A Parallel Packet Screen

for High Speed Networks

Carsten Benecke
benecke@fwl.dfn.de
Contents

- Packet screens
- Performance bottleneck
- How to use parallel processing?
- Prototype parallel packet screen
- Performance, reliability, and security
Packet screens

A well known building block for firewalls

- Coarse grained access control:
  - Access control based on packets
  - Examine ports, addresses, options, and flags
  - Permit or deny packets to pass on

- Protection for ...
  - Bastion hosts
  - Secure subnetworks
  - DMZ
  - ...

Packet screens

A well known building block for firewalls (2)

Simple PS

PS protects bastion

PS protects DMZ

PS protects DMZ
Packet screens

A well known building block for firewalls (3)
Packet screen between two networks
Packet screens

A well known building block for firewalls (4)
Packet screen protects bastion host
Packet screens

A well known building block for firewalls (5)
Packet screen protects bastion host

Diagram: A network diagram showing a bastion host connected to two networks, external net and internal net, through a packet screen (PS). The diagram illustrates the flow of communication between the networks and the bastion host.
Packet screens

A well known building block for firewalls (6)
Packet screen protects DMZ
Packet screens

A well known building block for firewalls (7)
Packet screens enclose DMZ

external net  PS  DMZ  PS  internal net

bastion host
Packet screens

A well known building block for firewalls (8)

Typical implementation:

- Part of router SW (black box)
  - “Fast”
  - “For free”
  - Router may be expensive
  - Administration is complex
  - Not optimized for access control
    - Complex features?
    - Support for new features?
    - Does it scale?
Packet screens

A well known building block for firewalls (9)

Typical implementation:

- Filter software (on workstations):
  - Optimized for access control
  - Support for integration of new features
  - Complex features (e.g., ‘stateful screening’)
  - Administration of standard HW
  - Workstations are inexpensive
  - Slow
  - Does not scale well
Performance bottleneck of PS

Setup

- Throughput depends on packet size and number of filter rules
  - UDP packet generator (*netperf*)
  - SW to configure filter rules (*IP-Filter*)

- Worst case scenario
  - Multiple communicating computers
  - Many different policies
    ⇒ no caching possible, each filter rule is applied to every packet!
Performance bottleneck of PS

Packet throughput

UDP Datagrams (DG) filtered by a Packet Screen
64 Kbyte Socket Size

theoretical Throughput
snd = rcv H2H
rcv, 0 FR
rcv, 10 FR
rcv, 50 FR
rcv, 100 FR
Parallel processing

Two problems to be solved

• How to distribute the packet load:
  – All instances need access to the packets
    ⇒ Mechanism needed to copy packets

• How to share the load:
  – All packets have to be screened
  – All instances should participate
    ⇒ Mechanism needed to select packets at each instance/processor
Parallel processing

Distributing the packet load

- Solution for distribution problem:
  - Hubs copy packets to parallel instances
  - All instances process the same code
  - Only ‘off-the-shelf’ equipment needed
Sharing the packet load

Requirements for a good packet selection algorithm

- Map each packet onto exactly one instance
- Distributed, not centralized
- Independent of number of instances
- Must be fast (additional overhead)
- Balance the load
Sharing the packet load

Prototype implementation

- Solution for load sharing:
  - Calculate: \([\text{IP checksum}] \mod [\text{number of instances}]\)
  - If result matches PID, screen packet
    else discard packet
- Handle ARP-packets as well
- Adaptable to handle connections
- Adaptable to heterogenous hardware
Prototype performance

Hub based approach (unidirectional packet flow)

UDP Datagrams (DG) filtered by Parallel Packet Screen
64 Kbyte Socket Size, Fast Ethernet, 0 Rules

snd, rcv (4WS)

max. theoretical Throughput, 1
snd
rcv (4WS)
rcv (3WS)
rcv (2WS)
rcv (1WS)

Looks nice but...
Prototype performance

Hub based approach (bidirectional packet flow)

UDP Datagrams (DG) and Throughput (Mbit/s) filtered by Parallel Packet Screen (4WS)
64 Kbyte Socket Size, Fast Ethernet, Hub, 0 Rules

max. theoretical Throughput (duplex)
snd sum hub
rcv sum hub
rcv Mbit hub

⇒ Many collisions on the hubs!
Prototype performance

**Evaluation: Hub based approach**

+ Good scaling characteristic
+ Even “old” HW is usable
+ Same level of utilization $\Rightarrow$ performance matches theoretical model
+ Easy to implement, easy to adapt

- Collisions cause throughput to decrease
- Why use hubs in switched networks?
Switch based approach

Solution for switched networks

- Replace hubs with switches
  - Separation of collision domains
  - Store and forward
  - 100MBit/s in each direction
    ⇒ Collision rate decreases, throughput increases

- More complicated configuration:
  - Combine instances to a multicast group
  - Adjust address resolution to MC address
  - Switches have to be secured
- Parallel PS is no longer transparent
Prototype performance

Switch based approach (bidirectional packet flow)

UDP Datagrams (DG) and Throughput (Mbit/s) filtered by Parallel Packet Screen (4WS)
64 Kbyte Socket Size, Fast Ethernet, Switch, 0 Rules

max. theoretical Throughput (duplex)
snd sum switch
rcv sum switch
rcv Mbit switch
Prototype performance

Evaluation: switch based approach

+ Similar to hub based approach
+ Also performs well in switched networks
+ Adaptable to:
  - VPNs (parallel encryption/decryption)
  - Parallel NAT
  - Parallel monitoring/audit \( \Rightarrow \) IDS

- Management more complicated
- No longer transparent
Performance, reliability, and security

Performance drawback due to failures

+ No security problem

But:
- Reliability decreases (parallel setup)
- Throughput decreases (due to failures)
- Failure detection required
- Automatic reconfiguration required
Performance, reliability, and security

Performance drawback due to failures (2)

UDP Datagrams (DG) filtered by Parallel Packet Screen
64 Kbyte Socket Size, Fast Ethernet, 0 Rules

Reconfiguration required for reliability and performance!
Conclusions

Ongoing work

- Simple distributed algorithm
- ‘Off-the-shelf’ equipment for:
  - Scalable screening
  - Scalable monitoring
  - Scalable VPNs

Ongoing work:

- Build and evaluate security components with increased survivability!