CVE-2023-40745

Alessio Esposito Inchiostro
Trabajo final de Explotaciòn de vulnerabilidades en sistemas software - 62240

CVE-2023-40745

LibTIFF is vulnerable to an **integer overflow**. This flaw allows remote attackers to cause a denial of service (application crash) or possibly **execute an arbitrary code via a crafted tiff image**, which triggers a **heap-based buffer overflow**.

Caracterización de la vulnerabilidad

Tipo de vulnerabilidad

- Integer wraparound/overflow causes heap buffer overflow -> DoS or out of bound write
- Write-what-where, similar a las vulnerabilidades vistas en la Pràctica 3

Mètricas

CVSS 3.x Severity and Vector Strings:

CNA: Red Hat, Inc. **Base Score:** 6.5 MEDIUM **Vector:** CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:N/A:H

- Nivel de riesgo mediano (**Base score 6.5**)
- **Impact score** bajo (**3.6**) → DoS muy facil, casi imposible el contròl de flujo de ejecuciòn
- **Exploitability score** alto (**2.8**) → Vector red, pero necesita interacciòn de usuario

Mètricas - II

CVSS 3.x Severity and Vector Strings:

CNA: Red Hat, Inc. **Base Score:** 6.5 MEDIUM **Vector:** CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:N/A:H

- Vector de ataque: network, el màs fácil que utilizar (ej: online TIFF converter)
- Necesaria interacciòn de usuario pero no permisos
- Ataque DOS (Availability), por eso de riesgo medio/bajo

Fuentes de las informaciones

- NVD
- Repository GitLab
- Issue con POC y commit de fix

Descripción técnica

Adversary model

- (Para la ejecuciòn de còdigo arbitrario) ASLR no está activado en el sistema objetivo, o existe una forma de eludirlo
- (Para la ejecuciòn de còdigo arbitrario) Las tècnicas de stack protection no están activadas en el sistema objetivo, o existe una forma de eludirlas
- El atacante es capaz de crear ficheros .tiff con payload dañino y pasar dichos ficheros al ejecutable con la vulnerabilidad

Funcionamiento de la vulnerabilidad

- Write-what-where
- Todas las versiones antes de la 4.6.0 son vulnerables
- cpSeparateBufToContigBuf es similar a un memcpy
- iskew puede ser negativo si *tilew*spp* es bastante grande
- Esto causa un integer overflow que causa buffer overflow

```
1.  iskew = imagew - tilew * spp;
2.  if (colb + tilew * spp > imagew) {
3.            uint32_t width = imagew - colb;
4.            int oskew = tilew * spp - width;
5.            cpSeparateBufToContigBuf(bufp + colb + s * bytes_per_sample, tilebuf,
6.                                     nrow, width / (spp * bytes_per_sample),
7.                                     oskew + iskew, oskew / spp, spp,
8.                                     bytes_per_sample);
9.          } else
10.           cpSeparateBufToContigBuf(bufp + colb + s * bytes_per_sample, tilebuf,
11.                                    nrow, tw, iskew, 0, spp, bytes_per_sample);
```

```
uint32_t imagew = TIFFRasterScanlineSize(in);
uint32_t tilew = TIFFTileRowSize(in);
```

Estos dos datos se leen directamente desde el header de la imagen TIFF; que puede ser modificado por ser dañino

```c
static void cpSeparateBufToContigBuf(uint8_t *out, uint8_t *in, uint32_t rows,
                                     uint32_t cols, int outskew, int inskew,
                                     tsample_t spp, int bytes_per_sample)
{
    while (rows-- > 0)
    {
        uint32_t j = cols;
        while (j-- > 0)
        {
            int n = bytes_per_sample;

            while (n--)
            {
                *out++ = *in++;
            }
            out += (spp - 1) * bytes_per_sample;
        }
        out += outskew;
        in += inskew;
    }
}
```

Fix de la vulnerabilidad

Merged **Fix for ticket #591** `ArieHaenel/libtiff:maste…` into `master`

Overview 0 | Commits 1 | Pipelines 1 | **Changes 1**

tools/tiffcp.c

```
1756 1756        }
     1757  +
     1758  +     if ( (imagew - tilew * spp) > INT_MAX ){
     1759  +         TIFFError(TIFFFileName(in),
     1760  +                   "Error, image raster scan line size is too large");
     1761  +         return 0;
     1762  +     }
     1763  +
1757 1764        iskew = imagew - tilew * spp;
1758 1765        tilebuf = limitMalloc(tilesize);
1759 1766        if (tilebuf == 0)
```

Otras contramedidas

La vulnerabilidad no se considera muy peligrosa porque sì puede causar DoS, pero necesita màs prerrequisitos para alcanzar una toma de control de flujo de ejecuciòn.

Otras contramedidas - II

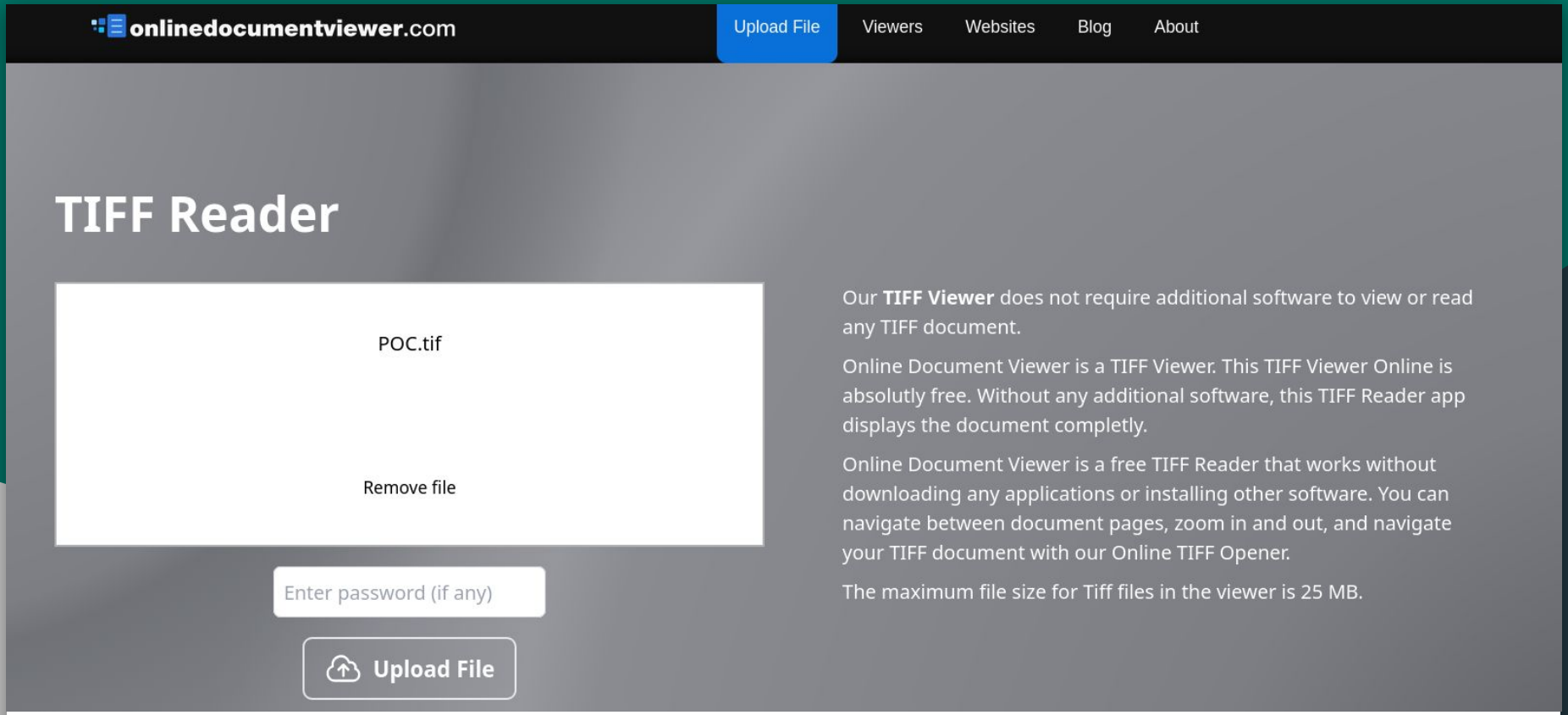
- Canarios de pila: Detectan el stack smashing e impiden que se tome el control de **%eip**.
- ASLR: Si de alguna manera se logra tomar el control de **%eip**, puede dificultar la utilización de otras funciones del ejecutable al desorganizar las direcciones de memoria.
- Bit NX: Impide que se inyecte y ejecute código malicioso en la región de memoria donde la vulnerabilidad causa la escritura.

Proof of concept

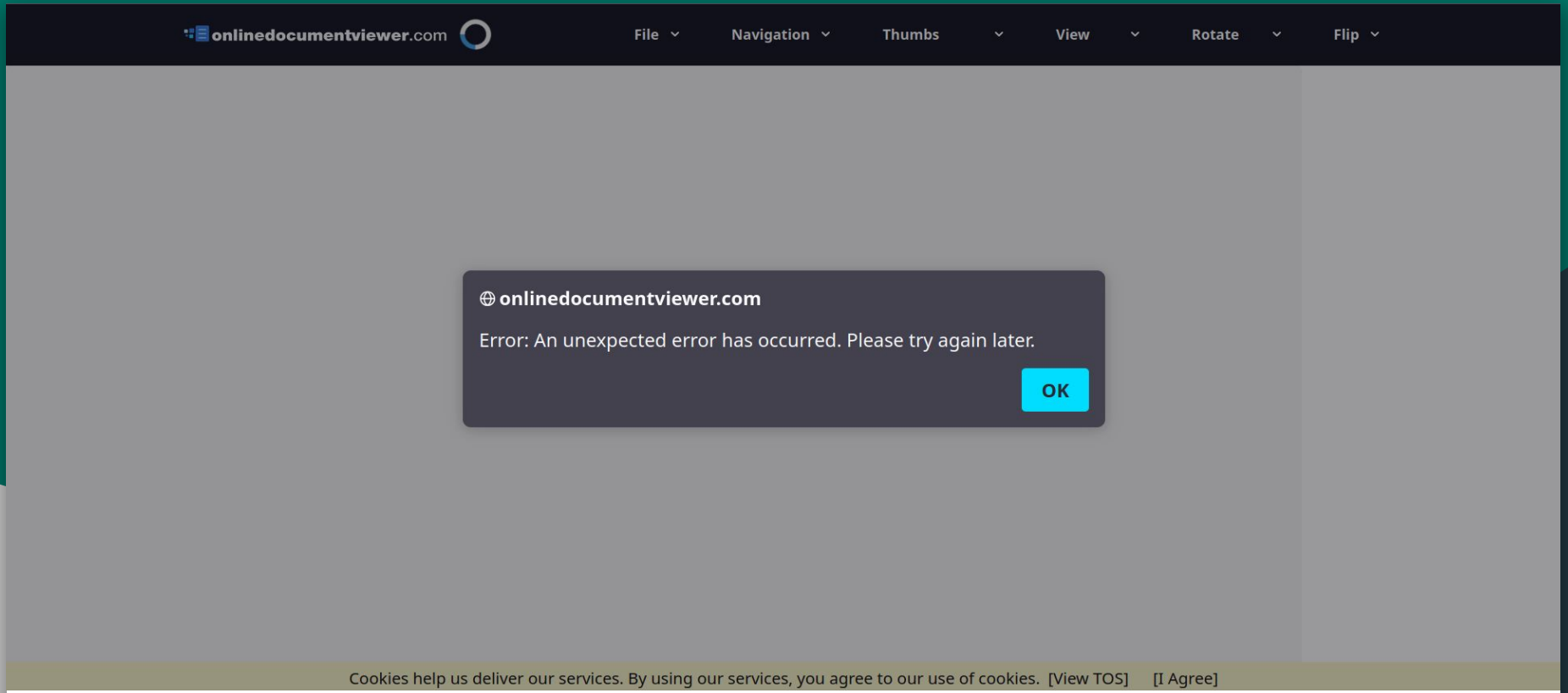
(Demostración práctica)

```
conte@scrappy:~/Downloads$ file POC.tif
POC.tif: TIFF image data, little-endian, direntries=17, height=2, bps=56266,
n=upper-left, width=10256
```

Se crea un fichero tiff malicioso con header que describen una imagen de tamaño muy grande



Se pasa la imagen dañina a un servicio online o a otra persona



El intentar leer o copiar la imagen causa un error






Preguntas y dudas