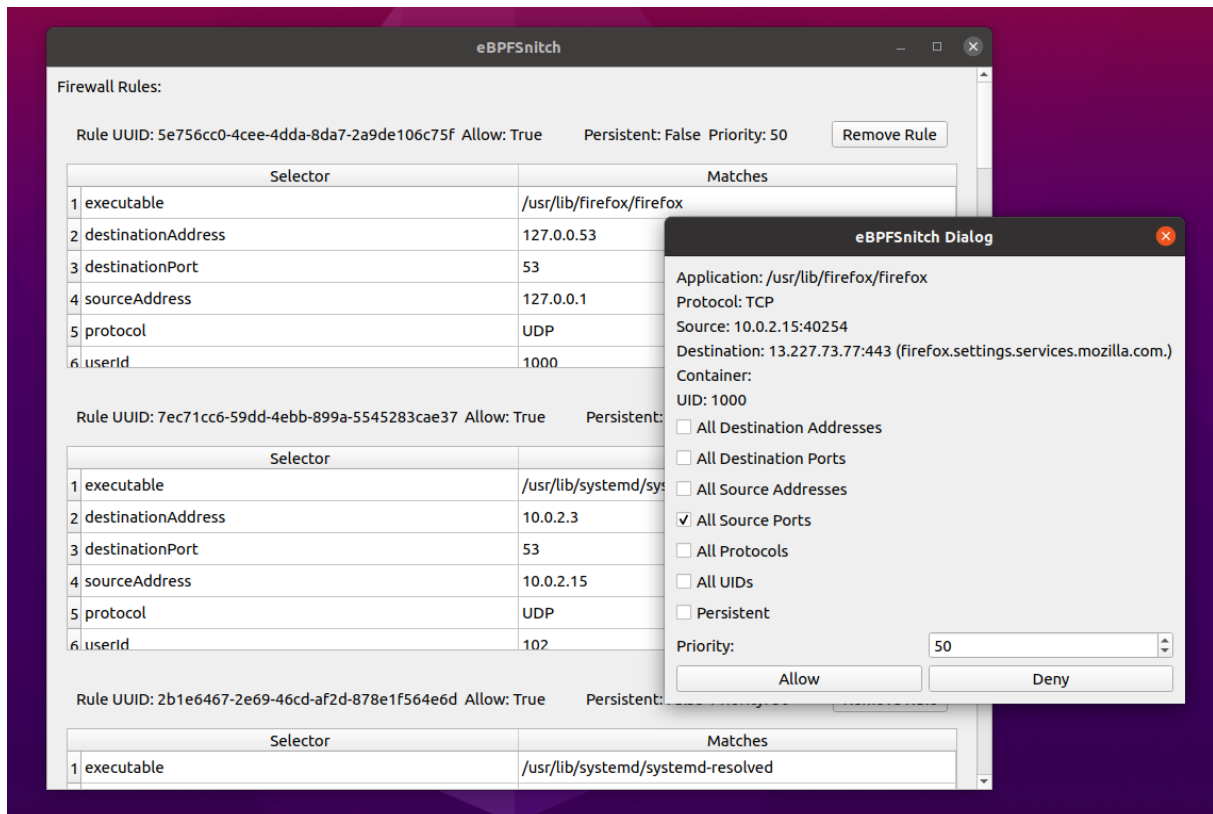

eBPFSnitch

eBPFSnitch is a Linux Application Level Firewall based on eBPF and NFQUEUE. It is inspired by OpenSnitch and Douane but utilizing modern kernel abstractions - without a kernel module.

The eBPFSnitch daemon is implemented in C++ 20. The control interface is implemented in Python 3 utilizing Qt5.



Disclaimer

This is an experimental project. The security of this application has not been audited by a 3rd party, or even myself. There are likely mechanisms by which it could be bypassed. Currently the daemon control socket is unauthenticated and an attacker could impersonate the user interface to self authorize.

Features

eBPFSnitch supports filtering all outgoing IPv4 / IPv6 based protocols (TCP / UDP / ICMP / etc). Filtering incoming connections should be supported in the near future.

A core goal of this project is to integrate well with containerized applications. If an application is running in a container that container can be controlled independently of the base system or other containers.

Additionally targeting can occur against specific system users. Blanket permissions for every instance of Firefox for every user are not required.

Daemon Configuration

eBPFSnitch is configured via command line arguments. The available arguments can be listed with `--help`:

```
1 eBPFSnitch Allowed options:
2 -h [ --help ]           produce help message
3 -v [ --version ]        print version
4 --remove-rules          remove iptables rules
5 --group arg             group name for control socket
6 --rules-path arg        file to load / store firewall rules
```

Control socket authorization

The control interface and daemon communicate utilizing a Unix socket. By default the socket can be accessed by any system user. It is recommended to associate a specific group with the socket to limit access. For example `--group='wheel'`.

Firewall rule persistence

Firewall rules that are marked as persistent are stored on the filesystem in a JSON encoding. By default, the current working directory is used to store the file `rules.json`. To specify a custom path use the `--rules-path` option.

System requirements

eBPFSnitch currently requires a recent kernel. The minimum supported version is Linux 5.8. This required version may be lowered in the future.

How firewall rules operate

Each rule is comprised of a set of clauses and a verdict. Each clause matches a property of a packet to value. If every clause in a rule matches, then the packet matches the rule and the verdict for that rule is used (allow / deny).

Rules are sorted by a configured priority. Each rule is tried until a match is found and a verdict can be determined. If no rule matches a packet, the daemon will send a query to the interface which then displays a dialog asking to create a new rule to match that packet.

By default rules are not persisted to disk. When the daemon restarts rules will be lost. If through the dialog you check the `persistent` box, the new rule will be saved to disk and be active when the daemon is restarted.

Installation with a package manager

eBPFSnitch is currently only available on the Arch user repository. Other distributions will require building from source manually.

```
1 # installation using the yay aur helper
2 yay -S ebpfSnitch
3 # start daemon
4 sudo systemctl start ebpfSnitchd
5 # start the ui
6 ebpfSnitch
```

Compilation instructions

If a package is not available for your distribution you can build eBPFSnitch from scratch as follows:

Dependencies

C++: pthread, libbpf, netfilter_queue, spdlog, fmt, nfnetlink, boost, libmnl

Python: PyQt5

Installing dependencies on Arch

```
1 sudo pacman -S clang cmake bpf libbpf libnetfilter_queue spdlog boost
  libmnl \
2     nlohmann-json conntrack-tools python3 python-pyqt5 vim
```

Installing dependencies on Ubuntu 21.04 (minimum version)

```
1 sudo apt-get install cmake clang libboost-all-dev libspdlog-dev \  
2     libnfnetlink-dev libmnl-dev linux-tools-common nlohmann-json3-dev \  
3     libbpf-dev linux-tools-generic conntrack python3 python3-pyqt5 \  
4     xxd libnetfilter-queue-dev
```

Installing dependencies on OpenSuse TumbleWeed

The program can be compiled on OpenSuse TumbleWeed, it took me an hour of fiddling. I'm not sure about the exact packages, but this works:

```
1 sudo zypper install make gcc *boost* cmake clang bpftool *bpf* nlohmann  
  * spd* *netfilter*
```

Some packages are not in the default repo but required libmnl libnfnetlink libnetfilter_queue

Setting up the daemon

From the eBPFSnitch repository directory:

```
1 mkdir build  
2 cd build  
3 cmake ..  
4 make  
5 sudo ./ebpfsnitchd
```

Starting the GUI

From the eBPFSnitch repository directory:

```
1 python3 ui/ebpfsnitch/entry.py
```