# Testing for Account Enumeration and Guessable User Account (Hesap Numaralandırma ve Tahmin Edilebilir Kullanıcı Hesabı Testi)

## Summary (Özet)

Bu testin kapsamı, uygulamanın kimlik doğrulama mekanizmasıyla etkileşimde bulunarak bir dizi geçerli kullanıcı adı toplanmasının mümkün olup olmadığını doğrulamaktır. Bu test, geçerli bir kullanıcı adı verildiğinde, ilgili şifreyi bulmak mümkünse test cihazının doğruladığı kaba kuvvet testi için yararlı olacaktır.

Genellikle, web uygulamaları, yanlış yapılandırmanın bir sonucu olarak veya bir tasarım kararı olarak sistemde bir kullanıcı adının ne zaman var olduğunu ortaya çıkarır. Örneğin, bazen yanlış kimlik bilgilerini gönderirken, kullanıcı adının sistemde mevcut olduğunu veya sağlanan şifrenin yanlış olduğunu belirten bir mesaj alırız. Elde edilen bilgiler, bir saldırgan tarafından sistemdeki kullanıcıların bir listesini kazanmak için kullanılabilir. Bu bilgiler, örneğin kaba bir güç veya varsayılan kullanıcı adı ve şifre saldırısı yoluyla web uygulamasına saldırmak için kullanılabilir.

Test cihazı, belirli taleplerin uygulamanın farklı şekillerde cevaplanmasına neden olup olmadığını anlamak için uygulamanın kimlik doğrulama mekanizmasıyla etkileşime girmelidir. Bu sorun, kullanıcı geçerli bir kullanıcı adı sağladığında web uygulamasından veya web sunucusundan yayınlanan bilgiler geçersiz bir kullanıcıyı kullandıklarından farklı olduğu için vardır.

Bazı durumlarda, geçersiz bir kullanıcı adı veya geçersiz bir şifre kullanıldığı için sağlanan kimlik bilgilerinin yanlış olup olmadığını ortaya koyan bir mesaj alınır. Bazen, testçiler bir kullanıcı adı ve boş bir şifre göndererek mevcut kullanıcıları sayabilir.

## Test Objectives (Test Hedefleri)

- Kullanıcı tanımlama ile ilgili süreçleri gözden geçirin *(örneğin* kayıt, giriş, vb.).
- Yanıt analizi yoluyla mümkün olan yerlerde kullanıcıları numaralandırın.

## How to Test (Nasıl Test Edilir)

Kara kutu testinde test cihazı, belirli uygulama, kullanıcı adı, uygulama mantığı, sayfadaki kayıtlardaki hata mesajları veya şifre kurtarma tesisleri hakkında hiçbir şey bilmiyor. Uygulama savunmasızsa, test cihazı doğrudan veya dolaylı olarak kullanıcıları numaralandırmak için yararlı bazı bilgileri ortaya koyan bir yanıt mesajı alır.

### HTTP Response Message (HTTP Yanıt Mesajı)

#### Testing for Valid Credentials (Geçerli Kimlik Bilgilerinin Test Edilmesi)

Geçerli bir kullanıcı kimliği ve geçerli bir şifre gönderdiğinizde sunucu yanıtını kaydedin.

> Bir web proxy kullanarak, bu başarılı kimlik doğrulamadan alınan bilgileri (HTTP 200 Yanıt, yanıtın uzunluğu) fark edin.

#### Testing for Valid User with Wrong Password (Yanlış Şifre ile Geçerli Kullanıcı İçin Test)

Şimdi, test cihazı geçerli bir kullanıcı kimliği ve yanlış bir şifre eklemeye çalışmalı ve uygulama tarafından oluşturulan hata mesajını kaydetmelidir.

Tarayıcı aşağıdaki mesaja benzer bir mesaj görüntülemelidir:


*Şekil 4.3.4-1: Kimlik Doğrulama Başarısız Oldu*

Kullanıcının varlığını aşağıdaki gibi ortaya koyan herhangi bir mesajın aksine:

```
Login for User foo: invalid password
```

Bir web proxy kullanarak, bu başarısız kimlik doğrulama girişiminden alınan bilgileri (HTTP 200 Yanıt, yanıtın uzunluğu) bildirin.

**Testing for a Nonexistent Username** (**Var Olmayan Bir Kullanıcı Adı Testi**)

Şimdi, test cihazı geçersiz bir kullanıcı kimliği ve yanlış bir şifre eklemeye çalışmalı ve sunucu cevabını kaydetmeli (sözcü kullanıcı adının uygulamada geçerli olmadığından emin olmalıdır). Hata mesajını ve sunucu cevabını kaydedin.

Test cihazı mevcut olmayan bir kullanıcı kimliğine girerse, aşağıdakilere benzer bir mesaj alabilirler:


*Şekil 4.3.4-3: Bu Kullanıcı Aktif Değil*

Ya da aşağıdaki gibi bir mesaj:

Genel olarak uygulama, farklı yanlış isteklere aynı hata mesajı ve uzunluğu ile yanıt vermelidir. Yanıtlar aynı değilse, test cihazı araştırmalı ve iki yanıt arasında bir fark yaratan anahtarı bulmalıdır. Örneğin:

1. İstemci isteği: Geçerli kullanıcı/yanlış şifre
2. Sunucu yanıtı: Şifre doğru değil
3. İstemci isteği: Yanlış kullanıcı/yanlış şifre
4. Sunucu yanıtı: Kullanıcı tanınmadı

Yukarıdaki yanıtlar, müşterinin ilk istek için geçerli bir kullanıcıya sahip olduklarını anlamasını sağlar. Böylece, bir dizi olası kullanıcı kimliği talep eden ve cevabı gözlemleyen uygulama ile etkileşime girebilirler.

İkinci sunucu yanıtına bakıldığında, test cihazı geçerli bir kullanıcı adı tutmadıkları gibi anlar. Böylece aynı şekilde etkileşime girebilir ve sunucu cevaplarına bakan geçerli bir kullanıcı kimliği listesi oluşturabilirler.

## Other Ways to Enumerate Users (Kullanıcıları numaralandırmanın diğer yolları)

Testçiler kullanıcıları aşağıdaki gibi çeşitli şekillerde numaralandırabilir:

## Analyzing the Error Code Received on Login Pages (Giriş Sayfalarında Alınan Hata Kodunu Analiz Etmek)

Bazı web uygulamaları, analiz edebileceğimiz belirli bir hata kodu veya mesaj yayınlar.

## Analyzing URLs and URLs Re-directions (URL'leri ve URL'lerin Yeniden Yönlendirmelerini Analiz Etmek)

Örneğin:

- `http://www.foo.com/err.jsp?User=baduser&Error=0`
- `http://www.foo.com/err.jsp?User=gooduser&Error=2`

Yukarıda görüldüğü gibi, bir test cihazı web uygulamasına bir kullanıcı kimliği ve şifre sağladığında, URL'de bir hatanın meydana geldiğinin bir mesaj göstergesi görürler. İlk durumda kötü bir kullanıcı
kimliği ve kötü şifre sağladılar. İkincisi, iyi bir kullanıcı kimliği ve kötü bir şifre, böylece geçerli bir kullanıcı kimliğini tanımlayabilirler.

## (URI Probing)

Bazen bir web sunucusu, mevcut bir dizin için bir istek alırsa veya almazsa farklı tepki verir. Örneğin bazı portallarda her kullanıcı bir dizinle ilişkilendirilir. Testçiler mevcut bir dizine erişmeye
çalışırsa, bir web sunucusu hatası alabilirler.

Web sunucularından alınan yaygın hatalardan bazıları şunlardır:

- 403 Yasak hata kodu
- 404 Hata kodu bulunamadı

Örnek:

- `http://www.foo.com/account1` Web sunucusundan alıyoruz: 403 Yasak
- `http://www.foo.com/account2` Web sunucusundan alıyoruz: 404 dosyası Bulunmadı

İlk durumda kullanıcı var, ancak test cihazı web sayfasını görüntüleyemiyor, ikinci durumda kullanıcı “aclu2” mevcut değil. Bu bilgiyi toplayarak test cihazları kullanıcıları sayabilir.

## Analyzing Web Page Titles (Web Sayfası Başlıklarını Analiz Etmek)

Testçiler, sorunların kullanıcı adı veya şifre ile olup olmadığını ortaya koyan belirli bir hata kodu veya mesaj alabilecekleri web sayfasının başlığında yararlı bilgiler alabilirler.

Örneğin, bir kullanıcı bir uygulamaya kimlik doğrulayamaz ve başlığı benzer olan bir web sayfası alırsa:

- `Invalid user`
- `Invalid authentication`

## Analyzing a Message Received from a Recovery Facility (Bir Kurtarma Tesisinden Alınan Bir Mesajın Analizi)

Bir kurtarma tesisi (yani unutulmuş bir şifre işlevi) kullandığımızda, savunmasız bir uygulama, bir kullanıcı adının var olup olmadığını ortaya koyan bir mesajı döndürebilir.

Örneğin, aşağıdakilere benzer mesajlar:

- `Invalid username: email address is not valid or the specified user was not found.`
- `Valid username: Your password has been successfully sent to the email address you registered with.`

## Friendly 404 Error Message (Dostu 404 Hata Mesajı)

Dizin içinde bulunmayan bir kullanıcı talep ettiğimizde her zaman 404 hata kodu alamaz. Bunun yerine, bir görüntü ile “200 tamam” alabiliriz,  bu durumda belirli resmi aldığımızda kullanıcının var olmadığını varsayabiliriz. Bu mantık diğer web sunucusu yanıtına uygulanabilir; hile, web sunucusu ve web uygulama mesajlarının iyi bir analizidir.

## Analyzing Response Times (Yanıt Zamanlarını Analiz Etmek)

Yanıtların içeriğine bakmanın yanı sıra, yanıtın aldığı süre de göz önünde bulundurulmalıdır. Özellikle talebin harici bir hizmetle etkileşime neden olduğu

durumlarda (unutulmuş bir şifre e-postası göndermek gibi), bu, istenen kullanıcının geçerli olup olmadığını belirlemek için kullanılabilecek yanıta birkaç yüz milisaniye ekleyebilir.

## Guessing Users (Tahmini Kullanıcılar)

Bazı durumlarda kullanıcı kimlikleri, yönetici veya şirketin belirli politikaları ile oluşturulur. Örneğin, sıralı sırayla oluşturulan bir kullanıcı kimliğine sahip bir kullanıcı görüntüleyebiliriz:

```
CN000100
CN000101
...
```

Bazen kullanıcı adları bir REALM takma adı ve ardından sıralı sayılarla oluşturulur:

- R1001 - REALM1 için kullanıcı 001
- R2001 - REALM2 için kullanıcı 001

Yukarıdaki örnekte, kullanıcı kimliklerini oluşturan basit kabuk komut dosyaları oluşturabilir ve geçerli kullanıcı kimliklerini ayırt etmek için bir web sorgusunu otomatikleştirmek için web gibi bir araçla bir talep gönderebiliriz. Bir senaryo oluşturmak için Perl ve bukle de kullanabiliriz.

Diğer olasılıklar şunlardır: - kredi kartı numaralarıyla ilişkili kullanıcı kimlikleri veya bir desene sahip genel sayılar. - gerçek isimlerle ilişkili kullanıcı kimlikleri, örneğin Freddie Mercury'nin birkullanıcı kimliğine sahipse, o zaman Roger Taylor'ın “rtaylor” kullanıcı kimliğine sahip olduğunu tahmin edebilirsiniz.

Yine, bir LDAP sorgusundan veya örneğin belirli bir etki alanından Google bilgi toplamasından alınan bilgilerden bir kullanıcı adını tahmin edebiliriz. Google, belirli sorgular yoluyla veya basit bir kabuk komut dosyası veya araç aracılığıyla etki alanı kullanıcılarının bulunmasına yardımcı olabilir.

> Kullanıcı hesaplarını numaralandırarak, önceden tanımlanmış sayıda başarısız sondadan sonra hesapları kilitleme riskiniz (uygulama politikasına göre). Ayrıca, bazen, IP adresiniz

> uygulama güvenlik duvarı veya İtme Önleme Sistemi ile ilgili dinamik kurallarla yasaklanabilir.

### Gray-Box Testing (Gri-Kutu Testi)

**Testing for Authentication Error Messages** (Kimlik Doğrulama Hatası Mesajları için Test)

Uygulamanın, başarısız bir kimlik doğrulama üreten her istemci isteği için aynı şekilde yanıt verdiğini doğrulayın. Bu sorun için kara kutu testi ve gri kutu testi, web uygulamasından alınan mesajların veya hata kodlarının analizine dayanarak aynı konsepte sahiptir.

> Uygulama, her başarısız kimlik doğrulama girişimi için aynı şekilde cevap vermelidir.
>
> Örneğin: *Gönderilen kimlik bilgileri geçerli değildir*

## Remediation (Düzeltme)

Uygulamanın, işlem sırasında giriş yapılan geçersiz hesap adı, şifre veya diğer kullanıcı kimlik bilgilerine yanıt olarak tutarlı jenerik hata mesajları döndürün.

Sistemi üretime geçirmeden önce (veya güvenilmez bir ağa maruz bırakmadan) önce varsayılan sistem hesaplarının silindiğinden emin olun.

## Tools (Araçlar)

- OWASP Zed Saldırı Proxy (ZAP)
- Curl
- PERL'

## Referance (Referanslar)

- Marco Mella, Sun Java Erişim ve Kimlik Yöneticisi Kullanıcıları Numaralandırma
- Kullanıcı adı Sayım Güvenlik Açıkları