Firewalls and Filters

Linux Systems Security

Firewall

- A physical barrier designed to slow or prevent the spread of fire
- In computer networks, a mechanism to slow or prevent the passage of network traffic
- Several firewall software packages have come and gone over the past 20 years, iptables is ubiquitous for Linux currently
- Many front-end software packages have been created to manage iptables for you because iptables is considered difficult to work with directly

netfilter.org

- Netfilter is the home of several packet filtering projects including iptables, which is used in most modern Linux kernels
- GPLv2 licensed, open source, in active development since approximately 1999
- Corporate sponsors include Watchguard, LinuxCare Inc., Connectiva, Sophos, and many others

Stateful vs. Stateless

- 1st generation packet filters were stateless network layer filters - each packet was examined on an individual basis and decisions about it were based solely on the contents of that packet
- 2nd generation packet filters incorporated connection information and could make stateful decisions as well - SPI
- 3rd generation adds application awareness and can make decisions based on unexpected traffic patterns - deep packet inspection

NAT

- NAT was developed to deal with limited address space in IPV4
- It was quickly recognized that it also provided the function of hiding internal addresses making reconnaissance more difficult for attackers
- Many firewalls provide NAT as an added tool for slowing attackers

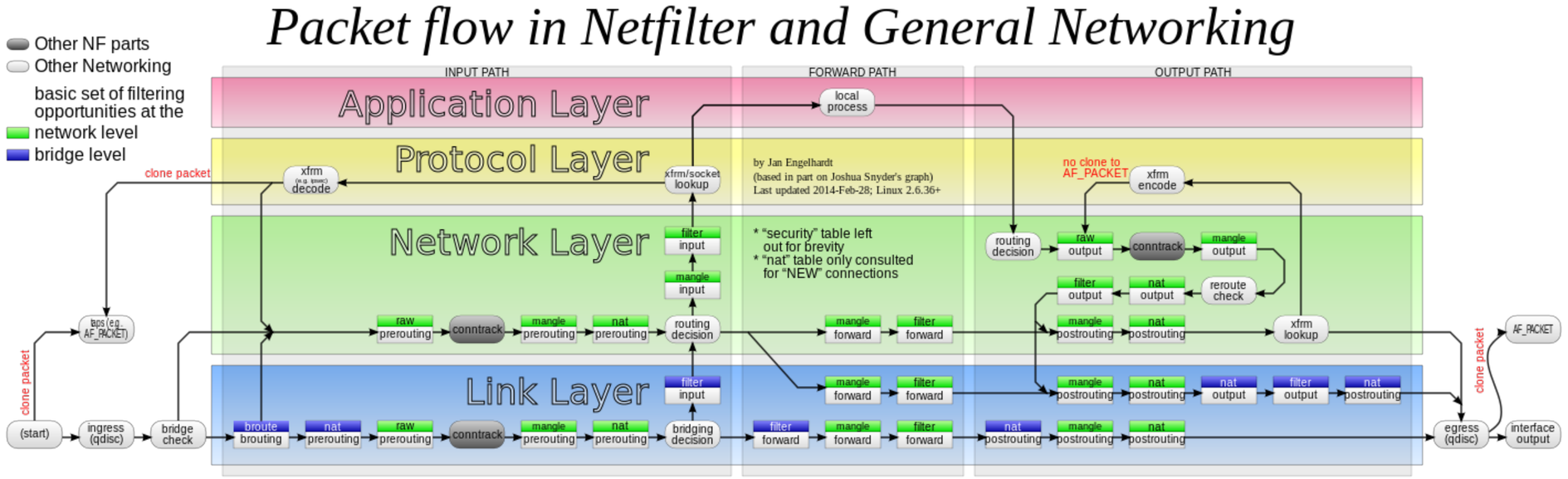
Proxies

- A proxy is a software device which provides a middleman for connections and can perform additional filtering of traffic
- Useful for implementing more complex application-specific rules such as url-based filtering
- Email MTAs can perform a proxy function for email
- Firewalling external connections from non-proxy hosts can add a layer of protection against internal hosts which have been compromised or have misuse attempted on them

iptables Tables

- iptables uses 5 built-in tables as the basis for managing traffic
- The filter table is the default table used to manage traffic
- The NAT table is used to perform address modifications in order to provide NAT and is applied to new connections
- The mangle table is used to modify packets in specialized ways
- The raw table is for intercepting and customizing connection tracking
- The security table is for MAC rules, such as from SELinux and takes effect after the filter table rules
- Tables contain chains of rules

Packet flow



"Netfilter-packet-flow" by Jengelh - Own work, Origin SVG PNG. Licensed under CC BY-SA 3.0 via Commons - https://commons.wikimedia.org/wiki/File:Netfilter-packet-flow.svg#/media/File:Netfilter-packet-flow.svg

iptables Chains

- A chain is a sequence of rules
- INPUT, OUTPUT, and FORWARD are the built-in chains
- INPUT is applied to packets destined for this host from network interfaces
- OUTPUT is applied to packets generated by this host
- FORWARD is applied to packets not generated by, or destined for, this host
- A chain also has a policy, which is what happens to packets not specified in the rules
- Create your own chains with iptables -N, delete them with iptables -X

iptables Chain Policy

- Each rule in a chain can specify parameters to identify packets that the rule applies to and an action to take if the packet matches the parameters
- If a packet is compared to all the rules and does not match any of them, the policy for the chain is applied to the packet
- The default policy after installation is ACCEPT
- Other policies available include DROP and REJECT

iptables Rules

- Each rule in a chain can have a number of parameters including a target
- Typical parameters might include
  - chain name
    interface name
    protocol (name or number from /etc/protocols)
    source address name/number/cidr range and/or port name or number from /etc/services
    destination address name/number/cidr range and/or port name or number from /etc/services
    jump target
- Builtin targets include ACCEPT, DROP, REJECT, LOG
- Additional targets can be other chains which allows you to clarify your chains
- Extensions can also be targets - see iptables-extensions(8)

iptables Command

- iptables -V to get version info
- iptables -L [-v] to get config summary
- iptables -S to show rules in iptables command line format
- iptables -A to append rules to a chain
- iptables -I to insert rules into a chain other than at the end
- iptables -F to flush rules from a chain
- ip6tables command builds rules for IPV6, or in newest distro releases use the options to specify ipv6 with the iptables command

iptables Examples

iptables -A INPUT -i lo -j ACCEPT ; iptables -A OUTPUT -o lo -j ACCEPT

iptables -A INPUT -i eth0 -p tcp --dport ssh -j ACCEPT

iptables -A INPUT -i eth0 -p tcp ! --dport ssh -j LOG --log-prefix "DENIED: "

iptables -A OUTPUT -o eth0 -p tcp --sport 22 -j ACCEPT

iptables -A OUTPUT -o eth0 -p tcp ! --sport 22 -j LOG --log-prefix "DENIED: "

iptables -P INPUT DROP

iptables -P OUTPUT DROP

What common network traffic might break because of this? How would you discover what was broken?

iptables Extensions

- Extensions exist for iptables and add packet matching capabilities using modules as well as new targets to give more options about what to do with matched traffic
- -m option can be used to specify a module to process matching traffic
- Some modules permit options
- Interesting modules:
  limit, connlimit, conntrack, iprange, multiport, comment
- Interesting targets:
  LOG, REDIRECT, TEE

http://ipset.netfilter.org/iptables-extensions.man.html

iptables Persistence

- iptables is a memory-based utility
- To have the rules take effect at boot, we need to use software that not only installs the rules, but saves those rules for reinstallation at next boot
- Most higher level packages that try to automate firewall management save the rules you create
- You can install the iptables-persistent package and save your rules to /etc/iptables/rules.v{46} using ip[6]tables-save
- You can use one or more of several packages intended to manage an iptables configuration

Common Attack Handling

- Drop or limit pings from all non-local hosts, limiting icmp rates across the board can help against smurfs
- Drop packets sourced from private netblocks which you aren't using yourself
- Drop malformed packets using --tcp-flags, port scans often use these
- Configure appropriate kernel tuning parameters to increase resilience to attacks
- Modern Linux kernel is quite robust in major distros, most attacks are on services so block or limit them and use whatever config options are available to you in those services

iptstate

- top-style tool for observing connection states
- Requires at least one rule that uses conntrack or state extension in order to provide state capture
- help screen available with h key, shows current sort and display settings

UFW

- Uncomplicated Firewall
- A command line utility to simplify firewall management
- Uses pre-configured rulesets for common configurations, with catch-all rules in /etc/ufw
- It is a front end to the iptables command, but conflicts are probable if you use both to set up your firewall - instead use the pre and post rules files in ufw to set up custom rulesets
- Provides enable/disable and configuration save
- gufw is a graphical frontend to ufw

https://help.ubuntu.com/lts/serverguide/firewall.html

https://help.ubuntu.com/community/Gufw

ipkungfu

- Another frontend to iptables (there are many, e.g. https://taufanlubis.wordpress.com/2007/09/23/need-proctection-for-your-ubuntu/)
- Uses a relatively friendly configuration file and supports automatic config at boot
- Groups many rule ideas into simpler concepts and makes them options in config files
- Designed and intended for use on a router, but with configuration changes can be used on any machine

https://help.ubuntu.com/community/firewall/ipkungfu

fail2ban

- fail2ban is a package that can scan log files looking for repeated login failures and then block the source hosts using iptables
- It does not require chain DROP policy, so if you don't have a deny policy, it will still work
- fail2ban knows many common log file formats such as ssh, web servers, email servers, ftp, and many applications that sit on top of those services
- see /etc/fail2ban/filter.d for the logs it knows, /var/log/fail2ban.log to see what it has been doing when running
- copy /etc/fail2ban/jail.conf to /etc/fail2ban/jail.local and modify to enable or configure jails
- fail2ban.org, 2014 PyCon video: https://www.youtube.com/watch?v=xcXheAWy7cU#t=190

Additional Filtering

- Proxy servers (email, web, etc.) can be set up
- Use iptables to prevent connections for proxied services that try to bypass the proxies
- Proxies can do application-level filtering