Linux Firewall Automation with nftables & Rust

`whoami`

Jasper Wiegratz

- Solution Architect @ SVA
  - OpenShift 📦
  - Ansible 🛠️
- Academia:
  - Linux Networking
  - Container Security
- Rustacean 🦀

Agenda

1. Context
2. Intro to Linux Packet Filtering
3. Automation Options & Challenges
4. Rust-Based SDK for nftables
5. Lessons, Insights, What's next
6. Q&A & Discussion

Intro & Context

MUD and the NAMIB Project

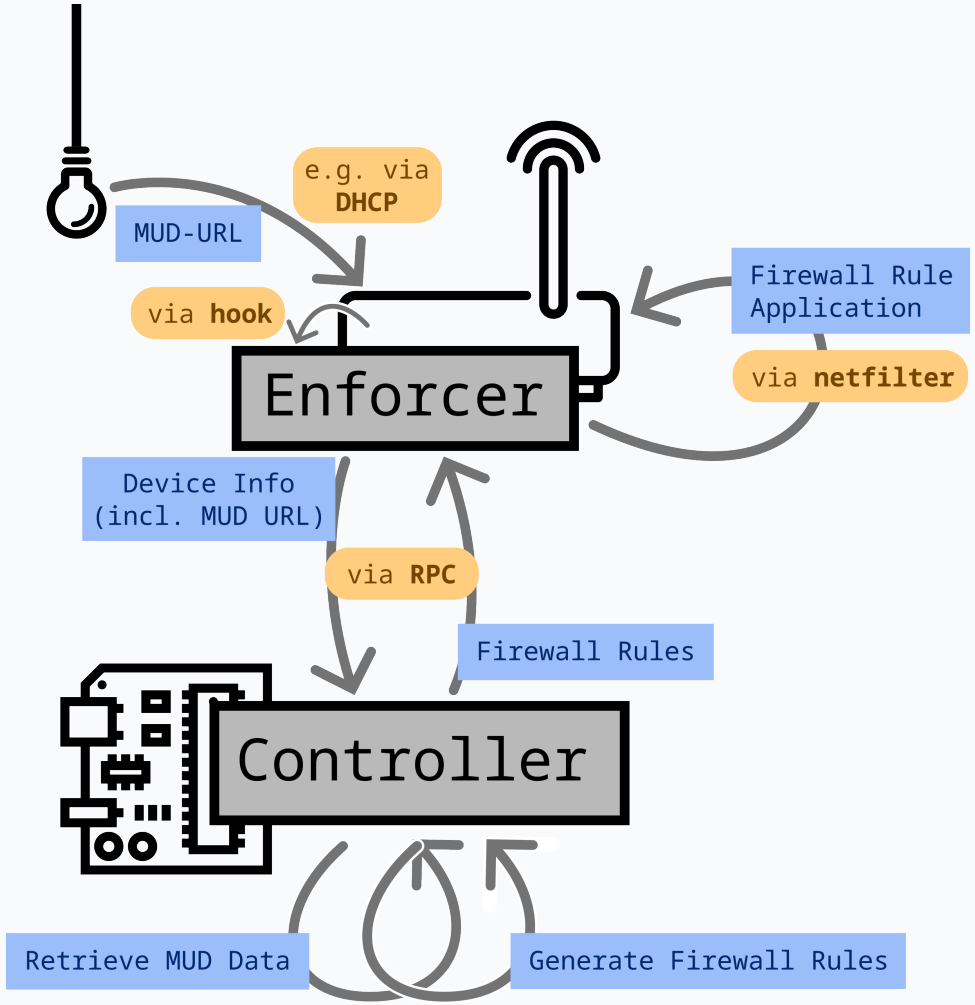
NAMIB
Network Access Makes IoT Better

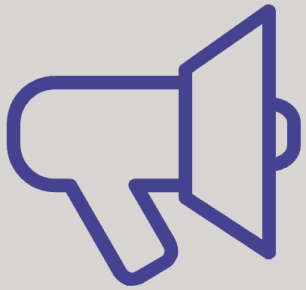
Reality of Home Networks:

- Single (W)LAN Network
- Unrestricted Access to Internet
- Unrestricted Local Access

The Cure?

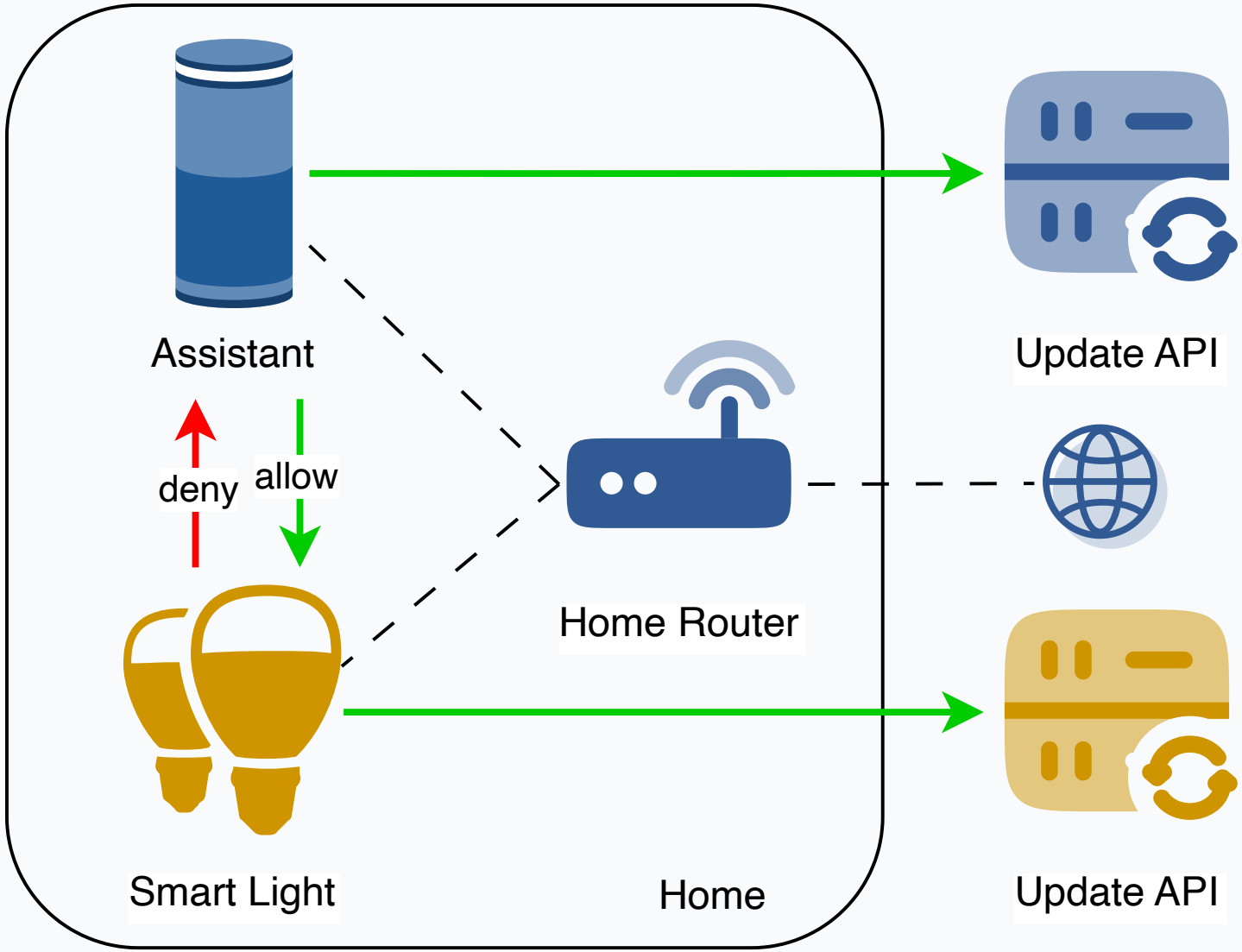
- Microsegmentation
- Tailored Access Policies





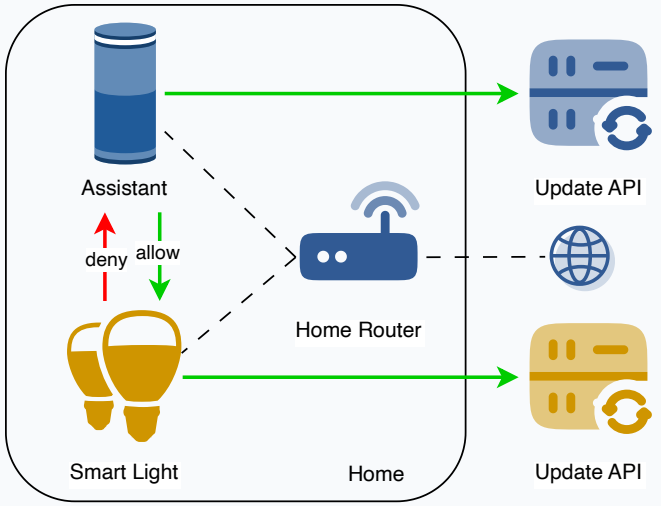
Intro to Linux Packet Filtering

Example: Desired IoT data flow



Network policies with iptables

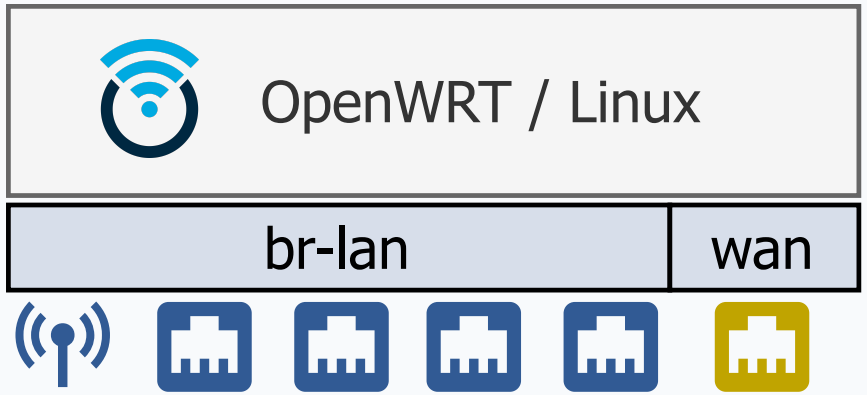
- **iptables** for routed IPv4 traffic
- **ip6tables** for routed IPv6 traffic
- Home networks:
  - Only local ↔️ internet




```
iptables -A OUTPUT -s 192.168.1.150 -d 192.168.1.165 ⏎
  -p tcp --dport 9999 -j ACCEPT # Assistant to Bulb ⚠️

iptables -A OUTPUT -s 192.168.1.150 -d 134.102.121.162 ⏎
  -p tcp --dport 443 -j ACCEPT # Assistant to Update API

iptables -P OUTPUT DROP # drop all other output
```

Local network policies with ebtables

- **ebtables** for bridge traffic
- implementing a filtering switch/bridge
- can inspect *higher* protocols
  - IPv4, IPv6, ARP
- Home networks:
  - within/across LAN/WLAN




```
# Allow ethernet from Assistant to Lamp on tcp/9999
ebtables -A FORWARD -s 36:8d:95:12:2d:30 -d 4b:fc:3a:d3:9f:8d ⏎
  -p IPv4 --ip-proto tcp --ip-dport 9999 -j ACCEPT

ebtables -P FORWARD DROP # drop all other output
```

nftables: one framework to rule them all

IP traffic

```
table inet filter {
  chain output {
    # drop by default
    type filter hook output priority 0; policy drop;

    # allow Assistant to Update API
    ip saddr 192.168.1.150 ip daddr 134.102.121.162 ⏎
      tcp dport 443 accept
  }
}
```

Bridge traffic

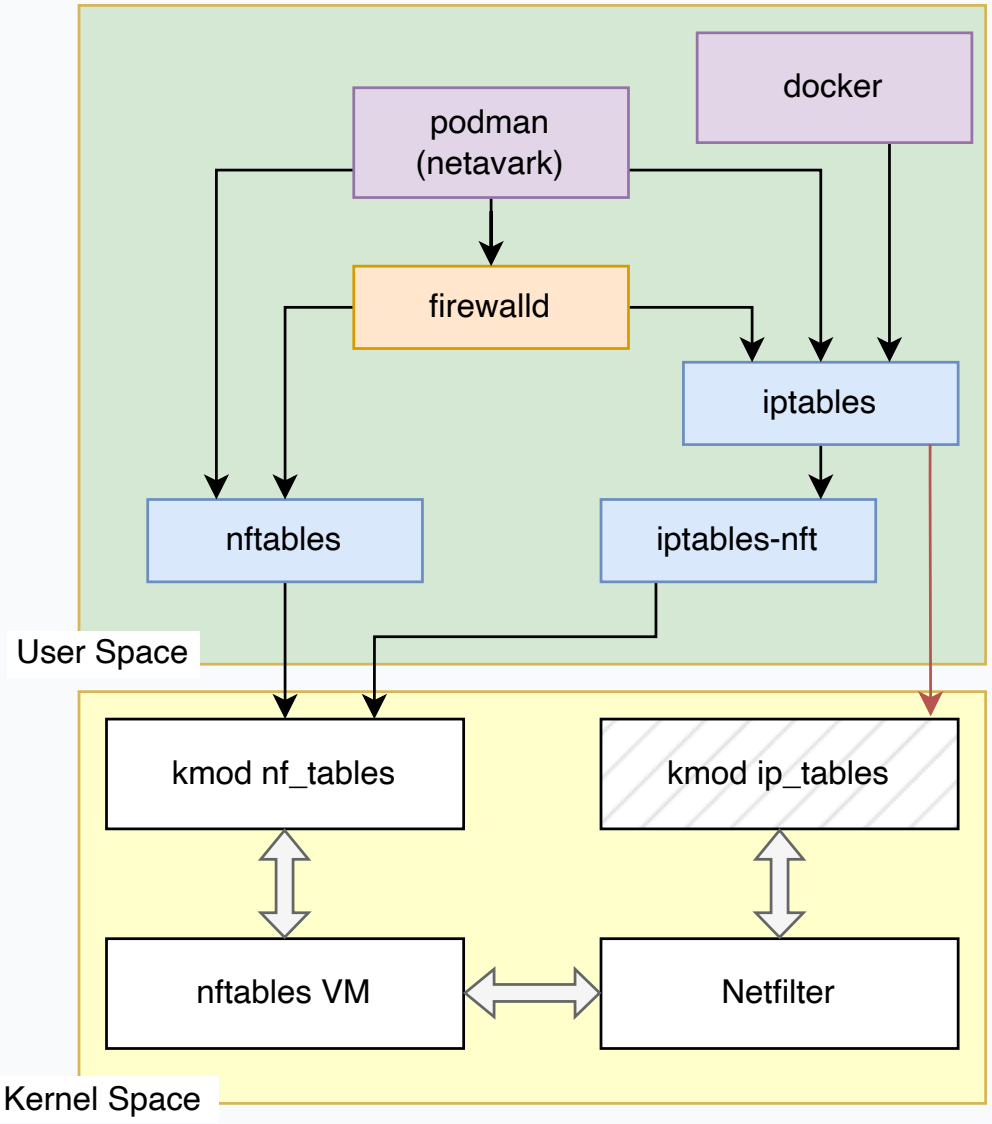
```
table bridge filter {
  chain forward {
    type filter hook forward priority 0; policy drop;

    # allow Assistant to Bulb
    ether saddr 36:8d:95:12:2d:30 ⏎
      ether daddr 4b:fc:3a:d3:9f:8d ⏎
      ip protocol tcp tcp dport 9999 accept
  }
}
```

xtables vs. nftables

	xtables	nftables
Available	2001 (ipchains)	2014
Families	iptables, ip6tables, arptables, ebtables	ip, ip6, inet, arp, bridge, netdev
Interfaces	Imperative	**Declarative**, Imperative
Schema	Per family	Unified
Persistence	`iptables-save/restore`	`/etc/nftables.conf`

Automation Options & Challenges

Challenging ecosystem



- nftables replaces iptables
  - Fedora 32
  - RHEL 8
  - Debian 10
  - Ubuntu 21.10
- **Goal**: automate with highest abstraction framework
  - firewalld, ufw
  - nftables
  - iptables

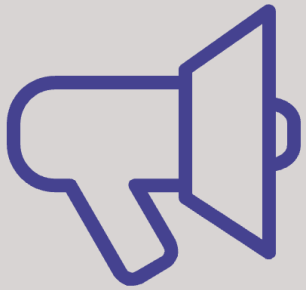
Nftables Automation Options

Static: .nft includes

- RHEL: `/etc/sysconfig/nftables.conf`
- Debian: `/etc/nftables.conf`

Dynamic

- CLI: `nft add rule inet filter input tcp dport 25 drop`
- JSON API: `cat nft.json | nft -j -f-; nft -j list ruleset`
- Libraries for native API or **JSON API**

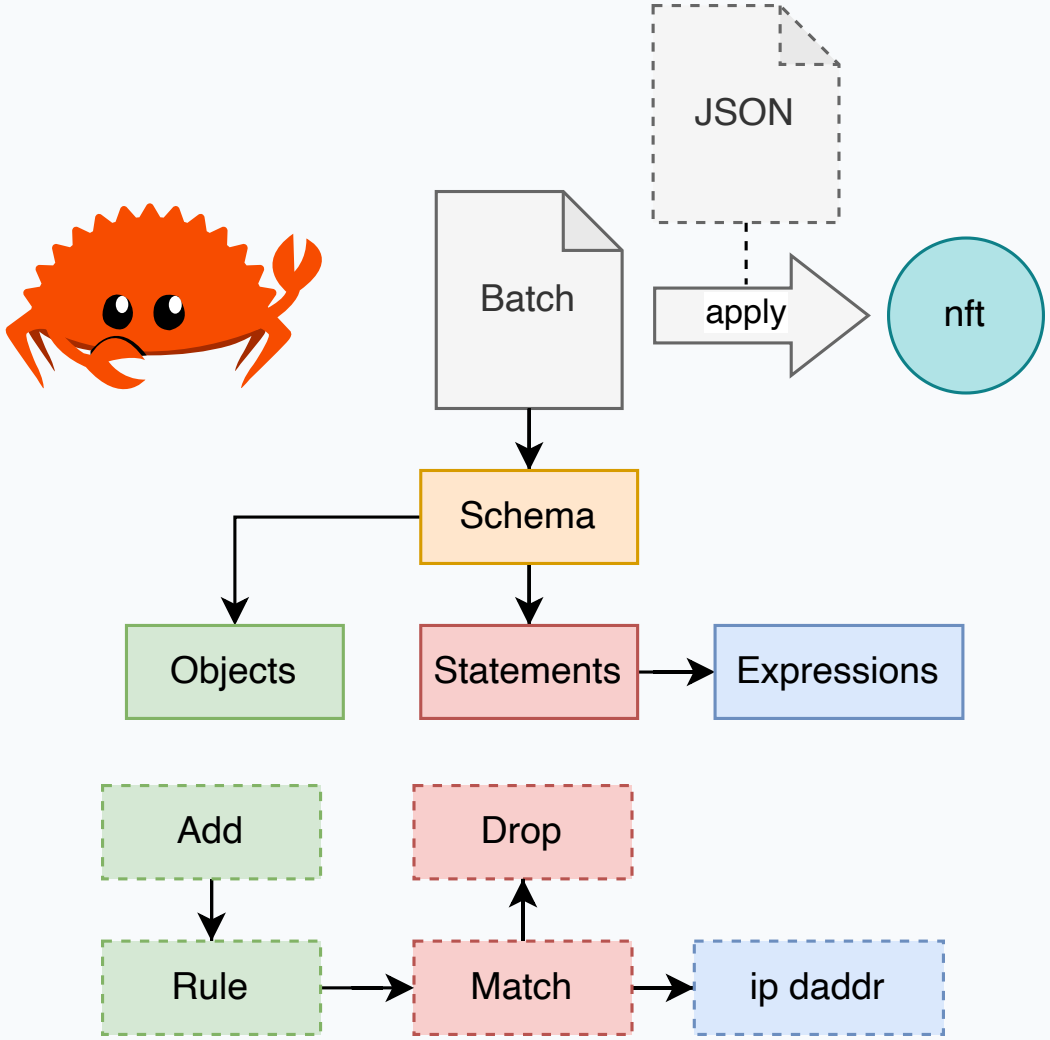


Rust-Based SDK for nftables

nftables-rs: Rust SDK for nftables

- Requires *nftables* with JSON support
- 100% safe Rust, no need to rewrite
- 100+ nftables types
  - 59 structs, 48 enums
- (100-x)% complete ✅, 85% documented ⏫

Example: IP-based Drop Rule



nftables-rs: Real World Applications

NAMIB

Network Access Makes IoT Better

podman

- NAMIB Enforcer (2021): Microsegmentation with MUD for OpenWRT
- Netavark (2023): Podman Network Backend

- wellenbrecher 🌊 (2023): Pixelflut Server
- ic (2024): The Internet Computer Protocol
- letmein (2024): Authenticated Port Knocker
- flow (2024): BGP flowspec executor

src: media.ccc.de/v/pixelflut-eh15, github.com/containers/podman

Lessons & Insights

Lessons learned from nftables-rs

- nftables: integration with JSON API vs native API
  - more **portable**
  - and **easier to maintain**

- nftables lacks formal JSON schema
  - required for correctness verification
  - can be derived from nftables-rs
  - *man pages* may be incomplete

Lessons Learned: Rootless testing via netns

```
Connecting to vm podman-machine-default. To close connection, use `~.` or `exit`
```

What's next?

nftables-rs: What's next?

- Higher-level abstraction (without being firewalld)
  - Macros? DSL?
- Derive JSON Schema for nftables ⭐
- Comprehensive Test Suite ⭐
  - nftables Test Fuzzing

Links & Recommended Reading

- nftables-rs: github.com/nftables-rs
- nftables Wiki: wiki.nftables.org
  - nftables in 10 minutes
  - Netfilter hooks
  - Main differences with iptables
- `man 5 libnftables-json`
- Projects: netavark, wellenbrecher, letmein
- LWN: BPF as a safer kernel programming environment (2022)

Your own idea?

Build something amazing and see where you'll end.

Thanks!