

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 sanitize filenames when (un)zipping

master · OTP-29.0-rc2 ... 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;
+   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
1567      +             ~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, Op0, Z) ->
1561      1574      ok = zlib:inflateInit(Z, -?MAX_WBITS),
      ↓

```

lib/stdlib/test/zip_SUITE.erl

...

	↑↑	@@ -22,7 +22,7 @@
22	22	-export([all/0, suite/0, groups/0, init_per_suite/1, end_per_suite/1,
23	23	init_per_group/2, end_per_group/2, borderline/1, atomic/1,
24	24	bad_zip/1, unzip_from_binary/1, unzip_to_binary/1,
25	-	zip_to_binary/1,
25	+	zip_to_binary/1, sanitize_filenames/1,
26	26	unzip_options/1, zip_options/1, list_dir_options/1, aliases/1,
27	27	openzip_api/1, zip_api/1, open_leak/1, unzip_jar/1,
28	28	unzip_traversal_exploit/1,
	⇕	@@ -40,7 +40,8 @@ all() ->
40	40	unzip_to_binary, zip_to_binary, unzip_options,
41	41	zip_options, list_dir_options, aliases, openzip_api,
42	42	zip_api, open_leak, unzip_jar, compress_control, foldl,
43	-	unzip_traversal_exploit, fd_leak, unicode, test_zip_dir].
43	+	unzip_traversal_exploit, fd_leak, unicode, test_zip_dir,
44	+	sanitize_filenames].
44	45	
45	46	groups() ->
46	47	[].
	↓ ↑	@@ -90,22 +91,27 @@ borderline_test(Size, TempDir) ->
90	91	{ok, Archive} = zip:zip(Archive, [Name]),
91	92	ok = file:delete(Name),
92	93	
94	+	RelName = filename:join(tl(filename:split(Name))),
95	+	
93	96	%% Verify listing and extracting.
94	97	{ok, [#zip_comment{comment = []},
95	-	#zip_file{name = Name,
98	+	#zip_file{name = RelName,
96	99	info = Info,
97	100	offset = 0,
98	101	comp_size = _}} = zip:list_dir(Archive),
99	102	Size = Info#file_info.size,
100	-	{ok, [Name]} = zip:extract(Archive, [verbose]),
103	+	TempRelName = filename:join(TempDir, RelName),
104	+	{ok, [TempRelName]} = zip:extract(Archive, [verbose, {cwd, TempDir}]),
101	105	
102	-	%% Verify contents of extracted file.
103	-	{ok, Bin} = file:read_file(Name),

104	-	<code>true = match_byte_list(X0, binary_to_list(Bin)),</code>
106	+	<code>%% Verify that absolute file was not created</code>
107	+	<code>{error, enoent} = file:read_file(Name),</code>
105	108	
109	+	<code>%% Verify that relative contents of extracted file.</code>
110	+	<code>{ok, Bin} = file:read_file(TempRelName),</code>
111	+	<code>true = match_byte_list(X0, binary_to_list(Bin)),</code>
106	112	
107	113	<code>%% Verify that Unix zip can read it. (if we have a unix zip that is!)</code>
108	-	<code>zipinfo_match(Archive, Name),</code>
114	+	<code>zipinfo_match(Archive, RelName),</code>
109	115	
110	116	<code>ok.</code>
111	117	
↓		<code>@@ -1054,3 +1060,21 @@ run_command(Command, Args) -></code>
1054	1060	<code>end</code>
1055	1061	<code>end)().</code>
1056	1062	
1063	+	<code>sanitize_filenames(Config) -></code>
1064	+	<code>RootDir = proplists:get_value(priv_dir, Config),</code>
1065	+	<code>TempDir = filename:join(RootDir, "borderline"),</code>
1066	+	<code>ok = file:make_dir(TempDir),</code>
1067	+	
1068	+	<code>%% Create a zip archive /tmp/absolute in it</code>
1069	+	<code>%% This file was created using the command below on Erlang/OTP 28.0</code>
1070	+	<code>%% 1> rr(file), {ok, {_, Bin}} = zip:zip("absolute.zip",</code>
		<code>[{" /tmp/absolute", <<>, #file_info{ type=regular, mtime={{1970,1,1},{0,0,0}},</code>
		<code>size=0 }}, [memory]], rp(base64:encode(Bin)).</code>
1071	+	<code>AbsZip =</code>
		<code>base64:decode(<<"UESDBBQAAAAAAAAAIewAAAAAAAAAAAAAAAAANAAAAL3RtcC9hYnNvbHV0ZVBL</code>
		<code>AQIUAXQAAAAAAAAAIewAAAAAAAAAAAAAAAAANAAAAAAAAAAAAACkAQAAAAVdG1wL2Fic29sdXRlU</code>
		<code>EsFBgAAAAABAAEA0wAAACsAAAAAAA==">>),</code>
1072	+	<code>Archive = filename:join(TempDir, "absolute.zip"),</code>
1073	+	<code>ok = file:write_file(Archive, AbsZip),</code>
1074	+	
1075	+	<code>TmpAbs = filename:join([TempDir, "tmp", "absolute"]),</code>
1076	+	<code>{ok, [TmpAbs]} = zip:unzip(Archive, [verbose, {cwd, TempDir}]),</code>
1077	+	<code>{error, enoent} = file:read_file("/tmp/absolute"),</code>
1078	+	<code>{ok, <<>} = file:read_file(TmpAbs),</code>

1079 +

1080 + `ok.`



Comments 0



Please [sign in](#) to comment.