1.1.1 Kullanılmamış Değişkenler (Unused Variables) (CWE-563)

Açıklık Önem Derecesi: Düşük

Açıklığın Etkisi: Kod kalitesinin düşmesi

Açıklığın Barındıran Dosyalar/Satırlar:

Proje Dosyası/Dosya Adı	Satır Numarası

Açıklığın Açıklaması:

Bir değişken (variable) / bir nesne değişkeni (instance variable) tanımlandığında veya bir değer atamasına tabi tutulduğunda, fakat sonra hiçbir kapsamda (scope'ta) kullanılmadığında "Kullanılmamış Değişkenler (CWE-563)" açıklığı olarak ele alınır. Bu açıklık uygulama güvenliğinde doğrudan bir risk yerine dolaylı bir risk oluşturur. Şöyle ki bu açıklık kodun okunurluğunu azaltıcı bir faktördür. Gelecekte uygulama ve kaynak kodları daha da büyüdüğünde okunurluk azalacağı gibi ekstradan okunurluğu azaltan bu gibi unsurlar arta bırakmak gelecekte uygulama güvenliğini sağlama noktasında negatiflik oluşturacaktır.

Bu problemi göstermek adına örnek kod bloklarına yer verilmiştir:

C - Güvensiz Kod Bloğu:

```c
#include <stdio.h>

// ...

int deneme(){
    // ...

    r = getName();
    r = getNewBuffer(buf);

    // ...
}

// ...
```

Güvensiz C kod örneğinde r değişkeni bir değer atamasına tabi tutulmamaktadır. Fakat değer kullanılmadan üzerine yeni bir değer atamasında daha tabi tutulmaktadır. Burada bir problem söz konusudur. İlk değer boşuna atanmaktadır ya da dalgınlıkla ilk değerin kullanımına yer

verilmemiştir. Bunlar gelecekte kodlamalar daha da kalabalıklaştığında kodların anlaşılırlığını düşürücü etkenlerdir.

C++ - Güvensiz Kod Bloğu:

```
#include <stdio.h>

// ...

int deneme2(){
    // ...

    int i, j;

    for (i=0; i < outer; i++) {
        for (i=0; i < inner; i++) {
            // ...
        }
    }

    // ...
}

// ...
```

C++ güvensiz kod örneğinde i ve j indis değişkenleri tanımlanmıştır. Fakat yazım hatası nedeniyle içiçe for döngülerinden içerideki for döngüsünde j yerine yine i indisi kullanılmıştır. Bu da gelecekte kodlamalar daha da kalabalıklaştığında kodların anlaşılırlığını düşürücü etkenlerden biridir.

Kurum uygulamada kullanılmamış değişken tespit edilmiştir.

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

Açıklığın Önlemi:

Kodlamadaki hatalar giderilmeli ve geleceğe daha temiz bir proje bırakılmalıdır. Çünkü gelecekte güvenliği temin etme şimdikinden daha güç olacaktır.

Referanslar:

1. https://cwe.mitre.org/data/definitions/563.html
2. https://vulncat.fortify.com/en/detail?category=Poor%20Style&subcategory=Variable%20Never%20Used#C%2fC%2b%2b