Network Firewall

Networks Lab Project

Group No. 8

Darshan Kumar, 21112040

Sanidhya Singh, 21114091

Sanidhya Bhatia, 21114090

Lakshya Joshi, 21114054

Project Description

The project is a software-based Network Firewall that can selectively block network packets received and sent on the device. It is made in Linux, for Linux.

The packets are intercepted using Netfilter hook-functions, where we drop or allow the packet based on its properties between the Network and Link Layer, before or after routing depending on whether it is an inbound or outbound packet. We have configured and coded the Netfilter hooks to filter the packets according to all the rules the user has added through the CLI, which we have created with all the needed options.

Project Implementation

The Netfilter hooks are injected into the kernel as kernel modules, not needing to recompile the kernel every time we insert our modules. Both the CLI and the networking functions are written in C/C++, which are compiled into kernel-specific output `*.ko*` binaries.

The modules are injected into the kernel with `*insmod*` command and removed after use with the `*rmmod*` command. The creation, compilation and insertion into the kernel is automated through a parser script which builds the needed kernel module based on the rule added. The rule is now active and the firewall within the kernel will start blocking any packet which satisfy any rule.

Since this involved extending the kernel (which led us to several crashes initially), therefore we used Vagrant to create virtual development environments, and used VirtualBox as L2-HyperVisor.

Key Components of the Project

1. *Command Line Interface (CLI)*: Our firewall has a custom command line interface where users can `*add*` and `*remove*` rules for packet filtering.
   Flags for the add command are *-p* (protocol), *-s* (source IP), *-d* (destination IP), and *-port* (port Number).
   Also, `*list*` and `*clear*` commands are used to display and remove all the rules respectively.

2. *Netfilter Hooks*: These hooks operate between the Network and the Link Layer. When packets are hooked, the conditions specified by the rules are checked, and the packet is dropped or accepted accordingly.

3. *Kernel Modules*: Hook-functions can directly be injected without recompiling the entire kernel using Linux Kernel Modules. This makes it much easier to add functionality to our kernel.

Contributions

The project needed a good amount of initial research, which all 4 of us participated in as a combined effort, laying out the objective and its execution plan.

The systematic ideation and architecture of the project was mainly done by Darshan. The command-line interface was worked on by Sanidhya Singh, deciding the commands and parameters' format, and executing them in code. The parser to generate modules based on user input was created by Lakshya and Sanidhya Bhatia.

Since Netfilter and Kernel Modules were the main components of the project, all 4 of us contributed almost equally and participated in the extensive debugging involved with Kernel Modules. Integration of CLI with Kernel Modules and Final testing was done by Darshan and Sanidhya Singh and extracting information about the packets like IP addresses, protocols, and port numbers from corresponding structs in code, when packets are hooked, was done by Sanidhya Bhatia and Lakshya.

Other significant parts of the project and most of the debugging and research was done together with every member participating.

Thank You for Reading!

References: https://www.netfilter.org/, https://github.com/torvalds/linux/tree/master/include, https://www.vagrantup.com/, https://sysprog21.github.io/lkmpg/