

1.1.1 Döngü Koşulunda Kontrol Edilmemiş Girdi (Unchecked Input for Loop Condition) (CWE-606)

Açıklık Önem Derecesi: Orta

Açıklığın Etkisi: Servis dışı kalma, uzaktan komut çalıştırma

Açıklığın Barındıran Dosyalar/Satırlar:

Proje Dosyası/Dosya Adı	Satır Numarası

Açıklığın Açıklaması:

Uygulamalar tekrarlı işlemleri uygulamak için for ve while gibi döngüler kullanırlar. Bu döngülerin sınırları dinamik girdiler ile belirleniyorsa (örn; kullanıcı girdisi ile veya uygulamanın üzerinde koştuğu sistemdeki bir kaynak ile) ve dinamik girdiler denetlenmeden döngüde kullanılıyorsa bu durumda "Unchecked Input for Loop Condition (CWE-606)" açıklığı var denir. Bu açıklık ile saldırganlar oldukça yüksek dinamik girdi değeri girerek döngüde takılı kalınmasına ve diğer işlemlere devam edilememesine sebep olabilirler.

Not:

Off By One Error (CWE-193) ile Unchecked Input for Loop Condition (CWE-606) açıklığı benzerdirler. İkisi de uygulamalardaki döngülerin beklenenden fazla çalışmasını ele alırlar. Fakat Off By One Error 'da geliştirici tarafından sağlanan yanlış / aşırı döngü limit değerleri söz konusu iken Unchecked Input for Loop Condition'da saldırgan tarafından sağlanan yanlış / aşırı döngü limit değerleri söz konusudur.

Bu açıklığı somutlaştırmak için çeşitli programlama dillerinde örneklere yer verilmiştir:

Java - Güvensiz Kod Bloğu:

```
// ENG: Loop Condition Is Not Bounded By Any Value

protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {

    int loopCount = 0;
    try{
        loopCount = Integer.parseInt(request.getParameter("loopCount"));
    } catch(NumberFormatException e){
        return DEFAULT_VAL;
    }
    for(int i=0; i < loopCount; i++){
        //Do Something
    }
}
```

Java - Güvenli Kod Bloğu:

```
// Loop Condition is Bounded With MAX_LOOPS

protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {

    int loopCount = 0;
    try{
        loopCount = Integer.parseInt(request.getParameter("loopCount"));
    } catch(NumberFormatException e){
        return DEFAULT_VAL;
    }
    if(loopCount > MAX_LOOPS){
        loopCount = MAX_LOOPS;
    }
    for(int i=0; i < loopCount; i++){
        //Do Something
    }
}
```

Güvensiz java kod bloğunda istemci taraftan gelen bir dinamik girdi for döngüsünün üst limiti olarak tayin edilmiştir. Denetleme olmadığından bu üst limit aşırı fazla girilerek döngüde takılı kalınmasına neden olabilir.

Güvenli java kod bloğunda ise istemci taraftan gelen bir dinamik girdi belirli bir maksimum değer ile kıyaslanıp büyükse maksimum değer kadar, değilse girdinin orijinal değeri kadar kalacak şekilde ayarlanmıştır. Bu şekilde for döngüsünde üst limit sınırlandırılmış olur. Bu sınırlandırma güvenli kodlamadır.

PHP - Güvensiz Kod Bloğu:

```
<?php

// Loop Condition Is Not Bounded By Any Value

$loopCount = $_GET['loop_count'];

for( $i = 0; $i < $loopCount; $i++ ) {
    // Do something...
}
```

PHP - Güvenli Örnek (I):

```
<?php

// Loop Condition is Bounded With a Hardcoded Value

$loopCount = $_GET['loop_count'];
$max_loops = 50;

if ($loopCount < $max_loops) {
    for( $i = 0; $i < $loopCount; $i++ ) {
        // Do something...
    }
}
```

PHP - Güvenli Örnek (II):

```
<?php

// Range Function Output is Bounded With a Hardcoded Value

$loopCount = $_GET['loop_count'];
$max_loops = 50;

if ($loopCount > -$max_loops and $loopCount < $max_loops) {
    foreach(range(0, $loopCount) as $i) {
        // Do something...
    }
}
```

PHP güvensiz kod bloğu örneği de aynı java da olduğu gibi istemci taraftan gelen bir dinamik girdi for döngüsünün üst limiti olarak tayin edilmiştir. Denetleme olmadığından bu üst limit aşırı fazla girilerek döngüde takılı kalınmasına neden olabilir.

PHP güvenli kod bloğu (I)'de ise istemci taraftan gelen bir dinamik girdi belirli bir maksimum değer ile kıyaslanıp büyükse for döngüsüne hiç girilmemektedir, küçükse for döngüsüne girilmektedir. Bu kontrol güvenli kodlamadır.

PHP güvenli kod bloğu (II)'de ise istemci taraftan gelen bir dinamik girdi yine belirli bir maksimum değer ile kıyaslanıp büyükse foreach döngüsüne hiç girilmeyecek şekilde, küçükse foreach döngüsüne girilecek şekilde tasarlanmıştır. Bu kontrol güvenli kodlamadır.

Python - Güvensiz Kod Bloğu:

```
# Loop Condition Is Not Bounded By Any Value

@app.route('/fib')
def fib():
    result = [0,1]
    limit = int(request.args.get('limit'))
    for i in range(0, limit):
        result.append(result[i] + result[-1])
    return ','.join(map(str, result))
```

Python - Güvenli Kod Bloğu:

```
# Loop Condition is Bounded With MAX_ITERATION

@app.route('/fib')
def fib():
    result = [0,1]
    limit = int(request.args.get('limit'))
    if limit > MAX_ITERATION:
        limit = MAX_ITERATION
    for i in range(0, limit):
        result.append(result[i] + result[-1])
    return ','.join(map(str, result))
```

Güvensiz python kod bloğunda istemci taraftan gelen bir dinamik girdi for döngüsünün üst limiti olarak tayin edilmiştir. Denetleme olmadığından bu üst limit aşırı fazla girilerek döngüde takılı kalınmasına neden olabilir.

Güvenli python kod bloğunda ise istemci taraftan gelen bir dinamik girdi belirli bir maksimum değer ile kıyaslanıp büyükse maksimum değer kadar, değilse girdinin orijinal değeri kadar

kalacak şekilde ayarlanmıştır. Bu şekilde for döngüsünde üst limit sınırlandırılmış olur. Bu sınırlandırma güvenli kodlamadır.

Kurum uygulamada Unchecked Input for Loop Condition (CWE-606) açıklığı tespit edilmiştir.

.....BULGU.....

Açıklığın Önlemi:

Bu açıklık hakkında verilebilecek tavsiyeler şu şekildedir:

- İdealde dinamik girdilerin bir döngüde iterasyon sayısı belirlerken kullanılması tavsiye edilmemektedir. Bu kullanımdan mümkünse kaçınılmalıdır.
- Fakat bu kullanım gerçekten gerekliyse dinamik girdi önce girdi doğrulamaya tabi tutulmalıdır ve ardından döngüde iterasyon sayısı belirlemede kullanılmalıdır.

Referanslar:

1. <http://cwe.mitre.org/data/definitions/193.html>
2. <https://cwe.mitre.org/data/definitions/606.html>