

erlang / otp Public

<> Code Issues 365 Pull requests 145 Actions Projects Wiki

Commit c242e64

u3s committed on Sep 2, 2025 · ✓ 1 / 1 · Verified

ssh: ssh_sftpd verify path size for client data
- reject max_path exceeding the 4096 limit or according to other option value

master (#10155) · OTP-29.0-rc2 ··· OTP-27.3.4.3

1 parent c388a2d commit c242e64

2 files changed +94 -33 lines changed

↑ Top ⚙

Filter files...

- lib/ssh
 - src
 - ssh_sftpd.erl
 - test
 - ssh_sftpd_SUITE.erl

2 files changed +94 -33 lines changed

Search within code ⚙

lib/ssh/src/ssh_sftpd.erl

```

@@ -57,6 +57,7 @@ Specifies a channel process to handle an SFTP subsystem.
57 57      file_handler,          % atom() - callback module
58 58      file_state,           % state for the file callback module
59 59      max_files,           % integer >= 0 max no files sent during
                                READDIR
60 +      max_path,           % integer > 0 - max length of path
60 61      options,             % from the subsystem declaration
61 62      handles              % list of open handles
62 63      %% handle is either {<int>, directory, {Path, unread|eof}} or

```

↓ ↑	@@ -86,6 +87,11 @@ Options:	
86	87	limit. If supplied, the number of filenames returned to the SFTP client per
87	88	<code>`REaddir`</code> request is limited to at most the given value.
88	89	
90	91	+ <code>-`max_path`</code> - The default value is <code>`4096`</code> . Positive integer value
91	92	+ represents the maximum path length which cannot be exceeded in
92	93	+ data provided by the SFTP client. (Note: limitations might be also
93	94	+ enforced by underlying operating system)
94	95	+
89	96	- <code>`root`</code> - Sets the SFTP root directory. Then the user cannot see any files
90	97	above this root. If, for example, the root directory is set to <code>`/tmp`</code> , then
91	98	the user sees this directory as <code>`/`</code> . If the user then writes <code>`cd /etc`</code> , the
↓ ↑	@@ -98,6 +104,7 @@ Options:	
98	104	Options :: [{cwd, string()}
99	105	{file_handler, CbMod {CbMod, FileState}}
100	106	{max_files, integer()}
107	107	+ {max_path, integer()}
101	108	{root, string()}
102	109	{sftpd_vsn, integer()}
103	110],
↓ ↑	@@ -149,8 +156,12 @@ init(Options) ->	
149	156	{Root0, State0}
150	157	end,
151	158	MaxLength = proplists:get_value(max_files, Options, 0),
159	159	+ MaxPath = proplists:get_value(max_path, Options, 4096),
152	160	Vsn = proplists:get_value(sftpd_vsn, Options, 5),
153	161	- {ok, State#state{cwd = CWD, root = Root, max_files = MaxLength,
161	162	+ {ok, State#state{cwd = CWD,
162	163	+ root = Root,
163	164	+ max_files = MaxLength,
164	165	+ max_path = MaxPath,
154	166	options = Options,
155	167	handles = [], pending = <<>>,
156	168	xf = #ssh_xfer{vsn = Vsn, ext = []}}.
↓ ↑	@@ -259,6 +270,30 @@ handle_data(Type, ChannelId, Data0, State =	
259	270	#state{pending = Pending}) ->
259	270	handle_data(Type, ChannelId, Data, State#state{pending = <<>>})
260	271	end.

```

261 272
273 + handle_op(Request, ReqId, <<?UINT32(PLen), _/binary>>,
274 +         State = #state{max_path = MaxPath, xf = XF})
275 +     when (Request == ?SSH_FXP_LSTAT orelse
276 +         Request == ?SSH_FXP_MKDIR orelse
277 +         Request == ?SSH_FXP_OPEN orelse
278 +         Request == ?SSH_FXP_OPENDIR orelse
279 +         Request == ?SSH_FXP_READLINK orelse
280 +         Request == ?SSH_FXP_REALPATH orelse
281 +         Request == ?SSH_FXP_REMOVE orelse
282 +         Request == ?SSH_FXP_RMDIR orelse
283 +         Request == ?SSH_FXP_SETSTAT orelse
284 +         Request == ?SSH_FXP_STAT),
285 +         PLen > MaxPath ->
286 +         ssh_xfer:xf_send_status(XF, ReqId, ?SSH_FX_NO_SUCH_PATH,
287 +             "No such path"),
288 +         State;
289 + handle_op(Request, ReqId, <<?UINT32(PLen), _:PLen/binary, ?UINT32(PLen2),
290 +         _/binary>>,
291 +         State = #state{max_path = MaxPath, xf = XF})
292 +     when (Request == ?SSH_FXP_RENAME orelse
293 +         Request == ?SSH_FXP_SYMLINK),
294 +         (PLen > MaxPath orelse PLen2 > MaxPath) ->
295 +         ssh_xfer:xf_send_status(XF, ReqId, ?SSH_FX_NO_SUCH_PATH,
296 +             "No such path"),
297 +         State;
262 297     handle_op(?SSH_FXP_INIT, Version, B, State) when is_binary(B) ->
263 298         XF = State#state.xf,
264 299         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 46 +         max_path/1,
46 47         real_path/1,
47 48         relative_path/1,
48 49         relpath/1,

```

		@@ -71,9 +72,8 @@
71	72	-define(SSH_TIMEOUT, 5000).
72	73	-define(REG_ATTRS, <<0,0,0,1>>).
73	74	-define(UNIX_EPOCH, 62167219200).
74		-
75		- -define(is_set(F, Bits),
76		- ((F) band (Bits)) == (F)).
	75	+ -define(MAX_PATH, 200).
	76	+ -define(is_set(F, Bits), ((F) band (Bits)) == (F)).
77	77	
78	78	%%-----
79	79	%% Common Test interface functions -----
		@@ -86,6 +86,7 @@ all() ->
86	86	[open_close_file,
87	87	open_close_dir,
88	88	read_file,
	89	+ max_path,
89	90	read_dir,
90	91	write_file,
91	92	rename_file,
		@@ -180,7 +181,8 @@ init_per_testcase(TestCase, Config) ->
180	181	{sftpd_vsn, 6}]],
181	182	ssh:daemon(0, [{subsystems, SubSystems} Options]);
182	183	_ ->
183		- SubSystems = [ssh_sftpd:subsystem_spec([])],
	184	+ SubSystems = [ssh_sftpd:subsystem_spec(
	185	+ [{max_path, ?MAX_PATH}]]],
184	186	ssh:daemon(0, [{subsystems, SubSystems} Options])
185	187	end,
186	188	
		@@ -333,6 +335,23 @@ read_file(Config) when is_list(Config) ->
333	335	
334	336	{ok, Data} = file:read_file(FileName).
335	337	
	338	+ %%-----
	339	+ max_path(Config) when is_list(Config) ->
	340	+ PrivDir = proplists:get_value(priv_dir, Config),

```

341 +   FileName = filename:join(PrivDir, "test.txt"),
342 +   {Cm, Channel} = proplists:get_value(sftp, Config),
343 +   %% verify max_path limit
344 +   LongFileName =
345 +       filename:join(PrivDir,
346 +           "t" ++ lists:flatten(lists:duplicate(?MAX_PATH, "e"))
347 +           ++ "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 +
356 +   %%-----
357 +   read_dir(Config) when is_list(Config) ->
358 +       PrivDir = proplists:get_value(priv_dir, Config),
359 +
360 +       @@ -388,35 +407,33 @@ rename_file(Config) when is_list(Config) ->
361 +
362 +       PrivDir = proplists:get_value(priv_dir, Config),
363 +       FileName = filename:join(PrivDir, "test.txt"),
364 +       NewFileName = filename:join(PrivDir, "test1.txt"),
365 +
366 +       ReqId = 0,
367 +
368 +       LongFileName =
369 +           filename:join(PrivDir,
370 +               "t" ++ lists:flatten(lists:duplicate(?MAX_PATH, "e"))
371 +               ++ "st.txt"),
372 +
373 +       {Cm, Channel} = proplists:get_value(sftp, Config),
374 +
375 +       -
376 +       {ok, <<?SSH_FXP_STATUS, ?UINT32(ReqId),
377 +           ?UINT32(?SSH_FX_OK), _/binary>>, _} =
378 +       rename(FileName, NewFileName, Cm, Channel, ReqId, 6, 0),
379 +
380 +       -
381 +       NewReqId = ReqId + 1,
382 +
383 +       -
384 +       {ok, <<?SSH_FXP_STATUS, ?UINT32(NewReqId),
385 +           ?UINT32(?SSH_FX_OK), _/binary>>, _} =
386 +       rename(NewFileName, FileName, Cm, Channel, NewReqId, 6,
387 +           ?SSH_FXP_RENAME_OVERWRITE),

```

```

404 -
405 -     NewReqId1 = NewReqId + 1,
406 -     file:copy(FileName, NewFileName),
407 -
408 -     %% No overwrite
409 -     {ok, <<?SSH_FXP_STATUS, ?UINT32(NewReqId1),
410 -         ?UINT32(?SSH_FX_FILE_ALREADY_EXISTS), _/binary>>, _} =
411 -     rename(FileName, NewFileName, Cm, Channel, NewReqId1, 6,
412 -         ?SSH_FXP_RENAME_NATIVE),
413 -
414 -     NewReqId2 = NewReqId1 + 1,
415 -
416 -     {ok, <<?SSH_FXP_STATUS, ?UINT32(NewReqId2),
417 -         ?UINT32(?SSH_FX_OP_UNSUPPORTED), _/binary>>, _} =
418 -     rename(FileName, NewFileName, Cm, Channel, NewReqId2, 6,
419 -         ?SSH_FXP_RENAME_ATOMIC).
420 +
421 +     Version = 6,
422 +     [begin
423 +         case Action of
424 +             {Code, AFile, BFile, Flags} ->
425 +                 ReqId = req_id(),
426 +                 ct:log("ReqId = ~p,~nCode = ~p,~nAFile = ~p,~nBFile =
427 + ~p,~nFlags = ~p",
428 +                     [ReqId, Code, AFile, BFile, Flags]),
429 +                 {ok, <<?SSH_FXP_STATUS, ?UINT32(ReqId), ?UINT32(Code),
430 + _/binary>>, _} =
431 +                 rename(AFile, BFile, Cm, Channel, ReqId, Version,
432 +                     Flags);
433 +             {file_copy, AFile, BFile} ->
434 +                 {ok, _} = file:copy(AFile, BFile)
435 +         end
436 +     end ||
437 +     Action <-
438 +     [{?SSH_FX_OK, FileName, NewFileName, 0},
439 +      {?SSH_FX_OK, NewFileName, FileName, ?SSH_FXP_RENAME_OVERWRITE},
440 +      {file_copy, FileName, NewFileName},
441 +      %% no overwrite
442 +      {?SSH_FX_FILE_ALREADY_EXISTS, FileName, NewFileName, ?
443 + SSH_FXP_RENAME_NATIVE},

```

```

433 +      {?SSH_FX_OP_UNSUPPORTED, FileName, NewFileName, ?
      SSH_FXP_RENAME_ATOMIC},
434 +      %% max_path
435 +      {?SSH_FX_NO_SUCH_PATH, FileName, LongFileName, 0}]],
436 +      ok.
420 437
421 438      %%-----
422 439      mk_rm_dir(Config) when is_list(Config) ->
      ↓
      ↑
      @@ -1078,3 +1095,12 @@ encode_file_type(Type) ->
1078 1095
1079 1096      not_default_permissions() ->
1080 1097          8#600. %% User read-write-only
1098 +
1099 + req_id() ->
1100 +     ReqId =
1101 +         case get(req_id) of
1102 +             undefined -> 0;
1103 +             I -> I
1104 +         end,
1105 +         put(req_id, ReqId + 1),
1106 +         ReqId.

```

Comments 0



Please [sign in](#) to comment.