# Denial-of-Service [Gligor, 84]

``A group of otherwise-authorized users of a specific service is said to deny service to another group of otherwise-authorized users if the former group makes the specified service unavailable to the latter group for a period of time which exceeds the intended (and advertised) waiting time"

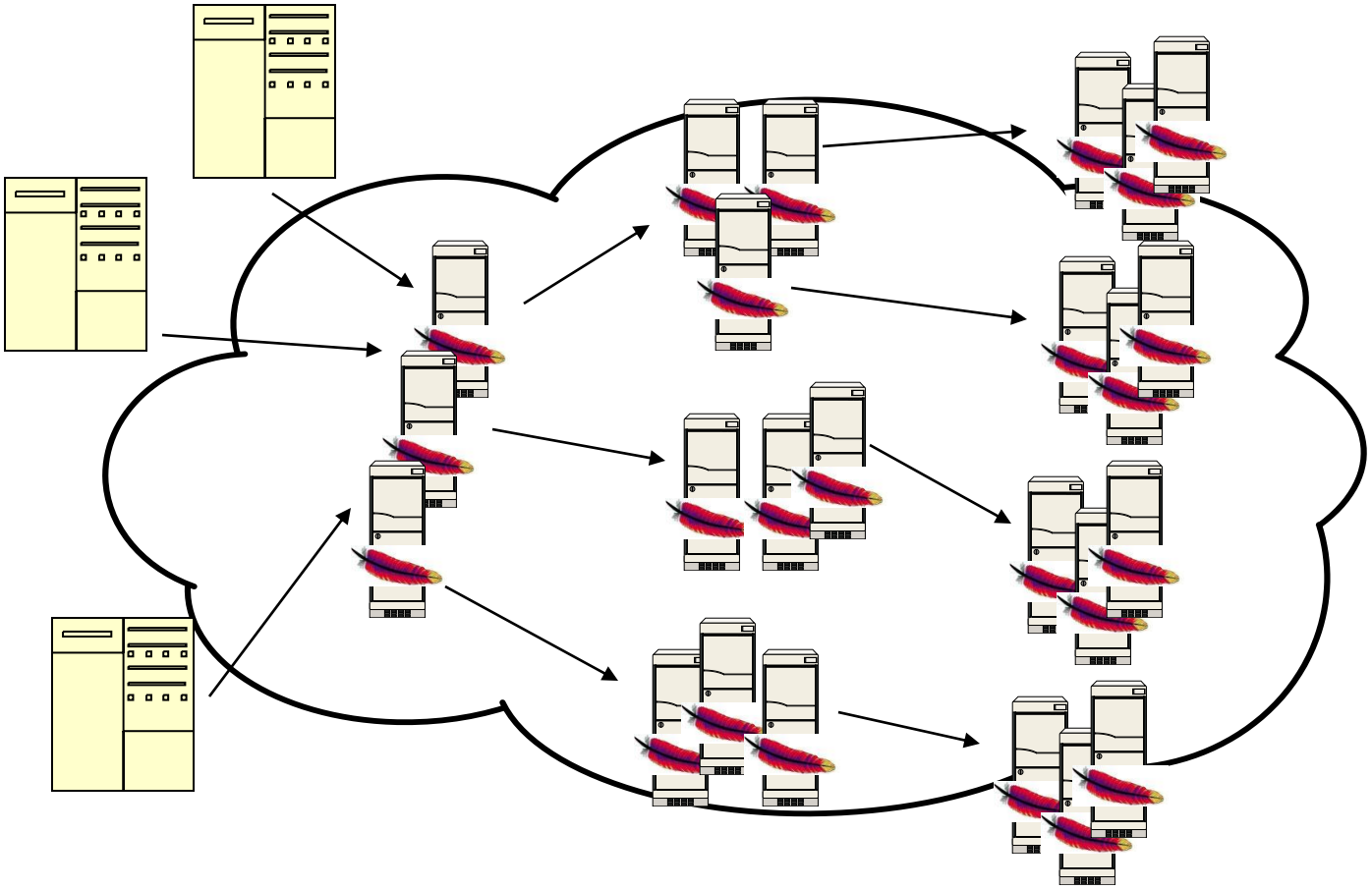
# DoS Attacks

- DoS attacks aim at reducing legitimate utilization of network and/or server resources through:
  - resource destruction (exploit bugs in the OS)
  - resource exhaustion
    - vulnerability exploitation (e.g., SYN attack)
    - brute-force flooding
      - Network-level (e.g., lots of packets as in UDP floods)
      - Service-level (e.g., flash crowds)

# Service-level DoS

- A large number of attack hosts request service from the victim server at a high rate. For instance,
  - download files from an FTP server, or
  - get web pages from an WWW server

# Front-ends

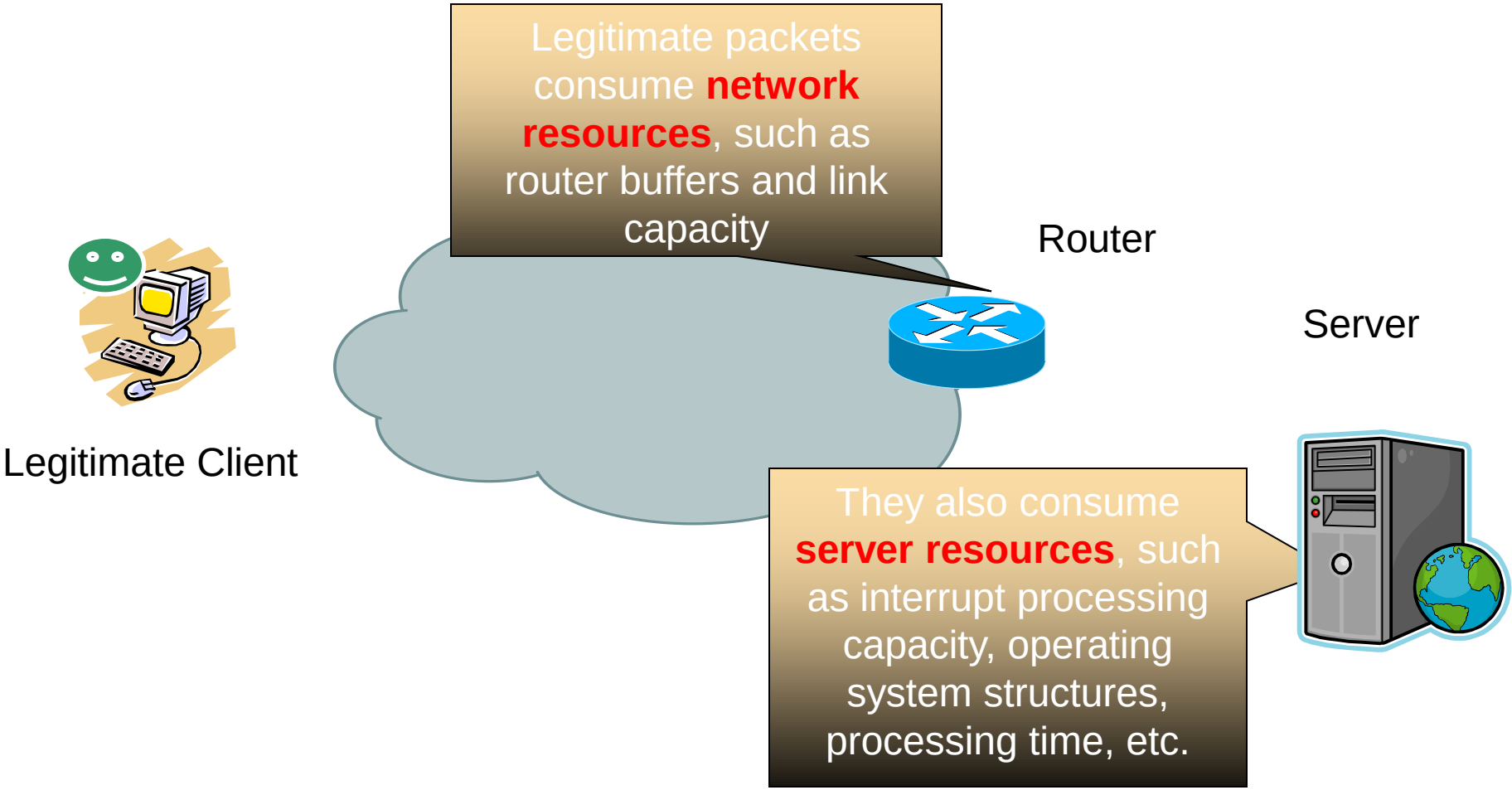


- Front-ends form a tree with the back-ends as its logical root.

# Front-ends (contd.)

- Tree level of each front-end depends on its attack tolerance
- Front-ends can be the bottleneck that gets attacked. It usually can withstand a good amount of attack traffic.
- To join the network (or reconfigure), a front-end performs:
  - Parent registration
  - Address registration

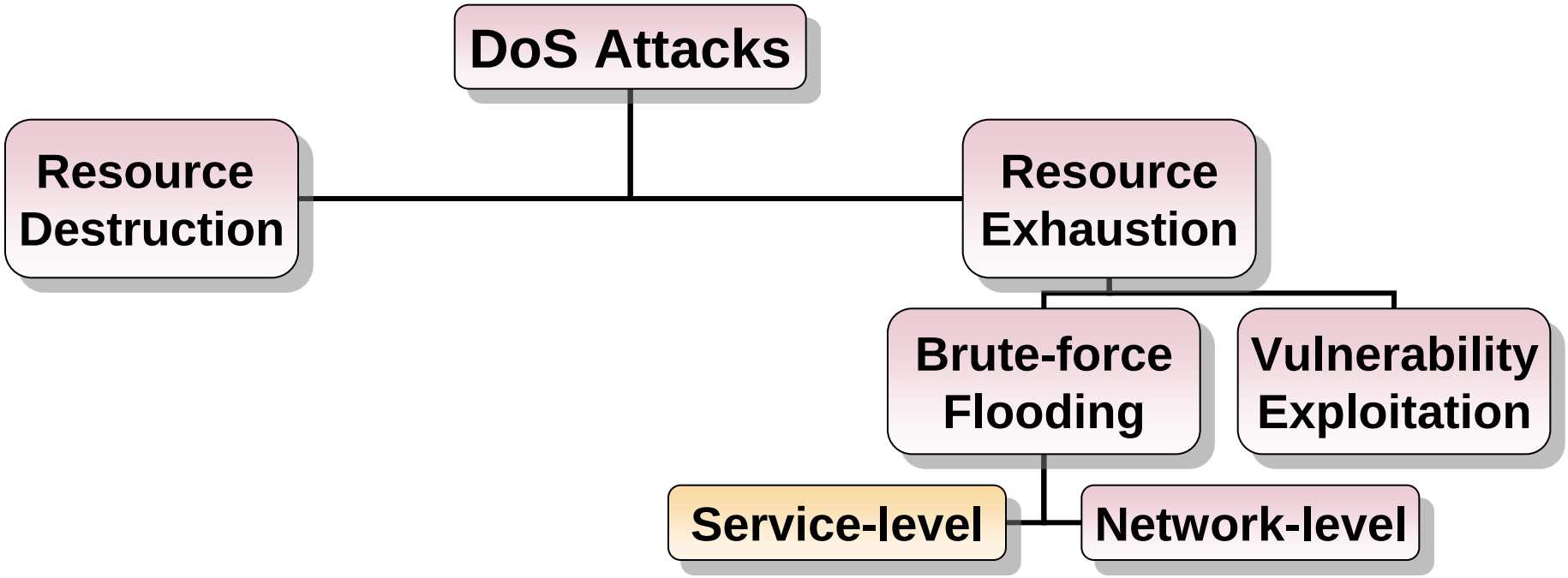
# DoS Attacks (1/4)



# DoS Attacks (2/4)

- Network-level DoS attacks flood network resources
- Service-level DoS attacks exploit vulnerabilities to crash servers
- Service-level DoS attacks flood server resources, so that legitimate clients' packets will be dropped…

# Our Focus: Service-level Flooding DoS



# The DoS Problem

Distinguish attack packets/requests from legitimate packets/requests

- quickly
- accurately (low false positives and false negatives) and
- efficiently (small overhead)

Primary metrics

- Legitimate Response Time
- Legitimate Throughput

# State-of-the-art

| | Prevention | Detection/ Recovery | Mitigation |
|---|---|---|---|
| Network-level | Network-level puzzles | PacketScore;<br>RED-PD;<br>Heavy-hitter detection;<br>DCAP; Pushback;<br>MOVE; Capabilities;<br>IP Hopping | Replication;<br>Overlay-based |
| Service-level | Application-level puzzles;<br>Reservation-based Schemes | DDoS Shield;<br>Shadow Honeypots;<br>Kill-Bots | Replication |

# Honeypots [Spitzner][Provos]

- Honeypots are:
  - decoy resources to trap attackers
  - useful in detecting worm-infected hosts
- However, honeypots are
  - at fixed locations
  - separate from real servers

DoS Attackers can evade honeypots

# Roaming Honeypots [Khattab]

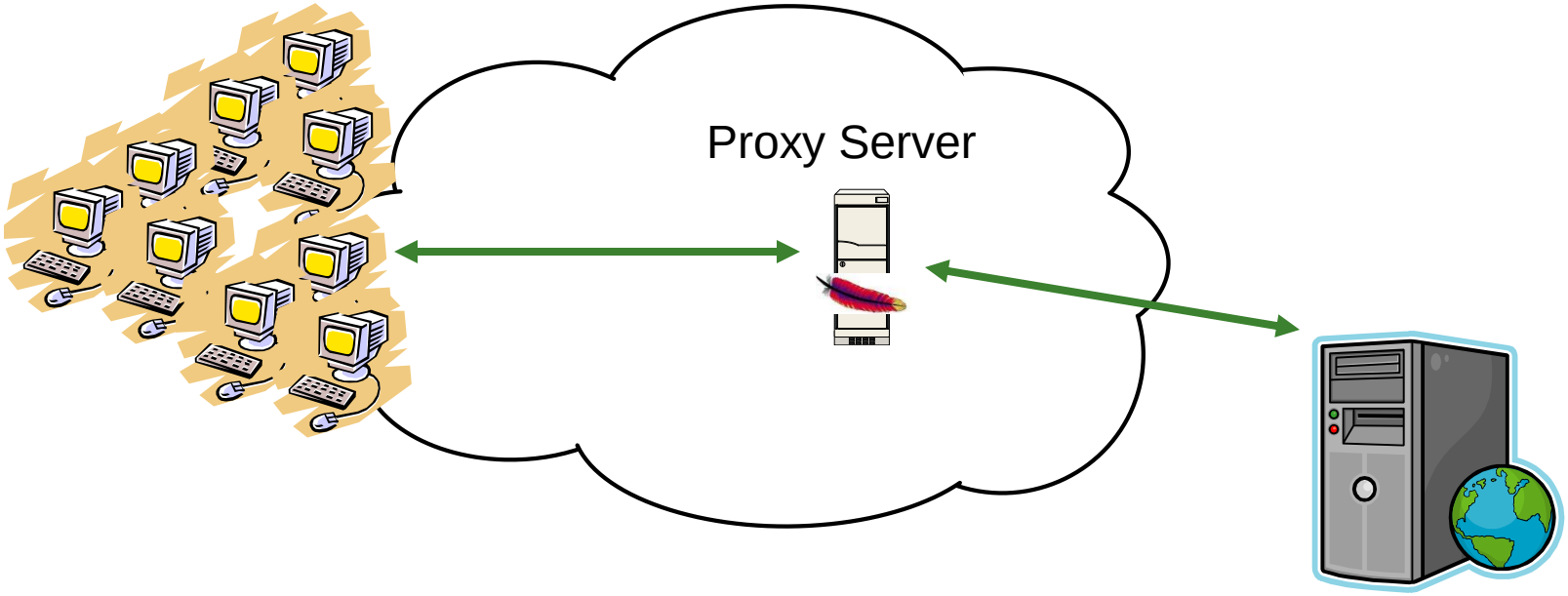
In roaming honeypots, the locations of honeypots are:

- continuously changing
- unpredictable to non-compliant attackers
- disguised within servers

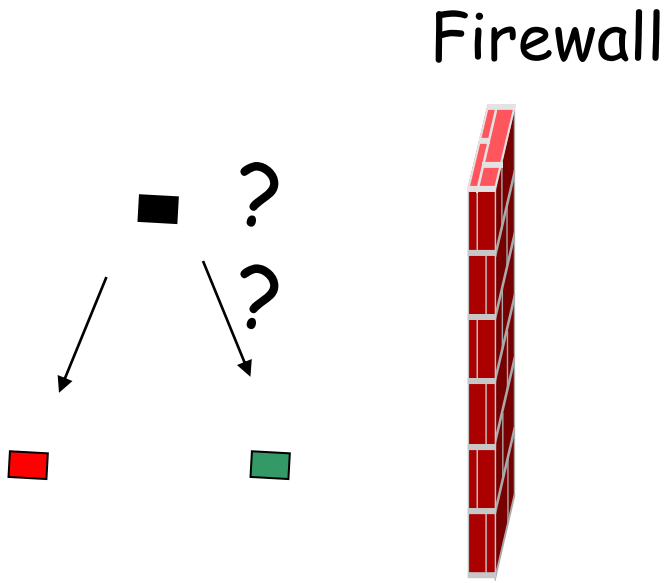
# Main Assumption

Unique, un-spoofable user identifier

(dealing with proxy servers is an open problem)



# Packet Filtering in firewalls




- White-list:
  - allow packets from certain users/Ips.
  - Not Scalable, because list grows with number of users
- Black list:
  - do not allow certain IPs or users.
  - More Scalable:
    # attackers << # users