Securing the network

Dorin RAUTU
dorin.rautu@enseeiht.fr

12/04/2019

Topics

- What is a firewall
- Packet filtering
- iptables

What is a firewall

- Definition
- Network Functions
- Examples

What is a firewall

- A mechanism used to block unwanted traffic from the network
- Can be implemented:
  - On a network device
    - Cisco router
  - Like a dedicated device
    - Cisco ASA
    - Fortinet Fortigate
  - On an end-device (host or server)
    - ZoneAlarm
    - Windows Firewall
    - Netfilter / iptables

Why do we need them

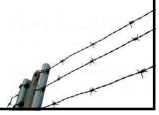
- The Internet is not a safe place
- The local network can always be the target of an attack:
  - Network scanning
    - Ping sweep
    - Sniffing
    - Port scan
  - By DoS (Denial of Service) or DDoS (Distributed DoS)
    - Smurf attack
    - SYN flood
  - Forcing access
    - Finding a password (dictionary or brute-force)
    - Buffer overflow
    - Man-in-the-middle

The role of a firewall

- Network scanning
  - The attacker is trying to find the PCs and services running on them
    - Example: ICMP echo request to a broadcast address discovers all machines on the network
  - A Firewall can:
    - Block vulnerable ports
    - Block external connection initiation
    - Block response to ICMP echo request

The role of a firewall

- DoS or DDoS attack
  - Generally based on generating a large amount of traffic, overloading the network or server
  - Because of overloading, new traffic is likely to be ignored
  - A Firewall can:
    - Monitor the number of TCP Half-Open sessions on server and close them if they cross a threshold
    - Block directed broadcasts



Firewall types

- Stateful firewall
  - Tracks the operating state of network connections
  - It can distinguish legitimate packets
  - Only packets matching a known active connection are let through
- Stateless firewall
  - Restrict or block packets based on source and destination addresses or other static values
  - They are not “aware” of traffic patterns or data flows

iptables

- Functions
- Structure
- iptables tables
- Predefined chains

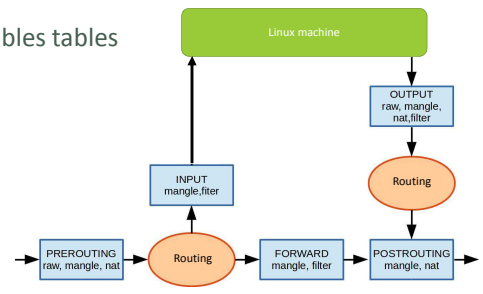
iptables

- Allows a Linux machine to:
  - Filter the packages
  - Translate address
  - Rewrite fields of a package
  - Configured by writing **rules**
- The iptables rules are composed of two main sections:
  - Pattern
    - What values the fields in the package must have to act on them
  - Action
    - What operation the Linux machine will perform on the package

Iptables tables

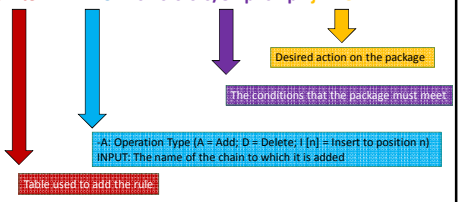
- Filter
  - It contains rules that specify what traffic can pass and what traffic should be discarded
  - Example:
    - An external address has repeatedly failed to connect to a Linux server through SSH
    - You should add a filtering rule that blocks any traffic from this address
- Nat
  - Contains rules for address translation in the NAT (Network Address Translation) process
  - Example:
    - A private address must access an Internet server
    - You should add a NAT rule that rewrites the private source address with a public address
    - Upon return, the address will be rewrited back
- Mangle
  - Contains rules for specialized packet alteration
  - Example:
    - Modifying packet fields like: TOS(Type of Service), TTL(Time to Live), MARK(packet labeling)

Iptables tables



How to use it

ubuntu# iptables **–t filter –A INPUT –s 10.0.0.0/8 –p icmp –j DROP**



How to use it

ubuntu# iptables **–t filter** –A INPUT –s 10.0.0.0/8 –p icmp –j DROP

- The default table is the filter
- The rule could therefore be shortened as:
  - ubuntu# iptables –A INPUT –s 10.0.0.0/8 –p icmp –j DROP
- The options allowed for this parameter are:
  - filter
  - nat
  - mangle
  - raw
    - Used to set up connection monitoring exceptions

How to use it

ubuntu# iptables –t filter **–A INPUT** –s 10.0.0.0/8 –p icmp –j DROP

- **INPUT** can be replaced with any other predefined chain
- New chains can also be created by the administrator
- Allowed operations are:
  - **–A** -- append — Adds a rule at the end of the chain
  - **–D** -- delete — Deletes a rule
  - **–L** -- list
  - **–F** -- flush — Deletes all rules
  - **–N** -- new-chain — Creates a new chain
  - **–X** -- delete-chain — Deletes a chain
  - **–P** -- policy — Modifies the global policy of the chain

How to use it

ubuntu# iptables –t filter –A INPUT **–s 10.0.0.0/8 –p icmp** –j DROP

- Traffic selection is based on package information
- Without specifying a protocol, rules can be made containing:
  - Input interface (**-i**)
  - Output Interface (**-o**)
  - Destination IP Address (**-d**)
  - Source IP Address (**-s**)

How to use it

ubuntu# iptables –t filter –A INPUT –s 10.0.0.0/8 –p icmp **–j DROP**

- The operation that will be done on the package
- In iptables terminology, **j** comes from jump and **DROP** is a target
- Can be omitted
  - In this case, the rule does nothing, but the rule counter will be incremented
- Common targets are:
  - ACCEPT: The package is accepted
  - DROP: The package is discarded
  - REJECT: The package is discarded, but the sender is notified
  - LOG: A record is added to the system logs

How it works

- When a packet arrives, it is sequentially evaluated according to each rule in a chain (from top to bottom – inserting a rule to a defined positions is important)
- If a match is made on a rule with an **ACCEPT** or **DROP** target, the processing ends and the package is accepted or discarded
- What happens if you do not match on any rule?

iptables policies

- Each predefined chain has a default policy
- User-created chains CAN NOT have default policy
- The policy is a target that is chosen for each package that does not match any of the chain rules
- Implicit policy: ACCEPT
- The policy of a chain can be changed:

```
ubuntu# iptables –P FORWARD DROP
```

iptables extensions

- Often, IP addresses and physical interfaces are not enough to implement security requirements
  - Can I only allow access to the HTTP service?
  - Can TCP connections be set in one direction only?
  - Can incoming pings be dropped, still keeping the ping to the outside?
- Iptables allows activation of **extensions**, modules that offer new possibilities in specifying the rules
- Extensions are enabled with -p (protocol) or -m (modules)
- The most important extensions are:
  - tcp
  - udp
  - ICMP

iptables extensions

- The tcp extension allows you to filter traffic by:
  - Destination port **--dport --destination-port**
  - Source port **--sport --source-port**
  - TCP flags (SYN, ACK, FIN, etc.) **--tcp-flags, --syn**
- The icmp extension allows for traffic filtering by:
  - ICMP package type **--icmp-type <type>** where type can be:
    - echo-request
    - echo-reply
    - time-exceeded
- For all the type values, you can run:

```
ubuntu# iptables -p icmp -h
```