Project 4 - Linux iptables

CSE497b - Spring 2007
Introduction Computer and Network Security
Professor Jaeger
www.cse.psu.edu/~tjaeger/cse497b-s07/
Project Goals

• Specify iptables rules for your Playpen VM
  – for the INPUT chain only

• Sources
  – Write rules for interaction with 2 machines
    • 130.203.83.76
    • 130.203.83.75

• Rules
  – Prevent all UDP
  – Permit ICMP (ping), but limit message size
    • 1000 bytes from 75 and 10000 bytes from 76
  – TCP
    • 130.203.83.75 sends to specific ports (no one else can use)
    • 130.203.83.76 sends to specific ports (no one else can use)
    • Also, some content filtering of packets
Project environment

• ICMP via ping
  – We will submit ping requests to your Playpen
  – Only allowed ones should result in a response

• TCP via nc
  – nc for netcat
  – nc -l -p <port> creates a server
  – nc -p <clientport> <addr> <port> connects a client
  – We supply the server program, client program
    • and expected output

• Due April 20 at 5:00
  – A bash script containing a sequence of iptables rules
  – Need to have the server program running at this time, so we can test!
Practical Firewall Implementations

• Primary task is to filter packets
  – But systems and requirements are complex

• Consider
  – All the protocols and services
  – Stateless vs. stateful firewalls
  – Network function: NAT, forwarding, etc.

• Practical implementation: Linux iptables
Netfilter hooks

- Series of hooks in Linux network protocol stack
- At each Netfilter hook
  - An iptable rule set is evaluated
- Hook placements
iptables Concepts

• Table
  – All the firewall rules

• Chain
  – List of rules associated with the chain identifier
  – E.g., hook name

• Match
  – When all a rule’s field match the packet (protocol-specific)

• Target
  – Operation to execute on a packet given a match
iptables Commands

- `iptables [-t <table_name>] <cmd> <chain> <plist>`
- Commands
  - Append rule to end or specific location in chain
  - Delete a specific rule in a chain
  - Flush a chain
  - List a chain
  - Create a new user-specified chain
  - Replace a rule
Test it out

• PING on localhost
  – `ping -c 1 127.0.0.1`

• Add iptables rule to block
  – `iptables -A INPUT -s 127.0.0.1 -p icmp -j DROP`

• Try ping

• Delete the rule
  – `iptables -D INPUT 1`
  – `iptables -D INPUT -s 127.0.0.1 -p icmp -j DROP`
  – `iptables -F INPUT`
Testing

• Use loopback to test the rules locally on your Playpen
  – IP address 127.0.0.1

• ICMP
  – submit ping requests to 127.0.0.1 as above

• TCP
  – submit requests to 127.0.0.1 at specific port
    – server
      • nc -l -p 3750
      • listen at port 3750
    – client
      • nc -p 3000 localhost 3750
      • send from port 3000 to localhost at port 3750
WARNING!

• Be careful!
  – You can lock yourself out of your Playpen

• Only write rules for the target IP addresses
  – localhost, 130.203.83.75, and 130.203.83.76

• Do not write any rules containing ssh

• We will have to restart your Playpen if you lock yourself out (not available 24/7)
Targets

- Define what to do with the packet at this time
  - **ACCEPT/DROP**
  - QUEUE for user-space application
  - LOG any packet that matches
  - REJECT drops and returns error packet
  - RETURN enables packet to return to previous chain
  - `<user-specified>` passes packet to that chain
iptables Rule Parameters

• Destination/Source
  – IP address range and netmask
• Protocol of packet
  – ICMP, TCP, etc
• Fragmented only
• Incoming/outgoing interface
• Target on rule match
Per Protocol Options

• Specialized matching options for rules
  – Specific to protocol

• TCP
  – Source/destination ports
  – SYN
  – TCP flags
Examples

- `iptables -A INPUT -s 200.200.200.2 -j ACCEPT`
- `iptables -A INPUT -s 200.200.200.1 -j DROP`
- `iptables -A INPUT -s 200.200.200.1 -p tcp -j DROP`
- `iptables -A INPUT -s 200.200.200.1 -p tcp --dport telnet -j DROP`
- `iptables -A INPUT -p tcp --destination-port telnet -i ppp0 -j DROP`
Match

- Different means for matching packet content
- Lots of different modules
  - Only a few supported on your Playpen (lucky you)
- To specify a match
  - `iptables -A INPUT -p tcp -m string --algo bm --string ‘exe’`
    - matches to packet with content containing ‘exe’
  - `iptables -A INPUT -p tcp -m length --length 10:100`
    - matches to packet with length between 10 and 100 bytes
    - Also, can specify ‘greater than 10’ by 10:
- There are many others, but these are what you’ll need to know