/system/bin/linker64

Android

oK1W\*

\_\_libc\_init

\_ZNSt3\_\_1lsIcNS\_11char\_traitsIcEENS\_9allocatorIcEEEERNS\_13basic\_ostreamIT\_T0\_EES9\_RKNS\_12basic\_stringIS6\_S7\_T1\_EE

\_ZNSt3\_\_124\_\_put\_character\_sequenceIcNS\_11char\_traitsIcEEEERNS\_13basic\_ostreamIT\_T0\_EES7\_PKS4\_m

\_ZN7android4base11GetPropertyERKNSt3\_\_112basic\_stringIcNS1\_11char\_traitsIcEENS1\_9allocatorIcEEEES9\_

\_ZN7android4base11SetPropertyERKNSt3\_\_112basic\_stringIcNS1\_11char\_traitsIcEENS1\_9allocatorIcEEEES9\_

\_ZN7android4base9ShouldLogENS0\_11LogSeverityEPKc

\_ZN7android4base10LogMessageC1EPKcjNS0\_11LogSeverityES3\_i

\_ZN7android4base10LogMessage6streamEv

\_ZNSt3\_\_113basic\_ostreamIcNS\_11char\_traitsIcEEElsEj

\_ZN7android4base10LogMessageD1Ev

\_ZN7android4base5SplitERKNSt3\_\_112basic\_stringIcNS1\_11char\_traitsIcEENS1\_9allocatorIcEEEES9\_

\_Z21ReadMiscMemtagMessageP19misc\_memtag\_messagePNSt3\_\_112basic\_stringIcNS1\_11char\_traitsIcEENS1\_9allocatorIcEEEE

\_Z22WriteMiscMemtagMessageRK19misc\_memtag\_messagePNSt3\_\_112basic\_stringIcNS2\_11char\_traitsIcEENS2\_9allocatorIcEEEE

getopt

\_ZNSt3\_\_113basic\_ostreamIcNS\_11char\_traitsIcEEElsEi

ftruncate

\_\_errno

\_\_open\_2

\_ZdlPv

\_ZNKSt3\_\_121\_\_basic\_string\_commonILb1EE20\_\_throw\_length\_errorEv

abort

\_Znwm

strlen

\_ZNSt3\_\_112basic\_stringIcNS\_11char\_traitsIcEENS\_9allocatorIcEEE6appendEPKc

\_ZNSt3\_\_112basic\_stringIcNS\_11char\_traitsIcEENS\_9allocatorIcEEE6assignEPKc

memcmp

\_ZNSt3\_\_113basic\_ostreamIcNS\_11char\_traitsIcEEE6sentryC1ERS3\_

\_ZNSt3\_\_113basic\_ostreamIcNS\_11char\_traitsIcEEE6sentryD1Ev

\_ZNKSt3\_\_18ios\_base6getlocEv

\_ZNSt3\_\_16localeD1Ev

\_ZNKSt3\_\_16locale9use\_facetERNS0\_2idE

\_ZNSt3\_\_18ios\_base5clearEj

\_ZNKSt3\_\_112basic\_stringIcNS\_11char\_traitsIcEENS\_9allocatorIcEEE7compareEmmPKcm

android\_fdsan\_close\_with\_tag

android\_fdsan\_create\_owner\_tag

close

android\_fdsan\_exchange\_owner\_tag

\_ZN7android4base9ReadFullyENS0\_11borrowed\_fdEPvm

\_ZN7android4base10WriteFullyENS0\_11borrowed\_fdEPKvm

\_ZNSt3\_\_14cerrE

optarg

optind

\_ZNSt3\_\_15ctypeIcE2idE

memcpy

\_\_stack\_chk\_fail

memset

libc.so

LIBC

LIBC\_Q

libbootloader\_message.so

libbase.so

libc++.so

libm.so

libdl.so

APS2

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!!! YOU PROBABLY DO NOT NEED TO USE THIS !!!

!!! USE THE `arm64.memtag.bootctl` SYSTEM PROPERTY INSTEAD. !!!

!!! This program is an implementation detail that is used !!!

!!! by the system to apply MTE settings. !!!

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

force\_on

Check failed: android::base::ReadFully(raw\_fd, m, sizeof(\*m))

s:t:f:

(invalid mode)

memtag-kernel

to

, override:

raw\_fd

memtag-kernel-once

[-s PROPERTY\_NAME]

[-f PROPERTY\_NAME]

[none,][memtag,][memtag-once,][memtag-kernel,][memtag-kernel-once,][memtag-off,]

[default|force\_on|force\_off]

[-t PATH\_TO\_FAKE\_MISC\_PARTITION]

OPTIONS:

-s PROPERTY\_NAME

Sets the system property 'PROPERTY\_NAME' to the new MTE mode (if provided), or to

the current value from the /misc partition.

-f PROPERTY\_NAME

Used in combination with -s without a new MTE mode and sets the system property

'PROPERTY\_NAME' to 1 after reading the current value from the /misc partition

[none,][memtag,][memtag-once,][memtag-kernel,][memtag-kernel-once,][memtag-off,]

A set of MTE options to be applied, if provided. Multiple options may be

specified as a ','-delimited list, e.g. 'memtag,memtag-kernel'.

The options are described below:

- none: default settings for MTE for the product will be applied on next

reboot.

- memtag: MTE is persistently enabled in userspace upon the next reboot.

- memtag-once: MTE is enabled in userspace, only for the next reboot.

- memtag-kernel: MTE is persistently enabled in the kernel upon the next

reboot.

- memtag-kernel-once: MTE is enabled in the kernel, only for the next reboot.

- memtag-off: MTE is persistently disabled in both userspace and kernel upon

the next reboot.

- forced: the current state is the result of force\_on or force\_off in the next

argument. When the next argument is set back to "default", the

state will be cleared.

[default|force\_on|force\_off]

An alternative method of configuring the MTE options to be applied, if provided.

This control is generally to be used by device\_config only, and it overwrites

the previously described settings that are expected to be utilized by the user.

The options are described below:

- default: This flag is not overwriting the MTE mode, and so the setting

should be inherited from the userspace controls (if present), or the

default value from the bootloader's ROM.

- force\_on: MTE is persistently enabled in userspace, overwriting the userspace

setting.

- force\_off: MTE is persistently disabled in userspace and the kernel,

overwriting the userspace setting.

Check failed: UpdateProp(set\_prop, m)

USAGE:

default

Check failed: android::base::WriteFully(raw\_fd, &m, sizeof(m))

system/extras/mtectrl/mtectrl.cc

(ftruncate(raw\_fd, sizeof(misc\_memtag\_message))=

mode:

memtag

. Ignoring and setting

Check failed: UpdateProp(set\_prop, {})

none

(raw\_fd=

Check failed:

override:

, -1=

Failed to apply mode:

memtag-once

MTE mode in misc message contained unknown bits:

memtag-off

Unknown value for mode:

forced

force\_off

ftruncate(raw\_fd, sizeof(misc\_memtag\_message))

Check failed: value

Applied

(invalid override)

Failed to read memtag message:

Partially applied

#@9h

c@9h

#@9h

#@9h

#@9h

?k78

.text

.got

.note.android.ident

.got.plt

.rela.plt

.dynstr

.eh\_frame\_hdr

.gnu.version\_r

.interp

.data.rel.ro

.relr.dyn

.rela.dyn

.gnu.version

.dynsym

.gnu.hash

.eh\_frame

.note.gnu.build-id

.dynamic

.shstrtab

.tdata

.rodata

.gnu\_debugdata

.data

7zXZ

E<+o

\lf+E

,#vU9

EW7\

?g(d

3< "

!`;@

(JK!

b'Np

WQB

$3f-

U&2{

NRE

t-@0c|

shiba:/ # cat system/etc/init/mtectrl.rc

# Copyright (C) 2022 The Android Open Source Project

#

# Licensed under the Apache License, Version 2.0 (the "License");

# you may not use this file except in compliance with the License.

# You may obtain a copy of the License at

#

# http://www.apache.org/licenses/LICENSE-2.0

#

# Unless required by applicable law or agreed to in writing, software

# distributed under the License is distributed on an "AS IS" BASIS,

# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

# See the License for the specific language governing permissions and

# limitations under the License.

on property:arm64.memtag.bootctl=\* && property:ro.arm64.memtag.bootctl\_supported=1

wait\_for\_prop arm64.memtag.bootctl\_loaded 1

exec -- /system/bin/mtectrl ${arm64.memtag.bootctl:-none} ${persist.device\_config.runtime\_native\_boot.bootloader\_override:-default}

on property:persist.device\_config.runtime\_native\_boot.bootloader\_override=\* && property:ro.arm64.memtag.bootctl\_supported=1

wait\_for\_prop arm64.memtag.bootctl\_loaded 1

exec -- /system/bin/mtectrl ${arm64.memtag.bootctl:-none} ${persist.device\_config.runtime\_native\_boot.bootloader\_override:-default}

# adbd gets initialized in init, so run before that. this makes sure that the

# user does not change the value before we initialize it

on early-boot && property:ro.arm64.memtag.bootctl\_supported=1

exec\_background -- /system/bin/mtectrl -s arm64.memtag.bootctl -f arm64.memtag.bootctl\_loaded

on shutdown && property:ro.arm64.memtag.bootctl\_supported=1

# This doesn't use wait\_for\_prop to not stall the shutdown.

exec -- /system/bin/mtectrl ${arm64.memtag.bootctl:-none} ${persist.device\_config.runtime\_native\_boot.bootloader\_override:-default}

on property:persist.device\_config.runtime\_native\_boot.bootloader\_override=force\_on

setprop persist.sys.mte.permissive 1

# mtectrl.cc

https://android.googlesource.com/platform/system/extras/+/main/mtectrl/mtectrl.cc