

# Quick Introduction to eBPF: Info needed during hands-on session

## First eBPF Program

```
#include <linux/bpf.h>
#include <bpf/bpf_helpers.h>

SEC("xdp")
int xdp_main(struct xdp_md *ctx)
{
    bpf_printk("here");
    return XDP_PASS;
}

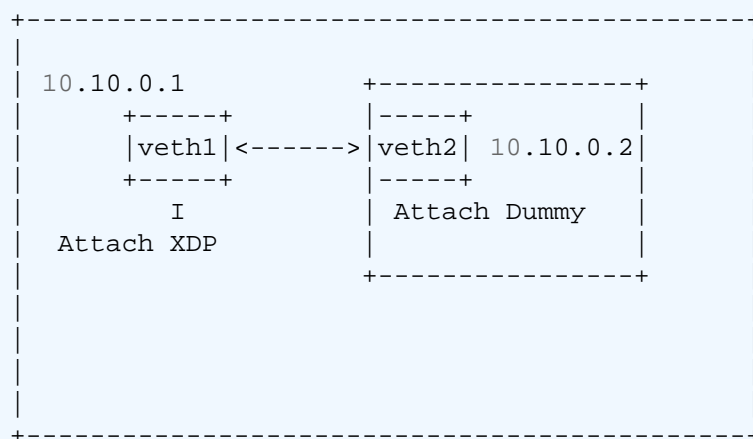
char LICENSE[] SEC("license") = "GPL";
```

## First Time Running eBPF Program

### Setup Environment:

```
sudo ip netns add n2
sudo ip link add veth1 type veth peer name veth2 netns n2
sudo ip link set veth1 up
sudo ip addr add 10.10.0.1/24 dev veth1

sudo ip netns exec n2 ip link set veth2 up
sudo ip netns exec n2 ip addr add 10.10.0.2/24 dev veth2
```



### Compile eBPF Program:

```
clang -S \
    -target bpf \
    -g -O2 -emit-llvm \
    -o NAME.bpf.ll NAME.bpf.c
```

```
llc -mcpu=probe -march=bpf -filetype=obj -o NAME.bpf.o NAME.bpf.ll  
bpftool gen skeleton NAME.bpf.o name SKEL_NAME > NAME.skel.h
```

#### Compile Loader Program:

```
clang -g -O2 -o ./loader ./loader.c -lbpf -lelf
```

#### Reading BPF Trace Logs:

```
sudo cat /sys/kernel/tracing/trace_pipe
```

## BPFTOOL

#### Listing attached eBPF Networking Programs:

```
sudo bpftool net
```

#### Listing Loaded eBPF Programs

```
sudo bpftool prog
```

## Generating Packets

#### Running NetCat Server (listen for packets):

```
nc -l -u 10.10.0.1 8080
```

#### Running NetCat Sending Packets:

```
printf "hello world\n" | nc -W 1 -N -u 10.10.0.1 8080
```

## Using IPROUTE2 To Load XDP Programs

```
#!/bin/bash  
ip link set dev veth2 xdp off  
sudo ip link set dev veth2 xdp obj first.bpf.o sec xdp  
  
on_signal() {  
    ip link set dev veth2 xdp off  
    exit 0  
}  
  
trap "on_signal" SIGINT SIGHUP  
echo Hit Ctrl-C  
while [[ true ]]; do  
    sleep 5  
done
```

## Install Dependencies On Ubuntu 22.04

```
# General stuff
sudo apt update
sudo apt install -y gcc-multilib build-essential libelf-dev linux-tools-`uname -r`
# LLVM/Clang v15
wget https://apt.llvm.org/llvm.sh
sudo bash ./llvm.sh 15
# Libbpf
# install a prebuilt version of follow the instructions at
# https://github.com/libbpf/libbpf
```