Testing for Bypassing Authorization Schema (Yetkilendirme Şemasını Atlama Testi)

Summary (Özet)

Bu tür bir test, yetkililik şemasının, ayrılmış işlevlere ve kaynaklara erişim sağlamak için her rol veya ayrıcalık için nasıl uygulandığını doğrulamaya odaklanmaktadır.

Test cihazının değerlendirme sırasında ve uygulamanın kimlik doğrulama sonrası aşamada yürütülmesini talep eden her özel rol için aşağıdakileri doğrulamak gerekir:

- Kullanıcı doğrulanmamış olsa bile bu kaynağa erişmek mümkün mü?
- Kayıt olduktan sonra bu kaynağa erişmek mümkün mü?
- Farklı bir role veya ayrıcalığa sahip bir kullanıcıya erişilebilir olması gereken işlevlere ve kaynaklara erişmek mümkün mü?

Uygulamaya idari bir kullanıcı olarak erişmeye çalışın ve tüm idari işlevleri takip edin.

- Test cihazı yönetim dışı bir kullanıcı olarak oturum açması halinde idari işlevlere erişmek mümkün mü?
- Bu idari işlevleri farklı bir role sahip ve bu eylemin kime reddedilmesi gerektiği için bir kullanıcı olarak kullanmak mümkün mü?

Test Objectives (Test Hedefleri)

- Yatay veya dikey erişimin mümkün olup olmadığını değerlendirin.

How to Test (Nasıl Test Edilir)

- Kaynaklara erişin ve işlemleri yatay olarak yürütün.

- Kaynaklara erişin ve işlemleri dikey olarak yürütün.

Testing for Horizontal Bypassing Authorization Schema (Yatay Bypass Yetkilendirme Şeması için Test)

Her işlev, belirli rol veya başvurunun yürütmesini talep etmek için, aşağıdakileri doğrulamak gerekir:

- Aynı role veya ayrıcalıkla farklı bir kimliğe sahip bir kullanıcıya erişilebilir olması gereken kaynaklara erişmek mümkün mü?
- Farklı bir kimliğe sahip bir kullanıcıya erişiminin açık olması gereken kaynaklarla ilgili işlevleri yerine getirmek mümkün mü?

Her rol için:

1. Aynı ayrıcalıklara sahip iki kullanıcıyı kaydedin veya oluşturun.
2. İki farklı oturumu aktif (her kullanıcı için bir tane) kurmak ve saklayın.
3. Her talep için, ilgili parametreleri ve oturum tanımlayıcısını bir tokenden iki tokene değiştirin ve her jeton için yanıtları teşhis edin.
4. Yanıtlar aynıysa, aynı özel verileri içeriyorsa veya diğer kullanıcıların kaynakları veya verileri üzerinde başarılı bir şekilde çalışırsa bir uygulama savunmasız olarak kabul edilecektir.

Örneğin, öyle olduğunu varsayalım. `viewSettings` İşlev, uygulamanın her hesap menüsünün aynı rolüne sahip olup, aşağıdaki URL'yi talep ederek erişmek mümkündür: `https://www.example.com/account/viewSettings`. . Ardından, aşağıdaki HTTP isteği, çağırırken oluşturulur. `viewSettings` Fonksiyon:

```
POST /account/viewSettings HTTP/1.1
Host: www.example.com
[other HTTP headers]
Cookie: SessionID=USER_SESSION

username=example_user
```

Geçerli ve meşru cevap:

```
HTTP1.1 200 OK
[other HTTP headers]

{
  "username": "example_user",
  "email": "example@email.com",
  "address": "Example Address"
}
```

Saldırgan bu talebi aynı şekilde yürütmeye çalışabilir `username` Parametre:

```
POST /account/viewCCpincode HTTP/1.1
Host: www.example.com
[other HTTP headers]
Cookie: SessionID=ATTACKER_SESSION

username=example_user
```

Saldırganın yanıtı, verilerini içeriyorsa `example_user` Daha sonra uygulama, bir kullanıcının diğer kullanıcıların verilerini okuyabileceği veya yazabileceği yanal hareket saldırıları için savunmasızdır.

Testing for Vertical Bypassing Authorization Schema (Dikey Yetkilendirme Şemasını Atlama Testi)

Dikey bir yetki bypass, bir saldırganın kendi rollerinden daha yüksek bir rol aldığı durumuna özgüdür. Bu bypass için test, her rol için dikey yetkilendirme şemasının nasıl uygulandığını doğrulamaya odaklanmaktadır. Her işlev, sayfa, özel rol veya başvurunun yürütmesi talebi için, aşağıdakilerin mümkün olup olmadığını doğrulamak gerekir:

- Yalnızca daha yüksek bir rol kullanıcıya erişilebilir olması gereken kaynaklara erişin.
- Sadece daha yüksek veya belirli bir rol kimliğine sahip bir kullanıcı tarafından faaliyete geçmesi gereken kaynaklar üzerinde işlevler çalıştırın.

Her rol için:

1. Bir kullanıcıya kaydolun.
2. İki farklı role dayanan iki farklı oturum oluşturun ve sürdürün.
3. Her talep için, oturum tanımlayıcısını orijinalinden başka bir rolün oturum tanımlayıcısına değiştirin ve her biri için yanıtları değerlendirin.
4. Bir uygulama, daha zayıf ayrıcalıklı oturum aynı verileri içeriyorsa veya daha yüksek ayrıcalıklı işlevlerde başarılı işlemleri gösterirse savunmasız olarak kabul edilecektir.

Banking Site Roles Scenario (Bankacılık Sitesi Rolleri Senaryosu)

Aşağıdaki tablo, bir bankacılık sitesindeki sistem rollerini göstermektedir. Her rol, etkinlik menüsü işlevselliği için belirli izinlere bağlanır:

ROL	İHALE	EK IZIN
Yönetici	Tam Kontrol	Silin
Yönetici	Değiştirin, Ekleyin, Oku	Ekleyin
Personel	Oku, Değiştir	Modify
Müşteri	Sadece Oku	

Uygulama: Aşağıdaki durumlarda savunmasız kabul edilecektir:

1. Müşteri yönetici, yönetici veya personel fonksiyonlarını işletmektedir;
2. Personel kullanıcısı yönetici veya yönetici işlevlerini çalıştırabilir;
3. Yönetici yönetici işlevlerini çalıştırabilir.

San bakalım `deleteEvent` İşlev, uygulamanın yönetici hesap menüsünün bir parçasıdır ve aşağıdaki URL'yi talep ederek erişmek mümkündür: `https://www.example.com/account/deleteEvent`. . Ardından, aşağıdaki HTTP isteği, çağırırken oluşturulur. `deleteEvent` Fonksiyon:

```
POST /account/deleteEvent HTTP/1.1
Host: www.example.com
[other HTTP headers]
Cookie: SessionID=ADMINISTRATOR_USER_SESSION
```

```
EventID=1000001
```

Geçerli yanıt:

```
HTTP/1.1 200 OK
[other HTTP headers]

{"message": "Event was deleted"}
```

Saldırgan aynı isteği yerine getirmeye çalışabilir:

```
POST /account/deleteEvent HTTP/1.1
Host: www.example.com
[other HTTP headers]
Cookie: SessionID=CUSTOMER_USER_SESSION

EventID=1000002
```

Saldırganın isteğinin yanıtı aynı verileri içeriyorsa `{"message": "Event was deleted"}` Uygulama savunmasızdır.

Administrator Page Access (Yönetici Sayfası Erişimi)

Yönetici menüsünün yönetici hesabının bir parçası olduğunu varsayalım.

Uygulama, yönetici dışındaki herhangi bir rolün yönetici menüsüne erişebilmesi durumunda savunmasız olarak kabul edilecektir. Bazen, geliştiriciler sadece GUI seviyesinde yetkilendirme doğrulaması gerçekleştirir ve işlevleri yetkilendirme doğrulaması olmadan bırakırlar, böylece potansiyel olarak bir güvenlik açığı ile sonuçlanır.

Testing for Access to Administrative Functions (İdari İşlevlere Erişim Testi)

Örneğin, öyle olduğunu varsayalım. `addUser` İşlev, uygulamanın idari menüsünün bir parçasıdır ve aşağıdaki URL'yi talep ederek erişmek mümkündür.
`https://www.example.com/admin/addUser` . .

Ardından, aşağıdaki HTTP isteği, çağırırken oluşturulur. `addUser` Fonksiyon:

```
POST /admin/addUser HTTP/1.1
Host: www.example.com
[...]

userID=fakeuser&role=3&group=grp001
```

Daha fazla soru veya değerlendirme aşağıdaki yönde olacaktır:

- İdari olmayan bir kullanıcı bu talebi yürütmeye çalışırsa ne olur?
- Kullanıcı oluşturulacak mı?
- Eğer öyleyse, yeni kullanıcı ayrıcalıklarını kullanabilir mi?

Testing for Access to Resources Assigned to a Different Role (Farklı Bir Role Atanan Kaynışlara Erişim Testi)

Çeşitli uygulamalar kullanıcı rollerine göre kaynak kontrolleri kurar. Bir S3 kovasına bir kariyer formuna yüklenen bir örnek özgeçmiş veya CV'ler (mürrik ve vitae) alalım.

Normal bir kullanıcı olarak, bu dosyaların konumuna ulaşmayı deneyin. Onları geri alabilir, değiştirebilir veya silebilirseniz, uygulama savunmasızdır.

Testing for Special Request Header Handling (Özel İstek Başlık Taşıması Testi)

Bazı uygulamalar standart olmayan başlıkları destekler. `X-Original-URL` ya da `X-Rewrite-URL` Başlık değerinde belirtilen ile isteklerde hedef URL'yi geçersiz kılmaya izin vermek için.

Bu davranış, uygulamanın istek URL'sine dayalı erişim kontrolü kısıtlaması uygulayan bir bileşenin arkasında olduğu bir durumda kullanılabilir.

İstek URL'sine dayalı erişim kontrol kısıtlaması türü, örneğin internet'ten maruz kalan bir yönetim konsoluna erişimi engelleme olabilir. `/console` ya da `/admin` . .

Başlık için desteği tespit etmek `X-Original-URL` ya da `X-Rewrite-URL` Aşağıdaki adımlar uygulanabilir.

1. Send a Normal Request without Any X-Original-Url or X-Rewrite-Url Header (1. X-Original-Url veya X-Rewrite-Url Başlıklı

Olmadan Normal Bir Talep Gönderin)

```
GET / HTTP/1.1
Host: www.example.com
[...]
```

2. Send a Request with an X-Original-Url Header Pointing to a Non-Existing Resource (2. Mevcut Olmayan Bir Kaynağın İşareti Olan X-Ornal Bir Başlıkla Bir Talep Gönderin)

```
GET / HTTP/1.1
Host: www.example.com
X-Original-URL: /donotexist1
[...]
```

3. Send a Request with an X-Rewrite-Url Header Pointing to a Non-Existing Resource (3. Mevcut Olmayan Bir Kayr Kaynağına Işaret Eden Bir X-Rewrite-Url Başlıklı Bir İstek Gönderin)

```
GET / HTTP/1.1
Host: www.example.com
X-Rewrite-URL: /donotexist2
[...]
```

Her iki talebin yanıtı, kaynağın bulunmadığı belirteçleri içeriyorsa, bu uygulamanın özel istek başlıklarını desteklediğini gösterir. Bu belirteçler, yanıt gövdesinde HTTP yanıt durumu kodunu veya yanıt gövdesinde “bulut bulunmamış kaynak” mesajını içerebilir.

Bir kez başlık için destek `X-Original-URL` ya da `X-Rewrite-URL` Onaylandıktan sonra, erişim kontrol kısıtlamasına karşı bypass geçiciliği, beklenen isteği uygulamaya göndererek, ancak ana istek URL'si olarak ön uç bileşeni tarafından "izin verilen" bir URL'yi belirterek ve gerçek hedef
URL'yi belirten bir URL'yi iletilebilir.
`X-Original-URL` ya da `X-Rewrite-URL` Desteklenene

bağlı olarak başlık. Her ikisi de desteklenirse, bypassın hangi başlığın etkili olduğunu doğrulamak için birbiri ardına deneyin.

4. Other Headers to Consider (4. Dikkate alınması gereken diğer başlıklar)

Genellikle yönetici panelleri veya idari ilgili işlevsellik parçaları yalnızca yerel ağlarda müşteriler tarafından erişilebilir, bu nedenle çeşitli proxy'leri kötüye kullanmak veya ilgili HTTP başlıklarını erişim sağlamak için iletmek mümkün olabilir. Test edilmesi gereken bazı başlıklar ve değerler şunlardır:

- Başlıklar:
  - `X-Forwarded-For`
  - `X-Forward-For`
  - `X-Remote-IP`
  - `X-Originating-IP`
  - `X-Remote-Addr`
  - `X-Client-IP`
- Değerler
  - `127.0.0.1` (ya da içindeki herhangi bir şey `127.0.0.0/8` ya da `::1/128` Adres alanları)
  - `localhost`
  - Herhangi bir şey RFC1918 Adres:
    - `10.0.0.0/8`
    - `172.16.0.0/12`
    - `192.168.0.0/16`
  - Yerel adresleri bağlayın: `169.254.0.0/16`

Not: Adres veya ana bilgisayar ile birlikte bir bağlantı elemanı dahil olmak, web uygulaması güvenlik duvarları vb. Gibi kenar korumalarını atlamaya yardımcı olabilir. Örneğin: `127.0.0.4:80` , `127.0.0.4:443` , `127.0.0.4:43982`

Remediation (Düzeltme)

İzinsiz erişimin gerçekleşmemesini sağlamak için kullanıcılar, roller ve kaynaklar üzerinde en az ayrıcalık ilkelerini kullanın.

Tools (Araçlar)

- OWASP Zed Attack Proxy (ZAP)
  - ZAP add-on: Access Control Testing
- Port Swigger Burp Suite
  - Burp extension: AuthMatrix
  - Burp extension: Autorize

References (Referanslar)

OWASP Application Security Verification Standard 4.0.1, v4.0.1-1, v4.0.1-4, v4.0.1-9, v4.0.1-16