CVE-2016-3076

描述：Heap-based buffer overflow in the j2k\_encode\_entry function in Pillow 2.5.0 through 3.1.1 allows remote attackers to cause a denial of service (memory corruption) via a crafted Jpeg2000 file.

软件：Pillow-2.6.x

源码：Pillow/libImaging/Jpeg2KEncode.c

出现位置： 476行

关键源码：

state->buffer = malloc (tile\_width \* tile\_height \* components \* prec / 8);

pack(im, state->buffer, pixx, pixy, pixw, pixh);

Note：This vulnerability requires a particular value for height \* width such that height \* width \* components \* precision overflows, at which point the malloc will be for a smaller value than expected. The buffer that is allocated will be ((height \* width \* components \* precision) mod (2^31) / 8), where components is 1-4 and precision is either 8 or 16. Common values would be 4 components at precision 8 for a standard RGBA image.

内存循环复制：

typedef void (\*j2k\_pack\_tile\_t)(Imaging im, UINT8 \*buf,

unsigned x0, unsigned y0,

unsigned w, unsigned h);

static void

j2k\_pack\_l(Imaging im, UINT8 \*buf,

unsigned x0, unsigned y0, unsigned w, unsigned h)

{

UINT8 \*ptr = buf;

unsigned x,y;

for (y = 0; y < h; ++y) {

UINT8 \*data = (UINT8 \*)(im->image[y + y0] + x0);

for (x = 0; x < w; ++x)

\*ptr++ = \*data++;

}

}

static void

j2k\_pack\_i16(Imaging im, UINT8 \*buf,

unsigned x0, unsigned y0, unsigned w, unsigned h)

{

UINT8 \*ptr = buf;

unsigned x,y;

for (y = 0; y < h; ++y) {

UINT8 \*data = (UINT8 \*)(im->image[y + y0] + x0);

for (x = 0; x < w; ++x) {

\*ptr++ = \*data++;

\*ptr++ = \*data++;

}

}

}

static void

j2k\_pack\_la(Imaging im, UINT8 \*buf,

unsigned x0, unsigned y0, unsigned w, unsigned h)

{

UINT8 \*ptr = buf;

UINT8 \*ptra = buf + w \* h;

unsigned x,y;

for (y = 0; y < h; ++y) {

UINT8 \*data = (UINT8 \*)(im->image[y + y0] + 4 \* x0);

for (x = 0; x < w; ++x) {

\*ptr++ = data[0];

\*ptra++ = data[3];

data += 4;

}

}

}

static void

j2k\_pack\_rgb(Imaging im, UINT8 \*buf,

unsigned x0, unsigned y0, unsigned w, unsigned h)

{

UINT8 \*pr = buf;

UINT8 \*pg = pr + w \* h;

UINT8 \*pb = pg + w \* h;

unsigned x,y;

for (y = 0; y < h; ++y) {

UINT8 \*data = (UINT8 \*)(im->image[y + y0] + 4 \* x0);

for (x = 0; x < w; ++x) {

\*pr++ = data[0];

\*pg++ = data[1];

\*pb++ = data[2];

data += 4;

}

}

}

static void

j2k\_pack\_rgba(Imaging im, UINT8 \*buf,

unsigned x0, unsigned y0, unsigned w, unsigned h)

{

UINT8 \*pr = buf;

UINT8 \*pg = pr + w \* h;

UINT8 \*pb = pg + w \* h;

UINT8 \*pa = pb + w \* h;

unsigned x,y;

for (y = 0; y < h; ++y) {

UINT8 \*data = (UINT8 \*)(im->image[y + y0] + 4 \* x0);

for (x = 0; x < w; ++x) {

\*pr++ = \*data++;

\*pg++ = \*data++;

\*pb++ = \*data++;

\*pa++ = \*data++;

}

}

}