Testing for Bypassing Authentication Schema (Kimlik Doğrulama Şemasını Bypass için Test)

Summary (Özet)

Bilgisayar güvenliğinde kimlik doğrulama, bir iletişimin göndereninin dijital kimliğini doğrulamaya çalışma sürecidir. Böyle bir sürecin yaygın bir örneği, süreçteki günlüktür. Kimlik doğrulama şemasının test edilmesi, kimlik doğrulama işleminin nasıl çalıştığını anlamak ve kimlik doğrulama mekanizmasını aşmak için bu bilgileri kullanmak anlamına gelir.

Çoğu uygulama, özel bilgilere erişmek veya görevleri yürütmek için kimlik doğrulama gerektirirken, her kimlik doğrulama yöntemi yeterli güvenliği sağlayamaz. Güvenlik tehditlerinin ihmali, cehalet veya basit yetersizliği genellikle sayfadaki kütüğü atlayarak ve yalnızca kimlik doğrulama yapıldıktan sonra erişilmesi gereken bir dahili sayfayı doğrudan arayarak atılabilen kimlik doğrulama şemalarına neden olur.

Buna ek olarak, istekleri kurcalayarak ve uygulamayı kullanıcının zaten doğrulandığını düşünerek kandırarak kimlik doğrulama önlemlerini atlamak genellikle mümkündür. Bu, verilen URL parametresini değiştirerek, formu manipüle ederek veya sahte oturumlarla
gerçekleştirilebilir.

Kimlik doğrulama şeması ile ilgili sorunlar, tasarım, geliştirme ve dağıtım aşamaları gibi yazılım geliştirme yaşam döngüsünün (SDLC) farklı aşamalarında bulunabilir:

- Tasarım aşamasında hatalar, korunması gereken uygulama bölümlerinin yanlış bir tanımını, kimlik bilgilerinin iletimini güvence altına almak için güçlü şifreleme protokolleri uygulamama seçeneğini ve daha fazlasını içerebilir.
- Geliştirme aşamasında hatalar, giriş doğrulama işlevselliğinin yanlış uygulanmasını veya belirli dil için en iyi uygulamaları takip etmemesini

içerebilir.

- Uygulama dağıtım aşamasında, gerekli teknik becerilerde eksiklik veya iyi dokümantasyon eksikliği nedeniyle uygulama kurulumu (kurulaş ve yapılandırma faaliyetleri) sırasında sorunlar olabilir.

Test Objectives (Test Hedefleri)

- Kimlik doğrulamanın bunu gerektiren tüm hizmetlerde uygulanmasını sağlayın.

How to Test (Nasıl Test Edilir)

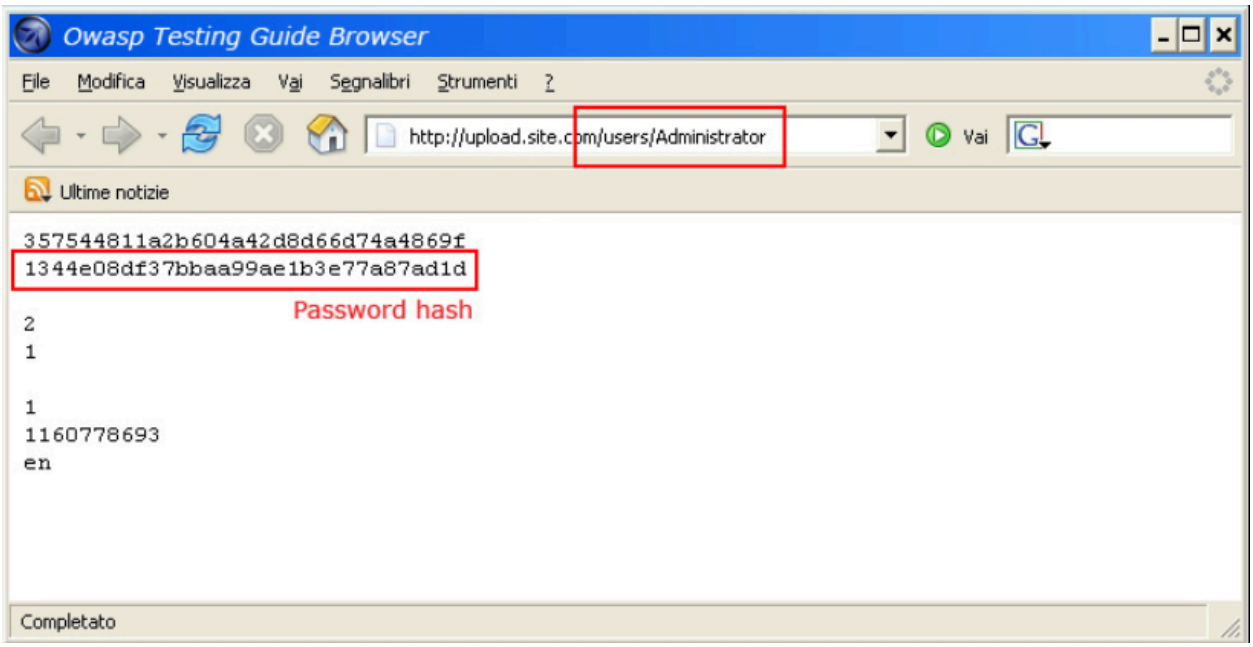
Black-Box Testing (Siyah-Kutu Testi)

Bir web uygulaması tarafından kullanılan kimlik doğrulama şemasını atlamak için çeşitli yöntemler vardır:

- Doğrudan sayfa isteği (zorla tarama)
- Parametre modifikasyonu
- Oturum kimliği tahmini
- SQL enjeksiyonu

Direct Page Request (Doğrudan Sayfa İsteği)

Bir web uygulaması yalnızca sayfadaki girişte erişim kontrolü uygularsa, kimlik doğrulama şeması atlanabilir. Örneğin, bir kullanıcı doğrudan zorunlu tarama yoluyla farklı bir sayfa isterse, bu sayfa erişim izni vermeden önce kullanıcının kimlik bilgilerini kontrol etmeyebilir. Bu yöntemi kullanarak test etmek için tarayıcınızdaki adres çubuğu aracılığıyla korunan bir sayfaya doğrudan erişme girişimi.



Şekil 4.4.4-1: Korumalı Sayfaya Doğrudan Talep

Parameter Modification (Parametre Değişikliği)

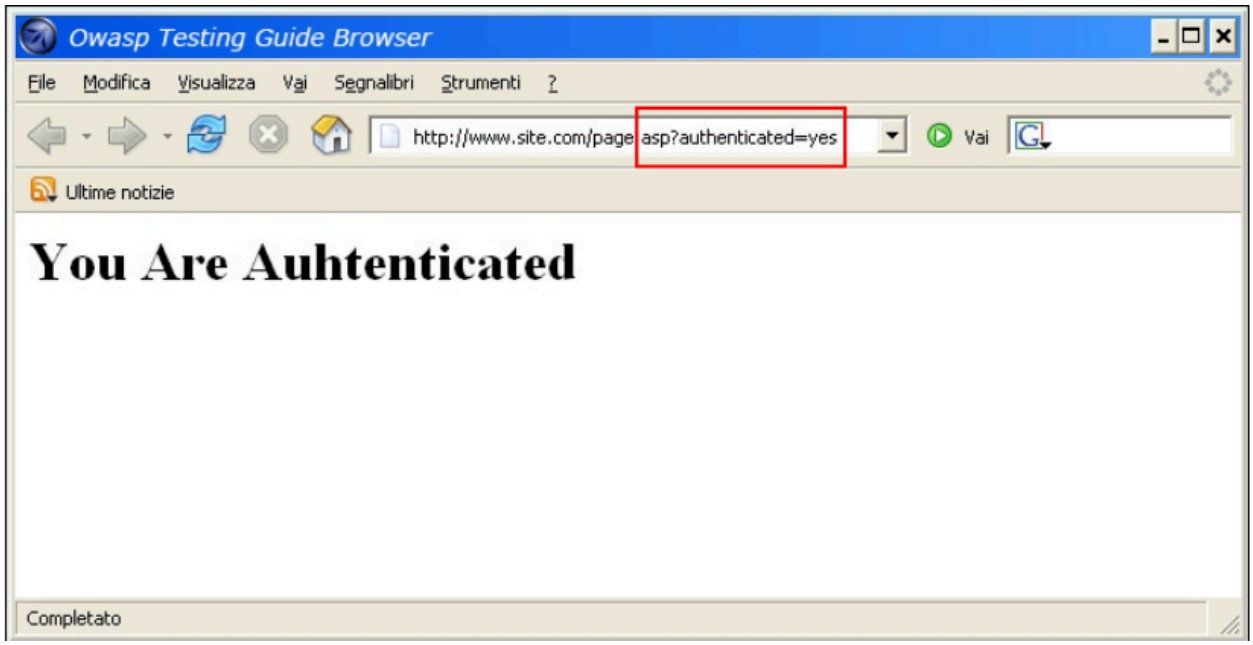
Kimlik doğrulama tasarımıyla ilgili bir diğer sorun, uygulamanın sabit bir değer parametreleri temelinde başarılı bir günlük doğrulamasıdır. Bir kullanıcı, geçerli kimlik bilgileri sağlamadan korunan alanlara erişmek için bu parametreleri değiştirebilir. Aşağıdaki örnekte, "otantikleşmiş" parametre, kullanıcının erişimini sağlayan "evet" değerine dönüştürülür. Bu örnekte, parametre URL'dedir, ancak bir proxy, parametreyi değiştirmek için de kullanılabilir, özellikle de bir POST talebinde form elemanları olarak parametreler gönderildiğinde veya parametreler bir çerezde depolandığında kullanılabilir.

```
http://www.site.com/page.asp?authenticated=no

raven@blackbox /home $nc www.site.com 80
GET /page.asp?authenticated=yes HTTP/1.0

HTTP/1.1 200 OK
Date: Sat, 11 Nov 2006 10:22:44 GMT
Server: Apache
Connection: close
Content-Type: text/html; charset=iso-8859-1
```

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
</HEAD><BODY>
<H1>You Are Authenticated</H1>
</BODY></HTML>
```

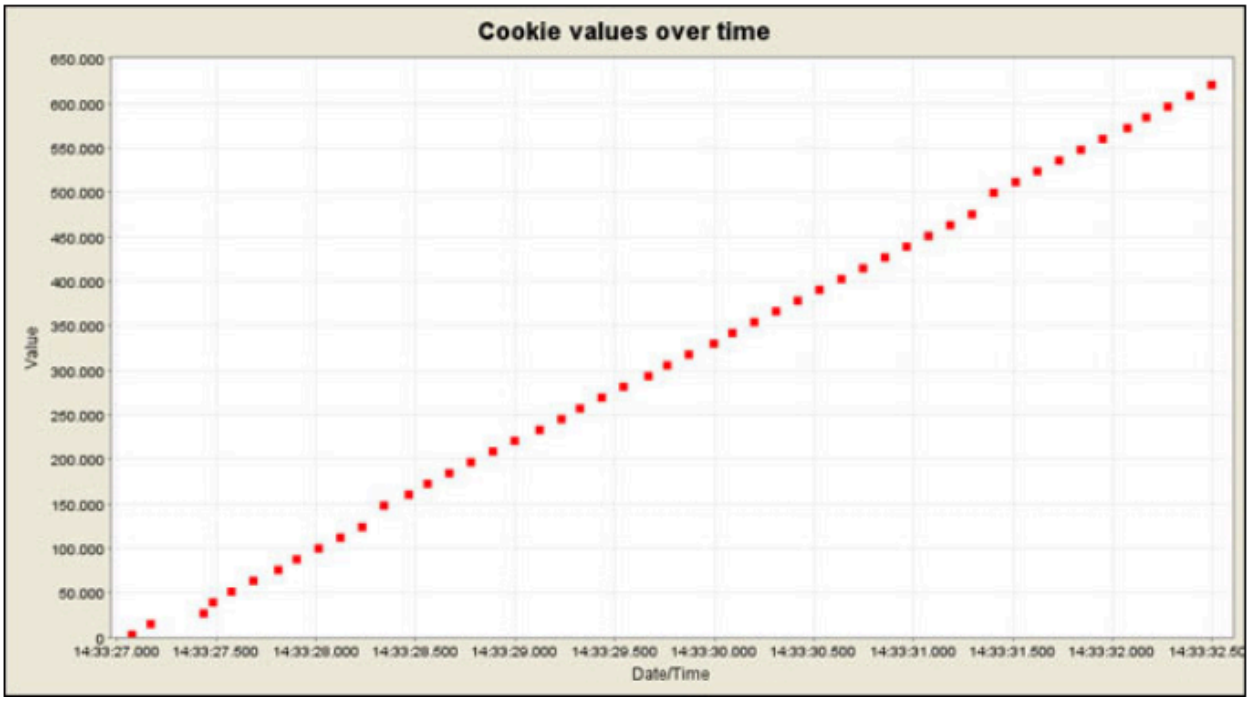


Şekil 4.4.4-2: Parametre Değiştirilmiş İstek

Session ID Prediction (Oturum ID Tahmini)

Birçok web uygulaması, oturum tanımlayıcıları (oturum Kimlikleri) kullanarak kimlik doğrulamayı yönetir. Bu nedenle, oturum Kimliği üretimi öngörülebilir ise, kötü niyetli bir kullanıcı geçerli bir oturum kimliği bulabilir ve uygulamaya yetkisiz erişim sağlayabilir ve daha önce doğrulanmış bir kullanıcıyı taklit edebilir.

Aşağıdaki rakamda, çerezlerin içindeki değerler doğrusal olarak artar, bu nedenle bir saldırganın geçerli bir oturum kimliğini tahmin etmesi kolay olabilir.



Şekil 4.4.4-3: Zaman Içinde Çerez Değerleri

Aşağıdaki rakamda, çerezlerin içindeki değerler sadece kısmen değişir, bu nedenle kaba bir kuvvet saldırısını aşağıda gösterilen tanımlanmış alanlara kısıtlamak mümkündür.

Session Identifier :	127.0.0.1/WebGoat WEAKID
Date	Value
2006/11/11 14:33:27	12430-1163252007028
2006/11/11 14:33:27	12431-1163252007138
2006/11/11 14:33:27	12432-1163252007247
2006/11/11 14:33:27	12433-1163252007435
2006/11/11 14:33:27	12434-1163252007544
2006/11/11 14:33:27	12435-1163252007653
2006/11/11 14:33:27	12436-1163252007763
2006/11/11 14:33:27	12437-1163252007872
2006/11/11 14:33:28	12438-1163252007982
2006/11/11 14:33:28	12439-1163252008091
2006/11/11 14:33:28	12440-1163252008200
2006/11/11 14:33:28	12442-1163252008310
2006/11/11 14:33:28	12443-1163252008419
2006/11/11 14:33:28	12444-1163252008528
2006/11/11 14:33:28	12445-1163252008638
2006/11/11 14:33:28	12446-1163252008747
2006/11/11 14:33:28	12447-1163252008857
2006/11/11 14:33:28	12448-1163252008966
2006/11/11 14:33:29	12449-1163252009075

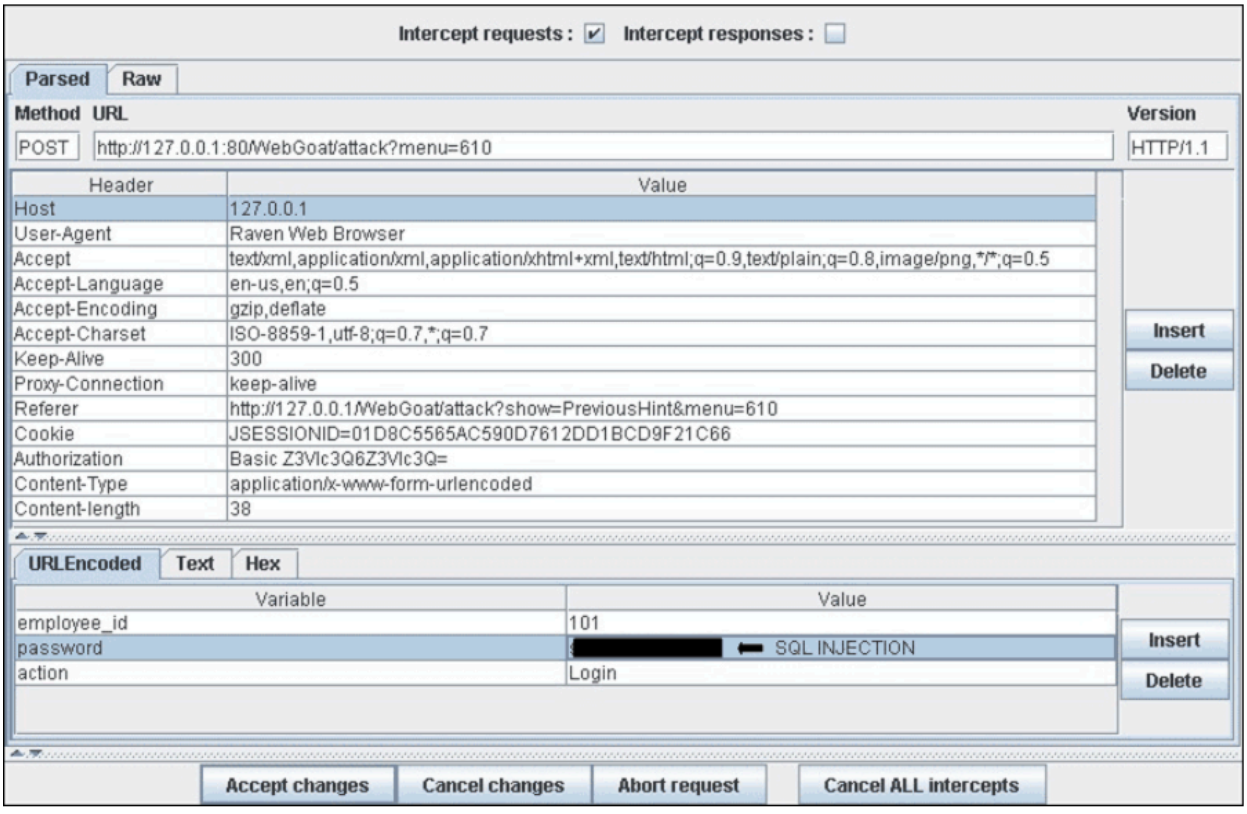
Şekil 4.4.4-4: Kısmen Değiştirilmiş Çerez Değerleri

SQL Injection (HTML Form Authentication) (SQL Injection (HTML Form Kimlik Doğrulama))

SQL Injection yaygın olarak bilinen bir saldırı tekniğidir. Bu bölüm, bu kılavuzun bu kılavuzun bu bölümün kapsamının ötesinde enjeksiyon tekniklerini açıklayan birkaç bölüm olduğu için bu tekniği ayrıntılı olarak tanımlamayacaktır.

Şekil 4.4.4-5: SQL Enjeksiyon

Aşağıdaki rakam, basit bir SQL enjeksiyon saldırısı ile kimlik doğrulama formunu atlamanın bazen mümkün olduğunu göstermektedir.



Şekil 4.4.4-6: Basit SQL Enjeksiyon Saldırısı

Gray-Box Testing (Gri-Kutu Testi)

Bir saldırgan, daha önce keşfedilen bir güvenlik açığından (örneğin, dizin geçidi) veya bir web deposundan (Açık Kaynak Uygulamaları) yararlanarak uygulama kaynak kodunu geri alabilsediyse, kimlik doğrulama sürecinin uygulanmasına karşı rafine saldırılar gerçekleştirmek mümkün
olabilir.

Aşağıdaki örnekte (PHPBB 2.0.13 - Kimlik Doğrulama Bypass Güvenlik Açığı), 5. satırda seri dışı () işlevi, sağlanan bir çerezi ortadan kaldırır ve değerleri $ row dizisinin içinde ayarlar. 10. satırda,
kullanıcının arka uç veritabanında depolanan MD5 şifre hashı ile karşılaştırılır.

```
if (isset($HTTP_COOKIE_VARS[$cookiename . '_sid']) {
    $sessiondata = isset($HTTP_COOKIE_VARS[$cookiename . '_data']) ? unserial
    $sessionmethod = SESSION_METHOD_COOKIE;
}
if(md5($password) == $row['user_password'] && $row['user_active']) {
    $autologin = (isset($HTTP_POST_VARS['autologin'])) ? TRUE : 0;
}
```

PHP'de, bir sicim değeri ile bir boolean değeri arasında bir karşılaştırma (1 ve `TRUE`) her zaman öyledir `TRUE`, bu yüzden aşağıdaki ipi tedarik ederek (önemli kısımdır `b:1`) için `unserialize()` İşlev, kimlik doğrulama kontrolünü atlamak mümkündür:

`a:2:{s:11:"autologinid";b:1;s:6:"userid";s:1:"2";}`

Tools (Araçlar)

- WebGoat
- OWASP Zed Attack Proxy (ZAP)

References (Referanslar)

Whitepapers ()

- Mark Roxberry: “PHPBB 2.0.13 vulnerability”
- David Endler: “Session ID Brute Force Exploitation and Prediction”