KubeCon | CloudNativeCon

Europe 2020

Virtual

CNI Deep Dive

Bruce Ma, Ant Financial
Bryan Boreham, Weaveworks

Welcome!

For more information:

- CNI Intro session https://sched.co/ZewR
- GitHub repos:
  - https://github.com/containernetworking/cni
  - https://github.com/containernetworking/plugins
- Slack - https://slack.cncf.io - topic #cni

Outline

- Introductions
- Recap: what is CNI?
- CNI Extensibility
- Recent CNI-related security issues
- Questions

Who are we?

Bruce Ma

Engineer, Ant Financial

Focus on container networking, “Fresh” CNI Maintainer

GitHub ID: mars1024

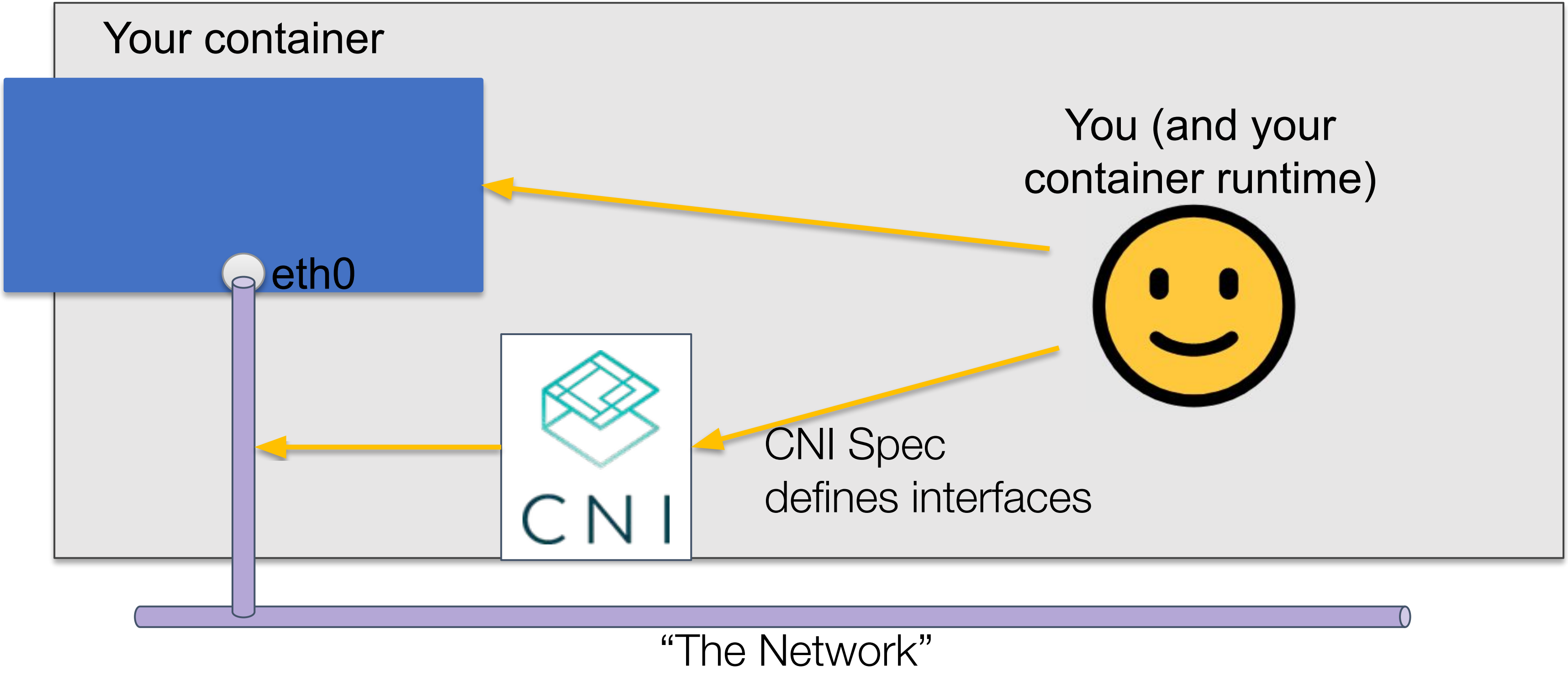
Bryan Boreham

Distinguished Engineer, Weaveworks

Maintainer on CNI, Weave Net and Cortex

GitHub: bboreham 🐦

What is CNI?


Your container
eth0
You (and your container runtime)
CNI
CNI Spec defines interfaces
“The Network”

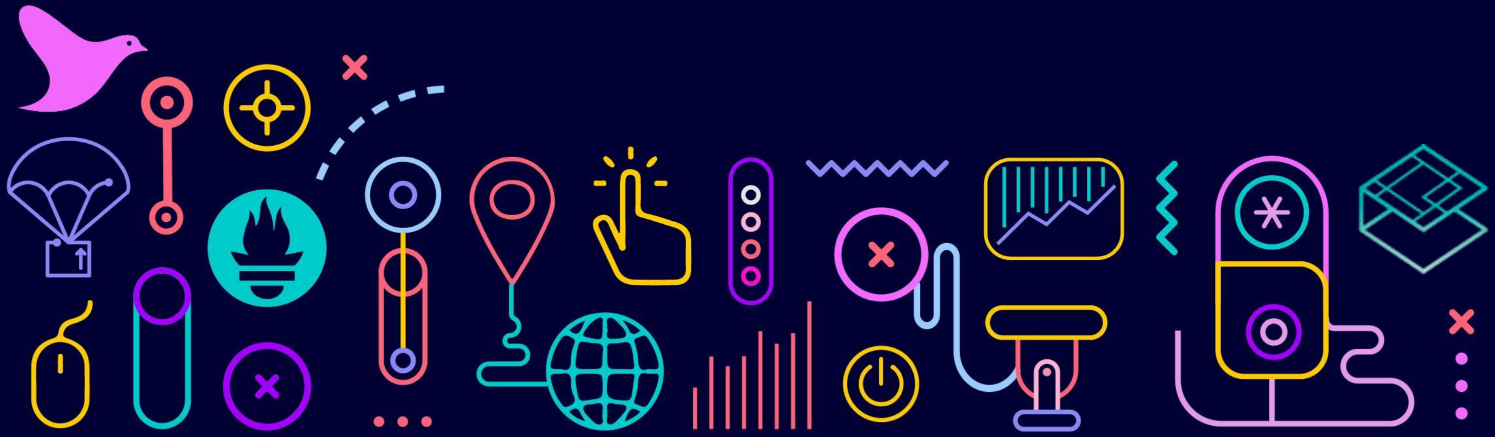
KubeCon | CloudNativeCon

Europe 2020

CNI Deep Dive

Looking into…

CNI Extensibility



CNI extensibility - overview

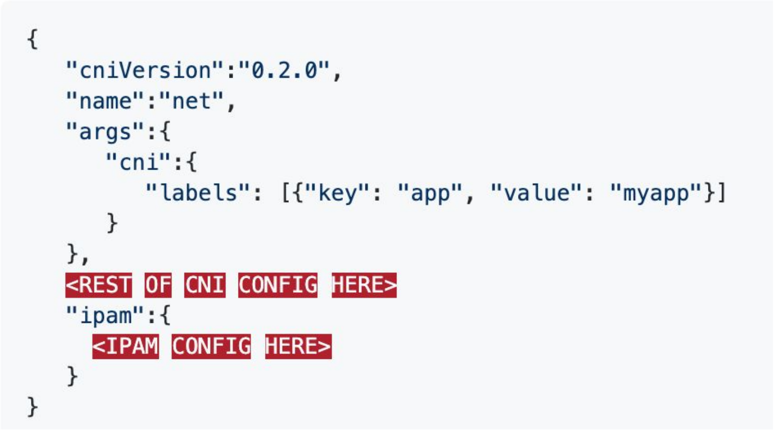
- Ref to CONVENTION.md
- Question
  - How to pass dynamic/custom information to plugins?
- Answer
  - Some ways to **override inputs** to plugins, the “extensibility” beyond the “spec”
- Use cases
  - IP/Subnet assignment for IPAM plugins
  - Bandwidth limits
  - User-defined host port mappings
  - Reset MAC/MTU for container interfaces
- Evolution
  - CNI_ARGS → args in config → Capabilities

CNI extensibility - CNI_ARGS

- History
  - Long time ago, born with the original CNI spec
- Format
  - Through process ENVs when calling plugin
  - Alphanumeric key-value pairs separated by semicolons, eg. "FOO=BAR;ABC=123"
- Advantages
  - debug friendly
  - Pass easily and flexibly
- Limitations
  - DEPRECATED!
  - Lowest priority
  - Not easy to hold structured data
- Question
  - If we don’t use envs to pass key-value pairs, what should we use instead?

CNI extensibility - args in config

- Answer
  - JSON network config is a better place
- History
  - Introduced in SPEC v0.2.0
  - Implemented in code release v0.4.0
- Sample
- Advantages
  - Easy to hold structured data
- Limitations
  - Container runtime have to implement the config-override function itself
  - Args will be pass to every plugin whatever it is needed or not

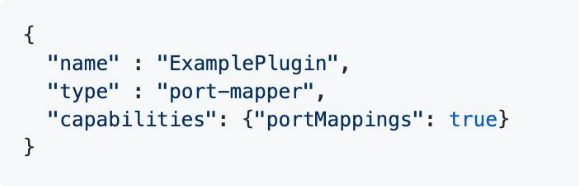


```
{
   "cniVersion":"0.2.0",
   "name":"net",
   "args":{
      "cni":{
         "labels": [{"key": "app", "value": "myapp"}]
      }
   },
   <REST OF CNI CONFIG HERE>
   "ipam":{
     <IPAM CONFIG HERE>
   }
}
```

- Question
  - Only a map to hold everything, full customization but control lost?
  - Pass all info to every plugin, information redundancy?

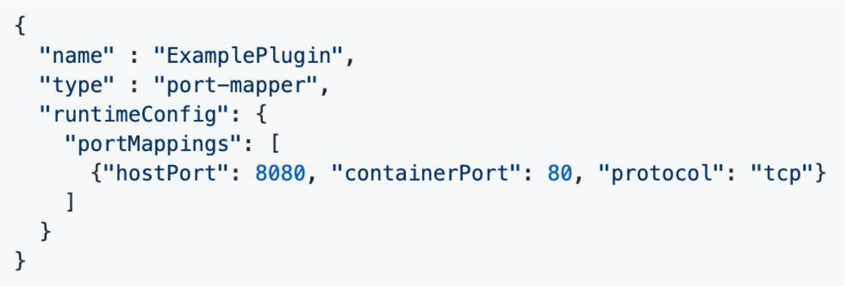
CNI extensibility - Capabilities

- Answer
  - CNI Capabilities(runtime configuration)
- Advantages
  - Plugin can define its additional fields in network config
    - Pre-validation
    - Remove redundant information
  - Libcni will override different field when calling different plugin
- How it works?



```
{
  "name" : "ExamplePlugin",
  "type" : "port-mapper",
  "capabilities": {"portMappings": true}
}
```

＋ User-defined CapabilityArgs Map ＝



```
{
  "name" : "ExamplePlugin",
  "type" : "port-mapper",
  "runtimeConfig": {
    "portMappings": [
      {"hostPort": 8080, "containerPort": 80, "protocol": "tcp"}
    ]
  }
}
```

CNI extensibility - Capabilities

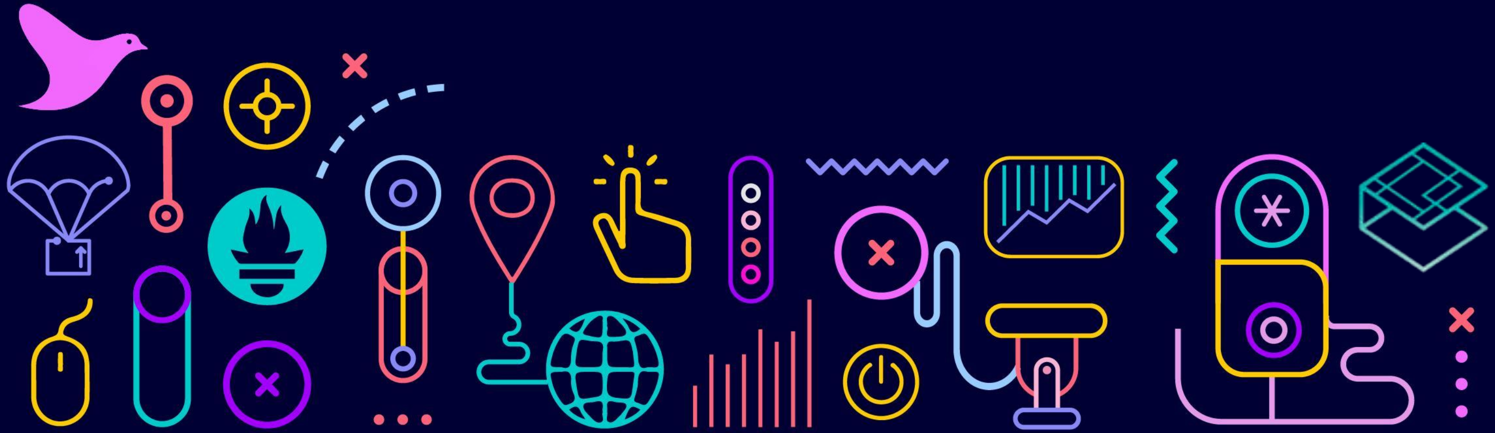
- Is **Capabilities** the final answer for dynamic configuration?
- Still have limitations
  - New capability **introduction cost**
    - Network config change
    - Code change in container runtime
  - Capability can **only** come from container runtime
    - Sometimes plugin need to pass dynamic configuration to one next plugin
- The future
  - Capability self-discovery (config and plugin may mismatch)
  - Capability pass over plugins (capture configurations from results?)
  - Provide more utility package about CNI extensibility

CNI extensibility - Comparison

	Age	Customization	Plugin info filter	Auto fill	Structured Data Supported
CNI_ARGS	oldest	Full	false	false	false
Args in config	old	Full	false	false	true
Capabilities	young	Limited	true	true	true

Now let’s look at…

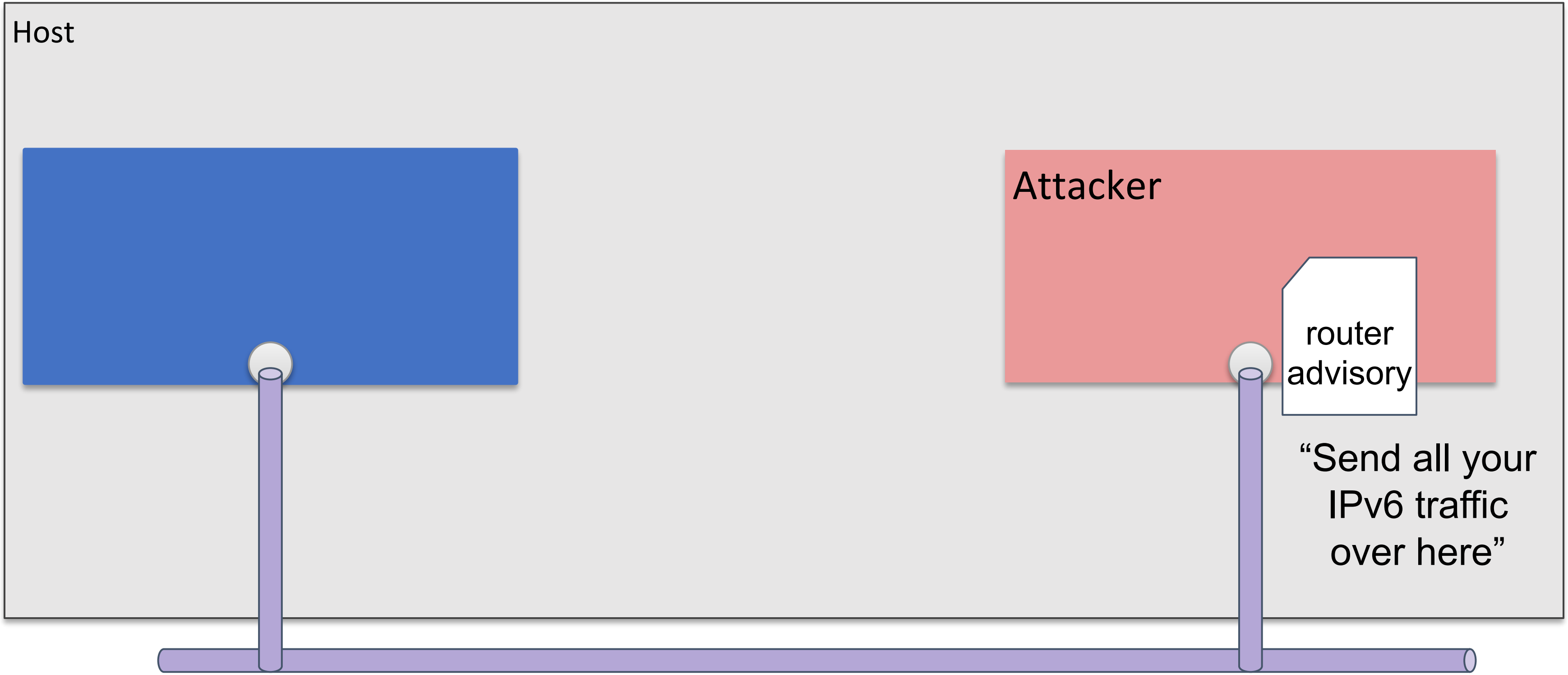
Recent CVEs



Recent CNI-related CVEs

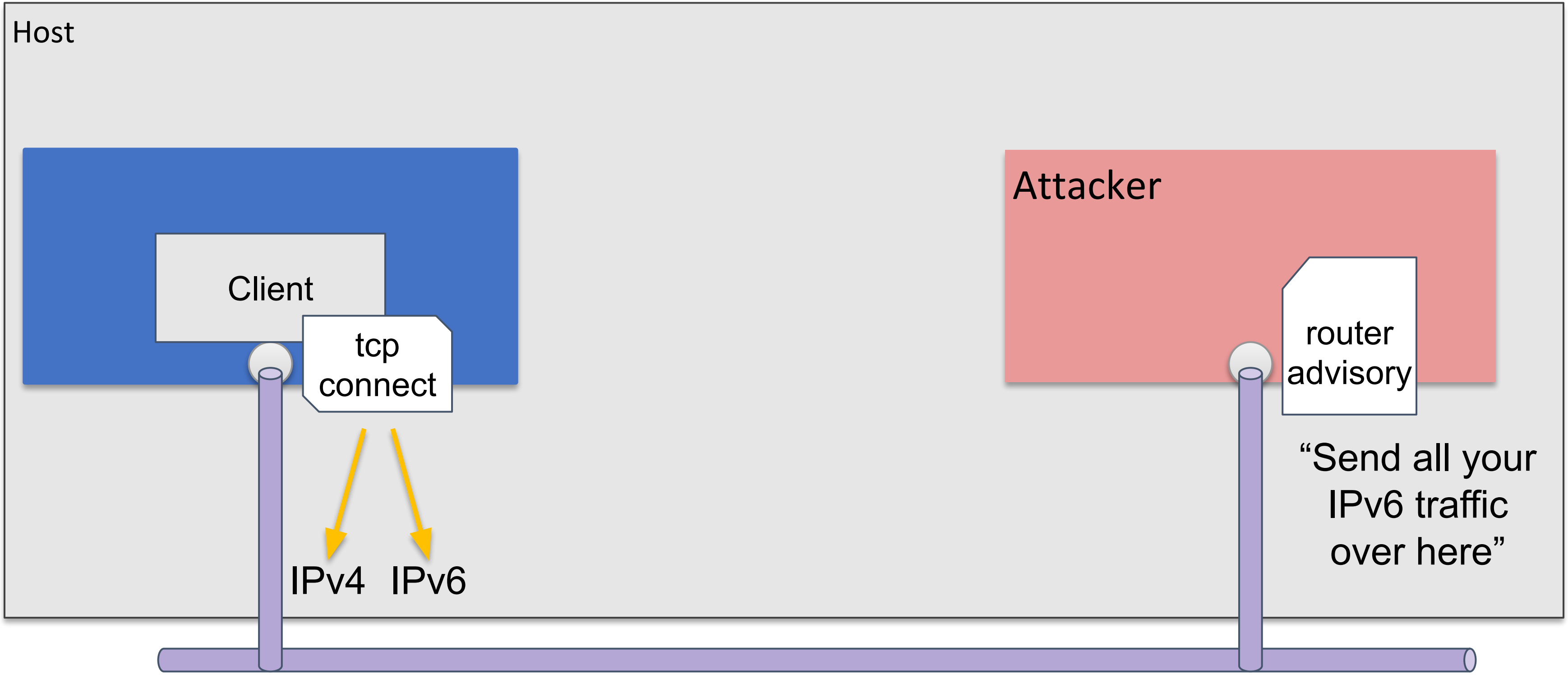
- CVE-2020-10749: Attacker sends IPv6 Router Advisory message and takes over some connections.
- CVE-2020-8558: Daemons listening on 127.0.0.1 are open to traffic from other nodes.

CVE-2020-10749



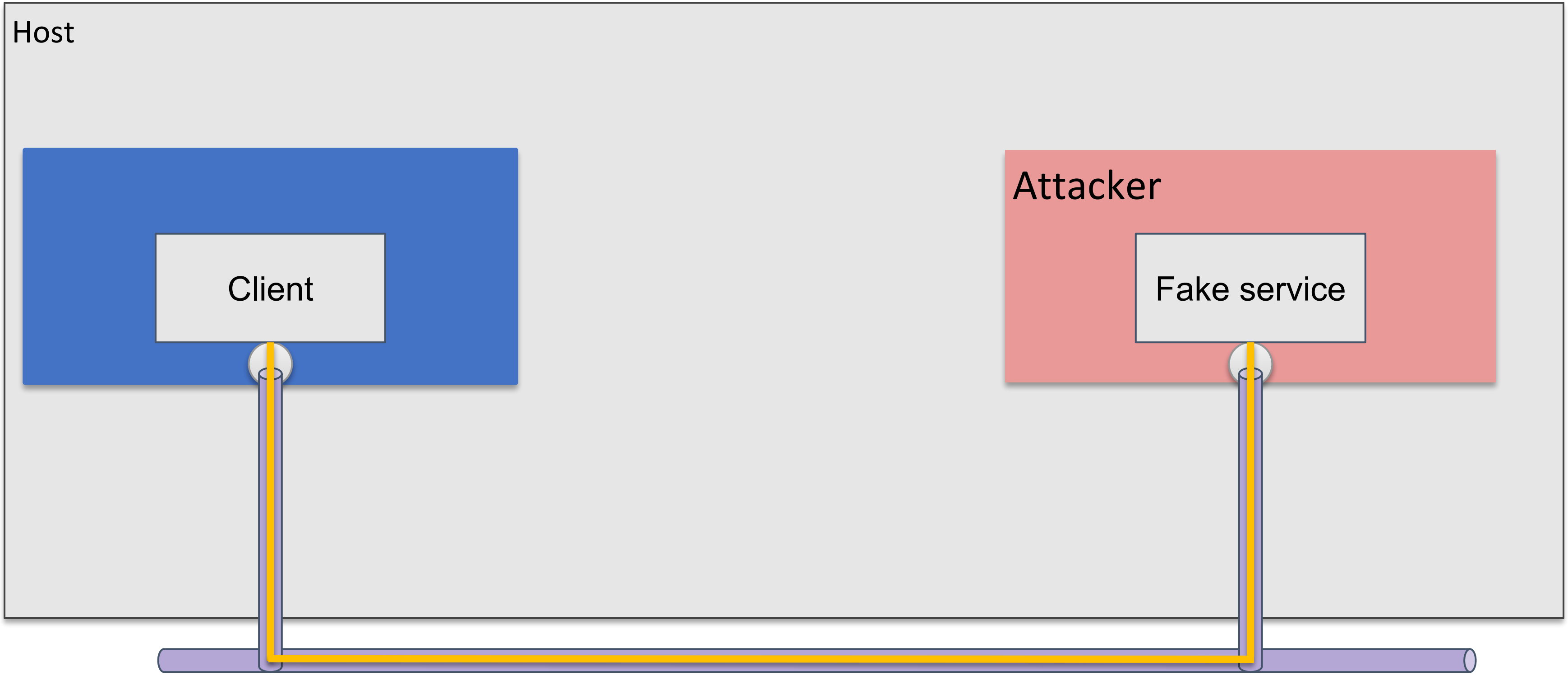
"Send all your IPv6 traffic over here"

CVE-2020-10749

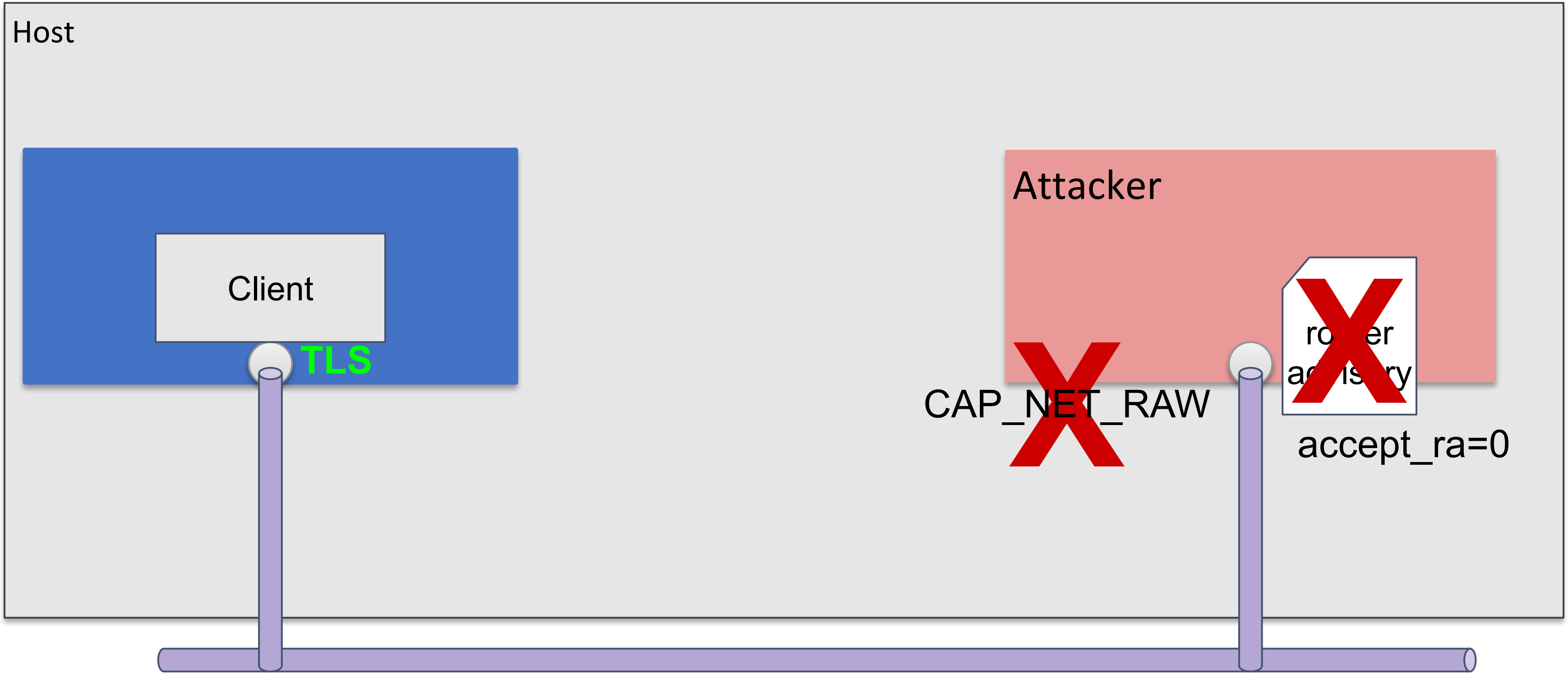


“Send all your IPv6 traffic over here”

CVE-2020-10749



CVE-2020-10749 Mitigations

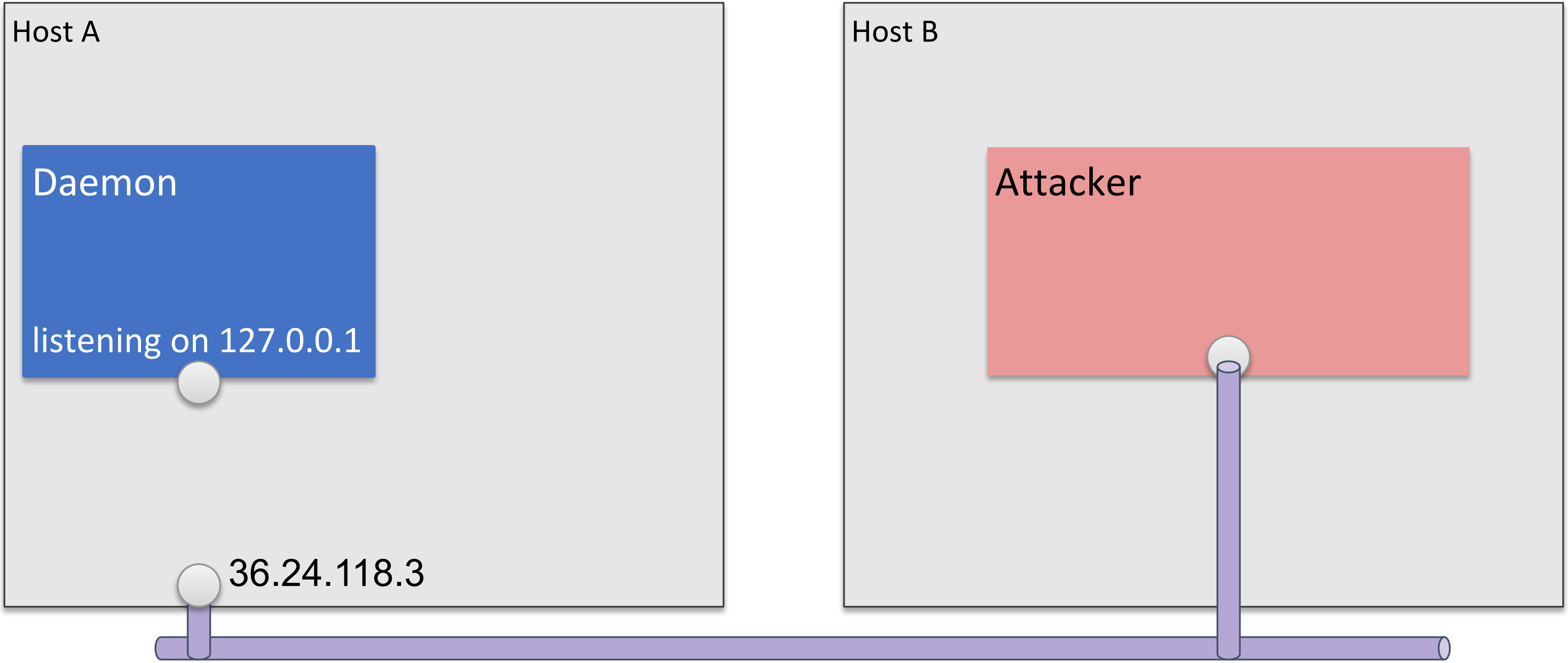


Recent CNI-related CVEs

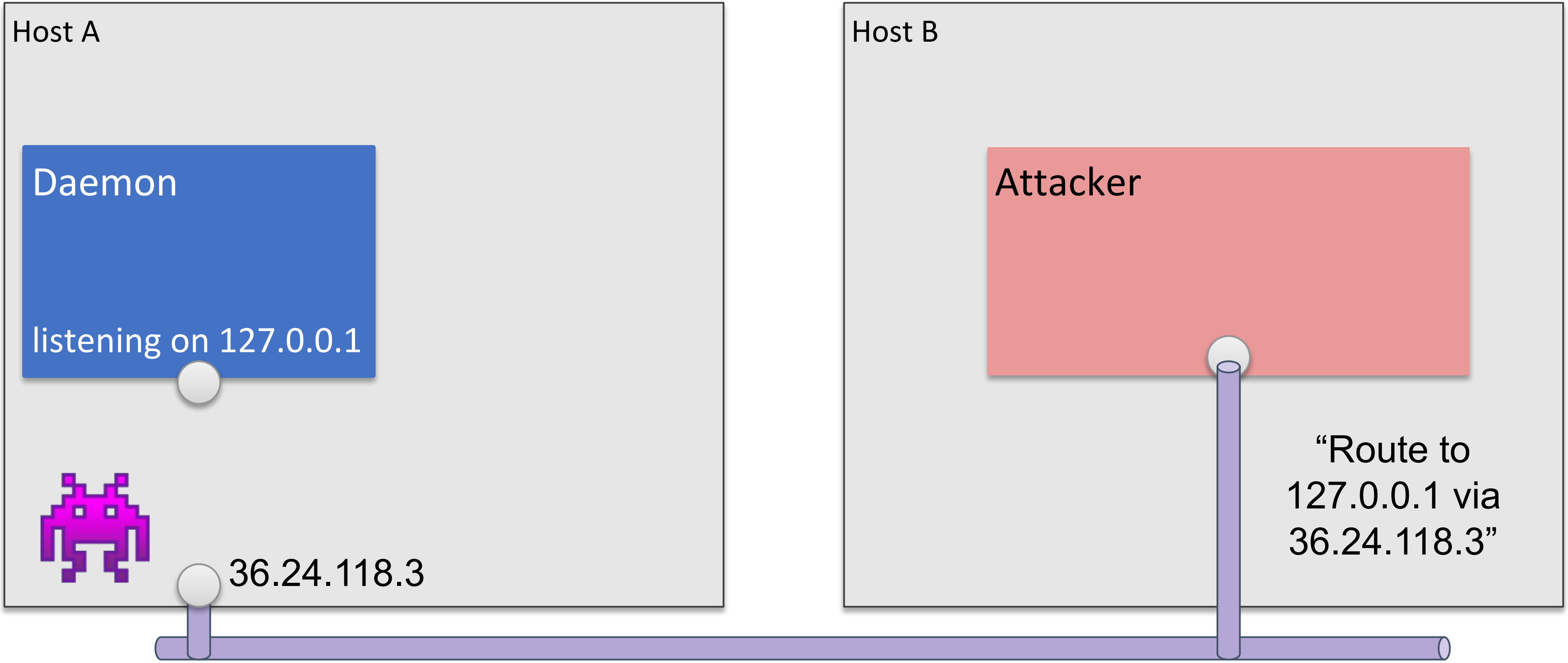
- CVE-2020-10749: Attacker sends IPv6 Router Advisory message and takes over some connections.

- CVE-2020-8558: Daemons listening on 127.0.0.1 are open to traffic from other nodes.

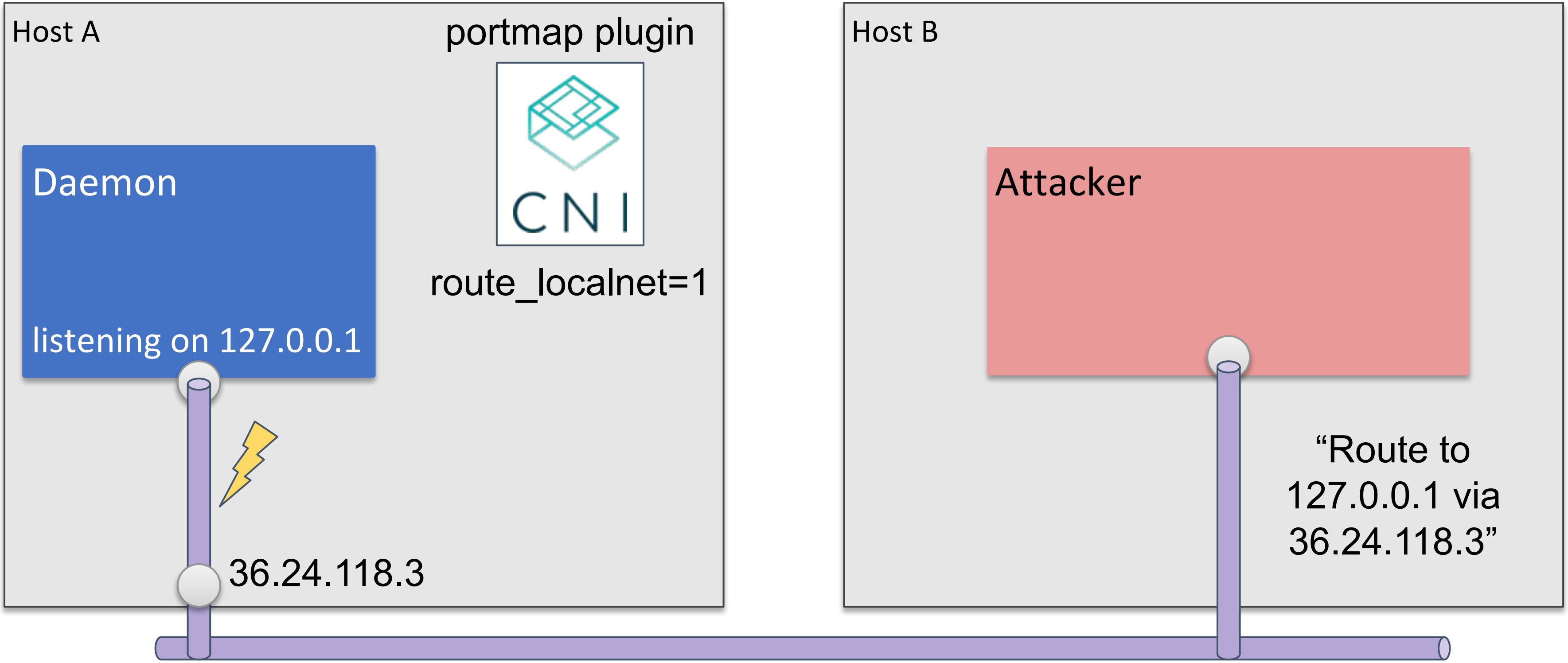
CVE-2020-8558



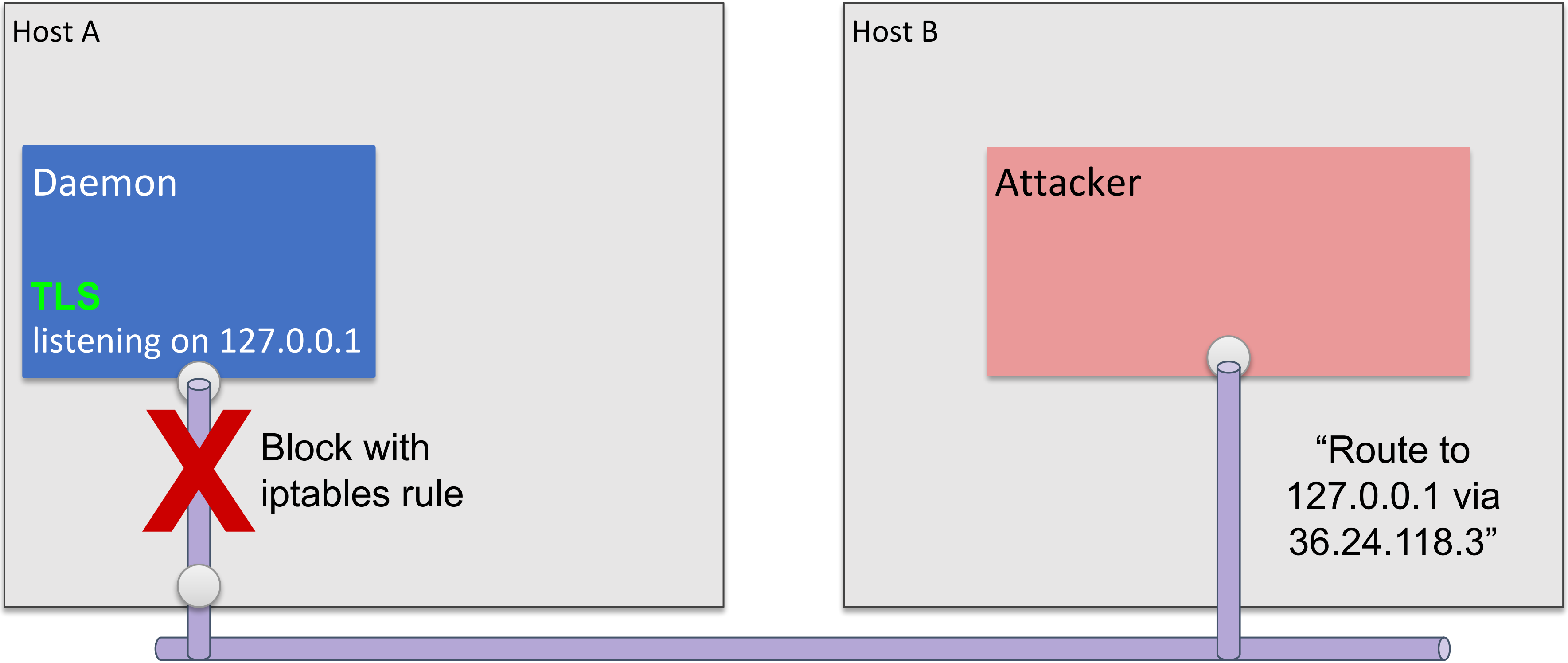
CVE-2020-8558



CVE-2020-8558



CVE-2020-8558 mitigations



Host A

Daemon

TLS

listening on 127.0.0.1

Block with iptables rule

Host B

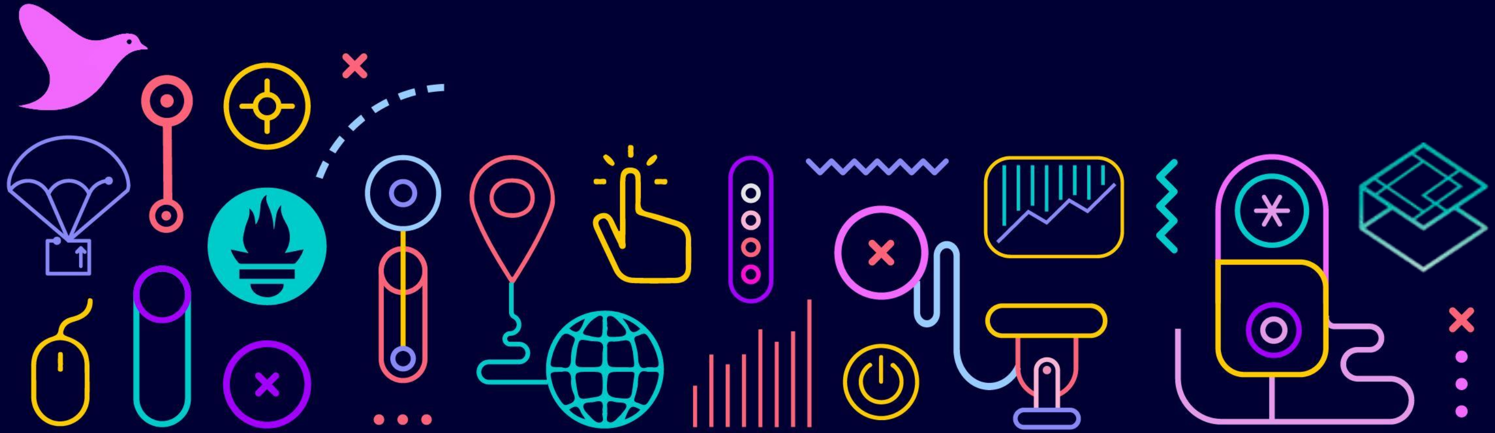
Attacker

"Route to 127.0.0.1 via 36.24.118.3"

Information and links

- CNI Intro session https://sched.co/ZewR
- GitHub repos:
    - https://github.com/containernetworking/cni
    - https://github.com/containernetworking/plugins
- Slack - https://slack.cncf.io - topic #cni
    - #sig-network for k8s-specific topics.

Questions?



KubeCon | CloudNativeCon

Europe 2020

Virtual

KEEP CLOUD NATIVE

CONNECTED