ME Mustafa Erdinç ZORBA

Web Yazılım Güvenliği - 1

Merhabalar, bu yazımda bir web projesini get sorgularına karşı nasıl güvenli hale getirebiliriz, mevcut güvenlik açıkları/tehlikeleri nelerdir, bunlardan nasıl korunuruz sorularına cevaplar bulacağız.

Öncelikle kendi yazdığımız sistemi ele alıp, bu sisteme ziyaretçi/saldırgan girdiğinde ne gibi tehlikeler yaratır bunları inceleyeceğiz.

Bildiğiniz gibi bir web sitesinde yapılan tüm istekler (request) get veya post şeklinde olmalıdır. Bunu biraz daha açacak olursak; Bir kullanıcı web sitenizde dolaşırken sürekli get veya postlar ile taleplerde bulunur, web sitesi ve server'da bu taleplere göre sonuçlar döndürür.

Bu yüzden ilk öncelikle inceleyeceğimiz konu istek (request) güvenliği.

Web sitesine yapılan isteklerde (request) ne gibi tehlikeler söz konusu olabilir ?

Öncelikle get ile başlayalım.

Bir web sitesinde yaptığınız tüm istekler, sunucuya herhangi bir değer post edilmediği sürece get türündedir. Basiteçe açıklamak gerekirse; www.merdincz.com veya www.merdincz.com/test.php?a=12&v=test arasında hiç bir fark yoktur. Sunucu kullanıcıdan herhangi bir değişken almaz aldığı tüm değişkenler url üzerindedir. (http requestlerden bahsetmiyorum)

İkinci verdiğim örnekten yola gidersek www.merdincz.com/test.php?a=12&v=test adresine girdiğinizde sunucu test.php ye a=12 v=test değişkenlerini gönderir. test.php'de buna göre sonuç döndürür.

Örnek test.php dosyamızın aşağıdaki gibi olduğunu düşünelim.

<?php
$a = $_GET["a"];
$v = $_GET["v"];
echo "Adınız: $a
Yaşınız: $v";
?>

merdincz.com/test.php?a=12&v=test adresine girdiğinizde karşınıza çıkacak değer;
Adınız: 12
Yaşınız: test

Şeklindedir.

Doğal olarak biz adres satırındaki a ve v değişkenlerinin yanındaki değerleri değiştirip çıkan sonucu da değiştirebiliriz.

Örnek vermek gerekirse adresimiz merdincz.com/test.php?a=gelecek&v=1 yaptığımızda sonuç;
Adınız: gelecek
Yaşınız: 1
Şeklinde olacaktır.

Peki bu ne gibi güvenlik açıkları doğuracaktır ?

Gördüğünüz gibi test.php'de herhangi bir filtreleme veya kontrol fonksiyonu yok. Dileyen kullanıcı a veya v değişkenin yanına istediği kodu yazarak çalıştırabilir.

Örnek vermek gerekirse

merdincz.com/test.php?a=<script>alert('document.cookie');</script>&v=12

yazarak site üzerinden kendi cookie bilgilerini görebilir. Kendi cookie bilgileri pek önemli değil ama bu bilgileri başka amaçlarda kullanabilir.(makalenin devamında xss ve cookie güvenliği konusunda bahsedicez)

Peki bu durumda ne yapmak gerekiyor ya kullanıcının url'yi düzeltmesini engellememiz ya da kullanıcının bir şekilde url'yi değiştirebileceğini hesaba katıp test.php'de sonuç gösterirken sonucu filtrelememiz gerekiyor.

İşi şansa bırakmamak ve riske etmemek adına her zaman ikinci yöntem seçilip test.php üzerinde kendi güvenliğimizi sağlamalıyız. Bunu nasıl yapabiliriz peki ?

Php'de bir çok fonksiyon ile girilen verinin html olup olmadığını verinin html olarak yorumlanmamasını sağlayabilirsiniz.

Bizim yapacağımız işlem adres satırını text olarak yorumlatmak bu yüzden **htmlspecialchars()** fonksiyonunu kullanacağız.

Bu fonksiyonun içine girilen tüm değerlerin çıktısı direk olarak verilir. Yani siz echo htmlspecialchars("**asd**") yazsanız dahi browserda sadece **asd** görürsünüz. Kalın bir şekilde **asd** yazmaz.

Bu fonksiyonu test.php'ye;

<?php
$a = $_GET["a"];
$v = $_GET["v"];
echo "Adınız: htmlspecialchars($a)
Yaşınız: $v";

?>

Şeklinde uyarlayabiliriz.

Normalde aynı değeri $v değişkeni için de kullanmamız gerekiyor ama onun için farklı bir örnek kullanıcaz.

Şu anda gelen değerin html olarak dışarı çıktı vermesini engelledik peki bu bizi güvenli bir hale getirdi mi ?

Tabii ki hayır!

test.php'yi yukarıdaki gibi değil aşağıdaki gibi olduğunu, veritabanından bilgi çektiğinizi düşünün.

<?php
$a = $_GET["a"];
$v = $_GET["v"];
echo "Adınız: htmlspecialchars($a)
";
$sql = mysql_query("Select * from uyeler where yas='$v'");
?>

(kodun tabii ki devamı olması gerekiyor)

Test.php'de şimdi yaptığımız işlem ise ekrana kullanıcının güvenli bir şekilde adını yazdırıp, veritabanından adres satırında yazan yaş'a sahip üyeleri aratmak.

Yani merdincz.com/test.php?a=ahmet&v=22 yazdığınız zaman karşınıza Adınız: Ahmet yazıp devamında 22 yaşındaki üyeleri getirmesi gerekiyor.

Peki bunda tehlike nedir ?

Girilen yaş değeri ile ilgili herhangi bir kontrolde bulunmadan değerin rakam olarak geldiğini varsayıp işlem yaptırıyoruz.

Bu durumda kullanıcı yaş dışında değerler girerek: Örneğin "22'a--" bizim sayfamızda çıkan sonucu manipüle edip, kendi istediği sql sorgularını çalıştırarak yönetim paneli şifrelerini vs.. elde edebilir (Sql injection konusuna ilerleyen makalelerde detaylıca giricez)

Peki şimdi ne yapmamız gerekiyor ?

Tabii ki $v değişkeninin sayı olup, olmadığını kontrol etmemiz ve sql bazlı sıkıntı yaşamamak için mysql_real_escape_string fonksiyonunu kullanmamız gerekiyor.

Test.php'yi bu gelişmeler ile tekrar modifiye edelim.

<?php
$a = $_GET["a"];
$v = $_GET["v"];
$yas = intval($v);
echo "Adınız: htmlspecialchars($a)
";
$sql = mysql_query("Select * from uyeler where yas='".mysql_real_escape_string($v)."'");
?>

Yaptığımızda kullanıcı merdincz.com/test.php?a=ahmet&v=12' OR ''=' şeklinde bir deneme yapsa dahi çabaları boşa gidecektir.

Küçücük test.php'de bu kadar önlem aldık peki bunlar bizim test.php'mizi yeteri kadar güvenli kıldı mı ? Tabii ki hayır.

Şimdi test.php dosyamızı şu şekilde hayal edelim.

<?php
$a = $_GET["a"];
$v = $_GET["v"];
$yas = intval($v);
echo "Adınız: htmlspecialchars($a)
";
include($v);
?>

Bu yaptığımız işlemin mantığı şu. Yine her zamanki gibi Adınız: yazdırıyoruz ardından ise $v değişkenine atanmış sayfayı çalıştırıyoruz.

Yani merdincz.com/test.php?a=Ahmet&v=haberler.php dediğimiz zaman Adınız: Ahmet yazıp haberler.php sayfası çalışıyor.

Peki bu gibi durumlarda ne gibi bir önlem alabiliriz veya bunun zararları nedir ?
a değikenine Ahmet v değişkenine ../../../../../../../../../etc/passwd şeklinde bir değişken girildiğini düşünelim.

Bu sayede Ekrana Ahmet yazar ama alt tarafta sunucunuzun etc/passwd bilgilerini yazar. (RFI ve LFI açıklarıyla da ileride daha detaylı bilgi paylaşıcam)

Peki böyle bir durumdan nasıl korunabilinir ?

Öncelikle get ile herhangi bir sayfayı include etmek son derece tehlikelidir. Bunun yerine gelen veriyi

if,else; switch,case gibi fonksiyonlar ile kontrol edip buna göre sayfa include edebilir sunucunuzda safe mode fonksyionunu on yapabilirsiniz. (php 6'da bu fonksiyona ihtiyaç duyulmayacak.)

Peki tüm bunları yaptığımızda GET isteklerine karşı tamamen güvenli oluyor muyuz ? Ne yazık ki hayır.

Her ne kadar amacım tek dökümanda tüm konulardan azar azar bahsetmek olsa da dökümana başlayınca konu giderek detaylandı. Bu yüzden dökümanı parçalayarak bol örnekli yazma kararı aldım.

GET isteklerine karşı daha bir çok örnek kullanılabilir. Örneğin browserdan alınan user-agent, refferer verileri vs..

Olayın mantığını kavradığınızı düşünerek get isteğine cevap veren bir sayfa oluşturduğunuzda bu sayfaya değişken üzerinden giren tüm verileri kullanıldığı alana göre kontrol etmeniz gerekmekte. Sql için mysql_real_escape_string ve verinin integer olup olmadığını kontrol etmek(integer sorgular için) güvenli bir yoldur. Çıktı verilen değişkenler için htmlspecialchars,addslashes, htmlentites gibi fonksiyonları kullanabilirsiniz.

Dökümana yeni başladım, diğer dökümanların yazılması uzun vakit alabilir. Bu süre zarfında aklınıza takılan herhangi bir soru(n) veya makaleme eklememi, düzeltmemi istediğiniz herhangi bir yer olması durumunda **** adresine mail atmanız yeterlidir.

Mustafa Erdinç ZORBA
23 Eyl. 2012
info@merdincz.com
www.merdincz.com