OAuth2.0协议

1. 定义

OAuth（开放授权）是一个开放标准，允许用户让第三方应用访问该用户在某一网站上存储的私密的资源（如照片，视频，联系人列表），而无需将用户名和密码提供给第三方应用。

1. 产生原因

例子：有一个“云冲印”的网站，可以将用户A存储在google的照片，冲印出来。A为了使用这个服务，必须让“云冲印”读取自己存储在google的照片。

？？“云冲印”如何获得用户的授权？？、

答案：给“云冲印”用户的用户名密码

一系列问题：

（1）“云冲印”缓存用户名密码，可能没有加密保护，一旦受到攻击，用户躺枪。

（2）“云冲印”没有权限限制，访问用户在google所有照片。

（3）用户无法撤销授权，除非修改密码。

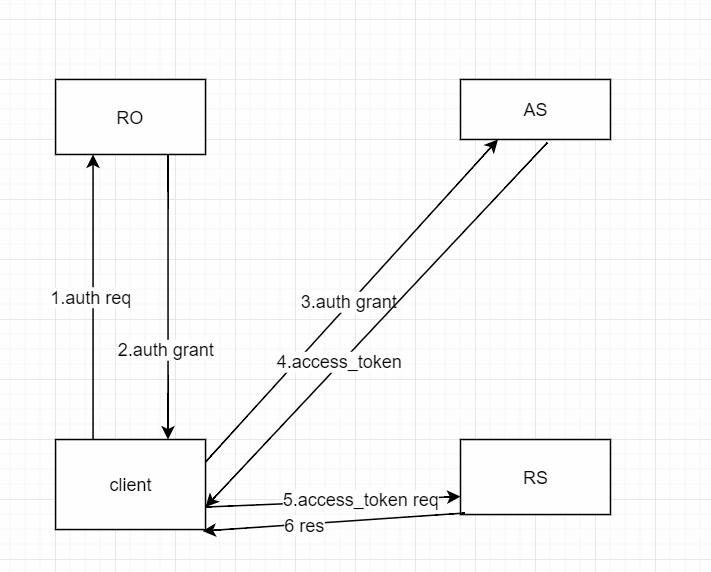
So，OAuth2.0运营而生。

1. 基础概念
2. Client ：第三方应用，例如“云冲印”、
3. RO（resource owner）：资源所有者，例如“A”
4. RS（resource server）：资源服务器，存储资源。例如“google照片”
5. AS（authorization server）：授权服务器，认证RO身份，为RO提供授权审批流程，并最终颁发授权令牌（Access\_token）
6. Oauth思路

OAuth在"客户端"与"服务提供商"之间，设置了一个授权层（authorization layer）。"客户端"不能直接登录"服务提供商"，只能登录授权层，以此将用户与客户端区分开来。"客户端"登录授权层所用的令牌（token），与用户的密码不同。用户可以在登录的时候，指定授权层令牌的权限范围和有效期。

"客户端"登录授权层以后，"服务提供商"根据令牌的权限范围和有效期，向"客户端"开放用户储存的资料。

1. 运行流程

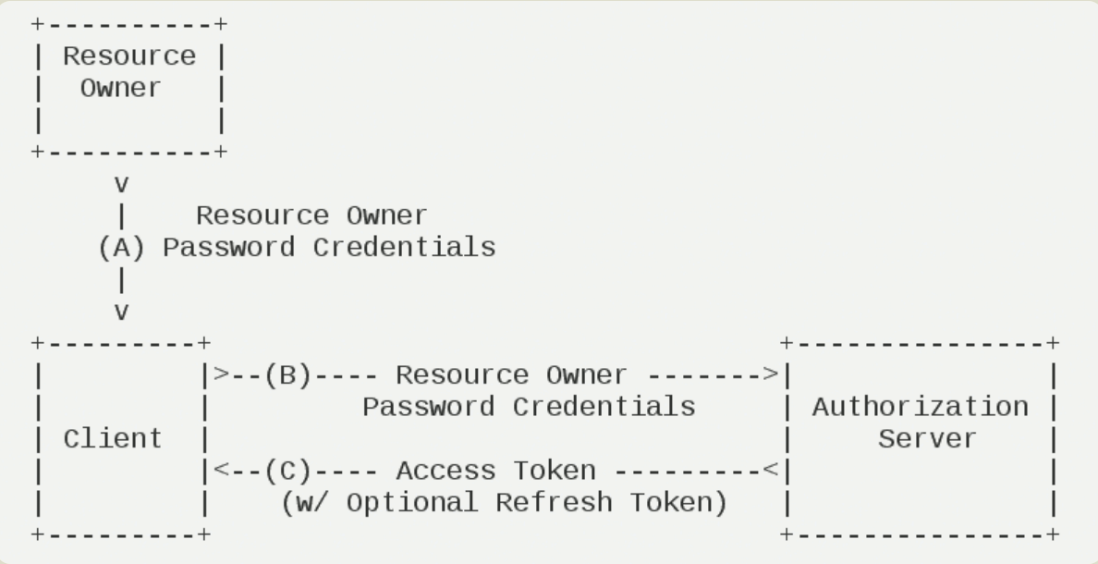


关键点：Ro 如何授权给client

三种方式：授权码模式、简化模式、密码模式。

1. 模式介绍

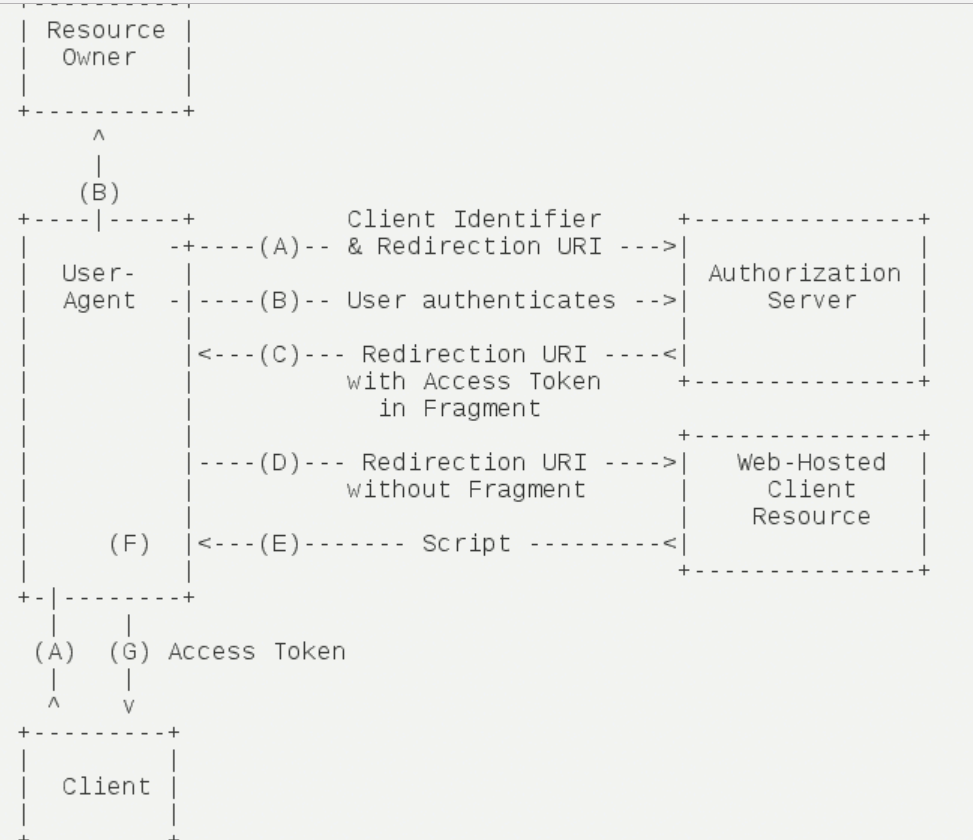
密码模式-----高度信任



步骤介绍：

1. 用户给客户的提供用户名密码
2. Client拿用户名密码去As验证，拿token
3. As返回token。

简化模式：-------web站点



步骤描述：

1. （A）client 向用户代理 request，302重定向到AS，请求参数client\_id 、scope、redirect\_uri、state 、access\_type、approval\_prompt。

例如：GET/authorize?response\_type=code&client\_id=s6BhdRkqt3&state=xyz&redirect\_uri=https%3A%2F%2Fclient%2Eexample%2Ecom%2Fcb HTTP/1.1

[注].state csrf攻击

1. （B）approval\_prompt = true，则AS通过交互界面得到用户的批准，false默认可以。
2. （C）返回access\_token

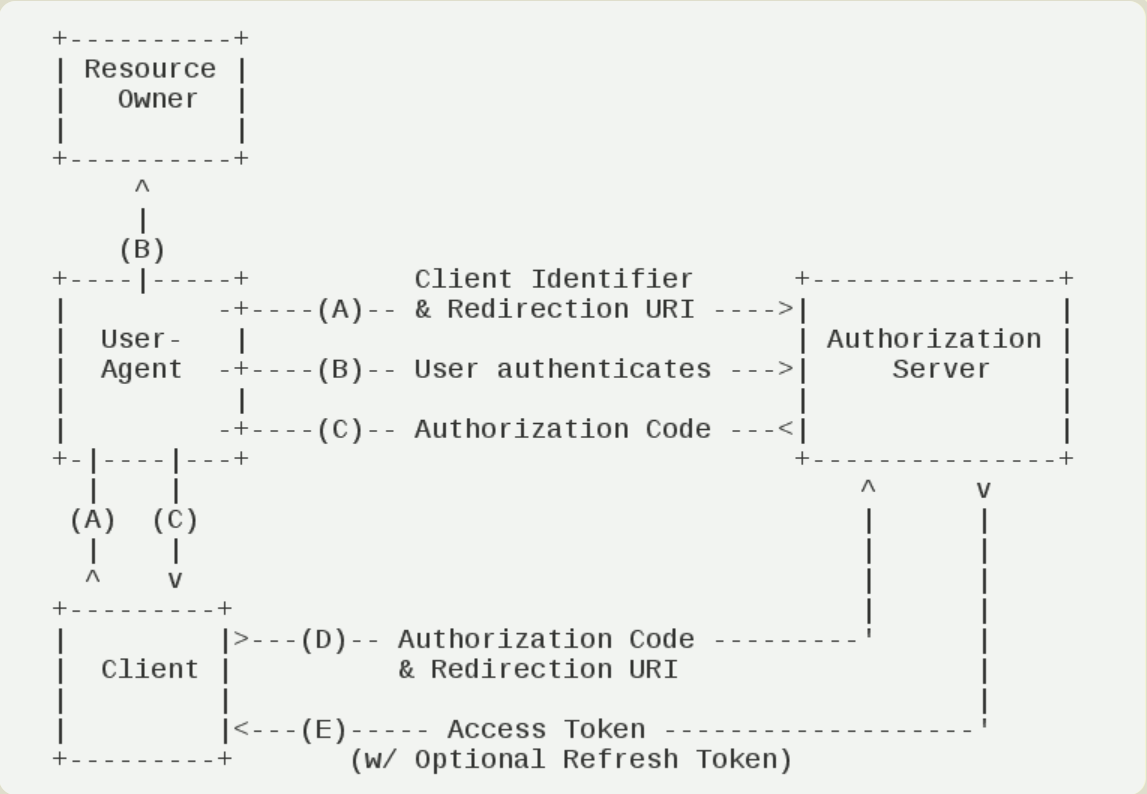
例如：

HTTP/1.1 302 Found

Location:http://example.com/cb#access\_token=2YotnFZFEjr1zCsicMWpAA&state=xyz&token\_type=example&expires\_in=3600

1. （D），访问Location的uri，
2. （E），返回的代码提取Location参数access\_token，获得资源。

授权码模式：------第三方应用（相比于上一种，多了Authorization\_code验证）



步骤描述：

1. （A）client 向用户代理 request，302重定向到AS，请求参数client\_id 、scope、redirect\_uri、state 、access\_type、approval\_prompt。当access\_type=offline时，AS将在颁发access\_token时，同时还会颁发一个refresh\_token。因为access\_token的有效期较短（如3600秒），为了优化协议执行流程，offline方式将允许Client直接持refresh\_token来换取一个新的access\_token

例如：GET/authorize?response\_type=code&client\_id=s6BhdRkqt3&state=xyz&redirect\_uri=https%3A%2F%2Fclient%2Eexample%2Ecom%2Fcb HTTP/1.1

[注].state csrf攻击

（2）approval\_prompt = true，则AS通过交互界面得到用户的批准，false默认可以。

（3）（C）用户同意授权，则重定向返回client authorization\_code 和state 例如：

HTTP/1.1 302 Found

Location:https://client.example.com/cb?code=SplxlOBeZQQYbYS6WxSbIA&state=xyz

（4）（D）向AS获取access\_code.

例如：grant\_type=authorization\_code&code=SplxlOBeZQQYbYS6WxSbIA

&redirect\_uri=https%3A%2F%2Fclient%2Eexample%2Ecom%2Fcb

（5）（E）返回access\_code 以及refresh\_token（若access\_type=offline）。

例如：

HTTP/1.1 200 OK

Content-Type: application/json;charset=UTF-8

Cache-Control: no-store

Pragma: no-cache

{

"access\_token":"2YotnFZFEjr1zCsicMWpAA",

"token\_type":"example",

"expires\_in":3600,

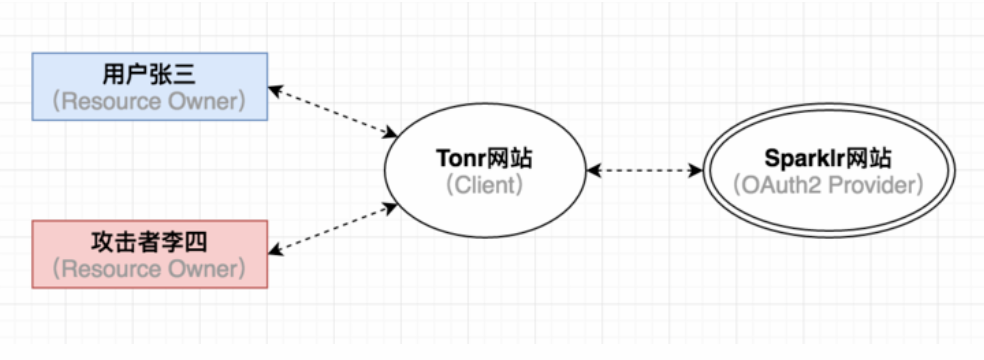
"refresh\_token":"tGzv3JOkF0XG5Qx2TlKWIA",

"example\_parameter":"example\_value"

}

1. 授权码模式详细介绍
2. OAuth2.0 存在的安全漏洞
3. csrf攻击，前提get请求没有state字段

发生场景：



Step1.攻击者李四登录Tonr网站，选择绑定Sparklr账号

Step2.Tonr重定向到Sparklr（没有state字段），然后返回Authorization\_code.

Step3.攻击者李四制造了一**个恶意web网页**，网页会向Sparklr发送带有李四的Authorization\_code的请求access\_token的页面

Step4.用户张三登录Tonr，但是没有任何账号绑定，然后张三访问李四的web页面，成功的从Sparklr获得access\_token.

Step5.攻击者李四可以成功的操作张三的Tonr账户信息。

本质：请求分步执行。

避免方式：添加state。

说明：恶意网页不是输入用户名、密码的网页。张三登录Tonr网站后，再去访问恶意web网页（web网站会向Sparklr发送带有李四的Authorization\_code的请求access\_token的页面），然后李四用access\_token成功登录张三的账号，前提是Tonr网站不是只能绑定一个OAuth提供方的账号。

b、redirect\_uri绕过导致授权劫持

本质：redirect\_uri校验失败。

例如:网易账号登录56网。

https://api.t.163.com/oauth2/authorize烦忙client\_id=II5coZy8DdAtKt7a&redirect\_uri=http%3A%2F%2Fapp.56.com%2Fcooperate%2Findex.php%3Faction%3DWeibo%26tag%3Dwy%26do%3DCheckLogin%26from%3Dregbox&response\_type=code&state=unk-qogvtqoomz

替换为：redirect\_uri=http%3A%2F%2Fapp.56.com%40wooyun.org

解析匹配redirect-uri：用 ./解析。

漏洞修补：正则表达式，rfc，很复杂。

1. 为什么要Authorization\_code

Redirect\_uri不安全，通过HTTP referrer伪造后，造成access\_token被盗用。