

erlang / otp Public

<> Code Issues 365 Pull requests 145 Actions Projects Wiki Se

# Commit 043ee3c



**u3s** committed on Sep 2, 2025 Verified

ssh: ssh\_sftpd verify path size for client data  
- reject max\_path exceeding the 4096 limit or according to other option value

master · OTP-29.0-rc2 ... OTP-26.2.5.15

1 parent [c1ce39f](#) commit 043ee3c

**3 files changed** +97 -33 lines changed

[↑ Top](#)

- ✓ lib/ssh
  - ✓ doc/src
    - ssh\_sftpd.xml
  - ✓ src
    - ssh\_sftpd.erl
  - ✓ test
    - ssh\_sftpd\_SUITE.erl

**3 files changed** +97 -33 lines changed

```

  ✓ lib/ssh/doc/src/ssh_sftpd.xml
  ↑... @@ -65,6 +65,14 @@
65 65      If supplied, the number of filenames returned to the SFTP client per
      <c>READDIR</c>
66 66      request is limited to at most the given value.</p>
67 67      </item>
68 +      <tag><c>max_path</c></tag>

```

```

69 + <item>
70 + <p>The default value is <c>4096</c>. Positive integer
71 + value represents the maximum path length which cannot be
72 + exceeded in data provided by the SFTP client. (Note:
73 + limitations might be also enforced by underlying operating
74 + system)</p>
75 + </item>
68 76 <tag><c>root</c></tag>
69 77 <item>
70 78 <p>Sets the SFTP root directory. Then the user cannot see any files

```

```

lib/ssh/src/ssh_sftpd.erl
@@ -52,6 +52,7 @@
52 52 file_handler, % atom() - callback module
53 53 file_state, % state for the file callback module
54 54 max_files, % integer >= 0 max no files sent during
READDIR
55 + max_path, % integer > 0 - max length of path
55 56 options, % from the subsystem declaration
56 57 handles % list of open handles
57 58 %% handle is either {<int>, directory, {Path, unread|eof}} or
@@ -65,6 +66,7 @@
65 66 Options :: [ {cwd, string()} |
66 67 {file_handler, CbMod | {CbMod, FileState}} |
67 68 {max_files, integer()} |
69 + {max_path, integer()} |
68 70 {root, string()} |
69 71 {sftpd_vsn, integer()}
70 72 ],
@@ -115,8 +117,12 @@ init(Options) ->
115 117 {Root0, State0}
116 118 end,
117 119 MaxLength = proplists:get_value(max_files, Options, 0),
120 + MaxPath = proplists:get_value(max_path, Options, 4096),
118 121 Vsn = proplists:get_value(sftpd_vsn, Options, 5),
119 - {ok, State#state{cwd = CWD, root = Root, max_files = MaxLength,
122 + {ok, State#state{cwd = CWD,
123 + root = Root,

```

```

124 +         max_files = MaxLength,
125 +         max_path = MaxPath,
120 126         options = Options,
121 127         handles = [], pending = <<>>,
122 128         xf = #ssh_xfer{vsn = Vsn, ext = []}}}.
⌵
@@ -222,6 +228,30 @@ handle_data(Type, ChannelId, Data0, State =
⌶
#state{pending = Pending}) ->
222 228         handle_data(Type, ChannelId, Data, State#state{pending = <<>>})
223 229         end.
224 230
231 + handle_op(Request, ReqId, <<?UINT32(PLen), _/binary>>,
232 +         State = #state{max_path = MaxPath, xf = XF})
233 +     when (Request == ?SSH_FXP_LSTAT orelse
234 +         Request == ?SSH_FXP_MKDIR orelse
235 +         Request == ?SSH_FXP_OPEN orelse
236 +         Request == ?SSH_FXP_OPENDIR orelse
237 +         Request == ?SSH_FXP_READLINK orelse
238 +         Request == ?SSH_FXP_REALPATH orelse
239 +         Request == ?SSH_FXP_REMOVE orelse
240 +         Request == ?SSH_FXP_RMDIR orelse
241 +         Request == ?SSH_FXP_SETSTAT orelse
242 +         Request == ?SSH_FXP_STAT),
243 +         PLen > MaxPath ->
244 +         ssh_xfer:xf_send_status(XF, ReqId, ?SSH_FX_NO_SUCH_PATH,
245 +             "No such path"),
246 +         State;
247 + handle_op(Request, ReqId, <<?UINT32(PLen), _:PLen/binary, ?UINT32(PLen2),
248 +         _/binary>>,
249 +         State = #state{max_path = MaxPath, xf = XF})
250 +     when (Request == ?SSH_FXP_RENAME orelse
251 +         Request == ?SSH_FXP_SYMLINK),
252 +         (PLen > MaxPath orelse PLen2 > MaxPath) ->
253 +         ssh_xfer:xf_send_status(XF, ReqId, ?SSH_FX_NO_SUCH_PATH,
254 +             "No such path"),
255 +         State;
225 255     handle_op(?SSH_FXP_INIT, Version, B, State) when is_binary(B) ->
226 256         XF = State#state.xf,
227 257         Vsn = lists:min([XF#ssh_xfer.vsn, Version]),
⌵

```

```

lib/ssh/test/ssh_sftpd_SUITE.erl
@@ -43,6 +43,7 @@
43 43      open_file_dir_v6/1,
44 44      read_dir/1,
45 45      read_file/1,
46 +      max_path/1,
46 47      real_path/1,
47 48      relative_path/1,
48 49      relpath/1,
@@ -72,9 +73,8 @@
72 73      -define(SSH_TIMEOUT, 10000).
73 74      -define(REG_ATTRS, <<0,0,0,0,1>>).
74 75      -define(UNIX_EPOCH, 62167219200).
75 -
76 - -define(is_set(F, Bits),
77 -     ((F band (Bits)) == (F)).
76 + -define(MAX_PATH, 200).
77 + -define(is_set(F, Bits), ((F band (Bits)) == (F)).
78 78
79 79      %%-----
80 80      %% Common Test interface functions -----
@@ -87,6 +87,7 @@ all() ->
87 87      [open_close_file,
88 88      open_close_dir,
89 89      read_file,
90 +      max_path,
90 91      read_dir,
91 92      write_file,
92 93      rename_file,
@@ -181,7 +182,8 @@ init_per_testcase(TestCase, Config) ->
181 182      {sftpd_vsn, 6}]]],
182 183      ssh:daemon(0, [{subsystems, SubSystems}|Options]);
183 184      _ ->
184 -      SubSystems = [ssh_sftpd:subsystem_spec([])],
185 +      SubSystems = [ssh_sftpd:subsystem_spec(
186 +          [{max_path, ?MAX_PATH}]]],
185 187      ssh:daemon(0, [{subsystems, SubSystems}|Options])

```

186	188	end,
187	189	
		@@ -334,6 +336,23 @@ read_file(Config) when is_list(Config) ->
334	336	
335	337	{ok, Data} = file:read_file(FileName).
336	338	
339		+ %%-----
340		+ max_path(Config) when is_list(Config) ->
341		+ PrivDir = proplists:get_value(priv_dir, Config),
342		+ FileName = filename:join(PrivDir, "test.txt"),
343		+ {Cm, Channel} = proplists:get_value(sftp, Config),
344		+ %% verify max_path limit
345		+ LongFileName =
346		+ filename:join(PrivDir,
347		+ "t" ++ lists:flatten(lists:duplicate(?MAX_PATH, "e"))
		++ "st.txt"),
348		+ {ok, _} = file:copy(FileName, LongFileName),
349		+ ReqId1 = req_id(),
350		+ {ok, <<?SSH_FXP_STATUS, ?UINT32(ReqId1), ?UINT32(?SSH_FX_NO_SUCH_PATH),
351		+ _/binary>>, _} =
352		+ open_file(LongFileName, Cm, Channel, ReqId1,
353		+ ?ACE4_READ_DATA bor ?ACE4_READ_ATTRIBUTES,
354		+ ?SSH_FXF_OPEN_EXISTING).
355		+
337	356	%%-----
338	357	read_dir(Config) when is_list(Config) ->
339	358	PrivDir = proplists:get_value(priv_dir, Config),
		@@ -389,35 +408,33 @@ rename_file(Config) when is_list(Config) ->
389	408	PrivDir = proplists:get_value(priv_dir, Config),
390	409	FileName = filename:join(PrivDir, "test.txt"),
391	410	NewFileName = filename:join(PrivDir, "test1.txt"),
392		- ReqId = 0,
		+ LongFileName =
		+ filename:join(PrivDir,
		+ "t" ++ lists:flatten(lists:duplicate(?MAX_PATH, "e"))
		++ "st.txt"),
393	414	{Cm, Channel} = proplists:get_value(sftp, Config),
394		-

```

395 - {ok, <<?SSH_FXP_STATUS, ?UINT32(ReqId),
396 -     ?UINT32(?SSH_FX_OK), _/binary>>, _} =
397 - rename(FileName, NewFileName, Cm, Channel, ReqId, 6, 0),
398 -
399 - NewReqId = ReqId + 1,
400 -
401 - {ok, <<?SSH_FXP_STATUS, ?UINT32(NewReqId),
402 -     ?UINT32(?SSH_FX_OK), _/binary>>, _} =
403 - rename(NewFileName, FileName, Cm, Channel, NewReqId, 6,
404 -     ?SSH_FXP_RENAME_OVERWRITE),
405 -
406 - NewReqId1 = NewReqId + 1,
407 - file:copy(FileName, NewFileName),
408 -
409 - %% No overwrite
410 - {ok, <<?SSH_FXP_STATUS, ?UINT32(NewReqId1),
411 -     ?UINT32(?SSH_FX_FILE_ALREADY_EXISTS), _/binary>>, _} =
412 - rename(FileName, NewFileName, Cm, Channel, NewReqId1, 6,
413 -     ?SSH_FXP_RENAME_NATIVE),
414 -
415 - NewReqId2 = NewReqId1 + 1,
416 -
417 - {ok, <<?SSH_FXP_STATUS, ?UINT32(NewReqId2),
418 -     ?UINT32(?SSH_FX_OP_UNSUPPORTED), _/binary>>, _} =
419 - rename(FileName, NewFileName, Cm, Channel, NewReqId2, 6,
420 -     ?SSH_FXP_RENAME_ATOMIC).
415 + Version = 6,
416 + [begin
417 +     case Action of
418 +         {Code, AFile, BFile, Flags} ->
419 +             ReqId = req_id(),
420 +             ct:log("ReqId = ~p,~nCode = ~p,~nAFile = ~p,~nBFile =
~p,~nFlags = ~p",
421 +                 [ReqId, Code, AFile, BFile, Flags]),
422 +             {ok, <<?SSH_FXP_STATUS, ?UINT32(ReqId), ?UINT32(Code),
_/binary>>, _} =
423 +                 rename(AFile, BFile, Cm, Channel, ReqId, Version,
Flags);
424 +             {file_copy, AFile, BFile} ->
425 +                 {ok, _} = file:copy(AFile, BFile)

```

```

426 +         end
427 +     end ||
428 +     Action <-
429 +         [{?SSH_FX_OK, FileName, NewFileName, 0},
430 +          {?SSH_FX_OK, NewFileName, FileName, ?SSH_FXP_RENAME_OVERWRITE},
431 +          {file_copy, FileName, NewFileName},
432 +          %% no overwrite
433 +          {?SSH_FX_FILE_ALREADY_EXISTS, FileName, NewFileName, ?
SSH_FXP_RENAME_NATIVE},
434 +          {?SSH_FX_OP_UNSUPPORTED, FileName, NewFileName, ?
SSH_FXP_RENAME_ATOMIC},
435 +          %% max_path
436 +          {?SSH_FX_NO_SUCH_PATH, FileName, LongFileName, 0}]],
437 +     ok.

```

421 438

422 439

%%-----

423 440

mk\_rm\_dir(Config) when is\_list(Config) ->



@@ -1087,3 +1104,12 @@ encode\_file\_type(Type) ->

1087 1104

1088 1105

not\_default\_permissions() ->

1089 1106

8#600. %% User read-write-only

1107 +

1108 + req\_id() ->

1109 + ReqId =

1110 + case get(req\_id) of

1111 + undefined -> 0;

1112 + I -> I

1113 + end,

1114 + put(req\_id, ReqId + 1),

1115 + ReqId.

## Comments 0



Please [sign in](#) to comment.