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

Açıklık Önem Derecesi: Düşük

Açıklığın Etkisi: Uygulama güvenliğinin sürdürülebilirliğini 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.

Açık tam dosya yolu kullanılmasının güvenliğe dokunan diğer yanlarına değinecek olursak eğer uygulama açık tam dosya yolu kullanıyorsa bu durum gizliliğin ihlaline yol açabilir. Gizliliğin ihlali ile kastedilen sunucuya sızan bir saldırganın uygulama dosyalarında gezinip dosyaları incelerken kaynak kodlardaki açık tam dosya yollarını (örn; c:\, d:\ veya /var/www/html v.b. ile başlayan yolları) görerek işletim sistemi teknolojisi türünü anlayabilmesidir. Böylece gizlilik saldırganın belli ölçüde içeri girmesiyle zaten ihlal olmuşken bir kademe daha ihlal edilmiş olur.

Açık tam dosya yolu kullanılmasının güvenliğe dokunan bir diğer yanı eğer uygulama açık tam dosya yolu kullanıyorsa ve ayrıca açık tam dosya yolunu veri okuma ve yazma için kullanıyorsa saldırganlar farklı açıklıklar yoluyla açık tam dosya yolundaki programın beklenen fonksiyonelliğinin üzerine yazma işlemi (override işlemi) yapabilirler, uygulamayı çalıştıran kullanıcılar ise açık tam dosya yolundaki programı çalıştırırken esasında açık tam dosya yolundaki saldırganca enfekte olmuş programı çalıştırabilirler. Bu da keyfi komut çalıştırma ile sonuçlanabilir.

Ayrıca windows sistemlerde sistem klasörlerinin ve kullanıcı profil klasörlerinin haricindeki tüm klasörlerde yetkili herhangi bir kullanıcının varsayılan olarak ful okuma ve yazma izni var olduğundan uygulama eğer açık tam dosya yolu ile bu dizinlerdeki bir programı kullanıyorsa yetkisiz/kötü niyetli bir kullanıcı bu korunmayan klasörlerdeki varolan yüklü uygulamaların/programların üzerine farklı açıklıklar üzerinden yazma yapabilir (zararlı kod ekleyebilir/bulaştırabilir). Böylece kurban bu enfekte olmuş açık tam dosya yolundaki programı çalıştırdığında zararlı kod tetiklenebilir ve saldırgan sistemin içinde derinlere inebilir.

Kurum uygulamanın gelecekte farklı bir sunucuya / platforma taşınabileceği olasılığını dikkate alarak kurum uygulamanın sürdürülebilirliğini temin etmek adına (örn; gelecekteki kendinize veya gelecekteki yeni geliştiricilere uygulamayı geliştirmede sürekliliği sağlama noktasında kolaylık sunmak adına), gizliliği sağlamak adına ve keyfi komut çalıştırılmasını önlemek adına kaynak kodda açık tam dosya yolu kullanılmamalıdır. Açık tam dosya yolu kullanımı ile;

- Uygulama kırılgan olur ve sürdürülebilirliği azalır,
- Gizlilik bir kademe ihlal olur,
- Farklı zafiyetler söz konusu olduğunda keyfi komut çalıştırma ile sonuçlanabilir.

Uygulamalar kaynak kodlarında açık tam dosya yolu kullandıklarında "Açık Tam Dosya Yolu (CWE-426)" açıklığı var 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:

```
// 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:::::::::

Şekil XXX. Kırılgan Uygulama

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
2. https://medium.com/@jordan.l.edmunds/please-stop-hard-coding-file-paths-609c769f9537

3. https://cwe.mitre.org/data/definitions/426