets are not leaked and securely stored by the browser



The protocol fulfills  $P$  with a **compromised** browser monitored by WPSE



# EXPERIMENTAL EVALUATION



- Manual investigation of 30 RPs for each IdP from Alexa top 100K
- Analyzed both **authorization code mode** and **implicit mode** of OAuth 2.0

## Security

- Leakage of sensitive data due to **advertisement libraries** (4 RPs)
- Lack or misuse of the **state parameter** (55 RPs)

## Compatibility

Problems due to **security critical deviations** in the **protocol flow** (7 RPs), e.g. auth code is sent twice, second time over HTTP

# A NEW ATTACK AGAINST GOOGLE IMPLEMENTATION OF SAML 2.0

- Similar to the **session swapping** attack presented before
- **Login CSRF** against Google Suite applications (Google Drive, GMail, ...)

# A NEW ATTACK AGAINST GOOGLE IMPLEMENTATION OF SAML 2.0

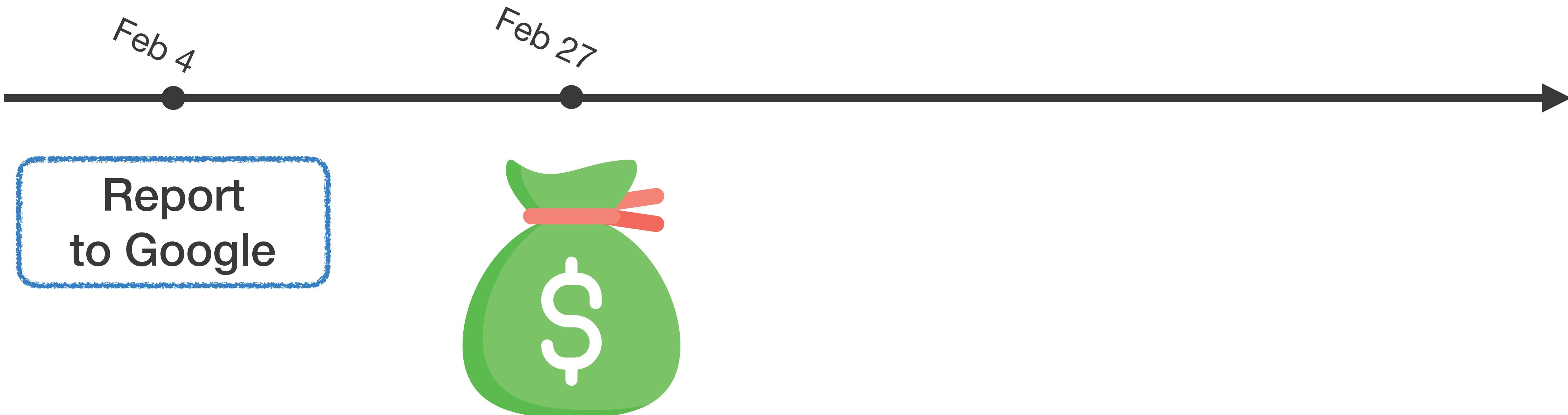
- Similar to the **session swapping** attack presented before
- **Login CSRF** against Google Suite applications (Google Drive, GMail, ...)

Feb 4



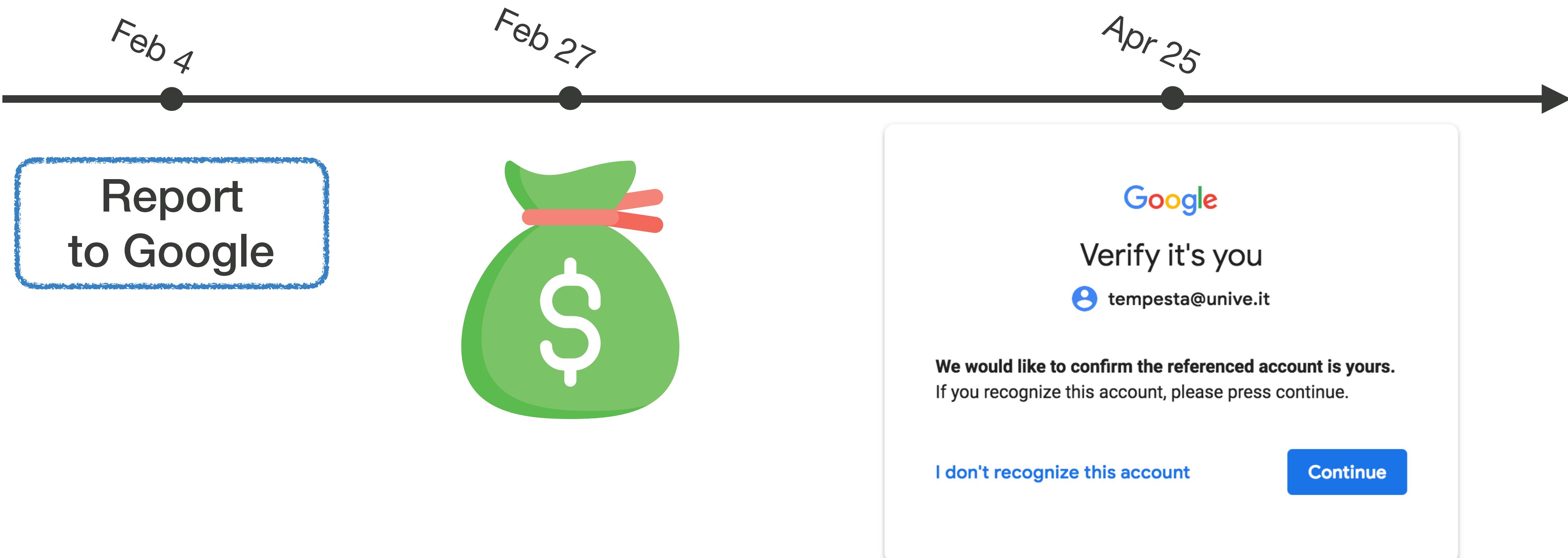
# A NEW ATTACK AGAINST GOOGLE IMPLEMENTATION OF SAML 2.0

- Similar to the **session swapping** attack presented before
- **Login CSRF** against Google Suite applications (Google Drive, GMail, ...)



# A NEW ATTACK AGAINST GOOGLE IMPLEMENTATION OF SAML 2.0

- Similar to the **session swapping** attack presented before
- **Login CSRF** against Google Suite applications (Google Drive, GMail, ...)



# SUMMING UP

**Lightweight policies on the client-side suffice to enforce provable security guarantees in web protocols**

# SUMMING UP

Lightweight policies on the client-side suffice to enforce provable security guarantees in web protocols



- Support for additional protocols e.g., **e-payments**
- **Automatic** techniques to **synthesize WPSE policies** from protocol specifications / browser traffic
- Embed WPSE into real **browsers**

# THANK YOU!

## QUESTIONS?



tempesta@unive.it



<https://sites.google.com/site/wpseproject/>