Network Security

OAuth

Dr. Ing. Simone Cirani

Parma, May 28th, 2013

Online Services and Private Data

- The evolution of online services, such as social networks, has had a huge impact on the amount of data and personal information disseminated on the Internet
- The information owned by online services is made available to third-party applications in the form of public Application Programming Interfaces (APIs), typically using HTTP as transport protocol and relying on the REpresentational State Transfer (REST) architecture
- Someone else, besides the entity which generates the information and the service that is hosting it, can access this information

Material and Credits

- http://tools.ietf.org/html/rfc5849 → OAuth 1.0 RFC
- http://tools.ietf.org/html/rfc6749 → OAuth 2.0 RFC
- http://oauth.net/ → website about OAuth 1.0 and 2.0
- http://hueniverse.com/oauth/guide/ → Guide to OAuth 1.0

A real-life example

- You have subscribed to an online service (“PhotoPrint”) which offers prints of pictures that you have uploaded to Facebook
- You have setup your privacy settings so that only your friends are allowed to see your pictures
- PhotoPrint isn’t your friend, it is a service...
- Big problem: how can PhotoPrint access your pictures?

Dr. Ing. Simone Cirani
Università degli Studi di Parma

Solution 1

- **Basic authentication**: Give your Facebook username and password to PhotoPrint
- PhotoPrint will log on to Facebook using your credentials
- PhotoPrint will navigate your photos and download them in order to be able to print them
- "Ok, I received my photo prints but I am not totally happy... I do not feel safe..."

Dr. Ing. Simone Cirani
Università degli Studi di Parma

Problems with Solution 1

- **If I give my username and password to PhotoPrint, it can use them to do anything as if it was me**, including sending and reading private messages, getting personal information, and so on (Facebook could not recognize who is accessing the service)
- **Never share your username and password with anybody**
- Other problem: if I subscribe to many services and I give my credentials to all of them, when I decide I no longer want one of them, I'd need to change username and password and re-enter them in all other services.

Dr. Ing. Simone Cirani
Università degli Studi di Parma

Solution 2

- **OAuth** (Open Authorization)

An open protocol to allow secure authorization in a simple and standard method from web, mobile and desktop applications.
http://oauth.net/

- OAuth is defined and specified in RFC 5849
- OAuth 2.0 in RFC 6749
- OAuth allows users to grant limited and controlled access to their personal data stored on some service to third-party applications

Dr. Ing. Simone Cirani
Università degli Studi di Parma

Who is using OAuth?

Service Provider	OAuth protocol version
Dropbox	1.0
Evernote	1.0
Facebook	2.0 draft 12
Flickr	1.0a
FourSquare	2.0
GitHub	2.0
Google	2.0
Google App Engine	1.0a
Instagram	2.0
LinkedIn	1.0a, 2.0
Microsoft	2.0
MySpace	1.0a
PayPal	2.0
Twitter	1.0a, 2.0
Yahoo!	1.0a

Solution 2

- When using OAuth, you will explicitly authorize PhotoPrint to gain access to your Facebook photos only
- The authorization grant is represented by a so-called **access token**
- The access token is a piece of information that identifies
  - the application
  - the user
  - the type of actions that the application can execute
- When PhotoPrint needs to access your pictures, it will exhibit its access token and Facebook, after verifying it, will let PhotoPrint grab the photos

Authorization

- Authorization defines what rights and services the end-user is granted once access is allowed
- Authentication and authorization are usually performed together
- Information can be retrieved and presented differently to users, depending on their identity
  - Is the requestor authorized to get this information?
  - How should the information be presented?
  - Should the information be presented entirely or partially?

OAuth roles

- OAuth defines three roles (or actors) involved in a service authorization scenario
- **Service Provider**: a service that is hosting a number of protected user-related resources (Facebook)
- **Service Consumer (or Client)**: a third-party application that is willing to access protected resources (PhotoPrint)
- **User**: an entity who is keeping some protected resources on the Service Provider (You)

OAuth credentials

- There are 3 types of credentials used in the OAuth protocol
- Each credential is associated with a secret; the secret is needed to perform request signing
- **Consumer key (and secret)**: used to authenticate the consumer
- **Request token (and secret)**: exchanged for an access token after “user’s consent”
- **Access token (and secret)**: used in requests that result in accessing protected resources
- Tokens are pieces of information used instead of username and password

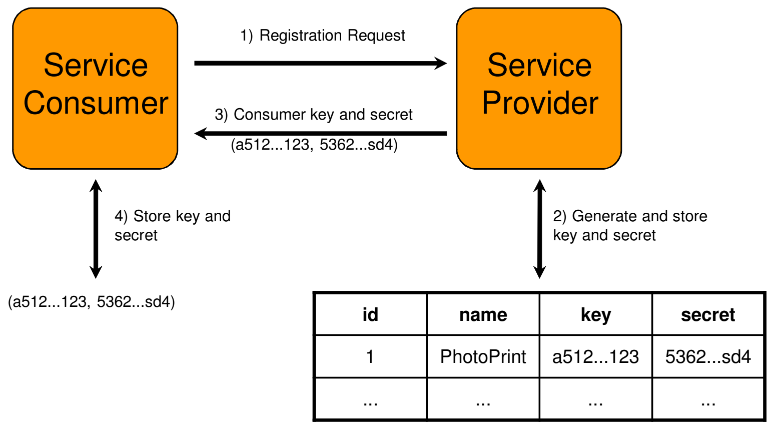
Dr. Ing. Simone Cirani
Università degli Studi di Parma

Service Consumer Registration

- Service Providers must recognize the applications (Service Consumers) that are going to request information
- Service Consumers are typically required to register themselves to the Service Provider
- The registration phase returns a consumer key and secret to the Service Consumer
- The Service Consumer is required to use these information when it interacts with the Service Provider, so that it can be correctly identified

Dr. Ing. Simone Cirani
Università degli Studi di Parma

Service Consumer Registration



id	name	key	secret
1	PhotoPrint	a512...123	5362...sd4
...	...	...	...

Dr. Ing. Simone Cirani
Università degli Studi di Parma

Digital Signatures

- OAuth uses digital signatures to verify authenticity and integrity of messages:
  - HMAC-SHA-1
  - RSA-SHA-1
  - PLAINTEXT
- The secret depends on the method used for digital signature.
  - PLAINTEXT and HMAC-SHA-1: the shared secret is a combination of the consumer secret and token secret
  - RSA-SHA-1: the consumer secret serves as a private key
- The signature includes a nonce to protect against replay attacks; a timestamp is also used to reduce storage (nonces must be used only once)
- 22

Dr. Ing. Simone Cirani
Università degli Studi di Parma

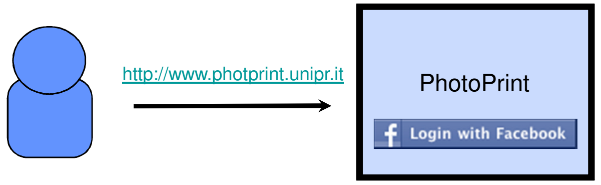
Getting an access token

- OAuth defines a redirection-based mechanism to let users grant privileges to client applications
- Redirection is performed on top of HTTP requests
- Problem: how can PhotoPrint retrieve an access token so that it can print some photos?
- Let’s start...

Dr. Ing. Simone Cirani
Università degli Studi di Parma

Getting an access token

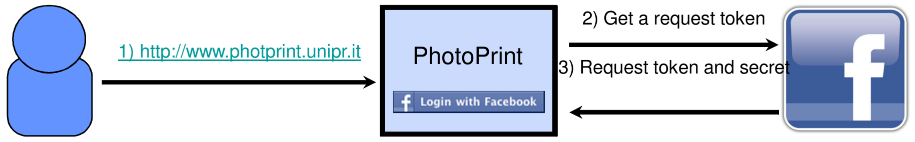
1. The user navigates on the client's website; the website needs access to the user's protected data



Dr. Ing. Simone Cirani
Università degli Studi di Parma

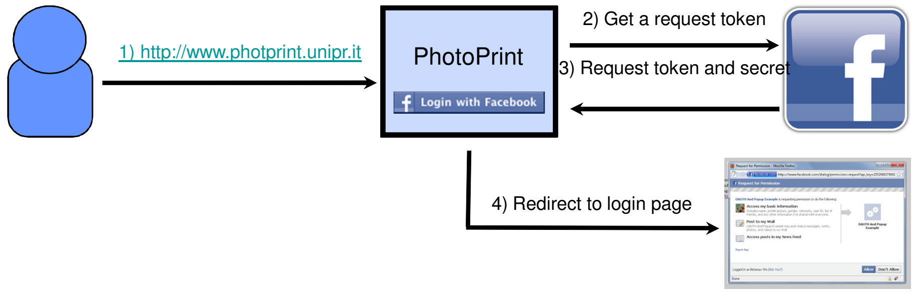
Getting an access token

2. The URL linked by the login button contains the request token. The request token is retrieved by a call to the Service Provider when the page is created; the request is signed using the consumer key and secret so that the SP can recognize who is requesting the RT



Dr. Ing. Simone Cirani
Università degli Studi di Parma

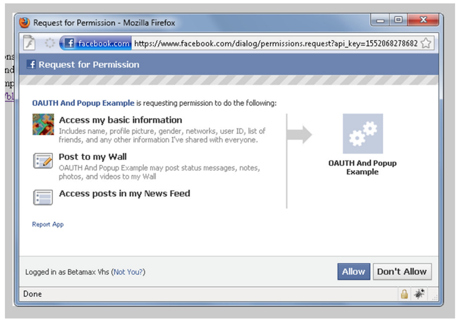
Getting an access token


1) http://www.photprint.unipr.it
PhotoPrint
Login with Facebook
2) Get a request token
3) Request token and secret
4) Redirect to login page

Dr. Ing. Simone Cirani
Università degli Studi di Parma

Getting an access token

5. The login page asks the user if he/she consents that client application will access protected data; typically the popup also states the types of actions that the client will perform on user's behalf.


Request for Permission - Mozilla Firefox

facebook.com https://www.facebook.com/dialog/permissions.request?api_key=1552068278682

Request for Permission

OAUTH And Popup Example is requesting permission to do the following:

Access my basic information
Includes name, profile picture, gender, networks, user ID, list of friends, and any other information I've shared with everyone.

Post to my Wall
OAUTH And Popup Example may post status messages, notes, photos, and videos to my Wall

Access posts in my News Feed

OAUTH And Popup Example

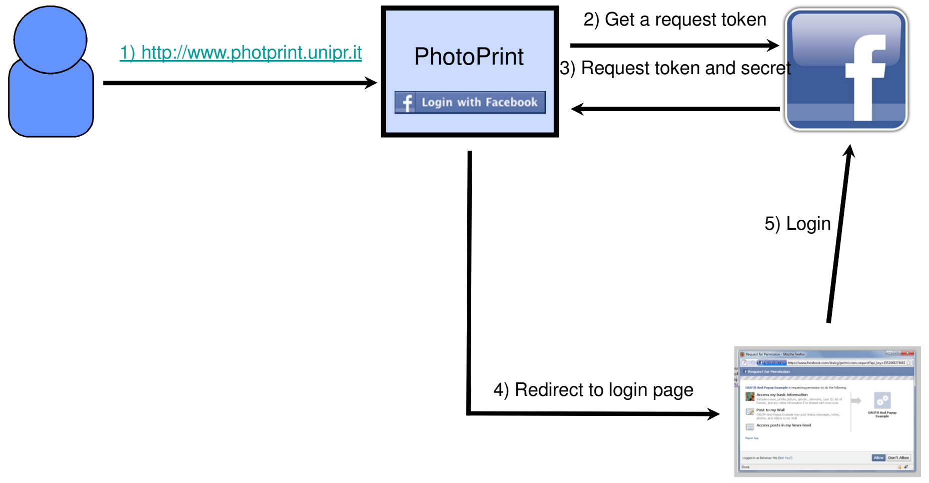
Report App

Logged in as Betamax Vhs (Not You?)

Allow | Don't Allow

Done

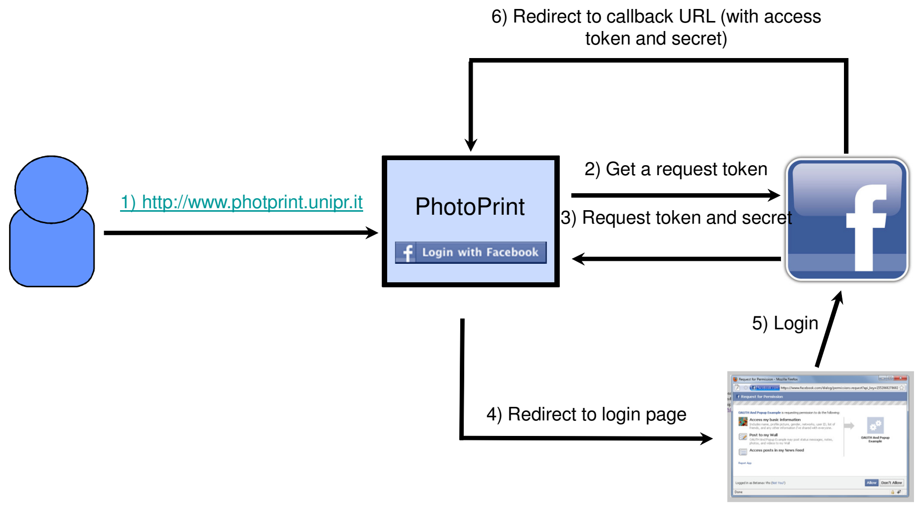
Getting an access token


1) http://www.photprint.unipr.it
PhotoPrint
Login with Facebook
2) Get a request token
3) Request token and secret
5) Login
4) Redirect to login page

Getting an access token

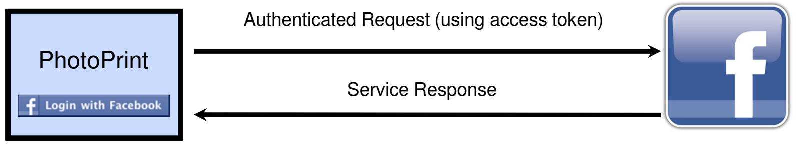
6. By logging in, the user explicitly grants privileges to the client application to perform actions on his/her behalf. The login form sends an HTTP POST request to the Service Provider which results in a redirection to the client's login callback URL. The callback URL is used to let the Service Provider exchange the request token contained in the login form for an access token and secret.

Getting an access token


6) Redirect to callback URL (with access token and secret)
1) http://www.photprint.unipr.it
PhotoPrint
Login with Facebook
2) Get a request token
3) Request token and secret
5) Login
4) Redirect to login page

Using the access token

Now the Service Consumer has received an access token and secret and it can use them to perform authenticated requests on behalf of the user



Dr. Ing. Simone Cirani
Università degli Studi di Parma

Signing requests

- An authenticated request includes several protocol parameters. Each parameter name begins with the "oauth_" prefix, and the parameter names and values are case sensitive
- Required parameters:
  - oauth_consumer_key
  - oauth_token
  - oauth_signature_method
  - oauth_timestamp
  - oauth_nonce
  - oauth_version

Dr. Ing. Simone Cirani
Università degli Studi di Parma

Signing request example

- Walkthrough taken from http://hueniverse.com/oauth/guide/authentication/
- The client would like to access a protected resource located at http://www.photoprint.unipr.it/print with the parameters
  - user=12345
  - size=medium
- The client has previously registered with the server and obtained the client identifier abcde and client secret zyxwv. It has executed the OAuth workflow and obtained an access token act123 and token secret act456
- To sign the request, the client is using the HMAC-SHA1 signature method and generated a nonce string xyzxyz and timestamp 1369735200

Dr. Ing. Simone Cirani
Università degli Studi di Parma

Signing request example

- The oauth_ parameters are:

parameter	value
oauth_consumer_key	abcde
oauth_token	act123
oauth_nonce	xyzxyz
oauth_timestamp	1369735200
oauth_signature_method	HMAC-SHA-1
oauth_version	1.0

- The URL query parameters (user and size) are UTF-8 encoded and then URL encoded
- All the parameters are then sorted by name, in order to create a univocal string

Dr. Ing. Simone Cirani
Università degli Studi di Parma

Signing request example

- The sorted parameters are:

parameter	value
oauth_consumer_key	abcde
oauth_nonce	xyzxyz
oauth_signature_method	HMAC-SHA-1
oauth_timestamp	1369735200
oauth_token	act123
oauth_version	1.0
size	medium
user	12345

Signing request example

- The parameters are concatenated into a single string using an encoding denoted as application/x-www-form-urlencoded
  - each key/value pair is transformed into a single string in the following way: key=value
  - all obtained strings are concatenated by adding a ‘&’ sign between them: key1=value1&key2=value2
- In our example, the resulting string is the following:

oauth_consumer_key=abcde&oauth_nonce=xyzxyz&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1369735200&oauth_token=act123&oauth_version=1.0&size=medium&user=12345

Signing request example

- Next the request URL is normalized as

scheme://authority:port/path

- The port is omitted if it is a standard HTTP port (80 or 443)
- Query parameters are omitted since they have already been included in the previous section
- Fragments (the part of the URL after the ‘#’ sign) are excluded
- In our example, the normalized URL is equal to the service URL:

http://www.photoprint.unipr.it/print

Signing request example

- The next step is to create the so-called **Signature Base String**
- The normalized URL and normalized parameters are URL-encoded (‘:’ becomes %3A, ‘/’ becomes %2F, ‘=’ becomes %3D, ‘&’ becomes %26, and so on). In our case, the URL becomes:

http%3A%2F%2Fwww.photoprint.unipr.it%2Fprint

- The Signature Base String is obtained by concatenating the uppercase HTTP method (GET, POST, ...), the normalized URL, and the normalized parameters with the ‘&’ sign
- In our example, the Signature Base String is:

GET**&**http%3A%2F%2Fwww.photoprint.unipr.it%2Fprint**&**oauth_consumer_key%3Dabcde%26oauth_nonce%3Dxyzxyz%26oauth_signature_method%3DHMAC-SHA1%26oauth_timestamp%3D1369735200%26oauth_token%3Dact123%26oauth_version%3D1.0%26size%3Dmedium%26user%3D12345

Signing request example

- The final step is to create the digital signature of the request
- First, the HMAC key K is computed by concatenating the client secret, the ‘&’ sign, and the access token secret

zyxwv&act456

- The signature is calculated as follows:
  1. HMAC of the Base Signature String using SHA-1 hash and K
  2. transform as string using Base64 encoding

3xkIuqoERka5vNmX4Z25wtAxYdw=

In case of RSA-SHA-1, the Base Signature String is signed using the consumer (private) key

In case of PLAINTEXT, the signature is the HMAC key K (no Base Signature String) - must use TLS

- The digital signature is added to the request in the oauth_signature parameter

Dr. Ing. Simone Cirani
Università degli Studi di Parma

Signing request example

- The final request sent to the Service Provider will be

```
GET /print?user=12345&size=medium HTTP/1.1Host:
www.photoprint.unipr.it:80Authorization: OAuth
realm="http://www.photoprint.unipr.it/print",   oauth_consumer_key="abcde",
oauth_token="act123",   oauth_nonce="xyzxyz",
oauth_timestamp="1369735200",   oauth_signature_method="HMAC-SHA1",
oauth_version="1.0",   oauth_signature="3xkIuqoERka5vNmX4Z25wtAxYdw%3D"
```

Dr. Ing. Simone Cirani
Università degli Studi di Parma

Signed Request Verification

- Upon receiving the request, the Service Provider will verify that the request has been correctly constructed, that the sender who it claims to be, and therefore can decide whether to serve the request or not
- The verification process performs the same steps as the signing process in order to produce the same Base Signature String and then verify if the key used for signing is the one that was constructed using the consumer secret and access token secret, which means that the sender is verified, and the request was not tampered with during transmission

Dr. Ing. Simone Cirani
Università degli Studi di Parma

2-legged and 3-legged Authentication

- The procedure explained so far is called a 3-legged authentication
- 3-legged authentication means that there are 3 parties involved in the scenario (Service Consumer, Service Provider, and User)
- Each leg relates to a party involved in the procedure
- When only 2 parties are involved, the procedure is called 2-legged authentication
  - the Service Consumer coincides with the User
  - this happens when the consumer key and secret are directly associated with the user's account
- *n*-legged authentication involves *n* parties; this can imply the delegation of the grant by the client to additional clients

Dr. Ing. Simone Cirani
Università degli Studi di Parma

OAuth 2.0

- OAuth 2.0 is the next evolution of the original OAuth protocol
- Focus on client developer simplicity
- Provides specific authorization flows for
  - web applications
  - desktop applications
  - mobile phones
  - living room devices

Dr. Ing. Simone Cirani
Università degli Studi di Parma

Where to go from here?

- Coding example: we'll create a simple Twitter application which will access personal data
- A list of libraries for different programming languages is available at http://oauth.net/code/
- Client applications:
  - Social Network integration
  - Social data mining and analysis
  - ...
- Service Providers:
  - create your own online service and protect your APIs with OAuth
- *OAuth as a Service (available thesis projects)*