

1.1.1 Dizilerin Public, Final ve Static Tanımlanması (Array Declared Public, Final and Static) (CWE-582)

Açıklık Önem Derecesi: Düşük

Açıklığın Etkisi: Sabit dizilerin modifiye edilebilmesi

Açıklığın Barındıran Dosyalar/Satırlar:

Proje Dosyası/Dosya Adı	Satır Numarası

Açıklığın Açıklaması:

Uygulamalarda dizi nesneleri / değişkenleri public, final ve static tanımlanabilmektedir. final kullanımı umulduğu üzere dizinin modifiye edilmesini önler ama dizinin elemanlarının modifiye edilmesini önlemek için yeterli değildir. final belirteci nesneye / değişkene bir dizinin sadece bir defa atanmasını zorlar, ancak dizi elemanlarına değerlerin yalnızca bir kez atanabilmesini garanti etmez. Böylesi bir durumda bir de dizi public tanımlanırsa zararlı bir program dizinin depolu değerlerini değiştirebilir. Bu encapsulation'ının ihlali anlamına gelir.

Bu problemi göstermek adına örnek java kod bloklarına yer verilmiştir:

Java - Güvensiz Kod Bloğu:

```
public class example {
    /**
     * Get a new random id.
     */

    public static final String[] ALLOWED_URLS = new String[]
{"http://amazon.com", "http://cnn.com"};

    public String getRandId() {
        Random random = new SecureRandom();
        byte[] randomBytes = new byte[csrfSize];
        random.nextBytes(randomBytes);
        return Hex.encodeHexString(randomBytes);
    }
}
```

Java güvensiz kod bloğu örneğinde bir dizi nesnesinin public, static ve final tanımlandığı gösterilmiştir. Burada kullanılan public, static ve final olan dizi nesnesi izinli uzak sistemler için beyaz liste güvenlik önlemini göstermektedir. Diziler tanımlanırken final kullanılsa bile umulanın aksine dizinin elemanları değiştirilebilir olduğundan dışarıdan uygulamanın hesap

edilmemiş farklı bir modülü ile veya zararlı kodlar ile bu beyaz liste önlemi modifiye edilebilir ve güvenlik önlemi atlatılabilir.

Java - Güvenli Kod Bloğu:

```
public class example {
    public static void main(String[] argv)
        throws Exception
    {
        try {

            // creating object of ArrayList<Character>
            List<Character> list = new ArrayList<Character>();

            // populate the list
            list.add('X');
            list.add('Y');

            // printing the list
            System.out.println("Initial list: " + list);

            // getting unmodifiable list
            // using unmodifiableList() method
            List<Character> immutablelist = Collections
            .unmodifiableList(list);

            // Adding element to new Collection
            System.out.println("\nTrying to modify" + " the
            unmodifiablelist");
            immutablelist.add('Z');
        }

        catch (UnsupportedOperationException e) {
            System.out.println("Exception thrown : " + e);
        }
    }
}
```

Java güvenli kod bloğunda ise bir dizi nesnesi değiştirilemez şekilde - ArrayList ile - tanımlanmıştır. Ayrıca Collection ile unmodifiableList() kullanarak dizinin elemanları da değiştirilemez yapılmıştır. Bu sayede diziye önce X ve Y değerleri eleman olarak eklenebilirken dizi elemanları değiştirilemez yapıldığında Z değeri diziye eleman olarak eklenemez duruma gelmiştir. Bu örnek dizi tanımlaması örneğin bir önceki beyaz liste önleminde kullanılarak uygulama güvende tutulabilir.

Public, static ve final tanımlanan sabit olduğu sanılan dizi hangi amaçla kullanılıyorsa o amaç üzere suistimal edilebilir.

- Örneğin sabit olduğu sanılan dizi kullanıcı rollerini tutmak için kullanılıyorsa dışarıdan modifiye edilerek yetki yükseltme saldırılarında kullanılabilir.
- Örneğin sabit olduğu sanılan dizi kritik bir veri taşıyorsa dışarıdan modifiye edilerek uygulamanın verisi ve çalışma şekli bozulabilir.

Yaşanabilecek olası problemlerin önüne geçmek için güvensiz dizi tanımlamasının düzeltilmesi önerilmektedir. Dizilerin güvensiz tanımlanmasına “Dizilerin Public, Final ve Static Tanımlanması (CWE-582)” açıklığı adı verilir.

Kurum uygulamada Güvenli Olmayan Dizi Tanımlaması (CWE-582) açıklığı tespit edilmiştir.

.....BULGU:.....

Açıklığın Önemi:

- Dizi (field'ının / instance variable'ının) erişilebilirliği azaltılmalıdır - örneğin private yapılmalıdır - veya dizi üyesinin tipi değiştirilemez tip yapılmalıdır.
- Dizi için uygun bir collection tipi kullanılabilir. Collection kullanarak geliştirici dizi elemanı güncellemelerinin izinli olup olmadığını kontrol edebilir. Geliştirici Collections.unmodifiableList() kullanarak collection'ın güncellenmesini önleyebilir. Bunların yanısıra Collection değiştirilemez olsa bile collection'da depolu tiplerin de değiştirilemez olduğundan emin olunmalıdır. Aksi takdirde sabit varsayılan elemanlar üzerinde istenmeyen değişiklikler yine görülebilir.

Referanslar:

1. <https://cwe.mitre.org/data/definitions/582.html>
2. <https://vulncat.fortify.com/en/detail?category=Unsafe%20Mobile%20Code&subcategory=Unsafe%20Array%20Declaration#Java%2fJSP>