1. Giới thiệu

Trong mô hình chứng thực client-server truyền thống, client yêu cầu truy cập một nguồn tài nguyên được bảo vệ trên một server bằng cách chứng minh rằng được cấp phép truy cập từ chủ sở hữu tài nguyên. Trong trường hợp muốn cho ứng dụng bên thứ ba cần truy cập vào những tài nguyên hạn chế này, chủ sở hữu tài nguyên ủy nhiệm cho bên thứ ba. Mô hình này có một vài giới hạn.

* Những ứng dụng của bên thứ ba phải lưu lại ủy quyền của người sở hữu tài nguyên để sử dụng về sau, thông thường ở dạng text.
* Cần các máy chủ để chứng thực mật khẩu, cho dù tính kém bảo mật cố hữu của mật khẩu.
* Những ứng dụng của bên thứ ba có thể thu thập toàn bộ nguồn tài nguyên giới hạn, trong thời gian vô hạn, bất chấp nỗ lực giới hạn thời gian truy cập hay giới hạn phạm vi truy cập vào nguồn tài nguyên này.
* Chủ sở hữu nguồn tài nguyên không thể bỏ quyền truy cập của môt bên riêng lẻ mà không bỏ quyền truy cập của toàn bộ các bên, và chỉ làm được bằng cách thay đổi mật khẩu truy cập của bên thứ ba.
* Sự thỏa hiệp của bất kỳ ứng dụng nào của bên thứ ba dẫn đến sự thỏa hiệp của mật khẩu người dùng cuối và tất cả dữ liệu được bảo vệ bởi mật khẩu đó.

OAuth giải quyết những vấn đề này bằng cách giới thiệu một tầng chứng thực và chia nhỏ các vai trò của client, với người sở hữu tài nguyên. Với OAuth, client yêu cầu truy cập vào những tài nguyên được kiểm soát bởi chủ sở hữu và lưu trữ bởi máy chủ tài nguyên, và phát hành những tập ủy quyền khác với của chủ sở hữu.

Thay vì sử dụng ủy quyền được cấp bởi chủ sở hữu để truy cập các tài nguyên hạn chế, client nhận được một khóa truy cập -- một chuỗi chỉ ra phạm vi, thời gian truy cập và một số thuộc tính khác. Khóa truy cập được cấp cho các clients của một ứng dụng bên thứ ba bởi một máy chủ chứng nhận với sự đồng thuận của chủ sở hữu tài nguyên. Client sử dụng khóa truy cập này để truy cập vào những tài nguyên hạn chế được lưu trữ trên máy chủ tài nguyên.

Ví dụ, một người dùng cuối (chủ tài nguyên) có thể cấp quyền cho một dịch vụ in ấn (client) truy cập vào những bức ảnh được bảo vệ của cô ta lưu trữ tại dịch vụ chia sẻ (máy chủ tài nguyên), mà không phải chia sẻ tên truy cập và mật khẩu với dịch vụ in ấn. Thay vào đó, cô ta ủy quyền trực tiếp một máy chủ tin cậy (máy chủ chứng thực) với dịch vụ chia sẽ, thứ sẽ phát hành giấy phép ủy nhiệm truy cập đặc biệt cho dịch vụ in ấn (khóa truy cập).

Đặc tả này thiết kế để dùng với HTTP [[RFC2616](https://tools.ietf.org/html/rfc2616)]. Việc sử dụng OAuth với những giao thức khác với HTTP là ngoài phạm vi của đặc tả.

Giao thức OAuth 1.0 [[RFC5849](https://tools.ietf.org/html/rfc5849)] , được phát hành như một tài liệu truyền thông, thứ được coi như là kết quả của một cộng đồng nhỏ. Những đặc tả tiêu chuẩn này xây dựng trên kinh nghiệm vận hành OAuth 1.0, cũng như các yêu cầu mở rộngtừ quá trình sử dụng được thu thập rộng rãi từ cộng đồng IETF. OAuth 2.0 không tương thích ngược với OAuth 1.0. Cả hai phiên bản có thể cùng tồn tại và các bản cài đặt có thể quyết định hỗ trợ cả hai. Tuy nhiên, mục đích của đặc tả này là bản cài đặt mới hỗ trợ OAuth 2.0 và bản OAuth 1.0 chỉ nhằm hỗ trợ những hệ thống đã vận hành. Cách cài đặt giao thức OAuth 2.0 có rất ít điểm chung với OAuth 1.0. Những nhà phát triển đã quen với OAuth 1.0 nên tiếp cận tài liệu này mà không liên tưởng đến cấu trúc cũng như chi tiết của bản OAuth 1.0.

* 1. Vai trò

OAuth định nghĩa 4 vài trò:

1. Chủ sở hữu tài nguyên

Một thực thể có khả năng cấp quyền truy cập vào nguồn tài nguyên được bảo vệ. Trong trường hợp chủ sở hữu là mộ người, thì tương đồng với người dùng cuối.

1. Máy chủ tài nguyên

Máy chủ lưu trữ nguồn tài nguyên được bảo vệ, có khả năng chấp thuận và phản hồi một yêu cầu sử dụng khóa truy cập.

1. Client

Một ứng dụng tạo ra yêu cầu truy cập nguồn tài nguyên lấy danh nghĩa của người sở hữu bằng sự ủy quyền của họ. Thuật ngữ “client” không mang một đặc trưng cụ thể nào (e.g, cho dù ứng dụng chạy trên máy chủ, máy bàn, hay những thiết bị khác)

1. Máy chủ chứng thực

Máy chủ phát hành khóa truy cập cho client sau khi chứng nhận thành công với chủ sở hữu và nhận được sự ủy quyền.

Sự giao tiếp giữa máy chủ chứng thực và máy chủ tài nguyên không nằm trong phạm vi của đặc tả này. Máy chủ chứng thực có thể nằm trên cùng một máy với máy chủ tài nguyên hoặc có thể nằm trên hai máy riêng rẽ. Một máy chủ chứng thực đơn lẽ có thể phát hành khóa-truy-cập cho nhiều máy chủ tài nguyên.

* 1. Giao thức.

+--------+ +---------------+

| |--(A)- yêu cầu chứng thực ->| chủ sở-hữu |

| | | tài nguyên |

| |<-(B)— chứng nhận ủy quyền ---| |

| | +---------------+

| |

| | +---------------+

| |--(C)— chứng nhận ủy quyền -->| máy chủ |

| Client | | chứng-thực |

| |<-(D)----- khóa truy cập ------| |

| | +---------------+

| |

| | +---------------+

| |--(E)----- khóa truy cập ----->| máy chủ |

| | | tài nguyên |

| |<-(F)-- tài nguyên giới hạn ---| |

+--------+ +---------------+

Mô hình trên chỉ ra sự giao tiếp của bốn vai trò thông qua các bước:

1. Client yêu cầu ủy quyền từ chủ sở hữu tài nguyên. Yêu cầu ủy quyền có thể gửi trực tiếp đến chủ sở hữu như trong mô hình này, hoặc gián tiếp thông qua máy chủ chứng thực làm trung gian.
2. Client nhận một chứng nhận (thể hiện sự ủy quyền của chủ sở hữu tài nguyên) thể hiện bằng một trong bốn cách ủy quyền được mô tả trong đặc tả này hoặc một cách riêng biệt nào đó. Chứng nhận này phụ thuộc vào phương pháp được client sử dụng và những cách ủy nhiệm được máy-chủ-chứng-thực hỗ trợ.
3. Client yêu cầu một khóa truy cập từ máy-chủ chứng-thực bằng cách trưng ra ủy nhiệm của chủ sở hữu.
4. Máy-chủ chứng-thực chứng nhận client và kiểm tra ủy quyền của chủ sở hữu, nếu hợp lệ, phát hành một khóa truy-cập.
5. Client yêu cầu những tài nguyên được bảo vệ từ máy chủ tài nguyên bằng cách cung cấp khóa truy-cập.
6. Máy chủ tài nguyên kiểm tra khóa truy cập, nếu hợp lệ thì phục vụ yêu cầu của client.

Cách thức để client nhận được ủy quyền của chủ sở hữu tài-nguyên (bước (A) và (B) ) gián tiếp qua máy-chủ chứng-thực được mô tả trong phần 4.1

* 1. Cấp phép (Authorization Grant)

Môt sự ủy quyền là một giấy phép thể hiện sự chấp thuận của chủ sở hữu tài-nguyên (để truy-cập nguồn tài nguyên được bảo vệ) được dùng để client nhận một khóa truy-cập. Đặc tả này xác định bốn cách để cấp phép – Mã chứng nhận (Authorization Code), Cấp phép ngầm (implicit), resource owner password credentials, và client credentials --

* + 1. Mã xác-thực (Authorization Code)

Mã chứng nhận lấy được bằng cách sữ dụng gián tiếp một máy-chủ chứng-thực. Thay vì yêu cầu chứng nhận trực tiếp từ chủ sở hữu, client hướng chủ sở-hữu đến một máy chủ chứng-thực thông qua user-agent của nó (định nghĩa tại [[RFC2616](https://tools.ietf.org/html/rfc2616)]), sau đó user-agent lại chuyên hướng chủ sở hữu lại client với một mã chứng nhận.

Trước khi chuyển hướng chủ sở hữu trở lại client với một mã- xác-thực, máy chủ chứng-thực xác nhận chủ sở hữu và nhận một ủy quyền. Bởi vì chủ sở hữu chỉ xác nhận với máy chủ chứng-thực, ủy nhiệm thư của người dùng sẽ không chia sẻ với client.

Mã chứng nhận có một vài lợi ích về bảo mật, như khả năng chức thực client, cũng việc chuyển khóa-truy-cập trực tiếp tới client mà không phải chuyển nó tới user-agent của chủ sở hữu và chứa nhiều nguy cơ lộ ra ngoài, bao gồm cả chủ sở hữu.

1.3.2 Chứng nhận ngầm (Implicit)

Chứng nhận ngầm là một chứng nhận dùng mã- xác-thực được đơn giản hóa theo hướng tối giản việc cài đặt client trên một browser sử dụng một ngôn ngữ kịch bản ví dụ như Javascript. Trong chứng thực ngầm, thay vì phát hành cho client một mã-chứng-nhận, client sẽ nhận được một khóa-truy-cập trược tiếp (như là kết quả của sự ủy quyền của chủ sở hữu). Cách xác thực là ngầm định, như thể không có ủy nhiệm thư nào được phát hành (ví dụ như một mã-xác-thực) (và sau đó dùng để lấy khóa-truy-cập).

Khi phát hành một khóa-truy-cập trong suốt quy trình xác thực ngầm, máy chủ chứng-thực không xác nhận client. Trong vài trường hợp, việc xác định client có thể kiểm tra thông qua URI được dùng để chuyển khóa-truy-cập đến client. Khóa-truy-cập có thể để lộ với chủ sở hữu hoặc những ứng dụng khác khi truy cập tác-nhân tương-tác của chủ sở hữu tài nguyên.

Chứng nhận ngầm cải tiến sự phản hồi và hiệu quả với một vài client (như client được cài đặt bên trong một ứng dụng trình duyệt), khi nó giảm số vòng lặp cần để lấy khóa-truy-cập. Tuy nhiên, lợi thế này có thể trở thành gánh nặng với việc bảo mật thông tin khi sử dụng chứng nhận ngầm, nhưng những thứ mô tả tại phần 10.3 và 10.16, đặc biệt khi việc cấp quyền bằng mã-xác-thực là khả thi.

1.3.3 Resource Owner Password Credentials

Chức thực mật khẩu của chủ sở hữu tài nguyên (i.e, username và password) có thể dùng trực tiếp như một kiểu sác nhận để lấy khóa-truy-cập. Các ủy nhiệm thư chỉ nên được dùng khi có sự đồng thuận cao về sự tin tưởng giữa chủ sở hữu tài nguyên và client (e.g., client là một phần của hệ điều hành của thiết bị hoặc một ứng dụng có đặc quyền cao), và khi các phương thức xác nhận khác không khả thi (như là dùng mã xác-thực).

Dù phương pháp chứng nhận này đòi hỏi client truy cập trực tiếp vào ủy nhiệm thư của chủ sở hữu, ủy nhiệm thư của người sở hữu được cùng cho một yêu cầu đơn lẻ và được chuyển đổi thành khóa-truy-cập. Loại chứng thực này có thể loại trừ nhu cầu lưu trữ chứng thư của người sở hữu để dùng trong tương lai, bằng sự chuyển đổi chứng -thư thành khóa-truy-cập dài-hạn hoặc khóa-đổi-mới.

1.3.4 Chứng-thư của client

Chứng-thư của client (hoặc dạng khác của việc xác thực client) có thể được dùng như một phương thức xác thực khi phạm vi xác thực bị giới hạn ở những tài-nguyên được bảo vệ dưới sự kiểm soát của client, hoặc những tài-nguyên đã được chấp nhận trước đó của máy chủ chứng thực. Chứng-thư của client được dùng như một phương pháp chứng thực tiêu biểu khi client hoạt động dưới sự ủy quyền của chính nó (client cũng đồng thời là chủ sở hữu) hoặc yêu cầu truy cập vào những tài nguyên đã được sự chấp nhận trước đó của máy-chủ chứng-thực.

* 1. Khóa-truy-cập (Access Token)

Khóa-truy-cập là chứng-thư được dùng trong việc truy cập những tài-nguyên được bảo vệ. Một khóa-truy-cập là một chuỗi đại diện cho việc chứng-nhận được phát hành cho client. Chuỗi này thường không có ý nghĩa với client. Những khóa bao hàm một phạm vi cụ thể và thời hạn của việc truy cập, được cấp phép bởi chủ sở hữu, và yêu cầu bởi máy-chủ tài-nguyên và máy-chủ chứng-thực.

Khóa có thể bao hàm một dấu chỉ nhận dạng người dùng, để trích ra thông tin cấp phép hoặc bao hàm thông tin cấp phép mà có thể kiểm chứng (i.e., một chuỗi khóa chứa một vài dữ liệu và một chữ kí). Các loại chứng-thư bổ xung khác, nằm ngoài phạm vi đặc tả này, có thể cần cho client trong việc sử dụng chuỗi khóa.

Khóa-truy-cập cung cấp một tầng-trừu-tượng , thay thế các cấu trúc xác thực khác (e.g., tên truy cập và mật khẩu) bằng một chuỗi-khóa đơn lẻ mà máy-chủ tài-nguyên có thể hiểu. Sự trừu-tượng này có thể làm cho việc phát hành khóa-truy-cập bị hạn chế hơn việc cấp quyền để lấy chúng, cũng như dỡ bỏ sự cần thiết phải có những máy-chủ tài-nguyên có thể hiểu một lượng lớn các phương thức xác thực.

Khóa-truy-cập có thể có nhiều định-dạng, cấu trúc, và cách thức sử dụng (e.g., các thuộc tính cryptographic) khác nhau tùy vào yêu cầu bảo mật của máy-chủ tài-nguyên. Các thuộc tính của khóa-truy-cập và cách thức dùng để truy cập vào những tài-nguyên được bảo vệ nằm ngoài phạm vi của đặc-tả này và được định nghĩa bởi các đặc-tả liên quan như [[RFC6750](https://tools.ietf.org/html/rfc6750)].

* 1. Khóa-Đổi-Mới

Khóa-đổi-mới là chứng-thư dùng để lấy khóa-truy-cập. Khóa-đổi-mới được phát hành cho client bởi máy-chủ chứng-thực và dùng để đổi lấy khóa-truy-cập mới khi khóa-truy-cập hiện tại hết hạn, hoặc để lấy thêm khóa-truy-cập cho các vùng cụ thể giới hạn hơn ( khóa-truy-cập có thể ngắn hạn hơn và có quyền hạn ít hơn so với khi được cấp phép bởi chủ sở hữu ) . Việc phát hành một khóa-đổi-mới là tùy vào sự thận trọng của máy-chủ chứng-thực . Nếu một máy-chủ chứng-thực phát hành một khóa-đổi-mới , nó được thêm vào khi phát hành khóa-truy-cập.

Một khóa-đổi-mới là một chuỗi diễn đạt sự cấp phép cho client bởi người sở hữu tài-nguyên. Chuỗi này thường vô nghĩa với client. Khóa này bao hàm một dấu chỉ được dùng để lấy ra thông tin cấp phép. Không giống khóa-truy-cập, khóa-thay-mới chỉ dùng duy nhất với máy-chủ chứng-thực và không bao giờ gửi tới máy-chủ tài-nguyên.

+--------+ +---------------+

| |--(A)------- Xác thực --------->| |

| | | |

| |<-(B)----------- Access Token -------------| |

| | & Refresh Token | |

| | | |

| | +----------+ | |

| |--(C)---- Access Token ---->| | | |

| | | | | |

| |<-(D)- Protected Resource --| Resource | | Authorization |

| Client | | Server | | Server |

| |--(E)---- Access Token ---->| | | |

| | | | | |

| |<-(F)- Invalid Token Error -| | | |

| | +----------+ | |

| | | |

| |--(G)----------- Refresh Token ----------->| |

| | | |

| |<-(H)----------- Access Token -------------| |

+--------+ & Optional Refresh Token +---------------+

1. client yêu cầu một khóa-truy cập bằng cách xác thực với máy-chủ chứng-thực và đưa ra một quyết định ủy-quyền.
2. Máy-chủ chứng-thực xác nhận client và kiểm tra quyết định ủy-quyền, và nếu hợp lệ, cung cấp một khóa-truy-cập và một khóa-đổi-mới.
3. Client yêu cầu truy cập nguồn tài-nguyên được bảo vệ từ máy-chủ tài-nguyên bằng cách cung cấp khóa-truy-cập.
4. Máy-chủ tài-nguyên kiểm tra khóa-truy-cập vào nếu hợp lệ thì phục vụ yêu cầu.
5. Bước (C) và (D) lặp lại cho đến khi khóa-truy-cập hết hạn. Nếu Client biết thời hạn của khóa-truy-cập, chuyển đến bước (G); nếu không tạo một yêu cầu truy cập tài nguyên được bảo vệ.
6. Khi khóa-truy-cập không còn hợp lệ, máy-chủ tài-nguyên sẽ trả ra một lỗi khóa-truy-cập không hợp lệ.
7. Client yêu cầu một khóa-truy-cập mới bằng cách chứng-thực với máy-chủ chứng-thực và cung cấp khóa-đổi-mới. Điều kiện để chứng thực client tùy thuộc vào loại client và chính sách an ninh của máy-chủ chứng-thực.
8. Máy-chủ chứng-thực chứng-thực client và kiểm tra khóa-đổi-mới, và nếu hợp lệ, cấp một khóa-truy-cập mới (và, có thể là khóa-đổi-mới mới).

Các bước (C), (D), (E) và (F) nằm ngoài phạm vi của tài liệu này, như được mô tả ở phần 7.

* 1. Phiên bản TLS
  2. Chuyển hướng HTTP
  3. Khả năng tương tác
  4. Ký hiệu:

xem [[RFC2119](https://tools.ietf.org/html/rfc2119)]

1. Đăng kí Client

Trước khi thiết lập giao thức, client đăng kí với máy-chủ chứng-thực. Cách client đăng kí với máy-chủ chứng-thực nằm ngoài phạm vi của tài liệu này, nhưng thông thường sẽ tương tác với người-dùng-cuối thông qua một mẫu đăng kí HTML.

Việc đăng kí của client không đòi hỏi phải tương tác trực tiếp giữa client và máy-chủ chứng-thực. Khi đã được hỗ-trợ bởi máy-chủ chứng-thực, việc đăng kí có thể dựa trên một ý nghĩa khác để thiết lập sự tin tưởng và có được các thuộc tính của client (e.g., URI-chuyển-hướng, loại client). Ví dụ, việc đăng kí có thể được thiết lập bằng cách sử dụng một công cụ riêng hoặc của một bên-thứ-ba cung cấp hoặc bằng cách máy-chủ chứng-thực yêu cầu client vào một kênh bảo-mật.

Khi client đăng kí, những nhà phát triển client NÊN:

* Chỉ ra loại client như mô tả tại phần 2.1
* Chỉ ra URI-chuyển-hướng của client được mô tả tại 3.1.2
* Thêm vào bất kì thông tin nào được yêu cầu bởi máy-chủ chứng-thực (e.g., tên ứng dụng, website, mô tả, ảnh logo, các điều khoản )
  1. Loại Client

OAuth định nghĩa hai loại client, dựa trên khả năng của client có thể bảo đảm tin cậy với máy-chủ chứng-thực (i.e., khả năng duy trì tính bảo mật của thông tin người dùng của chúng):

Confidential (tin cậy) : Client có khả năng duy trì độ tin cậy với những chứng-thư (e.g., client cài đặt trên một máy chủ bảo mật với sự truy cập giới hạn tới các chứng-thư của người dùng), hoặc khả năng bảo mật việc chứng-thực client sử dụng cách khác.

Public(công-cộng) : Client không có khả năng duy trì tính bảo mật cho những chứng-thư (e.g., client vận hành trên thiết bị được sử dụng bởi chủ sở-hữu tài-nguyên, như là một ứng-dụng native hay một ứng-dụng web)

Việc thiết kế loại client dựa trên định nghĩa của máy-chủ chứng-thực về chứng-thực an toàn và những mức độ của chứng-thư của client mà nó chấp nhận. Máy-chủ chứng-thực KHÔNG NÊN tạo ra các giả định về loại client.

Một client có thể được cài đặt như một tập hợp gồm các phần riêng rẽ, mỗi phần có loại client và môi trường bảo mật khác nhau (e.g., một client gồm một thành phần bảo mật dựa vào server và một phần công cộng dựa trên trình duyệt) . Nếu máy-chủ chứng-thực không cung cấp sự hỗi trợ cho những client này hoặc không cung cấp hướng dẫn về việc đặng kí, thì client NÊN đăng kí mỗi phần như một client riêng biệt.

Bản đặc tả này được thiết kế xung quanh những kiểu client sau:

Web Application: Một ứng dụng web là một client tin cậy vận hành trên một máy chủ web. Người sở hữu tài nguyên truy cập client thông qua một giao diện HTML thể hiện ở một tác nhân tương tác trêm một thiết bị được người sở hữu tài nguyên sử dụng. Chứng nhận client cũng như khóa-truy-cập được cấp cho client thì đều được cất giữ trên máy chủ web và không được cấp cho hoặc cho phép người dùng truy cập.

Ứng dụng dựa trên tác nhân tương tác: Một ứng dụng dựa trên tác-nhân tương-tác là một client công-cộng với nghĩa mã-nguồn của client có thể được tải từ một máy-chủ web và chạy với một tác-nhân tương-tác trong nó. (ví dụ, web-browser) trên thiết bị sử dụng bởi người sở hữu tài nguyên. Giao thức truyền dữ liệu và chứng-nhận dễ dàng truy cập (thường là mở) bởi chủ sở hữu tài nguyên. Khi những ứng dụng chứa tác-nhân tương-tác, chúng có thể tạo một trải nghiệm sử dụng liền mạch khả năng của tác-nhân tương-tác khi yêu cầu chứng-thực.

Ứng dụng native: Một ứng dụng native là một client công-cộng được cài đặt và vận hành trên thiết bị được sử dụng bởi người sở hữu tài nguyên. Giao thức dữ liệu và các chứng nhận có thể truy cập bởi người sở hữu tài nguyên. Giả thiết rằng chứng-nhận nào được gói trong ứng dụng cũng có thể bị trích xuất. Mặt khác, bằng việc cấp phát tự động những chứng nhận cũng như như khóa-truy-cập, khóa-đổi-mới làm ta nhận được độ bảo mật có thể chấp nhận được. Tối thiểu, những chứng-nhận này được bảo vệ khỏi những máy chủ thù địch mà ứng dụng có thể sẽ tương tác với. Trên một vài nền-tảng, những chứng-nhận này có thể được bảo vệ khỏi những ứng dụng khác nằm trên cùng thiết bị.

* 1. Định danh client

Máy chủ chứng thực cung cấp cho những client đã đăng kí một định danh – một chuỗi duy nhất đại diện cho thông tin đăng kí cung cấp bởi client. Định danh của client không phải là thứ gì bí mật; Nó được để lộ cho người sở hữu tài nguyên và KHÔNG ĐƯỢC sử dụng một mình để chứng thực client. Định danh của client là duy nhất đối với máy chủ chứng thực.

* 1. Xác thực client

Đối với client tin cậy, client và máy chủ chứng-thực thiết lập một phương thức chứng-thực client phù hợp với yêu cầu về bảo mật của máy chủ chứng thực. Máy chủ chứng thực CÓ THỂ chấp nhận bất kì dạng chứng thực nào đáp ứng đủ yêu cầu.

Những client tin cậy được cấp phát cho một tập chứng nhận dùng trong việc chứng-thực với máy-chủ chứng-thực (ví dụ: passowrd, public/private key pair).

Máy-chủ chứng-thực CÓ THỂ thiết lập một phương thức chứng thực với những client công-cộng. Tuy nhiên, máy-chủ chứng-thực KHÔNG ĐƯỢC dựa vào những chứng thực của client công-cộng với mục đích định danh client.

Client KHÔNG ĐƯỢC sử dụng nhiều hơn một phương thức chứng-thực cho mỗi yêu cầu.

* + 1. Mật khẩu của client.

Những client sở hữu mậu khẩu CÓ THỂ sử dụng đặc tả chứng thực HTTP Basic mô tả tại [[RFC2617]](https://tools.ietf.org/html/rfc2617) để chứng-thực với máy chủ chứng-thực. Định danh của client được biến đổi sự dụng thuật toán biến đổi “application/x-www-form-urlencoded” (Phụ Lục B), và giá trị đã biến đổi được dùng như tên-đăng-nhập; mật khẩu của client được biến đổi dùng cùng thuật toán và được dùng như mật khẩu. Máy chủ chứng-thực PHẢI hỗ trợ đặc tả chứng-chứng thực HTTP Basic để chứng thực các client đã được cấp cho một mật khẩu.

Ví dụ: Authorization: Basic czZCaGRSa3F0Mzo3RmpmcDBaQnIxS3REUmJuZlZkbUl3

Thêm nữa, máy chủ chứng thực CÓ THỂ hỗ trợ việc gom chứng thực client vào trong phần thân của yêu cầu sử dụng những tham số này:

client\_id : BẮT BUỘC . định danh của client được cung cấp cho client trong quá trình đăng kí.

client\_secret : BẮT BUỘC . chuỗi bí mật của client. Client nên bỏ qua tham số này nếu chuỗi bí mật là một chuỗi rỗng.

Việc gom chứng-nhận client vào trong phần thân của yêu cầu sử dụng hai tham số này là KHÔNG KHUYẾN KHÍCH và NÊN giới hạn ở những client không thể trược tiếp sử dụng đặc tả chứng-thực HTTP Basic (hoặc các đặc tả chứng thực HTTP dựa trên mật khẩu khác). Những thuộc tính chỉ nên gửi đi trong phần thân của yêu cầu và KHÔNG ĐƯỢC thêm vào phần URI của yêu cầu.

Ví dụ, một yêu cầu để cấp mới một khóa-truy-cập sử dụng phần thân của yêu cầu.

POST /token HTTP/1.1

Host: server.example.com

Content-Type: application/x-www-form-urlencoded

grant\_type=refresh\_token&refresh\_token=tGzv3JOkF0XG5Qx2TlKWIA

&client\_id=s6BhdRkqt3&client\_secret=7Fjfp0ZBr1KtDRbnfVdmIw

Máy chủ chứng-thực PHẢI yêu cầu sử dụng TLS khi gửi những yêu cầu sử dụng chứng thực dùng mật khẩu.

Khi phương thức chứng-thực client bao gồm một mật khẩu, Máy chủ chứng-thực PHẢI bảo vệ bất kỳ tiếp điểm nào có thể khiến nó tổn thương trước các cuộc tấn công brute-force.

* + 1. Các phương thức xác thực khác

Máy chủ chứng thực CÓ THỂ hỗ trợ bất kỳ đặc tả chứng thực HTTP nào phù hợp với yêu cầu bảo mật của nó. Khi sử dụng những phương thức chứng thực khác, máy chủ chứng-thực PHẢI định nghĩa một mối liên kết giữa định danh của client (bản ghi đăng kí) và đặc tả chứng-thực.

* 1. Client không đăng kí.

Đặc tả này không loại trừ những việc sử dụng cho những client không đăng kí. Tuy nhiên, việc sử dụng cho những client này quá xa với phạm vi của đặc tả này và yêu cầu những phân tích bảo mật và xem xét tác động của những tương tác của nó.

1. Các đầu mút của giao thức

Tiến trình chứng thực sử dụng hai đầu mút máy chủ chứng thực (HTTP resources):

* Đầu Chứng thực – sử dụng bởi client để được chứng nhận từ người sở hữu tài nguyên thông qua việc chuyển hướng qua tác-nhân-tương-tác .
* Đầu mút khóa-truy-cập – sử dụng bởi client để chuyển đổi một chứng-thực thành khóa-truy-cập, thường là với chứng-nhận của client.

Cũng như một đầu mút client:

* Đầu mút chuyển hướng – sử dụng bởi máy chủ chứng thực để phản hồi kèm với chứng-nhận cho client thông qua tác-nhân tương-tác của chủ sở hữu tài nguyên.

Không phải loại chứng thực nào cũng dùng cả hai loại đầu mút. Những chứng nhận mở rộng CÓ THỂ định nghĩa thêm những đầu mút cần thiết.

* 1. Các đầu chứng thực.

Đầu mút chứng thực được dùng để tương tác với chủ sở hữu của tài nguyên và nhận một sự ủy quyền. Máy củ chứng thực trước tiên PHẢI kiểm tra và xác định chủ sở hữu tài nguyên. Cách máy củ chứng thực xác định chủ sở hữu tài nguyên (như là, tên đăng nhập và mật khẩu, phiên làm việc ....) nằm ngoài phạm vi của đặc tả này.

Ý nghĩa của việc client xác định được vị trí của máy chủ chứng thực nằm ngoài phạm vi của tài liệu này nhưng vị trí thường được cung cấp trong tài liệu về dịch vụ.

URI của đầu mút CÓ THỂ bao gồm một thành phần được biến đổi theo “application/x-www-form-urlencoded” (Phụ lục B) ([[RFC3986] Section 3.4](https://tools.ietf.org/html/rfc3986#section-3.4)), thứ PHẢI được duy trì khi thêm những tham số truy vấn khác. URI của đầu mút PHẢI KHÔNG chứa các thành phần bị phân mảnh.

Khi yêu cầu kết quả từ đầu mút chứng thực dưới sự chứng nhận của người dùng và chứng nhận có dạng văn bản thuần (trong phản hồi của truy vấn HTTP), máy chủ chứng thực PHẢI yêu cầu dùng TLS như mô tả tại Section 1.6 khi gửi yêu cầu đến đầu mút chứng thực.

Máy chủ chứng thực PHẢI hỗ trợ việc dùng HTTP “GET” [[RFC2616](https://tools.ietf.org/html/rfc2616)] cho đầu mút chứng thực và cũng CÓ THỂ hỗ trợ thêm “POST”.

Tham số được gửi mà không có giá trị PHẢI làm sao để như thể chúng đã được loại bỏ khỏi yêu cầu. Máy chủ chứng thực PHẢI bỏ qua những tham số không xác định. Các tham số trong yêu cầu và phản hồi KHÔNG ĐƯỢC chứa nhiều hơn một.

* + 1. Loại phản hồi

Đầu mút chứng thực được dùng trong cấp quyền hạn dạng mã-chứng-nhận và bao hàm việc cấp quyền gián tiếp (?). Client liên lạc với máy chủ chứng thực để cân nhắc việc loại chứng nhận sử dụng tham số sau:

response\_type : BẮT BUỘC . Giá trị của tham số phải là một trong hai “code” để yêu cầu một mã-chứng-thực mô tả tại Section 4.1.1 hoặc “token” để yêu cầu một khóa-truy-cập (implicit grant) được mô tả tại Section 4.2.1, hoặc một giá trị mở rộng đã được đăng kí và theo mô tả tại Section 8.4.

Loại phản hồi mở rộng CÓ THỂ chứa một danh sách giá trị phân biệt bởi khoảng trắng (%x20) , thứ tự của các giá trị là không quan trọng.

Nếu một yêu cầu chứng thực thiếu tham số “response\_type”, hoặc nếu giá trị của nó không đúng, máy chủ chứng thực PHẢI trả về một lỗi mô tả tại Section 4.1.2.1.

* + 1. Các đầu chuyển hướng

Sau khi hoàn thành giao tiếp với chủ sở hữu tài nguyên, máy chủ chứng thực chuyển hướng tác-nhân tương-tác của chủ sở hữu trở lại client. Máy chủ chứng thực chuyển hướng tác-nhân tương-tác đến đầu mút chuyển hướng của client đã thiết lập trước đó với máy chủ chứng thực thông qua tiến trình đăng kí của client hoặc khi yêu cầu chứng thực được tạo.

URI của đầu mút chuyển hướng PHẢI là một URI hoàn chỉnh như định nghĩa tại [[RFC3986] Section 4.3](https://tools.ietf.org/html/rfc3986#section-4.3). URI của đầu mút CÓ THỂ bao gồm một thành phần được biến đổi theo “application/x-www-form-urlencoded”, thứ PHẢI được duy trì khi thêm các tham số truy vấn. URI của đầu mút KHÔNG ĐƯỢC chứa các đoạn phân mảnh.

* + - 1. Đầu mút yêu cầu chứng-nhận.

Đầu mút chuyển hướng NÊN đòi hỏi việc sử dụng TLS như mô tả ở Section 1.6 khi response\_type là “code” hoặc “token” , hoặc khi yêu cầu chuyển hướng sẽ dẫn đến việc truyền những chứng-nhận nhạy cảm qua một mảng mở. Tài liệu này không yêu cầu bắt buộc sử dụng TLS bởi vì tại thời điểm viết việc yêu cầu client thiết lập TLS làm môt rào cản lớn đòi hỏi nhiều lập trình viên. Nếu TLS không có sẵn, máy chủ chứng thực NÊN cảnh báo cho chủ sở hữu tài nguyên về đầu mút không bảo mật để chuyển hướng đến. (như là, hiển thị một thông báo trong suốt yêu cầu chứng thực).

Lack of transport-layer security can have a severe impact on the security of the client and the protected resources it is authorized to access. The use of transport-layer security is particularly critical when the authorization process is used as a form of delegated end-user authentication by the client (e.g., third-party sign-in service).

* + - 1. Các yêu cầu khi đăng kí

Máy chủ chứng thực PHẢI đòi hỏi những client sau phải đăng kí những đầu mút chuyển hướng của chúng:

* Những client công cộng.
* Những client bảo mật dùng kiểu chứng-nhận gián-tiếp.

Máy chủ chứng thực NÊN yêu cầu tất cả các client đăng kí đầu mút chuyển hướng trước khi sử dụng đầu mút chứng thực.

Máy chủ chứng thực NÊN yêu cầu client cung cấp URI hoàn chỉnh (client CÓ THỂ dùng tham số “state” để lưu vết những thay đổi qua mỗi yếu cầu chứng thực). Nếu việc đòi hỏi phải đăng kí URI chuyển hướng hoàn chỉnh là không khả thi, Máy chủ chứng thực NÊN yêu cầu việc đăng kí URI scheme, tác giả, và đường dẫn (chỉ cho phép client linh động thành phần URI chuyển hướng trong yêu cầu khi yêu cầu chứng thực).

Máy chủ chứng thực CÓ THỂ cho phép client đăng kí nhiều đầu mút chuyển hướng.

Lack of a redirection URI registration requirement can enable an attacker to use the authorization endpoint as an open redirector as described in Section 10.15.

* + - 1. Cấu hình động

Nếu có nhiều URI chuyển hướng được đăng kí, nếu chỉ một phần của URI chuyển hướng được đăng kí, hoặc nếu không URI chuyển hướng nào được đăng kí, client PHẢI thêm một URI chuyển hướng vào cùng yêu cầu chứng thực sử dụng tham số “redirect\_uri”

Khi một URI chuyển hướng được thêm vào một yêu cầu chứng thực, máy chủ chứng thực PHẢI so sánh và kiểm tra trùng với ít nhất một trong những URI chuyển hướng đã được đăng kí (hoặc với các thành phần của URI) như định nghĩa tại [[RFC3986](https://tools.ietf.org/html/rfc3986#section-6) ] Section 6, nếu có bất kì URI chuyển hướng nào đã đăng kí. Nếu bản đăng kí của client bao gồm URI chuyển hướng hoàn chỉnh, máy chủ chứng thực PHẢI so sánh hai URI sử dụng hàm so sách chuỗi đơn giản như được định nghĩa tại [[RFC3986] Section 6.2.1.](https://tools.ietf.org/html/rfc3986#section-6.2.1)

* + - 1. Các đầu mút không hợp lệ.

Nếu việc kiểm tra một yêu cầu chứng thực gặp thất bại do thiếu, không hợp lệ, URI chuyển hướng không trùng, máy chủ chứng thực NÊN chuyển lỗi đến người sở hữu tài nguyên và KHÔNG ĐƯỢC tự động điều hướng tác-nhân tương-tác đến URI chuyển hướng không hợp lệ.

* + - 1. Đầu mút nội dung

Yêu cầu chuyển hướng đến đầu mút của client thường xác định trong phần phản hồi tài liệu HTML, được thực hiện bởi tác-nhân tương-tác. Nếu phản hồi HTML được phục vụ trực tiếp như là kết quả của yêu cầu chuyển hướng, bất kể kịch bản nào được thêm vào tài liệu HTML sẽ được thực thi với đầy đủ quyền truy cập đến URI chuyển hướng và các chứng nhận mà chúng chứa.

Client KHÔNG NÊN chứa bất kỳ mã kịch bản của bên thứ ba nào (như là, công cụ phân tích của bên thứ ba, phần mở rông tính năng xã hội, mạng lưới quảng cáo) vào trong phản hồi của đầu mút chuyển hướng.

* 1. Đầu mút khóa-truy-cập

Đầu mút khóa-truy-cập được sử dụng bởi client để nhận một khóa-truy-cập bằng cách trình ra chứng-nhận cấp-quyền hoặc khóa-đổi-mới. Đầu mút khóa-truy-cập được dùng với mỗi chứng-nhận cấp-quyền ngoại trừ cho loại cấp-quyền gián-tiếp (vì một khóa-truy-cập được cấp trực tiếp).

Các cách mà client xác định được vị trí của đầu mút khóa-truy-cập nằm ngoai phạm vi của đặc tả này, nhưng vị trí thường được cung cấp qua tài liệu về dịch vụ.

URI của đầu mút CÓ THỂ bao gồm một thành phần biến đổi bởi “application/x-www-form-urlencoded” và phải duy trì khi thêm các tham số truy vấn. URI của đầu mút KHÔNG ĐƯỢC chứa các thành phần bị phân mảnh.

Khi các yêu cầu đến đầu mút khóa-truy-cập có truyền các chứng-nhận dưới dạng văn bản thuần (thông qua yêu cầu và phải hồi của giao thức HTTP), máy chủ chứng-thực PHẢI đòi hỏi việc sử dụng TLS như đả mô tả trong Section 1.6 khi chuyển các yêu cầu đến đầu mút khóa-truy-cập.

Client PHẢI dùng phương thức “POST” của giao thức HTTP khi tạo những yêu cầu tạo khóa-truy-cập.

Các tham số được gửi mà không có giá trị PHẢI được thể hiện sao cho như thể nó bị loại bỏ khỏi yêu cầu. Máy chủ chứng-thực PHẢI bỏ qua nhửng tham số không xác định. Các tham số trong yêu cầu và phản hồi KHÔNG ĐƯỢC chứa nhiều hơn một.

* + 1. Việc chứng nhận client.

Các client bảo mật hoặc các loại client khác được cấp ra chứng-nhận client PHẢI chứng thực với máy chủ chứng thực như đã mô tả trong Section 2.3 khi tạo yêu cầu đến đầu mút khóa-truy-cập. Việc chứng thực client hữu ích cho:

* Đảm bảo việc kết hợp của các khóa-đổi-mới và các mã-chứng-nhận cho client mà chúng được cấp cho. Việc chứng thực client là rất quan trọng khi một mã-chứng-nhận được chuyển đến một đầu mút thông qua một kênh không an toàn hoặc khi một URI chuyển hướng không được đăng kí hoàn chỉnh.
* Khôi phục từ việc một client bị tấn công bằng cách vô hiệu hóa nó hoặc thay đổi chứng nhận của nó, như vậy là ngăn chặn được những kẻ tấn công dùng những khóa-đổi-mới bị đánh cắp. Việc thay đổi một tập các chứng-nhận của client là nhanh hơn hẳn việc cấp lại hoàn toàn các khóa-đổi-mới.
* Những bài tập tốt nhất về cài đặt quản lý chứng thực, thứ đòi hỏi phải hình dung trước sự xoay vòng của chứng-nhận. Việc xoay vòng của toàn bộ tập khóa-đổi-mới có thể là thách thức, trong khi xoay vòng của một tập đơn lẻ các chứng-nhận client rõ ràng là dễ hơn.

Một client CÓ THỂ dùng tham số “client\_id” để định danh chính mình khi gửi các yêu cầu đến đầu mút khóa-truy-cập. Khi gửi các yêu cầu “authorization\_code” “grant\_type” đến đầu mút khóa-truy-cập, một client chưa chứng thực PHẢI gửi “client\_id” để ngăn chính client vô tình chấp nhận một mã được thiết kế cho một client với một “client\_id” khác. Nó bảo vệ client khỏi sự thay đổi của mã-chứng-thực. (nó không cung cấp thêm sự bảo mật nào trong việc bảo vệ tài nguyên).

* 1. Phạm vi Khóa-Truy-Cập

Các đầu mút chứng thực và các đầu mút khóa-truy-cập cho phép client khai báo phạm vi truy cập sử dụng tham số “scope” trong yêu cầu. Đến lượt mình, máy chủ chứng-thực sử dụng tham số “scope” trong phản hồi để thông báo cho client phạm vi mà khóa-truy-cập được cấp.

Giá trị của tham số “scope” là một danh sách được ngăn cách bởi khoảng trắng, có phân biệt hoa thường. Chuỗi này được định nghĩa bởi máy chủ chứng thực. Nếu giá trị chứa nhiều chuỗi cách nhau bởi khoảng trắng, thứ tự của chúng là không quan trọng, và mỗi chuỗi thêm một phạm vi truy cập vào trong phạm vi được yêu cầu.

scope = scope-token \* (SP scope-token)

scope-token = 1 \* (%x21 / %x23-5B / %x5D-7E)

Máy chủ chứng thực CÓ THỂ bỏ qua toàn bộ hoặc một phần những phạm vi được client yêu cầu, dựa vào chính sách của máy chủ chứng thực hoặc chỉ dẫn của chủ sở hữu tài nguyên. Nếu phạm vi của khóa-truy-cập đã được cấp khác với cái được yêu cầu bởi client, máy chủ chứng thực PHẢI thêm tham số “scope” để thông báo cho client những phạm vi thực sự được cấp phép.

Nếu client bỏ qua tham số “scope” khi yêu cầu chứng thực, máy chủ chứng-thực PHẢI hoặc là xử lý yêu cầu sử dụng một giá trị mặc định hoặc ngừng xử lý và chỉ ra phạm vi không hợp lệ. Máy chủ chứng thực NÊN có tài liệu về những yêu cầu của các phạm vi và giá trị mặc định. (nếu có)

1. Việc lấy chứng thực

Để yêu cầu một khóa-truy-cập, client lấy sự chứng-thực từ chủ sở hữu tài nguyên. Sự chứng-thực là sự chuyển tải theo dạng của một sự chứng-nhận cấp-quyền, thứ client sử dụng để yêu cầu khóa-truy-cập. OAuth định nghĩa 4 loại cấp-quyền: Mã-chứng-nhận, gián tiếp, mật khẩu của chủ sở hữu tài nguyên, và chứng-nhận client. Nó cũng cung cấp các hệ thống mở rộng để định nghĩa thêm các loại cấp quyền khác.

* 1. Authorization Code grant

Loại cấp quyên theo mã-chứng-thực được sử dụng để lấy cả hai khóa-truy-cập và khóa-đổi-mới và nó đơn giản hóa của các client tin-cậy. Vì đây là tiến trình dựa trên sự chuyển hướng, nên client phải có khả năng giao tiếp với tác-nhân tương-tác của chủ sở hữu (thường là một web-browser) và khả năng nhận các yêu cầu (từ sự chuyển hướng) từ máy chủ chứng-thực.

+----------+

| Chủ sở |

| hữu |

| |

+----------+

^

|

(B)

+----|-----+ định danh client +---------------+

| -+----(A)-- & URI chuyển hướng --->| |

| tác - | | máy chủ chứng |

| nhân tt -+----(B)—chứng nhận người dùng--->| thực |

| | | |

| -+----(C)— mã-chứng-thực ---<| |

+-|----|---+ +---------------+

| | ^ v

(A) (C) | |

| | | |

^ v | |

+---------+ | |

| |>---(D)------- mã-chứng-thực ---------' |

| Client | & URI chuyển hướng |

| | |

| |<---(E)----- khóa truy cập-------------------'

+---------+ (w/ tùy chọn khóa-đổi-mới)

1. Client bắt đầu tiến trình bằng cách điều hướng tác-nhân tương-tác của chủ sở hữu đến đầu mút chứng thực. Client kèm theo định danh, phạm vi yêu cầu, trạng thái, và một URI chuyển hướng để khi máy chủ chứng thực sẽ điều hướng tác-nhân tương-tác tới đó khi quyền truy cập được cấp (hoặc không).
2. Máy chủ chứng thực xác nhận chủ sở-hữu (thông qua tác-nhân tương-tác) và xác định xem chủ sở hữu chấp nhập cấp quyền cho yêu của client hay không.
3. Dựa vào quyết định của chủ sở-hữu tài nguyên, máy chủ chứng-thực chuyển hướng tác-nhân tương-tác trở lại client theo URI chuyển hướng được cấp trước đó (trong yêu cầu hoặc ở giai đoạn đăng kí của client). URI chuyển hướng bao gồm một mã chứng-thực và bất kì trạng thái nào được gửi kèm bởi client trước đó.
4. Client yêu cầu một khóa-truy-cập từ đầu mút khóa-truy-cập của máy chủ chứng thực bằng cách kèm theo mã-chứng-nhận có được trước đó. Khi tạo một yêu cầu, client xác nhận với máy chủ chứng thực. Client thêm URI chuyển hướng đã dùng để lấy mã-chứng-nhận cho việc kiểm tra.
5. Máy chủ chứng-thực xác nhận client, kiểm tra mã-chứng-thực và đảm bảo rằng URI chuyển hướng nhận được trùng với URI được dùng để chuyển hướng client ở bước (C). Nếu tất cả hợp lệ, máy chủ chứng-thực phản hồi lại với một khóa-truy-cập và, tùy theo, một khóa-đổi-mới.
   * 1. Yêu cầu chứng thực

Client cấu trúc URI của yêu cầu bằng cách thêm các tham số sau vào thành phần truy vấn của URI của đầu mút chứng-thực sử dụng biến đổi “application/x-www-form-urlencoded”:

response\_type : BẮT BUỘC . Giá trị PHẢI là “code”.

client\_id : BẮT BUỘC . định danh client như mô tả trong Session 2.2

redirect\_url : TÙY CHỌN . như mô tả tại Session 3.1.2.

scope : TÙY CHỌN . phạm vi của yêu cầu truy cập như mô tả tại Session 3.3

state : KHUYẾN DỤNG . Một giá trị sử dụng bởi client để duy trì trạng thái giữa giai đoạn yêu cầu và gửi lại. Máy chủ chứng thực đính kèm giá trị này khi chuyển hướng tác-nhân tương-tác trở lại với client. Tham số NÊN được dùng để ngăn chặn cross-site request forgery như mô tả tại Section 10.12.

Client điều hướng chủ sở hữu tài nguyên đến URI đã được cấu trúc sử dụng một phản hồi chuyển hướng của giao thức HTTP, hoặc bằng cách các cách khác khả dụng thông qua tác-nhân tương-tác.

Ví dụ, client chuyển hướng tác-nhân tương-tác để tạo câu truy vấn HTTP sử dụng TLS:

GET /authorize?response\_type=code&client\_id=s6BhdRkqt3&state=xyz

&redirect\_uri=https%3A%2F%2Fclient%2Eexample%2Ecom%2Fcb HTTP/1.1

Host: server.example.com

Máy chủ chứng thực kiểm tra yêu cầu để đảm bảo có đầy đủ các tham số cần thiết và có giá trị thích hợp. Nếu yêu cầu là hợp lệ, máy chủ chứng-thực xác nhận với chủ sở hữu tài nguyên và lấy một quyết định mang tính xác nhận (bằng cách hỏi chủ sở hữu tài nguyên hoặc bằng các thiết lập khác tương đương).

Khi một quyết định được đưa ra, máy chủ chứng-thực điều hướng tác-nhân tương-tác tới URI điều hướng mà client cung cấp sử dụng một phản hồi chuyển hướng của giao thức HTTP, hoặc các cách khác khả dụng thông qua tác-nhân tương-tác.

* + 1. Phản hồi của việc chứng thực

Nếu chủ sở-hữu cấp quyền cho yêu cầu truy cập, máy chủ chứng-thực cấp một mã-chứng-thực và phân phối nó đến client bằng cách thêm những tham số sau vào thành phần truy vấn của URI chuyển hướng sử dụng biến đổi “application/x-www-form-urlencoded”.

code : BẮT BUỘC. Mã-chứng-thực được tạo bởi máy chủ chứng-thực. Mã-chứng-thực PHẢI hết hạn nhanh chóng sau khi được cung cấp để tránh nguy cơ bị rò rỉ. Thời gian tối đa một mã-chứng-thực KHUYẾN CÁO là 10 phút. Client KHÔNG ĐƯỢC sử dụng mã-chứng-thực nhiều hơn một lần. Nếu một mã-chứng-thực được dùng nhiều hơn một lần, máy chủ chứng thực PHẢI ngăn chặn yêu cầu và NÊN yêu cầu cấp mới (khi có thể) tất cả các khóa đã được cấp trước đó từ mã-chứng-thực này. Mã-chứng-thực gắn liền với định danh client và URI chuyển hướng.

state : BẮT BUỘC nếu tham số “state” tồn tại trong yêu cầu chứng-thực của client. Giá trị chính xác là giá trị nhận được từ client.

Ví dụ, máy chủ chứng-thực chuyển hướng tác-nhân tương-tác bằng cách gửi phản hồi sau theo giao thức HTTP:

HTTP/1.1 302 Found

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

&state=xyz

Client PHẢI bỏ qua những tham số được trả về mà không xác định. Độ dài của chuỗi mã-chứng-thực không được xác định trong tài liệu này. Client nên tránh võ đoán về độ dài của mã, máy chủ chứng-thực NÊN có tài liệu về độ lớn của bất kì giá trị nào mà nó cung cấp.

* + - 1. Error Response

Nếu yêu cầu thất bại do thiếu sót, không hợp lệ hoặc không khớp URI chuyển hướng, hoặc nếu định danh của client thiếu hoặc không hợp lệ, máy chủ chứng thực NÊN thông báo cho chủ sở hữu tài nguyên về lỗi và KHÔNG ĐƯỢC tự động chuyển hướng tác-nhân tương-tác đến URI chuyển hướng không hợp lệ.

Nếu chủ sở hữu tài nguyên không đồng ý cấp quyền cho yêu cầu hoặc nếu yêu câu thất bại vì những nguyên nhân khác hơn là thiếu hoặc URI chuyển hướng không hợp lệ, máy chủ chứng thực thông báo cho client bằng cách thêm các tham số sau vào thành phần truy vấn của URI chuyển hướng sử dụng biến đổi “application/x-www-form-urlencoded”.

error : BẮT BUỘC . Một mã lỗi ASCII [[USASCII](https://tools.ietf.org/html/rfc6749#ref-USASCII)] như sau:

* invalid\_request : yêu cầu bị thiếu các tham số cần thiết, kèm theo một tham số không hợp lệ, kèm theo nhiều hơn một tham số (trùng nhau), hoặc các lỗi định dạng khác.
* unauthorized\_client : không thể xác định client để yêu cầu mã-chứng-thực.
* access\_denied : Chủ sở hữu tài nguyên hoặc máy chủ chứng-thực không chấp nhận yêu cầu.
* unsupported\_response\_type : Máy chủ chứng-thực không hỗ trợ việc lấy mã-chứng-thực dùng phương pháp này.
* invalid\_scope : phạm vi yêu cầu không hợp lệ, không xác định hoặc lỗi định dạng.
* server\_error : Máy chủ chứng thực gặp một trường hợp không mong muốn ngăn cản nó đáp ứng yêu cầu. (mã lỗi này là cần thết bởi một mã trạng thái 500 của giao thức HTTP không thể gửi cho client thông qua phản hồi chuyển hướng của giao thức HTTP).
* temporarily\_unavailable: Máy chủ chứng thực hiện không thể xử lý yêu cầu vì quá tải hoặc bảo trì máy chủ. (mã lỗi này là cần thiết bởi mã trạng thái 503 của giao thức HTTP không thể gửi tới client thông qua phản hồi chuyển hướng của giao thức HTTP)

Giá trị của tham số “error” KHÔNG ĐƯỢC thêm những kí tự nằm ngoài tập %x20-21 / %x23-5B / %x5D-7E.

error\_description : TÙY CHỌN . chuỗi ASCII con người đọc được cung cấp thêm thông tin, sử dụng hỗ trợ lập trình viên phía client hiểu hơn về lỗi đã phát sinh.

error\_uri : TÙY CHỌN . Một URI xác định một trang web con người đọc được có các thông tin về lỗi, sử dụng để cung cấp cho lập trình viên thông tin thêm về lỗi. Giá trị của tham số “error\_uri” PHẢI tương thích với cú pháp của URI và KHÔNG ĐƯỢC chứa các kí tự ngoài tập %x21 / %x23-5B / %x5D-7E.

state : BẮT BUỘC . Nếu tham số “state” được khai báo trong yêu cầu chứng thực của client. Nó có giá trị chính xác là những gì nhận được từ client.

Ví dụ, máy chủ chứng thực điều hướng tác-nhân tương-tác bằng cách gửi phản hồi HTTP sau:

HTTP/1.1 302 Found

Location: https://client.example.com/cb?error=access\_denied&state=xyz

* + 1. Access Token Request

Client tạo một yêu cầu đến đầu mút khóa-truy-cập bằng cách gửi những tham số sau định dạng “application/x-www-form-urlencoded” với một kí tự mã hóa UTF-8 nằm trong phần thân một yêu cầu của giao thức HTTP.

grant\_type : BẮT BUỘC. Giá trị PHẢI là “authorization\_code”.

code : BẮT BUỘC . mã-chứng-thực nhận được từ máy chủ chứng thực.

redirect\_uri : BẮT BUỘC . Nếu tham số “redirect\_uri” được thêm vào yêu cầu chứng thực như mô tả tại Section 4.1.1 và giá trị của chúng PHẢI được xác định.

client\_id : BẮT BUỘC . Nếu client chưa được chứng thực với máy chủ chứng thực như mô tả tại Session 3.2.1.

Nếu client thuộc loại đảm bảo hoặc client được cấp chứng-nhận (hoặc thuộc loại có yêu cầu chứng thực khác) , client PHẢI chứng-nhận với máy chủ chứng thực như mô tả tại Section 3.2.1

Ví dụ, client tạo một yêu cầu HTTP sử dụng TLS

POST /token HTTP/1.1

Host: server.example.com

Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW

Content-Type: application/x-www-form-urlencoded

grant\_type=authorization\_code&code=SplxlOBeZQQYbYS6WxSbIA

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

Máy chủ chứng thực phải :

* Yêu cầu việc xác nhận client cho những client tin cậy, hoặc cho bất kể client nào được cấp chứng-nhận (hoặc các loại yêu cầu về chứng-thực khác ).
* Xác nhận client nếu có yêu cầu chứng thực.
* Đảm bảo rằng mã-chứng-nhận được cấp để xác nhận client tin cậy, hoặc nếu là client công co65nt, đảm bảo rằng mãi được cấp cho “client\_id” trong yêu cầu.
* Xác nhận rằng mã-chứng-thực là hợp lệ, và
* Đảm bảo rằng tham số “redirect\_uri” có mặt nếu tham số “redirect\_uri” được thêm vào trong khi khởi tạo yêu cầu chứng thực như mô tả tại Section 4.1.1, và nếu đảm bảo rằng giá trị của chúng là xác định được.
  + 1. Access Token Response

Nếu yêu cầu cấp khóa-truy-cập là hợp lệ và được chứng nhận, máy chủ chứng-thực cấp một khóa-truy-cập và (tùy chọn) khóa-đổi-mới như mô tả tại Section 5.1 . Nếu yêu cầu chứng thực của client thất bại hoặc không hợp lệ, máy chủ chứng thực trả về một phản hồi về lỗi như mô tả tại Section 5.2.

Một ví dụ về một phản hồi khi yêu cầu thành công:

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. Cấp-quyền gián tiếp

Loại cấp quyền gian tiếp được sử dụng để lây khóa-truy-cập (không hỗ trợ việc cấp phát khóa-đổi mới) và tối ưu cho client công cộng vận hành URI chuyển hướng cụ thể. Những client này thường cài đặt trên trình duyệt sử dụng ngôn ngữ kịch bản như javascript.

Vì đây là tiến trình dựa trên nền tảng chuyển hướng, client phải có khả năng giao tiếp với tác-nhân-tương-tác của chủ sở hững tài nguyên (thường là một web browser) và có khả năng nhận yêu cầu đến (thông qua chuyển hướng) từ máy chủ chứng-thực.

Không giống loại cấp quyền theo mã-chứng-thực, ở đây client tạo những yêu cầu riêng biệt cho việc chứng nhận và cho việc lấy khóa-truy-cập, client nhận được khóa truy cập là kết quả của yêu cầu chứng-thực.

Loại cấp quyền gián tiếp này không bao gồm chứng thực client, và dựa vào sự hiện diện của chủ sở hữu tài nguyên và việc đăng kí URI chuyển hướng. Bởi khóa-truy-cập được mã hóa theo URI chuyển hướng, có thể để lộ cho chủ sở hữu tài nguyên và những ứng dụng khác nằm trên cùng một thiết bị.

+----------+

|Chủ sở hữu|

| tài |

| nguyên |

+----------+

^

|

(B)

+----|-----+ Định danh client +---------------+

| -+----(A)-- &URI chuyển hướng --->| |

| tác- | | Máy chủ |

| nhân -|----(B) —Xác định User -->| Chứng thực |

| tương | | |

| tác |<---(C)---URI chuyển hướng ----<| |

| | với khóa-truy-cập +---------------+

| | in Fragment

| | +---------------+

| |----(D)---URI chuyển hướng ---->| Web-Hosted |

| | without Fragment | Client |

| | | Resource |

| (F) |<---(E)------- Script ---------<| |

| | +---------------+

+-|--------+

| |

(A) (G) Khóa-truy-cập

| |

^ v

+---------+

| |

| Client |

| |

+---------+

1. Client mở đầu tiến trình bằng cách chuyển hướng tác-nhân tương-tác của chủ sở hữu tài nguyên đến đầu mút chứng thực. Client kèm theo nhận dạng của nó, phạm vi truy cập nó yêu cầu, trạng thái cục-bộ và một URI chuyển-hướng địa chỉ mà máy chủ chứng thực sẽ gửi lại cho tác-nhân tương-tác khi được cấp quyền.
2. Máy chủ chứng-thực chứng-nhận chủ sở hữu tài nguyên (thông qua tác-nhân tương-tác) và xác định chủ tài nguyên đồng ý cấp quyền hay không.
3. Dựa vào quyết định của chủ sở hữu tài nguyên, máy chủ chứng thực chuyển hướng tác-nhân tương-tác về Client sử dụng URI chuyển-hướng được cung cấp trước đó. URI chuyển hướng bao gồm khóa-truy-cập trong thành phần của URI.
4. Tác-nhân tương-tác làm theo hướng dẫn chuyển hướng bằng cách tạo một yêu cầu đến máy chủ web client client (URI chuyển hướng không bao gồm các thành phần được kèm thêm bởi máy chủ chứng thực) . Tác-nhân tương-tác giữ các thông tin được kèm thêm bởi máy chủ chứng-thực ở cục bộ.
5. Máy chủ web của client trả về một trang web (thường là một tài liệu HTML với một kịch bản được nhúng bên trong) có khả năng truy cập đầy đủ URI chuyển hướng bao gồm các thành phần mà máy chủ chứng-thực gửi kèm, và trích xuất khóa-truy-cập (và các tham số khác) được chứa trong thành phần mà máy chủ chứng-thực gửi kèm.
6. Tác nhân tương-tác thực thi kịch bản được cung cấp bởi máy chủ web của client, thứ trích xuất ra khóa-truy-cập.
7. Tác nhân tương-tác chuyển khóa-truy-cập này cho client.

Xem phần 1.3.1 và 9 để hiểu nền tảng của việc sử dụng cấp quyền gián tiếp.

Xem phần 10.3 và 10.16 để biết các vấn đề bảo mật quan trọng khi sử dụng cấp quyền gián tiếp.

* + 1. Authorization Request

Client tạo ra URI yêu cầu bằng cách thêm các tham số sau vào thành phần truy vấn của URI của đầu mút chứng-thực sử dụng chuyển đổi “application/x-www-form-urlencoded”.

response\_type : BẮT BUỘC , giá trị PHẢI là “token”.

client\_id : BẮT BUỘC, định danh của client mô tả tại phần 2.2

redirect\_uri : TÙY CHỌN , như định nghĩa tại phần 3.1.2

scope : TÙY CHỌN , phạm vị truy cập mô tả tại phần 3.3

state : KHUYÊN DÙNG , Một giá trị sử dụng bởi client để duy trì trạng thái giữa các quá trình truy vấn. Máy chủ chứng-thực kèm theo giá trị này khi chuyển hướng tác-nhân tương-tác về client. Tham số này NÊN sử dụng để tránh cross-site request forgery như mô tả tại phần 10.12

Ví dụ, client chuyển hướng tác-nhân tương-tác tạo ra try vấn HTTP sau sử dụng TLS:

GET /authorize?response\_type=token&client\_id=s6BhdRkqt3&state=xyz

&redirect\_uri=https%3A%2F%2Fclient%2Eexample%2Ecom%2Fcb HTTP/1.1

Host: server.example.com

Máy chủ chứng-thực kiểm tra truy vấn để đảm bảo rằng tất cả các thành phần truy vấn hợp lệ. Máy chủ chứng-thực PHẢI xác nhận URI chuyển-hướng phải trùng khớp với URI chuyển-hướng được đăng kí bởi client như mô tả tại phần 3.1.2

Nếu yêu cầu hợp lệ, máy chủ chứng-thực xác nhận chủ sở-hữu và nhận về một quyết định (bằng cách hỏi chủ sở-hữu hoặc bằng các thiết lập khác tương đương)

Khi quyết định được đưa ra, máy chủ chứng-thực hướng tác-nhân tương-tác đến URI chuyển-hướng được cung cấp trước đó sử dụng một phản hồi HTTP, hoặc một cách nào khác thông cua tác-nhân tương-tác.

* + 1. Access Token Response

Nếu chủ sở-hữu tài nguyên cấp quyền cho truy cập, máy chủ chứng-thực xuất ra một khóa-truy-cập và chuyển nó đến client bằng cách thêm các tham số sau vào thành phần của URI chuyển hướng sử dụng biến đổi “application/x-www-form-urlencoded”:

access\_token : BẮT BUỘC , khóa-truy-cập cấp phát bởi máy chủ chứng thực.

token\_type : BẮT BUỘC , loại của khóa vừa được cấp phát được mô tả tại phần 7.1, giá trị không phân biệt hoa thường.

expries\_in : KHUYÊN DÙNG, thời gian tồn tại của khóa-truy-cập biểu diễn bởi giây. Ví dụ, giá trị “3600” biểu thị rằng khóa-truy-cập sẽ hết hạn trong vòng 1 giờ kể từ khi phản hồi được tạo. Nếu cung cấp tham số này, máy chủ chứng-thực PHẢI cung cấp thời gian hết hạn thông qua cách này hay khác hoặc thông qua tài liệu về các giá trị mặc định.

scope : là TÙY CHỌN, trong trường hợp xác định phạm vi truy cập bởi client; là BẮT BUỘC trong các trường hợp khác. Phạm vi của khóa-truy-cập mô tả tại phần 3.3

state: là BẮT BUỘC nếu thông số “state” được client sử dụng, có giá trị chính xác là giá trị nhận được từ client.

Máy chủ chứng-thực KHÔNG ĐƯỢC phát hành khóa-đổi-mới.

Ví dụ, máy chủ chứng-thực chuyển hướng tác-nhân tương-tác bằng cách gửi một phản hồi HTTP:

Nhà phát triển nên ghi nhớ rằng tác-nhân tương-tác không hỗ trợ việc thêm các thành phần trong trường header “Location” của phản hồi. Nên các client sẽ phải sử dụng các phương pháp khác để chuyển hướng client hơn là dùng phản hồi 3xx của HTTP, Ví dụ trả về một trang HTML bao gồm một nút “continue” để chuyển hướng đến URI khi được nhấn.

Client PHẢI bỏ qua các tham số không xác định trong phản hồi. Kích thước của khóa-truy-cập không được định nghĩa trong tài liệu này. Client nên tránh việc đoán kích thước của giá trị trả về. Máy chủ chứng-thực NÊN cung cấp tài liệu về kích thước của bất kỳ giá trị nào mà nó phát sinh.

* + - 1. Error Response

Nếu việc yêu cầu thất bại do một thiếu sót, giá trị không hợp lệ, hoặc URI chuyển hướng không khớp, hoặc nếu định danh của client thiếu hoặc không hợp lệ, máy chủ chứng-thực NÊN thông báo cho chủ sở-hữu tài-nguyên về lỗi và KHÔNG ĐƯỢC tự động chuyển hướng tác-nhân tương-tác đến URI không hợp lệ.

Nếu chủ sở-hữu tài-nguyên không chấp nhận một yêu cầu truy cập hoặc việc yêu cầu thất bại vì các nguyên do khác hơn là thiết sót hoặc URI chuyển-hướng không hợp lệ, máy chủ chứng-thực báo cho client bằng cách thêm vào những tham số sau vào thành phần của URI chuyển-hướng sử dụng biến đổi “application/x-www-form-urlencoded”:

error : BẮT BUỘC, Một mã duy nhất dạng ASCII như sau:

* invalid\_request : Thiếu một tham số bắt buộc nào đó trong yêu cầu, hoặc một tham số nào đó có giá trị không hợp lệ, hoặc một tham số nào đó xuất hiện nhiều lần, hoặc các lỗi định dạng khác.
* unauthorized\_client : Không xác thực được client khi sử dụng phương thức xác thực này.
* access\_denied : Chủ sở hữu tài-nguyên hoặc máy chủ chứng thực không chấp nhận yêu cầu.
* unsupported\_response\_type : Máy chủ chứng-thực không hỗ trợ cấp phát khóa-truy-cập sử dụng phương thức này.
* invalid\_scope : phạm vi truy cập trong yêu cầu không hợp lệ, hoặc không hiểu, hoặc lỗi định dạng.
* server\_error : máy chủ chứng-thực rơi vào trường hợp không mong muốn cản trở nó thực hiện yêu cầu (mã lỗi này là cần thiết vì trạng thái 500 của giao thức HTTP không thể chuyển tới client thông qua một phản hồi chuyển hướng HTTP) .
* temporarily\_unavailable : máy chủ chứng thực hiện tại không thể xử lý các yêu cầu do quá tải hoặc trong giai đoạn bảo trì. (mã lỗi này là cần thiết vì trạng thái 503 của giao thức HTTP không thể chuyển tới client thông qua một phản hồi chuyển hướng HTTP).

Các giá trị cho tham số “error” KHÔNG ĐƯỢC bao gồm các kí tự nằm ngoài tập %x20-21 / %x23-5B / %x5D-7E.

error\_description : TÙY CHỌN . Một đoạn theo chuẩn ASCII cung cấp các thông tin thêm, sử dụng để hỗ trợ các nhà phát triển phía client hiểu được lỗi đã phát sinh. Giá trị cho tham số “error\_description” KHÔNG ĐƯỢC chứa các kí tự ngằm ngoài tập %x20-21 / %x23-5B / %x5D-7E.

error\_uri : TÙY CHỌN . Một URI chỉ định một trang web chứa thông tin về lỗi, sử dụng để cung cấp cho các nhà phát triển phía client những thông tin thêm về lỗi vừa phát sinh. Giá trị của tham số “error\_uri” PHẢI phù hợp với cú pháp của URI và KHÔNG ĐƯỢC chứa các kí tự nằm ngoài tập %x20-21 / %x23-5B / %x5D-7E.

state : là BẮT BUỘC nếu tham số “state” được sử dụng trong yêu cầu, giá trị chính xác là giá trị nhận được của phía client.

Ví dụ, máy chủ chứng-thực chuyển hướng tác-nhân tương-tác bằng các gửi phản hồi HTTP sau :

HTTP/1.1 302 Found

Location: https://client.example.com/cb#error=access\_denied&state=xyz

* 1. Resource Owner Password Credentials Grant

Kiểu cấp quyền thông qua mật khẩu của chủ sở hữu tài nguyên phù hợp với trường hợp khi mà chủ sở hữu tài nguyên tin tưởng mối quan hệ với client, như là hệ điều hành chạy trên thiết bị hoặc các ứng dụng đòi hỏi mức độ truy cập ở mức cao. Máy chủ chứng thực nên quan tâm đặc biệt khi cung cấp loại chứng thực này và chỉ chấp nhận nó khi các loại khác không khả thi.

Loại cấp quyền này phù hợp cho các client có khả năng lấy chứng-thư từ chủ sở hữu tài nguyên (thường là tên-đăng-nhập và mật khẩu, và thường là dưới hình thức gián tiếp). Nó cũng được dùng để nhập những clients đã dùng hệ thống bằng các hình thức xác thực trực tiếp như thông qua HTTP Basic hoặc Digest authentication sang OAuth để chuyển đổi sang từ chứng-thư sang khóa-truy-cập.

+----------+

|chủ sở hữu|

|tài nguyên|

| |

+----------+

v

| chứng-thư (mật khẩu)

(A) của chủ sở hữu tài nguyên

|

v

+---------+ +---------------+

| |>--(B)---- Resource Owner ------->| |

| | Password Credentials | máy chủ |

| Client | | Chứng thực |

| |<--(C)---- Access Token ---------<| |

| | (w/ Optional Refresh Token) | |

+---------+ +---------------+

1. Chủ sở hữu tài nguyên cung cấp cho client tên đăng nhập và mật khẩu.
2. Client yêu cầu một khóa-truy-cập từ đầu mút khóa-truy-cập của máy chủ chứng-thực bằng cách đính kèm chứng-thư nhận được từ chủ sở-hữu tài nguyên. Khi tạo yêu cầu, client xác thực với máy chủ chứng-thực.
3. Máy chủ chứng-thực chứng nhận client và kiểm tra chứng-thư của chủ sở hữu tài nguyên, và nếu hợp lệ, cấp một khóa-truy-cập.
   * 1. Authorization Request and Response

Phương pháp mà qua đó client lấy được chứng-thư của chủ sở-hữu tài nguyên nằm ngoài phạm vi của tài liệu này. Client PHẢI tránh sử dụng chứng-thư một khi đã nhận được khóa-truy-cập.

* + 1. Access Token Request

Client tạo một yêu cầu đến đầu mút khóa-truy-cập bằng cách thêm những tham số sau trong toàn bộ phần thân của câu truy vấn HTTP sử dụng biến đổi “application/x-www-form-urlencoded” và UTF-8 encoding.

grand\_type : BẮT BUỘC . Giá trị PHẢI là “password”

username : BẮT BUỘC. Tên-truy-cập của chủ sở hữu tài nguyên.

password : BẮT BUỘC. Mật-khẩu của chủ sở-hữu tài nguyên.

scope : TÙY CHỌN . phạm vi truy cập định nghĩa tại phần 3.3

Nếu loại client tin cậy hoặc client được cấp chứng-nhận (hoặc các loại tương tự), client PHẢI chứng thực với máy chủ chứng-thực như mô tả tại phần 3.2.1

Ví dụ, client tạo một truy vấn HTTP sử dụng lớp bảo mật.

POST /token HTTP/1.1

Host: server.example.com

Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW

Content-Type: application/x-www-form-urlencoded

grant\_type=password&username=johndoe&password=A3ddj3w

Máy chủ chứng-thực PHẢI :

* Yêu cầu client chứng-thực cho sự tin cậy hoặc bất kì chứng nhận nào được cấp phát (hoặc với các yêu cầu chứng-thực khác),
* Xác thực client nếu việc xác nhận client là cần thiết, và
* Kiểm tra chứng-nhận của chủ sở hữu tài nguyên sử dụng thuật toán kiểm tra có sẵn.

Một khi khóa-truy-cập đòi hỏi sử dụng mật khẩu của chủ sở hữu tài nguyên, máy chủ chứng-thực PHẢI bảo vệ các đầu mút chống lại các loại tấn công brute force. (như là sử dụng giới hạn hoặc tạo ra các cảnh báo).

* + 1. Access Token Response

Nếu yêu cầu khóa-truy-cập là hợp lệ và được chứng nhận, máy chủ chứng-thực xuất ra một khóa-truy-cập và khóa-đổi-mới nếu có dùng như mô tả tại phần 5.1. Nếu việc yêu cầu thất bại ở việc chứng thực client hoặc không hợp lệ, máy chủ chứng-thực trả về một phản hồi về lỗi như mô tả tại phần 5.2.

Một ví dụ về việc phản hồi sau khi yêu cầu thành công:

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. Client Credentials Grant

Client có thể yêu cầu một khóa-truy-cập sử dụng duy nhất chứng-nhận (hoặc những chứng-thư khác mang ý nghĩa chứng nhận) khi client đang yêu cầu truy cập vào nguồn tài nguyên được bảo vệ nằm trong tầm kiểm soát của nó, hoặc những tài nguyên của chủ sở hữu đã cấp phép trước đó cho máy chủ xác-thực (phương pháp để thực hiện nằm ngoài phạm vi của tài liệu này).

Loại cấp quyền theo chứng-thư PHẢI dùng chỉ với những client tin cậy.

+---------+ +---------------+

| | | |

| |>--(A)- Client Authentication --->| Authorization |

| Client | | Server |

| |<--(B)---- Access Token ---------<| |

| | | |

+---------+ +---------------+

1. Client xác nhận mình với máy chủ chứng thực và yêu cầu một khóa-truy-cập từ đầu-mút khóa-truy-cập.
2. Máy chủ chứng thực xác nhận client và nếu hợp lệ, xuất ra một khóa-truy-cập.
   * 1. Authorization Request and Response:

Vì việc chứng thực client được dùng thông qua phương pháp xác-nhận, nên không cần phải có yêu cầu chứng nhận nào.

* + 1. Access Token Request

Client tạo một yêu cầu đến đầu-mút khóa-truy-cập bằng cách thêm các tham số sau sử dụng biến đổi “application/x-www-form-urlencoded” với bộ mã hóa UTF-8 ở toàn bộ phần thân của truy vấn HTTP:

grand\_type : BẮT BUỘC , giá trị PHẢI là “client\_credentials”

scope : TÙY CHỌN. Phạm vi yêu cầu truy cập như mô tả tại phần 3.3

Client PHẢI xác-nhận với máy chủ chứng thực như mô tả tại phần 3.2.1.

Ví dụ, Client tạo truy vấn hTTP sau sử dụng lớp bảo mật.

POST /token HTTP/1.1

Host: server.example.com

Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW

Content-Type: application/x-www-form-urlencoded

grant\_type=client\_credentials

Máy chủ chứng-thực phải xác nhận client.

* + 1. Access Token Response

Nếu yêu cầu khóa-truy-cập là hợp lệ và được xác nhận, máy chủ chứng thực cấp phát một khóa-truy-cập như mô tả tại phần 5.1 . Một khóa-đổi-mới KHÔNG ĐƯƠC tạo kèm theo. Nếu yêu cầu thất bại do không xác nhận được client hoặc không hợp lệ, máy chủ chứng thực trả về một phản hồi về lỗi như mô tả tại phần 5.2

Một ví dụ về phản hồi khi yêu cầu thành công.

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,

"example\_parameter":"example\_value"

}

* 1. Extension Grants

Client sử dụng một loại cấp quyền mở rộng bởi loại cấp quyết đặc biệt sử dụng một URI tuyệt đối (định nghĩa bởi máy chủ chứng-thực) như là giá trị của tham số “grant\_type” tới đầu-mút khóa-truy-cập, và thêm vào bất cứ tham số nào cần thiết.

Ví dụ, để yêu cầu một khóa-truy-cập sử dụng một SAML 2.0 (Security Assertion Markup Language) để cấp quyền như định nghĩa tại [[OAuth-SAML2],](https://tools.ietf.org/html/rfc6749#ref-OAuth-SAML2) Client có thể tạo truy vấn HTTP sau sử dụng TLS.

POST /token HTTP/1.1

Host: server.example.com

Content-Type: application/x-www-form-urlencoded

grant\_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Asaml2-

bearer&assertion=PEFzc2VydGlvbiBJc3N1ZUluc3RhbnQ9IjIwMTEtMDU

[...omitted for brevity...]aG5TdGF0ZW1lbnQ-PC9Bc3NlcnRpb24-

Nếu yêu cầu là hợp lệ và xác nhận được, máy chủ chứng thực cấp phát một khóa-truy-cập và tùy lựa chọn mà có thể cấp phát khóa-đổi-mới như mô tả tại phần 5.1 . Nếu việc yêu cầu thất bại đo không xác nhận được hay do không hợp lệ, máy chủ chứng thực trả lại một phản hồi về lỗi như mô tả tại phần 5.2.

1. Cấp phát khóa-truy-cập

Nếu yêu cầu cấp phát khóa-truy-cập là hợp lệ và được xác-nhận, máy chủ chứng-thực cấp phát một khóa truy-cập và tùy theo việc sử dụng mà có thể cấp phát khóa-đổi-mới như mô tả tại phần 5.1 . Nếu việc yêu cầu thất bại do không xác nhận được client hoặc không hợp lệ, máy chủ chứng-thực trả về một phản hồi về lỗi như mô tả tại phần 5.2

* 1. Phản hồi khi yêu cầu thành công

Máy chủ chứng-thực phát sinh một khóa-truy-cập và tùy theo việc sử dụng mà phát sinh thêm khóa-đổi-mới, và sắp xếp phản hồi bằng cách thêm vào các tham số sau vào chiếm toàn bộ phần thân của phản hồi HTTP với mã trạng thái OK(200) :

access\_token : BẮT BUỘC, khóa-truy-cập được cấp phát bởi máy chủ chứng-thực.

token\_type : BẮT BUỘC , loại của khóa vừa được cập phát như mô tả tại phần 7.1 . Giá trị không phân biệt hoa thường.

expires\_in : KHUYÊN DÙNG , thời gian tồn tại tính bằng giây của khóa-truy-cập. Ví dụ, giá trị “3600” hàm ý rằng khóa-truy-cập sẽ hết hạn sau 1 giờ kể từ khi phản hồi được tạo. Nếu có dùng, máy chủ chứng-thực NÊN cung cấp giời gian hết hạn thông qua cách này hay khác hoặc thông qua tài liệu về giá trị mặc định.

refresh\_token : TÙY CHỌN , khóa-đổi-mới, có thể được dùng để đổi khóa-truy-cập mới sử dụng cùng loại cấp quyền như mô tả t ại phần 3.3.

scope : là TÙY CHỌN nếu phạm vi yêu cầu được xác định bởi client; là BẮT BUỘC ở các trường hợp khác. Phạm vi của khóa-truy-cập mô tả tại phần 3.3

Các tham số được thêm vào phần thân của phản hồi HTTP sử dụng nội dung loại “application/json” như định nghĩa tại [[RFC4627](https://tools.ietf.org/html/rfc4627)]. Những tham số được nén thành một JavaScript Object Notation (JSON) cấu trúc bằng các thêm các tham số vào node trên cùng. Tên tham số và các giá trị dạng chuỗi được thêm vào sử dụng JSON string. Giá trị số được thêm vào như JSON number. Thứ tự của tham số không quan trọng.

Ví dụ:

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"

}

Client PHẢI bỏ qua các giá trị không xác định trong phản hồi. Kích thước của khóa-truy-cập và các giá trị khác trong phản hồi từ máy chủ chứng-thực là rõ ràng. Client tránh tạo ra các định đoán về kích thược. Máy chủ chứng-thực NÊN có tài liệu về kích thước của bất cứ giá trị nào nó phát sinh.

* 1. Phản hồi về lỗi

Máy chủ chứng-thực phản hồi với một mã trạng thái HTTP 400(Bad Request) và đính kèm những tham số sau với phản hồi:

error : BẮT BUỘC . Một mã lỗi ASCII [[USASCII](https://tools.ietf.org/html/rfc6749#ref-USASCII)] như sau:

* invalid\_request : yêu cầu bị thiếu các tham số cần thiết, kèm theo một tham số không hợp lệ, kèm theo nhiều hơn một tham số (trùng nhau), hoặc các lỗi định dạng khác.
* unauthorized\_client : không thể xác định client để yêu cầu mã-chứng-thực.
* access\_denied : Chủ sở hữu tài nguyên hoặc máy chủ chứng-thực không chấp nhận yêu cầu.
* unsupported\_response\_type : Máy chủ chứng-thực không hỗ trợ việc lấy mã-chứng-thực dùng phương pháp này.
* invalid\_scope : phạm vi yêu cầu không hợp lệ, không xác định hoặc lỗi định dạng.
* server\_error : Máy chủ chứng thực gặp một trường hợp không mong muốn ngăn cản nó đáp ứng yêu cầu. (mã lỗi này là cần thết bởi một mã trạng thái 500 của giao thức HTTP không thể gửi cho client thông qua phản hồi chuyển hướng của giao thức HTTP).
* temporarily\_unavailable: Máy chủ chứng thực hiện không thể xử lý yêu cầu vì quá tải hoặc bảo trì máy chủ. (mã lỗi này là cần thiết bởi mã trạng thái 503 của giao thức HTTP không thể gửi tới client thông qua phản hồi chuyển hướng của giao thức HTTP)

Giá trị của tham số “error” KHÔNG ĐƯỢC thêm những kí tự nằm ngoài tập %x20-21 / %x23-5B / %x5D-7E.

error\_description : TÙY CHỌN . chuỗi ASCII con người đọc được cung cấp thêm thông tin, sử dụng hỗ trợ lập trình viên phía client hiểu hơn về lỗi đã phát sinh.

error\_uri : TÙY CHỌN . Một URI xác định một trang web con người đọc được có các thông tin về lỗi, sử dụng để cung cấp cho lập trình viên thông tin thêm về lỗi. Giá trị của tham số “error\_uri” PHẢI tương thích với cú pháp của URI và KHÔNG ĐƯỢC chứa các kí tự ngoài tập %x21 / %x23-5B / %x5D-7E.

Các tham số được thêm vào phần thân của phản hồi HTTP sử dụng nội dung loại “application/json” như định nghĩa tại [[RFC4627](https://tools.ietf.org/html/rfc4627)]. Những tham số được nén thành một JavaScript Object Notation (JSON) cấu trúc bằng các thêm các tham số vào node trên cùng. Tên tham số và các giá trị dạng chuỗi được thêm vào sử dụng JSON string. Giá trị số được thêm vào như JSON number. Thứ tự của tham số không quan trọng.

Ví dụ:

HTTP/1.1 400 Bad Request

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

Cache-Control: no-store

Pragma: no-cache

{

"error":"invalid\_request"

}

1. Đổi mới một khóa-truy-cập

Nếu máy chủ chứng thực cấp phát một khóa-đổi-mới cho client, client tạo một yêu cầu đổi mới đến đầu-mút khóa-truy-cập và thêm các tham số sau sử dụng biến đổi “application/x-www-form-urlencoded” với mã hóa UTF-8 trong toàn bộ phần thân truy vấn HTTP:

grand\_type : BẮT BUỘC , giá trị PHẢI là “refresh\_token”

refresh\_token : BẮT BUỘC , khóa-đổi-mới được cấp cho client.

scope : TÙY CHỌN , phạm vi của yêu cầu truy cập như mô tả tại phần 3.3 . Phạm vi yêu cầu PHẢI KHÔNG bao gồm bất kỳ phạm vi nào không nằm trong vùng đã được cấp phép bởi chủ sở hữu tài nguyên.

Bởi vì khóa-đổi-mới giống như chứng-nhận dài hạn dùng để yêu cầu thêm khóa-truy-cập, khóa-đổi-mới phụ thuộc vào client mà nó được phát sinh cho. Nếu client loại tin cậy hoặc client được cấp các chứng-thư (hoặc đủ các yêu cầu theo các cách chứng nhận khác), client PHẢI xác-nhận với máy chủ chứng-thực như mô tả tại phần 3.2.1

Ví dụ, client tạo truy vấn HTTP sử dụng lớp bảo mật.

POST /token HTTP/1.1

Host: server.example.com

Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW

Content-Type: application/x-www-form-urlencoded

grant\_type=refresh\_token&refresh\_token=tGzv3JOkF0XG5Qx2TlKWIA

Máy chủ chứng-thực PHẢI:

* Yêu cầu các client tin cậy hoặc bất kỳ client nào được cấp chứng từ xác nhận mình.
* Xác-nhận client nếu việc chứng thực client là cần thiết và đảm bảo rằng khóa-đổi-mới đã được cấp cho client đã chứng-thực, và.
* Kiểm tra khóa-đổi-mới.

Nếu hợp lệ và chứng thực được client, máy chủ chứng-thực cấp phát một khóa-truy-cập như mô tả tại phần 5.1. Nếu việc yêu cầu thất bại trong việc xác định hoặc không hợp lệ, máy chủ chứng-thực trả về một phản hồi về lỗi như mô tả tại phần 5.2.

Máy chủ chứng-thực CÓ THỂ cấp phát một khóa-đổi-mới mới, trong trường hợp đó client PHẢI bỏ đi khóa-đổi-mới cũ và thay thế nó bằng cái mới. Máy củ chứng-thực CÓ THỂ đào thải khóa-đổi-mới cũ sau khi cấp phát cái mới cho client. Nếu một khóa-đổi-mới mới được cấp phát, phạm vi truy cập của khóa-đổi-mới PHẢI client chỉ ra trong yêu cầu.

1. Truy cập nguồn tài nguyên được bảo vệ.

Client truy cập tài nguyên được bảo vệ bằng cách trình ra khóa-truy-cập cho máy chủ chứng-thực. Máy chủ tài-nguyên PHẢI kiểm tra khóa-truy-cập và đảm bảo rằng nó chưa hết hạn và phạm vi của nó bao gồm những tài nguyên được yêu cầu. Phương pháp sử dụng bởi máy chủ tài nguyên để kiểm tra khóa-truy-cập (cũng như là bất kì phản hồi lỗi nào) nằm ngoài phạm vi của tài liệu nào nhưng nó thường đòi hỏi một trao đổi và hợp tác giữa máy chủ tài nguyên và máy chủ chứng-thực.

Phương pháp mà client sử dụng khóa-truy-cập để xác-nhận với máy chủ tài nguyên phụ thuộc vào loại của khóa-truy-cập được máy chủ chứng-thực cấp. Thông thường, nó đòi hỏi sử dụng trường header “Authorization” của truy vấn HTTP [[RFC2617](https://tools.ietf.org/html/rfc2617)] với một lược đồ xác-nhận được định nghĩa bởi đặc tả về loại khóa-truy-cập được sử dụng, như là [[RFC6750](https://tools.ietf.org/html/rfc6750)]

* 1. Các loại khóa-truy-cập

Loại khóa-truy-cập cung cấp cho client với thông tin cần thiết để vận dụng thành công khóa-truy-cập để tạo yêu cầu truy cập nguồn tài nguyên được bảo vệ (dựa vào những thuộc tính đặc biệt). Client KHÔNG ĐƯỢC sử dung một khóa-truy-cập nếu nó không hiểu loại của khóa đó.

Ví dụ, loại khóa “bearer” định nghĩa tại [[RFC6750](https://tools.ietf.org/html/rfc6750)] được sử dụng đơn giản bằng cách thêm khóa-truy-cập vào yêu cầu:

GET /resource/1 HTTP/1.1

Host: example.com

Authorization: Bearer mF\_9.B5f-4.1JqM

Trong khi loại khóa “mac” định nghĩa tại [[OAuth-HTTP-MAC](https://tools.ietf.org/html/rfc6749#ref-OAuth-HTTP-MAC)] được sử dụng bằng cách phát sinh một khóa MAC (Message Authentication Code) kèm theo với khóa-truy-cập để kí đảm bảo các thành phần của truy vấn HTTP:

GET /resource/1 HTTP/1.1

Host: example.com

Authorization: MAC id="h480djs93hd8",

nonce="274312:dj83hs9s",

mac="kDZvddkndxvhGRXZhvuDjEWhGeE="

Những ví dụ trên được đưa ra chỉ nhằm cung cấp cái nhìn trực quan. Nhà phát triển được khuyến cáo tham khảo đặc tả [[RFC6750](https://tools.ietf.org/html/rfc6750)] và [[OAuth-HTTP-MAC](https://tools.ietf.org/html/rfc6749#ref-OAuth-HTTP-MAC)] trước khi sử dụng.

Mỗi định nghĩa của một loại khóa-truy-cập xác định những thuộc tính kèm theo (nếu có) gửi cho client cùng với tham số “access\_token” trong phản hồi. Nó cũng định nghĩa phương pháp xác-thực HTTP dùng kèm với khóa-truy-cập khi tạo truy vấn yêu cầu nguồn tài nguyên được bảo vệ.

* 1. Phản hồi về lỗi

Nếu việc yêu cầu tài nguyên thất bại, máy chủ tài-nguyên NÊN thông báo cho client về lỗi. Mặc dù chi tiết của phản hồi về lỗi nằm ngoài phạm vi của tài liệu này, tài liệu này thiết lập một số mục chung tại phần 11.4 cho các giá trị lỗi để chia sẽ giữa những lược-đồ chứng thực dùng khóa OAuth.

Các lược-đồ chứng-thực mới được thiết kế dựa trên chứng-thực khóa OAuth NÊN định nghĩa một hệ thống để cung cấp mã lỗi cho client, trong đó các giá trị lỗi được chấp nhận được đăng kí trong những mục-đăng-kí lỗi được thiết lập bởi đặc tả này.

Các lược-đồ CÓ THỂ giới hạn tập các mã lỗi trong tập con của các giá trị đã được đăng kí. Nếu mã lỗi được trả về sử dụng một tham số có tên nào đó thì tên đó NÊN là “error”.

Những lược-đồ khác có khả năng sự dụng chứng-thực khóa OAuth, nhưng không được thiết kế để sử dụng cho mục đích đó, CÓ THỂ đính kèm các giá trị lỗi như trên.

Những lược đồ chứng-thực mới CÓ THỂ chọn để cung cấp kèm theo tham số “error\_description” và “error\_uri” để trả về thông tin về lỗi như cách chúng được sử dụng trong tài liệu này.

1. Mở rộng
   1. Định nghĩa các loại khóa-truy-cập

Các loại khóa-truy-cập có thể được định nghĩa bằng một trong hai cách: đăng kí trong mục-đăng-kí khóa-truy-cập (theo thủ tục trong phần 11.1) hoặc bằng cách sử dụng một địa chỉ URI tuyệt đối duy nhất.

Việc sử dụng các loại theo URI NÊN được giới hạn ở một số phiên bản chính không được ứng dụng nhiều, và được cung cấp chi tiết cài đặt của máy chủ tài-nguyên nơi chúng được sự dụng.

Những loại khác PHẢI được đăng kí, tên của loại khóa-truy-cập PHẢI phù hợp với loại-tên ABNF. Nếu định nghĩa của loại bao gồm một lươc-đồ chứng-thực HTTP, tên của loại NÊN chỉ ra tên lược đồ chưng-thực HTTP được dùng (như định nghĩa bởi [[RFC2617](https://tools.ietf.org/html/rfc2617)]). Loại khóa-truy-cập “example” được cung cấp để dùng cho các ví dụ.

type-name = 1\*name-char

name-char = "-" / "." / "\_" / DIGIT / ALPHA

* 1. Định nghĩa tham số cho đầu mút mới.

Những tham số mới (trong truy vấn hoặc phản hồi) để dùng với đầu mút chứng thực hoặc đầu mút khóa-truy-cập được định nghĩa và đăng kí tại mục-đăng-kí tham số OAuth theo thủ tục tại phần 11.2.

Tên của tham số PHẢI phù hợp với param-name ABNF, và cú pháp giá trị của tham số PHẢI được định nghĩa rõ (như là, sử dụng ABNF , hoặc liên hệ tới cú pháp của một tham số có sẵn).

param-name = 1\*name-char

name-char = "-" / "." / "\_" / DIGIT / ALPHA

Các tham số không mở rông dùng trong một số phiên bản đặc biệt thì không thường được sử dụng và được chỉ ra trong tài liệu cài đặt chi tiết của máy chủ chứng thực nơi chúng được sử dụng. Các tham số đó NÊN sử dụng một tiền tố chỉ phiên bản đặc biệt để không xung đột với những giá tị đã được đăng kí (như là, bắt đầu với ‘companyname\_’).

* 1. Định nghĩa loại cấp quyền mới.

Loại cấp quyền mới có thể được định nghĩa bằng cách gán cho nó một địa chỉ URI tuyệt đối duy nhất để sử dụng tham số “grant\_type”. Nếu loại cấp phép mở rộng đòi hỏi những tham số thêm vào tại đầu-mút khóa-truy-cập, chúng PHẢI được đăng kí trong mục-đăng-kí tham số OAuth như mô tả tại phần 11.2

* 1. Định nghĩa loại phản hồi mới cho đầu mút chứng-thực.

Loại phản hồi mới dùng với đầu mút chứng-thực được định nghĩa và đăng kí trong mục-đăng-kí các loại phản hồi của đầu-mút chứng-thực theo thủ tục tại phần 11.3. Tên của loại phản hồi mới PHẢI phụ hợp với response-type ABNF .

response-type = response-name \*( SP response-name )

response-name = 1\*response-char

response-char = "\_" / DIGIT / ALPHA

Nếu một loại phản hồi chứa một hoặc nhiều kí tự khoảng-trắng (%20), nó được xem như một danh sách các giá trị ngăn cách nhau bởi khoảng trắng và thứ tự của các giá trị này không quan trọng. Chỉ một thứ tự cần đăng kí, nó sẽ bao gồm tất cả các hoán vị của tập giá trị.

Ví dụ, loại phản hồi “token code” không được định nghĩa bởi tài liệu này, tuy nhiên, một phần mở rộng có thể định nghĩa và đăng kí loại phản hồi “token code”. Một khi đã đăng kí, những kiểu kết hơp có cùng giá trị không thể dùng để đăng kí ví dụ “code token”, nhưng cả hai cùng có thể được sử dụng để chỉ cùng một loại phản hồi.

* 1. Định nghĩa thêm mã lỗi

Trong trường hợp khi các phần mở rộng của giao thức ( như là , các loại khóa-truy-cập, tham số mở rộng, hoặc loại cấp quyền mở rộng) đòi hỏi thêm vào những mã lỗi để sử dụng cùng với phản hồi lỗi của quá trình chứng thực (phần 4.1.2.1), phản hồi lỗi của quá trình cấp quyền gián tiếp (4.2.2.1), phản hồi lỗi của các khóa (phần 5.2), hoặc phản hồi lỗi trng quá trình truy cập tài nguyên (phần 7.2), những mã lỗi CÓ THỂ được định nghĩa.

Mã lỗi mở rộng PHẢI được đăng kí (theo thủ tục ở phần 11.4) nếu phần mở rộng mà chúng được dùng với là một loại khóa truy-cập đã đăng kí, một tham số đã đăng kí, hoặc một loại cấp quyền đã đăng kí. Những mã lỗi sử dụng với các phần mở rộng chưa đăng kí cũng CÓ THỂ được đăng kí.

Những mã lỗi PHẢI phù hợp với error ABNF và NÊN có phần tiền tố xác định tên rõ nhất có thể, Ví dụ, một lỗi chỉ ra một tập giá trị không hợp lệ đối với một tham số mở rộng có tên “example NÊN có thên là “example\_invalid”.

error = 1\*error-char

error-char = %x20-21 / %x23-5B / %x5D-7E

1. Ứng dụng Native

Những ứng dụng native là những client được cài đặt và thực thi trên thiết bị của chủ sở-hữu tài-nguyên (như là, ứng dụng desktop, ứng dụng mobile native). Các ứng dụng native đòi hỏi sự đánh giá cẩn thận về bảo mật, khả năng của nền tảng, kinh nghiệm người dùng.

Đầu-mút chứng thực yêu cầu giao tiếp giữa client và tác-nhân tương-tác của chủ sở-hữu tài-nguyên. Các ứng dụng native có thể yêu cầu các tác-nhân tương-tác mở rộng hoặc nhúng sẵn một tác nhân tương-tác bên trong ứng dụng. Ví dụ:

* Tác-nhân tương-tác mở rộng - ứng dụng native có thể bắt phản hồi từ máy chủ chứng thực sử dụng một URI chuyển-hướng với một lược-đồ đã đăng kí với hê-điều-hành để gọi client xử lý, sao/chép các chứng-thư, chạy một máy chủ web cục bộ, cài đặt một tác-nhân tương-tác mở rộng, hoặc cung cấp một URI chuyển hướng đến một máy chủ chứa tài-nguyên nằm dưới sự kiểm soát của client, những thứ giúp phản hồi đến được ứng dụng native.
* Tác-nhân tương-tác nhúng sẵn - ứng dụng native nhận lấy phản hồi bằng cách trực tiếp liên lạc với tác-nhân tương-tác được nhúng sẵn, bằng cách theo dõi các thay đổi trạng thái trong quá trình tải tài nguyên, hoặc truy cập vùng lưu trữ các cookie của tác-nhân tương-tác.

Khi chọn giữa tác-nhân mở rộng hay nhúng sẵn, lập trình viên nên đánh giá những yếu tố sau:

* Một tác-nhân tương-tác có thể cải thiện tốc độ, như là chủ tài nguyên có thể đã kích hoạt một phiên làm việc với máy chủ chứng-thực, nên không cần phải chứng-thực lại. Nó cung cấp một trải nghiệm người dùng thống nhất. Chủ sở-hữu tài-nguyên cũng có thể đặc các đặc-tính của tác-nhân tương-tác hoặc các phần mở rông để trợ giúp cho quá trình chứng-thực (như là, quản lý mật-khẩu , bộ đọc thiết bị bảo mật hai lớp).
* Một tác-nhân tương-tác có thể tốt hơn về mặt tiện dụng, như là không yêu cầu phải chuyển đỗi giữa các hoàn-cảnh sử dụng và mở cửa sổ mới.
* Một tác-nhân tương-tác tạo ra các thách thức về bảo mật bởi những người chủ sở-hữu tài-nguyên thì được chứng thực bên trong một cửa sổ không xác định và không có các lớp bảo vệ như với đa số tác-nhân tương-tác mở rộng. Một tác-nhân tương-tác nhúng sẵn hướng người dùng đến việc tin tưởng những yêu cầu không xác định của quá trình chứng-thực (tạo ra tiền đề để dễ bị tấn công bởi phishing attack).

Khi chọn giữa cấp quyền gián tiếp và loại cấp quyền thông qua mã chứng-nhận, nên suy xét những yếu tố sau:

* Những ứng dụng native sử dụng loại cấp quyên sử dụng mã-chứng-nhận NÊN dùng chúng mà không có chứng-thư, vì những ứng dụng native không có khả năng giữ những chứng từ này một cách tin cậy.
* Khi sử dụng cấp quyền gián tiếp, không được trả về một khóa-đổi-mới, yêu cầu lặp lại quá trình chứng thực một khi khóa-truy-cập hết hạn.

1. Các vấn đề về bảo mật

Vì là nền tảng có thể mở rộng và uyển chuyển, vấn đề bảo mật của OAuth dựa vào nhiều nhân tố. Những phần sau này cung cấp cho những nhà cài đặt những hướng dẫn bảo mật tập trung vào ba nhóm client mô tả tại phần 2.1: ứng dụng web, ứng dụng dựa vào tác-nhân tương-tác, ứng dụng native.

Một mô hình bảo mât OAuth đầy đủ và phân tích, cũng như là nền tảng kiến thức về thiết kế giao thức, được cung cấp tại [[OAuth-THREATMODEL](https://tools.ietf.org/html/rfc6749#ref-OAuth-THREATMODEL)] .

* 1. Chứng-thực người dùng

Máy chủ chứng-thực thiết lập chứng-thư client với các client dạng ứng dụng web để phục vụ cho quá trình chứng thực client. Máy chủ chứng-thực được kì vọng rằng đánh giá việc xác thực client mạnh mẽ hơn là việc dùng mật khẩu. Những client là ứng dụng web PHẢI đảm bảo độ tin cậy của mật khẩu của client và những chứng-thư khác.

Máy chủ chứng-thực KHÔNG ĐƯỢC phát sinh mật khẩu hoặc các chứng-thư khác cho ứng dụng native hoặc cho ứng dụng dựa vào tác-nhân tương-tác với mục đích chứng thực client. Máy chủ chứng-thực CÓ THỂ cấp phát một mật khẫu hoặc chứng-thư khác cho bản cài đặt đặc biệt của một ứng dụng native trên môt thiết bị đặc biệt.

Khi việc chứng-thực client là bất khả thi, máy chủ chứng-thực NÊN sử dụng những thứ khác để xác nhận danh tính của client – ví dụ, bằng việc yêu cầu đăng kí URI chuyển hướng của client hoặc nhờ chủ sở-hữu tài nguyên xác-nhận danh tính của client. Một URI chuyển hướng hợp lệ là chưa đủ để xác nhận danh tính của client khi yêu cầu chủ sở-hữu tài nguyên chứng-thực nhưng có thể dùng để tránh việc chuyển chứng-nhận đến client giả mạo sau khi nhận được xác-nhận của chủ sở-hữu tài-nguyên.

Máy chủ chứng-thực phải quan tâm đế vấn đề bảo mật khi giao tiếp với các client không xác nhận và lượng giá để giới hạn việc rò rỉ những chứng-nhận khác (như là khóa-đổi-mới) đã xuất cho những client này.

* 1. Client giả dạng

Một client nguy hiểm có thể giả dạng client khác và cố gắng truy cập vào tài nguyên được bảo vệ, nếu client bị giả thất bại, hoặc không thể trong việc đảm bảo chứng-thư của nó.

Máy chủ chứng-thực PHẢI xác-thực client bất kể khi nào có thể. Nếu máy chủ chứng-thực không thể xác-nhận client dựa trên các đặc điểm tự nhiên, máy chủ chứng-thực PHẢI yêu cầu đăng ký bất kỳ URI chuyển-hướng nào đã dùng để nhận phản hồi về việc xác thực và NÊN sử dụng những các khác để bảo vệ chủ sở-hữu tài-nguyên khỏi những client nguy hiểm. Ví dụ, chủ sở-hữu tài nguyên có thể nhờ sự giúp đỡ của chủ sở hữu hỗ trợ trong việc xác định danh tính client và nguồn gốc của chúng.

Máy chủ chứng-thực NÊN đảm bảo sự rõ ràng trong việc xác nhận chủ sở hữu và cung cấp cho chủ sở hữu thông tin về client và phạm vi cũng như thời gian hết hạn của yêu cầu. Mục đích là để chủ sở hữu xem xét những thông tin của client và có quyết định đồng ý hay không yêu cầu của client.

Máy chủ chứng thực KHÔNG NÊN tự động lặp lại việc chứng thực (mà không có tương tác của chủ sở hữu) mà không xác-nhận client hoặc dựa trên đánh giá để đảm bảo rằng yêu cầu lặp lại đó đến từ client chứ không phải kẻ giả danh.

* 1. Các khóa-truy-cập

Khóa-truy-cập (cũng như là bất kì thuộc tính nào của khóa-truy-cập) PHẢI được đảm bảo trong việc cất giữ và lưu chuyển, và chỉ chia sẽ nội bộ máy chủ chứng thực, các máy chủ tài nguyên mà khóa-truy-cập đó hợp lệ, và client được cấp phát khóa-truy-cập. Khóa-truy-cập PHẢI được lưu chuyển chỉ khi sử dụng TLS như mô tả tại 1.6 với việc chứng thực với máy chủ như định nghĩa tại [[RFC2818](https://tools.ietf.org/html/rfc2818)].

Khi sử dụng kiểu cấp-quyền gián tiếp, khóa-truy-cập được lưu chuyển trong thành phần của URI, thứ có thể rò rỉ cho các bên khác.

Máy chủ chứng-thực PHẢI đảm bảo rằng khóa-truy-cập không thể được tạo, chỉnh sửa hoặc làm giả bởi các bên khác.

Client NÊN yêu cầu khóa-truy-cập với phạm vi truy cập tối thiểu cần thiết. Máy chủ chứng-thực NÊN lấy danh tính client để lựa chọn cách đáp ứng phạm vi được yêu cầu và CÓ THỂ phát sinh khóa-truy-cập với ít quyền hạn hơn phạm vi được yêu cầu.

Đặc tả này không cung cấp phương pháp nào để máy chủ tài-nguyên có thể đảm bảo một khóa-truy-cập vừa được cung cấp bởi client có phải được phát sinh từ máy chủ chứng-thực cho client đó hay không.

* 1. Các khóa-đổi-mới

Máy chủ chứng thực CÓ THỂ cấp phát khóa-đổi mới cho các client là ứng dụng web và ứng dụng native.

Khóa-đổi-mới PHẢI được giữ và lưu chuyển an toàn, và chỉ chia sẻ giữa máy chủ chứng-thực và client được cấp phát. Máy chủ chứng thực PHẢI duy trì mối liên hệ giữa khóa đổi mới và client mà nó được cấp phát cho. Khóa-đổi-mới PHẢI được lưu chuyển khi và chỉ khi sử dụng TLS như mô tả tại 1.6 với việc chứng thực với máy chủ như định nghĩa tại [[RFC2818](https://tools.ietf.org/html/rfc2818)].

Máy chủ chứng thực PHẢI kiểm tra mối liên hệ giữa khóa-đổi-mới và danh tính của client bất kể khi nào danh tính của client có thể xác-nhận được. Khi việc xác-thực client là bất khả thi, máy chủ chứng thực nên triển khai các biện pháp để bảo vệ khóa-truy-cập khỏi việc bị sử dụng cho mục đí xấu.

Ví dụ, máy chủ chứng-thực có thể xoay vòng cấp phát khóa-đổi-mới mới với mỗi phản hồi yêu cầu truy cập khóa-đổi-mới. Khóa-đổi-mới cũ bị hết hạn nhưng vẫn tồn tại ở máy chủ chứng-thực . Nếu một khóa-truy-cập bị lộ và được sử dụng bởi cả kẻ tấn công và client, một trong hai sẽ đưa ra khóa hết hạn và máy chủ chứng-thực sẽ nhận ra các sai phạm.

Máy chủ chứng-thực PHẢI đảm bảo rằng các khóa-đổi-mới không thể được tạo, chỉnh sửa hoặc làm giả bởi các bên khác.

* 1. Mã Chứng-thực (chứng nhận ủy quyền)

Việc chuyển giao chứng-nhận ủy-quyền NÊN thực hiện thông qua một kênh an toàn, và client NÊN yêu cầu sử dụng TLS với URI chuyển hướng của mình nếu URI đó chỉ định một tài nguyên mạng. Khi chứng-nhận ủy-quyền được chuyển thông qua chuyển hướng tác-nhân tương-tác, chúng có nguy cơ bị phát hiện thông qua lịch sử của tác-nhân tương-tác và HTTP headers.

Chứng-nhận-ủy-quyền hoạt động như chứng-thư văn bản thuần túy, dùng để xác-nhận rằng chủ sở hữu đã ủy quyền tại máy chủ chứng-thực là cùng một người chuyển sang client để hoàn thành tiến trình. Cho dù client dựa vào chứng-nhận-ủy-quyền để chứng-thực nó được ủy-quyền của chủ sở-hữu, đầu mút chuyển hướng của client PHẢI sử dụng TLS.

Chứng-nhận ủy-quyền PHẢI có vòng đời ngắn và dùng một lần. Nếu máy chủ chứng-thực quan sát thấy nhiều giao dịch của một chứng-nhận-ủy-quyền yêu cầu khóa-truy-cập, máy củ chứng-thực NÊN loại bỏ tất cả những khóa-truy-cập đã được cấp dựa trên chứng-nhận-ủy-quyền bị đánh cắp.

Nếu có thể xác-nhận được client, máy chủ chứng thực PHẢI xác-nhận client và đảm bảo rằng chứng-nhận-ủy-quyền được cấp cho client đó.

* 1. Authorization code redirection URI manipulation

Khi yêu cầu chứng-thực sử dụng cấp quyền dựa trên chứng-nhận ủy quyền, client có thể xác định một URI chuyển hướng thông qua tham số “redirect\_uri”. Nếu kẻ tấn công có thể kiểm xoát giá trị của URI chuyển hướng, nó có thể làm máy chủ chứng-thực chuyển hướng tác-nhân tương-tác của chủ sở-hữu đến URI nằm đưới sự kiểm soát của kẻ tấn công cùng với chứng-nhận-ủy-quyền.

Kẻ tấn công có thể tạo một tài khoản tại một client hợp lệ và bắt đầu tiến trình chứng-thực. Khi tác-nhân-tương-tác của kẻ tấn công gửi đến máy chủ chứng thực để được cấp quyền, kẻ tấn công lấy URI xác-nhận được cấp bởi client hợp lệ và thay thế URI chuyển hướng của client với URI nằm dưới sự kiểm soát của kẻ tấn công và chứng-thực để truy cập vào client hợp lệ.

Đồng thời tại máy chủ chứng-thực, nạn nhân được hỏi như thường lệ, yêu cầu hợp lệ, và chứng-nhận yêu cầu. Nạn nhân được chuyển hướng đến đầu mút nằm dưới sự kiểm xoát của kể tấn công với chứng-nhận-ủy-quyền. Kẻ tấn công hoàn thành tiến trình chứng-thực bằng cách gửi chứng-nhận-ủy-quyền đến client sử dụng URI chuyển hướng gốc được cung cấp bởi client. Client chuyển chứng-nhận-ủy-quyền thành khóa-truy-cập và liên kết nó với tài khoản của kẻ tấn công, bây giờ kẻ tấn công có thể truy cập tài nguyên của nạn nhân (thông qua client).

Để chống lại loại tấn công này, máy chủ chứng-thực PHẢI đảm bảo rằng URI chuyển hướng sử dụng để yêu cầu chứng-nhận-ủy-quyền là URI được cung cấp khi chuyển đổi chứng-nhận-ủy-quyền thành khóa-truy-cập. Máy chủ chứng thực PHẢI yêu cầu các client công cộng và NÊN yêu cầu các client-đảm-bảo đăng ký URI chuyển hướng của mình. Nếu một URI chuyển hướng được cung cấp trong yêu cầu, máy chủ chứng thực phải kiểm tra nó với những URI đã đăng kí.

* 1. Chứng-thư mật-khẩu của chủ sở hữu tài nguyên.

Kiểu cấp quyền thông qua chứng-thư mật-khẩu thường được dùng cho các vấn đề duy trì hoặc sát nhập. Nó giảm thiểu nguy cơ từ việc lưu trữ tên-đăng-nhập và mật khẩu bởi client nhưng không làm giảm nhu cầu tiết lộ các chứng-thư có đặc quyền cao cho client.

Loại cấp-quyền này mang theo mình rủi ro cao hơn các loại cấp quyền khác bởi nó duy trì việc sử dụng mật-khẩu thứ mà giao thức này được lập ra để tránh. Client có thể làm ảnh hưởng đến mật-khẩu, hoặc mật-khẩu có thể không an toàn trước kẻ tấn công (như là, thông qua file log hoặc những mục lưu trữ khác của client).

Thêm nữa, bởi chủ sở hữu không kiểm xoát xuyên suốt quá trình chứng-thực (chủ sở hữu chỉ tham gia đến khi đưa chứng-thư cho client), client có thể xin khóa-truy-cập với phạm vi rộng hơn phạm vi chủ sở hữu mong muốn. Máy chủ chứng thực nên đánh giá phạm vi và giời gian hết hạn của khóa-truy-cập được cấp bởi dạng cấp quyền này.

Máy củ chứng-thực và client NÊN hạn chế sử dụng loại cấp quyền này và thay thế bởi các dạng khác bất cứ khi nào có thể.

* 1. Độ tin cậy của yêu cầu

Khóa-truy-cập, khóa-đổi-mới, mật khẩu của chủ sở hữu, và chứng-thư cho client KHÔNG ĐƯỢC lưu chuyển ở dạng nguyên mẫu, chứng-nhận-ủy-quyền KHÔNG NÊN lưu chuyển ở dạng nguyên mẫu.

Các tham số “state” và “scope” KHÔNG NÊN kèm theo những thông tin nhạy cảm của client hay chủ sở hữu ở dạng văn bản thuần, vì chúng có thể được lưu trữ ở nơi không an toàn hoặc chuyển qua các kênh không bảo mật.

* 1. Ensuring Endpoint Authenticity

Với mục đích ngăn chặng kiểu tấn công man-in-the-middle, máy chủ chứng-thực PHẢI yêu cầu sử dụng TLS với các máy chủ liên quan đến chứng thực như định nghĩa tại [[RFC2818](https://tools.ietf.org/html/rfc2818)] cho bất kỳ yêu cầu nào gửi đến các đầu mút chứng thực hoặc đầu mút khóa-truy-cập. Client PHẢI kiểm tra chứng nhận TLS của máy chủ chứng-thực như định nghĩa tại [RFC6125] và phù hợp với yêu cầu về nhận daan5t với máy chủ nhận đạng.

* 1. Credentials-Guessing Attacks

Máy chủ chứng-thực PHẢI ngăn chặng những kẻ tấn công bằng cách đoán các khóa, chứng-nhận-ủy-quyền, mã-đổi-mới, mật-khẩu của chủ sở-hữu, và các chứng-thư của client.

Xác xuất của các khóa được tạo ra bởi kẻ tấn công (và các chứng-thư khác) PHẢI nhỏ hơn hoặc bằng 2^(-128) và NÊN nhỏ hơn hoặc bằng 2^(-160).

Máy chủ chứng thực PHẢI sử dụng các cách khác để bảo vệ chứng-thư do người dùng cuối sử dụng.

* 1. Phishing Attacks

Sử phổ biến của OAuth và các giao thức tương tự có thể khiến người dùng cuối trở nên nhàm chán vì phải gõ lại password mỗi lần chuyển hướng websites. Nếu người dùng cuối không cẩn thận xác nhận những websites này trước khi nhập mật khẩu có thể khiến những kẻ tấn công đào sâu trải nghiệm này để trộm mật khẩu của họ.

Người cung cấp dịch vụ nên giúp người dùng nhận thức được sự nguy hiểm của kiểu tấn công phishing attack và nên cung cấp cơ chế để làm mọi thứ trở nên dễ dàng cho người dùng trong việc xác nhận danh tính với site của họ. Những nhà phát triển các ứng dụng client nên đánh giá các cài đặt bảo mật của việc giao tiếp giữa client với tác-nhân-tương-tác (như là dùng phần mở rộng, hay nhúng sẵn), và khả năng người dùng cuối kiểm tra việc xác nhận của máy chủ chứng thực.

Để giảm rủi ro từ phishing attack, máy chủ chứng-thực PHẢI yêu cầu sử dụng TLS trên mỗi đầu mút mà người dùng cuối tiếp xúc.

* 1. Cross-site Request Forgery (CSRF)

Làm một nổ lực của kẻ tấn công khiến cho tác-nhân tương-tác của nạn nhân chuyển hướng đến một URI nguy hiểm (như là , cung cấp cho tác-nhân tương-tác một liên kết sai, hình ảnh, hoặc chuyển hướng) đến một máy chủ không tin cậy (thường được thiết lập thông qua cookie hợp lệ đang hiện diện).

Một cuộc tấn công CSRF ngăn cảm URI chuyển hướng của client cho phép kẻ tấn công trích xuất chứng-nhận-ủy-quyền hoặc khóa-truy-cập, thức có thể là dẫn đến client sử dụng một khóa truy cập gắn với tài nguyên của kẻ tấn công hơn là của nạn nhân (như là, lưu trữ thông tin tài khoản ngân hàng của khách hàng đến một nguồn tài nguyên được bảo vệ và dưới sự kiểm soát của kẻ tấn công).

Client PHẢI cài đặt bảo vệ CSRF cho những URI chuyển-hướng. Điều này thường dẫn đến việc yêu cầu tất cả các truy vấn gửi đến đầu mút URI chuyển hướng đình kèm một giá trị thứ gắn kết truy vấn với trạng thái xác-thực của tác-nhân tương-tác (như là, một mã băm của phiên làm việc mà tác-nhân tương-tác được xác-thực) . Client NÊN sử dụng tham số “state” để chuyển giá trị này cho máy chủ chứng-thực khi tạo yêu cầu chứng-thực.

Một khi việc chứng-thực đã được người dùng cuối xác-nhận, máy chủ chứng-thực chuyển tác-nhân tương-tác của người dùng cuối trở lại với client với giá trị ràng buộc bắt buộc chứa trong tham số “state”. Giá trị ràng buộc cho phép client kiểm tra tính hợp lệ của yêu cầu bằng cách so giá trị này với trạng thái chứng-thực của tác-nhân tương-tác . Giá trị ràng buộc dùng để bảo vệ khỏi GSRF PHẢI chứa một giá trị không đoán định được (như mô tả tại 10.10), và trạng thái chứng-thực của tác-nhân tương-tác của client (như là cookie của phiên làm việc, lưu trữ cục bộ của HTML5 ) PHẢI được giữ ở nơi chỉ truy cập được bởi client và tác-nhân tương-tác (như, bảo vệ bởi chính sách same-origin ).

Một cuộc tấn công CSRF ngăn cản đầu mút chứng-thực của máy chủ chứng-thực có thể dẫn đến việc kẻ tấn công có được chứng thực của người dùng cuối cho một client nguy hiểm mà không đánh động hoặc cảnh báo gì tới người dùng cuối.

Máy chủ chứng-thực PHẢI cài đặt bảo vệ CSRF cho đầu mút chứng-thực và đảm bảo rằng một client nguy hiểm không thể có được chứng-thực mà không có sự cho phép rõ ràng của chủ sở hữu tài nguyên.

* 1. Clickjacking

Trong một cuộc tấn công clickjacking, kẻ tấn công đăng kí một client hợp lệ và thiết lập một site nguy hiểm , nó tải trang web của đầu mút chứng thực của máy chủ chứng thực vào trong một iframe ẩn nằm trên một tập các nút giả được bố trí cẩn thận đặt ngay dưới các nút quan trọng trong trang xác-thực. Khi người dùng cuối click vào một nút được hiển thị, thực tế họ click vào một nút ẩn trên trang xác-thực (như là nút “xác nhận”) . Việc này cho phép kẻ tấn công lừa lấy được sự cấp phép của người dùng cho client của chúng mà người dùng không hay biết.

Để ngăn ngừa dạng tấn công này, những ứng dụng native NÊN sử dụng một thành phần trình duyệt mở rộng thay vì trình duyệt nhúng sẵn bên trong ứng dụng khi yêu cầu người dùng cuối xác-nhận. Hầu hết những trình duyệt mới, việc ngăn ngừa iframe có thể làm bằng cách sử dụng header không chuẩn “x-frame-options”. Header này có thể có hai giá trị, “deny” và “sameorigin”, thứ sẽ chặn bất kì frame nào, hoặc những frame bởi site không cùng nguồn gốc. Tương tự, cho những trình duyệt cũ hơn, kĩ thuật Javascript frame-busting có thể dùng thay thế, nhưng sẽ không hiệu nghiệm trên tất cả các trình duyệt.

* 1. Code Injection and Input Validation

Một cuộc tấn công xâm nhập mã nguồn xuất hiện khi một dữ liệu nhập hoăc các ngoại-biến khác được dùng bởi ứng dụng không cẩn thận và có thể thay đổi giá trị để thay đổi logic của ứng dụng. Loại tấn công này cho phép kẻ tấn công truy cập được vào thiết bị chứa ứng dụng và dữ liệu của nó, khiến dừng dịch vụ, hoặc có nhiều ảnh hưởng nguy hiểm sâu rộng.

Máy chủ chứng-thực và client PHẢI cẩn thận (và kiểm tra khi có thể) bất kì giá trị nào nhận được – cụ thể, giá trị của các tham số “state” và “redirect\_uri”.

* 1. Open Redirectors

Máy chủ chứng-thực, đầu mút chứng-thực, và đầu mút chuyển hướng của client có thể không được cấu hình chắc chắn và hoạt động như những chuyển-hướng mở. Một chuyển-hướng mở là một đầu mút sử dụng một tham số để tự động chuyển hướng tác-nhân tương-tác đến địa chỉ ghi trong giá trị của tham số mà không kiểm tra.

Chuyển-hương-mở có thể được dùng trong phishing attack, hoặc bởi những kẻ tấn công để khiến người dùng truy cập vào các website nguy hiểm. Thêm nữa, nếu máy chủ chứng thực cho phép client đăng kí chỉ một phần của URI chuyển-hướng, kẻ tấn công có thể sử dụng chuyển-hướng-mở hoạt động trên một client để cấu trúc một URI chuyển hướng có thể qua được vòng kiểm tra của máy chủ chứng-thực và gửi chứng-nhận-ủy-quyền hoặc khóa-truy-cập đến đầu mút nằm dưới sự kiểm soát của kẻ tấn công.

* 1. Misuse of Access Token To Impersonate Resource Owner In Implicit Flow

Cho các client-công-cộng sử dụng quy trình chứng thực gián tiếp, tài liệu này không cung cấp phương pháp nào để client xác định khóa-truy-cập được cấp phát cho client nào.

Chủ sở hữu tài nguyên có thể sẽ chuyển quyền truy cập vào một tài nguyên thông qua cấp quyền cho một khóa-truy-cập cho một client nguy hiểm của kẻ tấn công. Điều này có thể do rò rỉ thông tin hoặc các nguyên do xấu khác. Kẻ tấn công có thể cũng lấy cắp khóa thông qua các cơ chế khác. Kể tấn công có thể giả dạng là chủ sở hữu bằng cách cung cấp khóa-truy-cập cho các client-công-cộng hợp lệ.

Trong một tiến trình gián tiếp (response\_type=token), kẻ tấn công có thể dễ dàng đánh tráo khóa trong phản hồi của máy chủ bằng cái đã được cấp phát cho kẻ tấn công trước đó.

Việc liên lạc giữa các máy chủ với những ứng dụng native dựa trên việc chuyển một khóa-truy-cập ở một kênh ngầm để xác định danh tính của người dùng của client có thể giống với việc kẻ tấn công tạo một ứng dụng mà có thể thâm nhập tùy ý trộm khóa-truy-cập.

Bất kỳ client-công-cộng nào giả định rằng chỉ có chủ sở hữu tài nguyên mới có thể đưa ra một khóa-truy-cập hợp lệ thì đều dễ bị tổn thương bởi loại hình tấn công này.

Loại tấn công này có thê lộ thông tin về chủ sở hữu tại client hợp lệ cho kẻ tấn công (client nguy hiểm). Điều này cũng có thể cho phép kẻ tấn công thực hiện các hành động trên client hợp lệ với cùng quyền hạn với chủ sở hữu tài nguyên.

Việc xác-thực chủ sở hữu cho client nằm ngoài phạm vi của tài liệu này. Bất kỳ đặc tả nào sử dụng tiến trình chức thực dạng như trao quyền hạn của người dùng hợp lệ cho client (như là dịch vụ đăng nhập của bên thứ ba) KHÔNG ĐƯỢC sử dụng tiến trình gián tiếp mà không có các cơ chế bảo mật cho phép client xác định khóa-truy-cập có được cấp phát cho nó sử dụng hay không (như là, hạn chế khóa-truy-cập)

1. IANA Considerations
   1. Oauth Access Token Types Registry
      1. Registration Template
   2. Oauth Parameters Registry
      1. Registration Template
      2. Initial Registry Contents
   3. Oauth Authorization Endpoint Respoinse Types Registry
      1. Registration Template
      2. Initial Registry Contents
   4. Oauth Extensions Error Registry
      1. Registration Template
2. References
   1. Normative References
   2. Informative References