Directory Enabled Distributed Packet Filtration System

A Scalable and High Performance Security Architecture

Siddhartha Gavirneni
sgavirne@eecs.ku.edu

Overview

- Motivation & Goals
- The Evolving Security Model
- The Distributed Firewall Architecture
- A Cost effective solution: Load Balancing

Part-I

Distributed Security Architecture

- Distributed Firewall Policy Management
- The DEN Initiative
- Directory Enabled Policy Management

Part-II

Policy Management

- KU and the Distributed Security Architecture
- Conclusion and Future Work

Motivation & Goals - 1

Motivation:

Existing monolithic firewall architectures

Goal:

Analyze the concepts of a distributed security architecture for large enterprise networks

Motivation & Goals - 2

Motivation:

High cost commercial firewalls

Goal:

A low cost solution: Load balancing of non-commercial firewalls/packet filters

Motivation & Goals - 3

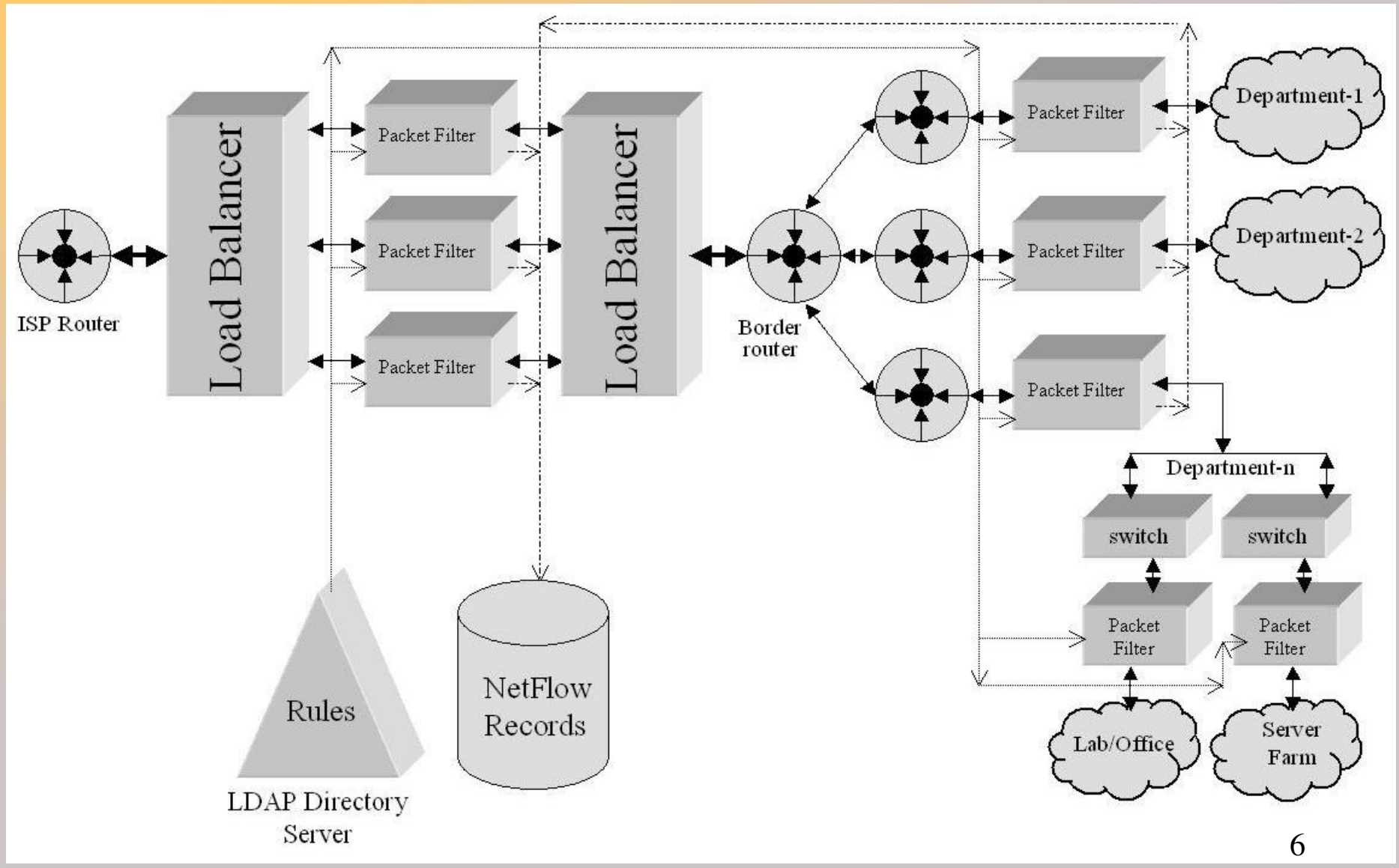
Motivation:

Maintaining the policies for all the firewalls in a distributed architecture, especially for a large network, is a mammoth task

Goal:

The Directory Enabled policy management system

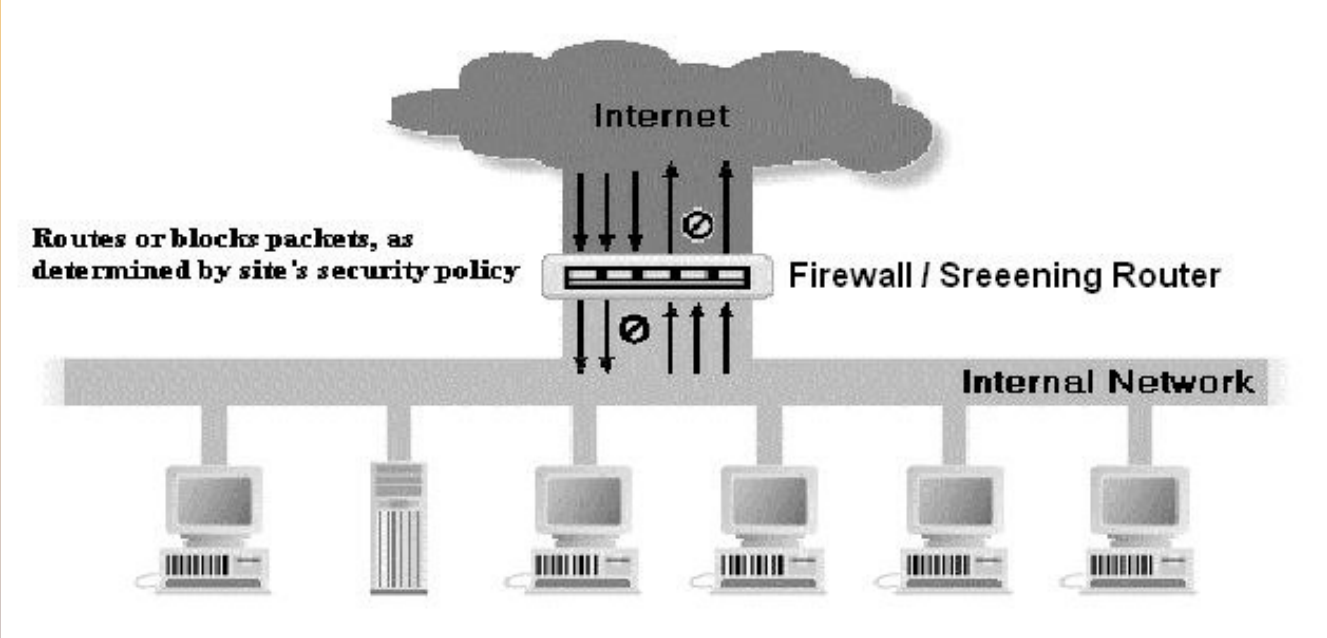
Motivation & Goals – The Complete Picture



The Evolving Security Model

Stage 0: No Firewall

Stage 1: Single Firewall Architecture

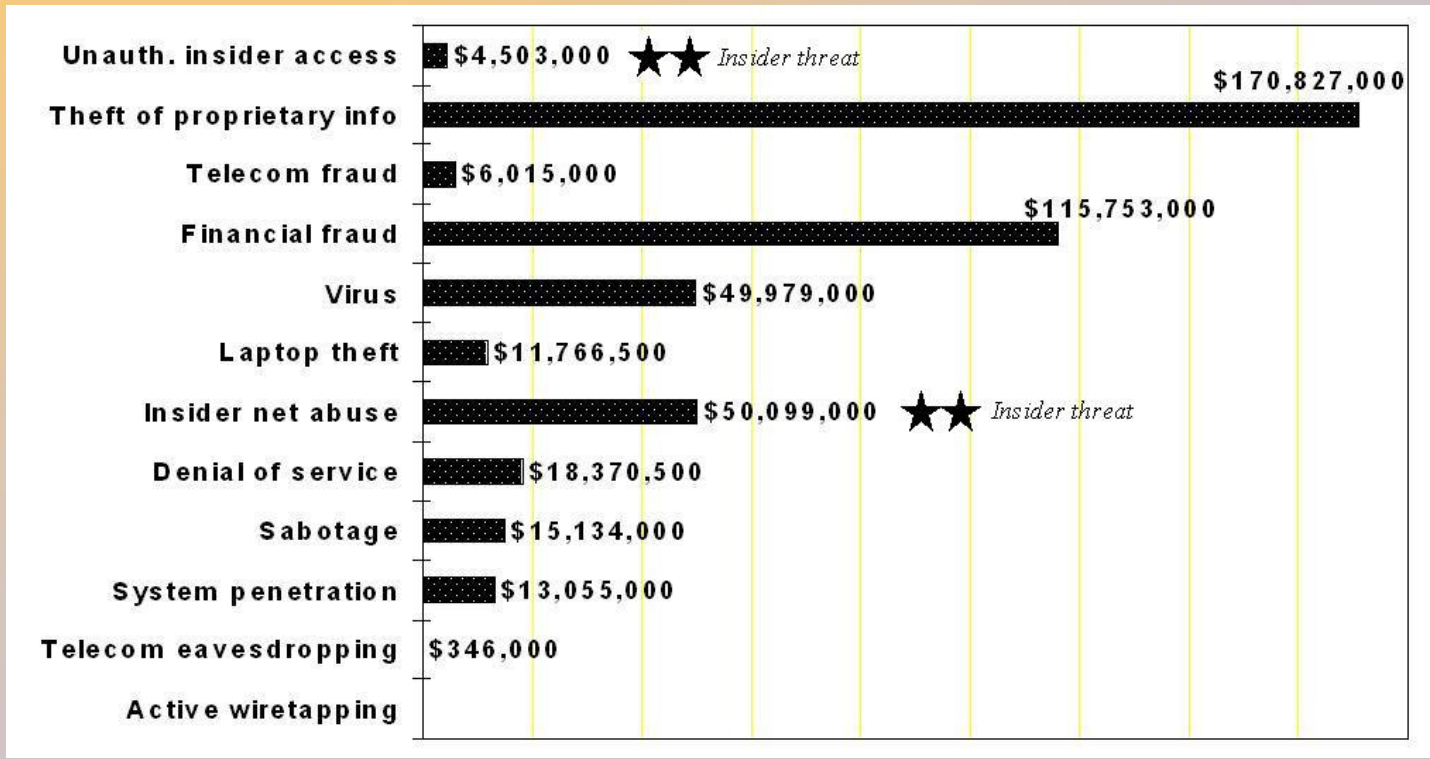


- Single point of protection – at the border
- Good enough for extremely small networks

The Evolving Security Model

Stage 1: Single Firewall Architecture - Drawbacks

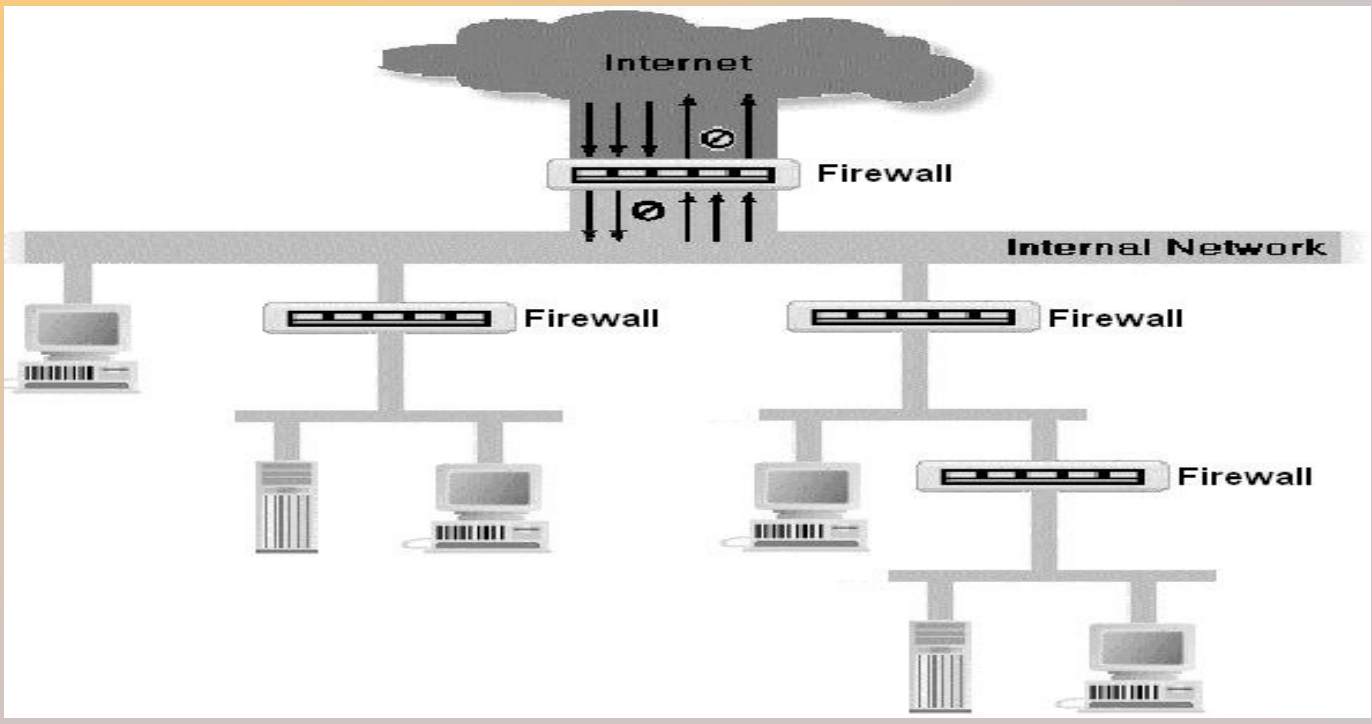
- Insider Threats



- Bandwidth Bottleneck
- Low Trust Level

The Evolving Security Model

Stage 2: Distributed Firewall Architecture

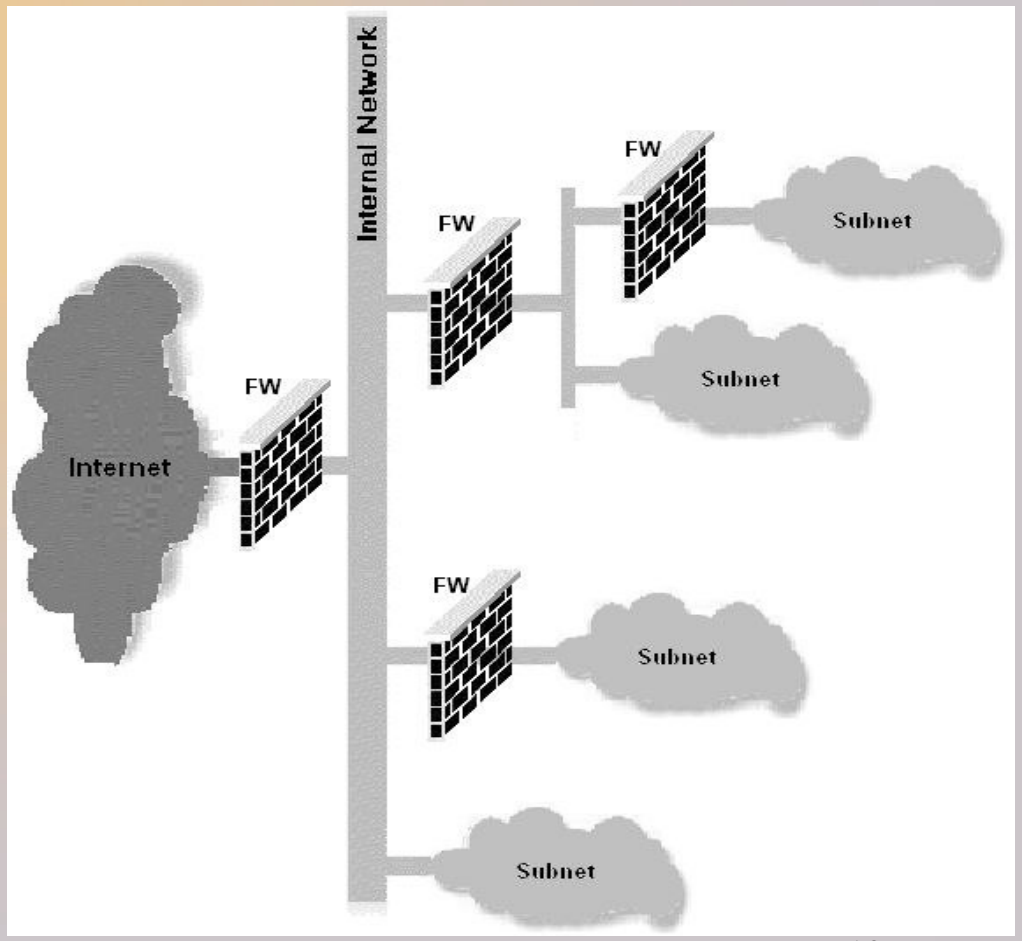


- Multiple points of protection
- Good for every network – small/large

The Distributed Firewall Architecture

"It is easier to secure a studio apartment than a mansion"

- Defense in Depth
- Numerous Choke Points
- Diversity of Defense
- Maintaining Simplicity
- Scalability
- High Performance



The Distributed Firewall Architecture

Major Issues

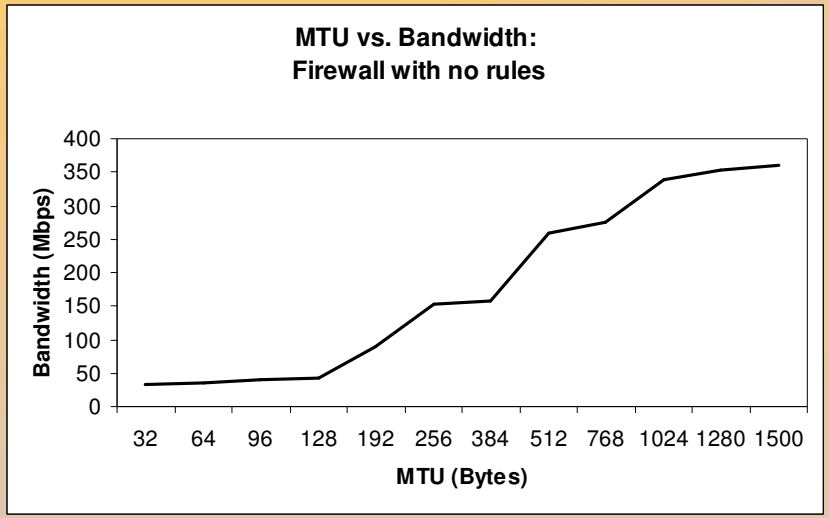
- Firewall Location: The network edge
  - Single host vs. Group of hosts
- Firewall Deployment:
  - Network topology vs. Security topology
- Firewall type
  - Commercial vs. Non Commercial

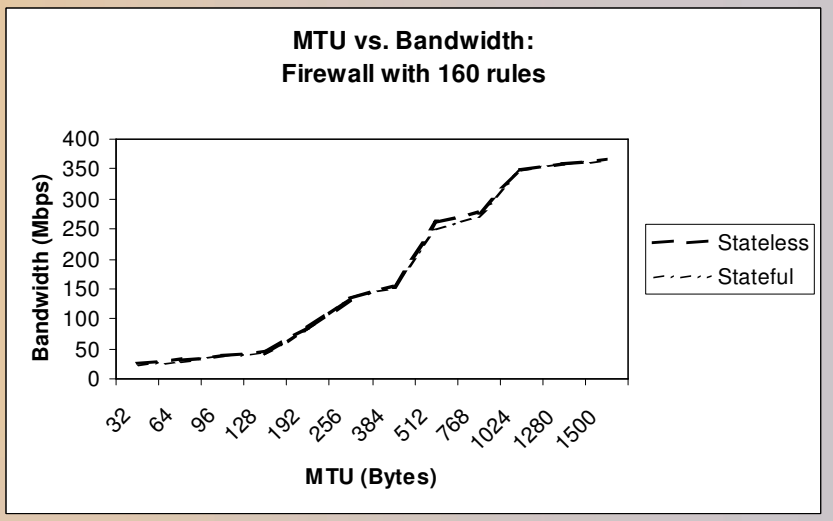
Low Cost Security: Load Balancing

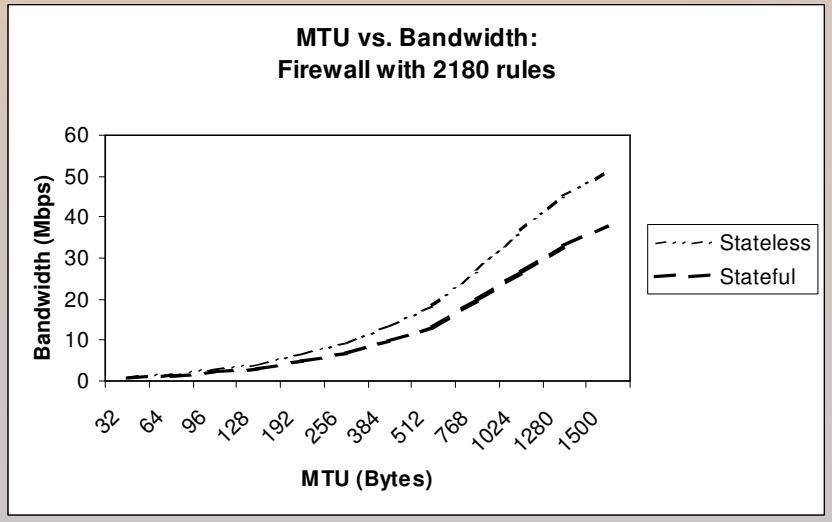
- Firewall is a bandwidth bottleneck
- Solution:
  - Better processor: not scalable
  - Parallel processing: the real solution
- Load balancing for Non Commercial firewalls
  - Low Cost
  - High Performance

Load Balancing for Firewalls

- Performance of a single firewall

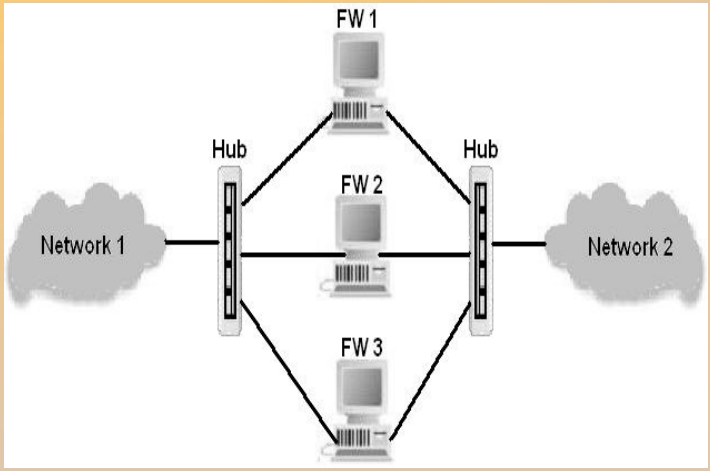






Load Balancing for Firewalls

- Case – 1: Firewall *selects* the packets to be processed

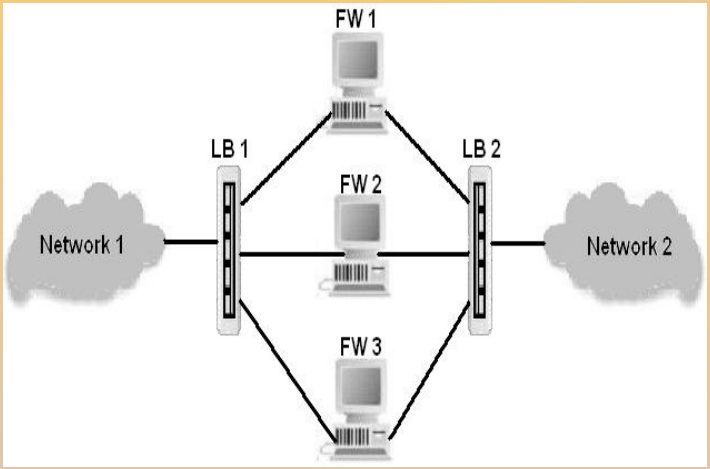


Processors	Speedup
1	1
2	1.82
3	2.3989
4	2.9557

- Drawbacks:
  - Firewalls do more than what they are supposed to do
  - Half duplex mode of the hubs
  - High number of collisions

Load Balancing for Firewalls

- Case – 2: Firewall *gets* the packets to be processed



Test	Speedup
One connection	1.40217
Two parallel connections: one in each direction	2.28814
Two parallel connections: both in same direction	2.45722

Number of processors = 2, Route based load balancing

- Advantages:
  - Firewalls do what they are supposed to do
  - Overcomes the half duplex limitations
  - Number of collisions not as high

Distributed Firewall Policy Management

- Who creates/manages the policies?
  - A central policy management committee
    - cannot ASK
    - cannot keep everyone happy
  - Individual network administrators
    - can ASK
    - no coordination
- How are the policies managed?
  - A centralized policy management system
    - Synchronization of policies
    - Ease of maintenance

"Directory Enabled Policy Management System"

Directory Enabled Network (DEN) Initiative

- What is a Directory?
  - Central storage for information about people, groups, and resources
  - Access by multiple processes, for multiple purposes
  - Operational lynchpin of almost all middleware services
- The DEN Initiative
  - Industry-standard specification for constructing and storing information related to a network's users, applications, resources, and data in a central directory.
  - Directory enabled software allows your enterprise to do everything it did before, only ***smarter***.
- LDAP: Lightweight Directory Access Protocol
  - Widely accepted open industry standard for directory access

Directory Enabled Policy Management

- LDAP schema for policy management
  - *Step-1*: Networked device registration
  - *Step-2*: Distributed firewall support

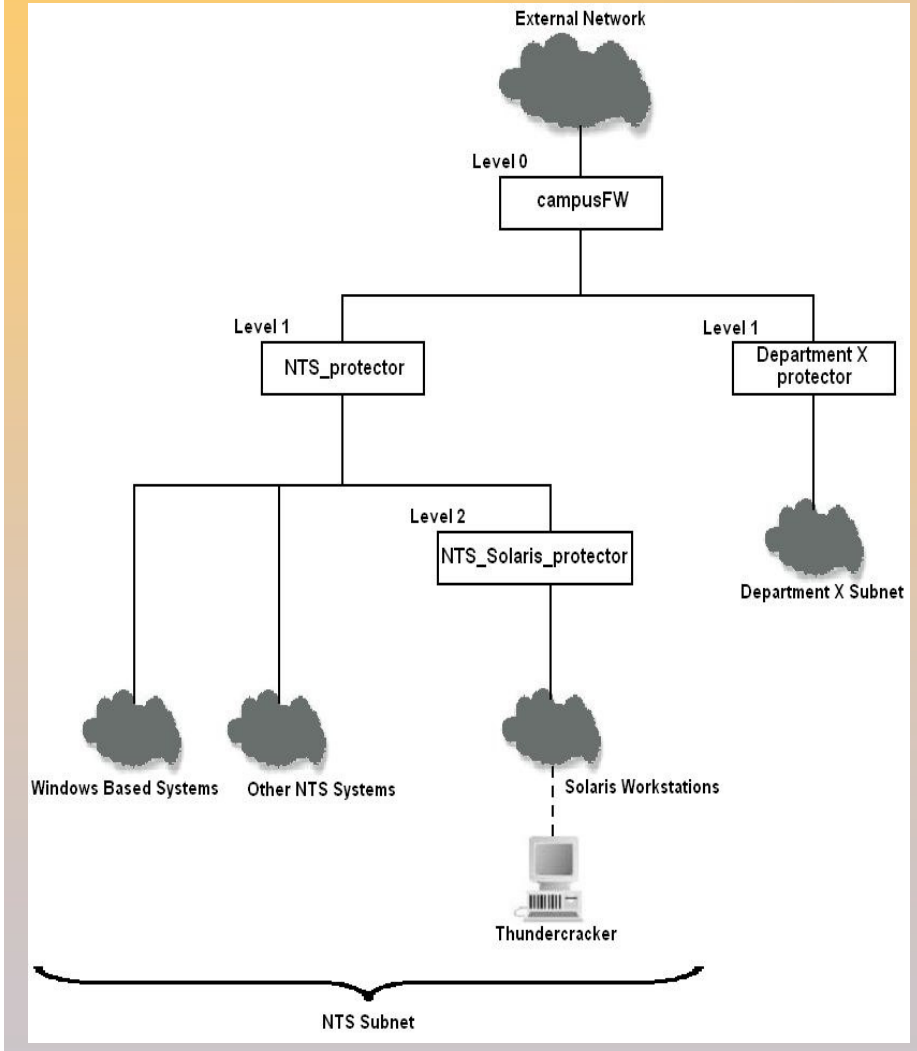
ObjectClasses:

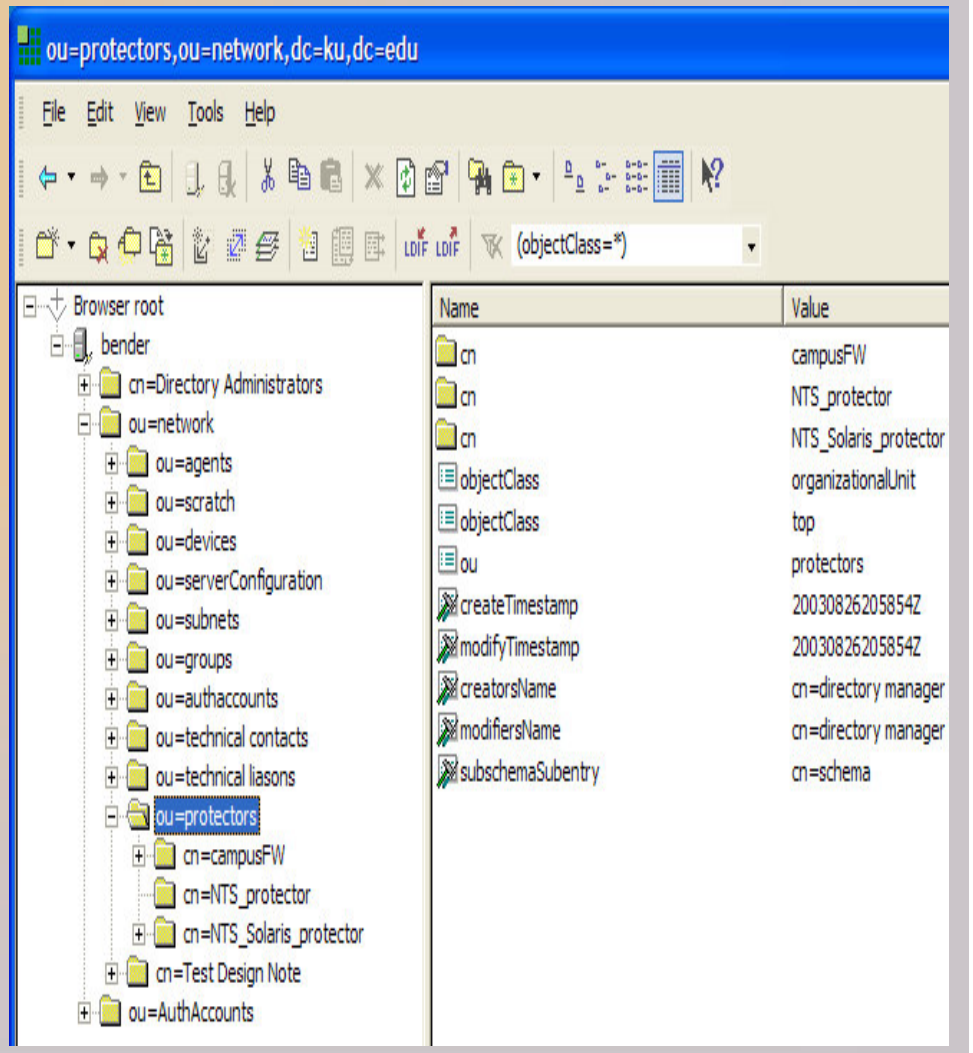
1. IPPacketFilterHost
2. IPPacketFilter

Interface names, MAC and IP addresses
Protected Network's DN
System Administrator
Type of firewall: forwarding / bridging
Filtration: stateless / stateful
Log files

Default policy: allow / deny
Protected internal IPs
Internal TCP/UDP services allowed/denied
External TCP/UDP services allowed/denied
ICMP types allowed/denied
Trusted internal/external IP addresses
Traffic to be logged

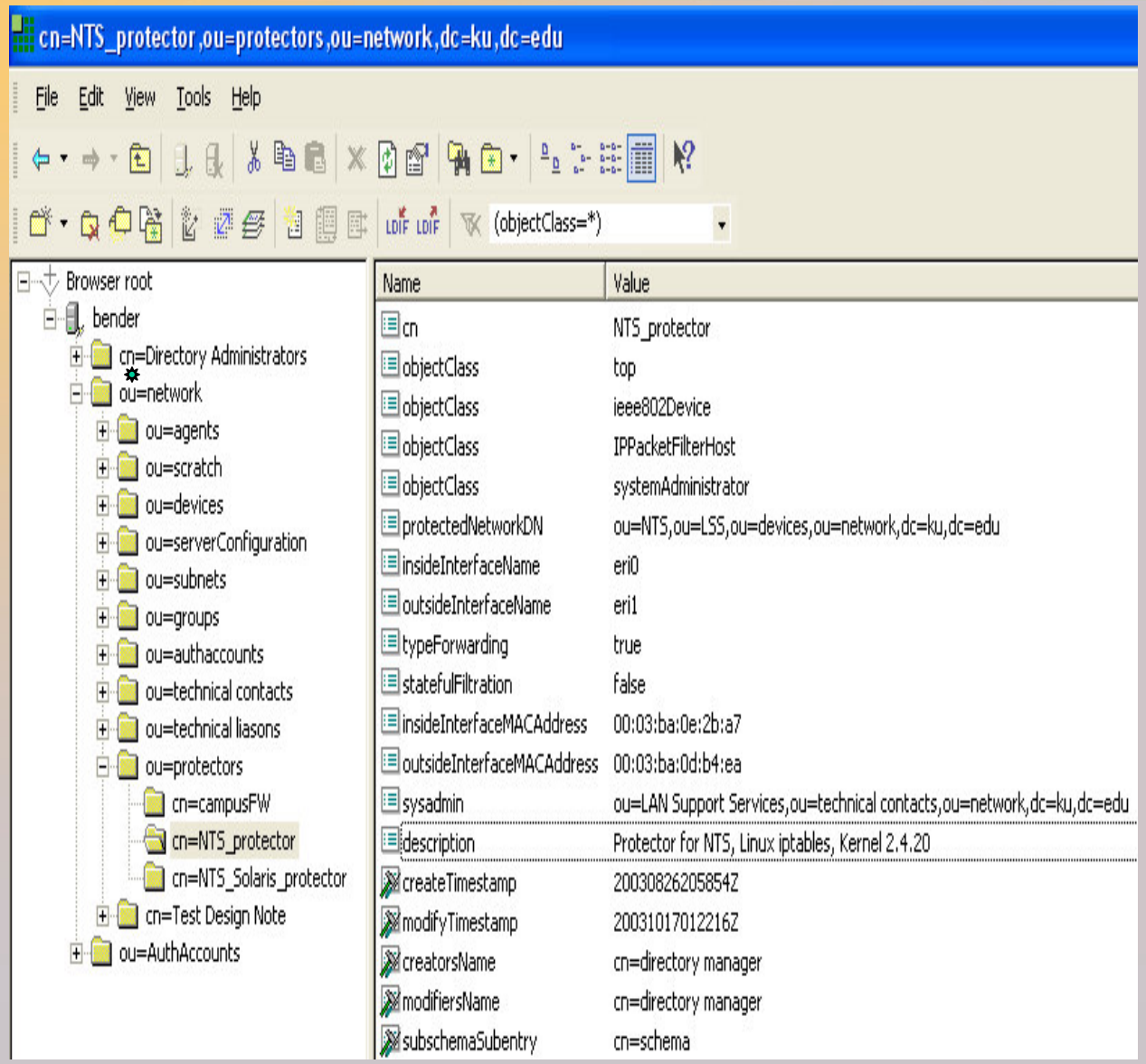
Example


External Network
Level 0
campusFW
Level 1
NTS_protector
Level 1
Department X protector
Level 2
NTS_Solaris_protector
Department X Subnet
Windows Based Systems
Other NTS Systems
Solaris Workstations
Thundercracker
NTS Subnet


ou=protectors,ou=network,dc=ku,dc=edu
File Edit View Tools Help
(objectClass=*)
Browser root
bender
cn=Directory Administrators
ou=network
ou=agents
ou=scratch
ou=devices
ou=serverConfiguration
ou=subnets
ou=groups
ou=authaccounts
ou=technical contacts
ou=technical liasons
ou=protectors
cn=campusFW
cn=NTS_protector
cn=NTS_Solaris_protector
cn=Test Design Note
ou=AuthAccounts
Name
Value
cn campusFW
cn NTS_protector
cn NTS_Solaris_protector
objectClass organizationalUnit
objectClass top
ou protectors
createTimestamp 20030826205854Z
modifyTimestamp 20030826205854Z
creatorsName cn=directory manager
modifiersName cn=directory manager
subschemaSubentry cn=schema

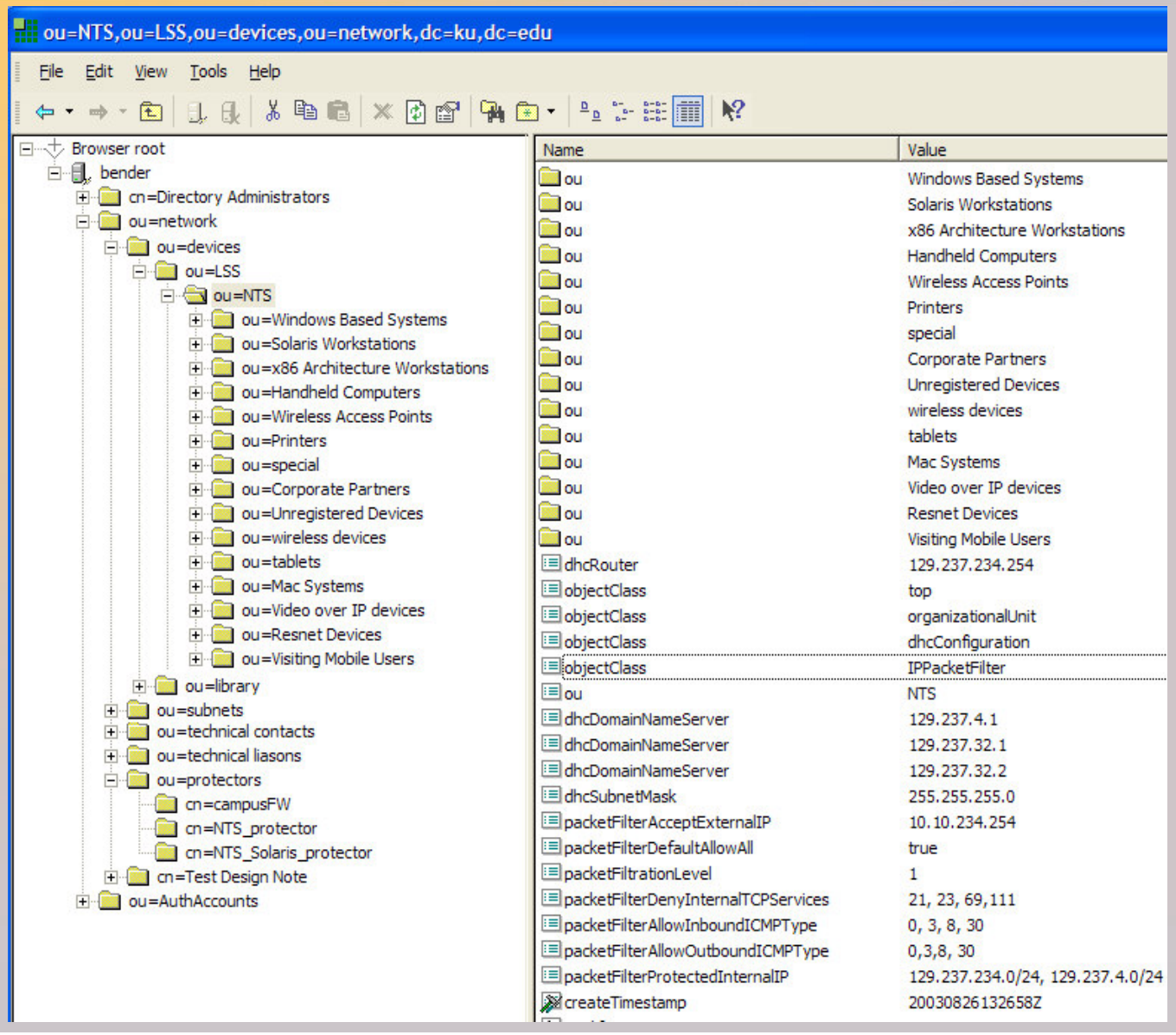
Example (contd.)

IPPacketFilterHost



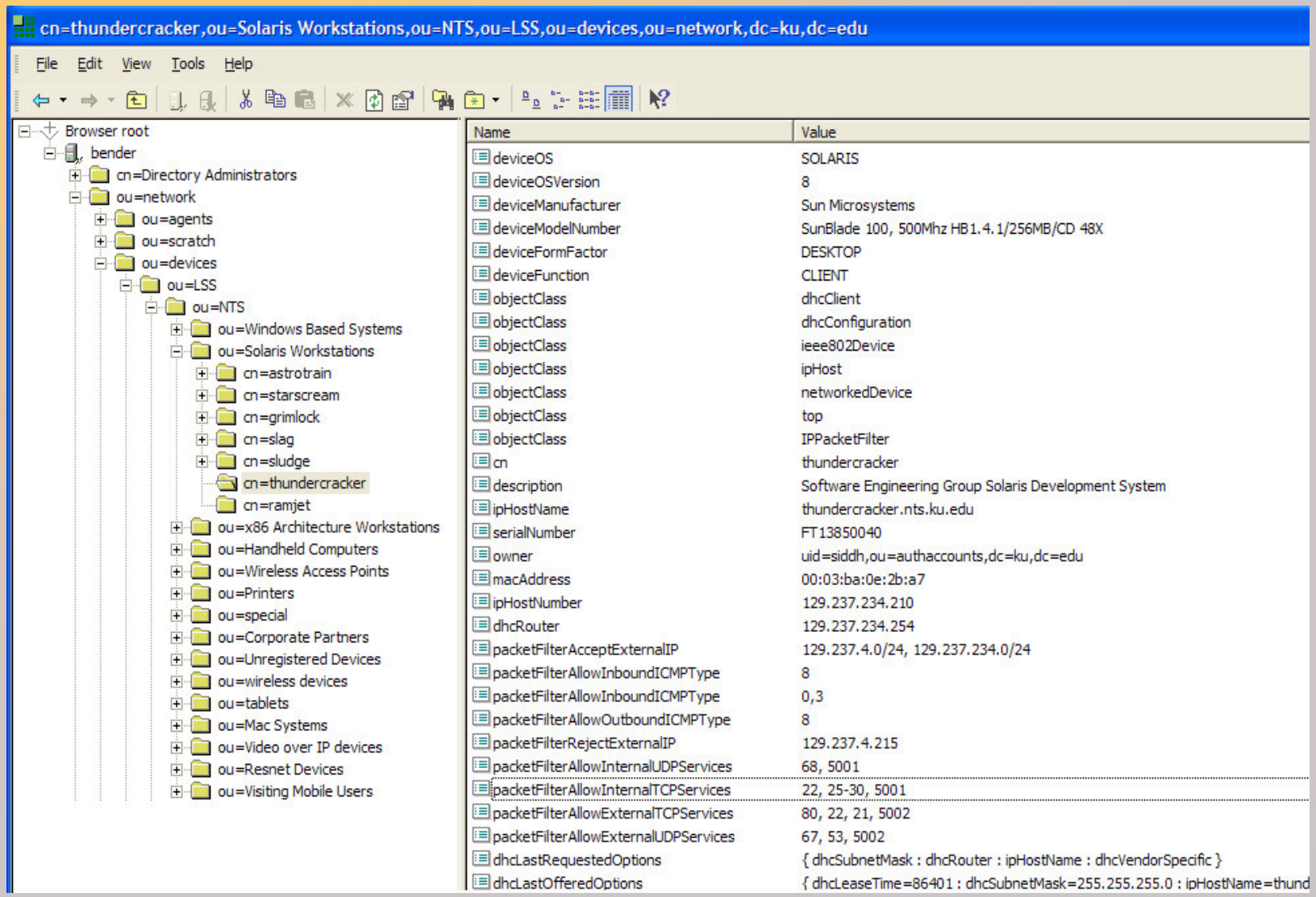
Example (contd.)

IPPacketFilter



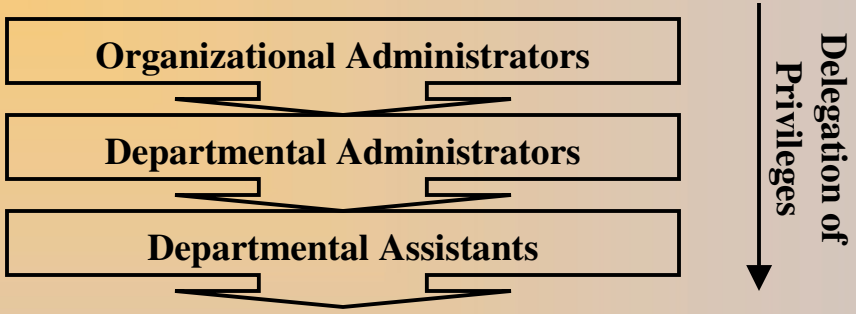
Example (contd.)

- Host-centric policy specification



Directory Enabled Policy Management (contd.)

- The Directory and the System/Network Administrators



  - Authentication/Authorization features
  - Access Control Lists
  - LDAP administration tools

Rule Generator

Policies in Directory ⟹ Firewall specific rules

Entries & Attributes

iptables, Drawbridge, OpenBSD pf, …

Cisco PIX, Checkpoint, …

- Two Phases
  - Firewall independent directory support system
    - Connection establishment
    - Search, retrieval and modification operations
    - Entry list for which rules are to be created
  - Firewall dependent rule creator
    - Rules in the firewall's language

DirectoryServer
DirectoryServerConfigFile
DirectoryServerInfo
PacketFilterDirectorySupport
PacketFilterProtectedNodes

CreateIptablesRules
CreatePIXRules

Directory Enabled Policy Management (contd.)

Advantages

- Ease of management.
- Delegated management.
- Flexible hierarchical model
- A high granularity of the security system is possible.
- Ability to achieve host-level security.
- Ease of synchronization and coordination.
- Highly scalable: hosts or group of hosts can be added or removed without much effort.
- Common language for different types of firewalls, both commercial and non-commercial.
- Flexible LDAP administration client tools.
- High speed search and security audit capability.
- Encrypted communication on the network with LDAPS.
- Identification, Authentication, and Authorization take place before changes can be made.
- Encrypted user credentials are stored in the directory and on the underlying file system.
- Protocol oriented communication via LDAP with external systems, i.e., ModPerl, or Java JNDI, or OpenLDAP APIs.
- Replication agreements with peer directory servers.
- Easy to load-balance, and easy to make backups via LDIF export.

KU and the Distributed Security Architecture

- The University Network
  - Lack of control over users
  - Loose confederation of autonomous entities
  - Academic culture and tradition of open access to information
  - Complex trust relationships between departments at various Universities
  - Excellent platforms for launching attacks
    - high bandwidth Internet
    - sophisticated computing capacity
    - insecure systems in dorms
- The University of Kansas
  - Number of students, faculty and staff: ~35000
  - Number of buildings: ~100
  - Number of hosts: ~20000
  - Internet 1 link: 70Mbps rate limited on 100Mbps connection

KU and the Distributed Security Architecture

- Firewalls that can be used: Cost effective solution
  - Factors:
    - Number of rules
    - Size of packets
    - Type of filtration: stateless or stateful
    - Number of flows (connections) passing through the firewall

KU

At the Border

- Number of rules: ~100
- Packet size: ~200 to 500 bytes
- Type of filtration: Stateless
- Number of flows: not an issue

For a Department

- Number of rules: ~200
- Packet size: ~200 to 500 bytes, on an average
- Type of filtration: Stateless or Stateful
- Number of unique flows: ~100 per minute

KU and the Distributed Security Architecture

- Proposed Setup

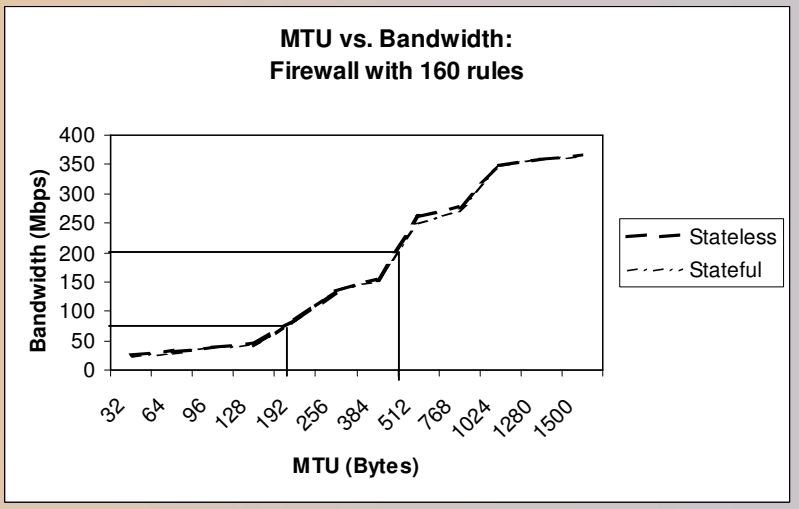
  Linux *iptables*, 2.4GHz, 512MB RAM, 512KB L2 cache, Intel GigE cards
  Route based load balancing (iproute2)

  Number of rules: **160**

  MTU of 200 bytes: **85.5Mbits/s**

  Default number of flows: **32760**

  Speedup with 2 firewalls: **2.45722**



KU

At the Border	For a Department
➢ Non-commercial firewalls	➢ Non-commercial firewalls
➢ Stateless filtration	➢ Stateless / Stateful filtration.
➢ Load balancing: at least 2 firewalls	➢ Load balancing: depends on department
➢ Load balancer need not worry about state	➢ Load balancer might have to keep track of state

- Policy Management – Already discussed in the examples

KU and the Distributed Security Architecture

- Example - The recent W32.Nachi worm attack
  - ↘ Scans the local class-b subnet (port 135), sends ICMP ping to potential victim
  - ↘ Connects to the infected machine on TCP port, range 666-765
  - ↘ Victim instructed to download the worm via TFTP

Problem faced with current architecture

- Few infected hosts in the internal network trying to infect other hosts
- Network flooded with ICMP ping packets
- Routers overloaded with excessively high number of flows

Steps taken

- Packet filter in the border router configured to block packets destined to TCP or UDP port 135
- Infected systems were identified and repaired

Did it really solve the problem?

- External ←→ Internal infection was stopped
- Takes time to isolate and repair infected systems
- In this time:
  - Each system generated 100,000 flows per minute, still infecting other systems
  - Backbone still flooded
  - Routers still overloaded

} *Management "nightmare"*

KU and the Distributed Security Architecture

- Example - The recent W32.Nachi worm attack (contd.)
  - How would the Directory Enabled Architecture help?
    - Quick response to security incidents
    - Every Firewall can be immediately configured
      - Prevents worm from spreading to areas outside the firewall
      - Traffic generated by the infected system remains within the subnet of that department
      - Removes the "extra" time given to an infected system for infecting other hosts in the campus
    - Firewall for an infected system can be immediately identified by looking up the directory
    - The other usual advantages of the distributed architecture
  - Steps involved
    - Enter the policy in the directory, for every firewall
    - Generate the rules for the firewalls
    - Inject the rules into the firewalls
    - Identify and repair the infected systems

} *Simple management*

Conclusion

- Distributed Security Architecture is the MOST SECURE
- It can be a LOW COST architecture
- The Directory Enabled Framework
  - helps efficiently maintain a distributed security architecture

    AND

  - retain the ability of the departmental administrators to make fine-grained decisions

Future Work

- More features for firewall maintenance
  - Timestamps
  - Rule distribution
- Rule generators for different types of firewalls
- Rule Minimization
- Managing firewall auditing
  - Logging facilities
  - Packet counters (netflow)
  - Usage based metering/ charging
- Integrating IDS into the firewall architecture

Acknowledgements

- First and foremost: NTS and everyone at NTS
- George Willard – An excellent supervisor
- Prof. Joseph Evans, Prof. Gary Minden, Prof. Victor Frost
- Brett Becker – the video setup, ITTC network information
- All my friends at KU