2.3. Client Authentication bölümü, istemci türü "confidential" (gizli) olan bir istemcinin, yetkilendirme sunucusuyla nasıl kimlik doğrulaması yapması gerektiğini açıklar. Bu bölümde, istemci doğrulaması için kullanılan yöntemlerin güvenlik gereksinimlerine dayalı olarak nasıl yapılandırılacağına dair önemli bilgiler bulunmaktadır.

1. Confidential Client (Gizli İstemci)

Confidential client, istemci kimlik bilgilerini güvenli bir şekilde saklayabilen bir istemci türüdür. Genellikle, istemci uygulaması, bir **sunucu** üzerinde çalışır ve istemci kimlik bilgileri (örneğin, parola, özel anahtar ve kamu anahtar çifti gibi) bu sunucuda güvenli bir şekilde saklanır.

- **Gizli İstemciler için Kimlik Doğrulama Yöntemleri**: Yetkilendirme sunucusu, gizli istemci türü için güvenlik gereksinimlerine uygun bir **kimlik doğrulama yöntemi** belirler. Bu, şifre tabanlı bir kimlik doğrulama, dijital sertifikalar veya kamu/özel anahtar çifti gibi farklı yöntemler olabilir. Yetkilendirme sunucusu, istemcinin güvenli bir şekilde kimliğini doğrulamak için bu kimlik doğrulama yöntemlerini kabul edebilir. Bu, istemcinin kimlik doğrulaması yapılırken sunucu ve istemci arasındaki güvenli iletişimi sağlar.
- **Client Credentials**: Gizli istemciler, genellikle bir **set** (takım) istemci kimlik bilgisi (client credentials) alır. Bu kimlik bilgileri, istemcinin yetkilendirme sunucusuyla güvenli bir şekilde kimliğini doğrulaması için kullanılır. Örneğin, istemci bir **parola** ya da **kamu/özel anahtar çifti** kullanabilir. Bu kimlik bilgileri istemciyle birlikte güvenli bir şekilde saklanır.

2. Public Client (Halka Açık İstemci)

Public client, istemci kimlik bilgilerini güvenli bir şekilde saklayamayan ve genellikle bir kullanıcının cihazında çalışan istemci türüdür. Örneğin, web tarayıcılarında veya mobil cihazlarda çalışan istemciler, istemci kimlik bilgilerini güvenli bir şekilde saklamakta zorluk yaşayabilirler. Bu nedenle, **public client** istemcileri için kimlik doğrulaması yapılmaz.

- Yetkilendirme sunucusu, **public client** istemcileri için kimlik doğrulama yöntemleri belirleyebilir. Ancak, public client kimlik doğrulaması **istemcinin tanımlanması** için kullanılmamalıdır. Yani, **public client** türündeki istemciler, kimlik doğrulama için **client authentication** yöntemlerine dayanmazlar.

3. Birden Fazla Kimlik Doğrulama Yöntemi Kullanılamaz

Bir istemci, aynı istek içinde birden fazla kimlik doğrulama yöntemi kullanmamalıdır. Bu, istemcinin güvenliğini ve iletişimini gereksiz yere karmaşıklaştırabilir ve sunucu için güvenlik açıklarına yol açabilir. Örneğin, istemci aynı istekte hem şifre hem de anahtar doğrulaması kullanamaz. İstemci yalnızca bir kimlik doğrulama yöntemi seçmeli ve bu yöntemi kullanmalıdır.

4. Özet

- **Confidential client** için, yetkilendirme sunucusu güvenlik gereksinimlerine uygun bir kimlik doğrulama yöntemi belirler ve istemci, bu yöntemle kimlik doğrulaması yapar.
- Gizli istemciler, genellikle bir set **client credentials** (kimlik bilgileri) alır ve bu bilgileri güvenli bir şekilde kullanır.
- **Public client** istemcileri, kimlik doğrulama için yalnızca token (erişim belirteci) gibi mekanizmalar kullanır, kimlik doğrulama amacıyla istemci bilgileri kullanılamaz.

- İstemci, her istekte yalnızca **bir** kimlik doğrulama yöntemi kullanabilir.

Bu bölüm, istemcinin, güvenli bir kimlik doğrulama süreciyle yetkilendirme sunucusuyla iletişim kurmasını ve doğru güvenlik gereksinimlerine dayalı olarak işlem yapmasını sağlar.

2.3.1. Client Password bölümü, istemcilerin kimlik doğrulamasını yapmak için **client password** (istemci parolası) kullanma yöntemini açıklar. Bu yöntem, istemcinin kimliğini doğrulamak için **HTTP Basic Authentication** şemasını kullanmayı içerir. Ayrıca, istemci kimlik bilgileriyle birlikte gönderilebilecek alternatif bir yöntem ve güvenlik önlemleri de tartışılmaktadır.

1. HTTP Basic Authentication

İstemciler, bir **client password** (istemci parolası) sahibi olduklarında, yetkilendirme sunucusuyla kimlik doğrulaması yapmak için **HTTP Basic Authentication** şemasını kullanabilirler. Bu şema, RFC 2617'de tanımlanmıştır.

- **Client Password Kullanımı**: İstemci, kendisini tanıtmak ve kimliğini doğrulamak için HTTP isteği içerisinde, **client identifier** (istemci kimlik bilgisi) ve **client password** (istemci parolası) bilgisini içerir. Bu bilgileri, "application/x-www-form-urlencoded" kodlama algoritması ile encode eder.
    - **Client Identifier**: İstemcinin benzersiz kimliğidir.
    - **Client Password**: İstemcinin gizli parolasıdır.

  Bu bilgilerin her biri, HTTP isteğinde uygun şekilde encode edilerek, **Authorization header** (Yetkilendirme başlığı) ile gönderilir. Örnek olarak:

Authorization: Basic czZCaGRSa3F0Mzo3RmpmcDBaQnIxS3REUmJuZlZkbUl3

Burada, `Basic` *anahtar kelimesi, temel kimlik doğrulamasının kullanılacağını belirtir ve ardından client identifier ile client password bilgileri encode edilerek yer alır.*

- **Yetkilendirme Sunucusunun Desteklemesi**: Yetkilendirme sunucusu, client password kullanan istemciler için **HTTP Basic Authentication** şemasını desteklemek zorundadır. Bu, istemcinin kimlik doğrulaması yapabilmesi için gereklidir.

2. Alternatif Kimlik Doğrulama Yöntemi

Yetkilendirme sunucusu, istemci kimlik bilgilerini **request body** (istek gövdesi) içinde de alabilir. Bu, client password yerine, istemci kimlik bilgilerini aşağıdaki parametrelerle göndermeyi içerir:

- **client_id**: İstemciye, kayıt sırasında verilen benzersiz kimlik bilgisi.
- **client_secret**: İstemcinin gizli parolası.

Bu yöntemin kullanımı, özellikle HTTP Basic Authentication şemasını veya başka bir parola tabanlı kimlik doğrulama şemasını doğrudan kullanamayan istemciler için uygundur. Ancak, bu yöntemin **request URI** (istek URI’sı) içinde yer almaması gerektiği belirtiliyor, sadece isteğin gövdesinde kullanılmalıdır.

Örnek kullanım:

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

3. TLS (Transport Layer Security) Zorunluluğu

Yetkilendirme sunucusu, istemci kimlik doğrulaması için parola tabanlı bir yöntem kullanıldığında, **TLS** (Transport Layer Security) protokolünün kullanılmasını zorunlu kılar. TLS, verilerin güvenli bir şekilde iletilmesini sağlar ve parola gibi hassas bilgilerin güvenliğini artırır. Bu, istemci kimlik bilgileri ile yapılan her istekte **TLS** kullanılması gerektiğini belirtir.

4. Brute Force (Kaba Kuvvet) Saldırılarına Karşı Koruma

Bu kimlik doğrulama yöntemi, **client password** (istemci parolası) içerdiği için, yetkilendirme sunucusu, kaba kuvvet saldırılarına karşı koruma sağlamak zorundadır. Kaba kuvvet saldırıları, kötü niyetli bir saldırganın parolaları tahmin etmeye çalışarak istemci kimlik doğrulamasını kırmaya çalıştığı saldırılardır. Yetkilendirme sunucusu, bu tür saldırılara karşı savunma mekanizmaları eklemelidir.

Özetle:

- **Client Password** yöntemi, istemcilerin **HTTP Basic Authentication** şemasını kullanarak kimlik doğrulaması yapmasına olanak sağlar. Bu şemada, istemci kimlik bilgileri (client identifier ve client password) HTTP isteği içerisinde gönderilir.
- Alternatif olarak, istemci kimlik bilgileri isteğin gövdesinde **client_id** ve **client_secret** parametreleri olarak da gönderilebilir, ancak bu yöntem genellikle önerilmez.
- **TLS** kullanımı zorunludur; bu, istemci parolasının güvenli bir şekilde iletilmesi için gereklidir.
- Ayrıca, sunucunun kaba kuvvet saldırılarına karşı savunmasız olmaması için gerekli güvenlik önlemleri alması gerekir.

Bu bölüm, istemcilerin güvenli bir şekilde kimlik doğrulaması yapmalarını sağlamak için önemli bilgiler ve güvenlik önlemleri sunar.

2.3.2. Other Authentication Methods *bölümü, yetkilendirme sunucularının istemci kimlik doğrulaması için* ***HTTP authentication*** *şemalarını esnek bir şekilde kullanabilmelerini açıklamaktadır. Bu bölümde, yetkilendirme sunucusunun* ***HTTP Basic Authentication*** *dışında başka kimlik doğrulama yöntemlerini de destekleyebileceği ve bu yöntemlerin nasıl uygulanması gerektiği belirtiliyor.*

1. Diğer Kimlik Doğrulama Yöntemleri (Other Authentication Methods)

Yetkilendirme sunucusu, **HTTP authentication** (HTTP kimlik doğrulama) için gereksinimlerini karşılayan herhangi bir uygun kimlik doğrulama şemasını destekleyebilir. Yani, sadece **HTTP Basic Authentication** değil, başka kimlik doğrulama yöntemleri de kullanılabilir.

MAY ifadesi, yetkilendirme sunucusunun ihtiyaca göre farklı kimlik doğrulama şemalarını kullanabileceği, ancak bunun zorunlu olmadığı anlamına gelir. Örneğin:

- **OAuth 2.0 Bearer Token** gibi daha modern ve güvenli kimlik doğrulama yöntemleri,
- **OAuth 2.0 Client Assertion** (istemci iddiaları) gibi alternatifler,
- **OAuth 2.0 JWT (JSON Web Token)** gibi teknolojiler,
- Diğer özel şemalar (örneğin, şirketin iç güvenlik standartlarına uygun özel şemalar) kullanılabilir.

2. İstemci Kimliği ve Kimlik Doğrulama Yöntemi Arasındaki Eşleştirme

Eğer yetkilendirme sunucusu, **diğer kimlik doğrulama yöntemlerini** kullanıyorsa, bu durumda **client identifier** (istemci kimliği) ile belirli bir kimlik doğrulama şeması arasında açık bir eşleştirme yapılması gerekmektedir. Yani, her istemci kaydı, kullanılan kimlik doğrulama şemasıyla ilişkilendirilmelidir.

- Örneğin, bir istemci kayıtlıysa ve bu istemci özel bir kimlik doğrulama yöntemi kullanacaksa, yetkilendirme sunucusu, bu istemci kaydını o yönteme uygun şekilde eşleştirmelidir.
- Bu eşleştirme, istemci kayıtlarında belirtilmeli ve yetkilendirme sunucusu bu eşleştirmeyi kullanarak hangi kimlik doğrulama şemasının uygulanacağını belirlemelidir.

3. Kimlik Doğrulama Yöntemi Belirlenmesi

- Yetkilendirme sunucusu, desteklediği kimlik doğrulama şemalarının her birinin güvenlik gereksinimlerini yerine getirdiğinden emin olmalıdır.
- Her kimlik doğrulama şeması, istemcinin güvenliğini sağlamak için **kimlik doğrulama ve yetkilendirme** süreçlerinde gereksinimlere uygun olmalıdır. Örneğin, bazı kimlik doğrulama yöntemleri, güvenli olmayan ağlarda kullanılmaya uygun olmayabilir ve bu nedenle sadece TLS (Transport Layer Security) ile korunmuş bağlantılarla sınırlı olmalıdır.

4. Eşleştirme Örneği

Bir istemci kaydında, istemci kimliği **client_id** belirtilir. Bununla birlikte, bu istemciye uygulanacak kimlik doğrulama şeması da tanımlanmalıdır. Örneğin:

- Eğer istemci, **OAuth 2.0 Client Assertion** kullanarak kimlik doğrulaması yapacaksa, yetkilendirme sunucusu bu istemcinin kaydını **Client Assertion** ile ilişkilendirebilir.
- Eğer istemci **Basic Authentication** kullanacaksa, bu kimlik doğrulama şeması da ilgili kayıtta belirtilir.

5. Özet

- **Yetkilendirme sunucusu**, **HTTP authentication** için yalnızca temel kimlik doğrulama şemaları değil, güvenlik gereksinimlerine uygun herhangi bir şemayı da destekleyebilir.
- Bu şemaların kullanılması durumunda, **client identifier** (istemci kimliği) ile ilgili kimlik doğrulama yöntemi arasında açık bir eşleştirme yapılmalıdır.
- Yetkilendirme sunucusunun, her kimlik doğrulama yönteminin güvenlik gereksinimlerini doğru şekilde yerine getirdiğinden emin olması gerekir.

Bu bölüm, OAuth 2.0 protokolünü daha esnek ve geniş bir uygulama yelpazesinde kullanılabilir kılmak için alternatif kimlik doğrulama yöntemlerine imkan tanır.