The bug is hidden in ImageX11.cpp, in function ImageDraw::operator Image(). At some point it asks X server for an image and the call fails returning NULL (at line 348). Later this NULL stored in pointer xim is accessed which inevitably leads to a crash.

Quick fix (i.e. removing the problem without really understanding what why the XGetImage fails):

```
ImageDraw::operator Image() const
{
 // ... a lot of code omitted here ...
 if(has_alpha) {
  xim = XGetImage(Xdisplay, alpha.dw, 0, 0, size.cx, size.cy, AllPlanes, ZPixmap);
  if(xim){   // added this check
   const byte *s = (const byte *)xim->data;
   t = ib;
   Buffer<RGBA> line(size.cx);
   for(int y = 0; y < size.cy; y++) {
    fmt.Read(line, s, size.cx, palette);
    for(int x = 0; x < size.cx; x++)
     (t++)->a = line[x].r;
    s += xim->bytes_per_line;
   }
   XDestroyImage(xim);
  }
 }
 Premultiply(ib);
 return ib;
}
```

As this fix bypasses the alpha processing (when the XGetImage fails), I would expect to see some visual artifacts, but I actually didn't notice anything. Maybe I was not looking hard enough or the effect is too subtle to see in the quick animation.

Also I have no idea why it makes problems in theide but I never seen it in any other app...

Best regards,
Honza