DD2497 Project Specification - Group 5

Tommie Andersson, André Brogärd, Henrik Kultala,
Hîvron Stenhav, Albin Winkelmann

November 12, 2021

1 Introduction

Microkernels often communicate, in some way, with external services for various reasons. By following the principle of least privilege (where each user, process etc has the least amount of privileges possible) you minimize the attack surface, and make it harder for potential malicious actors. Specifically for network traffic this can be accomplished through the usage of a so-called firewall.

The firewall is employed to ensure that only pre-defined traffic is allowed. This is a defensive mechanism to prevent remote exploits that could otherwise be possible on exposed processes, and prevent malicious processes from communicating with a remote host. Specifically it mitigates the effect of and prevents remote exploitation, where a remote adversary may try to gain access to the OS over the network.

A previous year in this course a Minix implementation of a firewall on the OS level was created. The firewall can for example block access on specified ports, and block specified source and destination IP-addresses. This firewall was later extended with increased security by making the firewall settings more expressive to support different rules for different processes. Thus, one can further restrict the network traffic to and from particular processes that don't need as much access as others.

We recognize the need to make this firewall even more expressive, and intend to extended it further by implementing firewall settings on a per user and/or group level. This would further improve the security of the micro-kernel as some users could have restricted internet traffic. Perhaps some users are only allowed to communicate to certain hosts, with a certain protocol, or they are not allowed to communicate at all over the network.

We aim to develop a service, which manages the per user and/or group permissions for network traffic, that also respects the firewall settings of the OS and process. A consequences of this is the threat of privilege escalation for traffic permissions that arises with different conflicting firewall settings for OS, processes and users. If for example a user is disallowed from having outgoing traffic on port 80, but this traffic is explicitly allowed in the OS firewall, perhaps this user could circumvent the more restrictive firewall rule. In the design of our service, we aim to address this potential privilege escalation in a "deny first" approach (not official terminology).

Our implemented service will either extend existing service(s) or be somewhat independent. Further investigation on how this service should be implemented is needed before we can say anything definitive, especially as it should work in conjunction with the existing solution and design choices.

2 Mandatory Requirements

Our first priority is to implement unique firewalls for all users and make sure that they cannot circumvent them from privilege escalation. With a system already in place to edit firewall rules from the project we are extending, this functionality will be preserved for privileged users to allow modifying their own firewalls. Meanwhile unprivileged users will not be allowed to modify any firewall, nor disable them.

Consequently, these are our mandatory requirements:

- one firewall per user
- unprivileged users can't edit firewall rules nor disable firewalls

  - a default firewall will be provide for these users

- privileged users can modify their own firewalls
- prevent privilege escalation happening due to more restrictive firewall rules being overshadowed by less restrictive rules from another firewall

3 Optional Requirements

After having satisfied our mandatory requirements, there are several extensions that would be worthwhile to our firewall. One natural extension would be to allow setting firewalls on groups in addition to users; making it possible to easily put certain restrictions on multiple users at once in a finer granularity than "privileged" and "unprivileged". Another desirable trait of our firewall would be to allow privileged users to modify other firewalls, making it more convenient to moderate other accounts. Further it would be nice to allow unprivileged users to modify their own firewalls by making them more restrictive, to increase their own security. Then there is also the case where a user A is allowed to execute a process as user B. This could potentially lead to privilege escalation (by user B having more access than user A), and should by handled.

Thus, our optional requirements are:

- in addition to a firewall per user; one firewall per group
  - somehow merge/prioritize firewalls from the user and all different groups the user is part of
- privileged users can modify firewalls of other users (is it binary in either privileged or unprivileged or are there multiple tiers of privilege?)
- unprivileged users can add more restrictive rules to their firewalls (rules that allow less) but not rules that allow more
  - they can also remove rules they added themselves, but not rules others have added
- handle the case of user A being allowed to execute a process as user B that has different privileges