erlang / otp Public

Code Issues 368 Pull requests 142 Actions Projects Wiki Se

Commit ba2f2bc

Erlang/OTP committed on Jun 16, 2025

```
Merge branch 'lukas/stdlib/zip-sanitize-filename/27/OTP-19653' into maint-27

* lukas/stdlib/zip-sanitize-filename/27/OTP-19653:
  stdlib: Properly sanatize filenames when (un)zipping
```

master · OTP-29.0-rc3 ••• OTP-27.3.4.1

2 parents 1c8d357 + b49c0b4 commit ba2f2bc

2 files changed +74 -12

Top

Filter files...

- lib/stdlib
  - src
    - zip.erl
  - test
    - zip_SUITE.erl

Search within code

lib/stdlib/src/zip.erl

```
@@ -1237,12 +1237,12 @@ get_filename({Name, _}, Type) ->
1237 1237     get_filename({Name, _, _}, Type) ->
1238 1238         get_filename(Name, Type);
1239 1239     get_filename(Name, regular) ->
1240      -      Name;
     1240 +      sanitize_filename(Name);
```

```
1241  1241    get_filename(Name, directory) ->
1242  1242        %% Ensure trailing slash
1243  1243        case lists:reverse(Name) of
1244        -     [$/ | _Rev] -> Name;
1245        -     Rev         -> lists:reverse([$/ | Rev])
      1244  +     [$/ | _Rev] -> sanitize_filename(Name);
      1245  +     Rev         -> sanitize_filename(lists:reverse([$/ | Rev]))
1246  1246        end.
1247  1247
1248  1248    add_cwd(_CWD, {_Name, _} = F) -> F;
                @@ -2365,12 +2365,25 @@ check_dir_level([_Dir | Parts], Level) ->
2365  2365    get_filename_extra(FileNameLen, ExtraLen, B, GPFlag) ->
2366  2366        try
2367  2367            <<BFileName:FileNameLen/binary, BExtra:ExtraLen/binary>> = B,
2368        -         {binary_to_chars(BFileName, GPFlag), BExtra}
      2368  +         {sanitize_filename(binary_to_chars(BFileName, GPFlag)), BExtra}
2369  2369        catch
2370  2370            _:_ ->
2371  2371                throw(bad_file_header)
2372  2372        end.
2373  2373
      2374  + sanitize_filename(Filename) ->
      2375  +     case filename:pathtype(Filename) of
      2376  +         relative -> Filename;
      2377  +         _ ->
      2378  +             %% With absolute or volumerelative, we drop the prefix and rejoin
      2379  +             %% the path to create a relative path
      2380  +             Relative = filename:join(tl(filename:split(Filename))),
      2381  +             error_logger:format("Illegal absolute path: ~ts, converting to
    ~ts~n",
      2382  +                                 [Filename, Relative]),
      2383  +             relative = filename:pathtype(Relative),
      2384  +             Relative
      2385  +     end.
      2386  +
2374  2387    %% get compressed or stored data
2375  2388    get_z_data(?DEFLATED, In0, FileName, CompSize, Input, Output, OpO, Z) ->
2376  2389        ok = zlib:inflateInit(Z, -?MAX_WBITS),
```

lib/stdlib/test/zip_SUITE.erl

```
@@ -25,7 +25,7 @@
25 25
26 26  -export([borderline/1, atomic/1,
27 27            bad_zip/1, unzip_from_binary/1, unzip_to_binary/1,
28    -          zip_to_binary/1,
   28 +          zip_to_binary/1, sanitize_filenames/1,
29 29            unzip_options/1, zip_options/1, list_dir_options/1, aliases/1,
30 30            zip_api/1, open_leak/1, unzip_jar/1,
31 31         unzip_traversal_exploit/1,
@@ -97,7 +97,7 @@ un_z64(Mode) ->
97 97        end.
98 98
99 99    zip_testcases() ->
100   -     [mode, basic_timestamp, extended_timestamp, uid_gid].
   100 +     [mode, basic_timestamp, extended_timestamp, uid_gid, sanitize_filenames].
101 101
102 102   zip64_testcases() ->
103 103       [unzip64_central_headers,
@@ -231,22 +231,27 @@ borderline_test(Size, TempDir) ->
231 231       {ok, Archive} = zip:zip(Archive, [Name]),
232 232       ok = file:delete(Name),
233 233
   234 +     RelName = filename:join(tl(filename:split(Name))),
   235 +
234 236       %% Verify listing and extracting.
235 237       {ok, [#zip_comment{comment = []},
236   -           #zip_file{name = Name,
   238 +           #zip_file{name = RelName,
237 239                     info = Info,
238 240                     offset = 0,
239 241                     comp_size = _}]} = zip:list_dir(Archive),
240 242       Size = Info#file_info.size,
241   -     {ok, [Name]} = zip:extract(Archive, [verbose]),
   243 +     TempRelName = filename:join(TempDir, RelName),
   244 +     {ok, [TempRelName]} = zip:extract(Archive, [verbose, {cwd, TempDir}]),
242 245
```

```
243       -     %% Verify contents of extracted file.
244       -     {ok, Bin} = file:read_file(Name),
245       -     true = match_byte_list(X0, binary_to_list(Bin)),
          246 +     %% Verify that absolute file was not created
          247 +     {error, enoent} = file:read_file(Name),
246       248
          249 +     %% Verify that relative contents of extracted file.
          250 +     {ok, Bin} = file:read_file(TempRelName),
          251 +     true = match_byte_list(X0, binary_to_list(Bin)),
247       252
248       253       %% Verify that Unix zip can read it. (if we have a unix zip that is!)
249       -     zipinfo_match(Archive, Name),
          254 +     zipinfo_match(Archive, RelName),
250       255
251       256       ok.
252       257
                  @@ -1619,6 +1624,50 @@ uid_gid(Config) ->
1619      1624
1620      1625       ok.
1621      1626
          1627 + sanitize_filenames(Config) ->
          1628 +     RootDir = get_value(pdir, Config),
          1629 +     TempDir = filename:join(RootDir, "sanitize_filenames"),
          1630 +     ok = file:make_dir(TempDir),
          1631 +
          1632 +     %% Check that /tmp/absolute does not exist
          1633 +     {error, enoent} = file:read_file("/tmp/absolute"),
          1634 +
          1635 +     %% Create a zip archive /tmp/absolute in it
          1636 +     %%   This file was created using the command below on Erlang/OTP 28.0
          1637 +     %%   1> rr(file), {ok, {_, Bin}} = zip:zip("absolute.zip",
                  [{"/tmp/absolute",<<>>,#file_info{ type=regular, mtime={{2000,1,1},{0,0,0}},
                  size=0 }}], [memory]), rp(base64:encode(Bin)).
          1638 +     AbsZip =
                  base64:decode(<<"UEsDBAoAAAAAAAAAISgAAAAAAAAAAAAAAAAANAAkAL3RtcC9hYnNvbHV0ZVVU
                  BQABcDVtOFBLAQI9AwoAAAAAAAAAISgAAAAAAAAAAAAAAAAANAAkAAAAAAAAAAACkAQAAAAAvdG1wL
                  2Fic29sdXRlVVQFAAFwNW04UEsFBgAAAAABAAEARAAAADQAAAAAAA==">>),
          1639 +     AbsArchive = filename:join(TempDir, "absolute.zip"),
          1640 +     ok = file:write_file(AbsArchive, AbsZip),
```

```
1641 +
1642 +     {ok, ["tmp/absolute"]} = unzip(Config, AbsArchive, [verbose, {cwd,
     TempDir}]),
1643 +
1644 +     zipinfo_match(AbsArchive, "/tmp/absolute"),
1645 +
1646 +     case un_z64(get_value(unzip, Config)) =/= unemzip of
1647 +         true ->
1648 +             {error, enoent} = file:read_file("/tmp/absolute"),
1649 +             {ok, <<>>} = file:read_file(filename:join([TempDir, "tmp",
     "absolute"]));
1650 +         false ->
1651 +             ok
1652 +     end,
1653 +
1654 +     RelArchive = filename:join(TempDir, "relative.zip"),
1655 +     Relative = filename:join(TempDir, "relative"),
1656 +     ok = file:write_file(Relative, <<>>),
1657 +     ?assertMatch({ok, RelArchive},zip(Config, RelArchive, "", [Relative],
     [{cwd, TempDir}])),
1658 +
1659 +     SanitizedRelative = filename:join(tl(filename:split(Relative))),
1660 +     case un_z64(get_value(unzip, Config)) =:= unemzip of
1661 +         true ->
1662 +             {ok, [SanitizedRelative]} = unzip(Config, RelArchive, [{cwd,
     TempDir}]);
1663 +         false ->
1664 +             ok
1665 +     end,
1666 +
1667 +     zipinfo_match(RelArchive, SanitizedRelative),
1668 +
1669 +     ok.
1670 +
1622 1671   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1623 1672   %%% Generic zip interface
1624 1673   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Comments 0

Please sign in to comment.