1.1.1 Açık Tam Dosya Yolu Kullanılması (Hardcoded Absolute Path) (CWE-426)

Açıklık Önem Derecesi: Düşük

Açıklığın Etkisi: Uygulama güvenliğinin sürdürebilirliğini azaltma

Açıklığın Barındıran Dosyalar/Satırlar:

Proje Dosyası/Dosya Adı	Satır Numarası

Açıklığın Açıklaması:

Açık tam dosya yolu kullanılması uygulamaları kırılgan yapar. Örneğin thick client (masaüstü) uygulamalar farklı farklı istemcilere (sistemlere/ortamlara) indirilip kurulduğunda uygulamadaki tam dosya yolları geçersiz olacağından uygun çalışmaz. Aynı şekilde thick client (masaüstü) uygulamaların kuruldukları sistemlerde/ortamlarda farklı sistem dilleri olabilir ve işletim sistemi mimarilerindeki sistem klasörleri farklı isimde olabilir. Örneğin; ispanyol windows makinelerde "C:\Program Files\" dosya yolunun "C:\Archivoc de programa (x86)" şeklinde olması gibi. Tam dosya yolu kullanılması bu durumlarda uygunsuz olacaktır.

Açık tam dosya yolu kullanılması aynı zamanda sunucu taraflı uygulamalar için de uygun değildir. Örneğin sunucu taraflı uygulama farklı bir sunucuya (sisteme/ortama) taşındığında - ki bu durum her daim bir ihtimal / olasılık olarak vardır - uygulamalardaki tam dosya yolları geçersiz olacağından uygun çalışmaz. Dolayısıyla bu kullanım esnek değildir. Ayrıca thick client (masaüstü) veya sunucu taraflı uygulamaların tasarımı veya gereksinimleri değişirse gelecekte uygulamanın yeni sürümlerinde açık tam dosya yolu kullanılması bakım problemlerine yol açar. Sonuç olarak tam dosya yolu kullanılması bir kod kalitesi bulgusu olarak gelecekte komplikasyonlara (karmaşıklığa) sebep olacağından güvenlik noktasında bir negatiflik oluşturur ve uygulamanın genel güvenlik seviyesini ideale seviyeye göre düşürür.

Tam dosya yolu kullanılmasının güvenliğe dokunan bir başka yanı eğer sunucu taraflı uygulamalar için veri okuma veya yazma işlevlerinde tam dosya yolu kullanılırsa bu durum gizliliğin ihlaline veya zararlı girdilerin programa eklenmesine yol açabilir. Gizliliğin ihlali ile kastedilen sunucuya sızan bir saldırganın uygulama dosyalarında gezinip dosyaları incelerken kaynak kodlardaki tam dosya yolunu görerek işletim sistemi teknolojisini anlayabilmesidir. Böylece gizlilik zaten ihlal olmuşken bir kademe daha ihlal edilmiş olur. Zararlı girdilerin programa eklenmesi ile kastedilen ise bazı spesifik koşullarda (yani başka açıklıkların (örn; güvensiz file upload açıklığının) birleştirilerek yapıldığı bir sızma girişiminde) kötü niyetli kullanıcıların uygulamanın mevcut işlevselliğini kendi emelleri doğrultusunda

kullanması (mevcut işlevselliği override etmesi) ve uygulamanın keyfi bir programı çalıştırmasını sağlamasıdır. Bu ikinci yolda saldırgan sunucuya yüklediği bir kodu sunucu taraflı uygulamanın kendisi üzerinden tetikleyebilir/çalıştırabilir ve sızma derinliğini genişletebilir. Ayrıca windows sistemlerde sistem klasörlerinin ve kullanıcı profil klasörlerinin haricindeki tüm klasörlerde yetkili herhangi bir kullanıcının varsayılan olarak ful okuma ve yazma izni olduğundan sunucu taraflı uygulama bunları koruyor varsayımına rağmen yetkisiz/kötü niyetli bir kullanıcı bu klasörlerdeki herhangi bir hassas veriye erişebilir ve bu korunmayan klasörlerdeki varolan yüklü uygulamaların/programların üzerine farklı açıklıklar üzerinden (örn; güvensiz file upload açıklığı üzerinden) yazma yapabilir (zararlı kod ekleyebilir/bulaştırabilir). Böylece saldırgan, kurban sistemde yüklü bu enfekte olmuş uygulamayı/programı esas sunucu taraflı uygulama üzerinden (tam dosya yolu üzerinden) zararlı kodu tetikleyebilir ve sistemin içinde daha derinlere inebilir.

Uygulamalar kaynak kodlarında tam dosya yolu kullandıklarında "Açık Tam Dosya Yolu (CWE-426)" açıklığı olarak işaretlenirler. Bu açıklığı örneklemek maksadıyla java dilinde bir örnek verilmiştir:

Java

```java
// GÜVENSİZ KOD
public File getLogFile() {
    String filename = "C:\Logs\myapp.log";
    File logFile = new File(filename);
    return logFile;
}
```

Java:

```java
// GÜVENLİ KOD
public File getLogFile() {
    Properties props = this.Properties;
    String filename = (String)props.get("logDirectory") +
(String)props.get("logFilename");
    File logFile = new File(filename);
    return logFile;
}
```

İlk kod bloğunda tam dosya yolu yer aldığından kod bloğu güvensizdir. İkinci kod bloğunda tam dosya yolu konfigurasyon dosyasından çekilerek kullanıldığından kod bloğu güvenlidir.

Kurum uygulamada "Açık Tam Dosya Yolu Kullanılması (CWE-426)" açıklığı tespit edilmiştir:

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

Açıklığın Önlemi:

- Hangi tür uygulama kullanılıyorsa kullanılsın uygulamaya açık tam dosya yolu (hardcoded absolute path) eklemeyin.

- Bunun yerine her sistemde/ortamda gerek duyulduğunda modifiye edilebilecek bir harici konfigurasyon dosyası kullanın ve bu dosyaya tam dosya yollarını depolayın.

- Alternatif olarak eğer dosya yolu eklenecek olan dosya uygulamanın kök dizini altında yer alıyorsa tam dosya yolu (absolute file path) kullanmak yerine göreceli dosya yolu (relative file path) kullanılabilir.

- Tüm çalıştırılabilir dosyaları korunaklı program dizininde (Windows'larda varsayılan olarak C:\Program Files\ dizini altında) depolamayı tercih edin.

- Linux'da veya farklı işletim sistemlerinde bir "system jail" (örn; chroot) kullanın ve tüm programları ve dosyaları orada depolayın.

Referanslar:

1. https://stackoverflow.com/questions/30776044/why-it-is-not-suggested-to-pass-hardcoded-absolute-path-name-to-file-object-cons