1.1.1 Finally Blok İçerisinde Return İfadesi Kullanılması (Return Inside Finally Block) (CWE-584)

Açıklık Önem Derecesi: Düşük

Açıklığın Etkisi: Dayanıklılık eksikliği

Açıklığın Barındıran Dosyalar/Satırlar:

Proje Dosyası/Dosya Adı	Satır Numarası

Açıklığın Açıklaması:

“Try-Finally” blok kullanımlarında - yani catch’in yer almadığı kullanımlarda - try bloğunda bir istisna fırladığında önce finally bloğuna gidilir, ardından istisna bir üst bloğa fırlar veya bu kullanımlarda try bloğunda bir istisna fırlamadığında yine önce finally bloğuna gidilir, ardından program akışı finally’den sonraki satır ile devam eder. “Try-Finally” blok kullanımlarında eğer finally bloğunda return ifadesi kullanılırsa try’da istisna fırlama satırı olsa dahi her ne şekilde olursa olsun try’dan sonra finally bloğuna gidildiğinden finally bloğundaki return ile dönüş olacaktır. Dolayısıyla return ile bir üst bloğa geçildiğinden try bloğundaki istisna üst bloğa fırlamamış olacaktır. Böylece bir istisna ele alınamamış / kaybolmuş olacaktır.

“Try-Catch-Finally” blok kullanımlarında ise try bloğunda bir istisna fırladığında ve istisna catch’te tanımlıysa catch bloğuna önce gidilir, ardından finally bloğuna gidilir. Eğer try bloğunda fırlayan istisna catch’de tanımlı değilse try’dan sonra doğrudan finally bloğuna gidilir ve finally bloğu tamamlandıktan sonra try’daki istisna bir üst bloğa fırlar. Bu durumda eğer finally bloğunda return ifadesi yer alırsa try’da fırlayan istisna catch bloğu pas geçilip finally’de return ile dönüş yapılacağından bir üst bloğa fırlamayacaktır ve kaybolmuş olacaktır.

“Try-Catch-Finally” blok kullanımlarında catch bloğunda bir istisna fırladığında ise önce finally bloğuna gidilir. finally bloğu tamamlandıktan sonra catch’deki istisna bir üst bloğa fırlar. Bu durumda eğer finally bloğunda return ifadesi yer alırsa catch’de fırlayan istisna finally’de return ile dönüş yapılacağından bir üst bloğa fırlamayacaktır ve kaybolmuş olacaktır.

“Try-Finally” kullanımlarında try’da fırlayan istisna veya “try-catch-finally” kullanımlarında try veya catch’de fırlayan istisnalar bir üst bloğa eğer finally bloğunda return yer alırsa fırlayamazlar. Bu ise bir istisnanın kaybı demektir ve hata kütüğünün elde edilememesi anlamına gelir. Dolayısıyla bu durum güvenliği etkiler.

Not:

Ayrıca eğer try veya catch bloğunda return ifadesi yer alırsa finally'de return'e yer verilmesi try / catch'in return'lerini baskınlar ve try-catch-finally'de dönüş finally'inin return'ü ile olur.

Bu açıklığı somutlaştırmak için JAVA dilinde bu istisna kaybı işleyişini gösteren P.O.C. örneğine yer verilmiştir:

JAVA - P.O.C:

```java
public class MagicTrick {

        public static class MagicException extends Exception { }

        public static void main(String[] args) {

                System.out.println("Bu büyülü kodun bir istisnayı " +
                                   "gözleriniz önünde yok etmesini " +
                                   "izleyin!");

                System.out.println("İlk olarak kullandığınız istisna" +
                                   "yönetimi:");
                try {

                        doMagic(false);

                } catch (MagicException e) {

                        // An exception will be caught here
                        e.printStackTrace();

                }

                System.out.println("Şimdi ise sihir vakti:");

                try {

                        doMagic(true);

                } catch (MagicException e) {

                        // No exception caught here, the finally block ate it
                        e.printStackTrace();

                }

                System.out.println("Tada!");
        }
```

```
        public static void doMagic(boolean returnFromFinally) throws
MagicException {

                try {

                        throw new MagicException();

                }
                finally {

                        if (returnFromFinally) {
                                return;
                        }

                }
        }
}
```

Çıktı:

Bu büyülü kodun bir istisnayı gözleriniz önünde yok etmesini izleyin!

İlk olarak kullandığınız istisna yönetimi:

… // >>> printStackTrace İstisna Hata Çıktısı Gelir

Şimdi ise sihir vakti:

// >>> printStackTrace İstisna Hata Çıktısı Gelmez

Tada!

P.O.C.'de gösterildiği gibi JAVA uygulamasında main() metodunda ilk olarak konsola "Bu büyülü kodun bir istisnayı gözleriniz önünde yok etmesini izleyin!" çıktısı ve sonra "İlk olarak kullandığınız istisna yönetimi:" çıktısı gelir. Ardından iki adet try-catch bloğu kullanımı gelir. İlk try-catch bloğu kullanımında try bloğunda doMagic(false) metodu çağırılır. Bu metodun içerisinde try-finally bloğu vardır. Bu try-finally bloğunda try'da bir istisna fırlatılır. İstisna bir üst bloğa fırlamadan önce finally'e girilir. doMagic metodu false argümanı ile çağırıldığından metot içerisindeki finally bloğunda yer alan if koşuluna girilmez ve return ifadesi çalışmadığından istisna bir üst bloğa (main() metoduna) sorunsuzca fırlar. İstisna bir üst bloğa (main() metodunda) fırladıktan sonra main() metodundaki ilk try-catch bloğunda catch try'da fırlayan istisnayı yakalar ve konsola istisna hata kaydını basar. Yani doMagic() metodundan fırlayan istisna hata kaydı sorunsuzca alınmış olur. Ardından main() metodundaki "Şimdi ise sihir

vakti:” çıktısı konsola yansır. Daha sonra main() metodundaki ikinci try-catch bloğuna gelinir. İkinci try-catch bloğu kullanımında try bloğunda doMagic(true) metodu çağırılır. Bu metodun içerisinde try-finally bloğu vardır. try-finally bloğunda try’da istisna fırlatılır. İstisna bir üst bloğa fırlamadan önce finally’e girilir. doMagic() metodu true argümanı ile çağırıldığından metot içerisindeki finally bloğunda yer alan if koşuluna bu sefer girilir ve if koşulunda return ifadesi yer aldığından return ile dönüş yapılır. doMagic() metodundaki try’da oluşan istisna bir üst bloğa (main() metoduna) bu nedenle fırlamaz. Return ile doMagic() metodundan main() metoduna dönüldükten sonra main() metodundaki ikinci try-catch bloğunda try’a istisna gelmeyeceğinden catch ile istisna kaydı hatası ekrana basılamaz. Son olarak konsola “Tada!” çıktısı yansır. Sonuç olarak doMagic() metodunda finally’de kullanılan return ifadesi dolayısıyla main() metodunda bir istisna kaydı kaybedilmiş olur.

Bu açıklığı somutlaştırmak için JAVA dilinde sade bir ilave örneğe de yer verilmiştir:

Java - Güvensiz Hal:

```
try {

        // ...
        throw IllegalArgumentException();

}
finally {

        return r;

}
```

Java - Güvenli Hal:

```
try {

        // ...
        throw IllegalArgumentException();

}
finally {

        // ...

}
```

“Java - Güvensiz Hal” kod bloğunda try bloğunda bir istisna fırlatılmaktadır. Ardından finally bloğunda return ifadesi yer almaktadır. Try’da fırlayan istisna bir üst bloğa fırlamadan önce finally’ye girilecektir. finally’de return ifadesi yer aldığından ise return ile dönüş yapılacaktır ve istisna fırlaması kaybolacaktır. “Java - Güvenli Hal” kod bloğunda ise try bloğunda yine bir istisna fırlatılmaktadır. Ardından finally bloğunda return yer almamaktadır. Try’da fırlayan istisna bir üst bloğa fırlamadan önce finally’ye girilecektir. Finally’de işlemler tamamlandıktan sonra istisna bir üst bloğa fırlayacaktır. Böylece istisna kaybolmamış olacaktır.

Finally blok’ları içerisinde return ifadesi kullanmak try’da veya catch’de fırlayacak istisnaların (exception’ların) yakalanamamasına / kaybolmasına sebep olmaktadır. Böylesi durumlar “Finally Blok İçerisinde Return İfadesi Kullanılması (CWE-584)” açıklığı olarak ele alınmaktadır. Kurum uygulamada bu açıklık tespit edilmiştir:

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

Açıklığın Önlemi:

- Finally blokları içerisinde asla return ifadesi kullanılmamalıdır.
- Finally bloğu bunun yerine temizlik (cleanup) kodlarına sahip olmalıdır.

Referanslar:

1. https://cwe.mitre.org/data/definitions/584.html
2. https://vulncat.fortify.com/en/detail?id=desc.structural.java.poor_error_handling_return_inside_finally#Java%2FJSP
3. https://rules.sonarsource.com/java/RSPEC-1143/
4. https://www.appmarq.com/public/tqi,1020068,Avoid-return-statement-in-finally-block
5. https://stackoverflow.com/questions/15225819/try-catch-finally-return-clarification
6. https://www.scientecheasy.com/2020/09/return-statement-in-try-catch-finally-block.html/