Page Table Check

Introduction

Page table check allows to harden the kernel by ensuring that some types of the memory corruptions are prevented.

Page table check performs extra verifications at the time when new pages become accessible from the userspace by getting their page table entries (PTEs PMDs etc.) added into the table.

In case of detected corruption, the kernel is crashed. There is a small performance and memory overhead associated with the page table check. Therefore, it is disabled by default, but can be optionally enabled on systems where the extra hardening outweighs the performance costs. Also, because page table check is synchronous, it can help with debugging double map memory corruption issues, by crashing kernel at the time wrong mapping occurs instead of later which is often the case with memory corruptions bugs.

Double mapping detection logic

Current Mapping	New mapping	Permissions	Rule
Anonymous	Anonymous	Read	Allow
Anonymous	Anonymous	Read / Write	Prohibit
Anonymous	Named	Any	Prohibit
Named	Anonymous	Any	Prohibit
Named	Named	Any	Allow

Enabling Page Table Check

Build kernel with:

- PAGE_TABLE_CHECK=y Note, it can only be enabled on platforms where ARCH_SUPPORTS_PAGE_TABLE_CHECK is available.
- Boot with 'page_table_check=on' kernel parameter.

Optionally, build kernel with PAGE_TABLE_CHECK_ENFORCED in order to have page table support without extra kernel parameter.