Lecture 4
Lab 2 Supplemental Notes

Print version of the lecture in *CPS633 Computer Security*

presented on Week of September 23, 2019

by Ali Miri ©2019from Department of Computer Science at Ryerson University

4.1

Contents

1 Background **1**

2 Packet Filtering FWs **2**
2.1 LKMs 2

3 Netfilter **3**
3.1 hooks 3
3.2 iptables 3
3.3 ufw 4

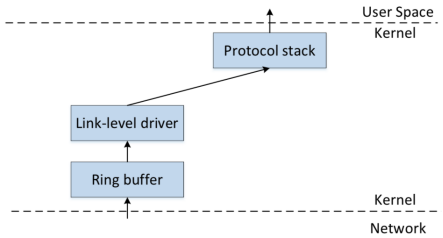
4 FW Evasion **5**
4.1 SSH Tunneling 5
4.2 Reverse SSH Tunneling 5

4.2

1 Background

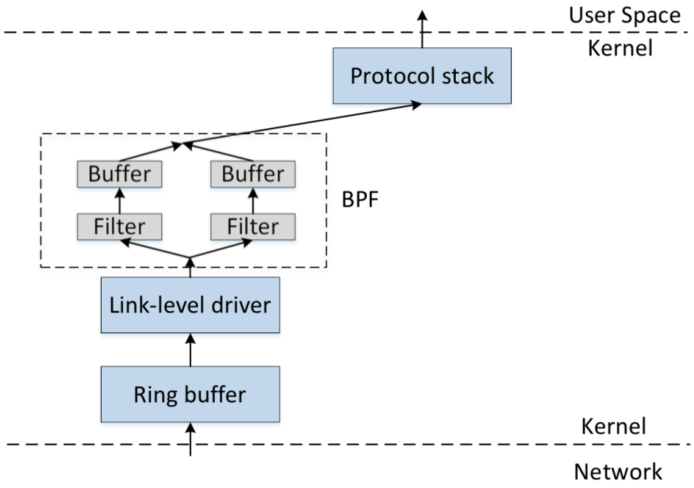
Packet Flow

- Most LANs use Ethernet and Wifi ← broadcast medium
- All nodes are connected to *single* shared medium → every node on the 'wire' will *hear* all the broadcasted frames
- NIC's job: copy all the frames arriving on the medium into its memory, and check its destination → if matches the NIC's MAC address, copy the frame into the kernel buffer



Most of the figures and content are from Chapter 17 of Wenliang Du, 'Computer and Network Security' book.

4.3

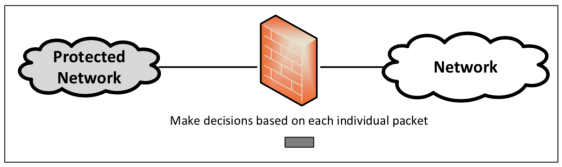


4.4

2 Packet Filtering FWs

Packet Filtering Firewalls

- Packet filtering is done inside the kernel ← need to run a kernel code



- *Loadable Kernel Modules* and *Netfilter*

4.5

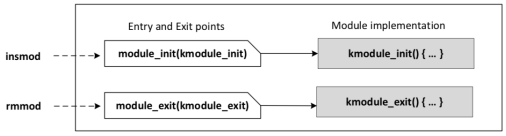
2.1 LKMs

Loadable Kernel Modules

Kernel modules

- are pieces of codes that can be loaded and unloaded on-demand at runtime
- do not run as specific process, but are executed in the kernel on behalf of the current process

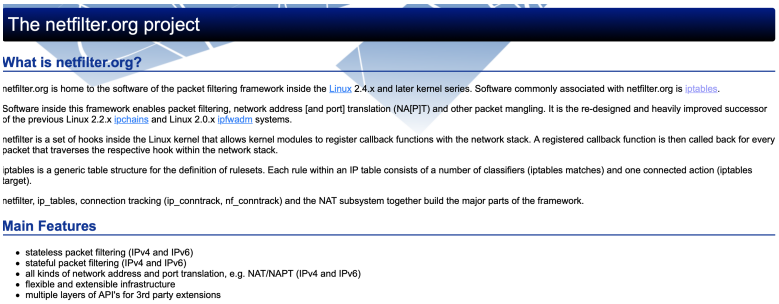
See Section 3 of the lab handout for simple examples



4.6

3 Netfilter

Netfilter



The netfilter.org project

What is netfilter.org?

netfilter.org is home to the software of the packet filtering framework inside the Linux 2.4.x and later kernel series. Software commonly associated with netfilter.org is iptables.

Software inside this framework enables packet filtering, network address [and port] translation (NA[P]T) and other packet mangling. It is the re-designed and heavily improved successor of the previous Linux 2.2.x ipchains and Linux 2.0.x ipfwadm systems.

netfilter is a set of hooks inside the Linux kernel that allows kernel modules to register callback functions with the network stack. A registered callback function is then called back for every packet that traverses the respective hook within the network stack.

iptables is a generic table structure for the definition of rulesets. Each rule within an IP table consists of a number of classifiers (iptables matches) and one connected action (iptables target).

netfilter, ip_tables, connection tracking (ip_conntrack, nf_conntrack) and the NAT subsystem together build the major parts of the framework.

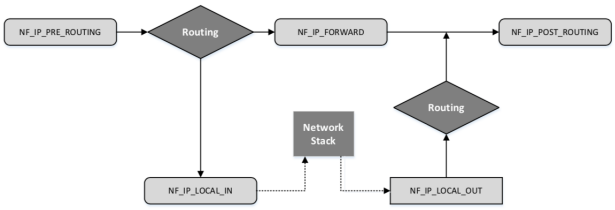
Main Features

- stateless packet filtering (IPv4 and IPv6)
- stateful packet filtering (IPv4 and IPv6)
- all kinds of network address and port translation, e.g. NAT/NAPT (IPv4 and IPv6)
- flexible and extensible infrastructure
- multiple layers of API's for 3rd party extensions

See `https://netfilter.org` 4.7

3.1 hooks

Netfilter Hooks for IPv4



4.8

3.2 iptables

iptables

- **iptables:** A simple, user-space firewall based on *netfilter* ← `Xtable`: kernel part implementation.
- check urlhttps://netfilter.org/projects/iptables/index.html, but also urlhttps://netfilter.org/projects/nftables/index.html
- It organized all its rules using a hierarchical structure
  - *tables*: all filtering rules in *filter*, *nat*, or *mangle* tables
  - *chains*: each table contains several chains, corresponding to *netfilter* hooks
  - *rules*: describe firewall filtering rules.

4.9

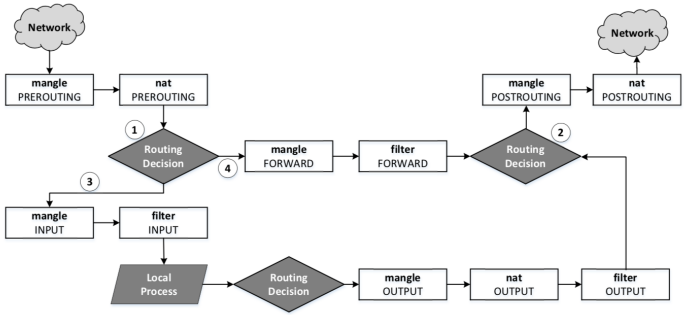
iptables 4.10

iptables

Traversing Chains and Rule Matching

Table 1: *iptables* Tables and Chains

Table	Chain	Functionality
filter	INPUT FORWARD OUTPUT	Packet filtering
nat	PREROUTING INPUT POSTROUTING	Modifying source or destination network addresses
mangle	PREROUTING INPUT FORWARD OUTPUT POSTROUTING	Packet content modification



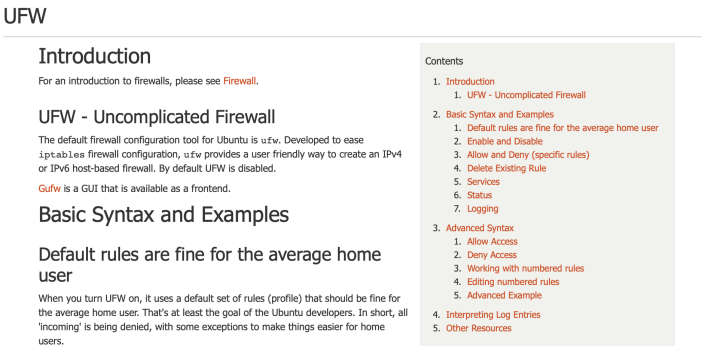
- Common actions: `ACCEPT`, `DROP`, or jump to a user-defined chain
- Functionalities of `iptables` can be extended using modules (may need to be downloaded and installed)

4.11

3.3 ufw

ufw

Uncomplicated Firewall (ufw): A user friendly `iptables` firewall configuration tool


UFW

Introduction

For an introduction to firewalls, please see Firewall.

UFW - Uncomplicated Firewall

The default firewall configuration tool for Ubuntu is `ufw`. Developed to ease `iptables` firewall configuration, `ufw` provides a user friendly way to create an IPv4 or IPv6 host-based firewall. By default UFW is disabled.

Gufw is a GUI that is available as a frontend.

Basic Syntax and Examples

Default rules are fine for the average home user

When you turn UFW on, it uses a default set of rules (profile) that should be fine for the average home user. That's at least the goal of the Ubuntu developers. In short, all 'incoming' is being denied, with some exceptions to make things easier for home users.

Contents

1. Introduction
   1. UFW - Uncomplicated Firewall
2. Basic Syntax and Examples
   1. Default rules are fine for the average home user
   2. Enable and Disable
   3. Allow and Deny (specific rules)
   4. Delete Existing Rule
   5. Services
   6. Status
   7. Logging
3. Advanced Syntax
   1. Allow Access
   2. Deny Access
   3. Working with numbered rules
   4. Editing numbered rules
   5. Advanced Example
4. Interpreting Log Entries
5. Other Resources

see `https://help.ubuntu.com/community/UFW`

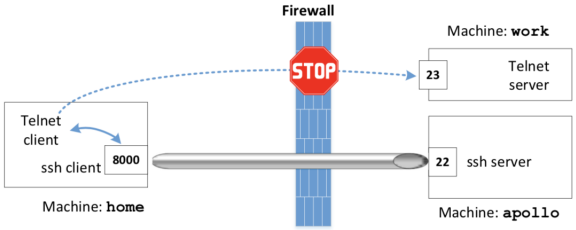
4.12

4 FW Evasion

4.1 SSH Tunneling

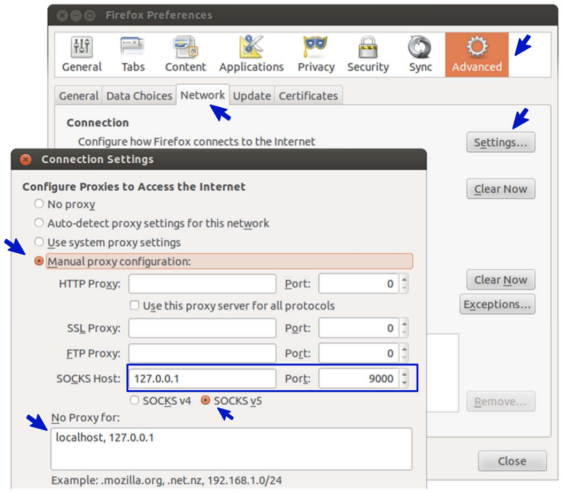
SSH Tunneling

- To evade firewall, need to hide the real purposes of network traffic
- A common technique is to sue VPN or SSH tunnel for ingress and egress traffic



4.13

SOCKS Proxy

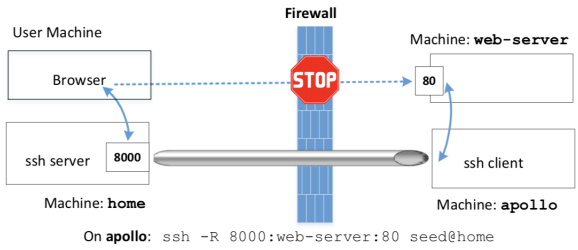


`Telnet` does not have a native SOCKS proxy support, but `HTTP` does.

4.14

4.2 Reverse SSH Tunneling

Reverse SSH Tunneling



4.15