erlang / otp Public

Code Issues 368 Pull requests 156 Actions Projects Wiki Se

Commit 578d400

Erlang/OTP committed on Jun 16, 2025

```
Merge branch 'lukas/stdlib/zip-sanitize-filename/OTP-19653' into maint-26

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

master · OTP-29.0-rc3 ••• OTP-26.2.5.13

2 parents 58a0c2a + 1060887 commit 578d400

2 files changed +49 -12 lines changed

Top

Filter files…

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

2 files changed +49 -12 lines changed

Search within code

lib/stdlib/src/zip.erl

```
@@ -833,12 +833,12 @@ get_filename({Name, _}, Type) ->
833 833     get_filename({Name, _, _}, Type) ->
834 834         get_filename(Name, Type);
835 835     get_filename(Name, regular) ->
836     -       Name;
    836 +       sanitize_filename(Name);
837 837     get_filename(Name, directory) ->
838 838         %% Ensure trailing slash
```

```
 839   839        case lists:reverse(Name) of
 840         -      [$/ | _Rev] -> Name;
 841         -      Rev         -> lists:reverse([$/ | Rev])
       840   +      [$/ | _Rev] -> sanitize_filename(Name);
       841   +      Rev         -> sanitize_filename(lists:reverse([$/ | Rev]))
 842   842        end.
 843   843
 844   844    add_cwd(_CWD, {_Name, _} = F) -> F;
            ↕     @@ -1550,12 +1550,25 @@ check_dir_level([_Dir | Parts], Level) ->
1550  1550    get_file_name_extra(FileNameLen, ExtraLen, B, GPFlag) ->
1551  1551        try
1552  1552            <<BFileName:FileNameLen/binary, BExtra:ExtraLen/binary>> = B,
1553        -           {binary_to_chars(BFileName, GPFlag), BExtra}
      1553  +           {sanitize_filename(binary_to_chars(BFileName, GPFlag)), BExtra}
1554  1554        catch
1555  1555            _:_ ->
1556  1556                throw(bad_file_header)
1557  1557        end.
1558  1558
      1559  + sanitize_filename(Filename) ->
      1560  +     case filename:pathtype(Filename) of
      1561  +         relative -> Filename;
      1562  +         _ ->
      1563  +             %% With absolute or volumerelative, we drop the prefix and rejoin
      1564  +             %% the path to create a relative path
      1565  +             Relative = filename:join(tl(filename:split(Filename))),
      1566  +             error_logger:format("Illegal absolute path: ~ts, converting to
            ~ts~n",
      1567  +                                 [Filename, Relative]),
      1568  +             relative = filename:pathtype(Relative),
      1569  +             Relative
      1570  +     end.
      1571  +
1559  1572    %% get compressed or stored data
1560  1573    get_z_data(?DEFLATED, In0, FileName, CompSize, Input, Output, OpO, Z) ->
1561  1574        ok = zlib:inflateInit(Z, -?MAX_WBITS),
            ↕
```

lib/stdlib/test/zip_SUITE.erl

```
@@ -22,7 +22,7 @@
-export([all/0, suite/0,groups/0,init_per_suite/1, end_per_suite/1,
          init_per_group/2,end_per_group/2, borderline/1, atomic/1,
              bad_zip/1, unzip_from_binary/1, unzip_to_binary/1,
-             zip_to_binary/1,
+             zip_to_binary/1, sanitize_filenames/1,
              unzip_options/1, zip_options/1, list_dir_options/1, aliases/1,
              openzip_api/1, zip_api/1, open_leak/1, unzip_jar/1,
          unzip_traversal_exploit/1,
@@ -40,7 +40,8 @@ all() ->
          unzip_to_binary, zip_to_binary, unzip_options,
          zip_options, list_dir_options, aliases, openzip_api,
          zip_api, open_leak, unzip_jar, compress_control, foldl,
-         unzip_traversal_exploit,fd_leak,unicode,test_zip_dir].
+         unzip_traversal_exploit,fd_leak,unicode,test_zip_dir,
+         sanitize_filenames].

 groups() ->
     [].
@@ -90,22 +91,27 @@ borderline_test(Size, TempDir) ->
     {ok, Archive} = zip:zip(Archive, [Name]),
     ok = file:delete(Name),

+    RelName = filename:join(tl(filename:split(Name))),
+
     %% Verify listing and extracting.
     {ok, [#zip_comment{comment = []},
-          #zip_file{name = Name,
+          #zip_file{name = RelName,
                     info = Info,
                     offset = 0,
                     comp_size = _}]} = zip:list_dir(Archive),
     Size = Info#file_info.size,
-    {ok, [Name]} = zip:extract(Archive, [verbose]),
+    TempRelName = filename:join(TempDir, RelName),
+    {ok, [TempRelName]} = zip:extract(Archive, [verbose, {cwd, TempDir}]),

-    %% Verify contents of extracted file.
-    {ok, Bin} = file:read_file(Name),
```

```
104       -     true = match_byte_list(X0, binary_to_list(Bin)),
      106 +     %% Verify that absolute file was not created
      107 +     {error, enoent} = file:read_file(Name),
105   108
      109 +     %% Verify that relative contents of extracted file.
      110 +     {ok, Bin} = file:read_file(TempRelName),
      111 +     true = match_byte_list(X0, binary_to_list(Bin)),
106   112
107   113       %% Verify that Unix zip can read it. (if we have a unix zip that is!)
108       -     zipinfo_match(Archive, Name),
      114 +     zipinfo_match(Archive, RelName),
109   115
110   116       ok.
111   117
          @@ -1054,3 +1060,21 @@ run_command(Command, Args) ->
1054  1060                end
1055  1061         end)().
1056  1062
      1063 + sanitize_filenames(Config) ->
      1064 +     RootDir = proplists:get_value(priv_dir, Config),
      1065 +     TempDir = filename:join(RootDir, "borderline"),
      1066 +     ok = file:make_dir(TempDir),
      1067 +
      1068 +     %% Create a zip archive /tmp/absolute in it
      1069 +     %%   This file was created using the command below on Erlang/OTP 28.0
      1070 +     %%   1> rr(file), {ok, {_, Bin}} = zip:zip("absolute.zip",
          [{"/tmp/absolute",<<>>,#file_info{ type=regular, mtime={{1970,1,1},{0,0,0}},
          size=0 }}], [memory]), rp(base64:encode(Bin)).
      1071 +     AbsZip =
          base64:decode(<<"UEsDBBQAAAAAAAAAIewAAAAAAAAAAAAAAAANAAAAL3RtcC9hYnNvbHV0ZVBL
          AQIUAxQAAAAAAAAAIewAAAAAAAAAAAAAAANAAAAAAAAAAAAAACkAQAAAAAvdG1wL2Fic29sdXRlU
          EsFBgAAAAABAAEAOwAAACsAAAAAAA==">>),
      1072 +     Archive = filename:join(TempDir, "absolute.zip"),
      1073 +     ok = file:write_file(Archive, AbsZip),
      1074 +
      1075 +     TmpAbs = filename:join([TempDir, "tmp", "absolute"]),
      1076 +     {ok, [TmpAbs]} = zip:unzip(Archive, [verbose, {cwd, TempDir}]),
      1077 +     {error, enoent} = file:read_file("/tmp/absolute"),
      1078 +     {ok, <<>>} = file:read_file(TmpAbs),
```

```
1079 +
1080 +     ok.
```

Comments 0

Please sign in to comment.