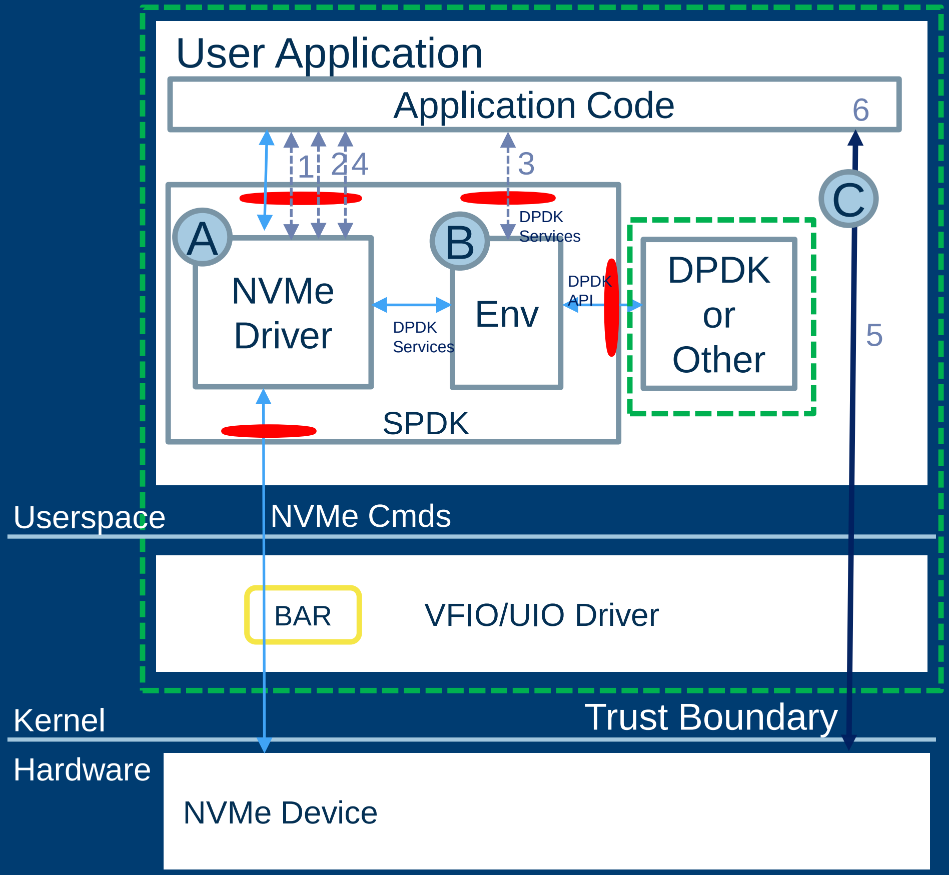
USE CASE: NVME DRIVER INTEGRATION

- A community member has decided to use the SPDK NVMe driver and integrate it directly into their application without leveraging any of the other SPDK components other than the environment abstraction layer which can be used w/DPDK or their own equivalent.
- No other SPDK components are used (bdev not included, etc).

USE CASE: NVME DRIVER INTEGRATION (SYSTEM DIAGRAM)



Assets:
A) NVMe Driver
B) Env layer
C) Data

High Level Flow:
1) App initializes NVMe controller
2) App allocates queue pairs for submitting and completing Ios, starts polling CQ for new entries
3) App allocates memory for data
4) App creates SQ entry and submits to NVMe driver
5) NVMe device transfers data and adds an entry to CQ when complete
6) App sees completion on CQ

Control/Function Calls ↔
Data ↔
Attack Surfaces ⬬
Trust Boundaries - - -

USE CASE: NVME DRIVER INTEGRATION (ATTACK SURFACES)

System Element	Compromise Type(s)	Assets exposed	Attack Method
App/NVMe Driver Interface	Invalid initialization, bad SQ/CQ entries	Data, NVMe Driver	Invalid/intercept data buffers in SQ/CQ entries
App/Env Interface	Invalid memory allocation, device enum	Data	Intercept data buffers
NVMe Driver/Env Interface	Invalid memory allocation, device enum	Data, NVMe driver	Corrupt the application with bad control information

USE CASE: NVME DRIVER INTEGRATION (THREAT MATRIX)

Assets Surface	NVMe Driver	Env	Data
App/NVMe Driver Interface	Yes	No	Yes
App/Env Interface	No	Yes	Yes
NVMe Driver/Env Interface	Yes	Yes	Yes

ADVERSARIES IN SCOPE

Persona	Motivation	Attacker Type	Starting Privilege Level	Skill and Potential Effort level
Malicious System User	Wants to snoop data/disrupt users on system	Unprivileged Software Adversary	User-level privileges	Proficient level of skill does not give up easily
Internet Attacker	Denial Of Service	Network Adversary	None	Unskilled, gives up easily

*system software adversary is out of scope because such adversaries have the permissions to defeat mitigations. The customer needs to ensure appropriate deployment policies are in place to prevent system level software adversaries

THREAT/ATTACK SURFACE MATRIX

Asset\Attack Surface	Network Interface	Vhost-Virtio	PF/VF mailbox	Filesystem	Unix fifo	virtio serial link
SPDK Application/SPDK Libs	Y (1)	Y (1)	Y (2)		Y (3)	Y (3)
Other Applications/VM's			Y (4)		Y (3)	Y (3)
CPU availability	Y (1)	Y (1)		Y (5)	Y (6)	Y (6)
CPU frequency				Y (5)	Y (3)	Y (3)
DPDK PMDS	Y (1)	Y (1)	Y (2)	Y (7)		
eBPF bytecode				Y (8)		
NIC resources	Y (9)		Y (2)	Y (7)		
Process memory Eg routing tables, encryption keys	Y (1)	Y (1)	Y (4)	Y (10)	Y (6)	Y (6)

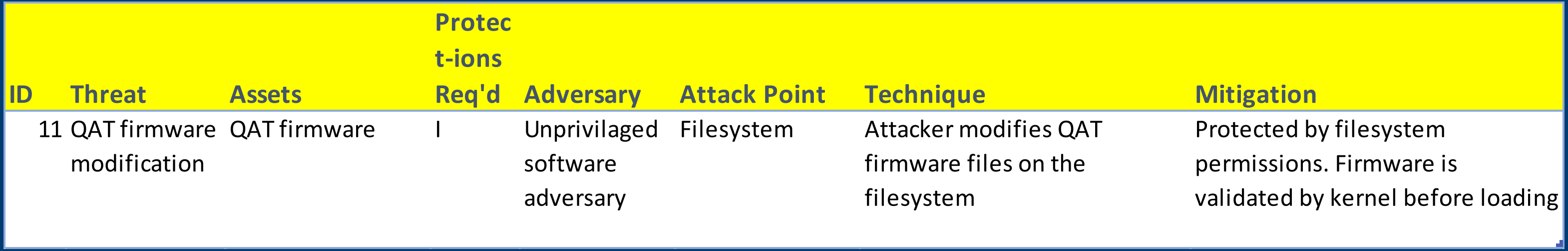
THREATS

ID	Threat	Assets	Protect-ions Req'd	Adversary	Attack Point	Technique	Mitigation
1	Bad Network Data	SPDK App, CPU availability, SPDK libs and DPDK PMDs, Process Memory	A C	Network Adversary	Network data, Vhost-virtio data, QAT data	Insert malformed packets/data causing buffer overflow or crash or other error, e.g. infinite loop, in application	SW to perform input validation checking on received data before use.
2	Invalid VF request	SPDK App, DPDK PMDs, NIC resources	A	Unprivilaged software adversary	NIC PF-VF mailbox	VF sends an invalid or illegal request to the PF causing crash of application/PF driver	SW to perform input validation on received mailbox messages when processing
3	CPU throttling	SPDK App, Other Apps/VMs, CPU frequency, SPDK libraries	A	Unprivilaged software adversary	Unix Fifo, virtio-serial link	Malicious container/VM sends a CPU throttling request for an application CPU, or CPU belonging to another VM/container	For VM's use one virtio-serial link per VM to prevent spoofing. CPU numbers are logical per VM, and need translation so VM cannot refer to another VM's cores. For containers/apps use one unix FIFO per app. Power management app must validate CPUs in request via SW
4	VF redirection	Other Apps/VMs, process memory	C	Unprivilaged software adversary	NIC PF-VF mailbox	VF makes request to trigger traffic from another VF to be routed to it instead, causing data leak and/or denial of service	HW provides one mailbox per VF preventing spoofing. SW to disallow one VF requesting resources for another

THREATS

ID	Threat	Assets	Protections Req'd	Adversary	Attack Point	Technique	Mitigation
5	CPU manipulation	CPU availability, CPU freqency	A	Unprivilaged software adversary	Filesystem	Attacker accesses sysfs and modifies the CPU parameters to hotplug out a cpu or change its freqency	Protected via OS permissions. Root access generally required
6	Invalid power request	CPU availability, process memory	A C	Unprivilaged software adversary	Unix Fifo, virtio-serial link	Invalid request sent, causing power management application to misbehave, e.g. via buffer overflow	SW to validate all inputs in range
7	PCI BAR access	DPDK PMDs, NIC resources, QAT availability	A C	Unprivilaged software adversary	Filesystem	Attacker accesses the PCI BARs through sysfs and uses those to manipulate the NIC resources	Protected via OS permissions. Root access generally required
8	eBPF modification	eBPF bytecode	I	Unprivilaged software adversary	Filesystem	eBPF bytecode on disk is modified by an attacker	Protected via OS permissions.
9	Invalid Packet Data	NIC resources, QAT availability	A	Network Adversary	Network data, Vhost-virtio data	Malicious data is sent which causes hardware lockup of the NIC or other HW e.g. QAT	HW implements protections to prevent invalid data causing lockup
10	Unauthorized memory access	Process memory	C	Unprivilaged software adversary	Filesystem	Attacker accesses the hugepage memory of a SPDK process through either the hugetlbfs filesystem or via unix socket for multiprocess	Hugepage files and unix sockets are protected by filesystem permissions. Generally can only be accessed by user running SPDK app.

THREATS



ID	Threat	Assets	Protect-ions Req'd	Adversary	Attack Point	Technique	Mitigation
11	QAT firmware modification	QAT firmware	I	Unprivilaged software adversary	Filesystem	Attacker modifies QAT firmware files on the filesystem	Protected by filesystem permissions. Firmware is validated by kernel before loading