Oturum Öncesi Tanımlı Oturum Kimliği Çerezi Açığı

Oturum öncesi tanımlı oturum kimliği çerezi açığı, az bilinen ve dünya genelinde çok sayıda sistemde yaygın olarak bulunan bir açıktır. Bunun başlıca nedeni, açığın ve dolaylı etkilerinin bilinirliğinin düşük olmasıdır. Türkiye'de bu açıktan ilk defa 02.04.2012 tarihinde TÜBİTAK Ulusal Bilgi Güvenliği Kapısı'nda yayınlanan "**Oturum Açma Sistemleri Tasarımında Güvenlik** [1]**"** başlıklı makalede bahsedilmiştir.

Web uygulamaları kimlik doğrulaması adımından başarı ile geçen kullanıcılar oturumlarını saklamak için Session ID/Cookie (çerez) bilgisini kullanırlar. Yaratılan bir çerezin içeriğinde kullanıcıya ait oturum kimliği de bulunur. Bu oturum kimlikleri kullanıcı başarılı bir kimlik doğrulama adımını geçtikten sonra özel olarak hazırlanmış tahmin edilmesi zor Session ID'lerinden oluşturulurlar ve hem sunucu tarafında hem de kullanıcı tarafında saklanırlar.

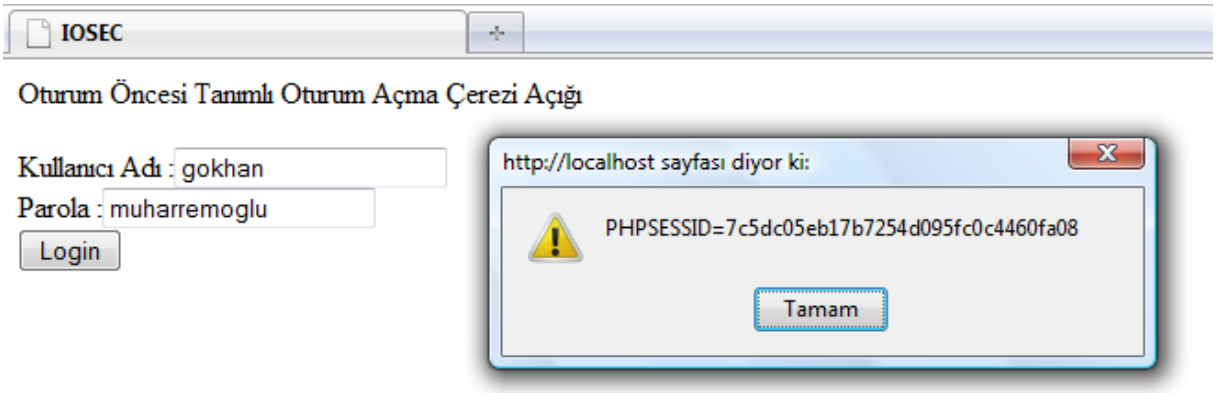
Bir çerez içeriğinde kimlik doğrulama (login/authentication) adımından önce bulunan Session ID, yani oturum kimliği, başarılı bir oturum açma işlemi sonrasında da kullanılmaya devam ederse, oturum açılmadan önceki çerezi çalan bir saldırgan, kullanıcı oturum açtıktan sonra da bu çerezi kullanarak sisteme yetkisiz erişim sağlayabilir. Kullanıcı giriş sayfalarında kimlik doğrulaması adımı geçilmeden yaratılan çerez bilgisinin kimlik doğrulaması adımından sonra da kullanılmaya devam ettiği durumlarda bu açığın varlığından söz edilebilir.

Açığın Oturum Çalma (Session Hijacking) saldırılarından biri olan Oturum Dinleme (Session Sniffing attack) saldırısındaki kullanım şekli, bütün saldırı sürecinin daha kullanıcı kendi kimlik bilgisini (login/authentication) girmeden uygulamanın kullanacağı oturum çerezini atamasıyla başlıyor olmasıdır. Saldırgan kullanıcıya bir oturum çerezi atamak zorunda olmadan oturum öncesi oluşturulmuş olan bu kimliği çalarak, oturum açma işlemini gerçekleştirmiş veya gerçekleştirecek olan kullanıcının oturumunu ele geçirir. Oturum sabitleme (Session Fixation attack) saldırısında ise saldırgan bildiği bir oturum çerezinin kullanıcı tarafından oturum açma işlemi gerçekleşmeden kullanılmasını sağlar ve böylece kullanıcının saldırgan tarafından atanmış çerez ile oturum açması sonucu oturumu ele geçirir veya kendi oturumundan devam etmesini sağladığı kullanıcının bilgi gizliliğini bozabilir. Bu duruma örnek olarak, kullanıcının kendi oturumunun değiştiğini fark etmemesi ve kendi (kredi kartı, özlük, vs.) bilgilerini saldırgana ait oturuma girerek kaydetmesi gösterilebilir.

Oturum öncesi tanımlı oturum açma çerezi açığının bulunduğu her sistemde, oturum açılması (authentication) sonrası yeni bir çerez oluşturulmaması nedeniyle **Oturum Sabitleme** saldırısından (Session Fixation attack) bahsedilebilir iken, Oturum Sabitleme saldırısının yapıldığı her sistemde bu açıktan söz edilemeyebilir. Oturum açma çerezinin kimlik bilgilerinin girildiği sayfada kimlik bilgileri girilmeden önce oluşturulduğu bir sistemde, kullanıcı girişi yapıldıktan sonra değiştirilmeyen bir çerez, aynı şekilde bir saldırgan tarafından önceden tanımlanabilir ve oturum sabitleme saldırısı gerçekleştirilebilir.

Oturum sabitleme saldırısının yapıldığı bir sistemde çerezin kimlik bilgilerinin girilmeden oluşturulmadığı durumlarda veya oluşturulduktan sonra "session_regenerate_id (PHP)" gibi bir fonksiyon ile değiştirildiği senaryolarda oturum öncesi tanımlı çerez açığı bertaraf edilmiş olsa dahi bu durum oturumun devam ettiği herhangi bir anda çerezin bir saldırgan tarafından değiştirilip sabitlenmesine mani olmamaktadır. Kendi oluşturduğu oturumu kullanıcıya atayan bir saldırgan, kendi oturumu üzerinden kullanıcının hareketlerini izleyebilir ve karşı tarafın kimlik bilgisini geçirmese dahi yetkisiz erişim sağlayabilir veya gizliliği bozabilir.

Kullanıcı giriş yapmadan önceki çerez bilgisi (Session ID) ve açıktan etkilenen PHP kodu



iosec_login.php

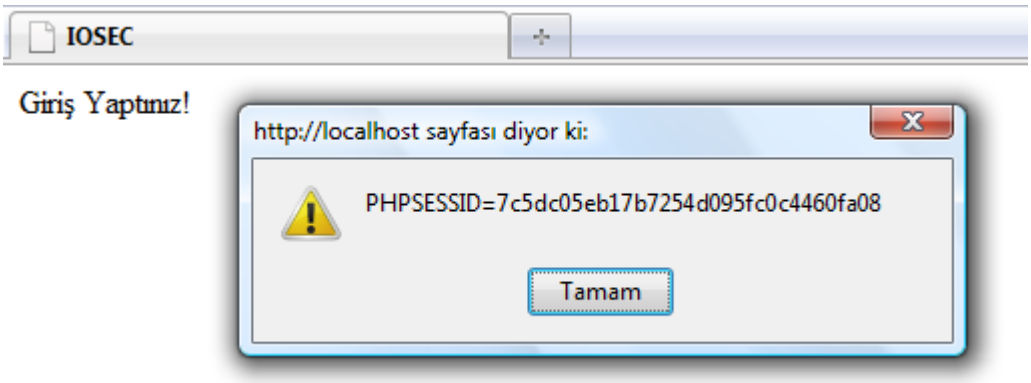
```php
<?php
session_start();
if (isset($_POST["submit"])) {
  if ($_POST["user"] == "gokhan" && $_POST["pass"] == "muharremoglu") {

    $_SESSION["username"] = $_POST["user"];

  }
}
?>

<html>
<head>
<title>IOSEC</title>
</head>
<body>
<?php
if (isset($_SESSION["username"])) {
  echo("Giriş Yaptınız!");
} else {
?>
<form method="post">
Oturum Öncesi Tanımlı Oturum Açma Çerezi Açığı <br><br>
Kullanıcı Adı :<input type="text" name="user" /><br />
Parola :<input type="text" name="pass" /><br />
<input type="submit" name="submit" value="Login" />
</form>
<?php
}
?>
</body>
</html>
```

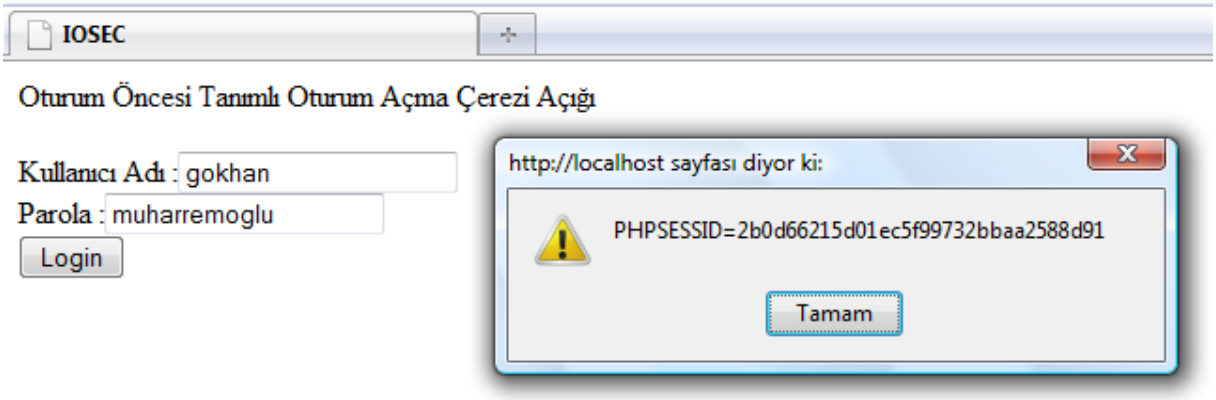
Kullanıcı giriş yaptıktan sonraki <u>değişmeyen</u> çerez bilgisi (Session ID)



Yukarıdaki ekran görüntülerinde oturum açıldıktan sonraki çerez bilgisinin kullanıcı adı ve parola girilmeden önceki çerez bilgisi ile aynı olduğu görülmektedir. Bu durumda oturum açılmamış bir sistem çerezi neredeyse oturum açılmış bir sistem çerezi ile aynı önem düzeyine sahiptir.

Aşağıdaki ekran görüntülerinde oturum sonrası oluşturulan çerez değiştirilerek session_regenerate_id() fonksiyonu ile açık giderilmiştir.

Kullanıcı giriş yapmadan önceki çerez bilgisi (Session ID) ve açıktan <u>etkilenmeyen</u> PHP kodu



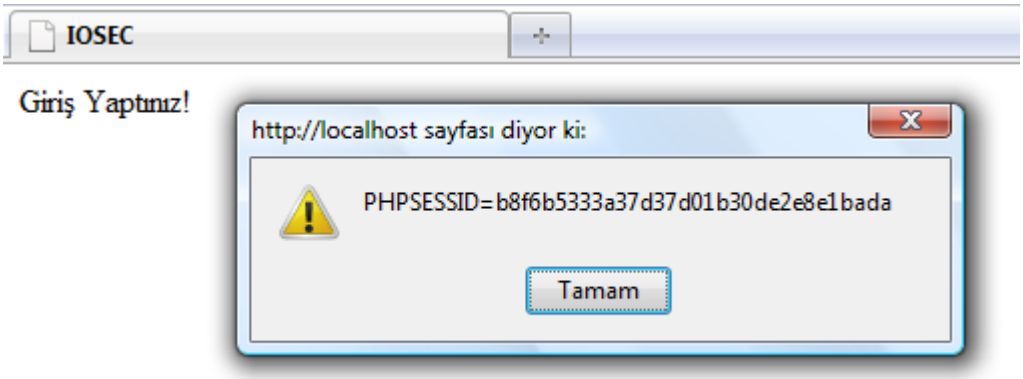
iosec_login.php

```php
<?php
session_start();
if (isset($_POST["submit"])) {
  if ($_POST["user"] == "gokhan" && $_POST["pass"] == "muharremoglu") {

    $_SESSION["username"] = $_POST["user"];
    session_regenerate_id();
  }
}
?>

<html>
<head>
<title>IOSEC</title>
</head>
<body>
<?php
if (isset($_SESSION["username"])) {
  echo("Giriş Yaptınız!");
} else {
?>
<form method="post">
Oturum Öncesi Tanımlı Oturum Açma Çerezi Açığı <br><br>
Kullanıcı Adı :<input type="text" name="user" /><br />
Parola :<input type="text" name="pass" /><br />
<input type="submit" name="submit" value="Login" />
</form>
<?php
}
?>
</body>
</html>
```

Kullanıcı giriş yaptıktan sonraki yeniden üretilmiş çerez bilgisi (Session ID)



Örnek PHP Kodu:

```php
<?php
session_start();
if (isset($_POST["submit"])) {
  if ($_POST["user"] == "gokhan" && $_POST["pass"] == "muharremoglu") {

    $_SESSION["username"] = $_POST["user"];
    session_regenerate_id();
  }
}
?>

<html>
<head>
<title>IOSEC</title>
</head>
<body>
<?php
if (isset($_SESSION["username"])) {
  echo("Giriş Yaptınız!");
} else {
?>
<form method="post">
Oturum Öncesi Tanımlı Oturum Açma Çerezi Açığı <br><br>
Kullanıcı Adı :<input type="text" name="user" /><br />
Parola :<input type="text" name="pass" /><br />
<input type="submit" name="submit" value="Login" />
</form>
<?php
}
?>
</body>
</html>
```

AÇIKTAN ETKİLENEN BİR SİSTEMDEKİ HTTP ÜST BAŞLIK BİLGİLERİ

İlk defa görüntülenen kullanıcı girişi ekranı (oturum öncesi)

(Status-Line) HTTP/1.1 200 OK
Server Apache/2.2.3 (CentOS)
Set-Cookie PHPSESSID=**8usd2oeo11a8cod9q3lnev9je2**; path=/
Expires Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma no-cache
Content-Type text/html
Content-Length 308
Date Tue, 10 Jul 2012 06:16:57 GMT
X-Varnish 1922993981
Age 0
Via 1.1 varnish
Connection keep-alive

Yollanan kimlik bilgileri

(Request-Line) POST /iosec_login_vulnerable.php HTTP/1.1
Host www.iosec.org
User-Agent Mozilla/5.0 (Windows; U; Windows NT 6.0; tr; rv:1.9.2.25) Gecko/20111212 Firefox/3.6.25 (.NET CLR 3.5.30729; .NET4.0E)
Accept text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language tr-tr,tr;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding gzip,deflate
Accept-Charset ISO-8859-9,utf-8;q=0.7,*;q=0.7
Keep-Alive 115
Connection keep-alive
Referer http://www.iosec.org/iosec_login_vulnerable.php
Cookie PHPSESSID=**8usd2oeo11a8cod9q3lnev9je2**
Content-Type application/x-www-form-urlencoded
Content-Length 42
POST DATA
user gokhan
pass muharremoglu
submit Login

Doğrulanan kimlik bilgileri ile açılan oturum

(Status-Line)	HTTP/1.1 200 OK
Server	Apache/2.2.3 (CentOS)
Set-Cookie	PHPSESSID=**8usd2oeo11a8cod9q3lnev9je2**; path=/
Expires	Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control	no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma	no-cache
Content-Type	text/html
Content-Length	308
Date	Tue, 10 Jul 2012 06:16:57 GMT
X-Varnish	1922993981
Age	0
Via	1.1 varnish
Connection	keep-alive

Bir saldırgan bu açığı kullanarak aşağıdaki gibi bir senaryo ile oturum hırsızlığı (Session Hijacking) saldırısı gerçekleştirebilir:

"Bir bankanın şubesinde ortak kullanılan ancak sadece bankanın Web sitesinin kullanıcı girişi sayfasının görüntülenmesine izin verilen bir terminal/kiosk sisteminde tarayıcı üzerinde bulunan adres çubuğu görünür ve değiştirebilir haldedir. Ancak adres çubuğuna girilen bankanın Web siteleri haricindeki bütün Web siteleri güvenlik duvarı aracılığı ile açılmadan doğrudan bankanın Web sitesine yönlendirilmektedir.

Adres çubuğuna "javascript:alert(document.cookie)" yazarak Oturum Öncesi Tanımlı Oturum Açma Çerezi'ni (Session ID) bir kağıda yazan saldırgan, bu bilgiyi kiosk ile oturum açan bütün kullanıcılar için sırayla kullanarak, kiosk ile giriş yapan bütün kullanıcıların oturumlarını çalabilir."

Benzer bir şekilde oturum açma ekranında HTTP kanalını kullanan ancak form verilerini HTTPS üzerinden yollayarak karışık mod (Mixed Mode) şifreleme kullanan bir Web sayfasında bu açığın bulunması, ağ üzerinde dinleme yapılarak oturumun daha HTTP kanalında iken çalınabilmesine olanak sağlamaktadır.

Oturum Öncesi Tanımlı Oturum Açma Çerezi açığının bu senaryoda tam olarak işlemesi için belli başlı "düşük" seviye olarak nitelendirilmiş açığın bankanın Web sitesinde bulunması gerekmektedir. Bunlardan bazıları şunlardır:

- User-Agent, IP kontrolünün yapılmaması (Same Origin Policy),
- Adres çubuğuna erişilebilir olma,
- Http-Only Cookie kullanılmaması,
- Oturum sonunda yeni Session ID atanmaması.

Buradaki senaryo ile beraber göz önünde bulundurulması gereken nokta, bilgi güvenliğinde "bir zincirin en zayıf halkası kadar sağlam olması" felsefesidir. Her zayıf halkanın (düşük-kritik seviye bir açığın) daha kolay kopabilir bir zincir meydana getirdiği unutulmamalıdır. Tek başına önemsiz gibi görünen birkaç düşük seviye açık, ciddi güvenlik ihlallerinin kapısını aralayabilir.

Açıktan etkilenen örnek sayfa:
http://www.iosec.org/iosec_login_vulnerable.php

[1]- http://www.bilgiguvenligi.gov.tr/siniflandirilmamis/oturum-acma-sistemleri-tasariminda-guvenlik.html

Gökhan Muharremoğlu
Bilgi Güvenliği Uzmanı
gokhan.muharremoglu@iosec.org