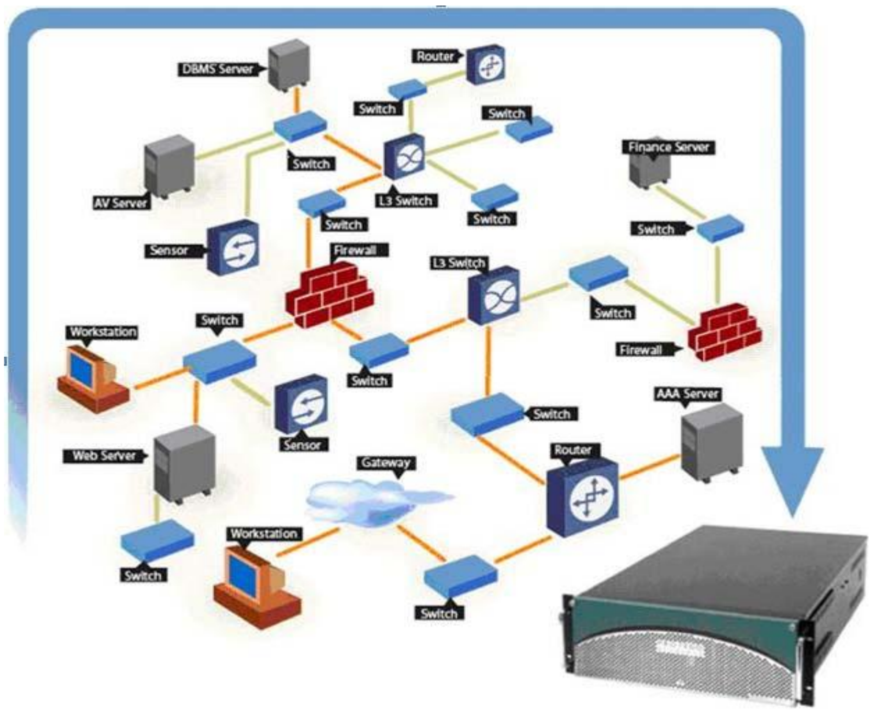
[MikrotikBrasil]
Routers & Wireless Systems



Layer II Security

Poland MUM – **Wrocław** - March 2010

Eng. Wardner Maia

[MikrotikBrasil]
Routers & Wireless Systems

Introduction

Name: Wardner Maia
Country: Brazil

→ Electronic/Telecommunications Engineer
→ Internet Service Provider since 1995
→ Wireless Internet Service Provider since 2000
→ Teaches Wireless for WISP’s since 2002, Mikrotik since 2006
→ Mikrotik Certified Trainer since June, 2007

Introduction

MD Brasil Information Technology and Telecommunications

→ Internet Service Provider, in the states of São Paulo and Minas Gerais
→ Mikrotik Distributor, equipment integrator
→ Mikrotik Training Partner
→ Consulting Services

www.mdbrasil.com.br
www.mikrotikbrasil.com.br

Target audience and objectives

Target Audience:

→ ISP's and WISP's that run small / medium growing networks

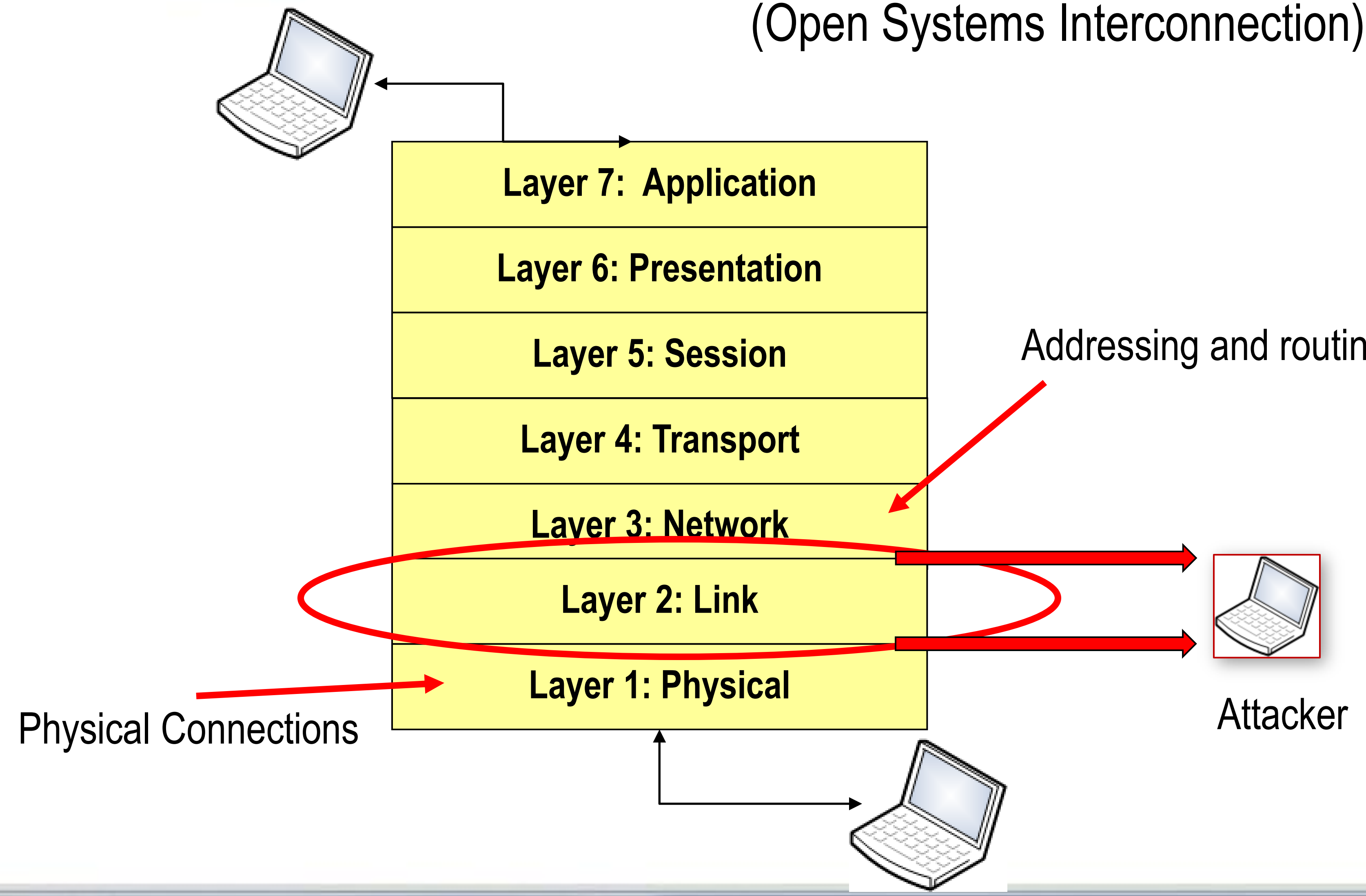
Objectives:

→ To discuss the most common network topologies and their issues regarding access security and network availability..

→ To understand conceptually the existing threats related to layer 2 vulnerabilities with practical demonstrations .

→ To discuss possible countermeasures using Mikrotik RouterOS listing the "best practices" to ensure security at this level of OSI model..

OSI Model
(Open Systems Interconnection)



Why Layer II ?

→Network Security is a broad question and should be viewed under different perspectives, from the physical to the application layer of OSI model.

→Security issues are quite independent for each layer and no matter how strong are the Security measures adopted for the upper ones, if a low layer is compromised the whole security is compromised. Authentication, confidentiality, integrity and availability must be guaranteed for all layers.

Layer 7: Application
Layer 6: Presentation
Layer 5: Session
Layer 4: Transport
Layer 3: Network
Layer 2: Link
Layer 1: Physical

Why Layer II ?

→ If compared to the many efforts focused in application and network layer, there are few ones regarding to the infrastructure breaches inherent to the existing L2 protocols weakness.

→ Good practices adopted to enhance Layer 2 security are important not only for the security itself, but to ensure a performance optimization, since a lot of garbage traffic can be dropped with appropriate measures.

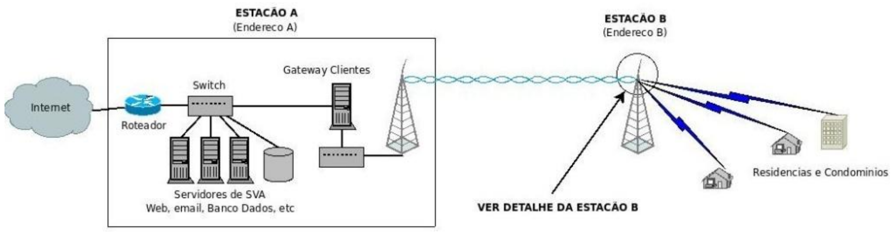
Layer 7: Application
Layer 6: Presentation
Layer 5: Session
Layer 4: Transport
Layer 3: Network
Layer 2: Link
Layer 1: Physical

AGENDA

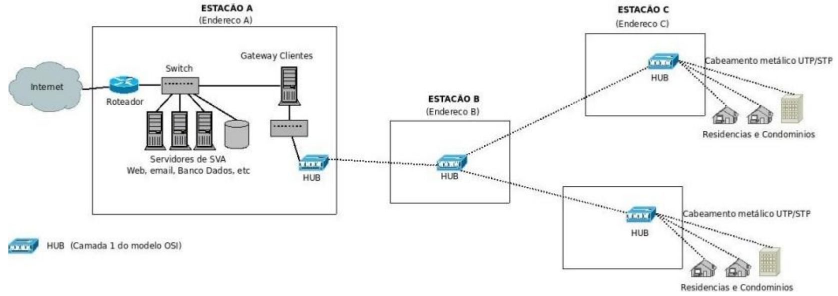
→Common topologies for IP Networks
→Bridging, Switching and Layer II Firewalls

→ Layer II attacks and protocol vulnerabilities:
- → CAM table overflow / neighborhood protocols explotation.
- → VLAN´s and Spanning Tree protocols explotation.
- → DHCP Starvation
- → ARP Cache poisoning – MitM Attack
- → Defeating users and providers Hotspot and PPPoE based
- → Wireless deauthentication attacks

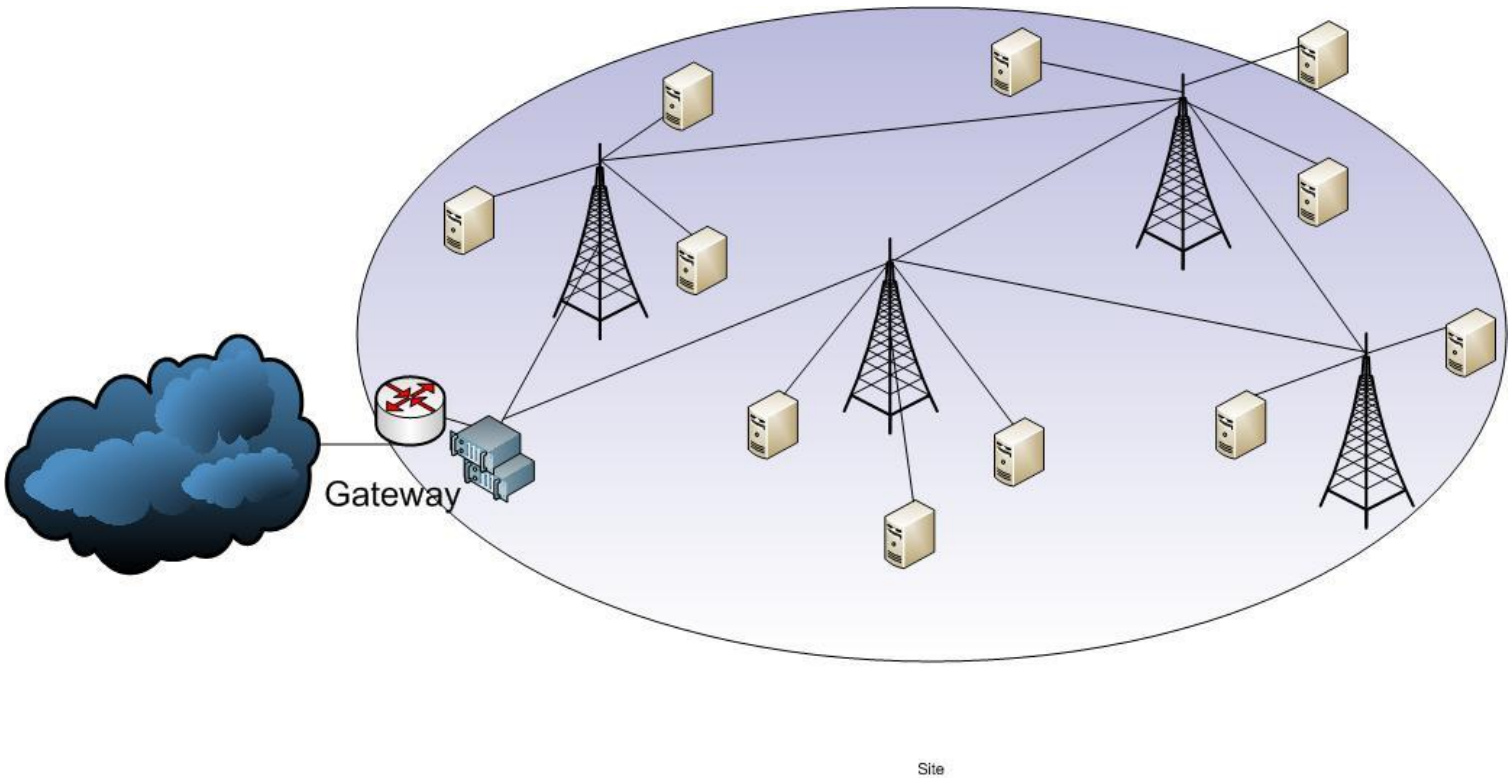
→ Countermeasures and best practices to face L2 issues using Mikrotik RouterOS



→ Common topologies for IP networks
→ Bridging x Switching
→ Layer II Firewalls (Bridge Filter)



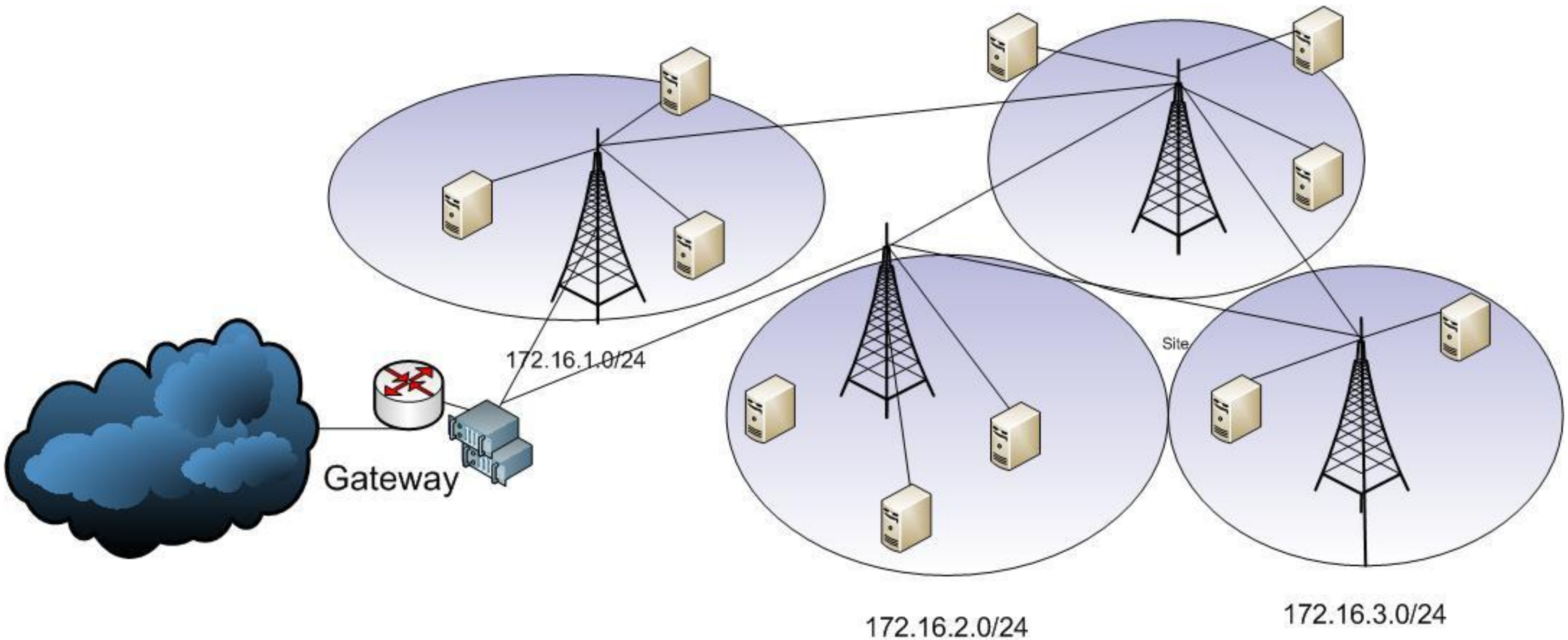
Typical Layer 2 Network



Customer gateway is border gateway

Just one broadcast domain

Typical Routed Network



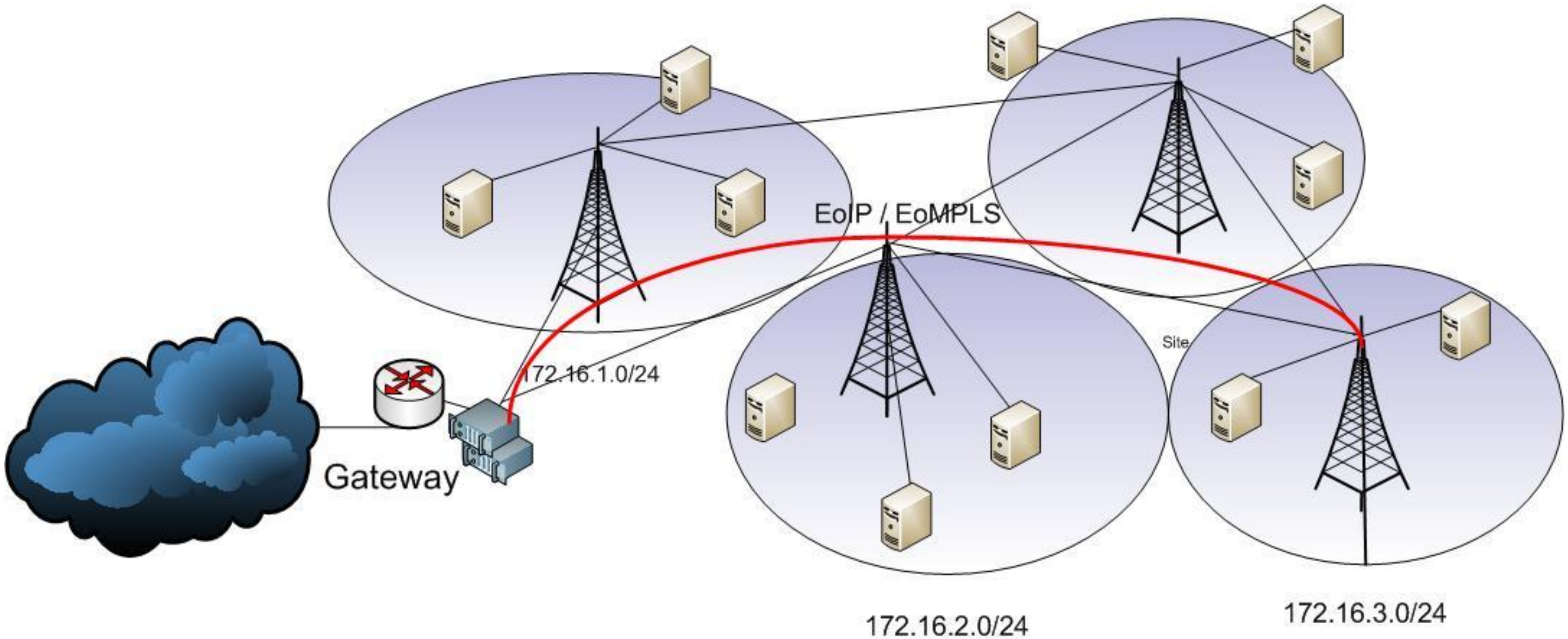
Customer’s gateway is distributed and close to it.

Segregated Broadcast domains

→ Even in such type of network there are bridging segments that should be watched

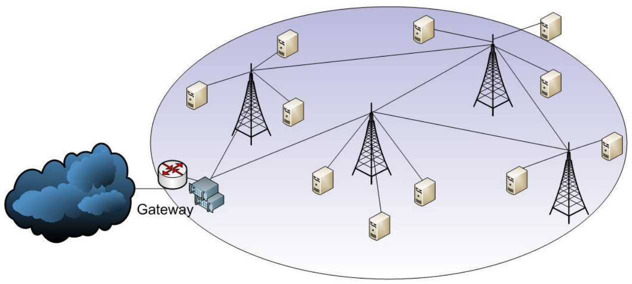
[MikrotikBrasil]
Routers & Wireless Systems

Typical Routed Network with concentrated gateway "Bridge over Routing"



Usually dynamic routing with transparent tunneling from the customer to the main gateway – (EoIP / EoMPLS, etc)

Layer 2 Networks



ATM, Frame Relay, MPLS (layer "2.5"), etc

We will focus on

- Bridged IP Networks:
  - → Fixed IP
  - → Dynamic IP with DHCP
  - → Hotspot
  - → Bridging over routing
- Layer II only network with PPPoE concentrator

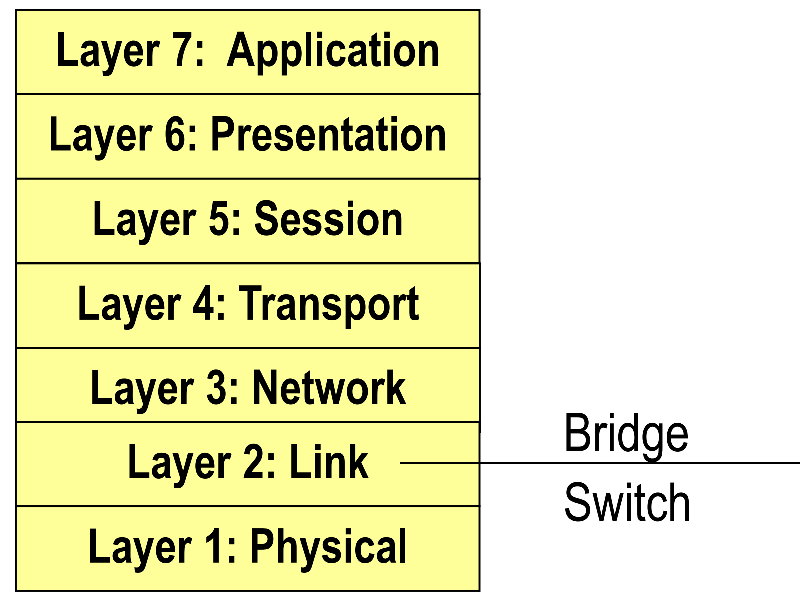
Bridging x Switching

Bridging x Switching

→ Both Bridging and Switching happen at layer II, but with a slightly difference

→Switching process is usually faster, because no processor cycle is required; Packets are forwarded at “wire speed”.

→ Since V.4, Mikrotik RouterOS support switching for some equipments.



Switching

→ The switch keeps a table with the MAC address connected to it, establishing a relationship with the port from where they were "learned"

→ When a MAC address does not exist in the table, it is sought in all ports.

→ The address space (Host table o CAM table) is limited and when it is full the switch forward the packets tor all ports behaving as it was a HUB!

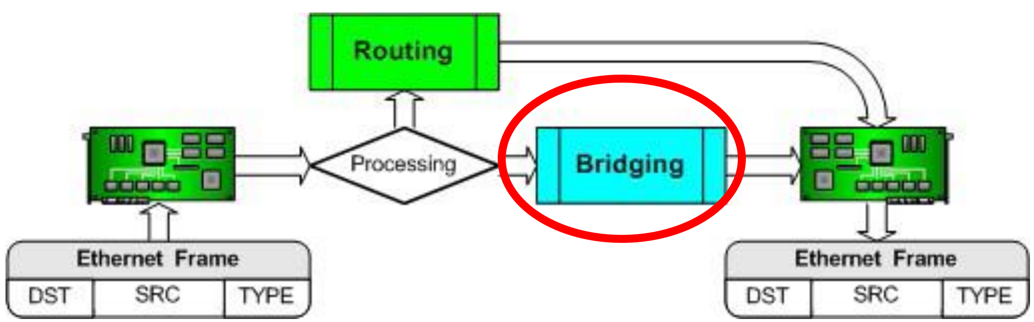
Feature	Atheros8316	Atheros7240	ICPlus175D	Other
Port Switching	yes	yes	yes	yes
Port Mirroring	yes	yes	yes	no
Host table	2k entries	2k entries	no	no
Vlan table	4096 entries	16 entries	no	no
Rule table	32 rules	no	no	no
	(RB450G)	(RB750)	(RB450)	

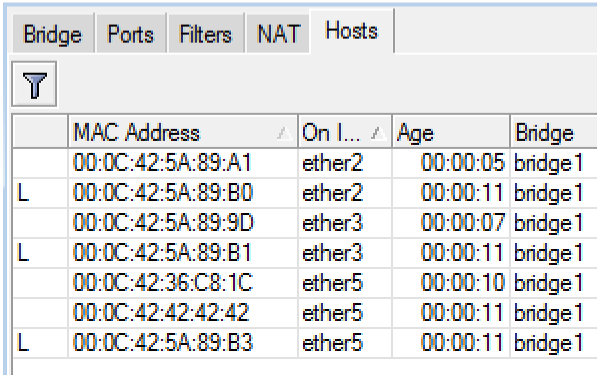
Bridging

→ Like the Switch, the Bridge keeps a table with the MAC addresses and ports. Each Bridge in the same segment has all MAC address that were “learned” from other Bridges.

→ The Host table does not have a fixed limit but is obviously limited by hardware memory resources

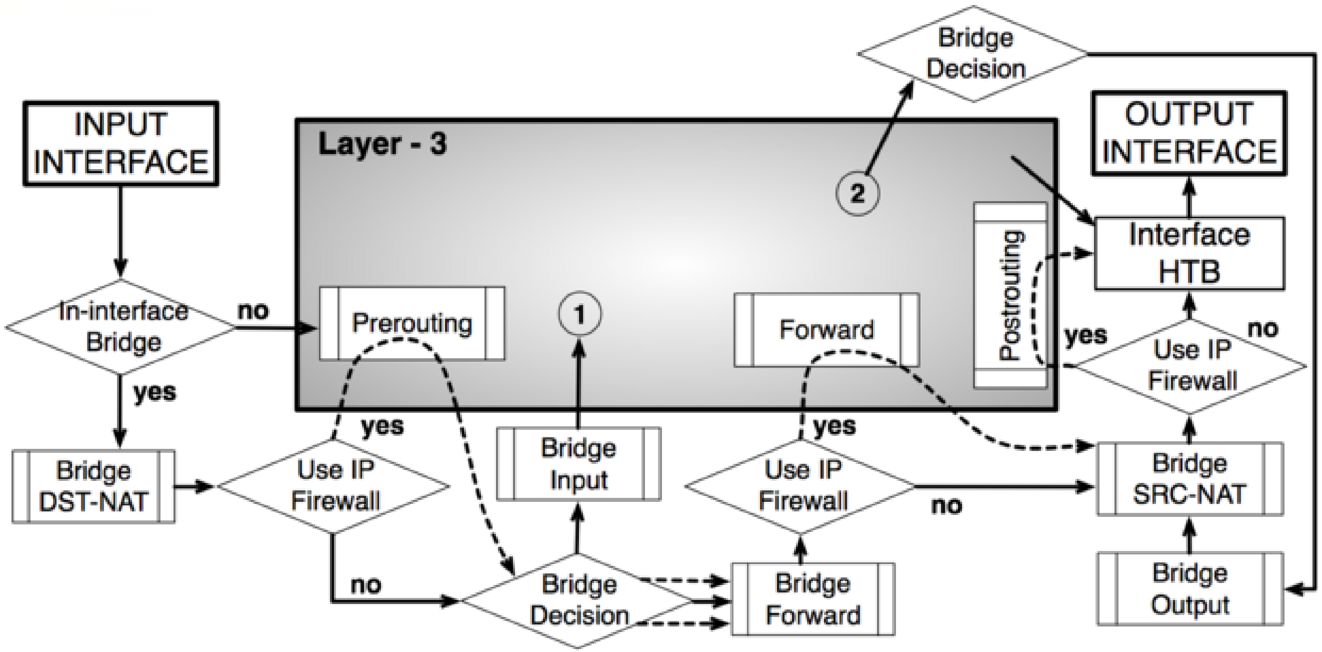
→ With RouterOS Bridging features is possible to inspect ethernet frames an to aply filters, marks, etc.

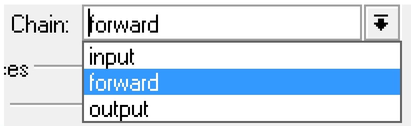


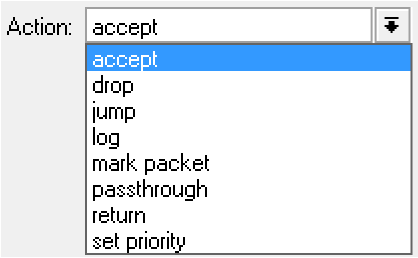


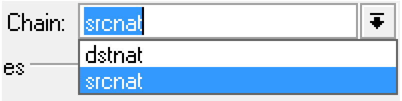
	MAC Address	On I...	Age	Bridge
	00:0C:42:5A:89:A1	ether2	00:00:05	bridge1
L	00:0C:42:5A:89:B0	ether2	00:00:11	bridge1
	00:0C:42:5A:89:9D	ether3	00:00:07	bridge1
L	00:0C:42:5A:89:B1	ether3	00:00:11	bridge1
	00:0C:42:36:C8:1C	ether5	00:00:10	bridge1
	00:0C:42:42:42:42	ether5	00:00:11	bridge1
L	00:0C:42:5A:89:B3	ether5	00:00:11	bridge1

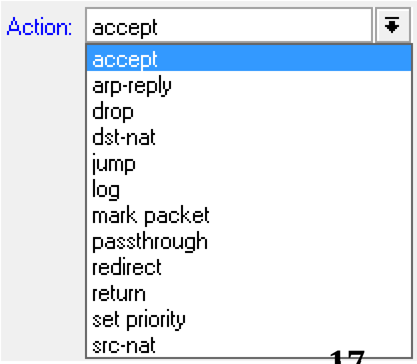
Layer 2 filters









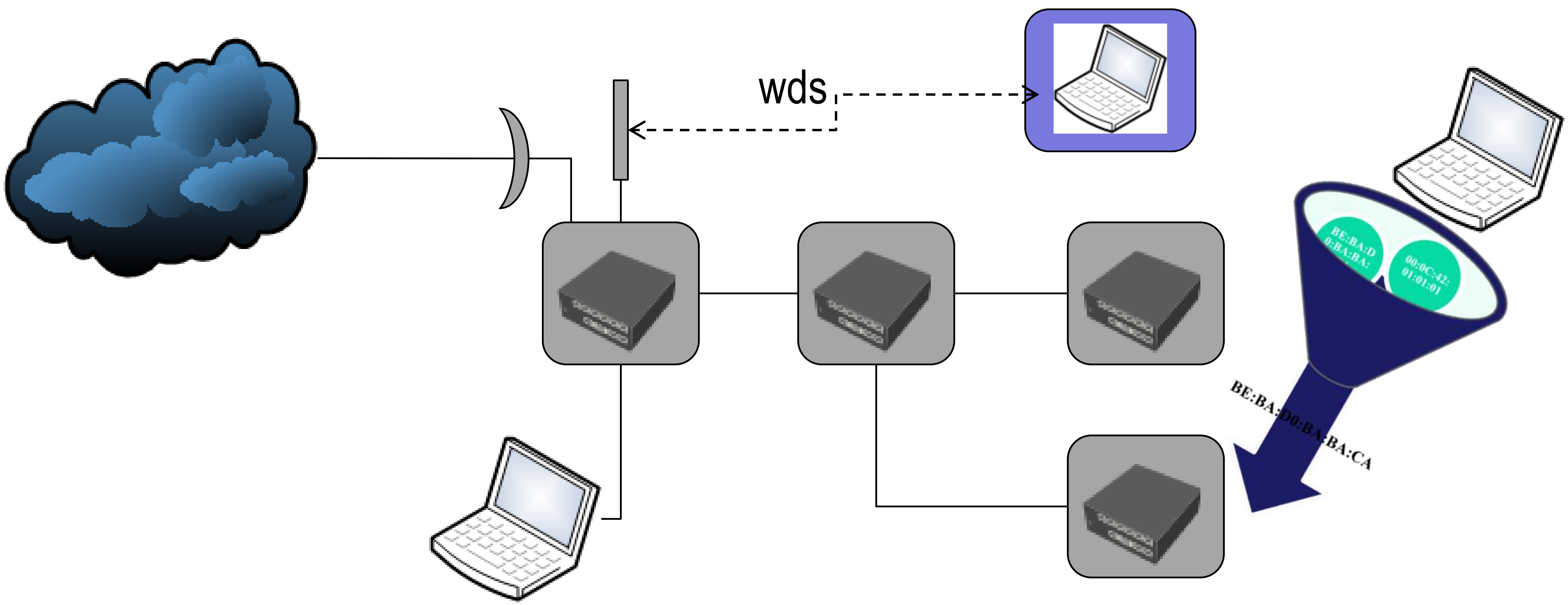


Layer 2 attacks

MAC Flooding

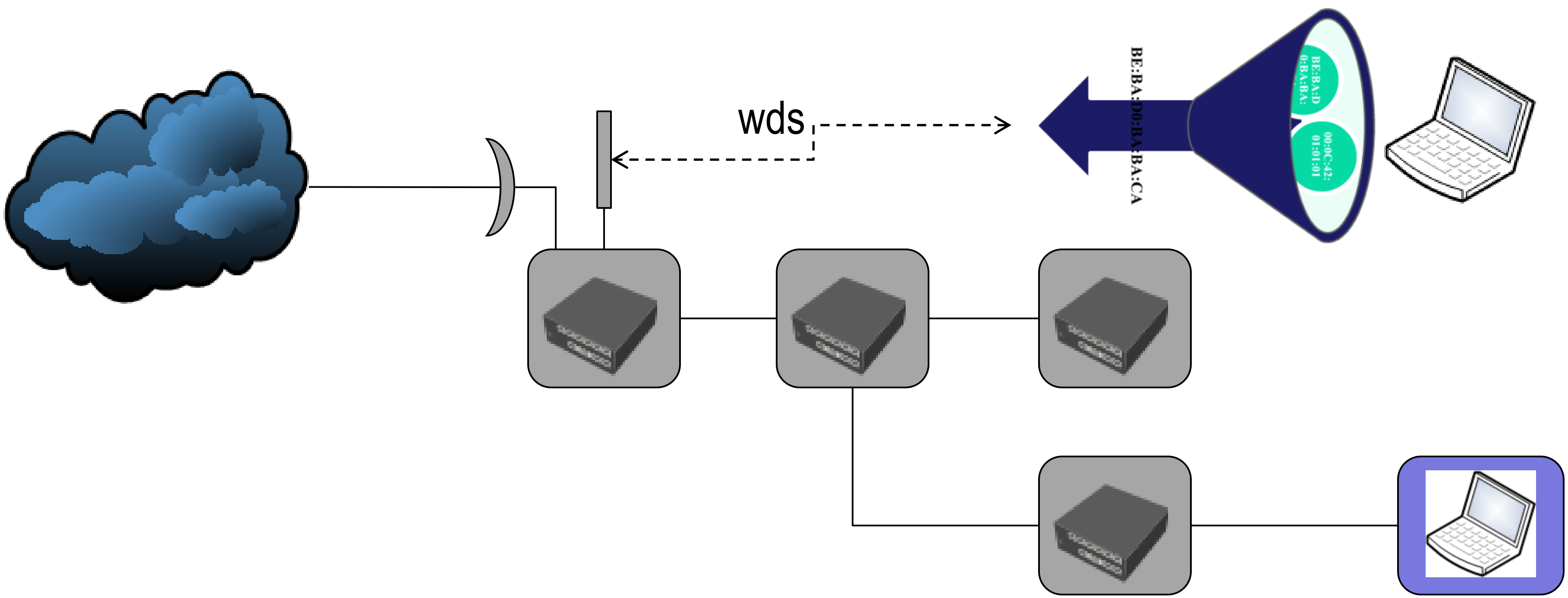
Attacks against Switches and Bridges
MAC flooding

There are a lot of tools designed with the purpose or “network security auditing” that you can flood a lot of MAC address at any point of the bridged structure.

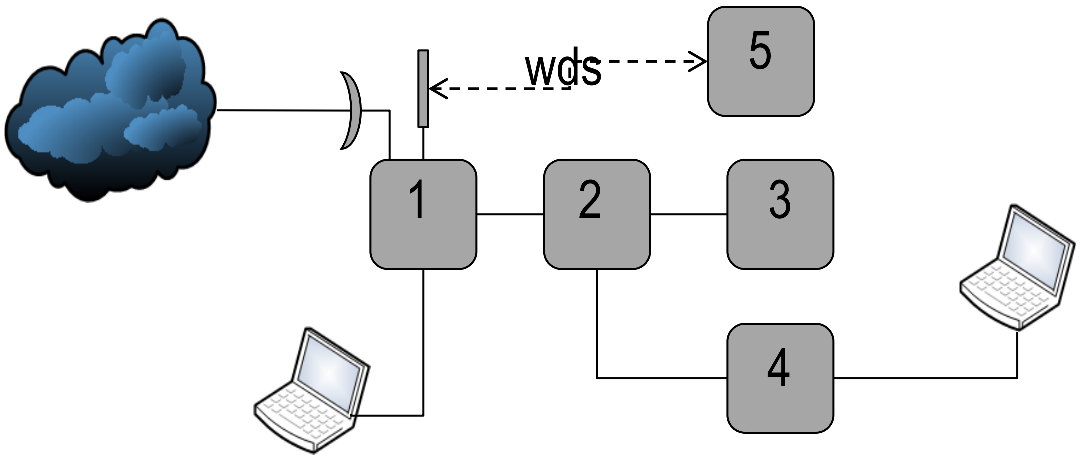


Attacks against Switches and Bridges
MAC flooding

The flooding can be launched from any port of the whole structure, even from Wireless interfaces in bridge mode.



Mac Flooding
DEMO



→ Lauching the attack from 4, we can see the effect in all bridged equipments.
→ Host tables increase very fast ant network performance goes down.

MAC flooding - Countermeasures

Switches:

→ Since the CAM table is limited, the attack does not cause DoS, but the switch starts to behave like a HUB, forwarding packets for all ports. Sniffing in promiscuous mode is possible.

→ When using Mikrotik RouterOS switching capability, there is nothing to do but only to avoid unauthorized people to have physical access on such structure.

→It would desirable some feature like Cisco's “port security” limiting the total of MAC addresses learned by each port.

[MikrotikBrasil]
Routers & Wireless Systems

MikroTik SwOS

Logout

Link | Forwarding | Statistics | VLAN | VLANs | Static Hosts | Hosts | SNMP | ACL | System

	Port1	Port2	Port3	Port4	Port5
Forwarding					
From Port 1	☐	☑	☑	☑	☑
From Port 2	☑	☐	☑	☑	☑
From Port 3	☑	☑	☐	☑	☑
From Port 4	☑	☑	☑	☐	☑
From Port 5	☑	☑	☑	☑	☐
Port Lock					
Port Lock	☐	☐	☐	☐	☐
Lock On First	☐	☐	☐	☐	☐
Port Mirroring					
Mirror Ingress	☐	☐	☐	☐	☐
Mirror Outgress	☐	☐	☐	☐	☐
Mirror To	◉	○	○	○	○
Bandwidth Limit					
Ingres Rate					
Outgres Rate					

MAC flooding - Countermeasures

Bridges:

→ Increasing the Host table “ad infinitum” the network will suffer delays, lost of packets, jitter, etc. The time to completely crash depends on equipment capabilities.

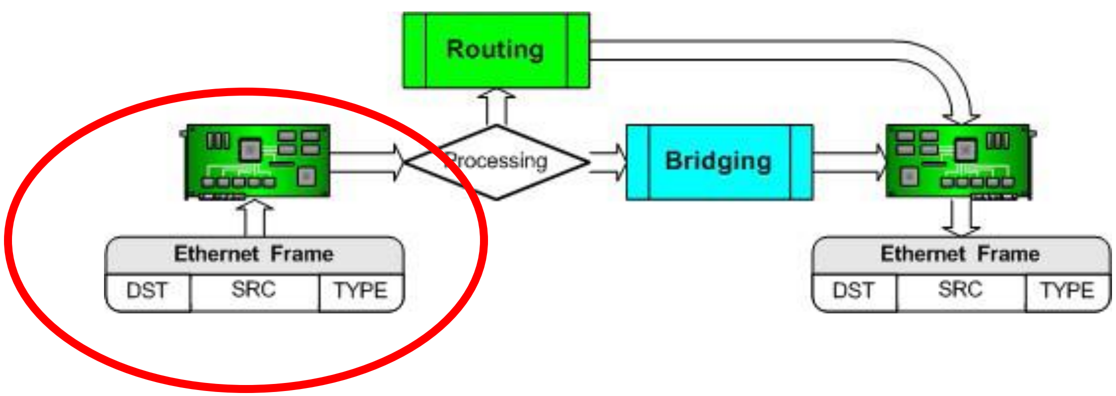
→ On the other hand, when using bridging we can inspect and apply filters to the ethernet frames.

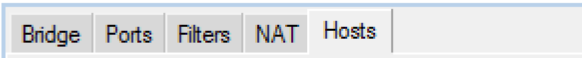
→Can we use Bridge Filter to thwart a MAC Flooding attack ?

???

MAC Flooding against Bridges Countermeasures

Why We cannot use Bridge Filter to thwart MAC flooding…





	MAC Address	On I...	Age	Bridge
	00:0C:42:5A:89:A1	ether2	00:00:05	bridge1
L	00:0C:42:5A:89:B0	ether2	00:00:11	bridge1
	00:0C:42:5A:89:9D	ether3	00:00:07	bridge1
L	00:0C:42:5A:89:B1	ether3	00:00:11	bridge1
	00:0C:42:36:C8:1C	ether5	00:00:10	bridge1
	00:0C:42:42:42:42	ether5	00:00:11	bridge1
L	00:0C:42:5A:89:B3	ether5	00:00:11	bridge1

→ Before passing through the filter, MAC's should be "learned" by the Bridge.

→ Because of this, Firewall Filter is useless to face this type of attack.

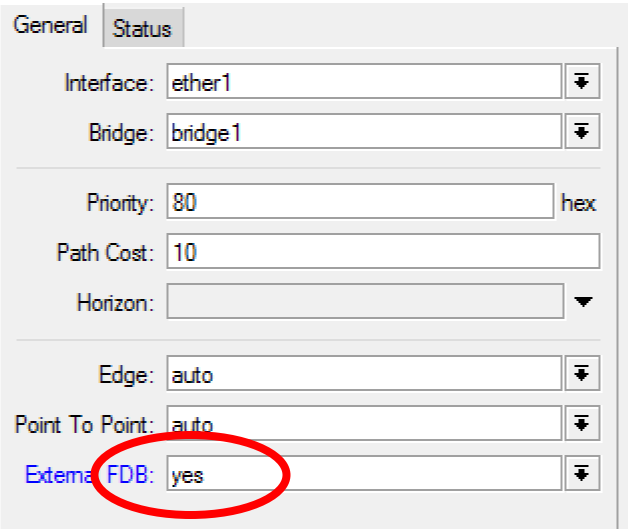
MAC Flooding against Bridges Countermeasures

Bridges:

→ It is possible to configure the border ports to search in an external Database an not in the host table. With this configuration (yes) there is no host table associated for that port.

→ This setting protects only the equipment where it is configured but the flood continue to compromise the other bridged.

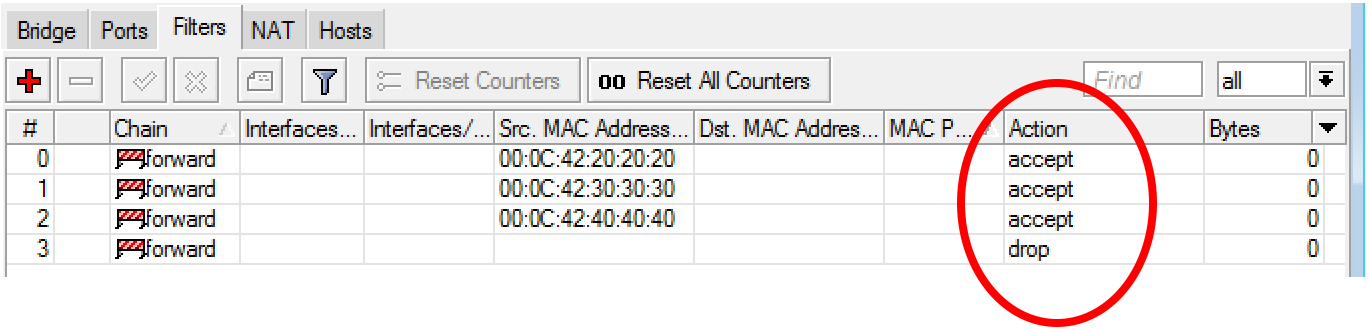
→ Fortunately, for the other equipment, we can use the Bridging filter features and accept only well known MAC addresses.



MAC Flooding against Bridges Countermeasures

So, MAC flooding countermeasure is only possible, combining external FDB for the border ports + Bridge Filter for intermediate hosts.

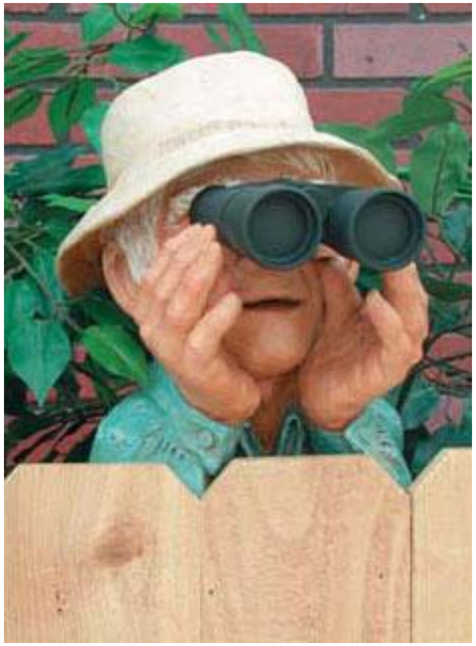
→ Because of with external FDB=yes turns the border bridge to act like a HUB, some kind of dynamic security could be achieved by means of a script that monitors the host table and turn on this setting only in case of anomalous behavior of the host table.



#		Chain	Interfaces...	Interfaces/...	Src. MAC Address...	Dst. MAC Addres...	MAC P...	Action	Bytes
0		forward			00:0C:42:20:20:20			accept	0
1		forward			00:0C:42:30:30:30			accept	0
2		forward			00:0C:42:40:40:40			accept	0
3		forward						drop	0

Layer II attacks

Exploiting Neighborhood
Discovery protocols

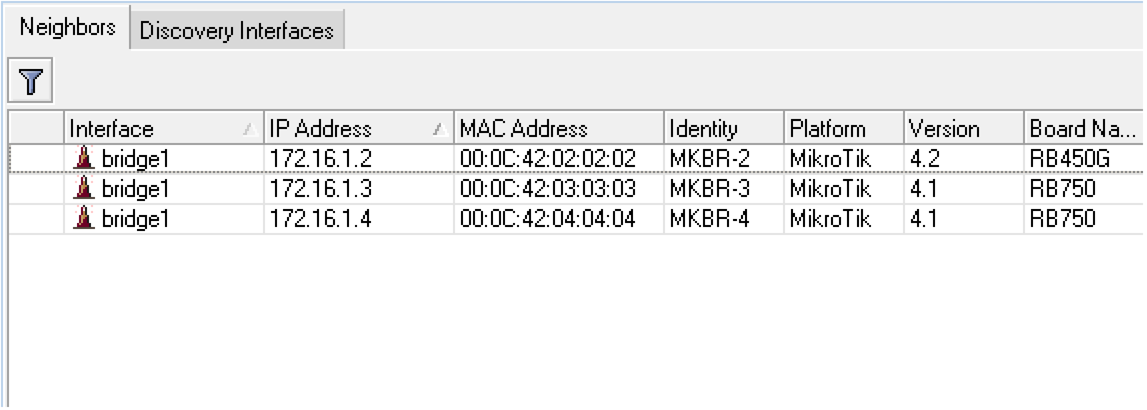


Exploiting Neighborhood Discovery Protocols

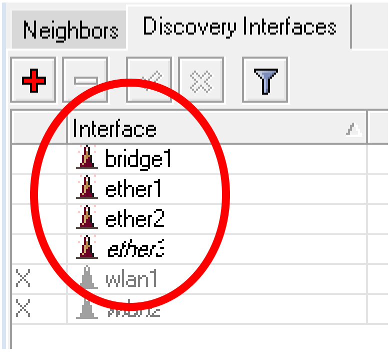
→ Neighbor Discovery Protocols are helpful for networking administrative tasks

→ Mikrotik RouterOS uses MNDP - Mikrotik Neighbor Discovery Protocol. (Cisco uses similar protocol – CDP – Cisco Discovery Protocol).

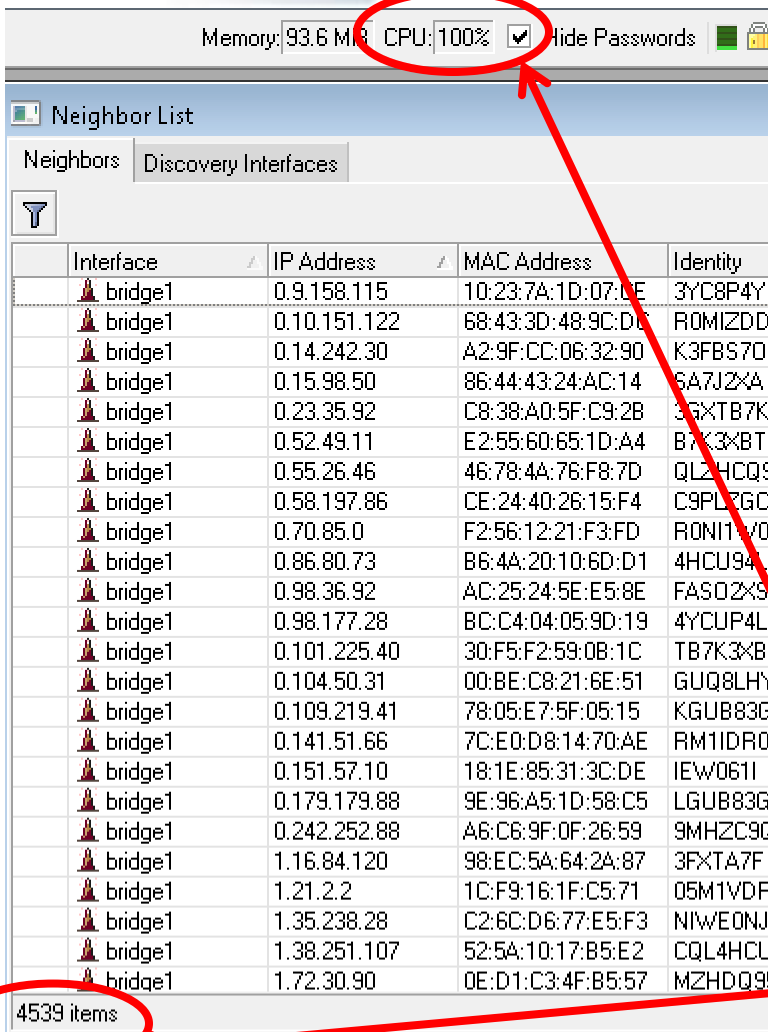
→ Both protocols are UDP based, broadcasting packets each 60 seconds over port 5678 and for all interfaces where the protocol is enabled.



Interface	IP Address	MAC Address	Identity	Platform	Version	Board Na...
bridge1	172.16.1.2	00:0C:42:02:02:02	MKBR-2	MikroTik	4.2	RB450G
bridge1	172.16.1.3	00:0C:42:03:03:03	MKBR-3	MikroTik	4.1	RB750
bridge1	172.16.1.4	00:0C:42:04:04:04	MKBR-4	MikroTik	4.1	RB750



Exploiting Neighborhood Discovery Protocols



→ Hacking tools developed to attack Cisco Routers can attack Mikrotik RouterOS too.

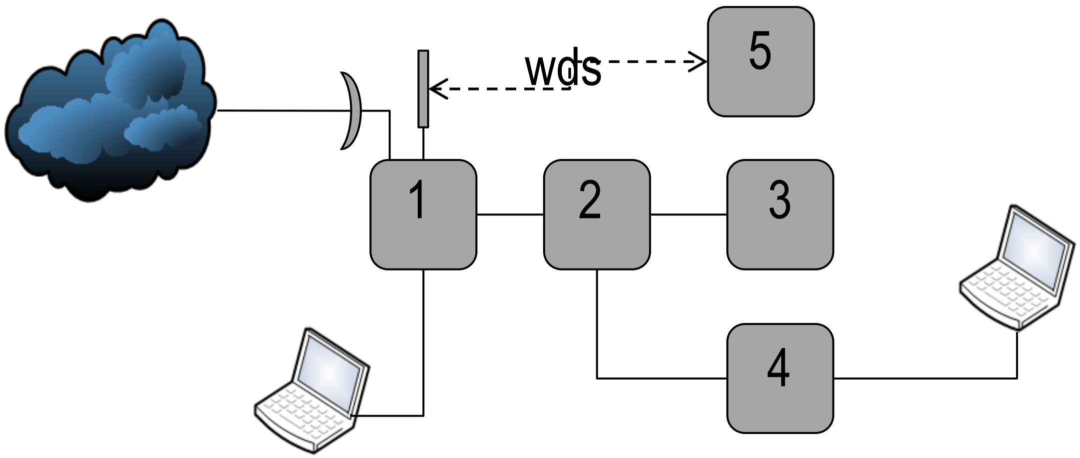
→ That tools can be used to get informations about the network or to cause Denial of Service.

→The attack can be triggered from any port of the Bridged Network and rapidly infects all hosts where the protocol is enabled.

15 seconds of attack against a RB433AH

Exploiting Neighborhood Discovery Protocols

DEMO



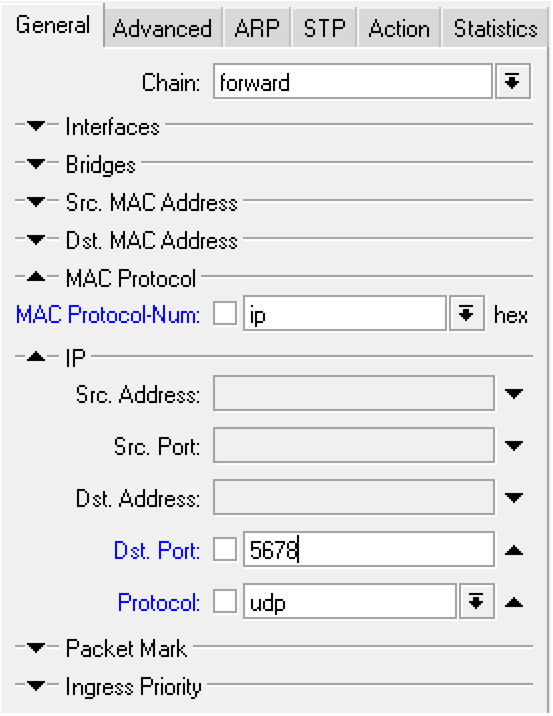
- Triggering the attack from 4
- Checking the effects at 1
- Protecting measures at 1
- Filtering at 4

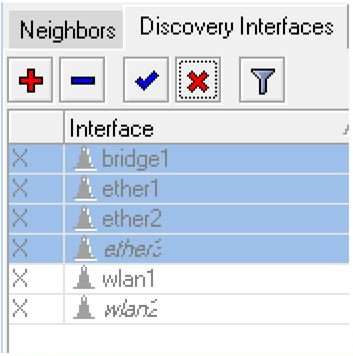
Neighborhood Discovery Protocols attacks Countermeasures

→ Disable MNDP for all interfaces

→Even with MNDP disabled, the traffic generated by such type of attack will be present and can cause performance problems. To block UDP port 5678 at all Bridge Fiters will drop this traffic.

→ Remember that each ethernet-like interface (EoIP, IPIP, static PPtP, etc) has MNDP enabled by default.





Attacking Layer 2

DHCP Starvation

DHCP Basics

DHCP runs in 4 steps:

1) The Client tries to find a DHCP server in his physical network segment

DHCP Discovery

Src-mac=<mac_do_cliente>, dst-mac=<broadcast>, protocolo=udp, src-ip=0.0.0.0:68, dst-ip=255.255.255.255:67

2) DHCP server offers (and reserves for a time) on IP address

DHCP Offer

Src-mac=<mac_do_DHCP-server>, dst-mac=<broadcast>, protocolo=udp, src-ip=<ip_do_DHCP-server>:68, dst-ip=255.255.255.255:67

DHCP Basics

3) The Client accepts the IP

DHCP Request

Src-mac=<mac_do_cliente>, dst-mac=<broadcast>, protocolo=udp, src-ip=0.0.0.0:68, dst-ip=255.255.255.255:67

4) The Server acknowledges the IP for the Client

DHCP Acknowledgment

Src-mac=<mac_do_DHCP-server>, dst-mac=<broadcast>, protocolo=udp, src-ip=<ip_do_DHCP-server>:68, dst-ip=255.255.255.255:67

DHCP Starvation

There are 2 types of DHCP starvation attack:

1) The attacker generates tons of DHCP and follow all steps getting all IP's available

2) Tha attacker generates tons of DHCP discovery packets but doesn't confirm them

Both technicques use random MAC addresses and can cause Denial of Service by means of consuming all availables IP's. The first attack is slower and persistent and the second one is faster and more volatile. The attacker's choice is based on which kind of damage he/she want to cause to the network.

DHCP Starvation

DHCP | Networks | Leases | Options | Alerts

Make Static | Check Status

	Address	Active Address	Active MAC Addre...	Active Hos...	Expires After	Status
D		172.16.1.250	00:16:D3:AD:25:F5	maia	2d 23:52:49	bound
D		172.16.1.254	3E:4D:E3:25:AC:95		00:00:20	offered
D		172.16.1.253	84:F3:C5:10:E6:F5		00:00:20	offered
D		172.16.1.252	80:FE:45:49:DC:30		00:00:20	offered
D		172.16.1.251	38:52:B0:3B:92:99		00:00:20	offered
D		172.16.1.249	9A:7F:69:51:0A:52		00:00:20	offered
D		172.16.1.248	E4:B1:FE:7B:FB:1D		00:00:20	offered
D		172.16.1.247	F2:B1:5C:36:B9:37		00:00:20	offered
D		172.16.1.246	FA:F6:79:0F:D8:09		00:00:20	offered
D		172.16.1.245	64:3B:C6:4B:D0:6E		00:00:20	offered

...

D		172.16.1.228	AA:76:E5:24:4B:9E		00:00:18	offered
D		172.16.1.227	D8:FD:2A:44:E7:27		00:00:18	offered
D		172.16.1.226	60:AE:2C:74:9F:FE		00:00:18	offered
D		172.16.1.225	74:6D:FF:1F:19:05		00:00:18	offered
D		172.16.1.224	18:87:80:08:CD:AC		00:00:18	offered
D		172.16.1.223	58:DF:F2:40:D1:1D		00:00:18	offered
D		172.16.1.222	EA:8B:DC:28:DA:...		00:00:18	offered
D		172.16.1.221	[illegible]		[illegible]	[illegible]

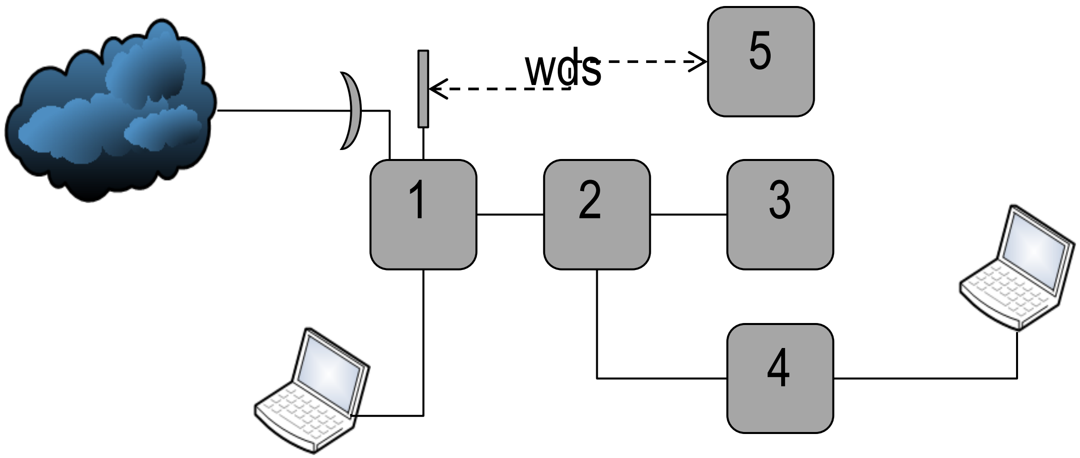
253 items

→ The attacker sends dhcp discovery packets using random MAC address and the server reserves IP's from its pool.

→ With the server without IP resources, the attacker can launch a Rogue DHCP server to catch users to his own IP, gateway and DNS configurations.

Less than 5 seconds can exhaust an entire Class C

DHCP Starvation
DEMO



- Launching the attack from 4
- Seeing the efects at 1 (DHCP Server)

DHCP Starvation Countermeasures

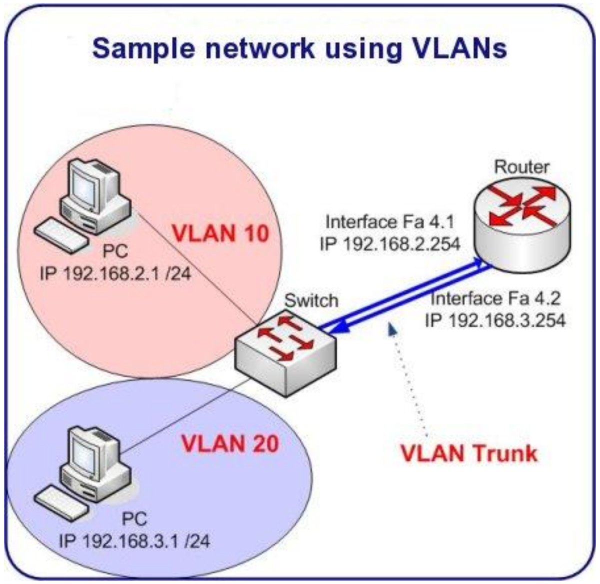
→ Appropriate Bridge Filter rules accepting only known MAC's

→ Use of static Leases at the DHCP Server

→ Radius o User Manager could be helpful

Atacking Layer 2

Exploiting Vlan´s



VLAN´s

A Vlan is a group of hosts with a common set of requirements that can communicate as if they were attached to the same broadcast domain regardless of their physical location. Vlans are usually used to:

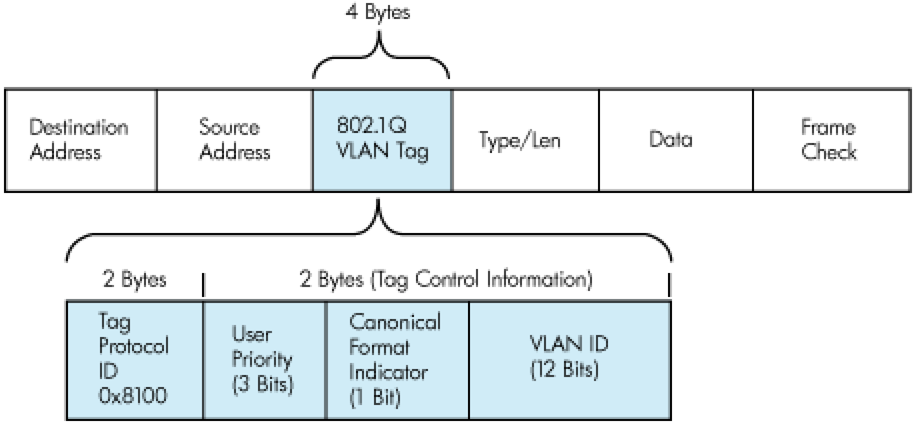
→ To create multiple layer 3 networks over a layer 2 structure.

→ To split traffic and broadcast domains limitation.

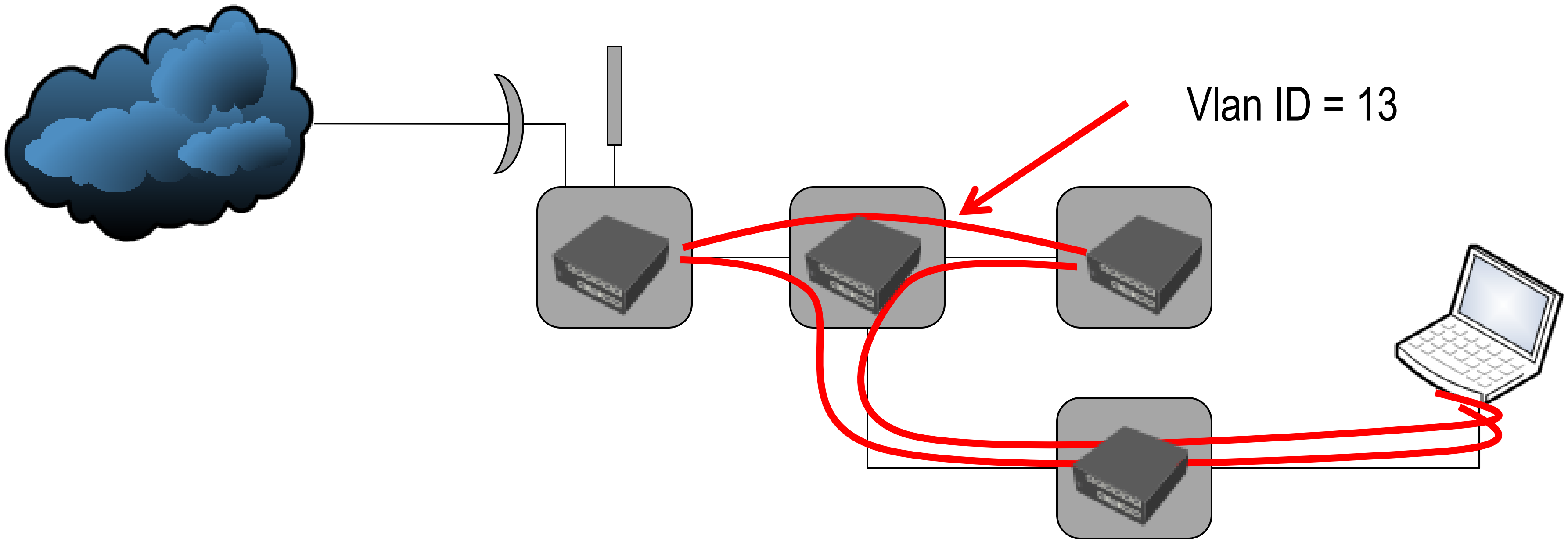
→ To apply particular QoS rules

→ To improve Security (?)

→ etc



Exploiting VLAN´s (802.1q)

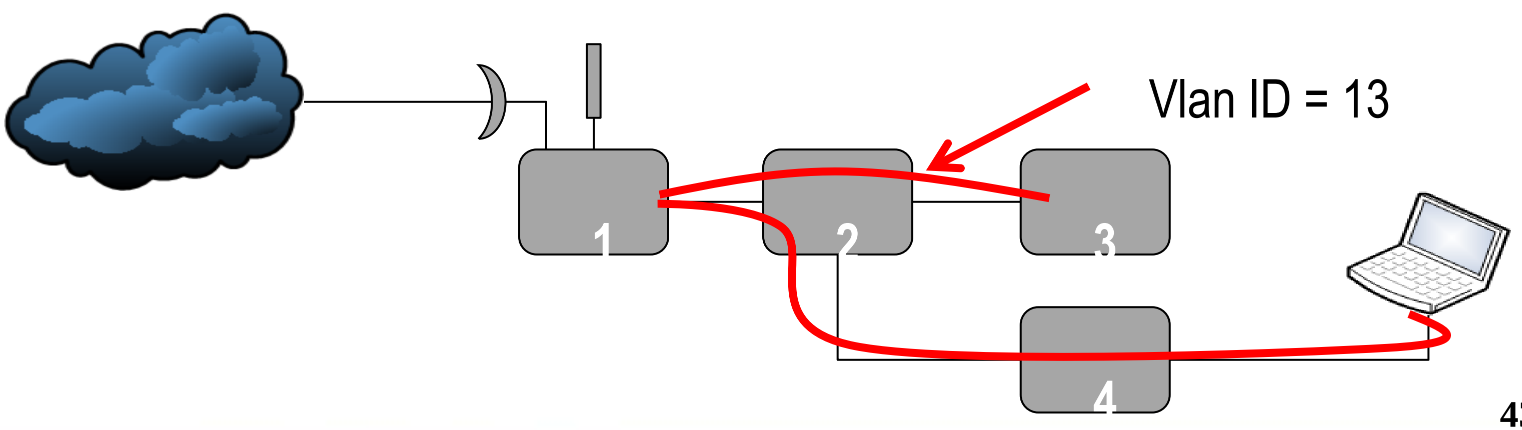


→ The first weakness is obvious – without proper protection any host with the same Vlan ID will participate on the Vlan group.

Exploiting VLAN´s

→ Vlan Proxy attack

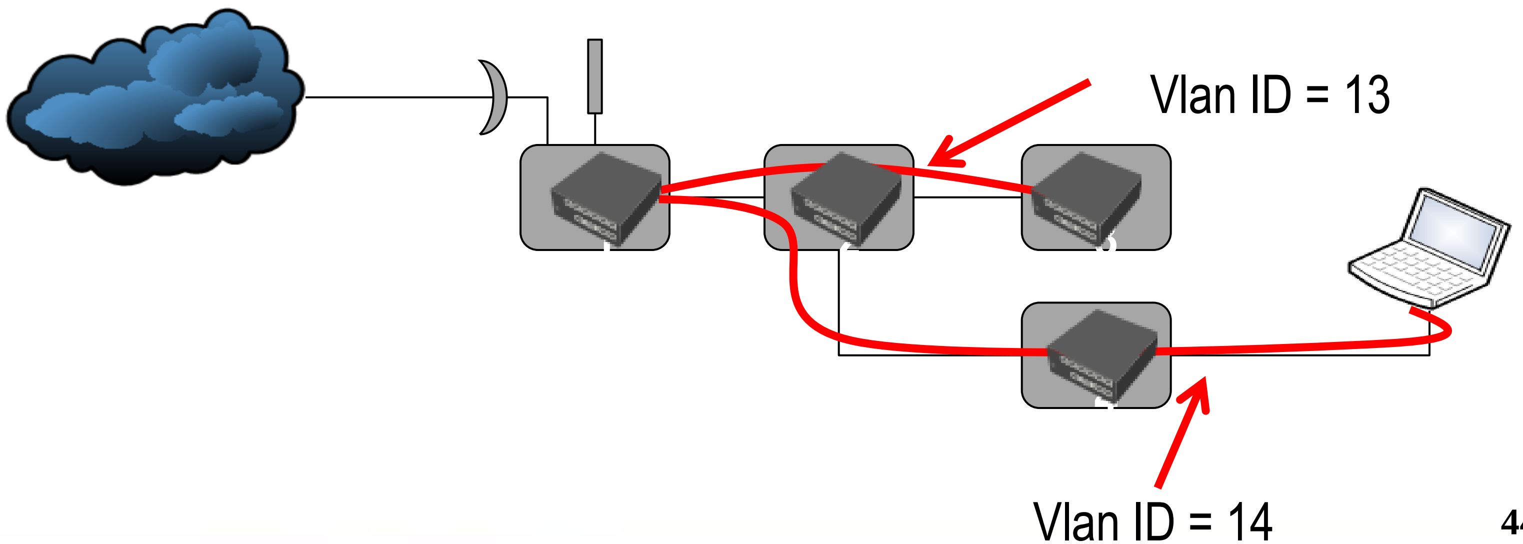
- Attacker sends a packet with his/her IP and MAC (4) as source, destination IP the victim (3) and destination MAC of the router (1) (usually the promiscuous port)
- The Router re-write the MAC and sends the packet to victim (3)
- This network attack works only for unidirectional traffic



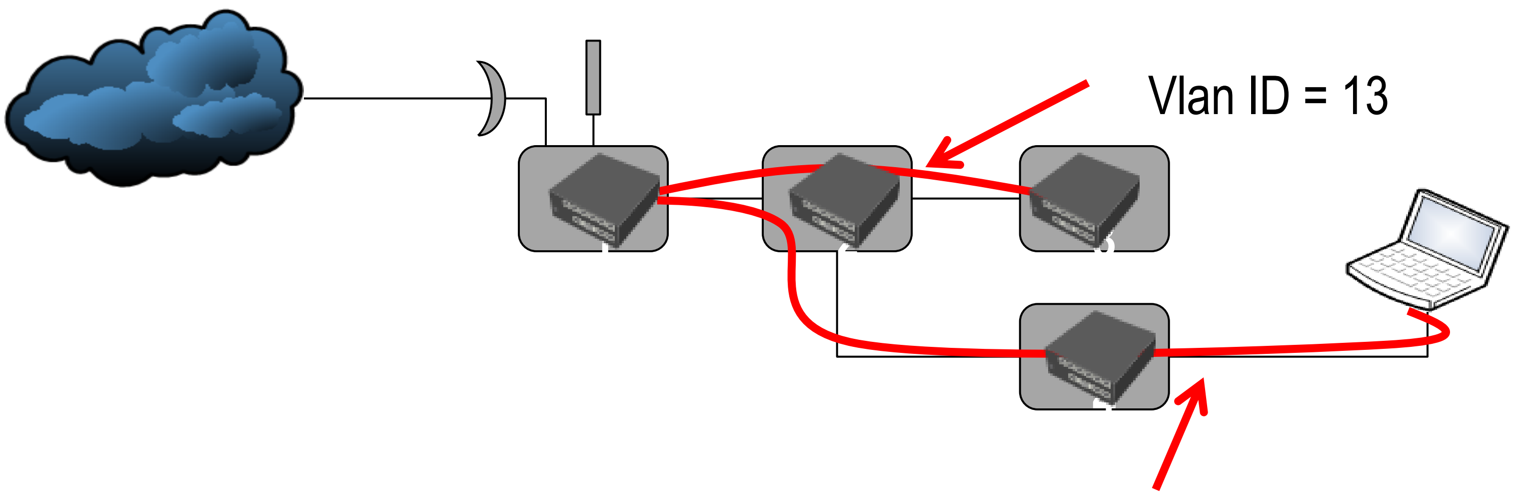
Exploiting VLAN´s

→ Vlan double tagging attack

- The attacker forms a packet with Vlan Tag ID = 13 (target victim) encapsulated with Vlan Tag ID = 14 (his/her segment)
- The switch (bridge) removes the Tag 14 and sends packet to Vlan 13
- Unidirectional attack.

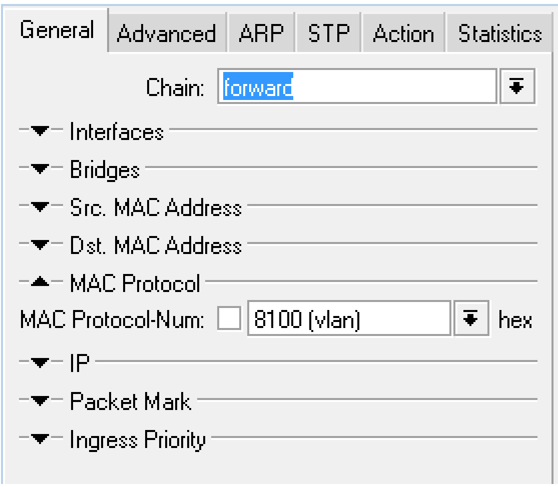


Vlan´s Explotation DEMO



- Vlan proxy attack
- Double tagging attack
- Limiting Vlan access

Exploiting VLAN´s Countermeasures



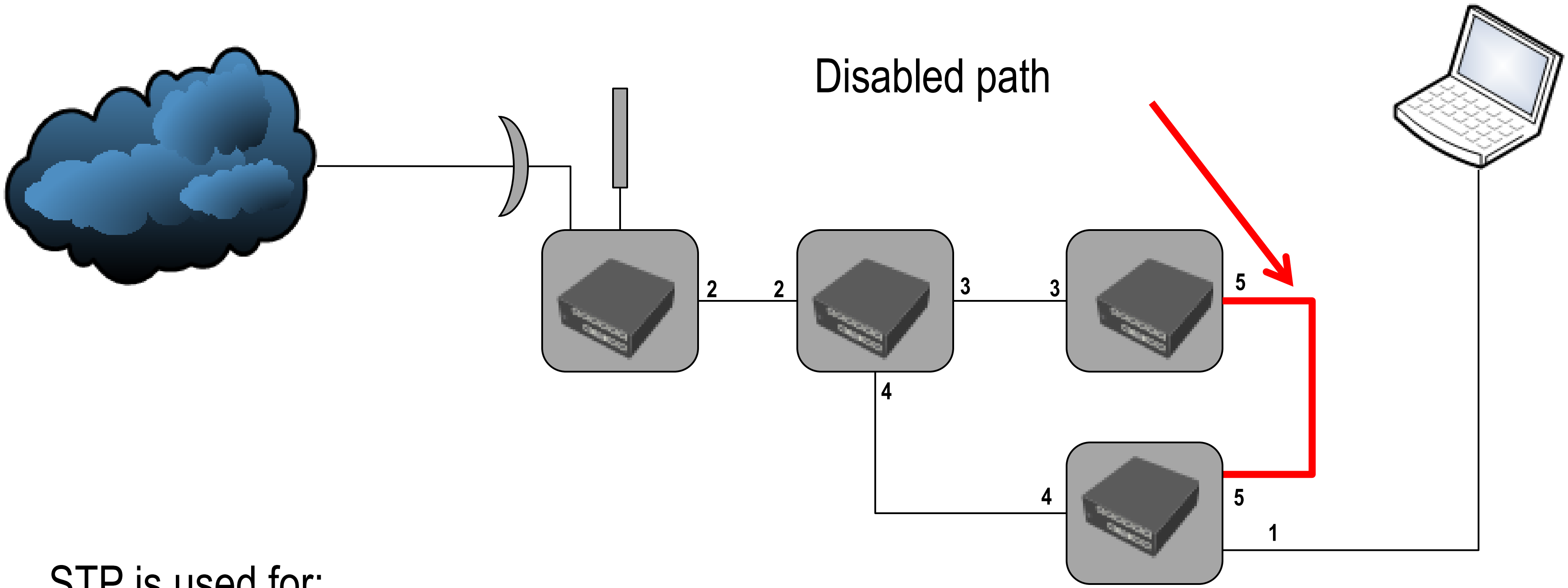
→ Blocking MAC protocol 8100 at all external ports that do not use a Vlan can prevent a attacker manually configure his/her device to participate on a Vlan.

→ Vlan proxy attacks and double tagging attacks from unknown clients could be avoid only by means of access control lists for all external ports. Legitimate clients could however deploy such type of attack.

Layer 2 attacks

Exploiting Spanning Tree Protocol

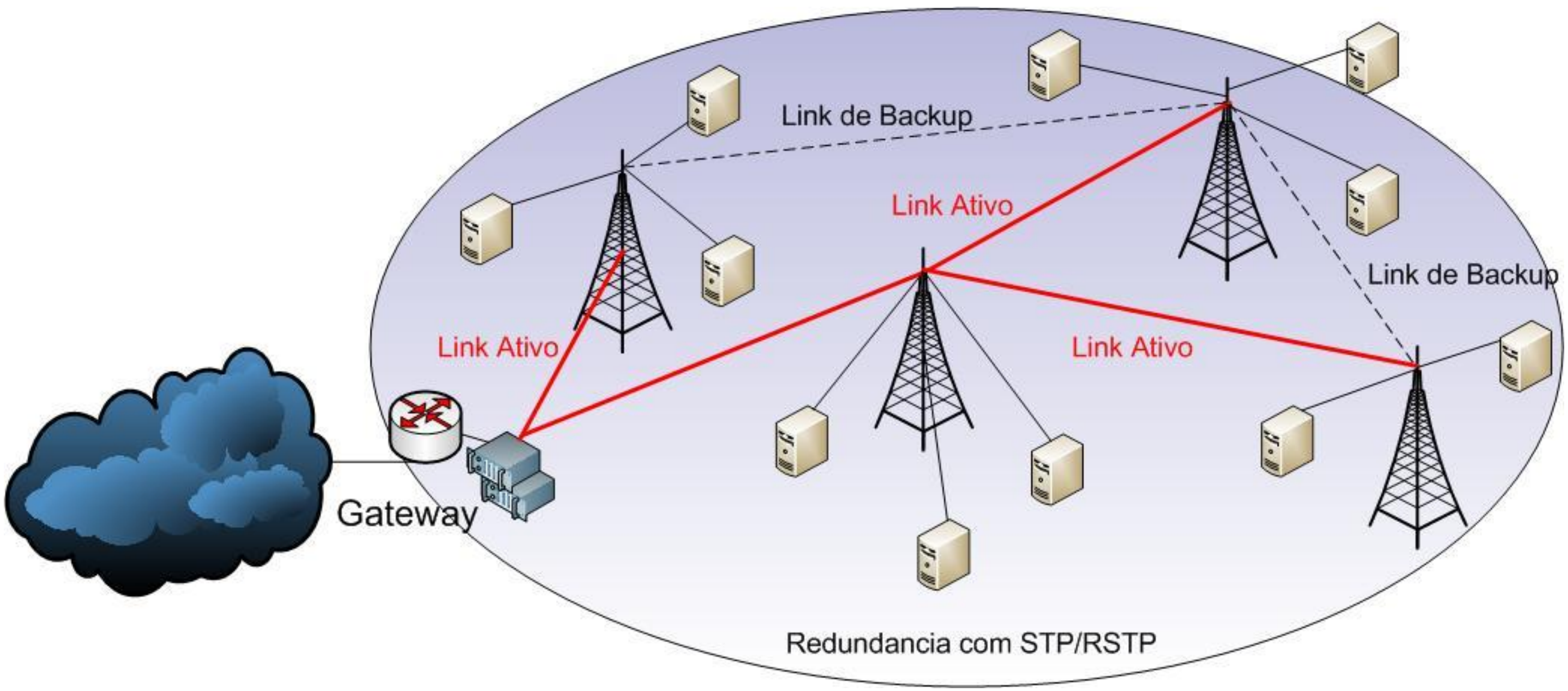
Spanning Tree applications



STP is used for:

→ to avoid looping in Bridged Networks with multiple path.
→ to provide redundancy when an active path goes down.

Spanning Tree applications



Spanning Tree x Rapid Spanning Tree (RSTP)

→RSTP was proposed by IEEE 802.1w in order to provide faster responses when adapting the network to topology changes

→RSTP works watching port states that can be:

- → Unknown (not yet determined)
- → Alternate (not part of the current active topology – backup)
- → Designated (the port is designated for a connected LAN)
- → Root (path to the root Bridge)

→ RSTP is much faster than STP, but they are fully compatible.

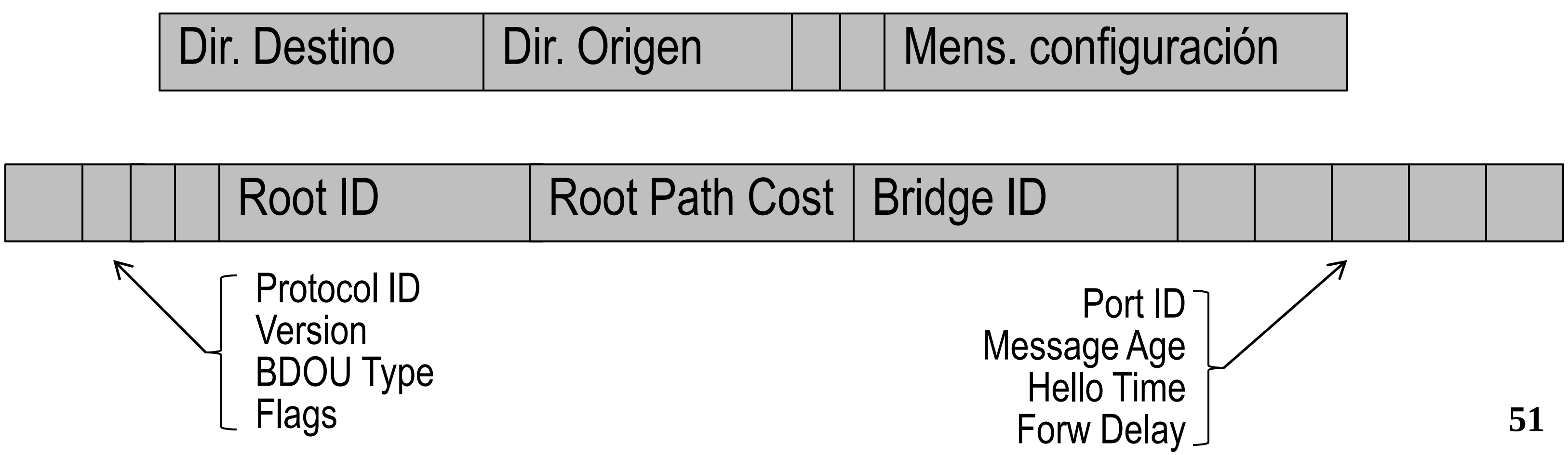
(R)STP Basics

→ The Spanning Tree Protocol elect among all the participating Bridges one Root Bridge (usually the lower Bridge ID)

→ Each device computes the shortest path from itself to the Root Bridge.

→ Each Bridge has a Root Port, where the communication to de Root Bridge is made.

→ All devices exchange BPDU (Bridge Protocol Data Unit) messages



(R)STP Basics

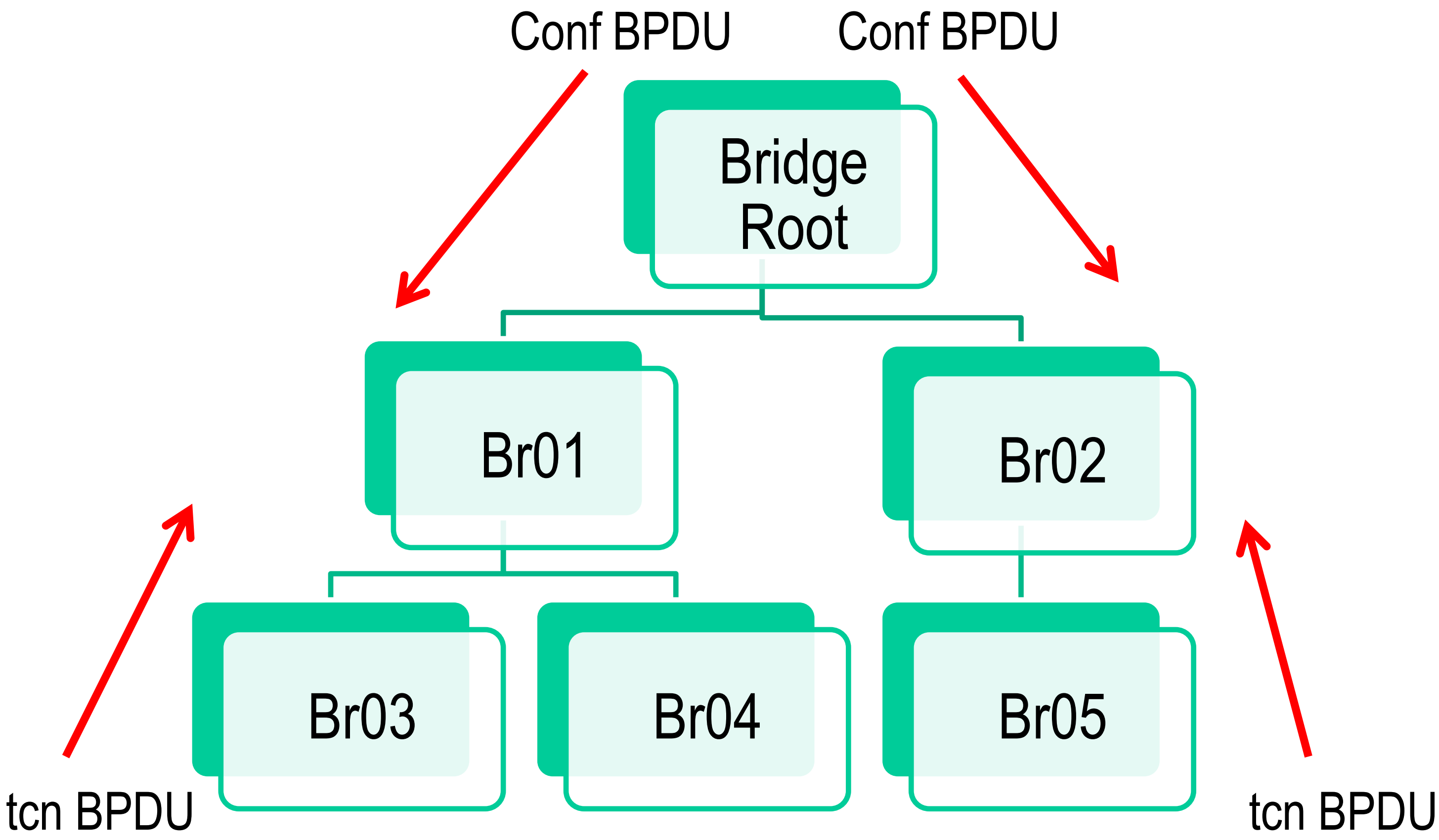
→ The Root Bridge periodically announces configuration messages to all other Bridges named **conf BPDU** (Configuration BPDU) with its source MAC address.

→ If topology changes at any network segment, the responsible Bridge for this segment sends messages telling about such modification. Such messages are named **tcn BPDU** – (Topology Change Notification BPDU)

				Root ID	Root Path Cost	Bridge ID					

Protocol ID	Version	Mes. Type

(R)STP Basics


Conf BPDU
Conf BPDU
Bridge Root
Br01
Br02
Br03
Br04
Br05
tcn BPDU
tcn BPDU

Exploiting STP and RSTP

Both STP and RSTP are wide open for attacks because there is no authentication in BPDU messages.

For this reason anyone that has access to Layer 2 can explore STP to launch DoS or MitM attacks

→ conf BPDU messages Flooding for DoS attacks
→ tcn BPDU messages Flooding for DoS attacks
→ Impersonating the Root Bridge by flooding conf BPDU mesages
→ Man-in-the-middle attack when having access to 2 bridges

Attacking (Rapid) Spanning Tree

→ Attacker sending conf BPDU message

firewall info	input: in:ether1 out:(none), src-mac 04:08:20:12:a9:75, dst-mac 01:80:c2:00:00:00, eth-proto 0026

→ Attacker sending tcn BPDU message

firewall info	input: in:ether1 out:(none), src-mac 04:08:20:12:a9:75, dst-mac 01:80:c2:00:00:00, eth-proto 0007

→ DoS attack based on tons of conf BPDU messages

firewall info	input: in:ether1 out:(none), src-mac 56:ea:a5:15:3e:6f, dst-mac 01:80:c2:00:00:00, eth-proto 0026
firewall info	input: in:ether1 out:(none), src-mac d2:50:ed:1e:48:31, dst-mac 01:80:c2:00:00:00, eth-proto 0026
firewall info	input: in:ether1 out:(none), src-mac 42:60:5b:79:2b:d4, dst-mac 01:80:c2:00:00:00, eth-proto 0026
firewall info	input: in:ether1 out:(none), src-mac 20:68:54:01:d9:1a, dst-mac 01:80:c2:00:00:00, eth-proto 0026
firewall info	input: in:ether1 out:(none), src-mac 18:f1:3a:59:72:0a, dst-mac 01:80:c2:00:00:00, eth-proto 0026
firewall info	input: in:ether1 out:(none), src-mac f6:89:e0:39:91:44, dst-mac 01:80:c2:00:00:00, eth-proto 0026

→ DoS attack based on tons of tcn BPDU messages

firewall info	input: in:ether1 out:(none), src-mac 82:f0:19:5c:7b:1c, dst-mac 01:80:c2:00:00:00, eth-proto 0007
firewall info	input: in:ether1 out:(none), src-mac d6:d8:2a:50:1e:5c, dst-mac 01:80:c2:00:00:00, eth-proto 0007
firewall info	input: in:ether1 out:(none), src-mac 88:63:b3:6b:18:f1, dst-mac 01:80:c2:00:00:00, eth-proto 0007
firewall info	input: in:ether1 out:(none), src-mac f8:52:21:43:6d:dd, dst-mac 01:80:c2:00:00:00, eth-proto 0007
firewall info	input: in:ether1 out:(none), src-mac 7e:0c:00:23:a5:0f, dst-mac 01:80:c2:00:00:00, eth-proto 0007
firewall info	input: in:ether1 out:(none), src-mac 32:b5:28:36:70:27, dst-mac 01:80:c2:00:00:00, eth-proto 0007

Attacking (Rapid) Spanning Tree

→ Attacker impersonating Root Bridge

firewall info	input: in:ether1 out:(none), src-mac 00:0c:42:03:04:04, dst-mac 01:80:c2:00:00:00, eth-proto 0026
firewall info	input: in:ether1 out:(none), src-mac 00:0c:42:03:04:04, dst-mac 01:80:c2:00:00:00, eth-proto 0026
firewall info	input: in:ether1 out:(none), src-mac 00:0c:42:03:04:04, dst-mac 01:80:c2:00:00:00, eth-proto 0026
firewall info	input: in:ether1 out:(none), src-mac 00:0c:42:03:04:04, dst-mac 01:80:c2:00:00:00, eth-proto 0026

Bridge | Ports | Filters | NAT | Hosts

Find

	Interface	Bridge	Priority (h...	Path Cost	Horizon	Role	Root Pat...
	ether1	bridge1	80	10		designated port	
I	ether2	bridge1	80	10		disabled port	
I	ether3	bridge1	80	10		disabled port	
	ether4	bridge1	80	10		root port	10
I	ether5	bridge1	80	10		disabled port	

Bridge | Ports | Filters | NAT | Hosts

Find

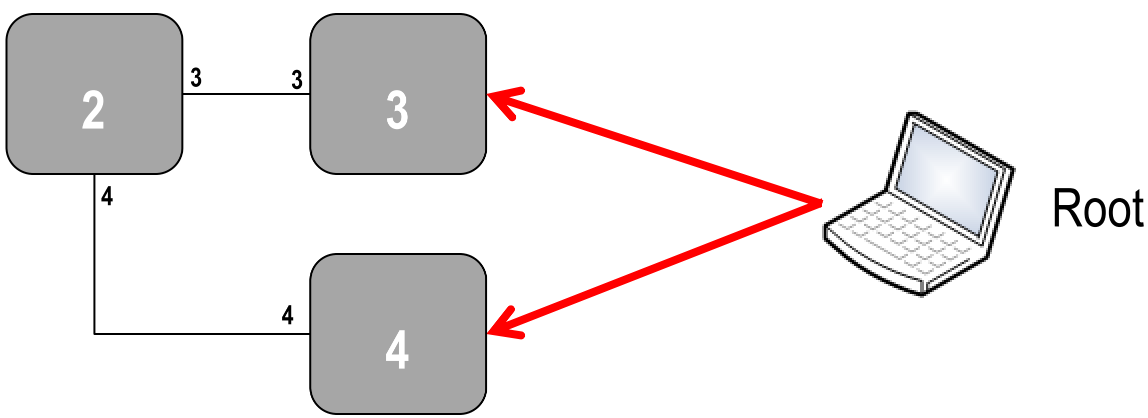
	Interface	Bridge	Priority (h...	Path Cost	Horizon	Role	Root Pat...
	ether1	bridge1	80	10		root port	20
I	ether2	bridge1	80	10		disabled port	
I	ether3	bridge1	80	10		disabled port	
	ether4	bridge1	80	10		designated port	
I	ether5	bridge1	80	10		disabled port	

Attacking (Rapid) Spanning Tree

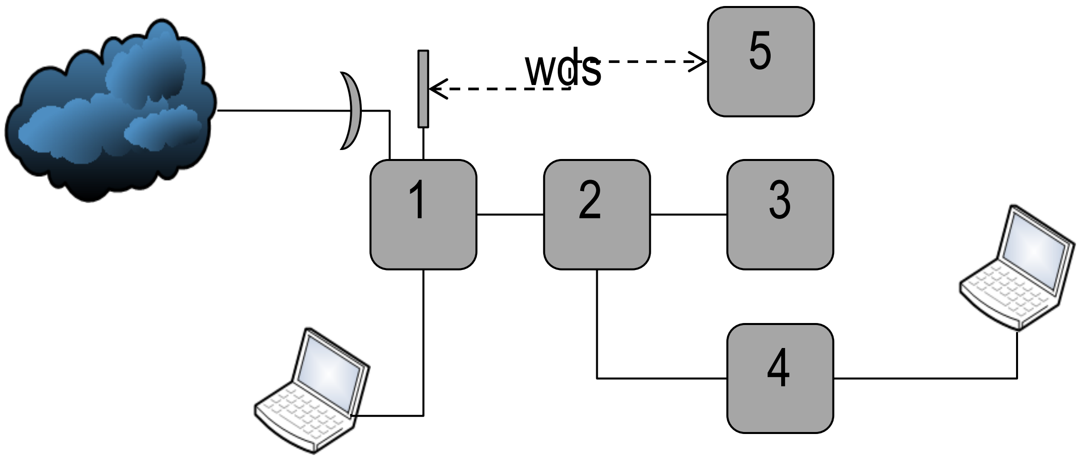
→ Attacker joining the (R)STP network

firewall info	input: in:ether1 out:(none), src-mac 00:0c:42:05:04:04, dst-mac 01:80:c2:00:00:00, eth-proto 0026
firewall info	input: in:ether1 out:(none), src-mac 00:0c:42:05:04:04, dst-mac 01:80:c2:00:00:00, eth-proto 0026
firewall info	input: in:ether1 out:(none), src-mac 00:0c:42:05:04:04, dst-mac 01:80:c2:00:00:00, eth-proto 0026
firewall info	input: in:ether1 out:(none), src-mac 00:0c:42:05:04:04, dst-mac 01:80:c2:00:00:00, eth-proto 0026
firewall info	input: in:ether1 out:(none), src-mac 00:0c:42:05:04:04, dst-mac 01:80:c2:00:00:00, eth-proto 0026

→ Attacker impersonating Root Bridge + MitM



Attacking (Rapid) Spanning Tree
DEMO



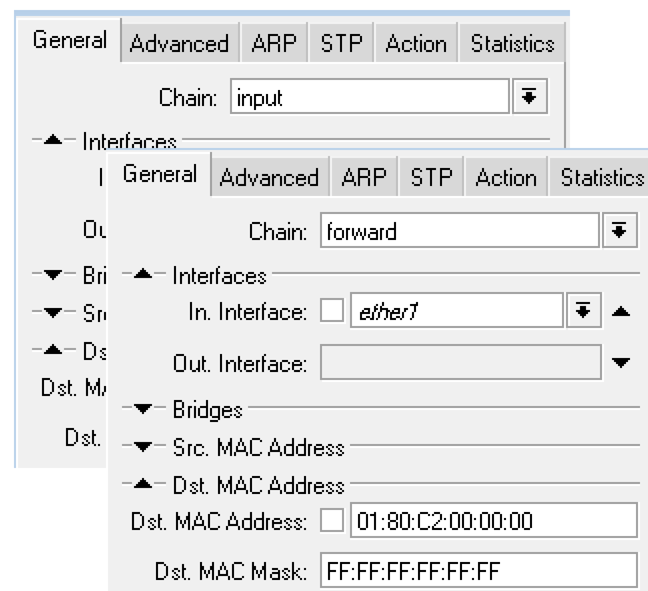
- DoS with conf and or tcn BPDU
- Joining STP Network
- Changing the Root port for one Bridge

Attacking (Rapid) Spanning Tree Countermeasures

Spanning Tree messages by default are sent to the MAC address below:

01:80:C2:00:00:00 .

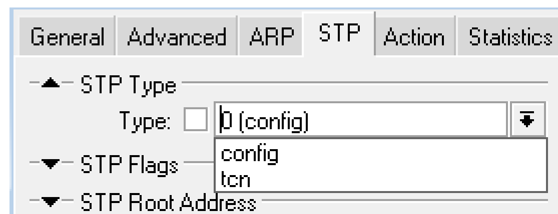
→ Filtering This MAC on border Bridges/Ports on both input and forward channels can avoid such attacks



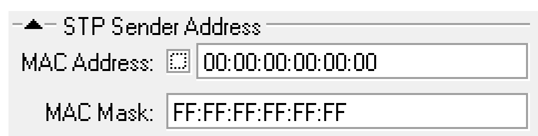
Attacking (Rapid) Spanning Tree Countermeasures

The Bridge Filter feature of Mikrotik RouterOS provide means to selectively filter BPDU messages using the classifiers:

→STP message type (conf BPDU or tcn BPDU)



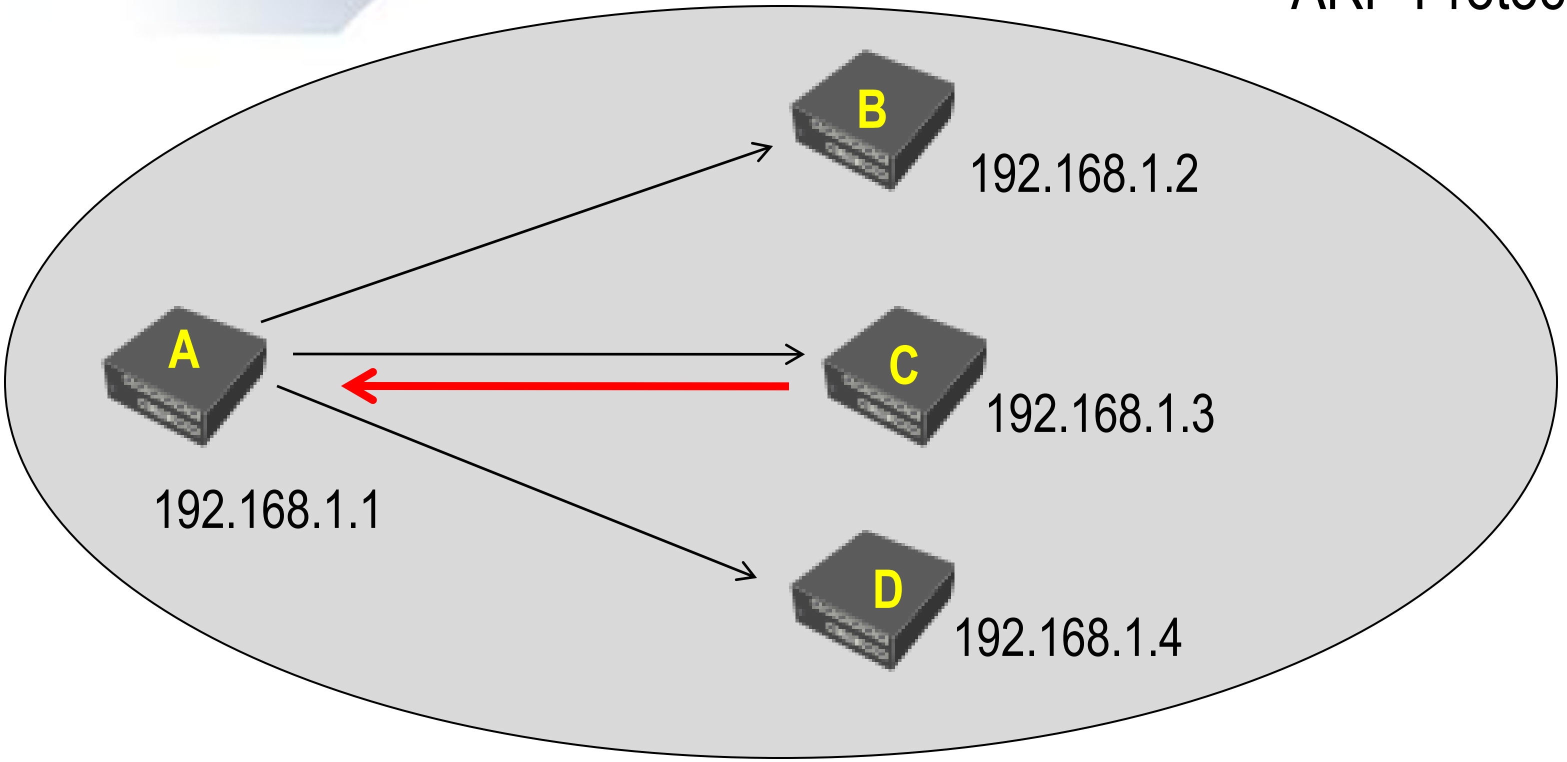
→ Sender MAC address



Layer 2 attacks

ARP Poisoning or ARP Spoof

ARP Protocol - Basics



→ **A asks all hosts:** "Who has the IP 192.168.1.3 ?"

→ **C answers to A:** "The IP 192.168.1.3 is on MAC CC:CC:CC:CC:CC:CC"

→ **A register in its arp table the pair:** 192.168.1.3, MAC CC:CC:CC:CC:CC:CC

ARP Poisoning

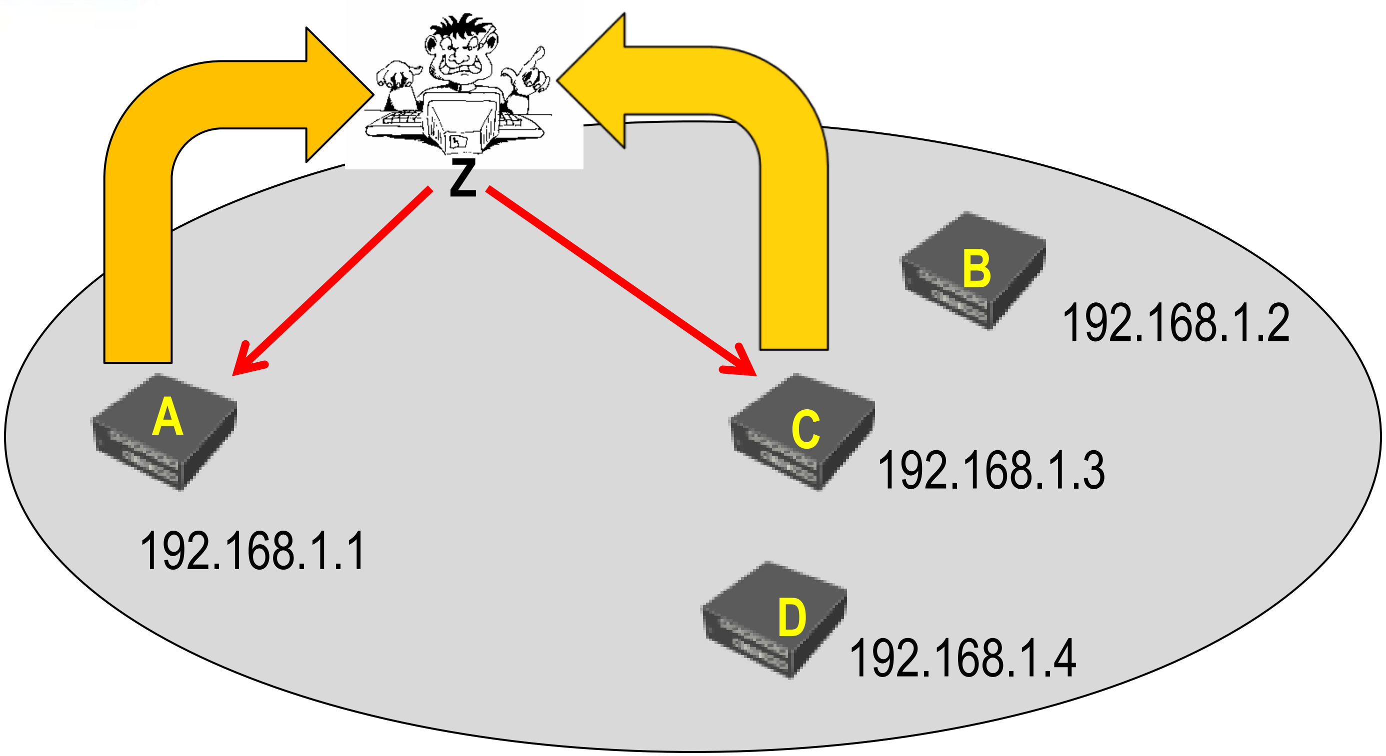
→ The attacker sends to a specific target host or to all hosts of the network, “gratuitous” arp messages saying that his MAC is the MAC belonging to whom he/she wants to spoof (usually the main gateway)

→ The victim or victims has their ARP tables poisoned and whenever they want to communicate through the gateway actually they send the packets to the attacker

→ The attacker sends to the gateway “gratuitous” arp messages announcing his MAC address as the MAC belonging to the victim.

→ Bidirectional attack is running now and all traffic from an to the victim could be sniffed/changed by the attacker.

ARP Poisoning

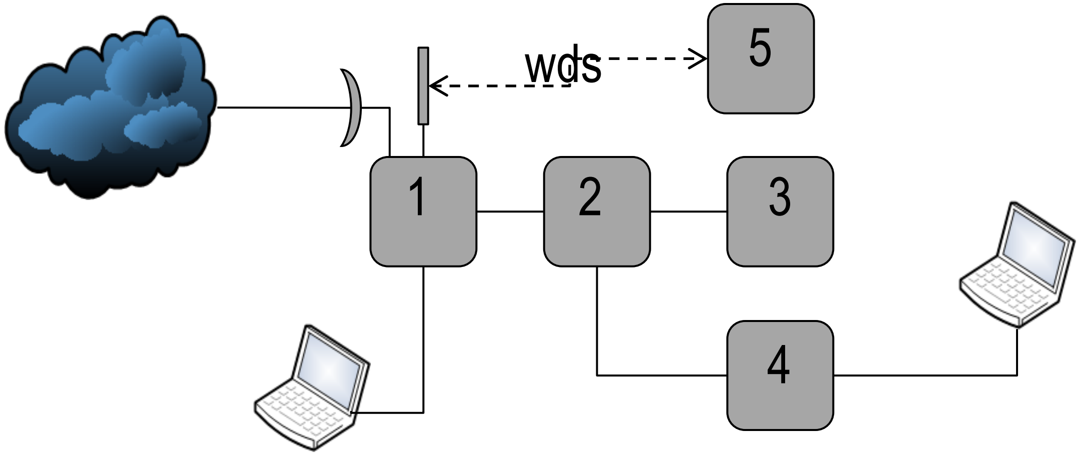


→ **Z says to A:** "The IP 192.168.1.3 is at MAC ZZ:ZZ:ZZ:ZZ:ZZ:ZZ"

→ **Z says to C:** "The IP 192.168.1.1 is at MAC ZZ:ZZ:ZZ:ZZ:ZZ:ZZ"

→ **A talk to C (and vice versa) through Z (Man-in-the-Middle)**

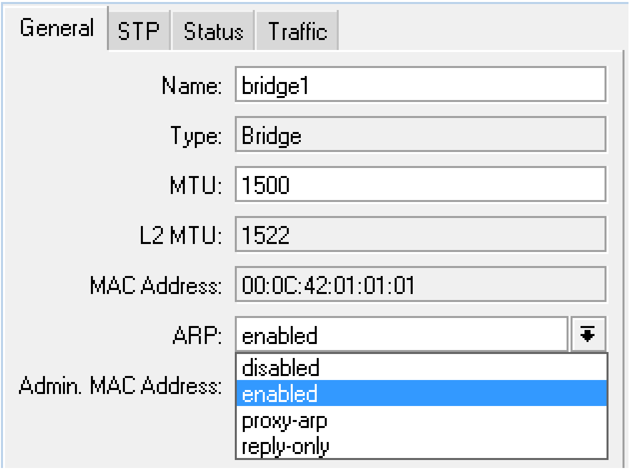
ARP Spoofing DEMO



- launching arp spoof attack from 4
- Checking on all other hosts
- Filtering the ARP

Arp Spoofing Countermeasures

1) Changing ARP protocol behavior



ARP disabled → all hosts must have ARP static entry's

ARP Reply-Only → In case of a multipoint system (e.g an Access Point), only the concentrator must have static entry's

Problems:

→ Static Arp in all hosts is a hard administrative task.

→ Reply-Only doesn't protect client side – Unidirectional attack is trivial. (Bidirectional requires a little bit more hacking ☺).

Arp Poisoning Countermeasures

2) Traffic isolation at layer 2

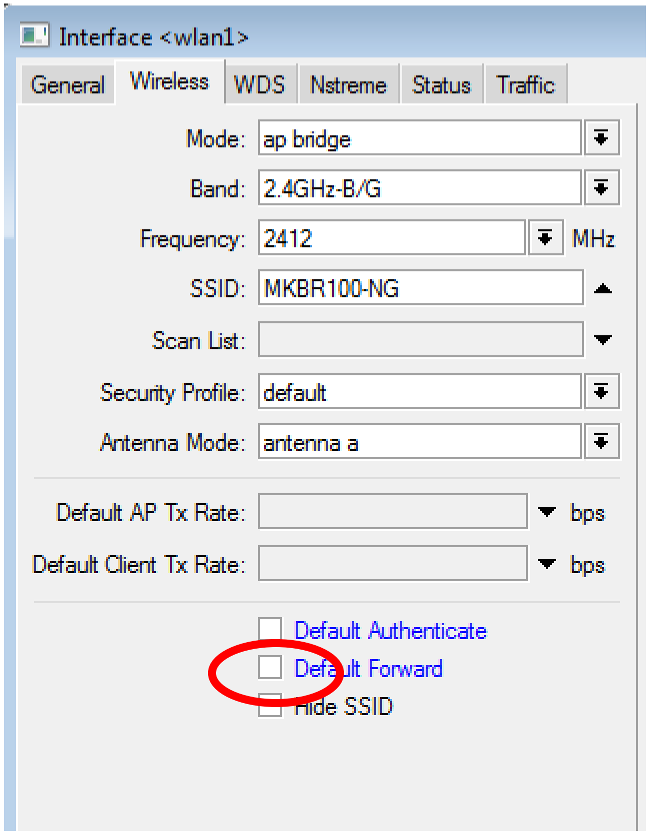
Considering a typical WISP network, the only valid traffic flow is from the client to the gateway and from the gateway to the clients. Ensuring only this flow is allowed We can thwart arp poisoning techniques because no one client will “see” the other.

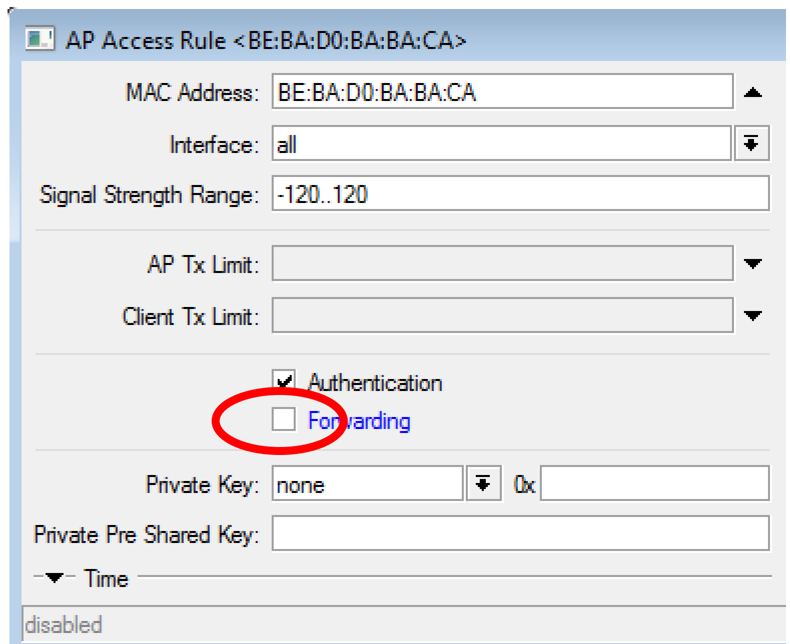
When working with Wireless AP, this isolation must be provided in 2 levels

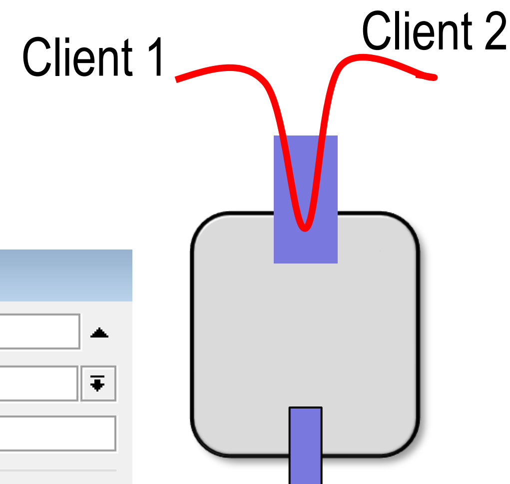
→ Wireless Interface level

→ All Bridged ports, wireless and ethernet

Layer 2 traffic isolation (for all Wireless cards)

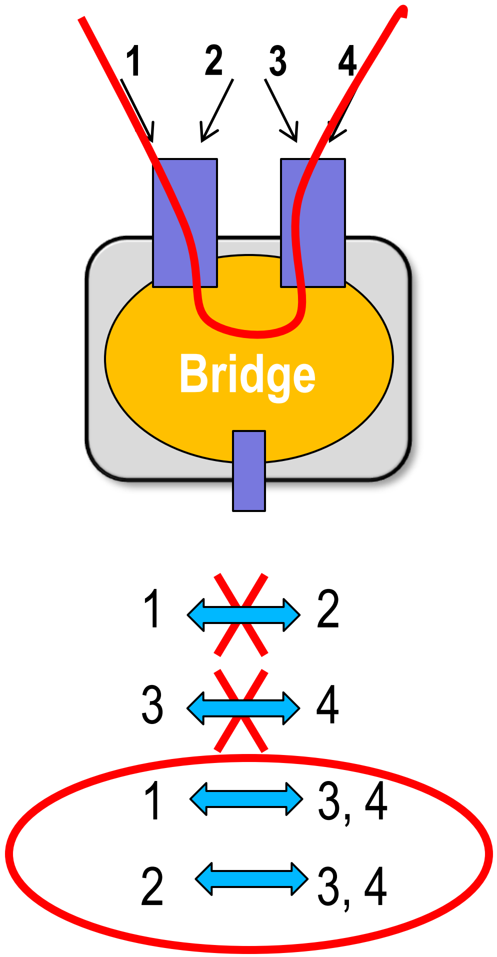


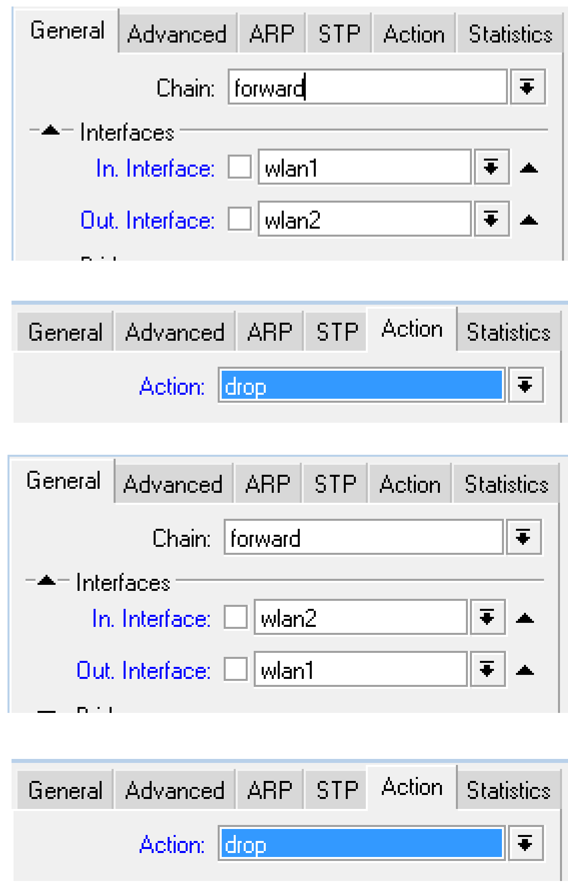




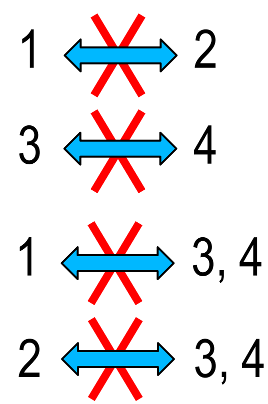
Default forward disabled at interface level and in the access list

Layer 2 traffic isolation
(2 Bridged Wireless card)

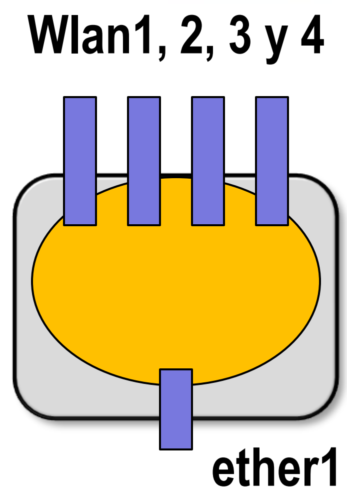




2 Rules



Layer 2 traffic isolation
(4 Bridged Wireless card)



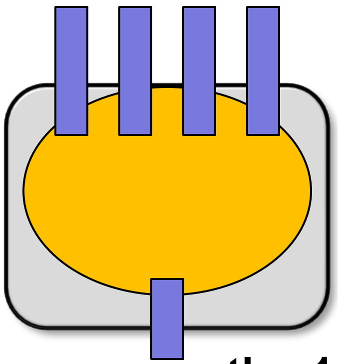
12 Rules?

Bridge | Ports | Filters | Broute | NAT | Hosts

Reset C

#	Chain	Interfaces...	Interfaces...
0	forward	wlan1	wlan2
1	forward	wlan2	wlan1
2	forward	wlan1	wlan3
3	forward	wlan3	wlan1
4	forward	wlan1	wlan4
5	forward	wlan4	wlan1
6	forward	wlan2	wlan3
7	forward	wlan3	wlan2
8	forward	wlan2	wlan4
9	forward	wlan4	wlan2
10	forward	wlan3	wlan4
11	forward	wlan4	wlan3

Layer 2 traffic isolation
(4 Bridged Wireless card)



12 Rules?

Bridge | Ports | Filters | Broute | NAT | Hosts

Reset C

#		Chain	Interfaces...	Interfaces...
0		forward	wlan1	wlan2
1		forward	wlan2	wlan1
2		forward	wlan1	wlan3
3		forward	wlan3	wlan1
4		forward	wlan1	wlan4
5		forward	wlan4	wlan1
6		forward	wlan2	wlan3
7		forward	wlan3	wlan2
8		forward	wlan2	wlan4
9		forward	wlan4	wlan2
10		forward	wlan3	wlan4
11		forward	wlan4	wlan3

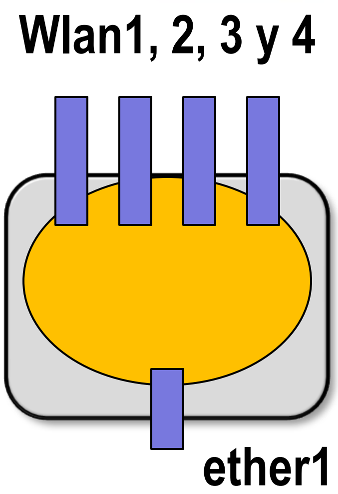
4 Rules

Bridge | Ports | Filters | Broute | NAT | Hosts

Reset

#		Chain	Interfaces...	Interfaces...
0		forward	wlan1	!ether1
1		forward	wlan2	!ether1
2		forward	wlan3	!ether1
3		forward	wlan4	!ether1

Layer 2 traffic isolation
(4 Bridged Wireless card)



12 Rules?

Bridge | Ports | Filters | Broute | NAT | Hosts

#	Chain	Interfaces...	Interfaces...
0	forward	wlan1	wlan2
1	forward	wlan2	wlan1
2	forward	wlan1	wlan3
3	forward	wlan3	wlan1
4	forward	wlan1	wlan4
5	forward	wlan4	wlan1
6	forward	wlan2	wlan3
7	forward	wlan3	wlan2
8	forward	wlan2	wlan4
9	forward	wlan4	wlan2
10	forward	wlan3	wlan4
11	forward	wlan4	wlan3

4 Rules

Bridge | Ports | Filters | Broute | NAT | Hosts

#	Chain	Interfaces...	Interfaces...
0	forward	wlan1	!ether1
1	forward	wlan2	!ether1
2	forward	wlan3	!ether1
3	forward	wlan4	!ether1

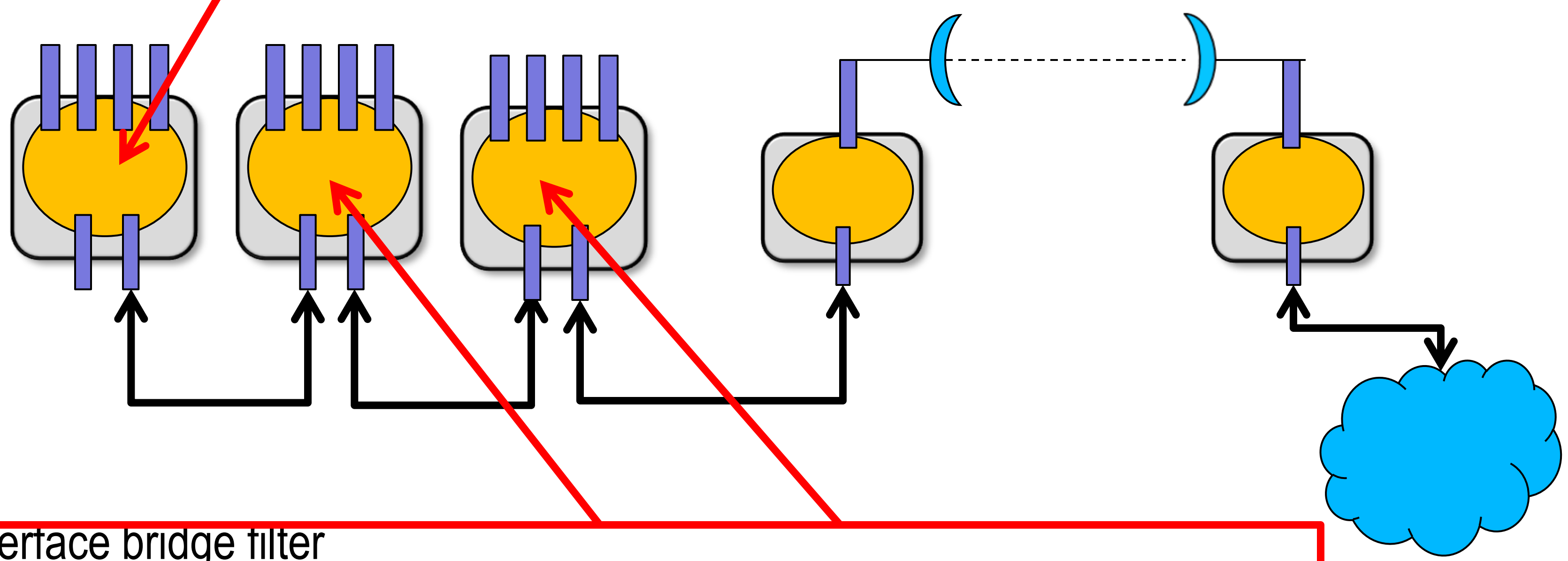
1 Rule !

Bridge | Ports | Filters | Broute | NAT | Hosts

#	Chain	Interfaces...	Interfaces...
0	forward	!ether1	!ether1

Layer 2 traffic isolation
(many Bridged equipments)

```
/interface bridge filter
add  chain=forward in-interface=!ether2
out-interface=!ether2  action=drop
```

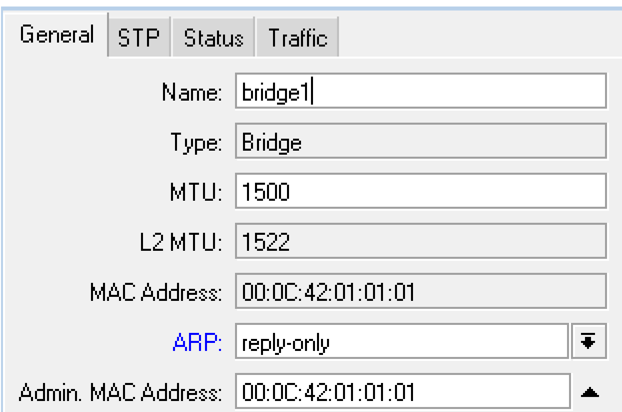


```
/interface bridge filter
add  chain=forward in-interface=ether1  out-interface=ether2 action=accept
add  chain=forward in-interface=ether2  out-interface=ether1 action=accept
add  chain=forward in-interface=!ether2 out-interface=!ether2  action=drop
```

Arp Spoofing Countermeasures

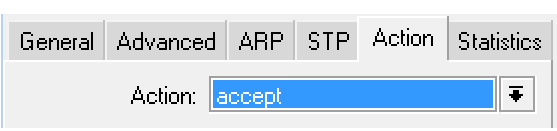
If the Bridged network has equipments without resources for isolation between clients, there is nothing to do but only try to minimize the effects or arp spoofing techniques.

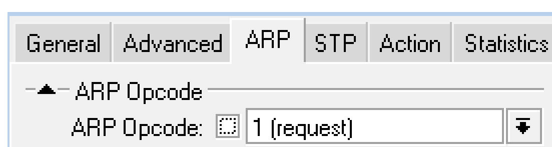
Below are some hints:

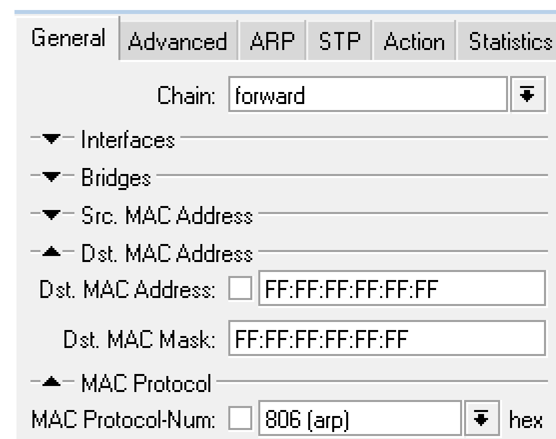


1 – Gateway with reply-only (static tables)

2 – Accepting arp requests from any host

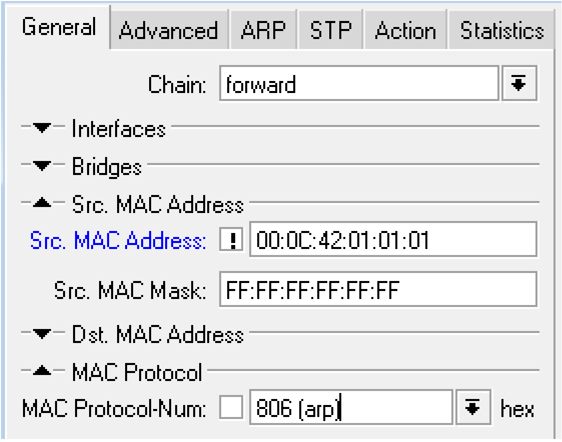


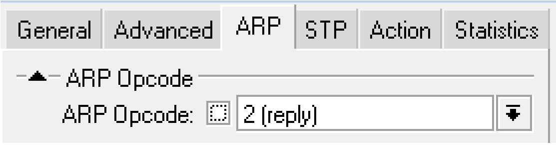




Arp Spoofing Countermeasures

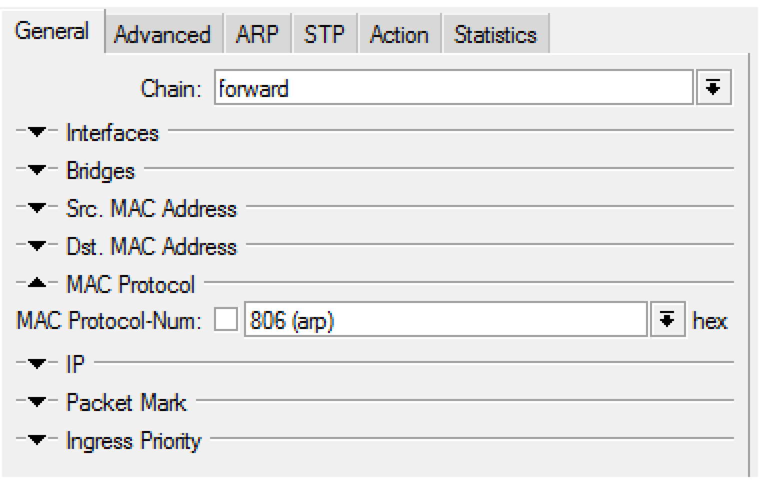
3 – Dropping any reply that has other source than the gateway

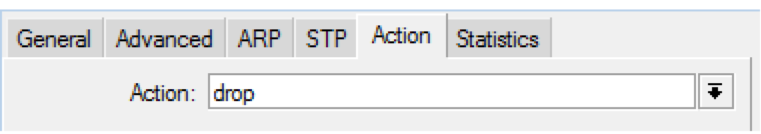


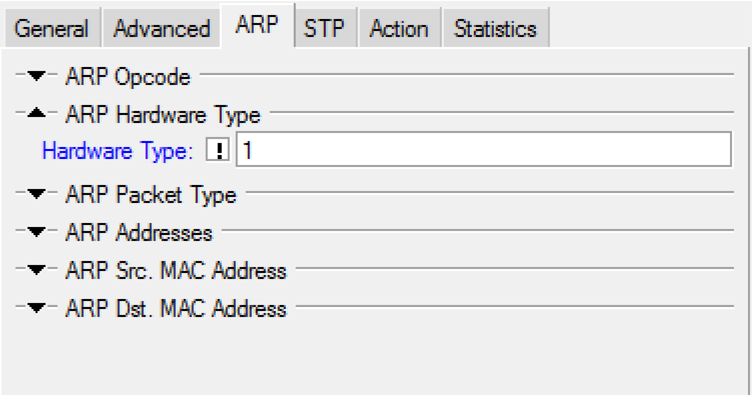


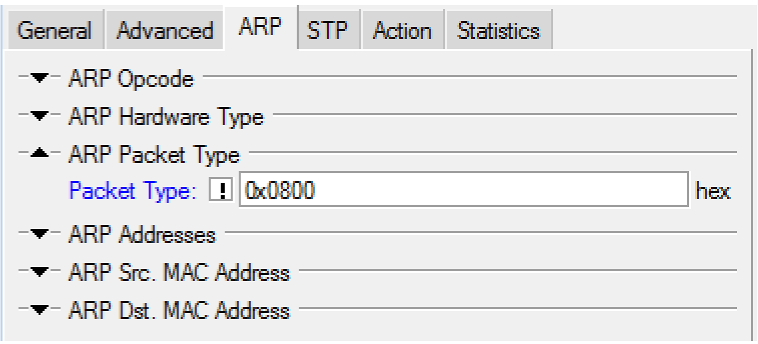
Arp Spoofing Countermeasures
Complementary measures

Is possible to get rid of some “insane traffic” at layer 2, dropping frames there are not Ethernet type or IPV4 traffic.





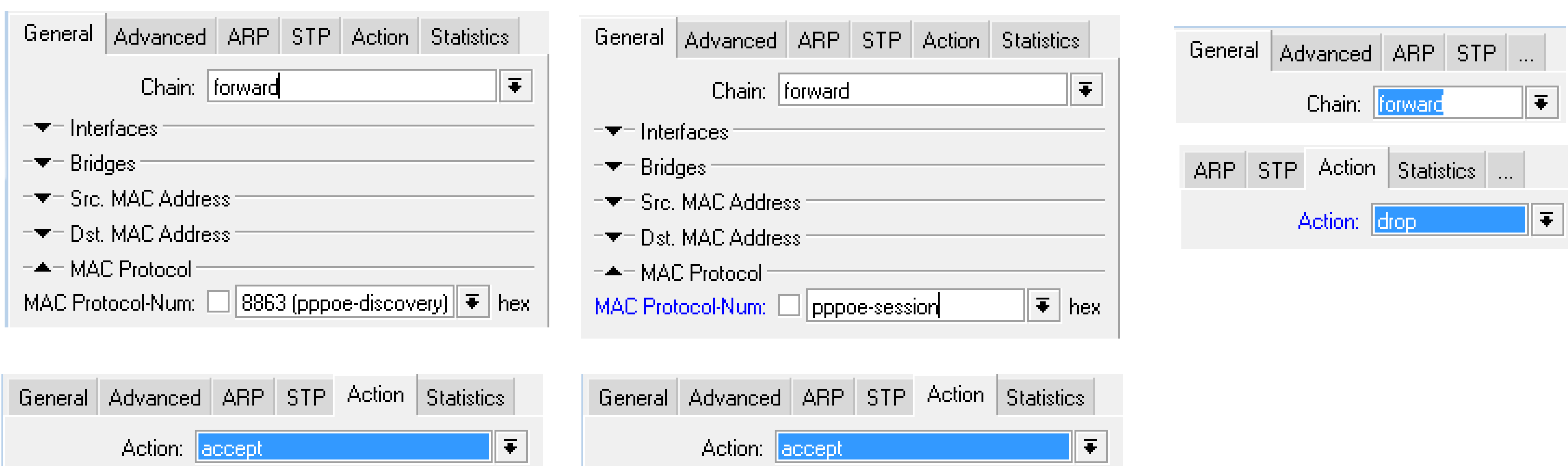




Arp Spoofing Countermeasures PPPoE only networks

→ Disable arp protocol in all interfaces

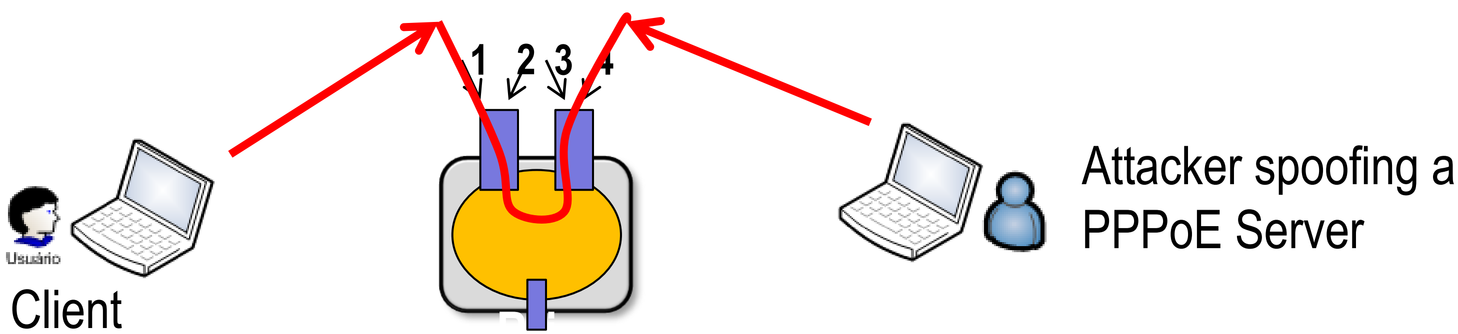
→ Configure Bridge Filters for all PPPoE interfaces accepting only PPPoE-discovery and PPPoE-session and dropping all the rest. This helps to get rid of a lot of useless traffic



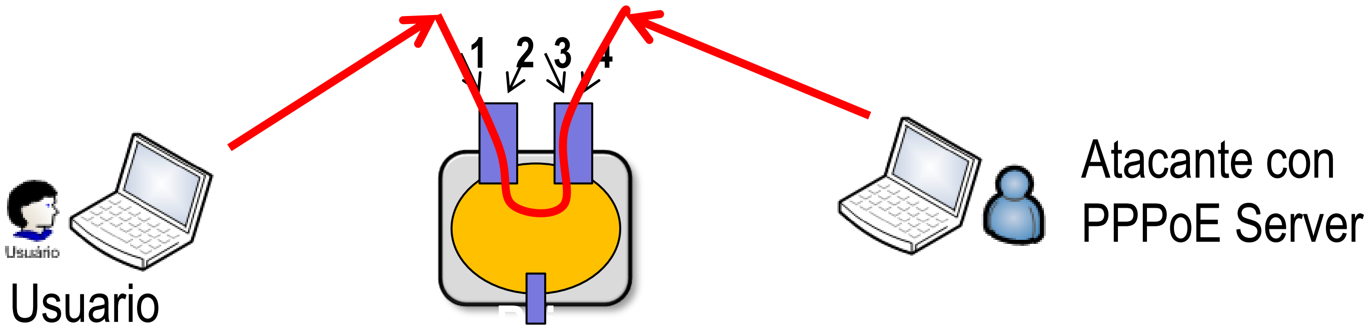
PPPoE only networks
Are the filters secure enough ?

→ Even with the filters presented in last slide, a PPPoE only Network can have security problems if the attacker is a associated client.

→Attacker can spoof a PPPoE Server, compromising the service or compromising other clients.



Arp Spoofing Countermeasures
PPPoE only networks



→ Disable default forward at all Wireless Interfaces and access lists

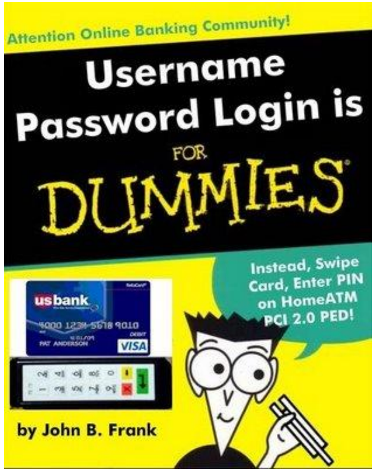
→ Configure the Bridge Filtes **BEFORE** allowing PPPoE traffic.

→ Accept PPPoE session and PPPoE discovery

→ Drop the remaining traffic

Layer 2 attacks

Attacking PPPoE and Hotspots



Attacking PPPoE and Hotspots

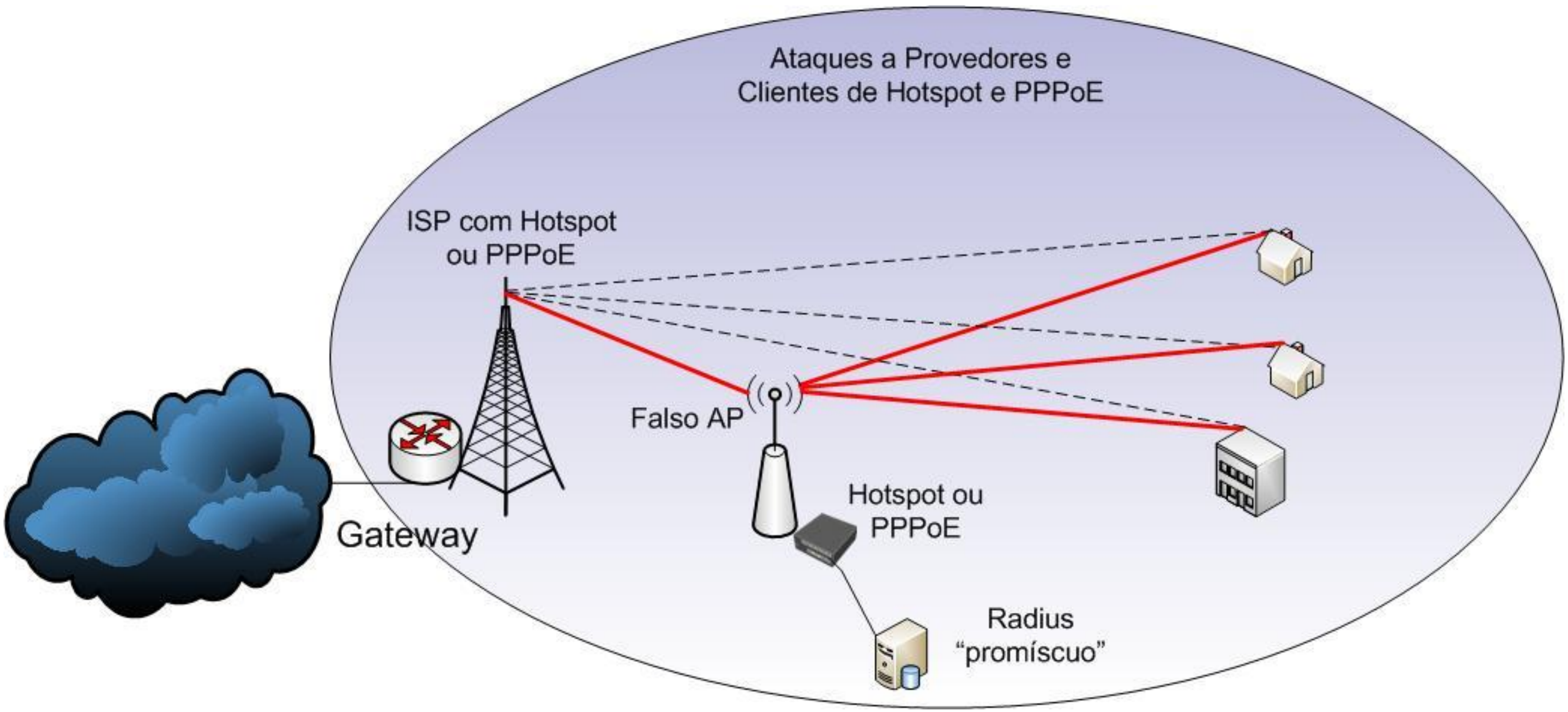
→ It is possible to deploy simple attacks, actually based on Layer 1 and layer 2 explotation, just launching an AP with the same SSID and Operation Band and with the same service (PPPoE o Hotspot)

→ Depending on the Power and physical location of the attacker nothing more is necessary. A DoS attack to the legitimate provider could do things faster.

→ The attack could be deployed with a lot of purposes, like DoS, PPPoE and Hotspot passwords theft, dns spoofing, etc.

→ To discover PPPoE/Hotspot passwords the attacker can use a "promiscuous" Radius Server.

Attacking PPPoE and Hotspots



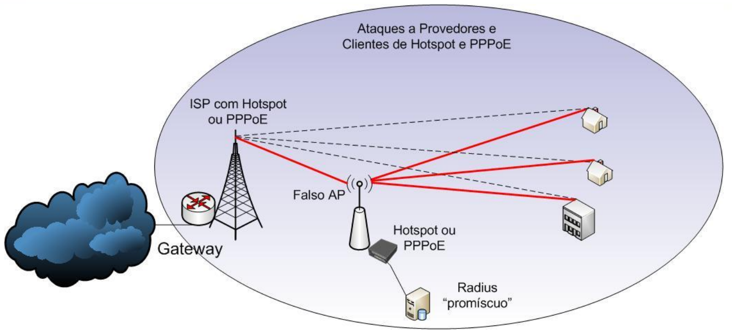
“Promiscuous” Radius Server

maia@maia-laptop:/etc/freeradius/radiusd.conf

```
…
# Log authentication requests to the log file
# allowed values: { no, yes }
log_auth = yes

# Log passwords with the authentication requests
# allowed values: { no, yes }
log_auth_badpass = yes
log_auth_goodpass = yes
…
```

Attacking PPPoE and Hotspots Countermeasures



→ Only a good encryption scheme can avoid such type of attacks. It is foolish consider that a Network without encryption is secure.

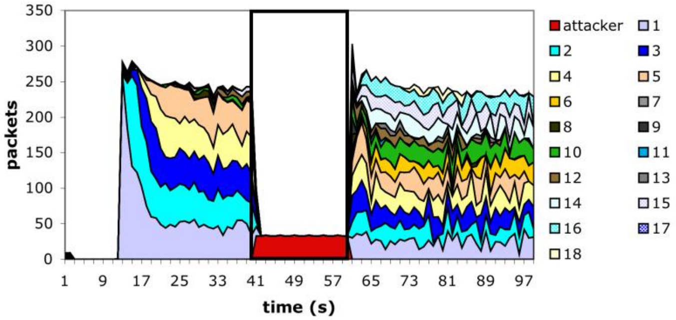
→ Encryption could be implemented in many ways, each one with proper advantages and weakness. The most secure method is with EAP-TLS with Certificates installed in all equipments. Unfortunately, there are practical limitations when using commodity hardware at client side.

→Mikrotik RouterOS has an intermediate solution with Pre Shared Keys exclusive for each client. Those keys can be administrated centralized with a Radius Server.

For details about such method, see http://mum.mikrotik.com – Brazil 2008

Atacando la capa 2

Deauthentication Attack



Denial of Service attacks against IEEE 802.11 Wireless Networks

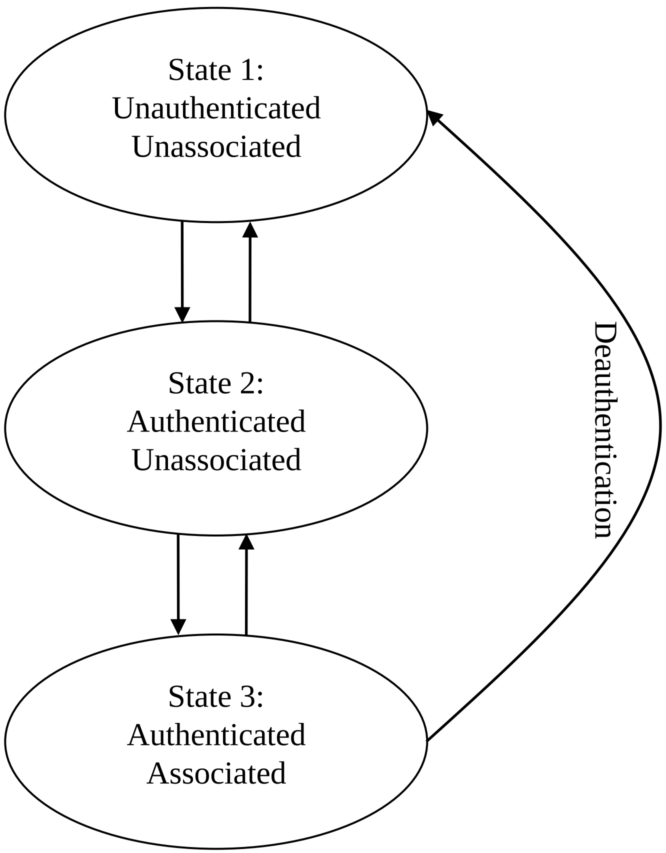
→ Attacks based on high RF power (Jamming) – layer 1

Since we are working on unlicensed bands, this is a potential risk and there is not much to do about, but only call the responsible authority for specrtum use. A good RF project could however help a lot to have a more robust network.

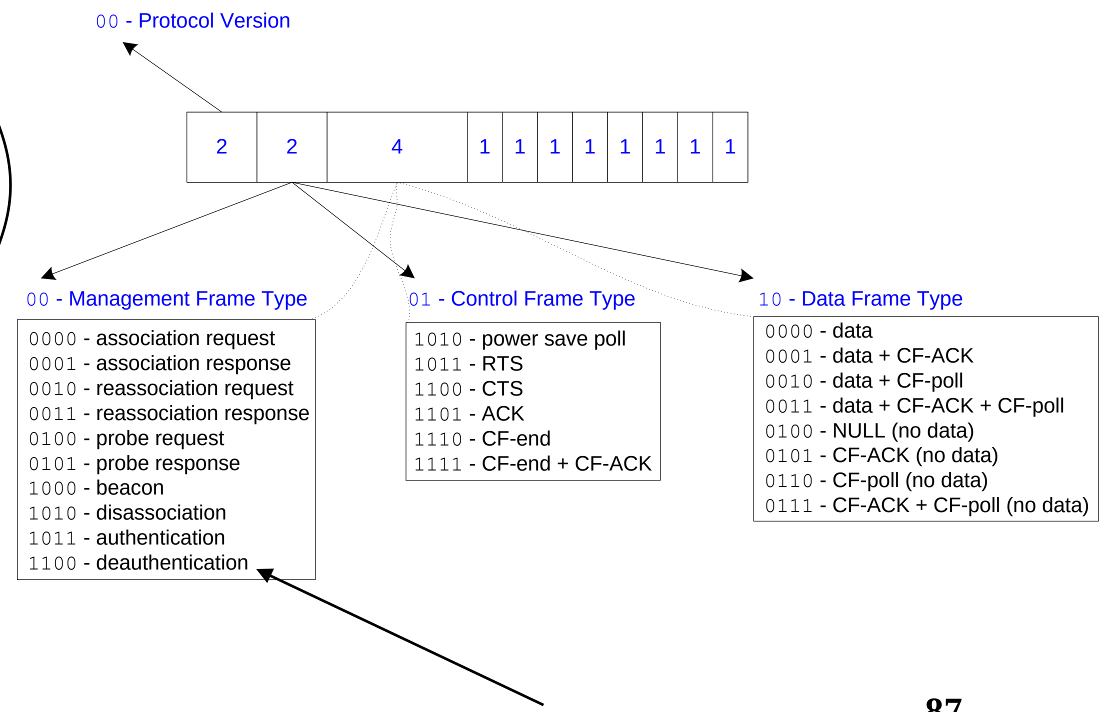
→ Protocol based attacks

The basis of those attacks are the existing vulnerabilities in control frames of 802.11 protocol. There is no authentication between wireless devices, a control frames can be forged by anyone.

Authentication Process



802.11 Types and Subtypes



Deauthentication attack

1 – The attacker uses any tool like airodump, kismet, wellenreiter, or even Mikrotik sniffer/snooper tool to find out:

→ Access Point MAC address

→ Client MAC Address

→ Channel in use

2 – Gets a position where can transmit to the AP (even with a weak signal)

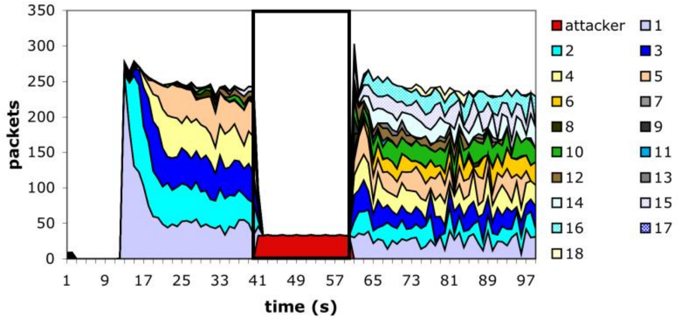
3 – Launches the attack asking the AP to de-authenticate the client;

This attack can be used not only for Denial of Service purposes, but also as support for other attacks like Man-in-the-middle in the air.

Atacando la capa 2

Deauthentication Attack

DEMO



[MikrotikBrasil]
Routers & Wireless Systems

Ataque de Deauth

maia@maia:~$ sudo my-l2-attacks –s 00:0C:42:AA:AA:AA –c 00:0C:42:CC:CC:CC
- - deauth=10 wlan0

```
09:54:01  Sending 64 direct DeAuth. STMAC:   [00:0C:42:CC:CC:CC]   [86|84 ACKs]
09:54:02  Sending 64 direct DeAuth. STMAC:   [00:0C:42:CC:CC:CC]   [111|99 ACKs]
09:54:03  Sending 64 direct DeAuth. STMAC:   [00:0C:42:CC:CC:CC]   [54|64 ACKs]
09:54:04  Sending 64 direct DeAuth. STMAC:   [00:0C:42:CC:CC:CC]   [138|130 ACKs]
09:54:07  Sending 64 direct DeAuth. STMAC:   [00:0C:42:CC:CC:CC]   [305|301 ACKs]
09:54:09  Sending 64 direct DeAuth. STMAC:   [00:0C:42:CC:CC:CC]   [318|311 ACKs]
09:54:12  Sending 64 direct DeAuth. STMAC:   [00:0C:42:CC:CC:CC]   [266|266 ACKs]
09:54:15  Sending 64 direct DeAuth. STMAC:   [00:0C:42:CC:CC:CC]   [322|316 ACKs]
09:54:17  Sending 64 direct DeAuth. STMAC:   [00:0C:42:CC:CC:CC]   [224|231 ACKs]
09:54:20  Sending 64 direct DeAuth. STMAC:   [00:0C:42:CC:CC:CC]   [346|344 ACKs]
```

Deauthentication attacks - countermeasures

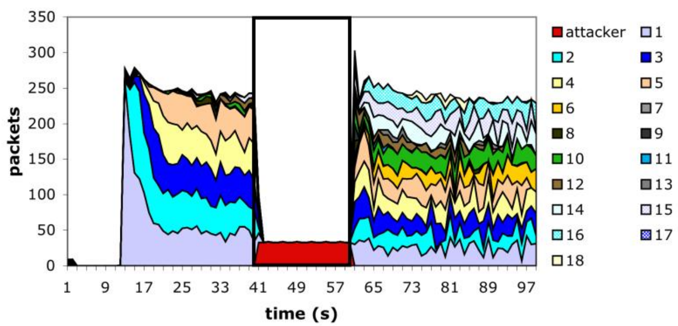
→ Once the problems with deauth attacks were revealed, some solutions were proposed, like the one below:

http://sysnet.ucsd.edu/~bellardo/pubs/usenix-sec03-80211dos-slides.pdf

→ At the MUM´s of Buenos Aires in 2007 and Krakow in 2008 some solutions using Mikrotik RouteOS were presented. Although there were only palliative solutions that could be adopted at that time..

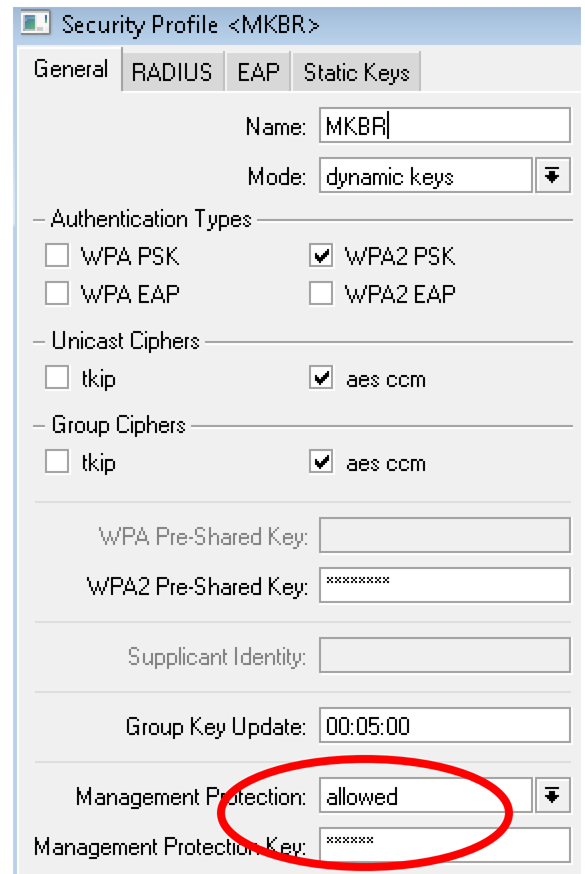
http://wiki.mikrotik.com/images/2/20/AR_2007_MB_Wireless_security_Argentina_Maia.pdf
http://mum.mikrotik.com/presentations/PL08/mdbrasil.pdf

Deauthentication Attack Countermeasures



→ Since V4 was released, with Mikrotik RouterOS is possible to authenticate control frames, turning the Deauth attack useless.

→ This is configured by means of a shared key between Mikrotik devices.



Layer 2 attacks and Coutermeasures
Conclusions

→ Networks where the physical access to Layer 2 is exposed to potential attackers, are under serious risks. Denial of Service attacks compromise network availability and other types of threats can affect users and the whole security no matter how secure is the network in respect to other layers.

→ Although Mikrotik RouterOS has a lot of features to implement security at Layer 2, some benefits of a L2 structure should be employed carefully and only in parts where the access is under a strong policy controlling physical addresses and deploying the appropriate filters.

→ Migrating a L2 network to a routed one can be a hard task at a first sight, but there are a lot of advantages when it comes to security. Migrating a dynamic routed network to a MPLS is much easier.

References

→ Cisco article– Safe Layer 2 Security in depth – version 2

→Seguridad en Capa 2 – Ing Gabriel Arellano

→ Layer 2 filtering and transparent frewalling – Cedric Blancher

→ Framework for Layer 2 attacks – Andres Berrueta / David Barroso

→ Messing up with WiFi public networks – Cedric Blancher

→ MUM Argentina 2007/ Poland 2008 / Brazil 2009 – Wireless links security

→ Mikrotik WIKI

Wardner Maia
maia@mikrotikbrasil.com.br
Phone: +55 1733447277
http://www.mdbrasil.com.br
http://www.mikrotikbrasil.com.br

MikroTik User Meeting
in Wroclaw, Poland
March 1-2, 2010

Dziękuję bardzo
Na zdrowie !

MUM
MikroTik User Meeting
in Wroclaw, Poland
March 1-2, 2010