1.1.1 Public Static Üye Değişkenin Final Olarak Tanımlanmaması (Public Static Field Not Marked Final) (CWE-500)

Açıklık Önem Derecesi: Düşük

Açıklığın Etkisi: Dayanıklılık eksikliği

Açıklığın Barındıran Dosyalar/Satırlar:

Proje Dosyası/Dosya Adı	Satır Numarası

Açıklığın Açıklaması:

Bir nesnenin public üye değişkenleri (member fields) harici class'lar tarafından değiştirilebilirler. Genellikle nesnelerin üye değişkenlerine harici class'ların doğrudan erişmesi istenmez. Örneğin iyi bir nesne yönelimli program tasarımında üye değişkenlerin diğer class'lara teşhir edilmesini önlemek için encapsulation (kapsülleme) kullanılır. Eğer sistem bu tarz üye değişkenlerin değiştirilemeyeceğini varsayarsa, fakat üye değişkenler değiştirilebilir ise bu durumda zararlı kodlar sistemin davranışını kötü yönde değiştirebilir.

Bu açıklığı somutlaştırmak için JAVA ve C++ örneklerine yer verilmiştir:

JAVA - Güvensiz Hal:

```
public class MyClass
{

        public static int ERROR_CODE = 100;

        //...

}
```

JAVA - Güvenli Hal:

```
public class MyClass
{

        public static final int ERROR_CODE = 100;

        //...

}
```

Kod bloğunu yorumlayacak olursak “JAVA - Güvensiz Hal” örneğinde ERROR_CODE üye değişkeni public ve static olarak tanımlanmıştır, fakat final olarak tanımlanmamıştır. Böylesi bir durumda zararlı bir kod bu hata kodunu (ERROR_CODE üye değişkeni değerini) değiştirebilir ve programın beklenmeyen şekilde davranmasına sebep olabilir.

Bir diğer JAVA örneği olarak şu verilebilir:

Java - Güvensiz Hal:

```
public class SomeAppClass {

        public static String appPropertiesFile = "app/Application.properties";

        // ...
}
```

Java - Güvenli Hal:

```
public class SomeAppClass {

        public static final String appPropertiesFile =
"app/Application.properties";

        //...
}
```

“Java - Güvensiz Hal” kod bloğunda public ve static tanımlanan değişkenin final olarak kullanılmaması gösterilmiştir. Bu durum zararlı kodlar ile public ve static üye değişkenin manipüle edilmesi ve üye değişkende geliştiricinin tanımladığı konfigürasyon dosyası yerine farklı bir dosyanın çağırılmasıyla sonuçlanabilir. Güvenli kod bloğunda ise public ve static üye değişkenin final ile güvenli şekilde kullanıldığı gösterilmiştir.

C++ - Güvensiz Hal:

```
class SomeAppClass {

        public:
                static string appPropertiesConfigFile =
"app/properties.config";

        // ...

}
```

C++ - Güvenli Hal:

```
class SomeAppClass {

        public:
                static const string appPropertiesConfigFile =
"app/properties.config";

        // ...
}
```

"C++ - Güvensiz Hal" kod bloğunda public ve static tanımlanan değişkenin final olarak kullanılmaması gösterilmiştir. Bu durum yine zararlı kodlar ile public ve static üye değişkenin manipüle edilmesi ve üye değişkende geliştiricinin tanımladığı konfigürasyon dosyası yerine farklı bir dosyanın çağırılmasıyla sonuçlanabilir. Güvenli kod bloğunda ise public ve static üye değişkenin final ile güvenli şekilde kullanıldığı gösterilmiştir.

Public ve static üye değişkenler final olarak tanımlanmadıklarında "Public Static Üye Değişkenin Final Olarak Tanımlanmaması (CWE-500)" açıklığı olarak ele alınırlar. Kurum uygulamada bu açıklık tespit edilmiştir:

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

Açıklığın Önlemi:

Public ve static tanımlanan üye değişkenler final olarak tanımlanmalıdırlar.

Referanslar:

1. https://cwe.mitre.org/data/definitions/500.html

2. https://vulncat.fortify.com/en/detail?id=desc.structural.java.poor_style_non-final_public_static_field#Java%2fJSP