XV multiple buffer overflows (update)

Von: Greg Roelofs <newt@pobox.com>
An: bugtraq@securityfocus.com
Datum: 11.04.2005 18:21

XV is a Unix/X11-based image viewer/converter with some editing capabilities. It has been distributed by John H. Bradley and the University of Pennsylvania as (shared-source) shareware for the last 15 years or so. Primary development appears to have ceased as of early 1995, and all forms of maintenance seem to have ended in 2000 or 2001; no part of the XV web site (trilon.com/xv) has been updated since then, as far as I can determine. The author does not respond to e-mail.

Last August, several input-validation vulnerabilities were described on this list by "infamous41md / sean," together with an exploit for one of them:

http://www.securityfocus.com/archive/1/372345

Various vendors, including Gentoo, SuSE, and OpenBSD, released patches addressing the problems, but the patches were incomplete. For example, the SuSE/Gentoo patch included this fragment (from http://bugs.gentoo.org/show_bug.cgi?id=61619):

```
--- xvpcx.c
+++ xvpcx.c    Tue Aug 24 13:12:15 2004
@@ -222,7 +222,14 @@
  byte *image;

  /* note:  overallocation to make life easier... */
-  image = (byte *) malloc((size_t) (pinfo->h + 1) * pinfo->w + 16);
+  int count = (pinfo->h + 1) * pinfo->w + 16;
+
+  if (count <= 0 || pinfo->h <= 0 || pinfo->w <= 0) {
+    pcxError(fname, "Bogus PCX file!!");
+    return (0);
+  }
+
+  image = (byte *) malloc((size_t) count);
  if (!image) FatalError("Can't alloc 'image' in pcxLoadImage8()");

  xvbzero((char *) image, (size_t) ((pinfo->h+1) * pinfo->w + 16));
```

(This is within the 8-bit code.) Because of the additive factors, count can be as large as 4295032848 == 65552 on machines with 32-bit integers; obviously that's positive. Setting the height to 65536 and the width to 65535 requires only 15 bytes of "fill" before the heap-overflowing exploit code can presumably begin.

The more general case is in the 24-bit code, and it affects almost all of the other 24-bit (RGB) formats, too:

```
@@ -250,17 +257,25 @@
 {
   byte *pix, *pic24, scale[256];
   int   c, i, j, w, h, maxv, cnt, planes, bperlin, nbytes;
+  int count;

   w = pinfo->w;  h = pinfo->h;

   planes = (int) hdr[PCX_PLANES];
   bperlin = hdr[PCX_BPRL] + ((int) hdr[PCX_BPRH]<<8);

+  count = w*h*planes;
+
+  if (count <= 0 || planes <= 0 || w <= 0 || h <= 0) {
+    pcxError(fname, "Bogus PCX file!!");
+    return (0);
+  }
+
   /* allocate 24-bit image */
-  pic24 = (byte *) malloc((size_t) w*h*planes);
+  pic24 = (byte *) malloc((size_t) count);
   if (!pic24) FatalError("couldn't malloc 'pic24'");

-  xvbzero((char *) pic24, (size_t) w*h*planes);
+  xvbzero((char *) pic24, (size_t) count);

   maxv = 0;
   pix = pinfo->pic = pic24;
```

In principle, "planes" can be as large as 255, but this function isn't reached unless it's exactly equal to 3. That's more than enough when w and h can each be as large as 65536, however. For formats that support 32-bit width and height values, even 1-bit images can cause wraparound to positive integers (which is what's not addressed in the existing patches).

In general, the fix is to use additional variables to hold the intermediate results of pairwise multiplication and to test that the expected inverse arithmetic operations hold for the results. Thus, for example, instead of malloc'ing a three-way product directly:

```
foo = (char *) malloc((size_t) w*h*3);          // int w, h;
```

....do something like this:

```
int npixels, bufsize;

npixels = w * h;
bufsize = 3 * npixels;
if (w <= 0 || h <= 0 || npixels/w != h || bufsize/3 != npixels) {
   FAIL();
}

foo = (char *) malloc((size_t) bufsize);
```

I have incorporated such fixes into all affected XV image decoders and updated my "jumbo patches" accordingly:

http://pobox.com/~newt/greg_xv.html

Since security fixes were/are not my primary motivation in making the patches, they include many other things as well; I apologize for that, but I don't have time to split things up and verify that everything still works as pieces.

Bruno Rohee has tested several other image-manipulating applications and found some of them to be affected, as well (though not necessarily in an exploitable way):

GwenView (Unix/KDE)
IrfanView (Win32)
ImageMagick (various)

He also found that the GIMP and Imlib-based viewers appear NOT to be affected.

CERT has assigned ID VU#622622 to this vulnerability. I have not received any notice of a CVE identifier.

Regards,

--

Greg Roelofs newt pobox com http://pobox.com/~newt/
Newtware, PNG Group, AlphaWorld Map, etc.