

# Commit 5a55fee



Erlang/OTP committed on Jun 13, 2025

Merge branch 'lukas/stdlib/zip-sanitize-filename/27/OTP-19653' into maint-28

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

# Conflicts:  
# lib/stdlib/test/zip\_SUITE.erl

🔗 master · 📁 OTP-29.0-rc3 ... OTP-28.0.1

2 parents [4ae92fb](#) + [b49c0b4](#) commit 5a55fee 📄

📁 2 files changed +74 -12 lines changed

↑ Top ⚙️

🔍 Filter files...

- ✓ 📁 lib/stdlib
  - ✓ 📁 src
    - 📄 zip.erl
  - ✓ 📁 test
    - 📄 zip\_SUITE.erl

📁 2 files changed +74 -12 lines changed

🔍 Search within code ⚙️

lib/stdlib/src/zip.erl

```

  ... @@ -1240,12 +1240,12 @@ get_filename({Name, _}, Type) ->
1240 1240  get_filename({Name, _, _}, Type) ->
1241 1241      get_filename(Name, Type);
1242 1242  get_filename(Name, regular) ->
1243      - Name;
1243 1243  + sanitize_filename(Name);

```

```

1244 1244     get_filename(Name, directory) ->
1245 1245         %% Ensure trailing slash
1246 1246         case lists:reverse(Name) of
1247 1247 -         [$/ | _Rev] -> Name;
1248 1248 -         Rev         -> lists:reverse([$/ | Rev])
1247 1247 +         [$/ | _Rev] -> sanitize_filename(Name);
1248 1248 +         Rev         -> sanitize_filename(lists:reverse([$/ | Rev]))
1249 1249         end.
1250 1250
1251 1251     add_cwd(_CWD, {_Name, _} = F) -> F;
@@ -2368,12 +2368,25 @@ check_dir_level([_Dir | Parts], Level) ->
2368 2368     get_filename_extra(FileNameLen, ExtraLen, B, GPFlag) ->
2369 2369         try
2370 2370             <<BFileName:FileNameLen/binary, BExtra:ExtraLen/binary>> = B,
2371 2371 -         {binary_to_chars(BFileName, GPFlag), BExtra}
2371 2371 +         {sanitize_filename(binary_to_chars(BFileName, GPFlag)), BExtra}
2372 2372         catch
2373 2373             _:_ ->
2374 2374                 throw(bad_file_header)
2375 2375         end.
2376 2376
2377 2377 +     sanitize_filename(Filename) ->
2378 2378 +         case filename:pathtype(Filename) of
2379 2379 +             relative -> Filename;
2380 2380 +             _ ->
2381 2381 +                 %% With absolute or volumerelative, we drop the prefix and rejoin
2382 2382 +                 %% the path to create a relative path
2383 2383 +                 Relative = filename:join(tl(filename:split(Filename))),
2384 2384 +                 error_logger:format("Illegal absolute path: ~ts, converting to
2385 2385 +                 ~ts-n",
2385 2385 +                     [Filename, Relative]),
2386 2386 +                 relative = filename:pathtype(Relative),
2387 2387 +                 Relative
2388 2388 +             end.
2389 2389 +
2377 2390     %% get compressed or stored data
2378 2391     get_z_data(?DEFLATED, In0, FileName, CompSize, Input, Output, Op0, Z) ->
2379 2392         ok = zlib:inflateInit(Z, -?MAX_WBITS),

```

```

lib/stdlib/test/zip_SUITE.erl
@@ -27,7 +27,7 @@
27 27
28 28     -export([borderline/1, atomic/1,
29 29         bad_zip/1, unzip_from_binary/1, unzip_to_binary/1,
30 30     -   zip_to_binary/1,
31 31     +   zip_to_binary/1, sanitize_filenames/1,
32 32         unzip_options/1, zip_options/1, list_dir_options/1, aliases/1,
33 33         zip_api/1, open_leak/1, unzip_jar/1,
34 34         unzip_traversal_exploit/1,
35 35     ]),
@@ -99,7 +99,7 @@ un_z64(Mode) ->
99 99     end.
100 100
101 101     zip_testcases() ->
102 102     -   [mode, basic_timestamp, extended_timestamp, capped_timestamp, uid_gid].
103 103     +   [mode, basic_timestamp, extended_timestamp, capped_timestamp, uid_gid,
104 104         sanitize_filenames].
105 105
106 106     zip64_testcases() ->
107 107     [unzip64_central_headers,
108 108     ],
@@ -233,27 +233,27 @@ borderline_test(Size, TempDir) ->
233 233     {ok, Archive} = zip:zip(Archive, [Name]),
234 234     ok = file:delete(Name),
235 235
236 236     +   RelName = filename:join(tl(filename:split(Name))),
237 237     +
238 238     %% Verify listing and extracting.
239 239     {ok, [#zip_comment{comment = []},
240 240     -   #zip_file{name = Name,
241 241     +   #zip_file{name = RelName,
242 242         info = Info,
243 243         offset = 0,
244 244         comp_size = _]}]} = zip:list_dir(Archive),
245 245     Size = Info#file_info.size,
246 246     -   {ok, [Name]} = zip:extract(Archive, [verbose]),
247 247     +   TempRelName = filename:join(TempDir, RelName),
248 248     +   {ok, [TempRelName]} = zip:extract(Archive, [verbose, {cwd, TempDir}]),

```

244	247	
245		- %% Verify contents of extracted file.
246		- {ok, Bin} = file:read_file(Name),
247		- true = match_byte_list(X0, binary_to_list(Bin)),
248		+ %% Verify that absolute file was not created
249		+ {error, enoent} = file:read_file(Name),
248	250	
251		+ %% Verify that relative contents of extracted file.
252		+ {ok, Bin} = file:read_file(TempRelName),
253		+ true = match_byte_list(X0, binary_to_list(Bin)),
249	254	
250	255	%% Verify that Unix zip can read it. (if we have a unix zip that is!)
251		- zipinfo_match(Archive, Name),
256		+ zipinfo_match(Archive, RelName),
252	257	
253	258	ok.
254	259	
		<div style="text-align: center;"> <span>⋮</span>  <span>↓</span>  <span>↑</span>  <span>⋮</span> </div>
		@@ -1654,6 +1659,50 @@ uid_gid(Config) ->
1654	1659	
1655	1660	ok.
1656	1661	
1662		+ sanitize_filenames(Config) ->
1663		+ RootDir = get_value(pdir, Config),
1664		+ TempDir = filename:join(RootDir, "sanitize_filenames"),
1665		+ ok = file:make_dir(TempDir),
1666		+
1667		+ %% Check that /tmp/absolute does not exist
1668		+ {error, enoent} = file:read_file("/tmp/absolute"),
1669		+
1670		+ %% Create a zip archive /tmp/absolute in it
1671		+ %% This file was created using the command below on Erlang/OTP 28.0
1672		+ %% 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)).
1673		+ AbsZip =
		base64:decode(<<"UESDBAoAAAAAAAAAISgAAAAAAAAAAAAAAAAAANAkAL3RtcC9hYnNvbHV0ZVVU
		BQABcDVtOFBLAQI9AwoAAAAAAAAAISgAAAAAAAAAAAAAAAAAANAkAAAAAAAAAAckAQAAAAAwdG1wL
		2Fic29sdXRlVVQFAAFwNW04UESFBgAAAAABAAEARAAAAADQAAAAAA==">>),
1674		+ AbsArchive = filename:join(TempDir, "absolute.zip"),

```

1675 +     ok = file:write_file(AbsArchive, AbsZip),
1676 +
1677 +     {ok, ["tmp/absolute"]} = unzip(Config, AbsArchive, [verbose, {cwd,
TempDir}]),
1678 +
1679 +     zipinfo_match(AbsArchive, "/tmp/absolute"),
1680 +
1681 +     case un_z64(get_value(unzip, Config)) /= unemzip of
1682 +         true ->
1683 +             {error, enoent} = file:read_file("/tmp/absolute"),
1684 +             {ok, <<>} = file:read_file(filename:join([TempDir, "tmp",
"absolute"]));
1685 +         false ->
1686 +             ok
1687 +     end,
1688 +
1689 +     RelArchive = filename:join(TempDir, "relative.zip"),
1690 +     Relative = filename:join(TempDir, "relative"),
1691 +     ok = file:write_file(Relative, <<>),
1692 +     ?assertMatch({ok, RelArchive}, zip(Config, RelArchive, "", [Relative],
[{:cwd, TempDir}])),
1693 +
1694 +     SanitizedRelative = filename:join(tl(filename:split(Relative))),
1695 +     case un_z64(get_value(unzip, Config)) := unemzip of
1696 +         true ->
1697 +             {ok, [SanitizedRelative]} = unzip(Config, RelArchive, [{:cwd,
TempDir}]);
1698 +         false ->
1699 +             ok
1700 +     end,
1701 +
1702 +     zipinfo_match(RelArchive, SanitizedRelative),
1703 +
1704 +     ok.
1705 +

```

```
1657 1706 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
1658 1707 %% Generic zip interface
```

```
1659 1708 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```



## Comments 0



Please [sign in](#) to comment.