black hat USA 2021
August 4-5, 2021
BRIEFINGS

PCIe Device Attacks: Beyond DMA

Exploiting PCIe switches, messages and errors

Hareesh Khattri; Nagaraju N Kodalapura (Raju); Nam N Nguyen;
Intel Corporation

Who We Are

Hareesh Khattri

Nagaraju N Kodalapura

Nam N Nguyen

- Offensive Security Researchers @ Intel
- Security research activities targeting datacenter platforms supporting Trusted Computing and Virtualization Technologies
- Contributed to a high-severity Intel Security Advisories (SA-00255, SA-00319, SA-00368), resulting in mitigation for 11 CVEs

AGENDA

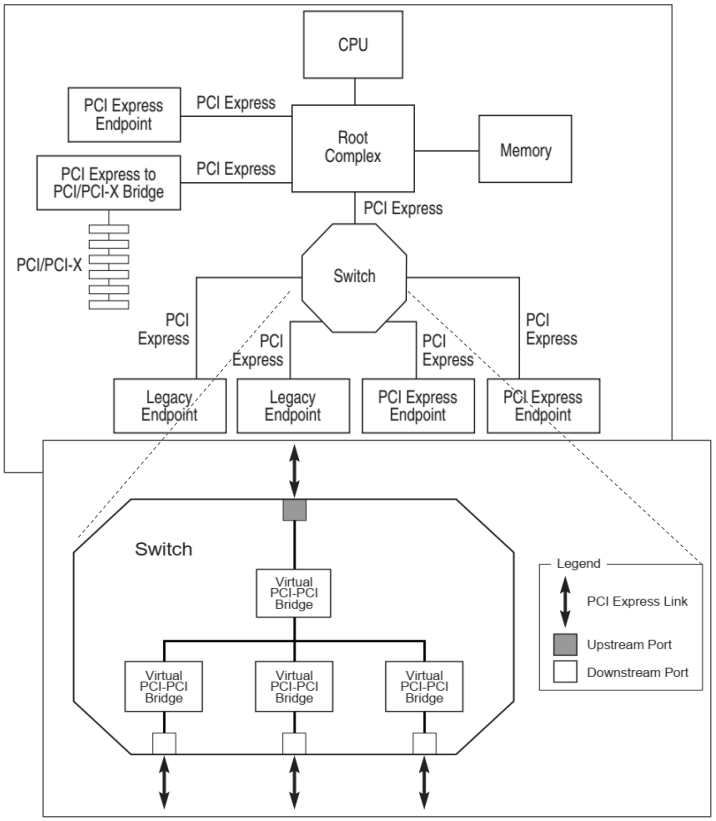
- **Introduction**
  - PCIe Switches overview
  - SRIOV Overview
  - Threat Model
- **PCIe Switches**
  - General hardware design
  - Attacking EEPROM
  - PCIe TLP Injector
  - Beyond DMA
- **PCIe Errors & SRIOV**
  - PCIe Advance error reporting (AER)
  - SRIOV DOS attack
- **Summary & Mitigations**

Introduction

PCIe Switches – Overview

- PCIe Root Complex (RC)
  - Root port of PCIe tree that connects CPU to IO peripherals
  - Can support Peer-to-peer communication between devices

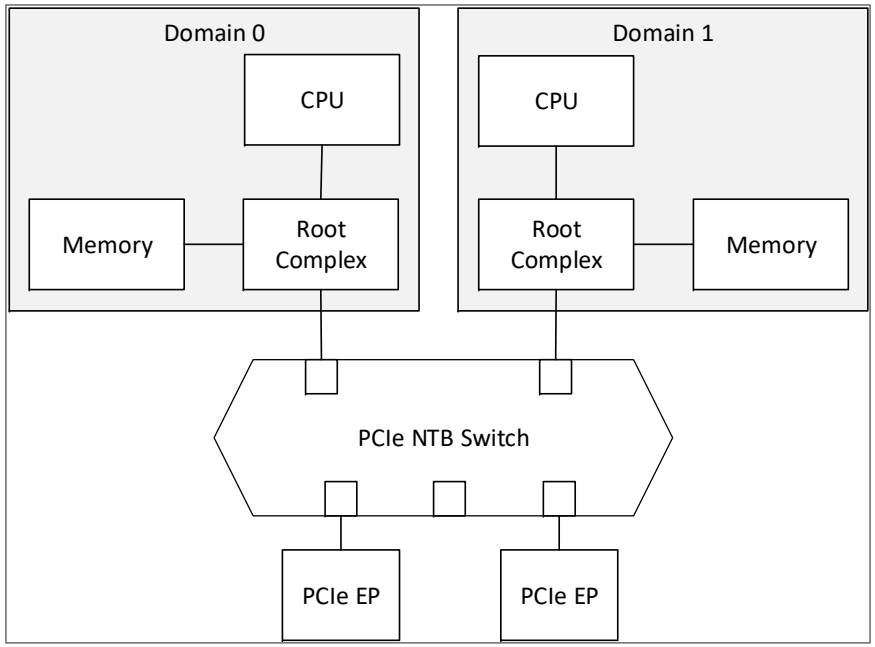
- PCIe Switches
  - "*A logical assembly of multiple virtual PCI-to-PCI Bridge devices*"
  - At software level appear as multiple PCIe bridges
  - Enable connecting and expanding PCIe tree by exposing more PCIe ports for pluggable PCIe devices.
  - Must support forwarding of all PCIe TLP packets between RC and PCIe Endpoint



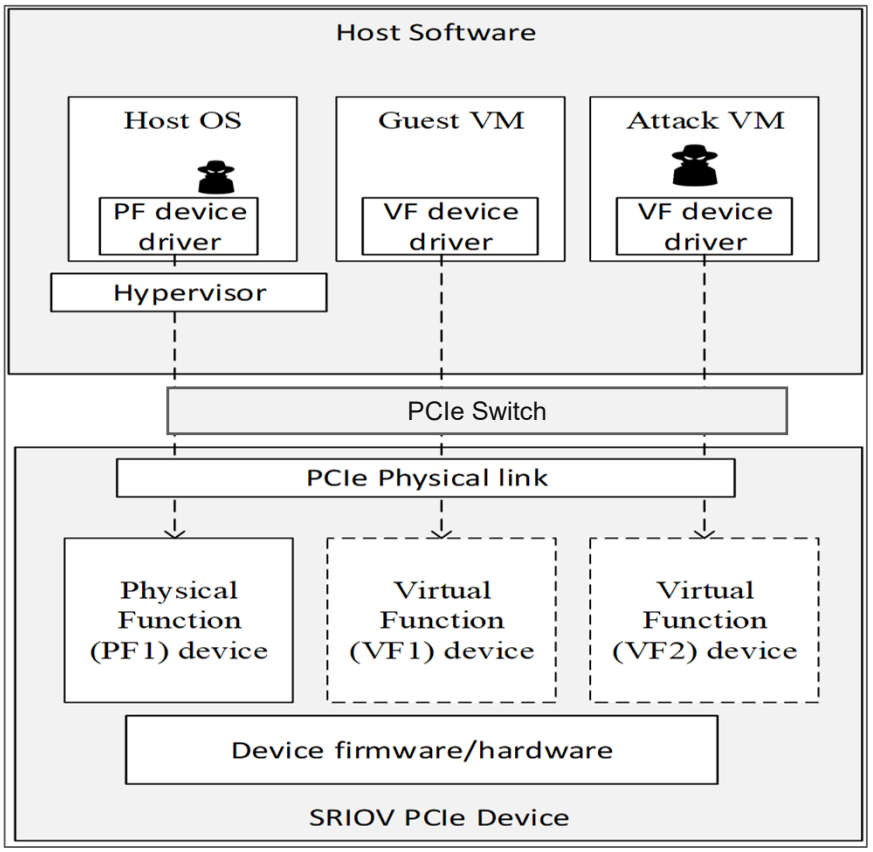
Pcie Base Specification 5.0

PCIe Switches – Overview

- PCIe Switches: Transparent vs Non-Transparent bridges
  - Transparent bridge: The PCIe switch exposes all connected downstream EPs to the RC directly in transparent mode
  - Non-Transparent bridge (NTB): The PCIe switch appears as EP device and connect two different PCIe busses with different RC
- NTB generally used to enable multiple RC connections and bridging



Threat Model

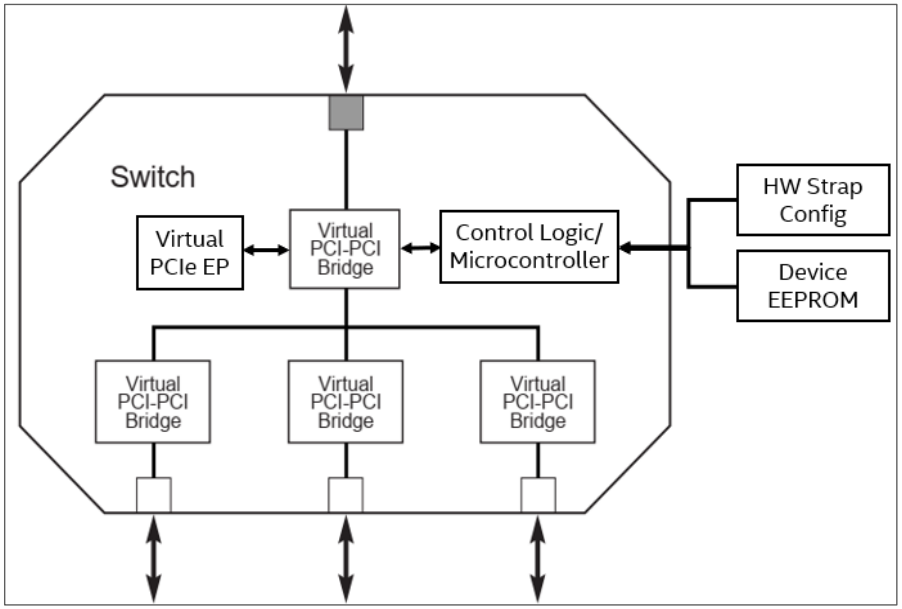


- Adversary Model
  - Malicious PCIe device targeting host OS & platform assets
  - Malicious Virtual machine (VM)/ virtual function (VF): targeting other VMs, Physical Function (PF) and platform hardware
  - Malicious PF targeting other platform trusted HW/FW components
  - System software (BIOS/ Hypervisor)
  - Physical attacker
- Let's look at a few attack vectors:
  - Malicious PF/ Host VM
    - Attacking EEPROM
    - PCIe TLP Injector
    - Beyond DMA
  - Malicious Guest VM: Injecting PCIe Errors w/ SRIOV

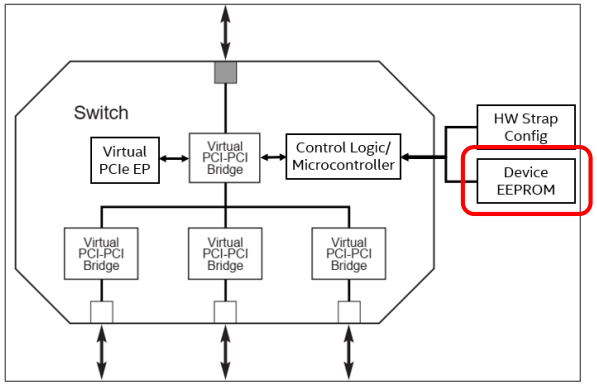
PCIe Switches

PCIe Switches – Hardware Design

- Switch controls
  - Exposed to Software through PCIe config space or MMIO mapped registers. Hypervisor will only allow EP to be exposed to Guest VMs
- Switch Configuration
  - EEPROM Data and hardware straps
  - Device firmware and patches (if switch has a microcontroller)
- Enable/disable debug modes
- Potential variations
  - Can support custom PCIe EP within switch used for switch configuration and management
  - Can include a microcontroller within Switch with firmware stored in EEPROM



Attacking EEPROM



EEPROM Data

- Port configuration & physical parameters
- Device firmware (if switch has a microcontroller)
- Software overrides for HW strap settings
- Patches for firmware and configuration

Possible attack: permanent denial of service by tampering with EEPROM contents

https://www.broadcom.com/products/pcie-switches-bridges/pcie-switches/pex8612

Serial EEPROM Random Write/Read Access

To access the serial EEPROM, a PCI Express or I^2C Master uses the following registers (Port 0, and also NT Port Virtual Interface if Port 0 is the NT Port):

- **Serial EEPROM Status and Control** (offset 260h)
- **Serial EEPROM Buffer** (offset 264h)
- **Serial EEPROM 3rd Address Byte** (offset 26Ch)

Table 6-1. Serial EEPROM Data

Location	Value	Description
0h	5Ah	Validation Signature
1h	00h	***Reserved***
2h	REG_BYTE_COUNT (LSB)	Configuration register Byte Count (LSB)
3h	REG_BYTE_COUNT (MSB)	Configuration register Byte Count (MSB)
4h	REGADDR (LSB)	1st Configuration Register Address (LSB)
5h	REGADDR (MSB)	1st Configuration Register Address (MSB)
6h	REGDATA (Byte 0)	1st Configuration Register Data (Byte 0)
7h	REGDATA (Byte 1)	1st Configuration Register Data (Byte 1)
8h	REGDATA (Byte 2)	1st Configuration Register Data (Byte 2)
9h	REGDATA (Byte 3)	1st Configuration Register Data (Byte 3)

Attacking EEPROM

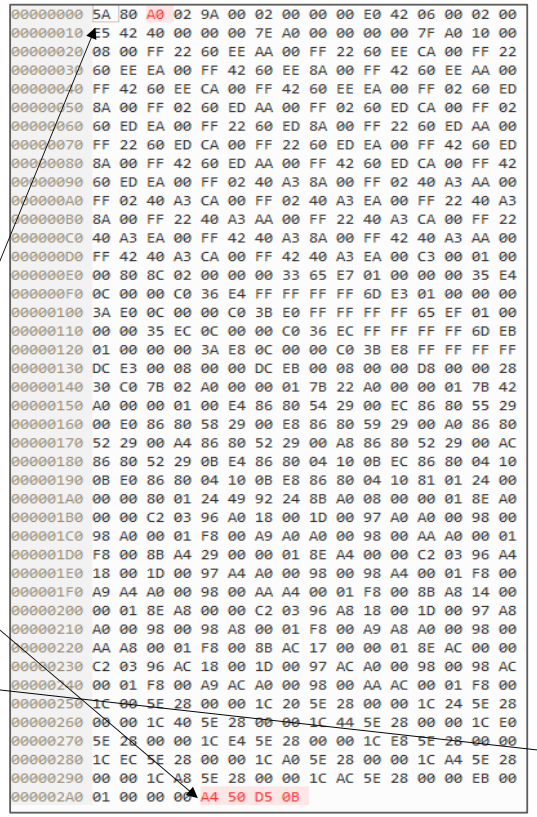
Table 6-1. Serial EEPROM Data

Location	Value	Description
0h	5Ah	Validation Signature
1h	00h	***Reserved***
2h	REG_BYTE_COUNT (LSB)	Configuration register Byte Count (LSB)
3h	REG_BYTE_COUNT (MSB)	Configuration register Byte Count (MSB)
4h	REGADDR (LSB)	1st Configuration Register Address (LSB)
5h	REGADDR (MSB)	1st Configuration Register Address (MSB)
6h	REGDATA (Byte 0)	1st Configuration Register Data (Byte 0)
7h	REGDATA (Byte 1)	1st Configuration Register Data (Byte 1)
8h	REGDATA (Byte 2)	1st Configuration Register Data (Byte 2)
9h	REGDATA (Byte 3)	1st Configuration Register Data (Byte 3)

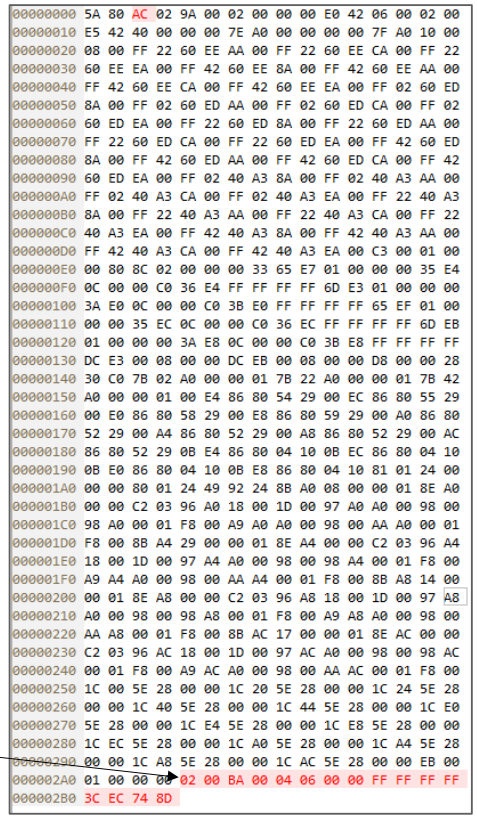
0x5A, 0x80 : EEPROM Signature
0x02,0xA0 : Byte Count

Last 4 bytes image CRC

Add New register writes to:
PCIe Revision ID , Device ID
Update Byte count & CRC



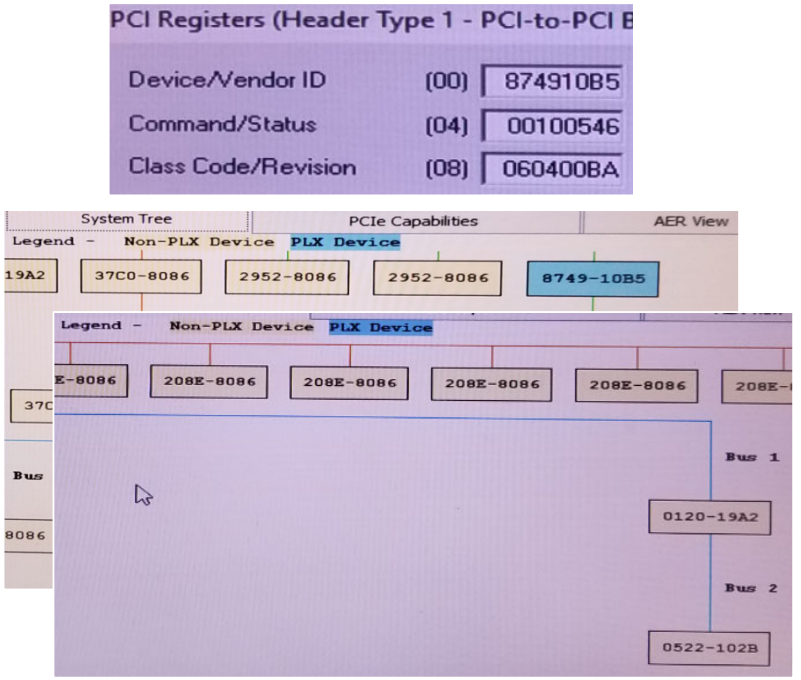
EEPROM Data from PLX Switch



Modified EEPROM Data

Attacking EEPROM

- **Writing to Read-only PCIe config space registers**
  - First tested changing Revision ID ➔ 0xBA
  - Writes using EEPROM method to extend new Register writes works!!
- **Now let's try a more interesting attack**
  - Modify Device ID , Vendor ID to 0xFFs
  - What is this going to result in?
    - 0xFF_FF is defined as Invalid Vendor ID & Device ID by PCI SIG
    - PCIe Switch and all downstream devices no longer show up in PCIe enumeration



After attack: PLX switch is not detected in PCIe enumeration

CVE-2019-14625, CVE-2020-8677

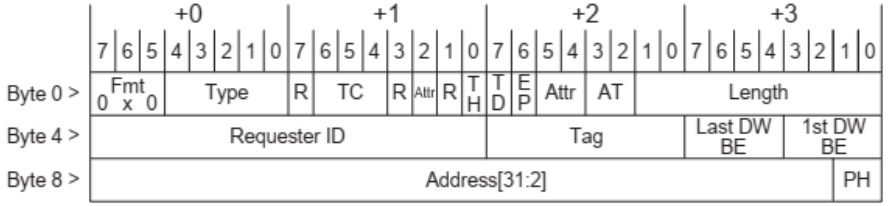
Impact: Persistent DoS on PCIe switch that is not SW recoverable

PCIe Switches – TLP Injector

- **PCIe Transaction layer packets (TLP)**
  - PCIe communication messages that define transaction types, address, size, data etc.
  - Raw TLP injection ability is key feature used in Rogue PCIe device attacks to target memory

- **PCIe Switches Bridge PCIe messages**
  - Must support forwarding of all PCIe TLP packets between RC and PCIe Endpoint
  - So, can we use the PCIe switches as a MiTM to modify/ inject PCIe TLP messages?

- **In general, PCIe switch design**
  - Raw TLP injector mode is not likely to be a functionally supported mode.
  - Might be possible through Debug features or internal firmware code execution.



Example PCIe TLP Format

PCIe Switches – TLP Injector

https://www.broadcom.com/products/pcie-switches-bridges/pcie-switches/pex8747
https://docs.broadcom.com/doc/12351854

PCIe Packet Generator
The PEX 8747 features a full-fledged PCIe Packet Generator capable of creating programmable PCIe traffic running at up to Gen 3 speeds and capable of saturating a x16 link. Using PLX's Software Development Kit (www.plxtech.com/sdk), designers can create custom

```
// Using 8747 Packet Generator @ [B3:00.0]
// Reset Generator to put into known state
mmr  570  40010000
// Small delay to let Generator reset
sleep 50
// Cmd #1: Mem Write (32-bit), Pyld:8B, Delay:5clk
// Write TLP & Command to RAM
mmr  56C  DEADBEEF
mmr  56C  DEADBEEF
mmr  56C  00004102
mmr  56C  FF00F0B4
mmr  56C  02000040
mmr  570  0001A000
mmr  56C  80250000
mmr  56C  80004000
mmr  570  00018000
// Start Generator
mmr  570  80050000
```

Packet generator script for Mem write TLP

TLP Bytes DW0 – DW 4
Length 0x02,
FMT 0x40 MemWr , etc
Writes 0xDEADBEEF to
Address 0x0241_0000 (32 bit)

Script writes to Pcie Config register offsets 0x560, 0x56c, 0x570
- These are SW exposed and control the packet generator

- Debug capability for Raw TLP injection is exposed to host software
- Hardware does not support disabling this feature
- CVE-2019-14626, CVE-2020-8676

Impact: Software can inject Raw TLP frames to target other platform components

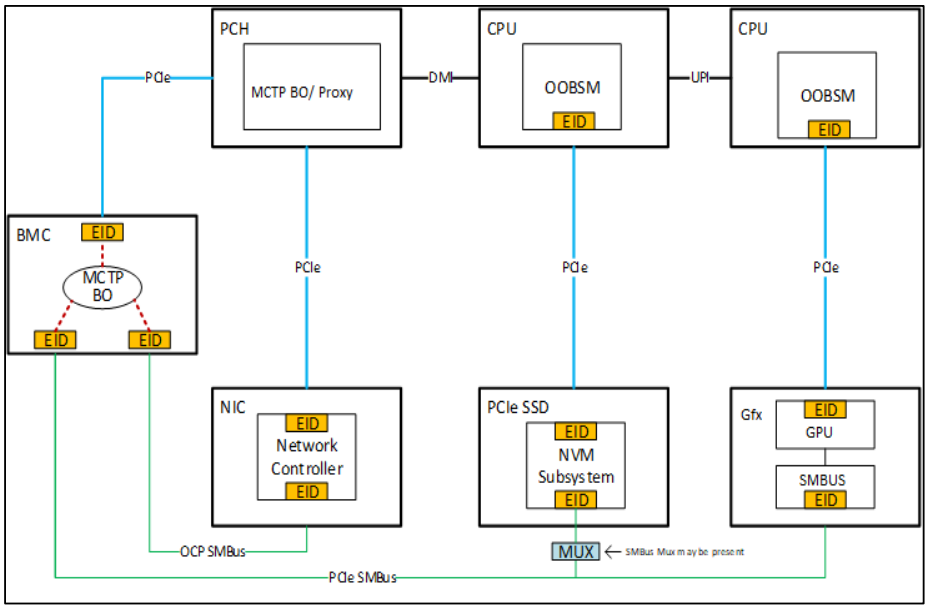
PCIe Switches – Beyond DMA

System Memory DMA, MMIO attacks are known, what else?

Transaction Type	Non-Posted or Posted
Memory Read	Non-Posted
Memory Write	Posted
Memory Read Lock	Non-Posted
IO Read	Non-Posted
IO Write	Non-Posted
Configuration Read (Type 0 and Type 1)	Non-Posted
Configuration Write (Type 0 and Type 1)	Non-Posted
Message	Posted

"The basic premise of MCTP is that higher layer protocols will fulfill security requirements (for example, confidentiality and authentication) for communication of management data. The MCTP protocol itself will not define any additional security mechanisms."

Management Component Transport Protocol (MCTP) uses these messages



Pcie Base Specification 5.0
DMTF, DSP0236, Management Component Transport Protocol (MCTP) Base Specification
DMTF, DSP0238, Management Component Transport Protocol (MCTP) PCIe VDM Transport Binding Specification

PCIe Switches- Beyond DMA

➢ MCTP message-based possible attacks

- Message spoofing: Possible by spoofing source/ destination endpoint ID of MCTP messages.
- Software send BMC privilege level platform management commands : Exploiting PCIe packet injector or other issues in PCIe EP devices
  - Messages can also be sent from PCIe switches

➢ Malformed messages to target MCTP message handling firmware

- This is feasible from PCIe devices or SMBUS devices that can generate MCTP messages.
  - Example: Fragmented MCTP messages using Start of message (SOM)/ End of message (EOM) flags to overflow past allocated memory. Issues in OpenBMC firmware here:
    - https://gerrit.openbmc-project.xyz/#/c/openbmc/libmctp/+/34301/
    - https://gerrit.openbmc-project.xyz/#/c/openbmc/libmctp/+/34300/

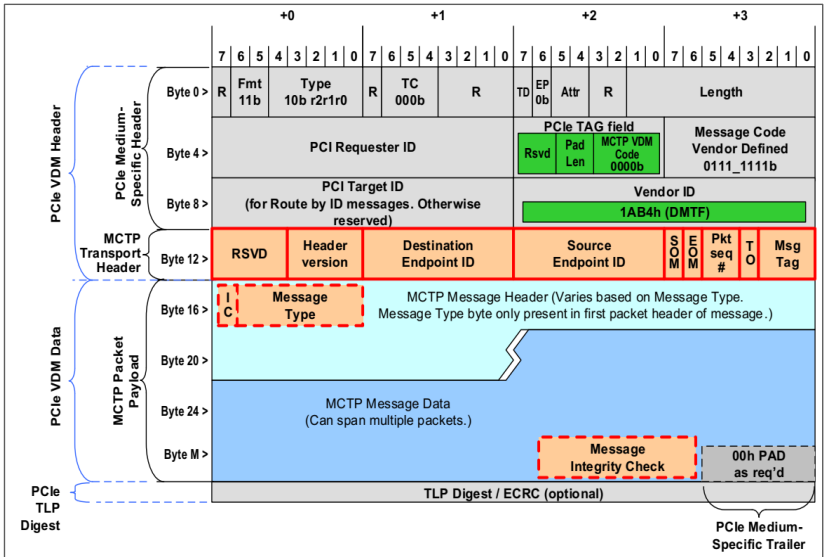


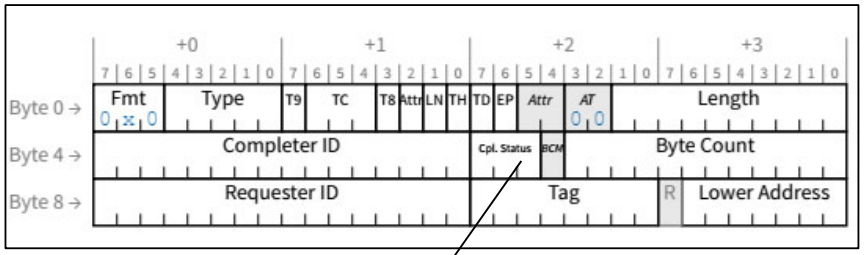
Figure 1 – MCTP over PCI Express Packet Format for PCIe 2.1

DMTF, DSP0238, Management Component Transport Protocol (MCTP) PCIe VDM Transport Binding Specification

PCIe Errors & SRIOV

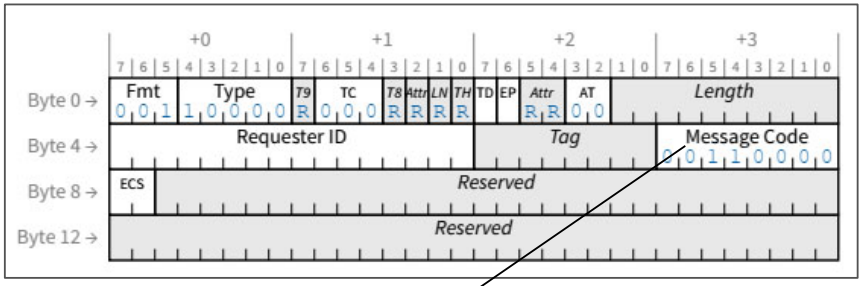
PCIe Error Messages

Errors signaled through Completion TLP



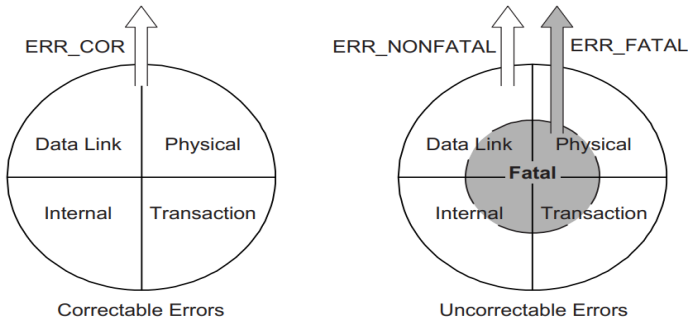
Cpl. Status[2:0] Field Value (b)	Completion Status
000	Successful Completion (SC)
001	Unsupported Request (UR)
010	Configuration Request Retry Status (CRS)
100	Completer Abort (CA)

Errors signaled through Message TLP



Name	Code[7:0] (b)	Routing r[2:0] (b)	Support			
			RC	Ep	Sw	Br
ERR_COR	0011 0000	000	r	t	tr	t
ERR_NONFATAL	0011 0001	000	r	t	tr	t
ERR_FATAL	0011 0011	000	r	t	tr	t

PCIe Advanced Error Reporting (AER)

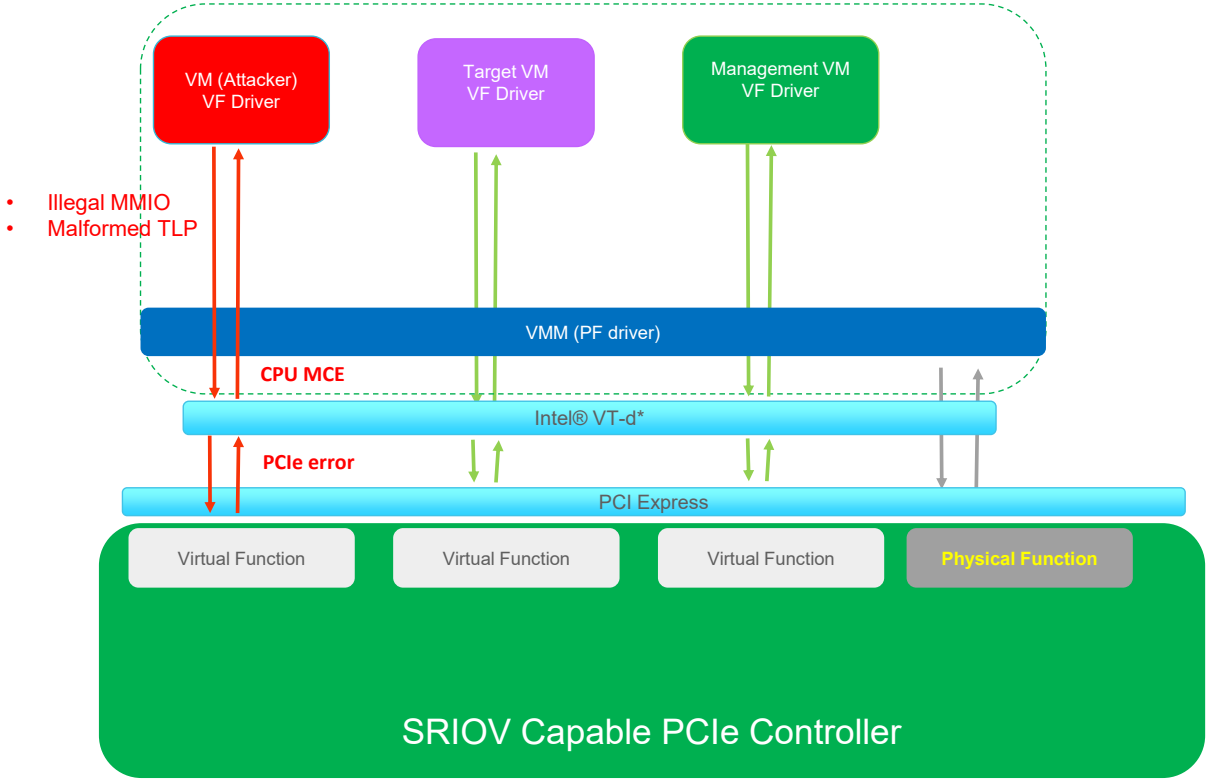


PCIe error classification

- **Uncorrectable Fatal Error**
  - PCIe end point device to send “Fatal Error” to root complex
- **Registers of interest (PCIe 5.0 Spec)**
  - Uncorrectable Error Mask Register
  - Uncorrectable Error Severity

SRIOV Attack Scenarios

- **Attack Scenario #1:** Malicious VF/VM attacking other VFs/VMs and PF
  - CVE-2019-0142, CVE-2019-0150
- **Attack Scenario #2:** Malicious VF/VM abusing its own Hardware (HW) resources to cause Platform DoS
  - CVE-2019-0143 and CVE-2019-0144



SRIOV exploit demo video

Summary & Mitigations

PCIe Switches – Mitigations

General Security protection PCIe switches

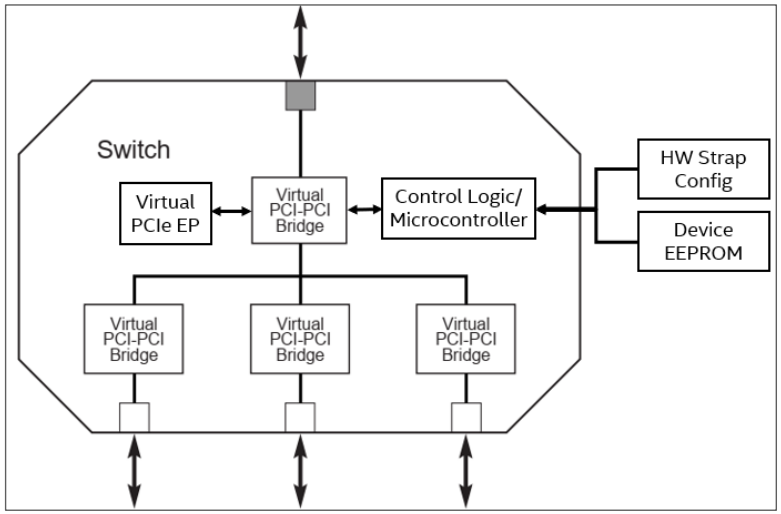
- Secure boot , Secure debug for protection of Switch controls. Supported in newer versions of Switches
- Host Software , Hypervisor , Management VM isolation of switch controls from Guest VM.

Attack 1: Serial EEPROM programming

- Platform board rework can be done to block writes to EEPROM after production using EEPROM write protection
- Hardware/ ROM based recovery mode in PCIe device and switches.

Attack 2: TLP injector

- In PCIe switches such debug modes need to be disabled or authenticated (secure debug)
- Platform can enable IOMMU protections to limit exposed memory regions to PCIe Switch and connected devices
- Enable PCIe Access Control Service Source validation (ACSCtl: SrcValid)
  - Upstream Requestor ID is checked against Secondary/Subordinate bus numbers.

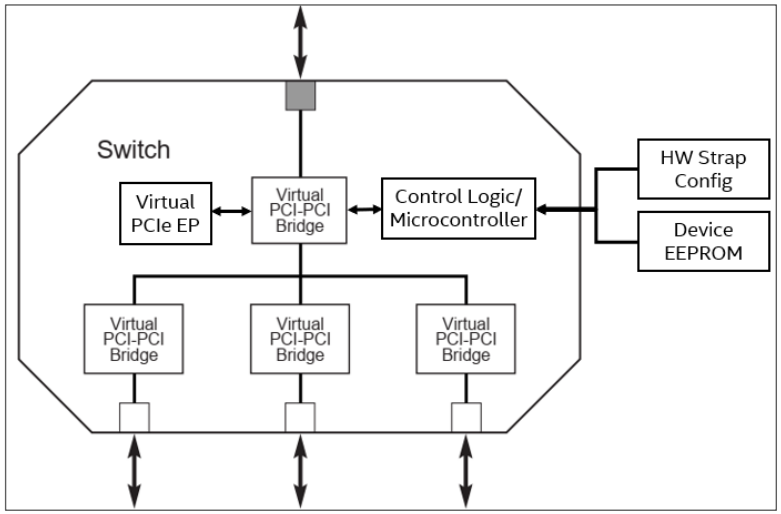


Security Mitigations - SRIOV

Attack Scenario	CVEs	Mitigation
Malicious VF/VM abusing its own HW resources to cause Platform DoS	CVE-2019-0143 CVE-2019-0144	PCIe device should ensure that it doesn't generate unconditional 'Uncorrectable Fatal Error" by default. The error generation should follow the PCIe AER configuration requirements
Malicious VF/VF attacking other VFs/VMs and PF	CVE-2019-0142	PF driver should block a VF driver from issuing high-privileged commands like PF Reset **Acknowledgement:** We would like to acknowledge Mr. Kumar Mangipudi (**narasimhakumar.mangipudi@latticesemi.com**) for his contribution to this CVE while he was part of Intel Corporation
VFID spoofing	CVE-2019-0150	PF driver should validate all the incoming parameters from a VF before consuming them

Other Areas

- **Other areas to investigate for security impact**
  - Peer-2-Peer transactions between devices connected through switches
  - Memory protections and isolations for downstream traffic
  - PCIe new security features:
    - Device security enhancements (DICE)
    - Integrity & Encryption for messages : SPDM, PCIe IDE



Notices and Disclaimers

Intel provides these materials as-is, with no express or implied warranties.

No product or component can be absolutely secure.

Intel and the Intel logo are trademarks of Intel Corporation in the United States and other countries.

*Other names and brands may be claimed as the property of others.

Q & A