

### 1.1.1 Boş Try Bloğu (Empty Try Blocks) (CWE-390)

**Açıklık Önem Derecesi:** Düşük

**Açıklığın Etkisi:** Dayanıklılık eksikliği

**Açıklığın Barındıran Dosyalar/Satırlar:**

Proje Dosyası/Dosya Adı	Satır Numarası

**Açıklığın Açıklaması:**

Bir try-catch-finally bloğu kullanıldığında eğer try bloğu boşsa bu durum uygulamanın güvenilir şekilde çalışmasını engeller.

Örneğin Java dilinde bu açıklığa sahip ve açıklığın kapatıldığı kod bloğu örnekleri verilmiştir.

JAVA - Güvensiz Hal (1):

```
try {  
  
}  
catch(SQLException e) {  
    log(e);  
}
```

JAVA - Güvensiz Hal (2):

```
try {  
    //rs = stmt.executeQuery(query);  
}  
catch(SQLException e) {  
    log(e);  
}
```

JAVA - Güvenli Hal:

```
try {  
    rs = stmt.executeQuery(query);  
}  
catch(SQLException e) {  
    log(e);  
}
```

Bu örneklerde kötü kod bloğunda try bloğu hiçbir işlevsel satır içermemektedir veya sadece yorum satırı içermektedir. Bu durum doğrudan olmasa da dolaylı yoldan bir güvenlik açıklığı oluşturmaktadır. İyi kod bloğunda ise try bloğu işlem uygulayan kod satırlarına sahiptir. Bu durum dolaylı güvenlik açıklığı oluşturmamaktadır.

Try bloğunun boş olması ölü kod olarak değerlendirilir. Ölü kod ise kodun daha zor okunmasına, daha zor anlaşılmasına ve daha zor sürdürülebilmesine / bakımının yapılabilmesine neden olur. Yani bu kod kalitesini olumsuz yönde etkiler. Try blokları boş olduğunda “Boş Try Bloğu (CWE-390)” açıklığı olarak işaretlenirler.

Kurum uygulamasında “Boş Try Bloğu (CWE-390)” açıklığı tespit edilmiştir:

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

### **Açıklığın Önlemi:**

Try blokları boş bırakılmamalıdır ve bu bloklara uygulamanın çalışırılığını sürdürücü ve güvenliğini koruyucu kod satırları girilmelidir.

### **Referanslar:**

1. <https://cwe.mitre.org/data/definitions/390.html>
2. [https://vulncat.fortify.com/en/detail?id=desc.structural.java.dead\\_code\\_empty\\_try\\_block](https://vulncat.fortify.com/en/detail?id=desc.structural.java.dead_code_empty_try_block)