



Intelligent
Systems
Lab

Administration of Operating Systems

Firewall Chapter 25

December 05, 2011

Firewall



Intelligent
Systems
Lab

- Single access point
 - castle moat and drawbridge gatehouse
- All traffic to/from an internal network passes through the firewall
 - data in the internal network is “good”
 - data outside is “bad”
- Single point of vulnerability — less bugs
- Packet filtering
 - only connections initiated from inside
 - blacklisted IPs, forbidden services/ports
- Application gateway
 - higher-level functions

Firewall Types



- Hardware firewall
 - specially programmed router
 - sits between a host and the Internet
 - or between LAN and the Internet
 - examines packets before forwarding them
 - drops those not fitting some policy
- Software firewall
 - a process on the machine being protected
 - running as part of the kernel
 - requires low-level access to the traffic
 - examines contents of network traffic
 - *and* processes initiating it

Tradeoffs



- Hardware firewall
 - specialised device
 - or a dedicated computer
 - centralised administration
 - does not “steal” CPU from users
 - potentially highly efficient
- Software firewall
 - usually more flexible
 - can create per-process rules
 - potentially easier to disable
 - firewall process can be killed
 - computer configuration can be changed

Firewall Policies



- Block traffic from a given IP
 - allow traffic from a given IP
- Block traffic to a given port
 - some applications can use dynamic port range
 - many use port 80
- Block all connections initiated from the Internet
 - incoming TCP packets with SYN flag set
- Block some protocol
 - block all UDP traffic
- Forbid some process from accessing a network
 - protection against trojans and viruses

Firewall Operation



- Packet analysis can only take you so far
 - for example, many p2p applications have configurable port numbers they will use
 - state-less vs state-full analysis
- It is also possible to analyse traffic contents
 - at least in principle
 - distinguishing applications based on that
 - very complex task
 - and specialised recognition routines are needed for each application to be blocked
 - can have hard time dealing with encryption
 - rarely implemented in stand-alone firewalls
 - but sometimes the only way to achieve a goal

NAT



- Network Address Translation
 - designed to “create” more IP addresses
- Ensures that all traffic passes through the firewall
 - computers do not have global IP address
 - they cannot be accessed from network
 - or access the network
 - in any other way than via NAT box
- A way simple and quite effective way to keep a host inaccessible from the Internet
 - incoming connections are refused by default
 - outgoing connections are allowed
 - listening ports need to be explicitly mapped

Virtual Private Network



- Two trusted sites connected by insecure network
 - an extension of the firewall idea
 - allows a user to treat several remote sites as belonging to a single LAN
- VPN — implemented in OS, router or firewall
 - transparently encrypt all network traffic as it is leaving the trusted zone
 - transparently decrypt all network traffic as it is entering the trusted zone
- Applications and end-users do not need to be aware of its existence
- Can be implemented on routers
 - or on individual user machines

Proxies



Intelligent
Systems
Lab

- An application which to client appears to be a server, and to server appears to be a client
- Needs to understand application protocol
 - but can, for example, filter web traffic based on URL addresses
 - strip attachments from incoming email
 - add advertisements or disclaimers to all outgoing email
 - allow operation of protocols like FTP behind a strict firewall
 - FTP uses varying port numbers
 - proxy can decide which ones need to be opened at any given time

iptables



- Utility for creating packet filtering rules
 - for Linux kernel module `netfilter`
- Flexible and extensible framework
- Manages *chains* of rules
 - PREROUTING
 - INPUT
 - FORWARD
 - OUTPUT
 - POSTROUTING
 - user-defined chains
- Allows for stateful packet filtering
 - decision is needed when *opening* connection

netfilter



- Intercepting and manipulating network packets
 - invoked by the packet reception/send routines
- Includes *connection tracking* module
 - keeping track of logical network connections
 - heuristics for common protocols
 - for example UDP and FTP
- Different connection states
 - NEW
 - ESTABLISHED
 - RELATED
 - INVALID
 - UNTRACKED

firestarter



Intelligent
Systems
Lab

- Graphical tool for configuring iptables
 - control inbound and outbound connections
 - establish NAT and port forwarding
 - managed DHCP server
- Easy to use
 - default setup works in most cases
- Limited flexibility
 - good for typical setups
- Allows real-time view of network activities
 - highlighting potential intrusion attempts
- Discontinued project
 - last stable release January 29, 2005



- Uncomplicated Firewall
- Easy to use
 - command-line tool
 - installed by default on Ubuntu systems
- Covers the whole functionality of iptables
 - but only some subsets are made easy
- `sudo ufw enable`
- `sudo ufw allow 22`
- `sudo ufw allow proto tcp from 192.168.0.2 to any port 22`
- `sudo ufw status`
- `sudo ufw disable`

gufw



- GUI for Uncomplicated Firewall
- `sudo apt-get install gufw`
- Three interfaces for creating rules
 - preconfigured
 - contains a number of typical setups
 - for various applications
 - simple
 - specify direction, protocol and port
 - advanced
 - specify address and port ranges
- Does not support setting up NAT

shorewall

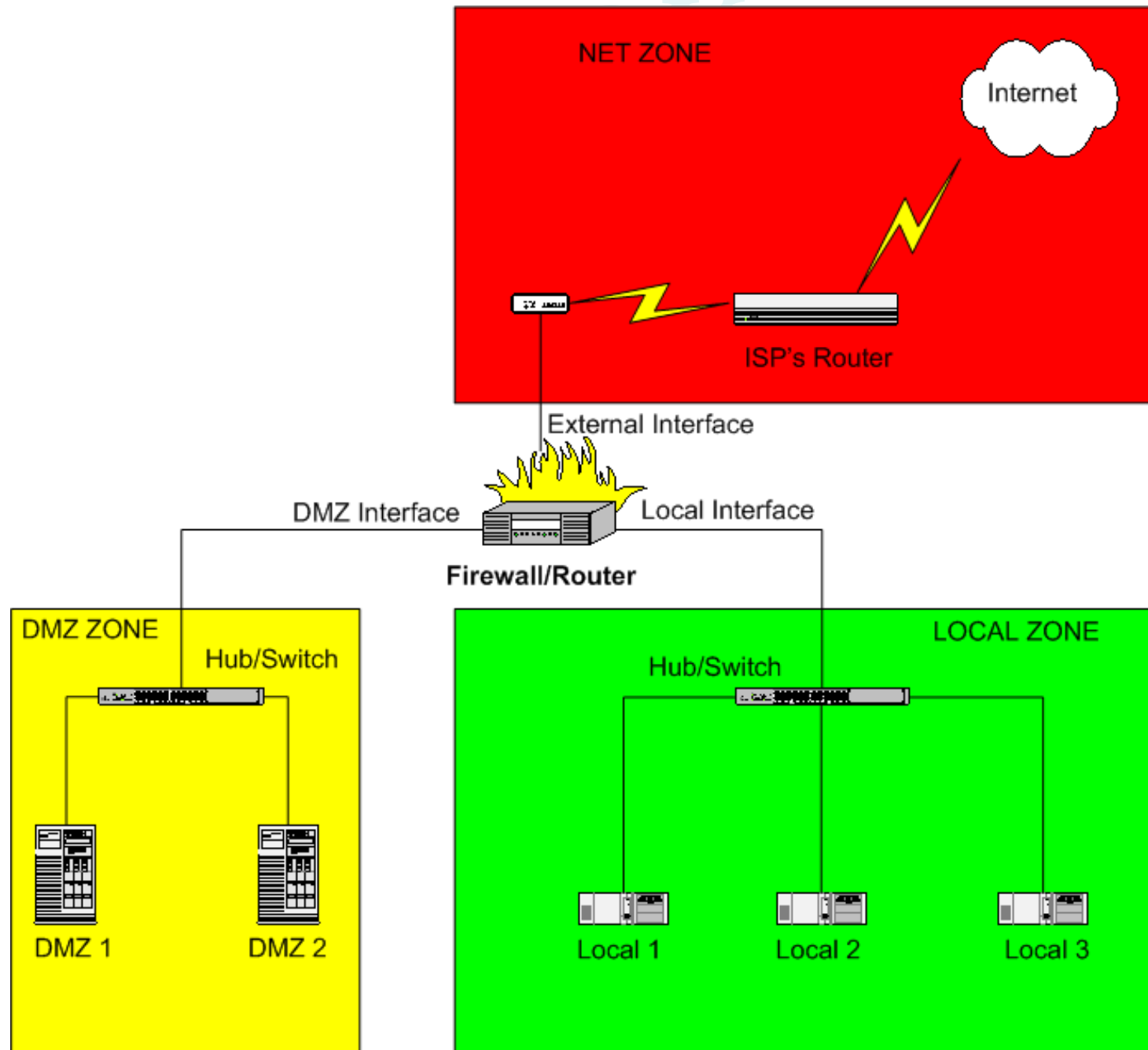


- More powerful configuration tool
- Allows for definitions of *zones*
 - individual computers
 - address ranges
 - subnets connected to different interfaces
- Each zone can have it's own rules
 - DMZ — DeMilitarized Zone
 - server zone / workstation zone
 - NAT zone
 - Internet zone
- Configured using multiple files
 - supported by Webmin

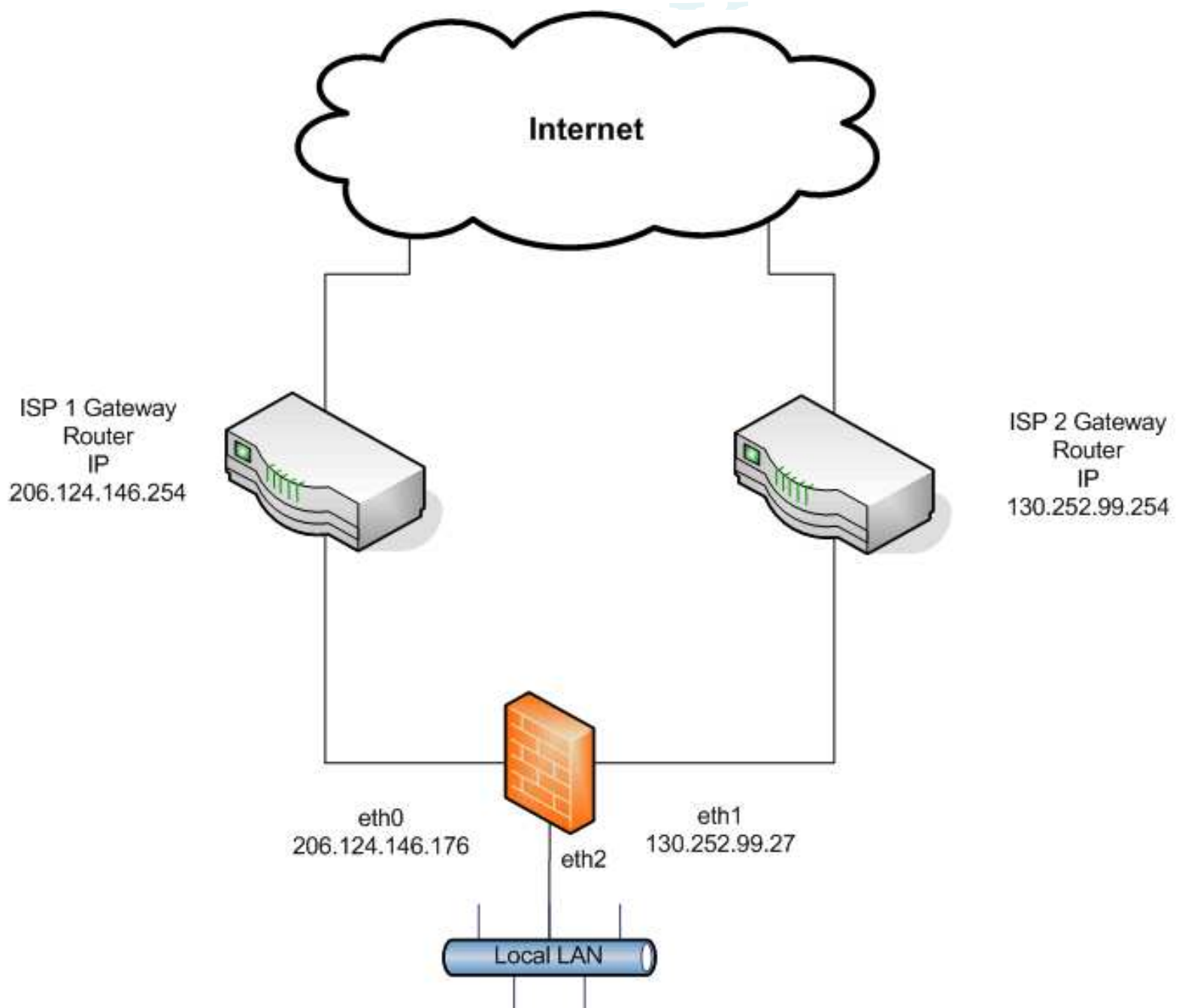
Example



Intelligent
Systems
Lab



Example





Intelligent
Systems
Lab

Questions?

