**# Title**: Forward Control Flow Guard Instruction runtime indicator

**# Status**: Submitted to industry standard forum

**# Document**: UEFI Specification Version 2.9

**# License**: SPDX-License-Identifier: CC-BY-4.0

**# Submitter**: [TianoCore Community](<https://www.tianocore.org>)

**# Summary of the change**

**[Background]**

Current X86 and ARM processors support hardware CPU based forward control flow guard, such as

* **Intel Control Flow Enforcement Technology (CET) - Indirect Branch Tracking (IBT)**. It puts **ENDBR** instruction at the jump target.
* **ARM v8.5 – Branch Target Identification (BTI)**. It puts **BTI** instruction at jump target.

An OS (such as Linux) may enable those hardware control flow to harden the OS kernel.

The UEFI Runtime Service might become a problem, because a UEFI firmware may or might not enable the forward control flow instruction in the runtime service image.

For legacy compatibility consideration, the OS may have to put the Runtime Service to a legacy code page region.

The long term should be: OS gets the BIOS enabling information to know if the Runtime Service code includes such forward control flow guard instruction (such as, x86 CET-IBT or ARM BTI).

**[Proposal]**

This proposal adds one bit information to report to the OS – if the BIOS runtime service code includes the forward control flow guard instruction.

Reference:

1. UEFI Specification 2.9 - www.uefi.org
2. Intel 64 and IA-32 Architectures Software Developer’s Manual (2021)
3. ARM Architecture Reference Manual (2019)

**# Benefits of the change**

The OS kernel may use this information to setup the forward control flow guard policy for runtime code.

**# Impact of the change**

This is an optional feature.

For compatibility, if the old kernel does not activate the CET-IBT but the new BIOS enabled CET-IBT, then the ENDBRANCH becomes NOP.

# Detailed description of the change [normative updates]

**UEFI Specification**

**4.6 EFI Configuration Table & Properties Table**

**EFI\_MEMORY\_ATTRIBUTES\_TABLE**

…

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
/\* EFI\_MEMORY\_ATTRIBUTES\_TABLE  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
typedef struct {  
UINT32** *Version* ***;*UINT32** *NumberOfEntries* ***;*UINT32** *DescriptorSize* **;  
UINT32** *ReservedFlags***;  
// EFI\_MEMORY\_DESCRIPTOR** *Entry [1]***;  
} EFI\_MEMORY\_ATTRIBUTES\_TABLE;**

*Version* The version of this table. Present version is 0x000000012

**…**

*ReservedFlags* Reserved bytesFlags to provide more information for the memory attributes.

#define EFI\_MEMORY\_ATTRIBUTES\_FLAGS\_RT\_FORWARD\_CONTROL\_FLOW\_GUARD 0x1

// BIT0 implies that Runtime code includes the forward control flow guard

// instruction, such as X86 CET-IBT or ARM BTI.

**# Special Instructions**

NO