Firewalls | NAT

INGI2347: COMPUTER SYSTEM SECURITY (Spring 2014)

Marco Canini

UCL Université catholique de Louvain

Lecture slides adapted from UCL INGI2347 by Gildas Avoine and Upenn CSE331 by Steve Zdancewic

Announcements

- First challenge deadline 24 Feb 23:59h
- From 18 Feb, Tuesdays lectures start at 8:30h
- Inform the instructor if you wish to take the exam in French by end of Feb

Firewalls and NAT

Problem:

- Protecting or isolating one part of the network from other parts
  - Prevent propagation of an attack while allowing legitimate traffic
- Need to filter or otherwise limit network traffic
  - How to configure this information?
- Questions:
  - What information do you use to filter?
  - Where do you do the filtering?

Types of Firewalls

- Software
  - Standard firewall software:

  Iptables, IPFILTER, IPFW, Ipcop

- Hardware
  - Specialized middlebox (also runs software as firmware):

  Cisco PIX, Juniper, WatchGuard, SonicWall, Barracuda

Software Firewalls

- Software firewalls inherit all vulnerabilities of the OS on which they run
- Software firewall architectures are well known, it's easier to exploit their vulnerabilities
  - Example: buffer overflows
- Software firewalls often have good performance
  - They benefit from rapid advances in PC hardware

Plan for today

Lecture 3

- Basic principles

- NAT
- Firewall features
- Firewall architectures
- Filtering rules

Basic Principles

Principles: The Seven Principles

- Least privileges
- Defense in depth
- Choke point
- Weakest link
- Deny by default
- User participation
- Simplicity

Principle: Least Privilege

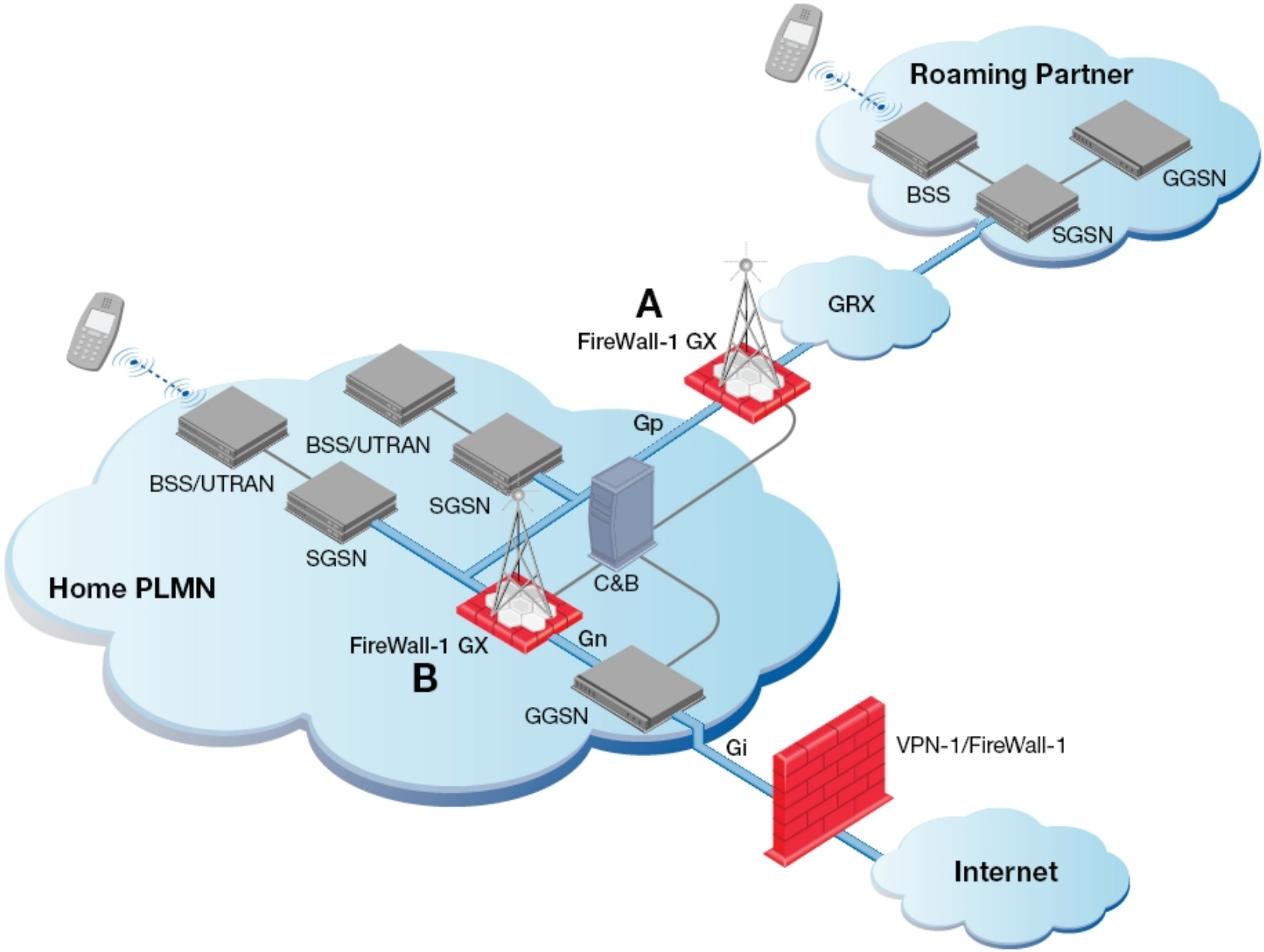
- Every part of the system must only have the minimal rights necessary to carry out its job
- Examples:
  - Regular users must not be administrators
  - Administrators must also use regular user accounts
  - A Web server runs under a non-privileged account
    - Unix: nobody
    - Windows: IUSR_machine_name
- Military's slogan: "Need to know"

Principle: Defense in Depth

- Layers of security are harder to break than a single defense
- Examples:
  - Anti-viruses on mail servers and on desktops
  - Patch machines even if they are protected by a firewall
  - Even if FTP connections are blocked by the firewall, workstations should not run FTP servers

Principle: Choke Point

- It is easier to control security if all data has to go through one given point
- Users should not be allowed to bypass the network policy
  - Example: use alternate Internet connection
- Interconnections with other companies must go through the firewall


Roaming Partner
BSS
GGSN
SGSN
GRX
A
FireWall-1 GX
Gp
BSS/UTRAN
BSS/UTRAN
SGSN
SGSN
C&B
Home PLMN
FireWall-1 GX
B
Gn
GGSN
Gi
VPN-1/FireWall-1
Internet

Principle: Weakest Link

- Attackers go after the easiest part of the system to attack
  - So improving that part will improve security the most
- Example:
  - Useless to install expensive anti-virus software for HTTP traffic if you do not also install one for SMTP traffic
- How do you identify it?
- Weakest link may not be a software problem
  - Social engineering
  - Physical security

Principle: Deny by Default

- It is better to prohibit all that is not explicitly authorized than to authorize all that is not explicitly prohibited
- We can never know in advance all the threats to which we will be exposed
- If we make an error, it is better to prohibit something useful than to allow an attack!

Principle: User Participation

- A protection system is efficient only if **all users support it**
- The goal of a firewall is to authorize all that is useful and at the same time avoid dangers
- A system that is too restrictive pushes users to be creative
  - Example: saving confidential email on personal's Gmail to read remotely
- We must understand the user's needs and make sure that reasons for restrictions are well understood by them

Principle: Simplicity

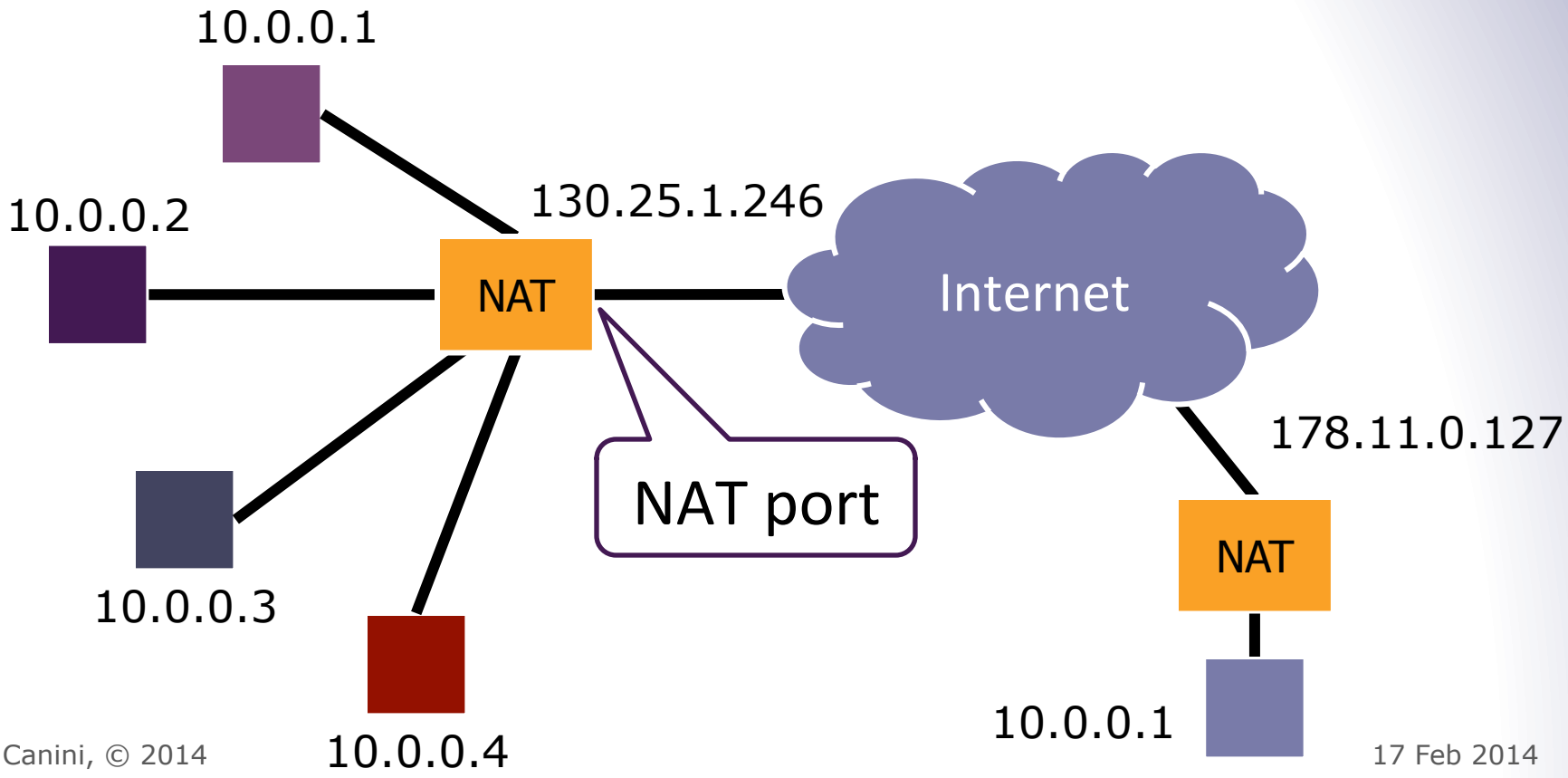
- Most security problems originate from human error
- Complexity leads to bugs and bugs lead to vulnerabilities
- Failsafe defaults
  - The default configuration should be secure
- In a simple system:
  - The risk of error is smaller
  - It is easier to verify its correct functioning
  - Especially in evolving networks and with several administrators

Network Address Translation

NAT

Network Address Translation (NAT)

- Idea: Break the invariant that IP addresses are globally unique



Dynamic NAT

- Basic principle: Maintain a table of the form:

<client IP> <client port> ⇄ <NAT ID>

- Outgoing packets (on non-NAT port):
  - Lookup client (source) IP address, client port in the mapping table
  - If not found, allocate a new unique NAT ID and replace client port with chosen NAT ID (same size as port = 2^{16})
  - If found, replace client port with previously allocated NAT ID
  - Replace client address with NAT address

Dynamic NAT

- Incoming packets (on NAT port)
  - Look up destination port number as NAT ID in port mapping table
  - If found, replace destination address and port with client entries from the mapping table
  - If not found, the packet is not for us and should be rejected
- Unused table entries expire periodically
  - Example: after 2-3 minutes
- Dynamic NAT doesn't allow establishing incoming connections
  - Good protection by default

Mapping Table Example

Protocol	Local IP	Local port	NAT IP	NAT ID	Peer IP	Peer port
TCP	192.168.0.1	1912	81.242.186.64	1912	192.178.100.4	80
TCP	192.168.0.1	1913	81.242.186.64	23745	192.178.100.4	80
TCP	192.168.0.2	1912	81.242.186.64	55468	212.27.63.3	80
UDP	192.168.0.3	18551	81.242.186.64	1912	83.170.84.81	26000

Mapping Table Example

- NAT ID must be unique

Protocol	Local IP	Local port	NAT IP	NAT ID	Peer IP	Peer port
TCP	192.168.0.1	1912	81.242.186.64	1912	192.178.100.4	80
TCP	192.168.0.1	1913	81.242.186.64	23745	192.178.100.4	80
TCP	192.168.0.2	1912	81.242.186.64	55468	212.27.63.3	80
UDP	192.168.0.3	18551	81.242.186.64	1912	83.170.84.81	26000

Mapping Table Example

- Mapping can hide auto-increasing port numbers

Protocol	Local IP	Local port	NAT IP	NAT ID	Peer IP	Peer port
TCP	192.168.0.1	1912	81.242.186.64	1912	192.178.100.4	80
TCP	192.168.0.1	1913	81.242.186.64	23745	192.178.100.4	80
TCP	192.168.0.2	1912	81.242.186.64	55468	212.27.63.3	80
UDP	192.168.0.3	18551	81.242.186.64	1912	83.170.84.81	26000

Mapping Table Example

- Protocol info. further demultiplexes mapping entries

Protocol	Local IP	Local port	NAT IP	NAT ID	Peer IP	Peer port
TCP	192.168.0.1	1912	81.242.186.64	1912	192.178.100.4	80
TCP	192.168.0.1	1913	81.242.186.64	23745	192.178.100.4	80
TCP	192.168.0.2	1912	81.242.186.64	55468	212.27.63.3	80
UDP	192.168.0.3	18551	81.242.186.64	1912	83.170.84.81	26000

Static NAT

- To allow incoming connections, we have to define certain static entries in the mapping table
- Typically we create one entry per protocol
  - Example: SSH (22), HTTP (80), SMTP (25), …
- Example:

Protocol	Local IP	Local port	NAT port	Peer IP & port
TCP	192.168.0.1	22	22	*
TCP	192.168.0.1	8080	80	*
UDP	192.168.0.3	26000	26000	*

Benefits of NAT

- Dynamic NAT only allows outbound connections established from internal network
  - External hosts can only contact internal hosts that appear in the mapping table, which are only added once they establish a connection
- Hides the internal network's structure
- Can simplify network administration
  - Divide network into small chunks
- Reuse IP address space
  - Original motivation behind NAT

IETF-allocated private addresses:

- 10.0.0.0 - 10.255.255.255
- 172.16.0.0 - 172.31.255.255
- 192.168.0.0 - 192.168.255.255

Drawbacks of NAT

- Rewriting IP addresses (and ports) isn't so easy:
  - Must validate/recalculate checksums
  - Certain protocols such as IPSec do not support packet modifications
  - Has to be aware of protocols that exchange IP addresses (e.g., FTP)
    - Must also look for IP addresses beyond packet headers and rewrite them
- Hinder throughput
- Breaks end-to-end principle
  - Prevents host-to-host connection establishment for hosts behind NAT
- Slow the adoption of IPv6?
- Limited filtering of packets

Firewall Features

Firewall Features

- Stateless vs Stateful
- Packet Analysis
- Filtering
- Network Address Translation
- Authentication
- Remote network access
- Encryption
- Logging

Stateless vs Stateful

- Stateless: without memory
  - Does not maintain state associated with observed packets
- Stateful: with memory
  - Maintain state associated with observed packets
  - Reconstructs each connection's state, or even certain protocols

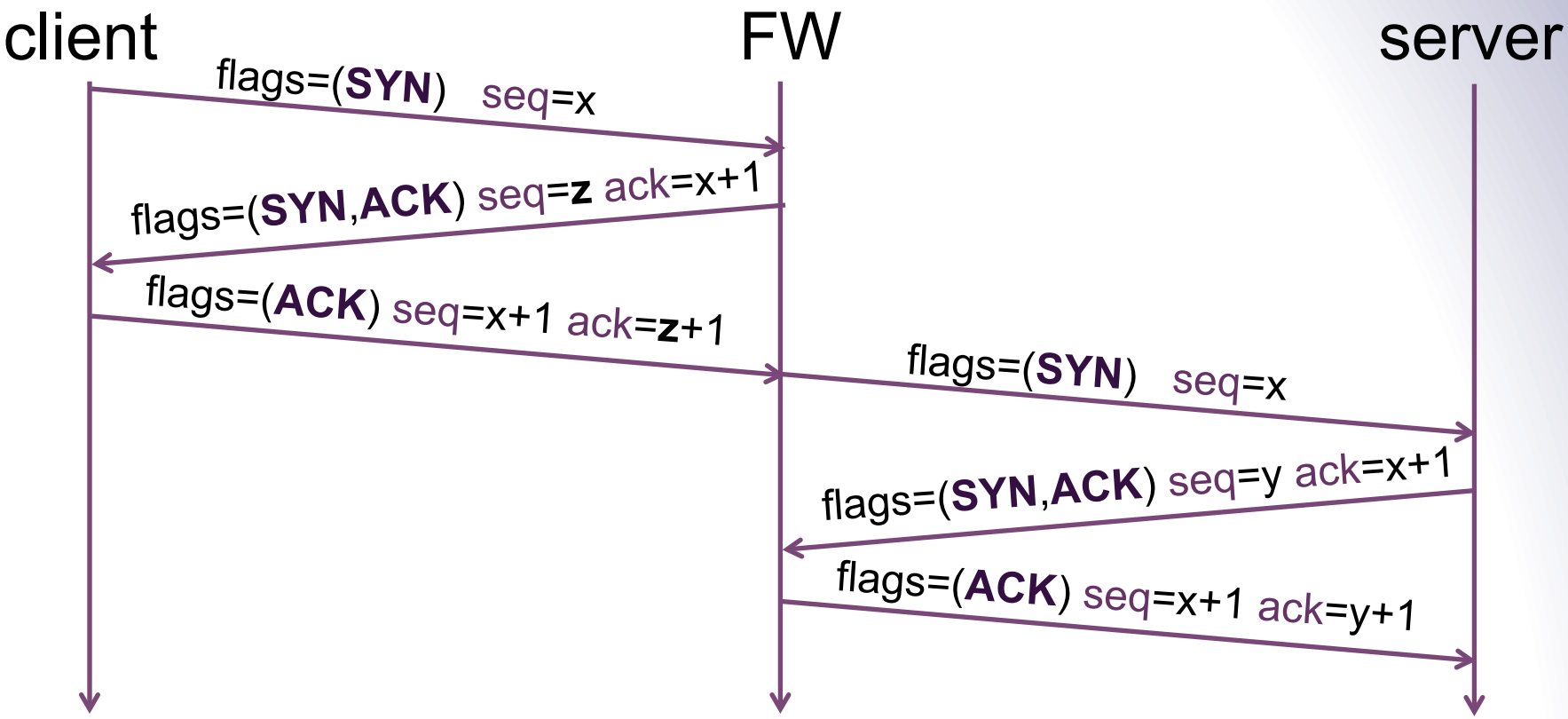
Stateful Firewall: Example TCP

- For each connection it knows what the next packet should look like
  - TCP flags, sequence numbers
- It can eliminate packets that do not fit in
- It can replace sequence numbers
  - Example: to randomize initial sequence numbers
- It can prevent SYN flooding

Protection Against SYN Flooding

- Simple:
  - FW keeps track of all attempts to open a connection
  - If it judges that a connection stays half-open for too long, it sends a RST to the server
- Advanced:
  - FW delays SYN packets and generates a SYN + ACK in place of the server
  - Only when it receives an ACK does it send the original SYN to the server

Protection Against SYN Flooding



- FW must adapt all sequence numbers

Packet Analysis (Deep Packet Inspection)

- Analyze an application protocol
  - Example: Block Skype
- Analyze packets to verify their format and content
- Eliminate unwanted packets
  - DoS, exploits, viruses
- Eliminate packets that do not correspond to the current protocol's state

Filtering

- Filtering helps limiting traffic to useful services
- Can be based on multiple criteria:
  - SRC or DST IP address
  - Protocols (TCP, UDP, ICMP, …) and port numbers
  - Flags and options (SYN, ACK, ICMP message type, …)
- Filtering of source addresses prevents IP spoofing
- Filtering of flags allows defining the direction in which connections can be established
- Cons of filtering?

Cons of Filtering

- Add-on security hampers network evolvability
  - Hard to deploy new protocols across the Internet
  - IPv6
  - Stream Control Transmission Protocol (SCTP)
  - Multipath TCP (MPTCP) designed to be compatible with existing middleboxes
    - Still not easy to deploy because certain middleboxes remove unknown TCP options

Authentication

- The FW can require authentication before letting a connection through
- Outbound: allows limiting Internet access only to privileged users
- Inbound: allows authorizing access to internal resources for offsite employees
- Authentication can be done based on a local database or by interaction with a central database

Remote Network Access

- A FW may realize a Virtual Private Network (VPN) service to allow remote users to access the LAN
  - More on VPNs in the IPSec lecture
- The remote user establishes an encrypted connection (a tunnel) with the FW
- The user finds himself just as if he were in the LAN

Encryption

- A FW can encrypt or decrypt traffic that traverses a less secure zone
- Examples:
  - Interconnection between remote sites via the Internet
  - Remote network access

Logging

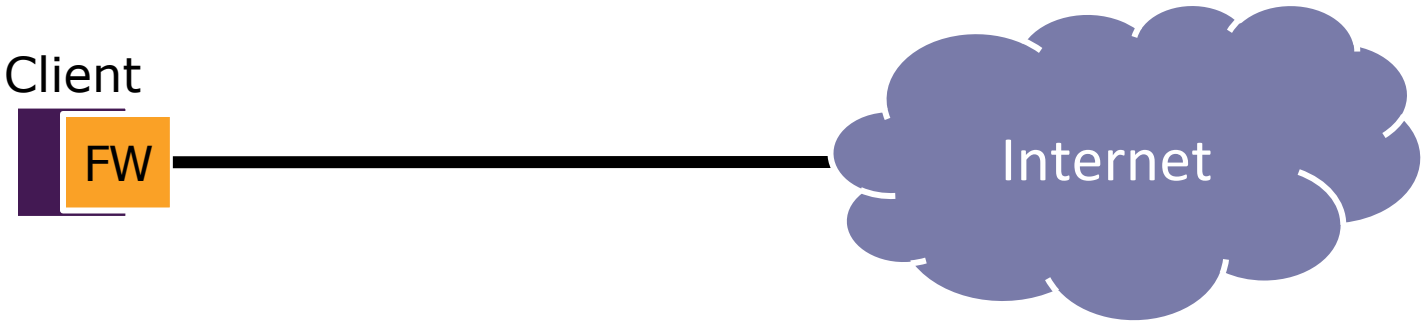
- It is important to be protected but we must also know when we are attacked and react accordingly
- Logs keep a trace of attack attempts
- They also allow verifying that the ports or destinations that we authorize are really needed
  - (least privilege)

Firewall Architectures

Firewall Architectures

- Personal Firewall
- NAT + Filtering
- FW with demilitarized zone
- Sandwiched demilitarized zone

Personal Firewall

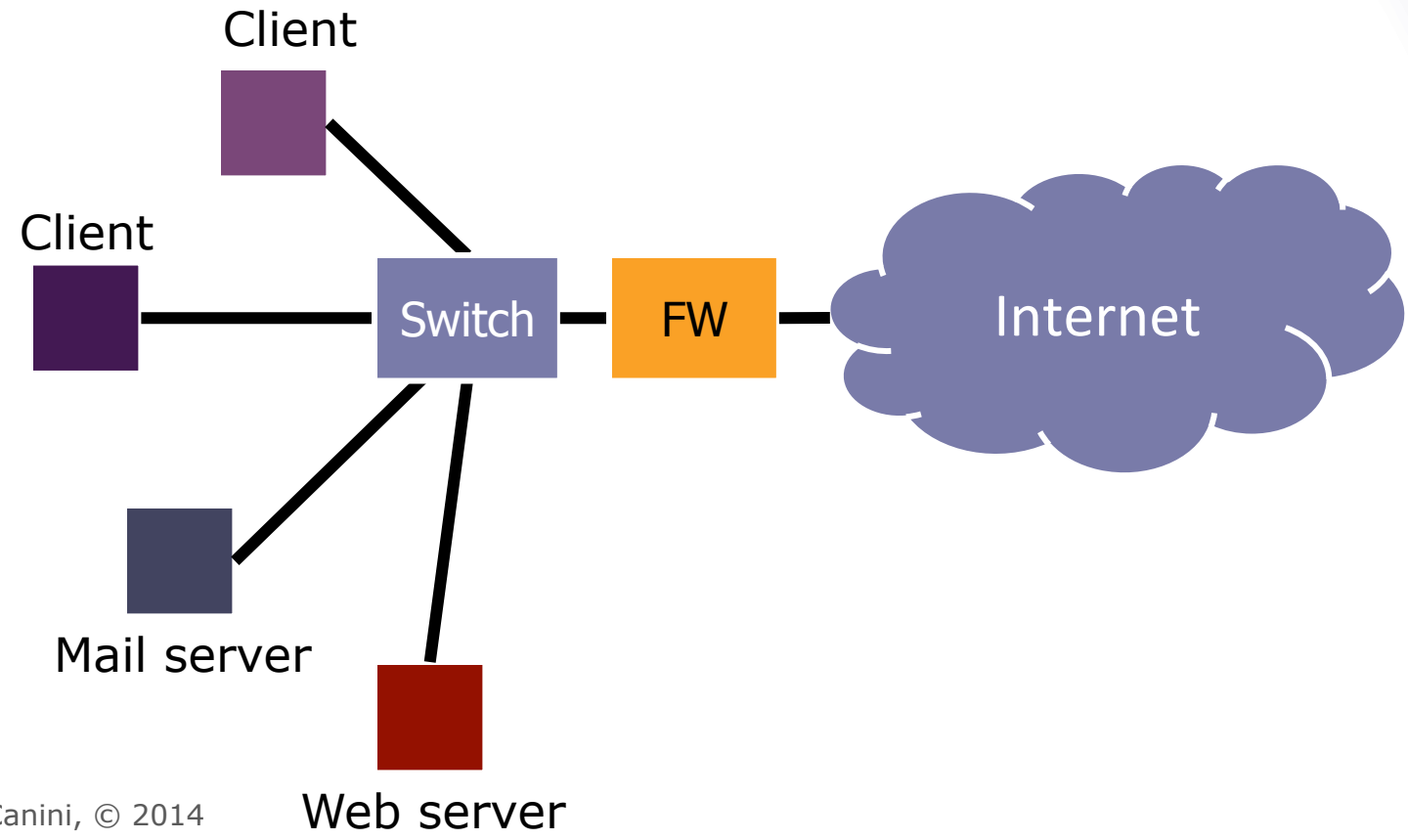


Personal Firewall

- The personal firewall initially prohibits all connections
- At each alarm, the user can authorize the application to connect

- Allows blocking backdoors, spywares, …
- An ideal complement to an anti-virus for safe surfing

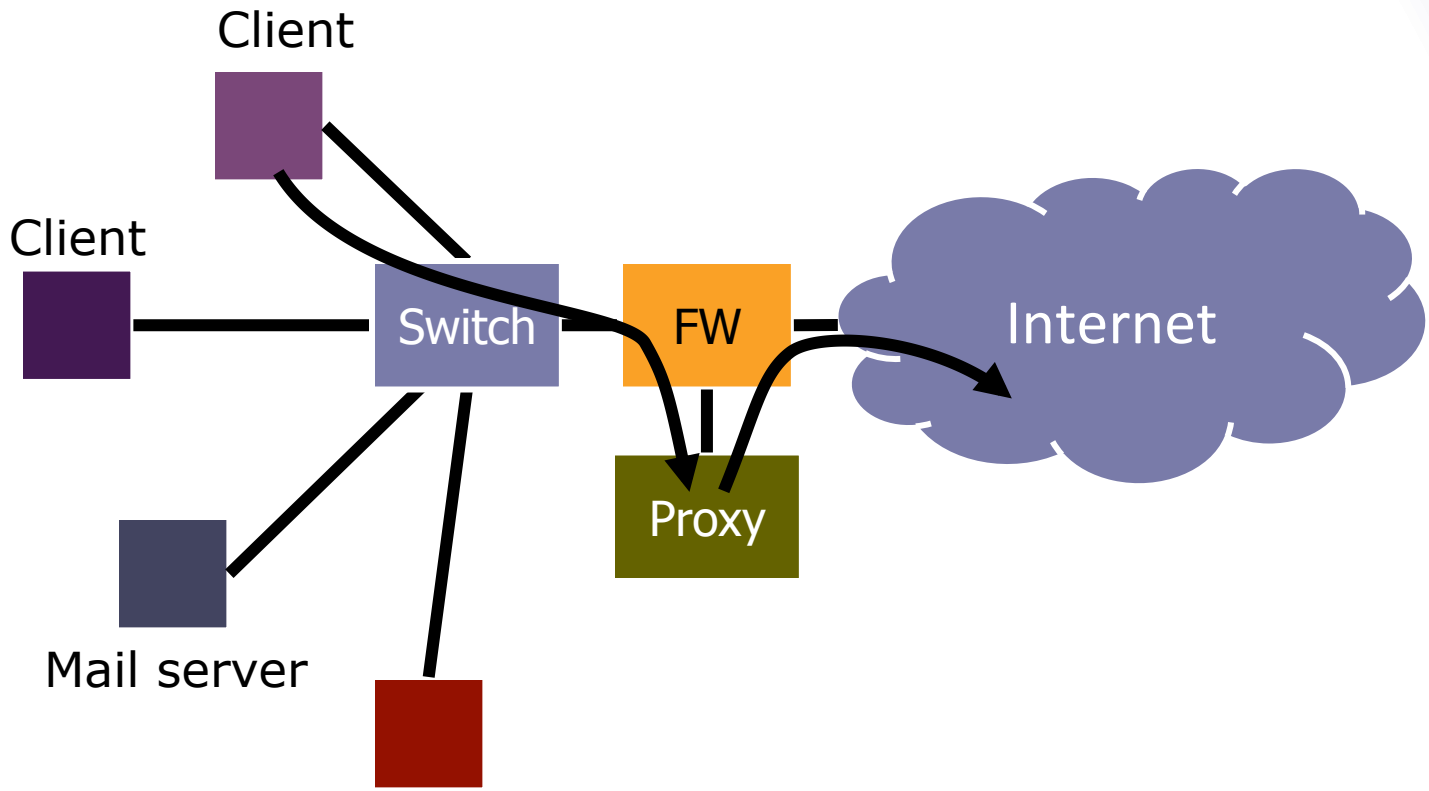
NAT + Filtering



NAT + Filtering

- Configuration
  - Dynamic NAT for all internal machines
  - Static NAT for all accessible servers
  - Outbound and Inbound filtering
- Limitations
  - No analysis of contents (virus) from Internet
  - Direct connections to internal servers (exploits, DoS)
- Applicability
  - Low security needs
  - Not for large public Web servers

Demilitarized Zone (DMZ) simple case



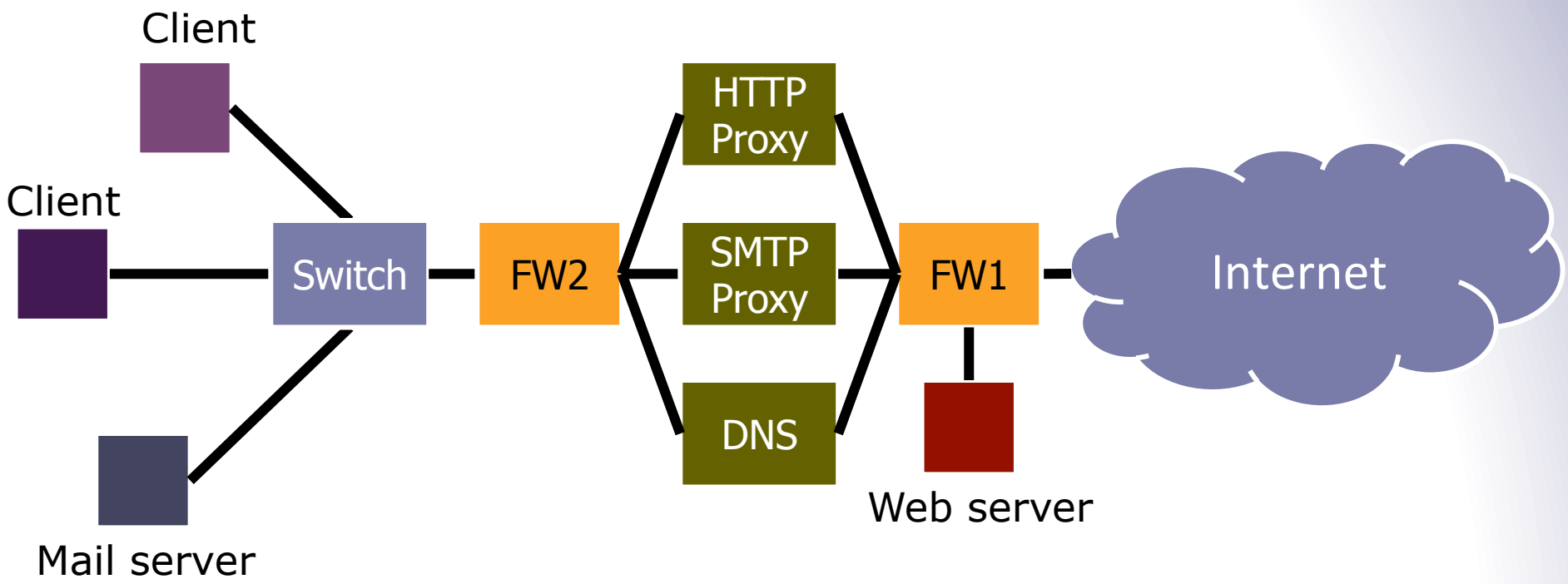
Demilitarized Zone (DMZ) simple case

- The DMZ is connected neither to the Internet, nor to the internal network
- Configuration
  - Internal machines can only connect to the proxy
  - Only the proxy can connect to the Internet
  - Outbound dynamic NAT
  - Inbound static NAT toward the proxy
  - Outbound and Inbound filtering

Demilitarized Zone (DMZ) simple case

- Limitations (**of the example**, not DMZ)
  - The firewall is a critical point
  - All services pass through the same proxy, a vulnerability on a single service can give access to all traffic
- Applicability
  - Medium security needs

Sandwiched DMZ



Sandwiched DMZ

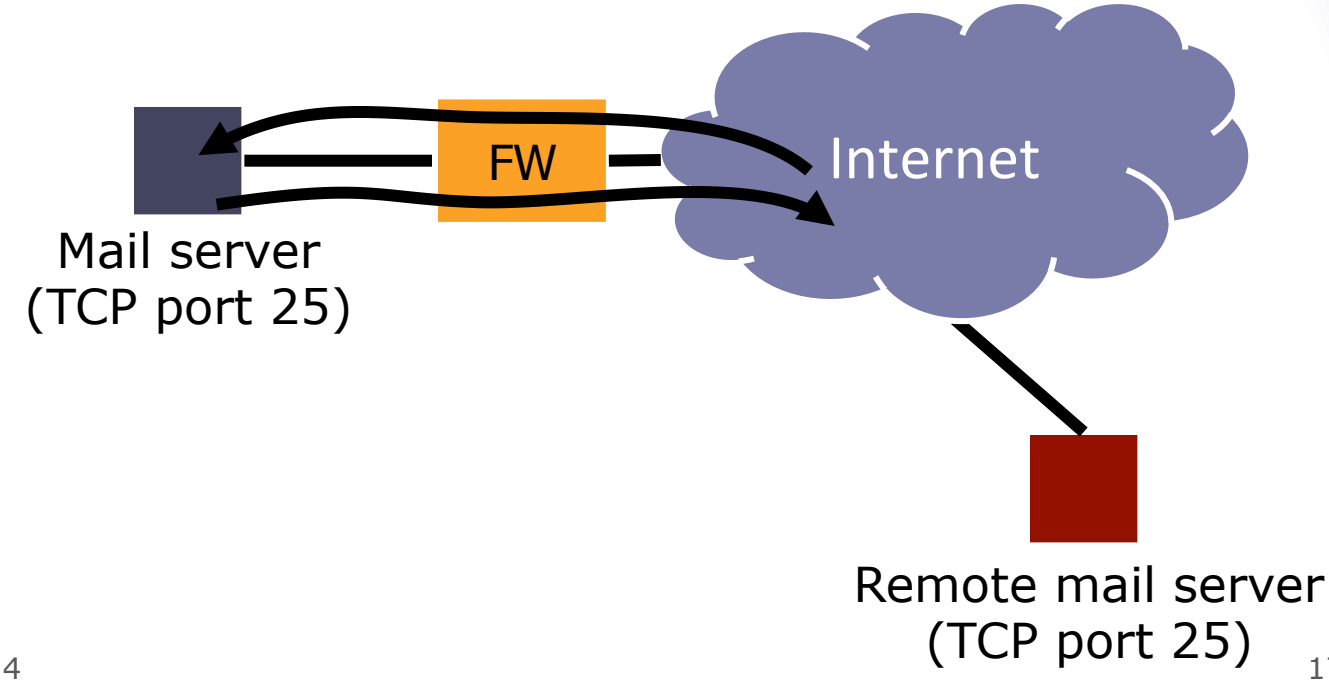
- Configuration
  - Internal machines can only connect to the proxies
    - (one proxy per protocol)
  - Only proxies can connect to the Internet
  - No routing in proxies
  - Outbound dynamic NAT, inbound static NAT
  - Outbound and Inbound filtering
- Applicability
  - High security needs

Filtering Rules

Filtering Rules (Organization)

- Filtering rules are specified in a list
- The firewall runs through the list until it finds a rule that applies
- The firewall executes the action specified by the matching rule and moves on to the next packet
- We create a last rule that prohibits all that has not been authorized

Filtering: A Simple Example



Filtering: A Simple Example

	Src	Port	Dst	Port	Proto	Action
1	Any	Any	128.3.3.1	25	tcp	allow
2	128.3.3.1	25	Any	Any	tcp	allow
3	128.3.3.1	Any	Any	25	tcp	allow
4	Any	25	128.3.3.1	Any	tcp	allow
5	Any	Any	Any	Any	tcp	deny

Filtering: A Simple Example

Problem:

- All ports of the server are accessible as long as the attacker chooses port 25 as source port!

Filtering: Corrected Example

	Src	Port	Dst	Port	Proto	ACK	Action
1	Any	Any	128.3.3.1	25	tcp	ACK = *	Allow
2	128.3.3.1	25	Any	Any	tcp	ACK = 1	allow
3	128.3.3.1	Any	Any	25	tcp	ACK = *	allow
4	Any	25	128.3.3.1	Any	tcp	ACK = 1	allow
5	Any	Any	Any	Any	Any	ACK = *	deny

- Specifying the ACK flag prevents sending of SYN packets and hence the establishment of connections

Filtering: Corrected Example

Problem:

- The attacker can still send unsolicited ACK packets (scanning, DoS)

Filtering: Stateful

	Src	Port	Dst	Port	Proto	Action
1	Any	Any	128.3.3.1	25	tcp	allow
3	128.3.3.1	Any	Any	25	tcp	allow
5	Any	Any	Any	Any	Any	deny

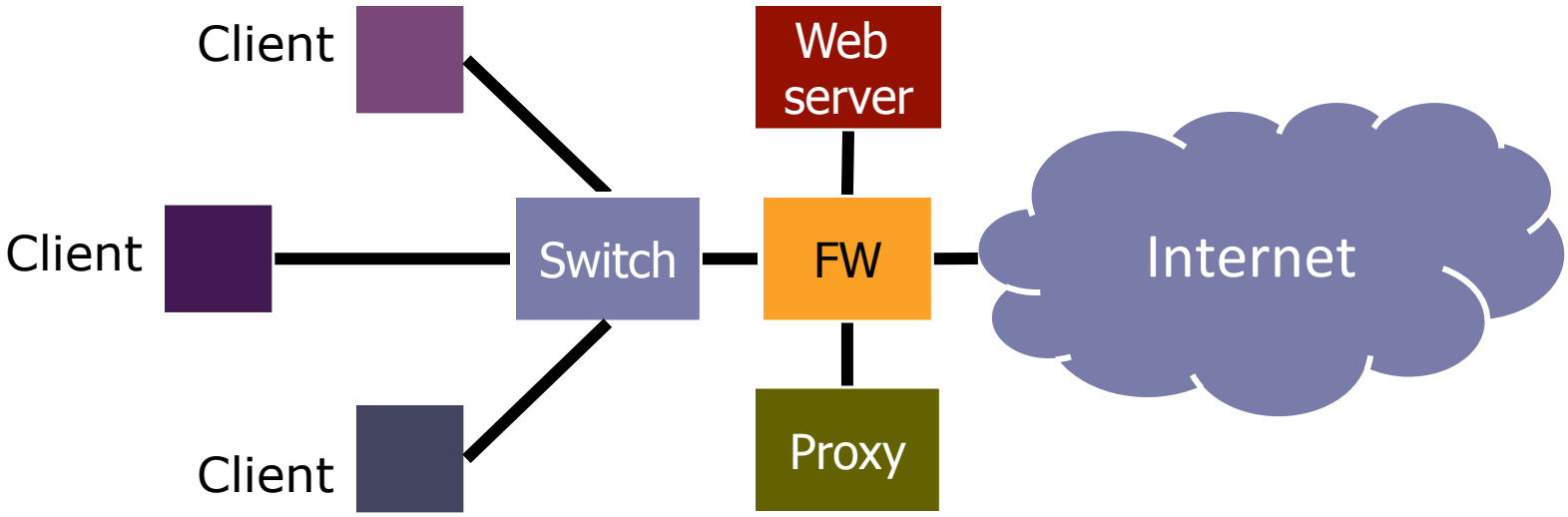
- Stateful FW knows about established connections and can automatically authorize returning traffic
- Safer:
  - No unsolicited packets
  - Simpler to configure, hence less errors

Organization of Filtering Rules

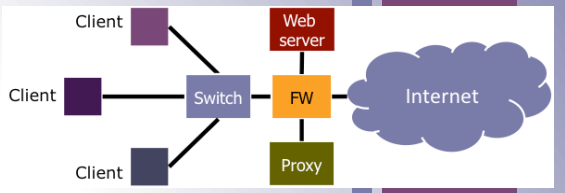
- The **order** in which the rules are specified matters!
- When there are many rules, it is important to organize them systematically

Organizing Rules: Example

- FW should allow inbound connections to the DMZ Web server
- FW should allow outbound connections to the Internet only through the DMZ Proxy

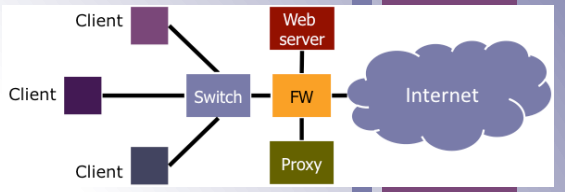


Organizing Rules: Example



	Src	Port	Dst	Port	Proto	Action
1	Any	Any	dmz-web	80	tcp	allow
2	Internal	Any	dmz-proxy	8080	tcp	allow
3	Internal	Any	Any	Any	tcp	deny
4	Any	Any	Any	Any	Any	deny

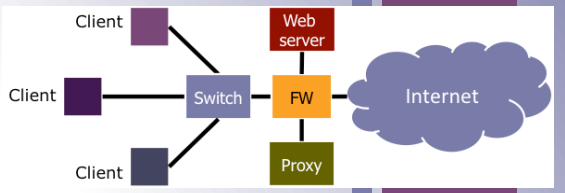
Organizing Rules: Example



	Src	Port	Dst	Port	Proto	Action
1	Any	Any	dmz-web	80	tcp	allow
2	Internal	Any	dmz-proxy	8080	tcp	allow
3	Internal	Any	Any	Any	tcp	deny
4	Any	Any	Any	Any	Any	deny

- Rule 1 allows internal machines to access dmz-web, while rule 3 should have prohibited it

Organizing Rules: Example



	Src	Port	Dst	Port	Proto	Action
1	Internal	Any	dmz-proxy	8080	tcp	allow
2	Internal	Any	Any	Any	tcp	deny
3	Any	Any	dmz-web	80	tcp	allow
4	Any	Any	Any	Any	Any	deny

- Rule 3 does not influence the internal traffic anymore

Organizing Rules: Method

- We define a security level for each zone
- We group rules by zones in descending order of security level
- Each groups consists of four parts:
  - Explicit authorizations for inbound traffic
  - General prohibition for inbound traffic
  - Explicit authorizations for outbound traffic
  - General prohibition for outbound traffic

Organizing Rules: 4-zone Example

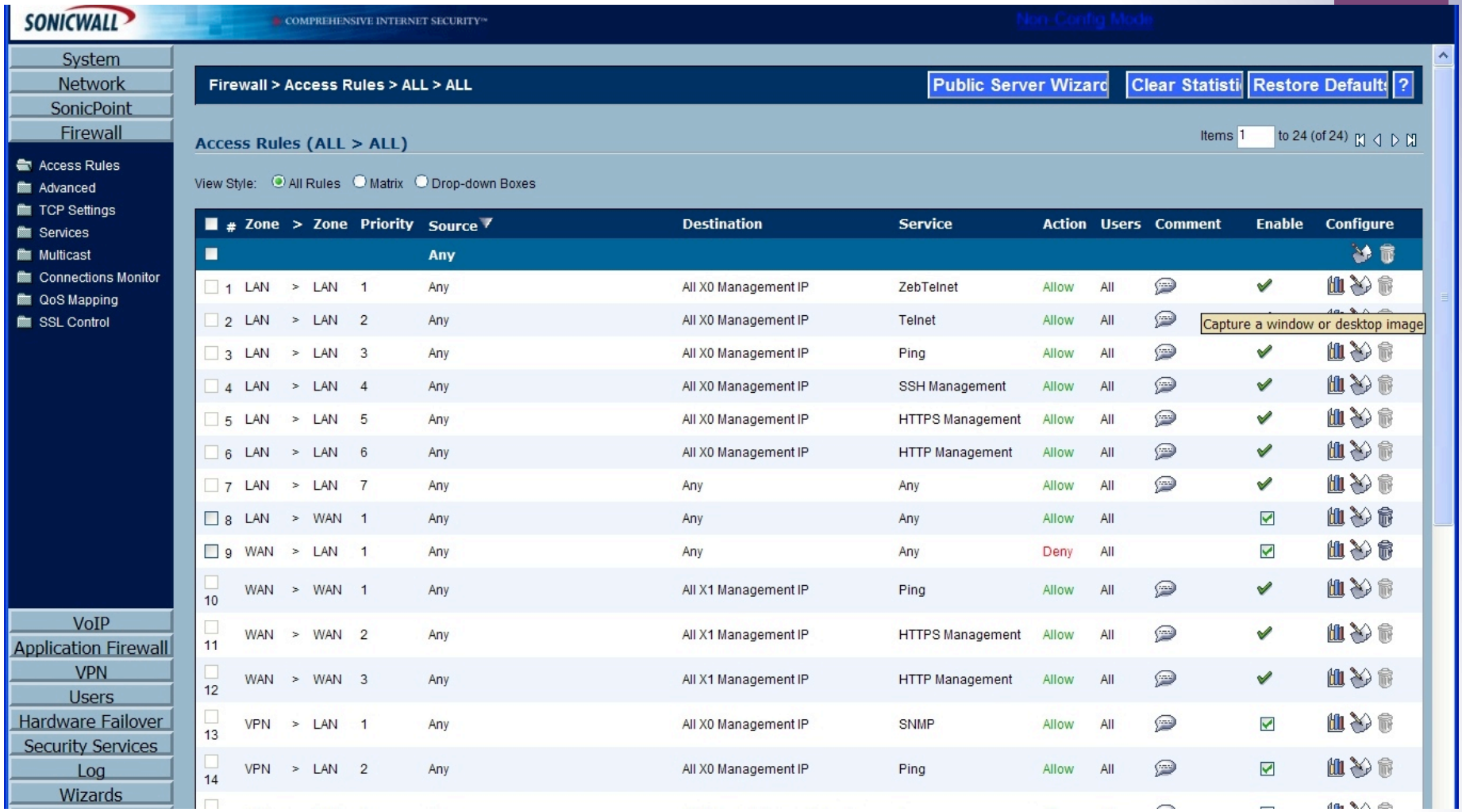
Zone	Rule	Src	Port	Dst	Port	Prot	Action
Zone 1	1	bob	any	alice	23	tcp	allow
	2	any	any	zone_1	Any	any	deny
	3	alice	any	bob	22	tcp	allow
	4	zone_1	any	any	any	any	deny
Zone 2	5	authorized traffic entering zone 2					allow
	6	other traffic entering zone 2					deny
	7	authorized traffic leaving zone 2					allow
	8	other traffic leaving zone 2					deny
Zone 3	9	authorized traffic entering zone 3					allow
	10	other traffic entering zone 3					deny
	11	authorized traffic leaving zone 3					allow
	12	other traffic leaving zone 3					deny
	13	any	any	any	any	any	deny

Organizing Rules: Properties

- For each zone, it is sufficient to declare the flow towards less secure zones
- The flow towards more secure zones cannot be influenced anymore (operation goal): "any" refers to lower levels
- A rule that implies 2 zones appears in the block related to the more secure zone
- The block related to the last zone is empty
- The last rule (any-any) must not be required
  - By activating logging on that rule we may detect possible errors

Example: SonicWall

https://sonicwall.com



Summary

- 7 basic principles of secure system design
- Firewalls + NAT
  - Protect networks, allow useful services
  - Stateful is safer
  - But more advanced protection requires looking into packet contents
  - How to deal with encrypted traffic?
  - Rule ordering matters!
    - Manual configuration leads to potential for errors

Any questions?

Stay tuned

+

Next time you will learn about

Proxies