Đăng nhập một lần và ủy quyền

* OpenID
* Oauth
* SSO

Nội dung:

[1]

<http://www.codehub.vn/OpenID-va-OAuth-Khac-Nhau-Nhu-The-Nao>

Hai khái niệm cơ bản

+ Authentication (xác thực) là việc xác thực danh tính của một người nào đó. Ví dụ như khi bạn thực hiện việc đăng nhập vào một tài khoản trên một website sử dụng username và password thì việc đăng nhập này dùng để xác thực danh tính người dùng.

+ Authorization (trao quyền) là việc người dùng U trao cho một ứng dụng A sử dụng các quyền của người dùng U này có trên một ứng dụng S. Ví dụ khi bạn trao quyền cho một website xem phim (ứng dụng A) quyền lấy các thông tin như email, họ và tên, ngày tháng năm sinh… của mình có trên Facebook (ứng dụng S). Tất nhiên khi đó bạn cần phải có một tài khoản Facebook trước đó.

Hiểu về hai khái niệm authentication và authorization ở trên giúp chúng ta dễ dàng tiếp cận tới các khái niệm là SSO, OpenID và OAuth hơn.

[2] SSO

<http://antoanthongtin.vn/Detail.aspx?CatID=8ab90f49-a562-4157-a607-d2474bf129a9&NewsID=1caffb79-e759-4855-91fc-76840dfc5424>

<http://aita.gov.vn/tin-tuc/2100/mot-vai-net-co-ban-ve-co-che-dang-nhap-mot-lan-sso-single-sin-on>

Giải pháp kiểm soát truy nhập hệ thống SSO

Với người dùng, việc mỗi lần sử dụng dịch vụ mới lại phải đăng ký thông tin tên đăng nhập và mật khẩu cho hệ thống xác thực khiến họ mệt mỏi khi phải nhớ quá nhiều thông tin tài khoản khác nhau và phải đăng nhập nhiều lần. Người dùng sẽ có xu hướng khai báo cùng một thông tin cho tài khoản ở nhiều dịch vụ. Đây là việc vô cùng nguy hiểm vì nếu bị lộ thông tin ở một dịch vụ, người dùng cũng có thể bị chiếm đoạt tài khoản ở các dịch vụ khác dùng chung thông tin này.

Với các nhà cung cấp dịch vụ, họ ngày càng có xu hướng mở rộng dịch vụ qua nhiều lĩnh vực khác nhau tạo thành các dịch vụ con. Các dịch vụ con này hoạt động gần như độc lập với nhau nên việc quản lý người dùng cũng rất phức tạp: quản lý riêng thì không tận dụng được sức mạnh của cụm dịch vụ mà quản lý chung thì cơ chế thế nào? Người quản trị phải đối mặt với một loạt vấn đề như bảo mật, mã hóa, lưu trữ cơ sở dữ liệu. Họ luôn nhận được các yêu cầu tạo lại mật khẩu và kích hoạt lại tài khoản người dùng, do họ gặp phải các sự cố về thông tin đăng nhập.

Xuất phát từ nhu cầu cả hai phía đó, các hệ thống cung cấp xác thực một lần (SSO) cho tất cả các tài khoản có yêu cầu bảo mật khác nhau ra đời. Giải pháp này vừa đảm bảo tính thuận tiện, vừa tăng độ bảo mật khi sử dụng. Hơn nữa, áp lực trong việc quản lý tài khoản đối với nhà quản trị hệ thống có thể được giảm xuống.

SSO là gì?

Single Sign On (SSO) hay xác thực một lần là cơ chế xác thực (authentication) yêu cầu người dùng đăng nhập để xác thực chỉ một lần rồi sau đó có thể truy cập vào nhiều ứng dụng trong một phiên xác thực. Ban đầu SSO chủ yếu dùng trong nội bộ một dịch vụ có nhiều dịch vụ con nhưng dần dần chúng được mở rộng ra cho cả dịch vụ bên ngoài sử dụng. Cơ chế hỗ trợ SSO nổi bật nhất là OpenId và SAML.

Ưu điểm của giải pháp SSO

+ Đối với người dùng:

Hệ thống SSO cung cấp cho người dùng công cụ xác thực đăng nhập một lần. Công cụ này xác thực người dùng đăng nhập vào mọi hệ thống có tích hợp giải pháp SSO, vì vậy, người dùng chỉ cần nhớ một cặp tên truy cập/mật khẩu duy nhất.

Khi cập nhật mật khẩu và tên đăng nhập, người dùng chỉ cần cập nhật một lần cho toàn bộ các hệ thống có yêu cầu xác thực người dùng.

Khi đã đăng nhập vào một hệ thống chứa các hệ thống con có tích hợp giải pháp SSO, người dùng có thể dễ dàng truy nhập vào các hệ thống này mà không cần phải xác thực lại trong một khoảng thời gian nào đó, tùy thuộc vào chính sách quản trị.

+ Đối với nhà cung cấp dịch vụ, quản trị hệ thống:

Người quản trị chỉ cần bảo mật và quản lý thông tin đăng ký của người dùng một lần, vì vậy có thể giảm dung lượng cơ sở dữ liệu và tránh được các xung đột nảy sinh do phải xử lý mật khẩu của các hệ thống khác nhau, tăng khả năng mở rộng và triển khai các chiến lược bảo mật.

Người quản trị có thể thay đổi và cập nhật thông tin được bảo mật của người dùng khi cần thiết, một cách dễ dàng hơn so với việc thay đổi ở từng hệ thống riêng lẻ mà người dùng đó được phép truy cập. Điều này rất hữu ích khi người dùng thay đổi vị trí của mình với các cấp độ bảo mật khác nhau.

Nhược điểm của giải pháp SSO

Việc xử lý một loạt các hệ thống con có tích hợp giải pháp SSO không đơn giản, bởi vì mỗi hệ thống con có thể hoạt động trên các nền phần cứng và phần mềm khác nhau. Vì vậy, khi cài đặt sẽ giải quyết nhiều vẫn đề liên quan đến tính tương thích và đồng bộ giữa các hệ thống.

Các kỹ thuật mang tính mở áp dụng với hệ thống hoặc chưa được chuẩn hóa, hoặc chưa được cung cấp, có thể gây mâu thuẫn và không tương thích với các sản phẩm khác.

Khi đã được xác thực thành công, người dùng có thể truy nhập vào nhiều ứng dụng trong hệ thống. Bởi vậy, nếu không được bảo mật tốt tại các hệ thống con, thì đối tượng tấn công khi tiếp cận vào một ứng dụng có thể tấn công vào toàn bộ hệ thống.

Thách thức đối với giải pháp SSO là làm việc theo kiến trúc, các cơ chế bảo mật độc lập đối với các nền ứng dụng. Để thuận tiện và đơn giản hơn khi giải quyết nhược điểm của SSO, người ta thường đi theo hướng xây dựng các ứng dụng sử dụng cơ sở hạ tầng bảo mật, xác thực người dùng chung, thông suốt giữa các ứng dụng. Điều này đòi hỏi phải có định dạng chung khi thể hiện các thông tin xác thực hoặc ủy nhiệm, sao cho tất cả các ứng dụng đều có thể hiểu và chấp nhận chúng. Đồng thời, phải đảm bảo được các ủy nhiệm là đúng đắn.

Mặc dù có một số nhược điểm, nhưng giải pháp SSO vẫn đang được triển khai rộng rãi ở nhiều nước trên thế giới, trong nhiều hệ thống thông tin có nhu cầu xác thực và bảo mật. Nó được đánh giá là một trong các giải pháp hiệu quả, tiện dụng và kinh tế khi triển khai trên diện rộng. Các vấn đề nêu trên sẽ được giải quyết trong thời gian tới khi chúng ta tăng cường chuẩn hóa và có các chính sách cụ thể cho vấn đề này.

Một số mô hình SSO thực tế

Các hệ thống xác thực SSO đang được sử dụng rộng rãi là: CAS (Central Authentication Service), WebAuth. RSA Single Sign On Manager, Open Single Sign - On (OpenSSO) hoạt động dựa trên Token, Java Open SSO (JOSSO).

+ Open SSO Enterprise

+ Central Authenticate Service (CAS)

[3] OpenID

<https://viblo.asia/p/tim-hieu-ve-openid-WwXKPXZMKEJ>

<http://openid.yahoo.com>

1 - OpenID là gì?

OpenID là một tiêu chuẩn mở và là một giao thức xác thực phân cấp.

Được phát triển bởi tổ chức phi lợi nhuận OpenID Foundation, OpenID cho phép user có thể được authen bởi rất nhiều website (replying parties) sử dụng service của bên thứ 3. Nó giảm được việc phải thiết lập riêng logic signup/login cho mỗi website, cho phép các user có thể login tới nhiều website mà không cần phải có những định danh và password riêng cho mỗi site.

Bạn có thể chọn việc liên kết thông tin với OpenID của bạn – thông tin này sẽ được share tới các website mà bạn đã truy cập (ví dụ như tên hoặc email). Với OpenID, bạn có thể kiểm soát được lượng thông tin mà bạn muốn cho phép các website bạn đến thăm biết được.

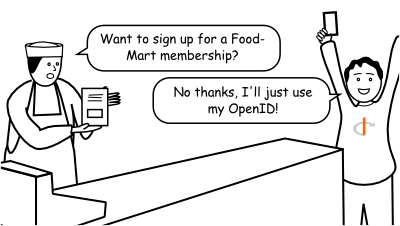
Với OpenID, chỉ duy nhất identity provider quản lý password, và provider này sẽ confirm identity của bạn tới các website mà bạn đến thăm, ko có một website nào có thể biết được password của bạn - đây là một yếu tố bảo mật rất cao.

OpenID nhanh chóng nhận được sự công nhận và sự dụng của cộng đồng web, với hơn 1 tỷ OpenID account được thiết lập và 50,000 webstie sử dụng phương thức OpenID để login. Một vài tổ chức lớn cho phép hoặc phát hành OpenIDs or tái sử dụng lại OpenID ví dụ như Google, Facebook, Yahoo!, Microsoft, AOL, MySpace, Sears, Universal Music Group, France Telecom, Novell, Sun, Telecom Italia v.v....



2 – Tại sao nên sử dụng OpenID

A – Đăng ký thành viên nhanh và dễ dàng



Với hình ảnh hoạt hình bên cạnh, bạn có thể hình dung sự tiện lợi này. Một người thu tiền hỏi anh khách hàng muốn đăng kí thành viên cho Food-Mart ko ? và khách hàng trả lời rất hào hứng "Nope. Tôi sẽ sử dụng OpenID của tôi". Một OpenID chính là cách để định danh chính bạn ko quan trọng bạn ghé thăm website nào. Nó như kiểu là bằng lái xe cho toàn bộ Internet vậy. Nhưng còn hơn thế nữa là bởi vì bạn có thể control được thông tin liên kết tới OpenID như là tên của bạn, địa chỉ mail của bạn, và bạn có thể chọn website nào được quyền thấy nó ? cái nào ko ? Điều này cũng đồng nghĩa với việc những website sử dụng OpenID sẽ ko hỏi đi hỏi lại bạn your name, your mail address lần nào cũng vậy.

B – Login nhanh và dễ dàng

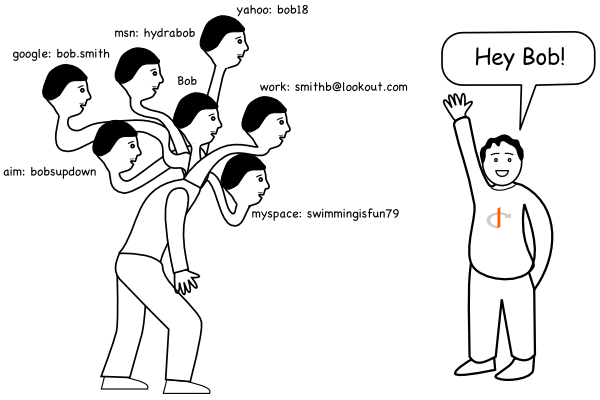
OpenID also simplifies signing in.

OpenID còn rất dễ dàng để login. Với OpenID bạn chỉ cần nhớ 1 username và 1 password. Bởi vì bạn login vào website với OpenID của bạn, cho nên bạn chỉ cần bảo mật với OpenID là đủ rồi. Password này của bạn thực tế là chỉ được quản lý bởi OpenID provider của bạn, ko phải là các website bạn đang login, khi bạn login vào OpenID provider sẽ bảo với các website đó là bạn đang login đấy và bạn là ai. Ko có bất kì một website nào nhìn được password của bạn, cho nên bạn ko cần phải quá bận tâm về vấn đề bảo mật khi sử dụng OpenID.

C – Tiệm cận đến khái niệm “web identity”

Bởi vì OpenID định danh bạn là duy nhất trên Internet, một khi bạn thiết lập mình như là người sử dụng OpenID, bất cứ khi nào ai đó nhìn thấy OpenID của bạn được sử dụng, bất cứ nơi nào trên Internet, họ sẽ biết rằng đó chính là bạn. Tương tự như vậy, nếu bạn truy cập vào một trang web mới và thấy rằng một người nào đó với OpenID của bạn bạn, bạn có thể gần như chắc chắn rằng đó là bạn của bạn.

Điều đó có thể khiến bạn lo lắng rằng OpenID là sẽ làm cho tất cả các hoạt động trực tuyến của bạn trở nên quá rõ ràng và dễ dàng bị phát hiện. Hãy yên tâm, mặc dù OpenID thống nhất thông tin về bạn, nhưng nó chỉ hợp nhất thông tin mà bạn đã cho phép public. Bạn có thể chọn, sử dụng OpenID, thông tin nào ? và thông tin đó ai sẽ thấy được



D – Liệu OpenID có an toàn?

OpenID là thực sự cũng ko kém bảo mật hơn or là an toàn hơn so với những gì bạn sử dụng ngay bây giờ. Sự thật là nếu ai đó có được tên truy cập và mật khẩu của OpenID của bạn, họ có thể chiếm đoạt định danh của bạn. Hầu hết các trang web cung cấp việc reset password thông qua email trong trường hợp nếu bạn đã quên nó, có nghĩa là nếu một người nào đó có được tài khoản e-mail của bạn, họ cũng có thể làm giống như trong trường hợp họ có được tên truy nhập và mật khẩu của OpenID của bạn. Họ có thể kiểm tra các trang web mà họ nghĩ rằng bạn có một tài khoản và yêu cầu thay mật khẩu bị quên. Tương tự như vậy, nếu một ai đó có quyền truy cập vào OpenID của bạn, họ có thể tìm kiếm trên Internet đến những nơi mà họ nghĩ rằng bạn có tài khoản và đăng nhập như bạn ... không có gì khác biệt về bảo mật ở đây.

Cho nên, ko liên quan đến việc bạn sử dụng OpenID hay ko, bạn nên cẩn thận về tài khoản của bạn - username và pwd. Khi bạn gõ username và pwd, hãy chắc chắn ràng bạn đang ở trên website mà bạn nghĩ là bạn muốn vào

Có nên giao toàn bộ định danh cho chỉ 1 website?

Tùy bạn, nếu bạn thích, bạn cũng có thể có nhiều OpenID, mỗi 1 cái lại có 1 thông tin khác nhau của bạn. Thực tế là có nhiều webstie cho phép bạn liên kết nhiều OpenID tới cùng 1 account. Nhưng việc làm này sẽ phá hỏng khái niệm đơn giản hóa của "chỉ có 1 username và pwd".

3 – OpenID Authentication

OpenID Authentication là trái tim của OpenID, nó bao hàm 3 concept chính:

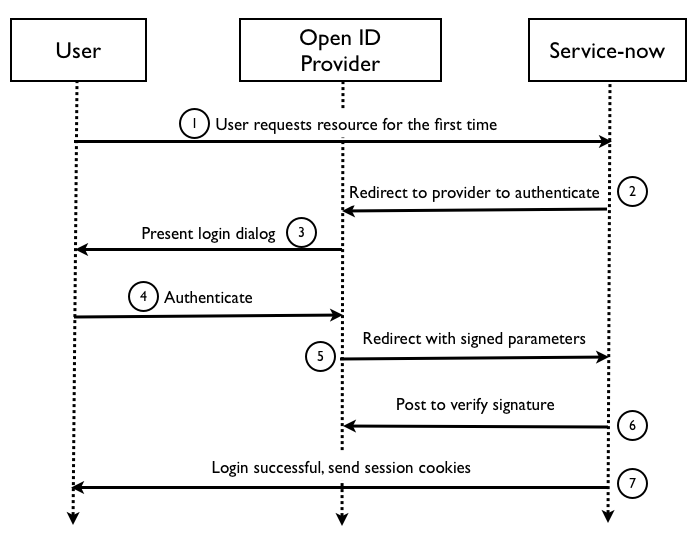
* The OpenID Indentifier: một string xác định duy nhất cho người sử dụng.
* The OpenID Relying Party (RP): một nguồn tài nguyên trực tuyến (có thể là một trang web, nhưng nó có thể là một tập tin, hình ảnh, hoặc bất cứ điều gì mà bạn muốn kiểm soát quyền truy cập vào) có sử dụng OpenID để xác định những người có thể truy cập nó.
* The OpenID Provider (OP): một trang web mà người sử dụng có thể yêu cầu một OpenID và sau đó đăng nhập và xác thực danh tính của người sử dụng RP.

OpenID hoạt động như thế nào?

Giả sử một người dùng đang cố gắng truy cập tài nguyên là một phần RP của website, và RP sử dụng OpenID. Để truy cập vào các nguồn tài nguyên, người sử dụng phải xuất trình OpenID của mình trong một form có thể được công nhận (bình thường) như OpenID. OpenID được mã hóa với vị trí của OP. RP sau đó lấy định danh của người dùng và chuyển hướng người dùng đến OP, nơi đó user sẽ được yêu cầu để chứng minh yêu cầu của mình sử dụng ID đó.

Hãy cùng điểm qua các thành phần chi tiết của OpenID và mỗi vài trò của nó trong cả quá trình xử lý.

<http://wiki.servicenow.com/index.php?title=OpenID#gsc.tab=0>



1. User requests resource for the first time: người dùng yêu cầu một tài nguyên từ RP (Service Now) lần đầu tiên chưa có xác thực từ OP (yêu cầu không chứa OpenID URL parameters)
2. Redirect to provider to authenticate: RP chuyển yêu cầu tới OP với yêu cầu xác thực
3. Present login dialog: OP đòi người dùng cung cấp tài khoản
4. Authenticate: Người dùng đăng nhập tài khoản và xác thực thành công
5. Redirect with signed parameters: OP gửi một yêu cầu, bao gồm cả OpenID URL tới RP
6. Post to verify signature”: RP xác nhận lại yêu cầu với OP
7. Login successful, send session cookies: RP đã được người dùng đăng nhập với tham số “user\_key”.

[4] OAuth

<https://laptrinhgo.blogspot.com/2016/11/bai-31-xac-thuc-mot-lan-va-cap-quyen.html>

<https://tinhte.vn/threads/oauth-la-gi-chuyen-gi-se-xay-ra-khi-ban-dang-nhap-website-bang-nick-facebook-google-twitter.1949091/>

<https://github.com/maraujop/requests-oauth2>

requests-oauth2 wants to provide the simplest and easiest way to do OAuth2 in Python.

<https://kipalog.com/posts/Tro-lai-co-ban--Oauth-2>

<http://requests-oauthlib.readthedocs.io/en/latest/oauth2_workflow.html>

Ưu điểm, ví dụ và cách hoạt động của OAuth

Vài năm trở lại đây, các trang web và ứng dụng di động thường cho phép người dùng sử dụng tài khoản các mạng xã hội hoặc các dịch vụ phổ biến khác để sử dụng dịch vụ thay vì phải đăng ký tài khoản

OAuth

Là cơ chế Authorization (cấp quyền)

Sau khi người dùng đăng nhập thành công với dịch vụ cung cấp OAuth, dịch vụ này sẽ trả về cho dịch vụ sử dụng một tok. en để có thể truy cập thông tin người dùng trên dịch vụ cung cấp.

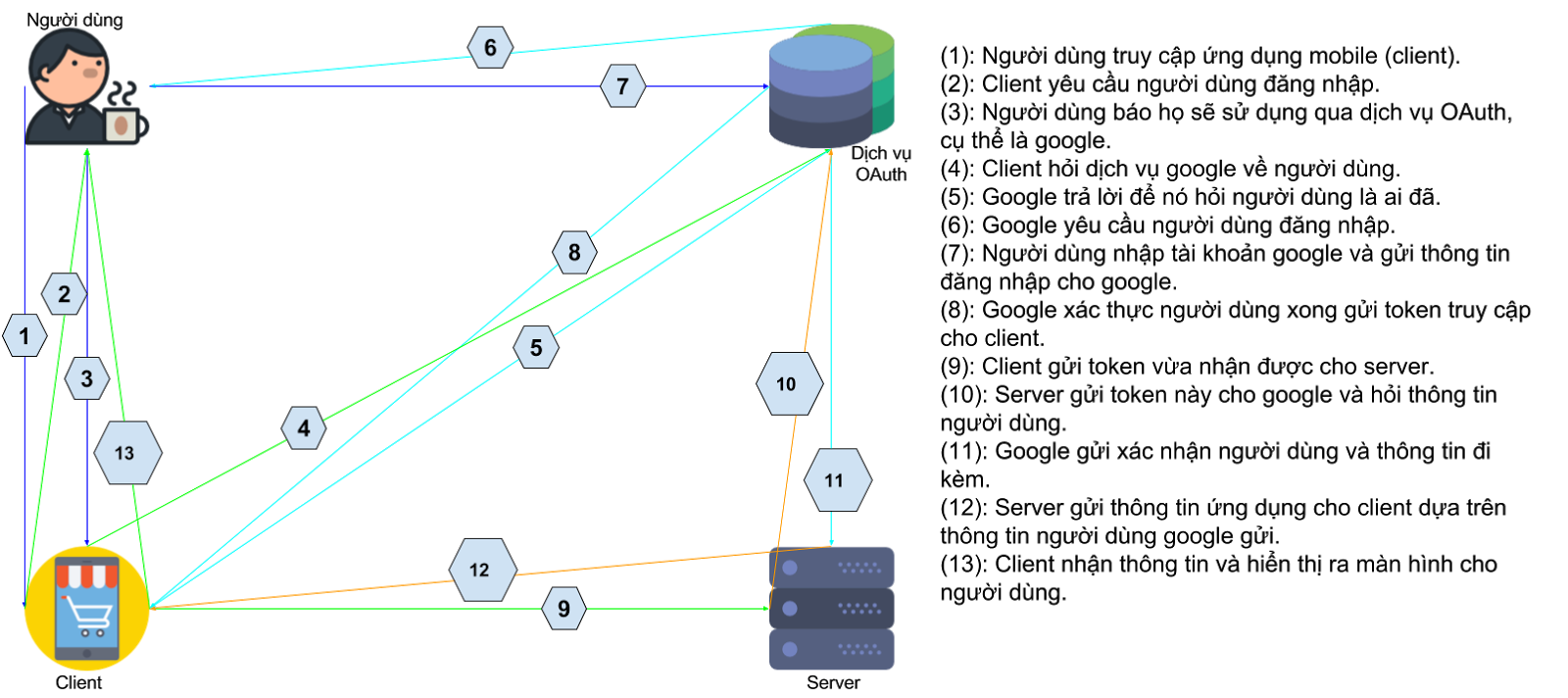
Điểm khác biệt giữa OAuth và SSO là cơ chế cấp quyền như OAuth chỉ cho phép dịch vụ sử dụng tiếp cận một số thông tin giới hạn của người dùng và người dùng quyết định cho hay không. (Dịch vụ xác thực thì cho phép toàn quyền tiếp cận).

Hiện nay có hơn 50 nhà cung cấp dịch vụ cấp quyền hỗ trợ OAuth. Tiêu biểu có thể kể đến Google, Facebook, Twitter, Microsoft, Amazon, Dropbox, Linkedin, Foursquare, Github, Instagram, Netflix, …

OAuth 2.0

Được làm mới từ phiên bản OAuth 1.0

Cơ chế cấp quyền OAuth 2.0 được minh họa qua ví dụ sau



Qua các bước ở trên chung ta thấy có 4 đối tượng trong quá trình cấp quyền với OAuth 2.0:

* Người dùng: trong thuật ngữ OAuth gọi là Resource Owner bởi người dùng chính là chủ tài nguyên
* Server: là nơi cung cấp dịch vụ ứng dụng cho người dùng
* Client: là trình duyệt, ứng dụng mobile hoặc dịch vụ khác
* Dịch vụ cấp quyền: là nơi lưu thông tin tài khoản người dùng, trong thuật ngữ OAuth gọi là Authorization Server. Còn gọi là OAuth server.

Trước tiên, để có thể yêu cầu OAuth Server cung cấp thông tin, client phải đăng ký với nó để được cấp quyền truy cập thông tin người dùng qua cặp ID và khóa. Cặp ID và khóa này là duy nhất trên OAuth server. Trong quá trình đăng ký, client phải cung cấp một số thông tin như tên dịch vụ, địa chỉ trang web, các quyền muốn truy cập,…

Có 4 hình thức cấp quyền khác nhau và mỗi cách sẽ có bước xử lý khác nhau:

Ủy quyền ngầm (Implicit) là hình thức cấp quyền phổ biến được minh họa trong ví dụ bên trên.

Cấp quyền OAuth 2.0 với Go

Go – dự án mã nguồn mở, được phát triển bởi các nhân viên Google; nhiều nét giống C; thích hợp cho việc xây dựng các phần mềm back-end, phần mềm máy chủ, và các công cụ hệ thống.

Thử kết nối sử dụng OAuth 2.0 với Facebook, Google Plus, Twitter qua goth – một packet hỗ trợ việc sử dụng OAuth 2.0 trong Go.

[5]

<https://stackoverflow.com/questions/1087031/whats-the-difference-between-openid-and-oauth>

