libextractor

GNU libextractor

Log | Files | Refs | Submodules | README | LICENSE

---

commit 6edfa653c048800e24a17f7e8cc2bb42659b8d01
parent 4478a2595f38996b969dc89c5df29e3a93e9c239
Author: Christian Grothoff <christian@grothoff.org>
Date: Tue, 30 Jun 2026 17:46:39 +0200

misc minor bugfixes

Diffstat:

M	ChangeLog	12	++++++++++-
M	src/main/extractor_datasource.c	42	++++++++++++++++++++++++++++++----------
M	src/main/extractor_ipc_w32.c	15	+++++++++------
M	src/main/extractor_plugin_main.c	5	+++--
M	src/main/extractor_plugins.c	16	++++++++++++----
M	src/main/extractor_plugpath.c	13	++++++++++++-
M	src/plugins/deb_extractor.c	30	++++++++++++++++++++++++--------
M	src/plugins/gif_extractor.c	19	+++++++++----------
M	src/plugins/nsfe_extractor.c	2	+-
M	src/plugins/ole2_extractor.c	25	++++++++++++++++++--------
M	src/plugins/pack.c	2	+-
M	src/plugins/qt_extractor.c	7	++++---
M	src/plugins/real_extractor.c	12	++++++------

13 files changed, 136 insertions(+), 64 deletions(-)

---

diff --git a/ChangeLog b/ChangeLog
@@ -1,6 +1,16 @@
Tue Jun 30 05:13:30 PM CEST 2026
Fix tIME chunk parser in PNG plugin.
-
+ Fix mime-type detection in QT plugin.
+ Fix wrong constant used in REAL plugin.
+ Fix multi-chunked decompression in DEB plugin.
+ Fix memory leak in GIF plugin (on error path).
+ Fix error handling in W32 IPC logic.
+ Fix some bit-flags set incorrectly for gzip-compressed headers.
+ Security fix to not honor LIBEXTRACTOR_PREFIX in case we
+ are linked into SUID binary and are running as root (which
+ is anyway a bad idea, but we can be more conservative).
+ Thanks to Haitam Lazaar for reporting.
+ Releasing GNU libextractor 1.16. -CG

Mon Jun 29 07:58:39 PM CEST 2026
Fix potential 4 MB on-stack memory allocation that could
diff --git a/src/main/extractor_datasource.c b/src/main/extractor_datasource.c
@@ -447,6 +447,12 @@ bfds_read (struct BufferedFileDataSource *bfds,

#if HAVE_ZLIB
+
+#define FHCRC 0x02
+#define FEXTRA 0x04

```
+#define FNAME 0x08
+#define FCOMMENT 0x10
+
 /**
  * Initializes gz-decompression object. Might report metadata about
  * compresse stream, if available. Resets the stream to the beginning.
@@ -474,10 +480,10 @@ cfs_init_decompressor_zlib (struct CompressedFileSource *cfs,
   if ( (-1 == rsize) ||
        (sizeof (hdata) > (size_t) rsize) )
     return -1;
-  if (0 != (hdata[3] & 0x4)) /* FEXTRA  set */
+  if (0 != (hdata[3] & FEXTRA))
     gzip_header_length += 2 + (hdata[10] & 0xff) + ((hdata[11] & 0xff) * 256);

-  if (0 != (hdata[3] & 0x8))
+  if (0 != (hdata[3] & FNAME))
   {
     /* FNAME set */
     char fname[1024];
@@ -512,7 +518,7 @@ cfs_init_decompressor_zlib (struct CompressedFileSource *cfs,
     gzip_header_length += len + 1;
   }

-  if (0 != (hdata[3] & 0x16))
+  if (0 != (hdata[3] & FCOMMENT))
   {
     /* FCOMMENT set */
     char fcomment[1024];
@@ -546,7 +552,7 @@ cfs_init_decompressor_zlib (struct CompressedFileSource *cfs,
       return 0; /* done */
     gzip_header_length += len + 1;
   }
-  if (0 != (hdata[3] & 0x2)) /* FCHRC set */
+  if (0 != (hdata[3] & FHCRC))
     gzip_header_length += 2;
   memset (&cfs->strm, 0, sizeof (z_stream));

@@ -569,13 +575,15 @@ cfs_init_decompressor_zlib (struct CompressedFileSource *cfs,
    *
    * ZLIB_VERNUM isn't defined by zlib version 1.1.4 ;
    * there might be a better check.
-   */if (Z_OK != inflateInit2 (&cfs->strm,
+   */
+  if (Z_OK !=
+      inflateInit2 (&cfs->strm,
 #ifdef ZLIB_VERNUM
-                              15 + 32
+                    15 + 32
 #else
-                              -MAX_WBITS
+                    -MAX_WBITS
 #endif
-                              ))
+                    ))
   {
     LOG ("Failed to initialize zlib decompression\n");
     return -1;
@@ -599,17 +607,24 @@ cfs_init_decompressor_zlib (struct CompressedFileSource *cfs,
  */
 static int
 cfs_init_decompressor_bz2 (struct CompressedFileSource *cfs,
-                           EXTRACTOR_MetaDataProcessor proc, void *proc_cls)
+                           EXTRACTOR_MetaDataProcessor proc,
+                           void *proc_cls)
 {
```

```
    if (0 !=
-       bfds_seek (cfs->bfds, 0, SEEK_SET))
+       bfds_seek (cfs->bfds,
+                  0,
+                  SEEK_SET))
    {
      LOG ("Failed to seek to start to initialize BZ2 decompressor\n");
      return -1;
    }
-   memset (&cfs->bstrm, 0, sizeof (bz_stream));
+   memset (&cfs->bstrm,
+           0,
+           sizeof (bz_stream));
    if (BZ_OK !=
-       BZ2_bzDecompressInit (&cfs->bstrm, 0, 0))
+       BZ2_bzDecompressInit (&cfs->bstrm,
+                             0,
+                             0))
    {
      LOG ("Failed to initialize BZ2 decompressor\n");
      return -1;
@@ -633,7 +648,8 @@ cfs_init_decompressor_bz2 (struct CompressedFileSource *cfs,
   */
 static int
 cfs_init_decompressor (struct CompressedFileSource *cfs,
-                       EXTRACTOR_MetaDataProcessor proc, void *proc_cls)
+                       EXTRACTOR_MetaDataProcessor proc,
+                       void *proc_cls)
 {
   cfs->result_pos = 0;
   cfs->fpos = 0;
diff --git a/src/main/extractor_ipc_w32.c b/src/main/extractor_ipc_w32.c
@@ -776,6 +776,7 @@ EXTRACTOR_IPC_channel_recv_ (struct EXTRACTOR_Channel **channels,
     if (dwresult == WAIT_OBJECT_0)
     {
       int ret;
+
       if (channels[i]->mdata_size == channels[i]->size)
       {
         /* not enough space, need to grow allocation (if allowed) */
@@ -784,6 +785,7 @@ EXTRACTOR_IPC_channel_recv_ (struct EXTRACTOR_Channel **channels,
           LOG ("Inbound message from channel too large, aborting\n");
           EXTRACTOR_IPC_channel_destroy_ (channels[i]);
           channels[i] = NULL;
+          continue;
         }
         channels[i]->mdata_size *= 2;
         if (channels[i]->mdata_size > MAX_META_DATA)
@@ -794,6 +796,7 @@ EXTRACTOR_IPC_channel_recv_ (struct EXTRACTOR_Channel **channels,
           LOG_STRERROR ("realloc");
           EXTRACTOR_IPC_channel_destroy_ (channels[i]);
           channels[i] = NULL;
+          continue;
         }
         channels[i]->mdata = ndata;
       }
@@ -809,18 +812,18 @@ EXTRACTOR_IPC_channel_recv_ (struct EXTRACTOR_Channel **channels,
       if (! bresult || (-1 == ret))
       {
         DWORD error = GetLastError ();
+
         SetErrnoFromWinError (error);
         if (! bresult)
           LOG_STRERROR ("ReadFile");
         EXTRACTOR_IPC_channel_destroy_ (channels[i]);
```

```
          channels[i] = NULL;
+         continue;
        }
-       else
-       {
-         memmove (channels[i]->mdata, &channels[i]->mdata[ret],
-                  channels[i]->size + bytes_read - ret);
-         channels[i]->size = channels[i]->size + bytes_read - ret;
-       }
+       memmove (channels[i]->mdata,
+                &channels[i]->mdata[ret],
+                channels[i]->size + bytes_read - ret);
+       channels[i]->size = channels[i]->size + bytes_read - ret;
      }
    }
    return 1;
diff --git a/src/main/extractor_plugin_main.c b/src/main/extractor_plugin_main.c
@@ -646,8 +646,9 @@ EXTRACTOR_plugin_main_ (struct EXTRACTOR_PluginList *plugin,
      CloseHandle (pc.shm_id);
 #else
    if ( (NULL != pc.shm) &&
-        (((void*) 1) != pc.shm) )
-     munmap (pc.shm, pc.shm_map_size);
+        (MAP_FAILED != pc.shm) )
+     munmap (pc.shm,
+             pc.shm_map_size);
    if (-1 != pc.shm_id)
    {
      if (0 != close (pc.shm_id))
diff --git a/src/main/extractor_plugins.c b/src/main/extractor_plugins.c
@@ -71,12 +71,16 @@ get_symbol_with_prefix (void *lib_handle,
             template,
             sym);
    /* try without '_' first */
-  symbol = lt_dlsym (lib_handle, name + 1);
+  symbol = lt_dlsym (lib_handle,
+                     name + 1);
    if (NULL == symbol)
    {
      /* now try with the '_' */
-     char *first_error = strdup (lt_dlerror ());
-     symbol = lt_dlsym (lib_handle, name);
+     const char *dlerr = lt_dlerror ();
+     char *first_error = (NULL == dlerr) ? NULL : strdup (dlerr);
+
+     symbol = lt_dlsym (lib_handle,
+                        name);
      if (NULL == symbol)
      {
        LOG ("Resolving symbol `%s' failed, "
@@ -84,7 +88,11 @@ get_symbol_with_prefix (void *lib_handle,
             "`%s' and `%s'.\n",
             name + 1,
             name,
-            first_error == NULL ? "out of memory" : first_error,
+            (NULL == first_error)
+            ? ( (NULL == dlerr)
+                ? "no error from ltdl!?"
+                : "out of memory")
+            : first_error,
             lt_dlerror ());
      }
      if (NULL != first_error)
diff --git a/src/main/extractor_plugpath.c b/src/main/extractor_plugpath.c
@@ -29,6 +29,7 @@
```

```
  #include <sys/types.h>
  #include <signal.h>
  #include <ltdl.h>
+#include <stdbool.h>

  #include "extractor_plugpath.h"
  #include "extractor_logging.h"
@@ -466,9 +467,19 @@ get_installation_paths (EXTRACTOR_PathProcessor pp,
    char *prefix;
    char *d;
    char *saveptr;
+  bool skip; /* true if we should skip GETENV */

    prefix = NULL;
-  if (NULL != (p = getenv ("LIBEXTRACTOR_PREFIX")))
+#if _GNU_SOURCE
+  #define GETENV secure_getenv
+    skip = false;
+#else
+  #define GETENV getenv
+    skip = (0 == geteuid ());
+#endif
+
+  if ( (! skip) &&
+       (NULL != (p = GETENV ("LIBEXTRACTOR_PREFIX"))) )
    {
      if (NULL == (d = strdup (p)))
      {
diff --git a/src/plugins/deb_extractor.c b/src/plugins/deb_extractor.c
@@ -372,7 +372,9 @@ processControlTGZ (struct EXTRACTOR_ExtractContext *ec,
    off = 0;
    while (off < size)
    {
-    if (0 >= (sret = ec->read (ec->cls, &data, size - off)))
+    if (0 >= (sret = ec->read (ec->cls,
+                               &data,
+                               size - off)))
      {
        free (cdata);
        return 0;
@@ -382,7 +384,9 @@ processControlTGZ (struct EXTRACTOR_ExtractContext *ec,
              sret);
      off += sret;
    }
-  bufSize = cdata[size - 4] + (cdata[size - 3] << 8) + (cdata[size - 2] << 16)
+  bufSize = cdata[size - 4]
+            + (cdata[size - 3] << 8)
+            + (cdata[size - 2] << 16)
              + (cdata[size - 1] << 24);
    if (bufSize > MAX_CONTROL_SIZE)
    {
@@ -395,17 +399,24 @@ processControlTGZ (struct EXTRACTOR_ExtractContext *ec,
      return 0;
    }
    ret = 0;
-  memset (&strm, 0, sizeof (z_stream));
-  strm.next_in = (Bytef *) data;
+  memset (&strm,
+          0,
+          sizeof (z_stream));
+  strm.next_in = (Bytef *) cdata;
    strm.avail_in = size;
-  if (Z_OK == inflateInit2 (&strm, 15 + 32))
+  if (Z_OK ==
+      inflateInit2 (&strm,
```

```
+                     15 + 32))
   {
     strm.next_out = (Bytef *) buf;
     strm.avail_out = bufSize;
-    inflate (&strm, Z_FINISH);
+    inflate (&strm,
+             Z_FINISH);
     if (strm.total_out > 0)
-      ret = processControlTar (buf, strm.total_out,
-                               ec->proc, ec->cls);
+      ret = processControlTar (buf,
+                               strm.total_out,
+                               ec->proc,
+                               ec->cls);
     inflateEnd (&strm);
   }
   free (buf);
@@ -462,6 +473,9 @@ struct ObjectHeader
  * @param ec extraction context provided to the plugin
  */
 void
+EXTRACTOR_deb_extract_method (struct EXTRACTOR_ExtractContext *ec);
+
+void
 EXTRACTOR_deb_extract_method (struct EXTRACTOR_ExtractContext *ec)
 {
   uint64_t pos;
diff --git a/src/plugins/gif_extractor.c b/src/plugins/gif_extractor.c
@@ -80,12 +80,8 @@ EXTRACTOR_gif_extract_method (struct EXTRACTOR_ExtractContext *ec)
   gif_file = DGifOpen (ec, &gif_READ_func, &gif_error);
   if ((gif_file == NULL) || (gif_error != 0))
   {
-    if (gif_file != NULL)
-#if GIFLIB_MAJOR < 5 || GIFLIB_MINOR < 1
-      EGifCloseFile (gif_file);
-#else
-      EGifCloseFile (gif_file, NULL);
-#endif
+    if (NULL != gif_file)
+      goto cleanup;
     return; /* not a GIF */
   }
 #endif
@@ -97,7 +93,7 @@ EXTRACTOR_gif_extract_method (struct EXTRACTOR_ExtractContext *ec)
                 "text/plain",
                 "image/gif",
                 strlen ("image/gif") + 1))
-    return;
+    goto cleanup;
   snprintf (dims,
             sizeof (dims),
             "%dx%d",
@@ -111,7 +107,7 @@ EXTRACTOR_gif_extract_method (struct EXTRACTOR_ExtractContext *ec)
                 "text/plain",
                 dims,
                 strlen (dims) + 1))
-    return;
+    goto cleanup;
   while (1)
   {
     if (GIF_OK !=
@@ -139,14 +135,17 @@ EXTRACTOR_gif_extract_method (struct EXTRACTOR_ExtractContext *ec)
       break;
     }
     while ( (GIF_ERROR !=
```

```
-              DGifGetExtensionNext (gif_file, &ext)) &&
+              DGifGetExtensionNext (gif_file,
+                                    &ext)) &&
             (NULL != ext) )
        ;                   /* keep going */
    }
+cleanup:
 #if defined (GIF_LIB_VERSION) || GIFLIB_MAJOR < 5 || GIFLIB_MINOR < 1
   DGifCloseFile (gif_file);
 #else
-  DGifCloseFile (gif_file, NULL);
+  DGifCloseFile (gif_file,
+                 NULL);
 #endif
 }

diff --git a/src/plugins/nsfe_extractor.c b/src/plugins/nsfe_extractor.c
@@ -289,7 +289,7 @@ auth_extract (struct EXTRACTOR_ExtractContext *ec,
    char *artist;
    char *copyright;
    char *ripper;
-   uint32_t left = size;
+   ssize_t left = size;
    void *data;
    const char *cdata;

diff --git a/src/plugins/ole2_extractor.c b/src/plugins/ole2_extractor.c
@@ -774,22 +774,28 @@ le_input_read (GsfInput *input,
    ssize_t ret;

    ec = li->priv->ec;
-   old_off = ec->seek (ec->cls, 0, SEEK_CUR);
+   old_off = ec->seek (ec->cls,
+                       0,
+                       SEEK_CUR);
    if (num_bytes
        != (ret = ec->read (ec->cls,
                            &buf,
                            num_bytes)))
    {
      /* we don't support partial reads;
- most other GsfInput implementations in this case
- allocate some huge temporary buffer just to avoid
- the partial read; we might need to do that as well!? */
-     ec->seek (ec->cls, SEEK_SET, old_off);
+       most other GsfInput implementations in this case
+       allocate some huge temporary buffer just to avoid
+       the partial read; we might need to do that as well!? */
+     ec->seek (ec->cls,
+               old_off,
+               SEEK_SET);
      return NULL;
    }
    if (NULL != optional_buffer)
    {
-     memcpy (optional_buffer, buf, num_bytes);
+     memcpy (optional_buffer,
+             buf,
+             num_bytes);
      return optional_buffer;
    }
    return buf;
@@ -852,7 +858,8 @@ le_input_class_init (LeInputClass *class)
    input_class->Dup = le_input_dup;
    input_class->Read = le_input_read;
```

```
    input_class->Seek = le_input_seek;
-   g_type_class_add_private (class, sizeof (LeInputPrivate));
+   g_type_class_add_private (class,
+                             sizeof (LeInputPrivate));
 }


@@ -949,7 +956,9 @@ EXTRACTOR_ole2_extract_method (struct EXTRACTOR_ExtractContext *ec)
          + (data512[729] << 24);
    fcb = data512[722] + (data512[723] << 8) + (data512[724] << 16)
          + (data512[725] << 24);
-   if (0 != ec->seek (ec->cls, 0, SEEK_SET))
+   if (0 != ec->seek (ec->cls,
+                      0,
+                      SEEK_SET))
    {
      /* seek failed!? */
      return;
diff --git a/src/plugins/pack.c b/src/plugins/pack.c
@@ -164,7 +164,7 @@ EXTRACTOR_common_cat_unpack (const void *buf,
          *ll |= ((long long) *bp++) << 40;
          *ll |= ((long long) *bp++) << 32;
          *ll |= ((long long) *bp++) << 24;
-         *ll |= ((long long) *bp++) << 18;
+         *ll |= ((long long) *bp++) << 16;
          *ll |= ((long long) *bp++) << 8;
          *ll |= ((long long) *bp++);
          ++ll;
diff --git a/src/plugins/qt_extractor.c b/src/plugins/qt_extractor.c
@@ -778,9 +778,10 @@ ftypHandler (const char *input,
        NULL != ftMap[i].ext;
        i++)
    {
-     if (0 != memcmp (ft->type,
-                      ftMap[i].ext,
-                      4))
+     if (0 ==
+         memcmp (ft->type,
+                 ftMap[i].ext,
+                 4))
      {
        addKeyword (EXTRACTOR_METATYPE_MIMETYPE,
                    ftMap[i].mime,
diff --git a/src/plugins/real_extractor.c b/src/plugins/real_extractor.c
@@ -83,10 +83,10 @@ struct ContentDescription
  * @param t type of the meta data
  */
 #define ADD(s,t) do { \
-     if (0 != ec->proc (ec->cls, "real", t, \
-                        EXTRACTOR_METAFORMAT_C_STRING, \
-                        "text/plain", s, strlen (s) + 1)) \
-     { return; } \
+          if (0 != ec->proc (ec->cls, "real", t, \
+                             EXTRACTOR_METAFORMAT_C_STRING, \
+                             "text/plain", s, strlen (s) + 1)) \
+          { return; } \
 } while (0)


@@ -357,7 +357,7 @@ extract_raff3 (struct EXTRACTOR_ExtractContext *ec,
      char x[clen + 1];

      memcpy (x,
-             &data[11 + RAFF4_HDR_SIZE + tlen + alen],
+             &data[11 + RAFF3_HDR_SIZE + tlen + alen],
```

```
               clen);
      x[clen] = '\0';
      ADD (x,
@@ -371,7 +371,7 @@ extract_raff3 (struct EXTRACTOR_ExtractContext *ec,
      char x[aplen + 1];

      memcpy (x,
-             &data[12 + RAFF4_HDR_SIZE + tlen + alen + clen],
+             &data[12 + RAFF3_HDR_SIZE + tlen + alen + clen],
              aplen);
      x[aplen] = '\0';
      ADD (x,
```