1.1.1 Hata Mesajı Yoluyla Bilgi İfşası (Information Exposure Through An Error Message) (CWE-209)

Açıklık Önem Derecesi: Düşük

Açıklığın Etkisi: Bilgi ifşası, Saldırı metotlarını belirlemede kolaylık sağlama

Açıklığın Barındıran Dosyalar/Satırlar:

Proje Dosyası/Dosya Adı	Satır Numarası

Açıklığın Açıklaması:

Uygulamalar bulundukları ortam hakkında, kullanıcıları hakkında veya uygulamanın ilişkili verileri hakkında hassas bilgi içeren hata mesajları oluşturabilirler. Hassas bilgiler kendi başına değerli bir bilgi olabilirler (örn; parola gibi) veya hassas bilgiler daha ciddi başka saldırılar yapmak için kullanılabilirler.

Saldırganlar daha odaklı saldırılar yapmak için hata mesajlarındaki içerikleri kullanabilirler. Örneğin hata mesajları path traversal (dizin gezinme) açıklıklarını sömürme teşebbüslerinde yüklü uygulamanın tam (full) dizin yolunu verebilir. Bundan hareketle bu bilgi hedeflenen dosyaya gitmek için uygun sayıda ".." üst dizin karakterinin seçilmesinde kullanılabilir. Örneğin hata mesajları SQL Enjeksiyonunun kullanıldığı bir saldırıda saldırı en başta başarılı olamasa da tüm bozuk çalışan sorguyu gözler önüne serebilir. Bundan hareketle sorgu mantığını, hatta sorguda kullanılan muhtemel parolaları ve diğer hassas verileri (örn; veritabanı field'larını (kolon adlarını)) ortaya çıkarabilir.

Uygulamalardaki istemciye dönülen hata mesajları yoluyla yaşanan bilgi ifşasına örnek olarak şu kod bloğu verilebilir:

JAVA:

```
// Güvensiz Kod

// ENG: Handle Exception by Printing To Output
// TR : İstisnayı Çıktıya Basarak İdare Etme

private void wrapCallToDB_Unsafe(HttpServletRequest request)
throws ServletException, IOException {

    String paramValue = request.getParameter("Param");

    try {
        callDbProc(paramValue);
    }
    catch (SQLException ex) {
        ex.printStackTrace();
    }
}
```

Bu örnekte try bloğundan fırlaması muhtemel bir hatanın catch ile yakalandığı ve catch bloğunda ise hata mesajının ekrana basıldığı görülmektedir. Hata mesajı ekrana basma metodu printStackTrace() stderr'e hata detaylarını basar. Burada hata mesajının ekrana basılması bu kod bloğunu güvensiz kılmaktadır. Bu kodun güvenli hali şu şekildedir:

Java:

```
// Güvenli Kod

// ENG: Write Exception Details to Log, Send
//      Generic Error Message
// TR : İstisna Detaylarını Log'a Yazdırma ve
//      Genel Bir Hata Mesajı Dönme

private void wrapCallToDB_SafePrintToLog(HttpServletRequest request)
throws ServletException, IOException {

    String paramValue = request.getParameter("Param");

    try {
        callDbProc(paramValue);
    } catch (SQLException ex) {
        writeExceptionToLog(ex);
        System.err.println("Veritabanı Hatası, detaylar için log dosyasına
bakınız.");
    }
}
```

Bu örnekte try bloğundan fırlaması muhtemel bir hatanın catch ile yakalandığı ve catch bloğunda ise hata mesajının ekrana basılması yerine önce log dosyasına basıldığı ve sonra ekrana genel bir (detay unsur içermeyen bir) hata mesajı basıldığı görülmektedir. Uygulamalar ekrana detay hata mesajları bastıklarında "Hata Mesajı Yoluyla Bilgi İfşası" açıklığı vardır şeklinde işaretlenirler.

Kurum uygulamasında "Hata Mesajları Yoluyla Bilgi İfşası" açıklığı tespit edilmiştir:

::::::::BULGU:::::::

Açıklığın Önlemi:

Bu açıklığın önlenmesinde takip edilmesi gereken hususlar şu şekildedir:

- İstisna (exception) bilgileri istemcilere doğrudan sergilenmemelidir. Bunun yerine bu bilgiler detayları ile beraber log dosyalarına log'lanmalıdır ve istemciye durum hakkında bilgilendirici, spesifik olmayan, genel bir hata mesajı sunulmalıdır.

- Uygulama kaynak kodlarında istisna (exception) fırlatabilen herhangi bir metot istisna işleyici (exception handler) bir blokla çevrelenmelidir. Fırlaması beklenen istisnalar için açık bir şekilde (explicitly) istisna işleyici (exception handler) tanımlanmalıyken fırlaması beklenmedik istisnalar için de varsayılan bir istisna işleyici (exception handler) tanımlaması dahil edilmelidir.

- Ayrıca "global istisna işleyici" (global exception handler) bir istisna işleyici (exception handler) ile çevrelenmemiş hataların uygulamadan çıkışını önlemek için yapılandırılmalıdır.

Referanslar:

1. https://cwe.mitre.org/data/definitions/209.html
2. https://www.geeksforgeeks.org/throwable-printstacktrace-method-in-java-with-examples/
3.