Tables and Chains

- How a packet traverses the inside of the kernel
  - You really need to understand this
  - Mistakes are easy to make
- 3 basic tables (categories of functionality)
  - filter (default)
  - nat
  - mangle
- Each table contains one or more chains
- User-defined chains may be specified in a table
- The pre-defined chains may then call any user-defined chains
Filter table

- Used for filtering
- Contains 3 chains
  - INPUT
  - OUTPUT
  - FORWARD
- Certain targets may not be used here
  - NAT targets
  - Mangle targets
- Filtering targets works perfectly
NAT table

- Used for Network Address Translation
- Only the first packet of a connection hits this table
  - Subsequent packets in the connection have the same action taken
  - Avoid pure filtering in this chain!
- Contains 3 chains
  - PREROUTING
  - POSTROUTING
  - OUTPUT
Mangle table

- Used for mangling packets
- Only the first packet in a connection hits this table
  - Subsequent packets in the connection have the same action taken
  - Avoid pure filtering in this chain!
- Contains 5 chains
  - PREROUTING
  - POSTROUTING
  - OUTPUT
  - INPUT
  - FORWARD
Iptables syntax

- iptables [ -t table ] command [options] [match] [target]
- Tables:
  - filter (default table)
  - nat
  - mangle
- Commands:
  - are we inserting, adding, or deleting a rule?
  - append, insert, replace, delete, list, policy, etc.
- Options:
  - verbose, line numbers, exact, etc.
- ...

Artjom Lind (artjom.lind@ut.ee)
28.10.2013
https://courses.cs.ut.ee/2013/syshald
Iptables syntax

• ...

• Match:
  - we’re telling the kernel what kind of packet(s) we’re looking for
  - dport, dst, sport, src, states, TCP options, owner, etc.

• Targets:
  - we’re telling the kernel what to do with the packet(s)
  - ACCEPT, DROP, REJECT, SNAT, DNAT, TOS, LOG, etc.
Commands

- **-A, –append**
  - `iptables –append INPUT yadda yadda yadda`

- **-D, –delete**
  - `iptables –delete INPUT yadda yadda yadda`
  - `iptables –delete INPUT $somenumber`

- **-R, –replace**
  - `iptables –replace INPUT $somenumber`

- **-I, –insert**
  - `iptables –insert INPUT $somenumber`
  - default insertion point is 1

- **-Z, –zero**
  - `iptables –zero INPUT`

- **-P, –policy**
  - `iptables –policy INPUT DROP`
matches

- Protocol
  - `-p, –protocol [!] [protocol]`
    - tcp, udp, icmp or all
    - Numeric value
    - `/etc/protocols`
- Destination IP & Port
  - `-d, –destination [!] address[/mask]`
  - Destination address
  - Resolvable (`/etc/resolve.conf`)
- `–dport, –destination-port [!] port[:port]`
  - Destination port
  - Numeric or resolvable (`/etc/services`)
  - Port range
matches

- Source IP & Port
  - -s, –source [!] address[/mask]
  - Source address
  - Resolvable (/etc/resolve.conf)
- –sport, –source-port [!] port[:port]
  - Source port
  - Numeric or resolvable (/etc/services)
  - Port range
matches

• Incoming and Outgoing interface
  – -i, –in-interface [!] interface
    • Input interface
    • mask
• -o, –out-interface [!] interface
  – Output interface
  – mask
targets

• ACCEPT
  - Accepts the packet
  - Ends further processing of the specific chain
  - Ends processing of all previous chains
  - Except other main chains and tables

• DROP
  - Drops the packet
  - No reply
  - Ends all further processing
targets

• REJECT
  - Drops packet
  - Returns a reply
    • User specified reply
    • Calculated reply
    • TCP-RST or ICMP errors
  - Ends all further processing

• RETURN
  - Returns from a chain to the calling chain
iptables -A INPUT \
-p tcp -m state –state NEW ! –syn -j \
REJECT –reject-with tcp-reset

- table : filter
- -A : appending rule to the chain
- chain : INPUT
- match : protocol = TCP

- -m : explicit match on state for a NEW packet
- ! : logical not
- –syn : SYN bit set and ACK, RST bits unset in packet
- mismatches packets in an established connection
- target : REJECT the packet and send TCP RST
- (closes the open connection gracefully)
Introduction

- E-Mail - method of exchanging digital messages from an author to one or more recipients
- E-Mail – easily accessible and popular internet service today
  - As a result also popular target of cyber-criminals and spammers.
- Employs several standards and protocols that define how the messages are composed and delivered
Introduction

- Developed by the Defense Advanced Research Projects Agency (DARPA) in 1969
- Introduced '@' sing to separate user's name and user's machine
- Initially adopted to ARPANET (later Internet) as it grown in popularity over older networks:
  - ? do you know Internet alternative networks?
Introduction

- Developed by the Defense Advanced Research Projects Agency (DARPA) in 1969
- Introduced '@' sing to separate user's name and user's machine
- Initially adopted to ARPANET (later Internet) as it grown in popularity over older networks:
  - BitNet, FidoNet, DecNet, CsNet ...
Introduction

- There different software components evolved in the process of message delivery:
  - Mail User Agent (MUA)
    - Reading and composing messages
      - Alpine, Netscape, Outlook, Thunderbird, ...
  - Mail Submission Agent (MSA)
    - Taking care of delivery from sender to MTA
  - Mail Transfer Agent (MTA)
    - Taking care of mail transfer between MSAs and MDAs
  - Mail Delivery Agent (MDA)
    - Taking care of mail delivery to final recepient
Mail Transfer Agent MTA

- Move email from one host or network to another
- In contrast to MSA and MDA, which move mail only within single system
- In other words:
  - MTAs are like mail trucks or planes that move mail between countries or cities
  - MDAs – are the letter carriers who deliver mail to users' mailboxes
  - MSAs – local mail droboxes
Mail User Agent (MUA)

- Mail Client
  - POP, POP3, IMAP protocol clients
  - Fetching the mail from remote Mailbox
  - Read/Write mails
  - Pushing outgoing to remote Mail Sending Agent
- SMTP Clients
Postfix Mail Transfer Agent

- Open Source secure mail transfer agent (MTA)
- Routing and delivering mail
- Default MTA for a number of Unix-like operation systems
Postfix

- Simpler design
- Modular architecture
- Easier to configure
- Less work to administer
- Designed for scalability, reliability and security as primal requirements
Postfix

- Is in Debian, Ubuntu main repositories
  - Critical security updates guaranteed
- Supporting
  - LDAP Lightweight Directory Access Protocol
    - maintaining distributed directory information services
    - Usually user profile data
  - SMTP Simple Mail Transfer Protocol
  - SASL Simple Authentication and Security Layer
  - TLS Transport Layer Security