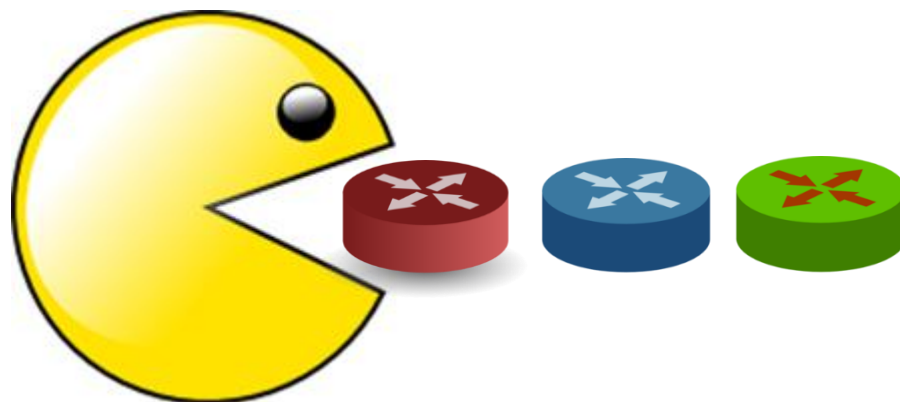


Routing Security



Hungary MUM – Budapest – March, 2011

Eng. Wardner Maia

Brazil

Introduction

Name: Wardner Maia

Country: Brazil

Electronic/Telecommunications Engineer

Internet Service Provider since 1995

Training Courses on Wireless since 2002

Mikrotik Certified Trainer since, 2007

Technical Director of company MD Brasil IT & Telecom

Member of board directors of LACNIC (<http://www.lacnic.org>)

MD Brasil Information Technology and Telecommunications

- ISP (Access and Hosting Services)
- Authorized Telecommunications operator in Brazil.
- Mikrotik Distributor and Training Partner.
- Consulting services

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

Target audience and Objectives

Target Audience:

ISP's and WISP's running or planning to run OSPF and BGP in their networks.

Objectives:

To understand conceptually the existing threats related to dynamic routing protocols caused by

- Intentional attacks
- self misconfigurations
- leak of measures to prevent misconfigurations from neighbors AS's.

To establish a set of Best Common Practices in Mikrotik RouterOS to avoid or minimize the above risks.

Why Routing Security ?

- The widely used routing protocols were created in early days of the Internet when security risks were not intense. .
- BGP, the protocol that glues together the largest and most complex network ever created, was born without any security concern.
- The same regarding to OSPF, nowadays the most popular dynamic Internal Gateway Protocol
- There are tons of known attacks against dynamic routing that can compromise, **confidentiality**, **integrity** and **availability** on networks of any size. Therefore, the whole Internet can be affected.

Why Routing Security ?

- Security in a wide meaning is not only related to intentional attacks but to incidents caused by misconfigurations and operating systems bugs.
- In recent past the Internet suffered regional and global problems caused by non-intentional administrators mistakes. The most notable:
 - Pakistan Telecom x Youtube
 - Mikrotik x Cisco bug (long as path bug)
- In the past 2 years we've seen several small ISP's growing up, getting their AS's and starting operating their own OSPF/BGP Networks.
- ***Are those new players well prepared to face the issues related to dynamic routing weakness ???***

What is routing security and what we will be discussing about ?

Security of the routing protocol itself



- “Semantics” that transport the routing information
- Algorithms used to select the best paths

Security of Topology information



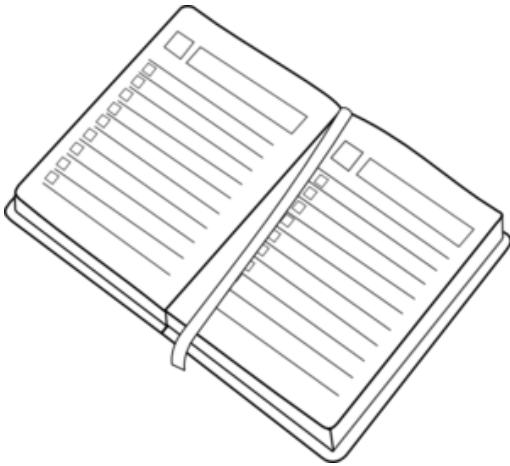
- Topology of the network carried by the routing protocol

Security of the involved Devices



- Routers than run the routing protocol (We will not cover devices protection in this presentation)

Agenda



1) Dynamic routing essentials

2) OSPF

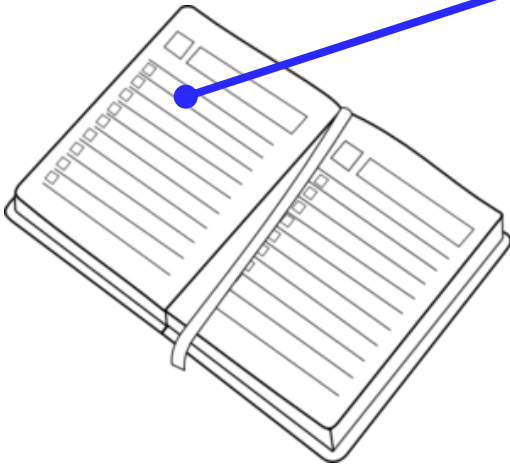
- OSPF Overview
- OSPF threats and countermeasures

3) BGP

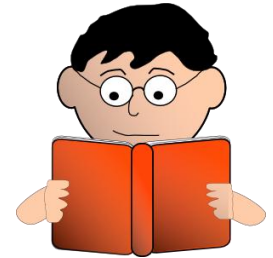
- BGP Overview
- BGP threats and countermeasures

4) Conclusions.

Agenda



1) Dynamic routing essentials



2) OSPF

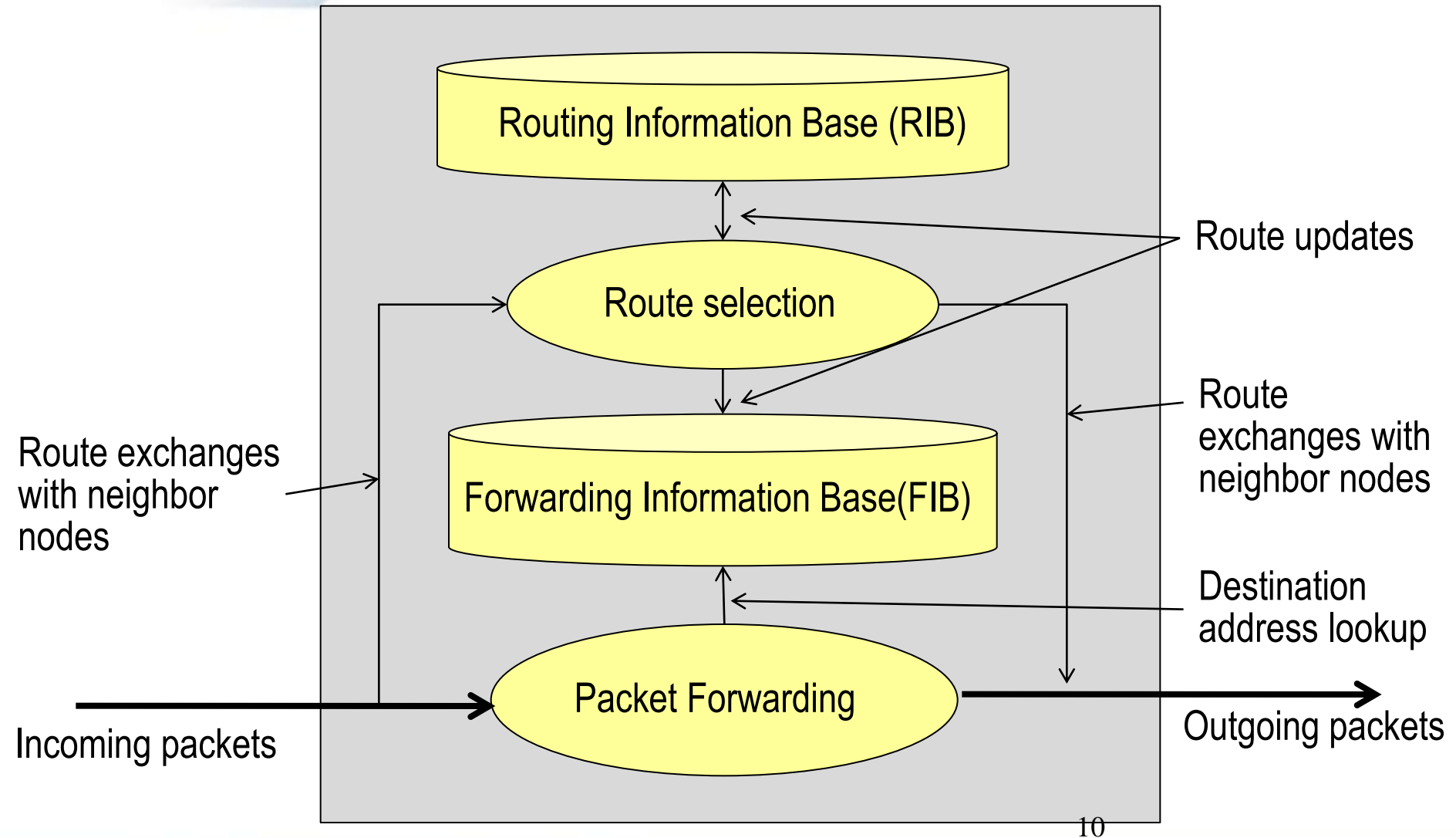
- OSPF Overview
- OSPF threats and countermeasures

3) BGP

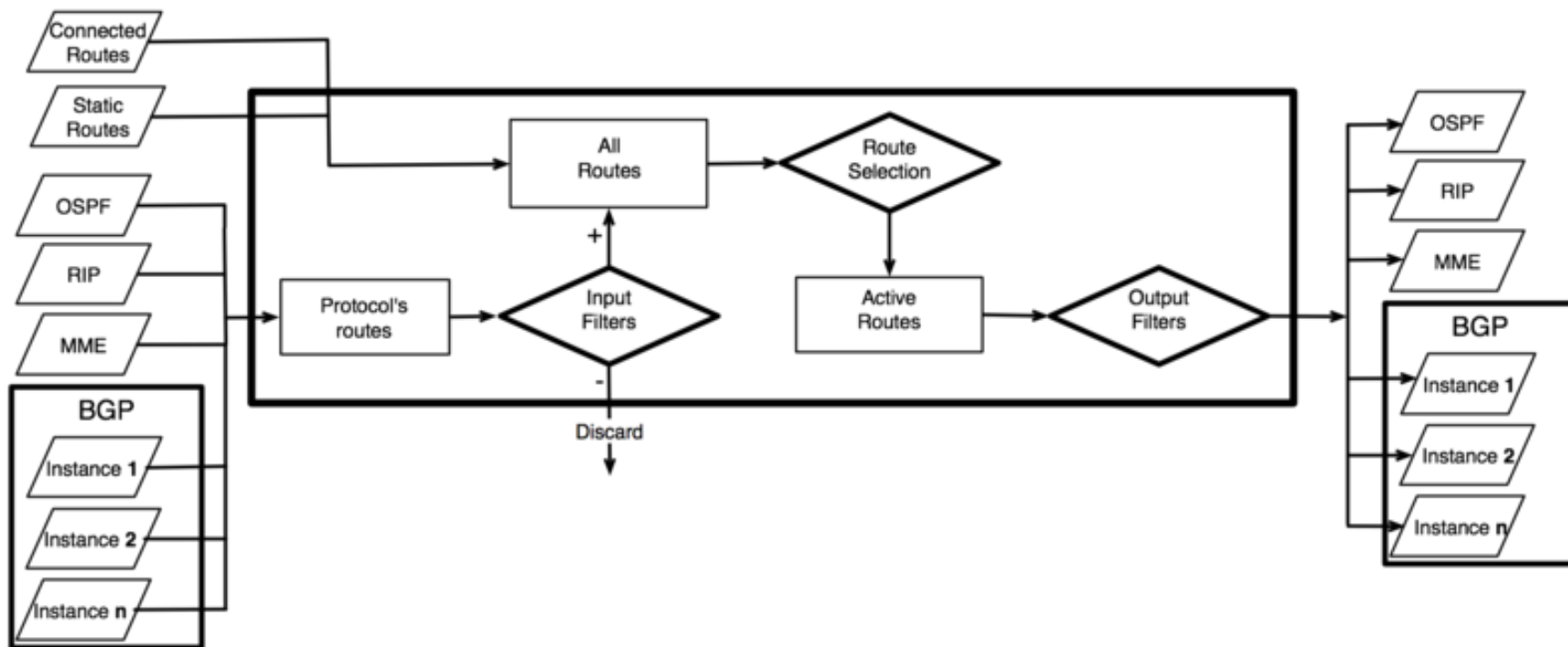
- BGP Overview
- BGP threats and countermeasures

4) Conclusions.

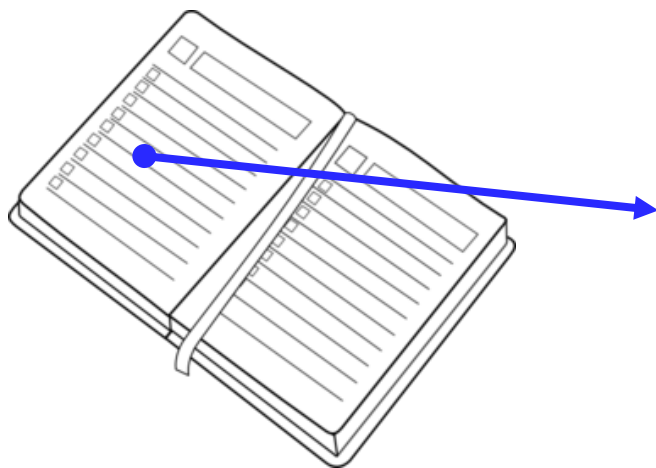
A Router, conceptually



Routing on Mikrotik RouterOS



Agenda



1) Dynamic routing essentials



2) OSPF

- OSPF Overview
- OSPF threats and countermeasures

3) BGP

- BGP Overview
- BGP threats and countermeasures

4) Conclusions.

OSPF

OSPF (Open Shortest Path First) is a “link-state” type protocol.

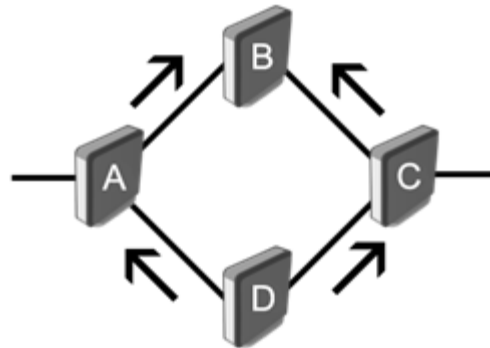
OSPF uses the Dijkstra algorithm to calculate the shortest path to a specific destination.

Characteristics of a link-state routing protocol:

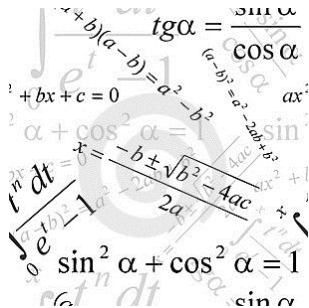
- Respond quickly to network changes;
- Send triggered updates when a network change occurs;
- Send periodic updates, known as link-state refresh, at longer intervals.

How OSPF works

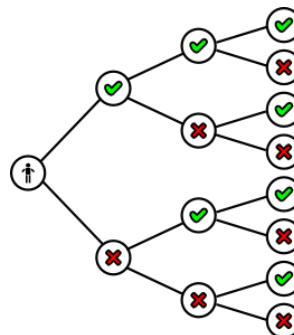
Hello packets discovers neighbors and build adjacencies between them



A Link State Database (LSDB) is constructed



Dijkstra algorithm runs



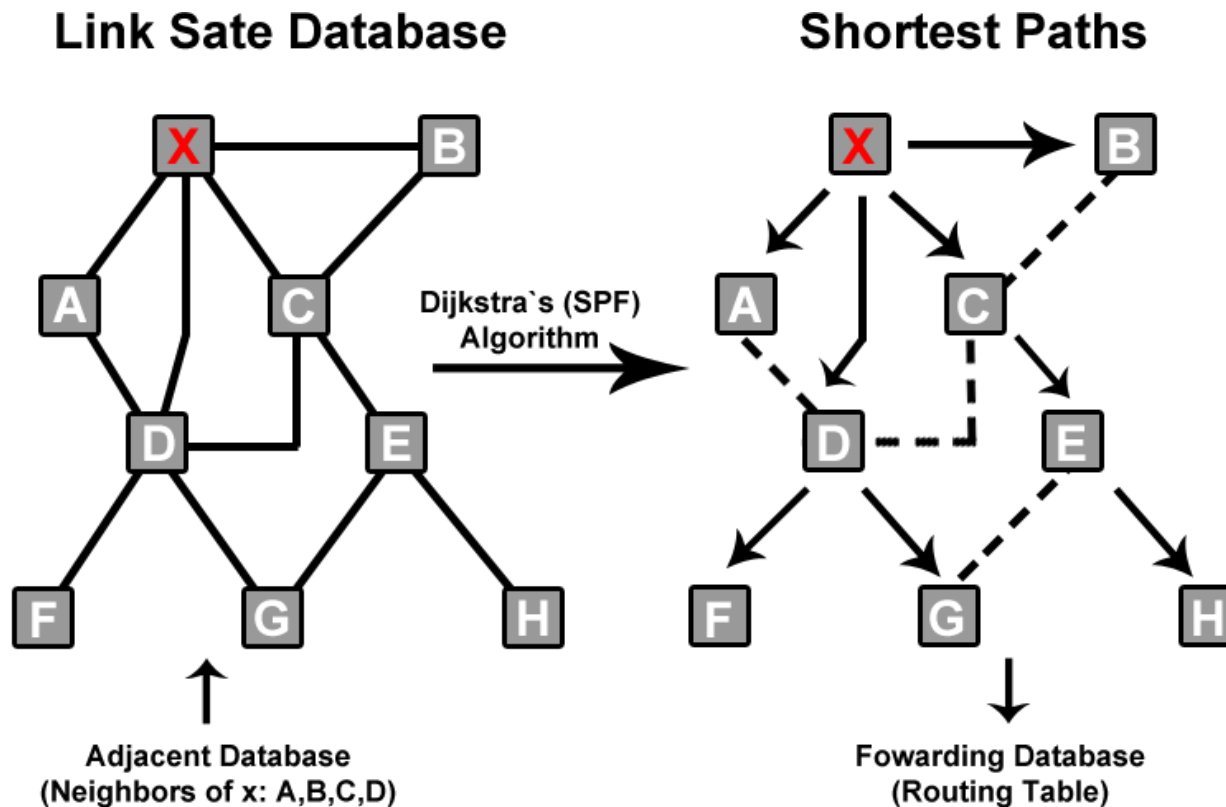
An SPF Tree is build

**LSA
Messages**

| Route List | | |
|--|--------------|-----------|
| Routes | Nexthops | Rules VRF |
| <div> + - ✓ ✗ 📄 🔍 </div> | | |
| | Dst. Address | Gateway |
| DAo | 0.0.0.0/0 | 192.168. |
| DAo | 10.0.1.3 | 192.168. |
| DAo | 10.0.1.4 | 192.168. |
| DAC | 10.0.1.5 | loopback |
| DAo | 10.0.1.6 | 192.168. |

The Forwarding table is formed

SPF Calculation



Assumes that all links are ethernet type with OSPF cost = 10

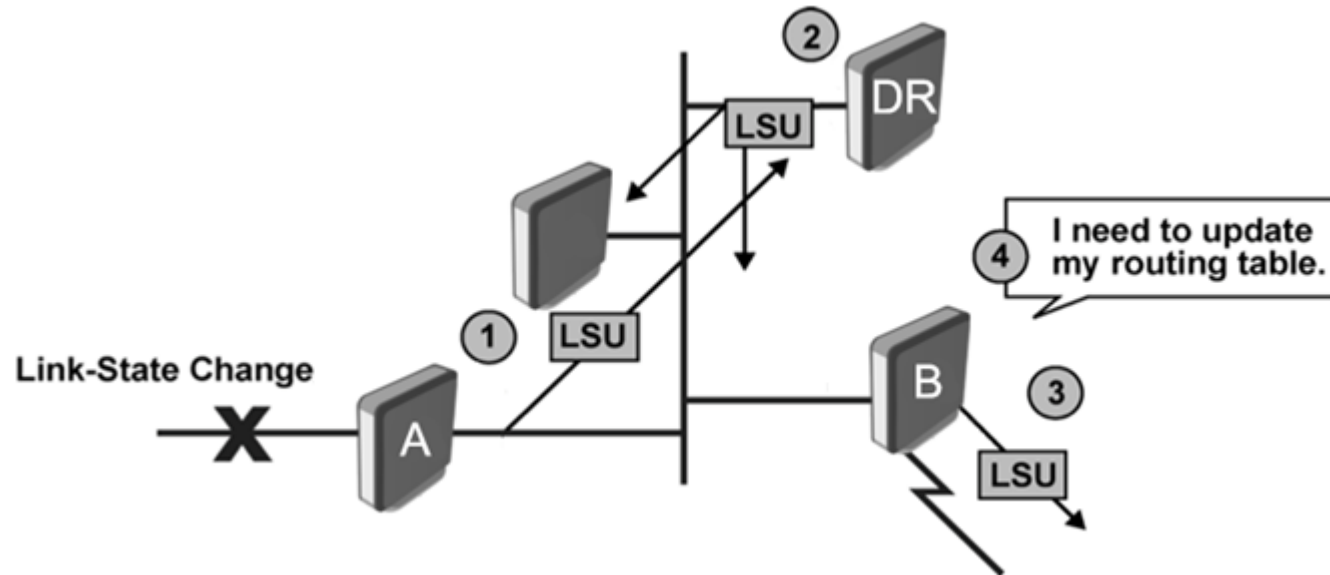
OSPF Link State messages

- LSA – Link State Advertise
- LSU – Link State Update
- LSR – Link State Request
- LSAck – Link State Acknowledgement



| OSPF | | | | | | | |
|--|----------|-------------|--------------|-------------|----------------|---------|--|
| Areas Area Ranges Virtual Links Neighbors NBMA Neighbors Sham Links LSA Routes AS Bo | | | | | | | |
| Y | | | | | | | |
| Instance | Area | Type | ID | Originator | Sequence Nu... | Age (s) | |
| as external | | | | | | | |
| default | | as external | 0.0.0.0 | 10.0.1.4 | 80001379 | 20 | |
| default | | as external | 0.0.0.0 | 10.0.1.3 | 80001379 | 75 | |
| network | | | | | | | |
| default | backbone | network | 192.168.1.10 | 10.16.16.16 | 80000f07 | 544 | |
| default | backbone | network | 192.168.1.14 | 10.16.16.16 | 80000eed | 1215 | |
| default | backbone | network | 192.168.1.6 | 10.0.1.5 | 80000f07 | 479 | |
| router | | | | | | | |
| default | backbone | router | 10.16.16.16 | 10.16.16.16 | 80001530 | 1562 | |
| default | backbone | router | 10.0.1.3 | 10.0.1.3 | 80001381 | 455 | |
| default | backbone | router | 10.0.1.5 | 10.0.1.5 | 80000f0a | 479 | |
| default | backbone | router | 10.0.1.4 | 10.0.1.4 | 80001524 | 1123 | |

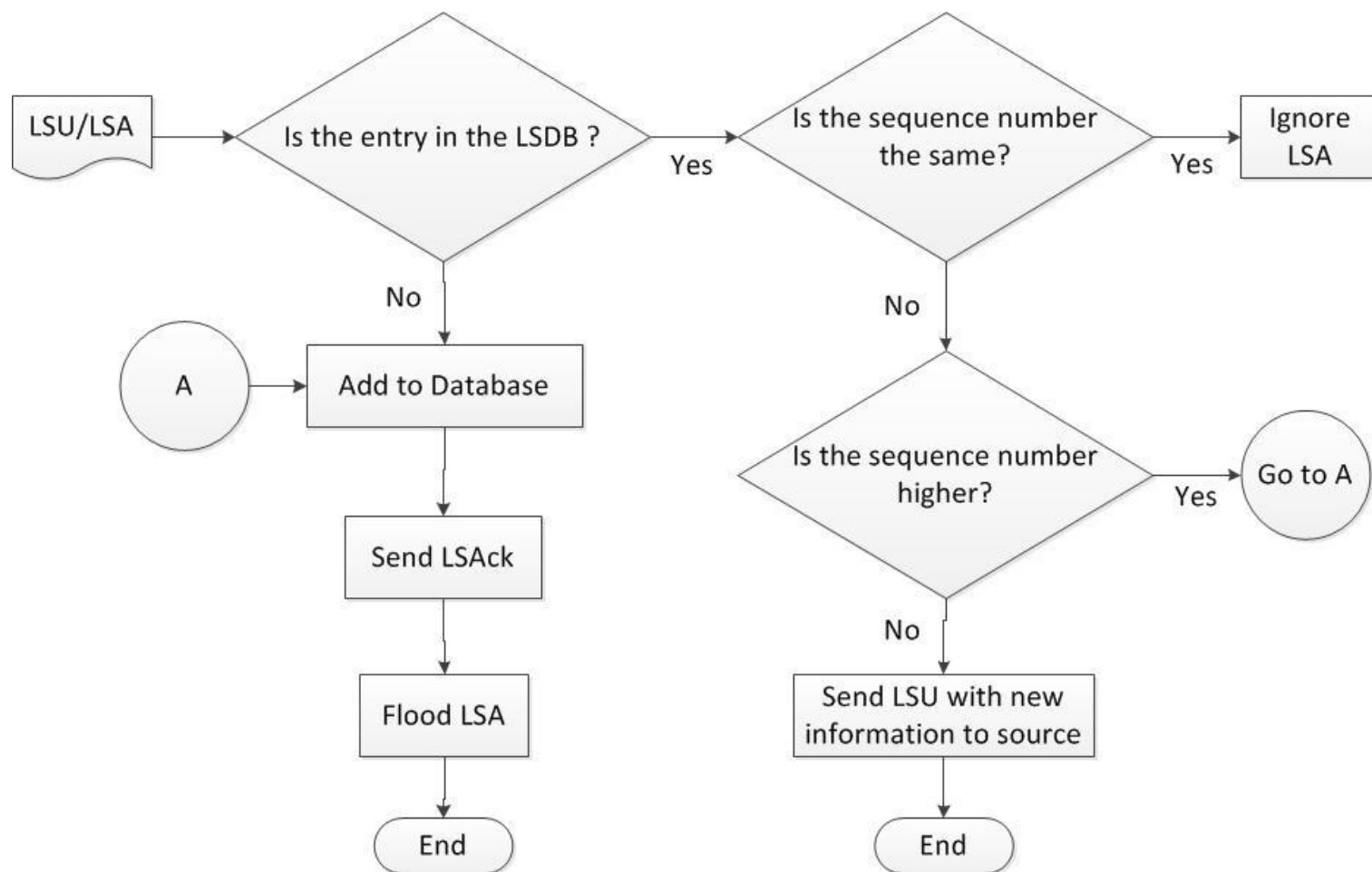
What about topology changes ?



When topology changes:

- LSU messages are flooded
- Databases are updated
- SPF (Dijkstra algorithm) runs again
- New Forwarding tables are generated.

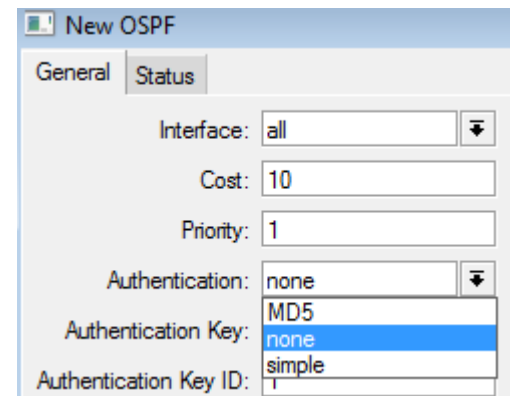
LSU/LSA Processing



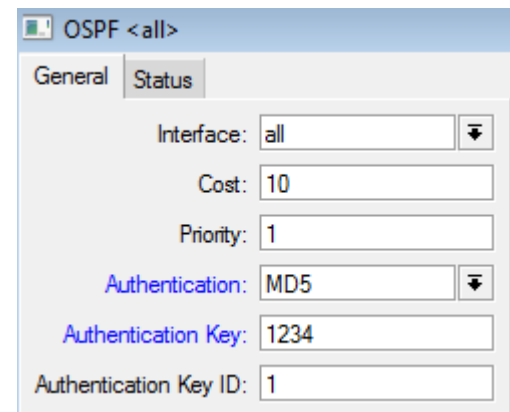
OSPF security

Authentication:

- By default, OSPF has no authentication
- Two authentication methods based on “pre shared” keys are possible:
 - Simple (password is transmitted in plain text)
 - MD5 (Message Digest authentication – MD5 hash)

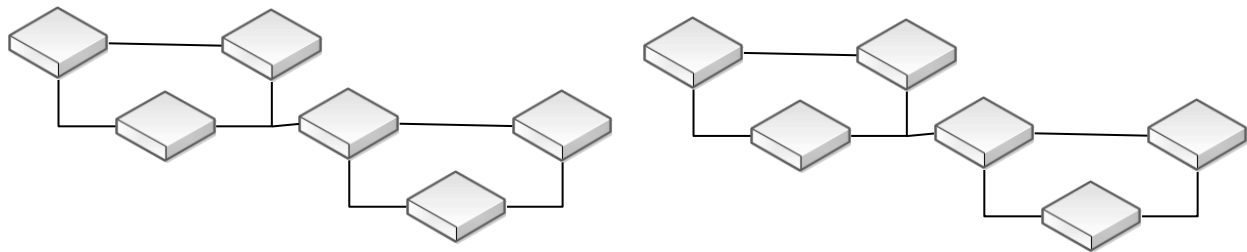


The 'New OSPF' window shows the 'General' tab. The 'Interface' is set to 'all', 'Cost' is 10, and 'Priority' is 1. The 'Authentication' dropdown is set to 'none'. The 'Authentication Key' dropdown shows 'MD5', 'none' (highlighted), and 'simple'. The 'Authentication Key ID' is set to 1.



The 'OSPF <all>' window shows the 'General' tab. The 'Interface' is set to 'all', 'Cost' is 10, and 'Priority' is 1. The 'Authentication' dropdown is set to 'MD5'. The 'Authentication Key' is set to '1234'. The 'Authentication Key ID' is set to 1.

Attacking OSPF



Attacks against OSPF

Basically, attacks against OSPF consist on forging Hello, LSA and LSU messages on behalf of authorized hosts, causing:

- Denial of service
- and / or
- Topology changes

Topology changes, leads to other threats like

- Eavesdropping
- Man-in-the-middle attack

OSPF

Resource Starvation Attacks 1/2

→ “Phantom LSAs” are Router/Network LSAs sent on behalf of non-existing OSPF peers. (no need to know the Authentication key)

→ These entries are ignored by the Shortest Path First (SPF) algorithm (do not produce topology changes)

→ “Phantom LSAs” are entered in the Link State Database and each entry is kept until “MaxAge” expires

Starvation attacks will work regardless encryption

OSPF

Resource Starvation Attacks 2/2

Memory Impact

- Bogus LSA's with an arbitrary source take up space in the topology table until the LSA ages out

CPU impact

- LSA's with bogus MD5 passwords invoke the MD5 function

Bandwidth impact

- Bogus LSA's and the associated legitimate response traffic could be disruptively high in large, densely populated areas.
- Bogus link state request packets can saturate a link with requests for nonexistent networks.

OSPF attacks

Forcing topology changes 1/2

An attacker can force topology changes by introducing false LSA Information

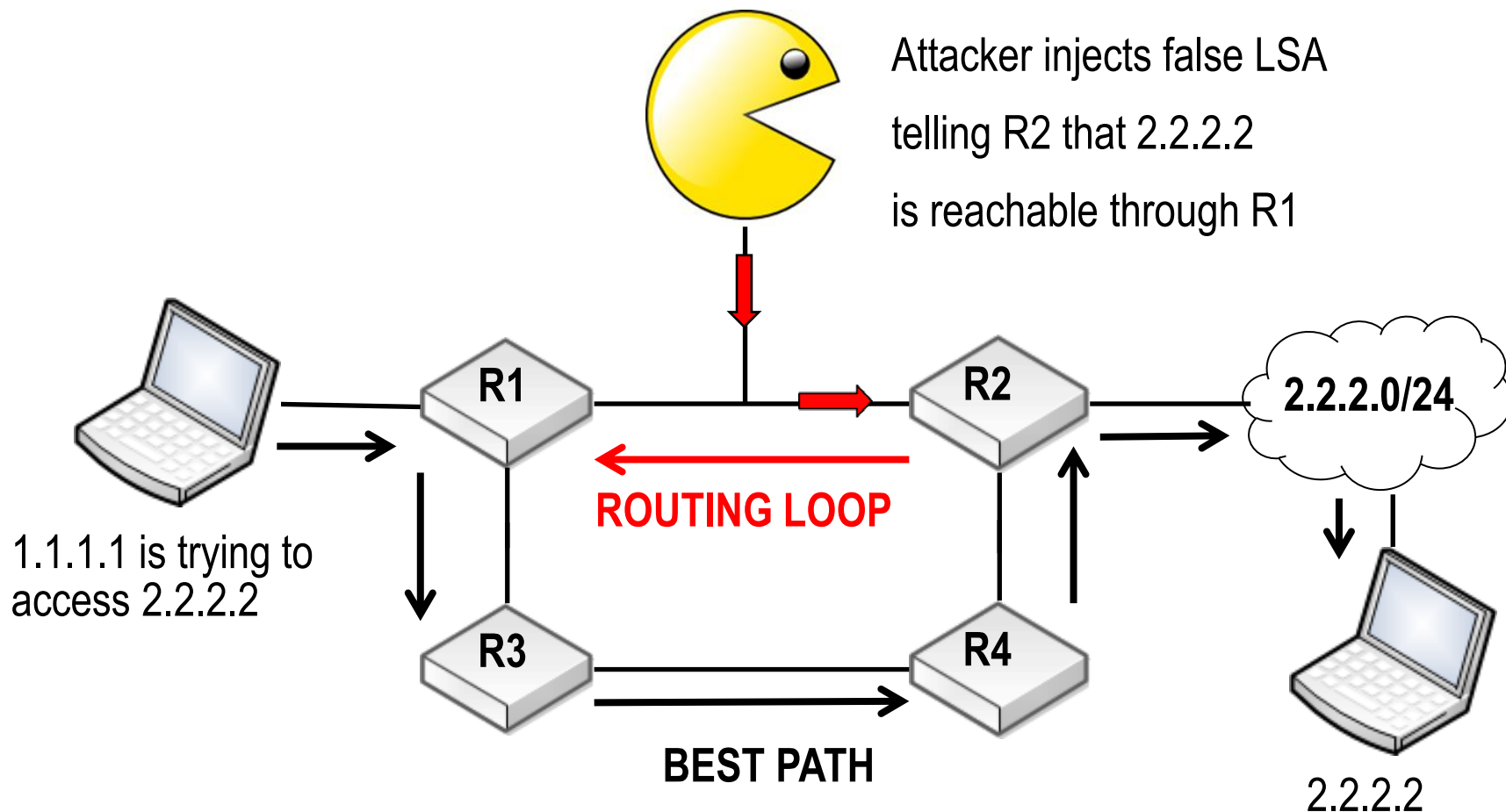
Pre-condition:

- absence of encryption.
- compromised pre shared key.

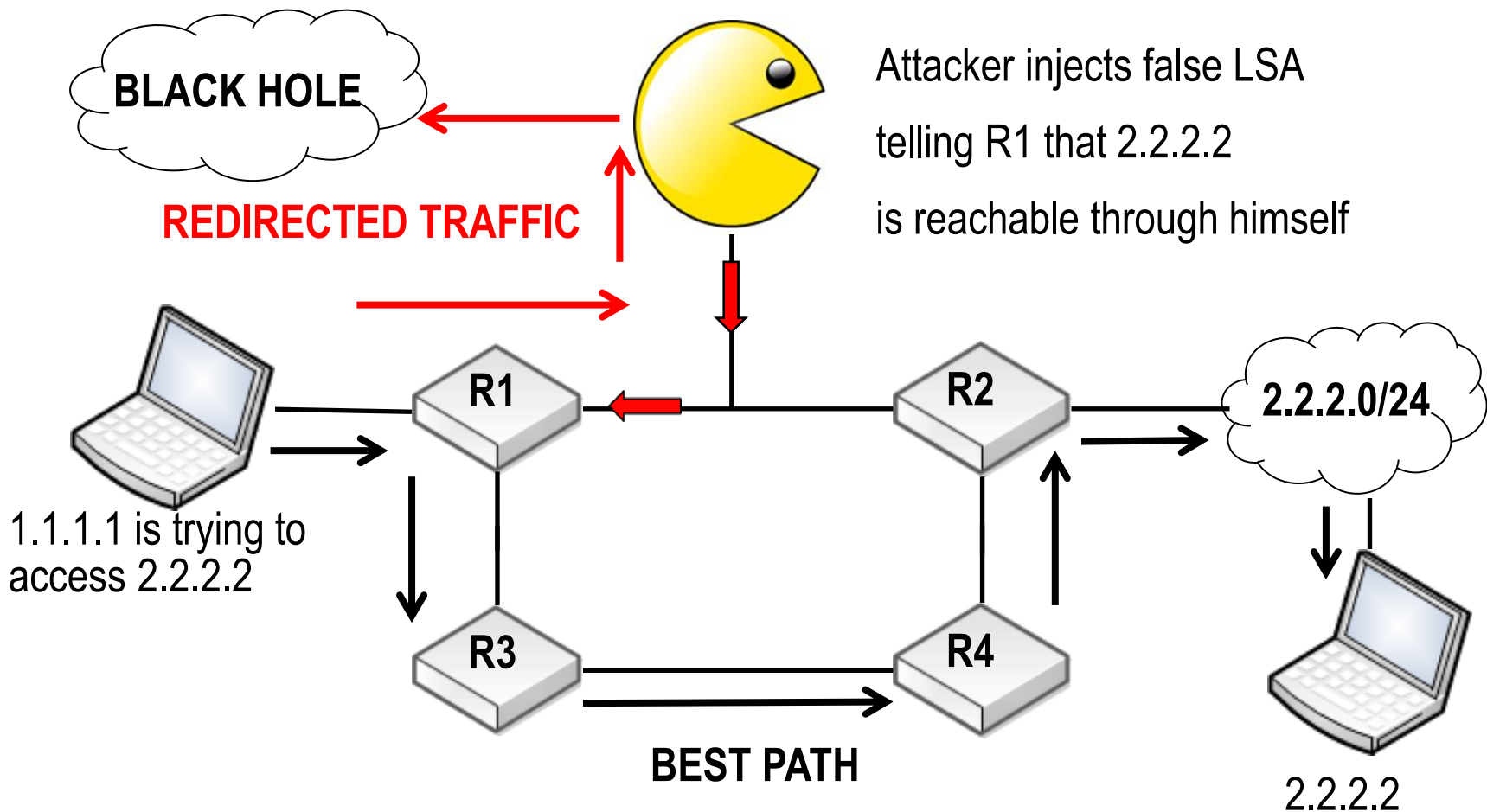
Impacts of Topology Changes

- Allow Eavesdropping
- Starve/Overload a network
- Unstable topology (loops, route-flapping)

Misdirecting traffic to form routing Loops

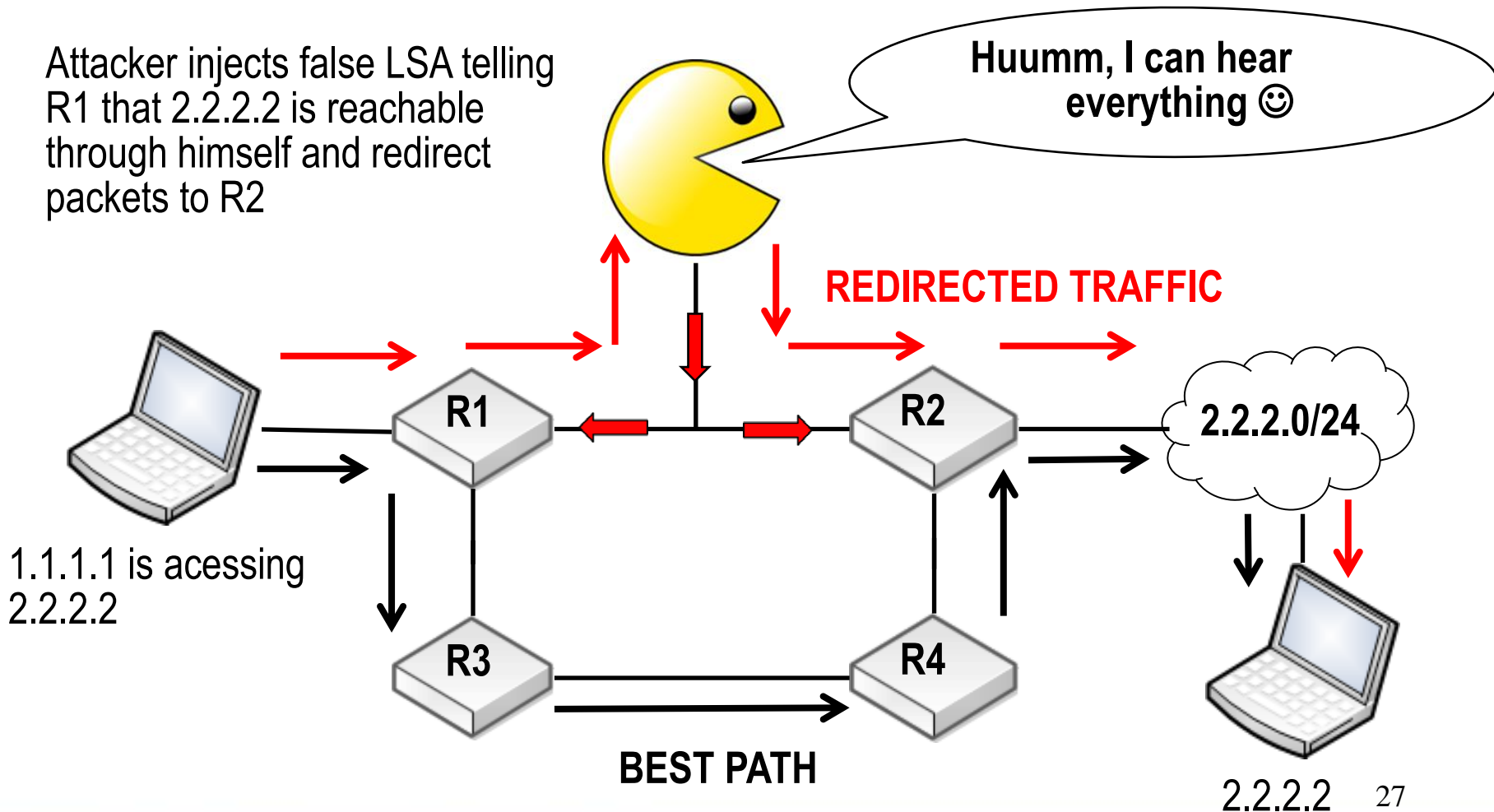


Missdirecting traffic to a black hole

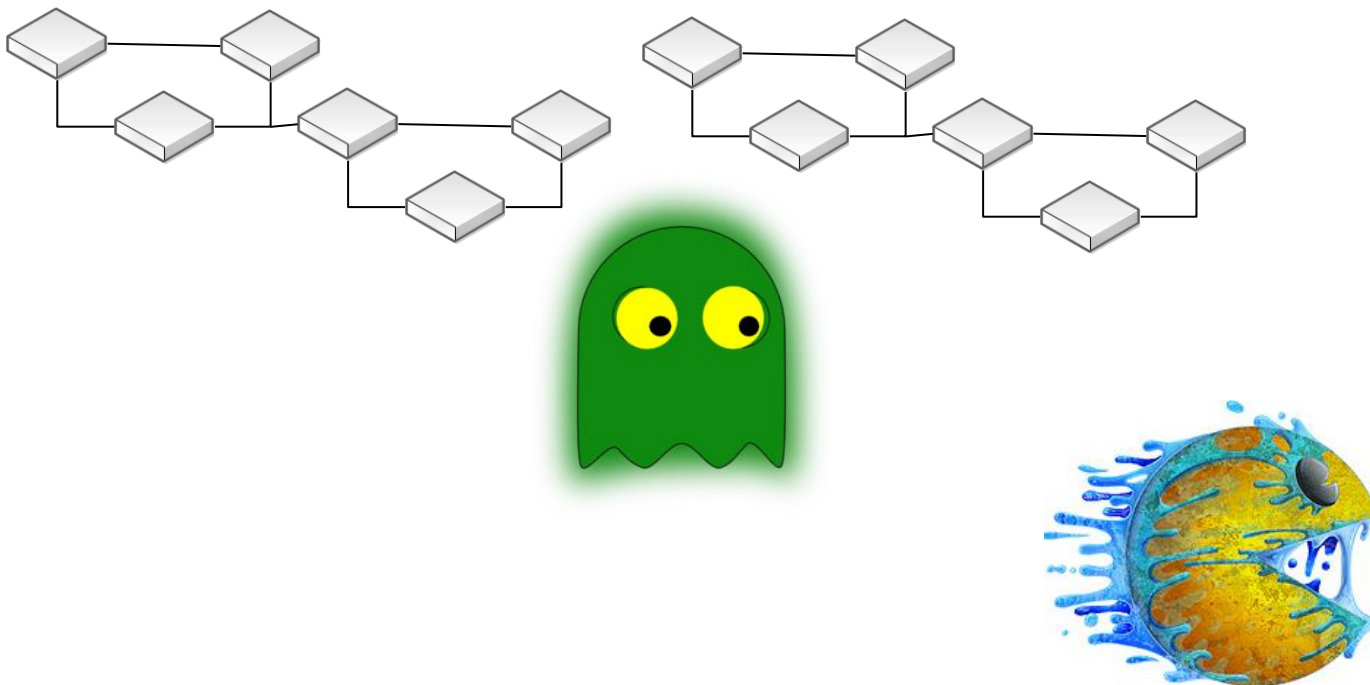


Eavesdropping/Man-in-the-middle

Attacker injects false LSA telling R1 that 2.2.2.2 is reachable through himself and redirect packets to R2



Protecting OSPF



Protecting OSPF (from the perspective of attacker's location)

From the point of view of attacker's location we can divide the possible attacks in;

External attacks

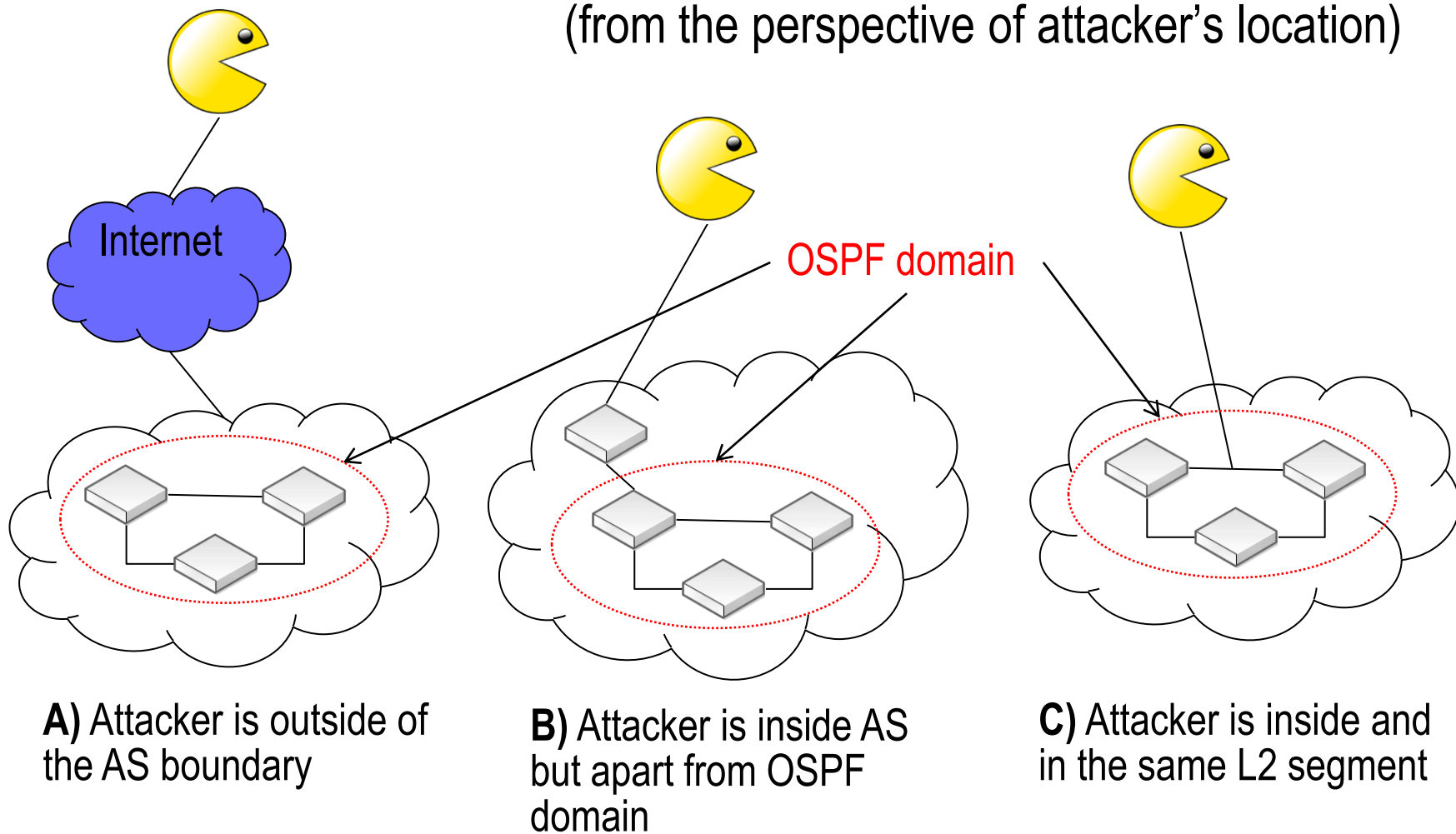
→ Attacker is outside of the Autonomous System (AS) boundary

Internal attacks

→ Attacker is inside the AS, in the same L2 network segment where OSPF is running

→ Attacker is inside the AS, but not in the same L2 network segment.

Attacks against OSPF (from the perspective of attacker's location)



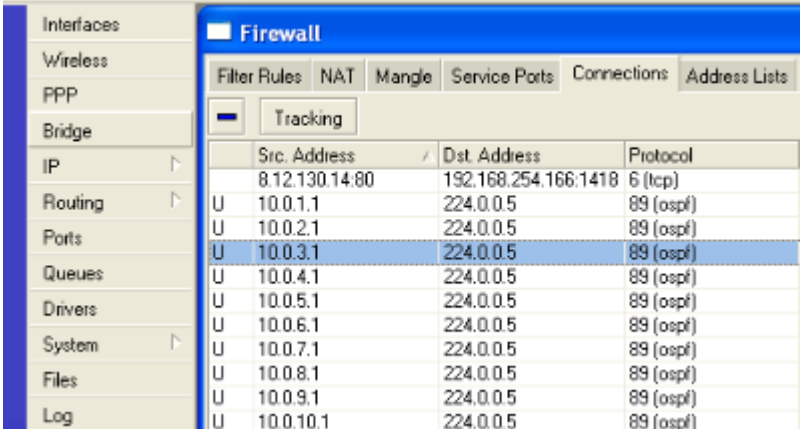
Attacks against OSPF

Attack example

MD Brasil customer using Mikrotik RouterOS 2.9x under attack from an outsider in 2007

→ Forged Source IP address.

→ Attacker generating packets with destination address 224.0.0.5 - Multicast Address AllSPFRouters



| Firewall | | | |
|---|----------------|----------------------|-----------|
| Filter Rules NAT Mangle Service Ports Connections Address Lists | | | |
| Tracking | | | |
| | Src. Address | Dst. Address | Protocol |
| | 8.12.130.14:80 | 192.168.254.166:1418 | 6 (tcp) |
| U | 10.0.1.1 | 224.0.0.5 | 89 (ospf) |
| U | 10.0.2.1 | 224.0.0.5 | 89 (ospf) |
| U | 10.0.3.1 | 224.0.0.5 | 89 (ospf) |
| U | 10.0.4.1 | 224.0.0.5 | 89 (ospf) |
| U | 10.0.5.1 | 224.0.0.5 | 89 (ospf) |
| U | 10.0.6.1 | 224.0.0.5 | 89 (ospf) |
| U | 10.0.7.1 | 224.0.0.5 | 89 (ospf) |
| U | 10.0.8.1 | 224.0.0.5 | 89 (ospf) |
| U | 10.0.9.1 | 224.0.0.5 | 89 (ospf) |
| U | 10.0.10.1 | 224.0.0.5 | 89 (ospf) |

Attacks against OSPF

A) Attacker is outside of the AS boundary (1/2)

Question: will such attack work ??

On physical point-to-point networks and Broadcast networks the IP destination is set to the Multicast address “AllSPFRouters” (224.0.0.5) .

On NBMA and all other network types (including virtual links), the majority of OSPF packets are sent as unicasts, i.e., sent directly to the other end of the adjacency. In this case, the IP destination is just the Neighbor IP address associated with the other end of the adjacency (see RFC 2326, section 10).

So, the answer is **YES**, the attack could work from any point of the Internet !

Attacks against OSPF

A) Attacker is outside of the AS boundary (2/2) - Countermeasures

Although external attacks are possible, the attacker should be able to send unicast OSPF packets destined to internal routers. To thwart such attacks, just follow the 2 hints below:

- Never, never run OSPF beyond your boundaries i.e. with networks under other administration.
- Deny protocol 89 (OSPF) at your border routers.

New Firewall Rule

General Advanced Extra Action Statistics

Chain: forward

Src. Address:

Dst. Address:

Protocol: ☐ ospf

Src. Port:

Dst. Port:

Any. Port:

P2P:

In. Interface: ☐ WAN

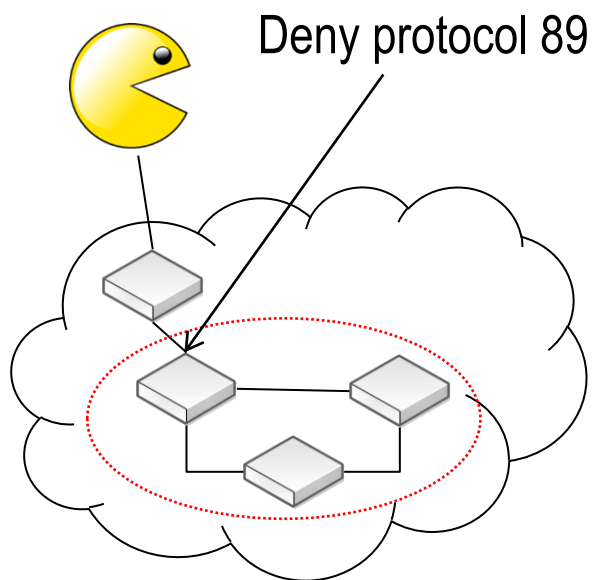
Firewall Rule <>

General Advanced Extra Action Statistics

Action: drop

Attacks against OSPF (from the perspective of attacker's location)

B) Attacker is inside the AS, but not in the same L2 network segment. (e.g. your client CPE)



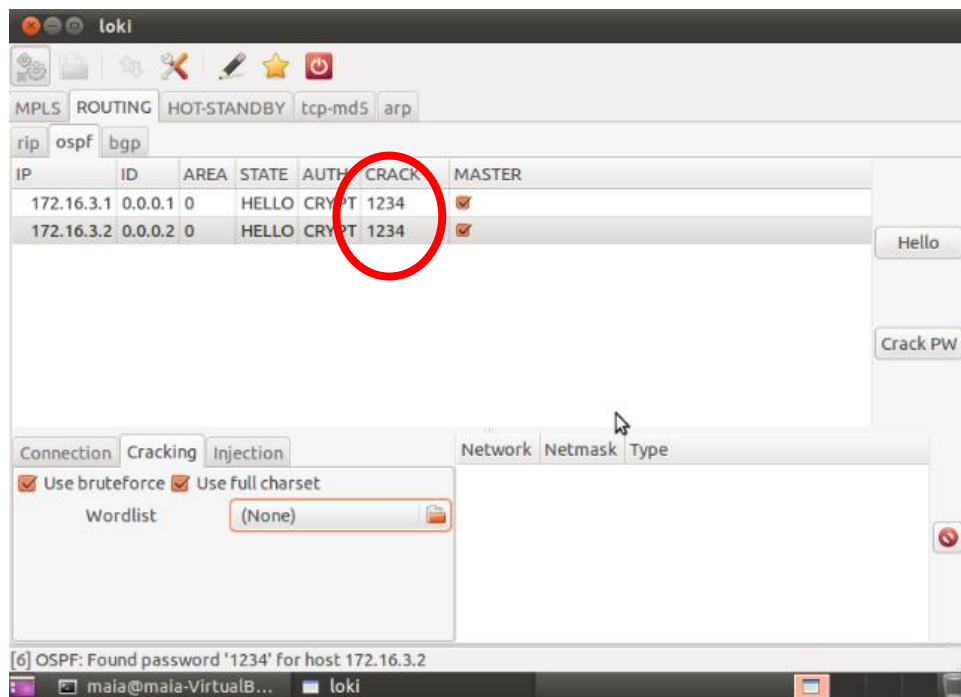
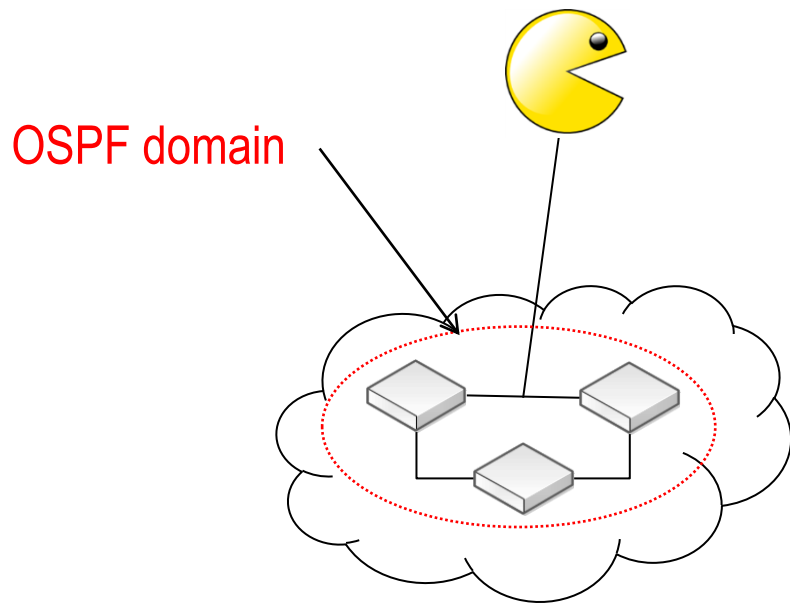
The same considerations from external AS attacks could be made. Countermeasures are similar:

→ Firewall rules should be placed at the boundaries of OSPF domain (**forward** an **input** chains)

Attacks against OSPF

C) Attacker is inside and in the same L2 segment (1/3)

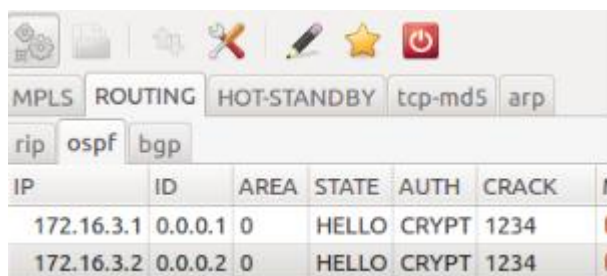
Without any doubt , this is the worst condition. Even with MD5 authentication OSPF can be exploited easily. On the net there are tools to explore this situation.



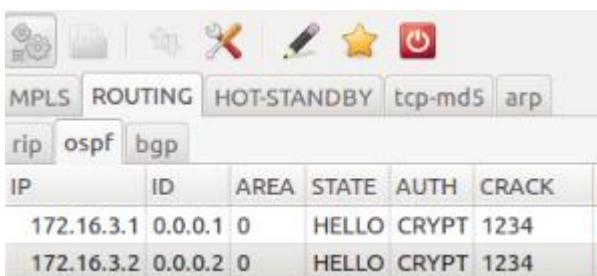
Attacks against OSPF

C) Attacker is inside and in the same L2 segment (2/3)

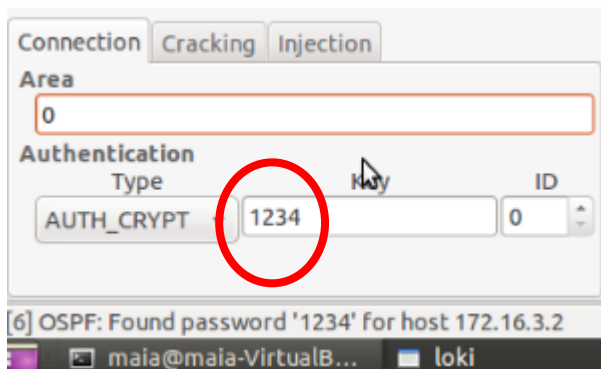
Once the pre shared key is compromised, attacker could do anything a real router could, since flooding LSA's for resource starvation, or impersonate a network router. Imagination and creativity will do the rest ☺



| IP | ID | AREA | STATE | AUTH | CRACK |
|------------|---------|------|-------|-------|-------|
| 172.16.3.1 | 0.0.0.1 | 0 | HELLO | CRYPT | 1234 |
| 172.16.3.2 | 0.0.0.2 | 0 | HELLO | CRYPT | 1234 |



| IP | ID | AREA | STATE | AUTH | CRACK |
|------------|---------|------|-------|-------|-------|
| 172.16.3.1 | 0.0.0.1 | 0 | HELLO | CRYPT | 1234 |
| 172.16.3.2 | 0.0.0.2 | 0 | HELLO | CRYPT | 1234 |



Connection Cracking Injection

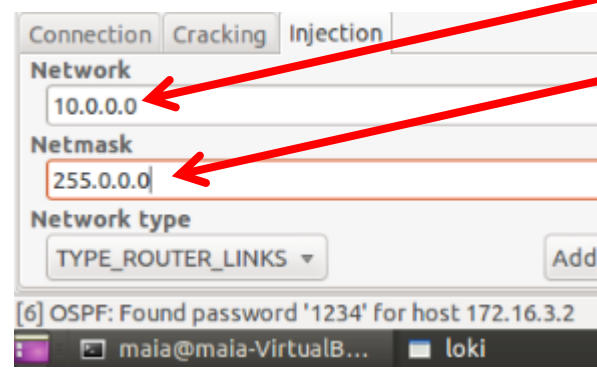
Area: 0

Authentication Type: AUTH_CRYPT

Key: 1234

ID: 0

[6] OSPF: Found password '1234' for host 172.16.3.2



Connection Cracking Injection

Network: 10.0.0.0

Netmask: 255.0.0.0

Network type: TYPE_ROUTER_LINKS

Add

[6] OSPF: Found password '1234' for host 172.16.3.2

Creating an arbitrary network

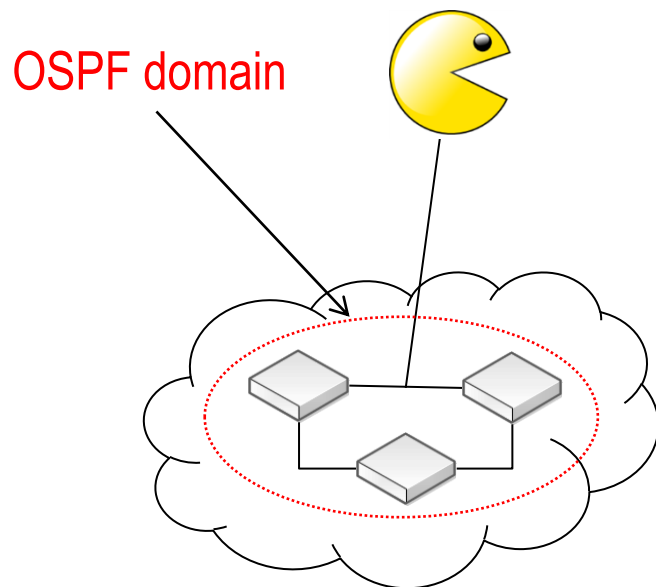
Attacks against OSPF

C) Attacker is inside and in the same L2 segment (3/3)

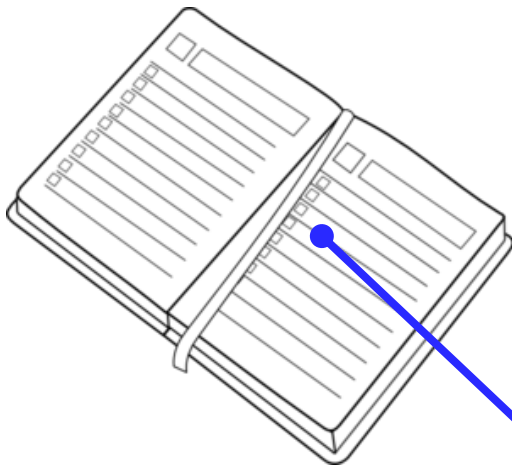
Countermeasures:

- Choosing a strong password will delay (but not avoid) the discovery. It's only a matter of time.
- The real solution is NOT TO SHARE L2 segments with outsiders.
- When L2 sharing could not be avoided, make sure to promote L2 isolation between hosts. Take a look on the presentation:

<http://mum.mikrotik.com/presentations/PL10/maia.pdf>



Agenda



1) Dynamic routing essentials ✓

2) OSPF

→ OSPF Overview ✓

→ OSPF threats and countermeasures ✓

3) BGP

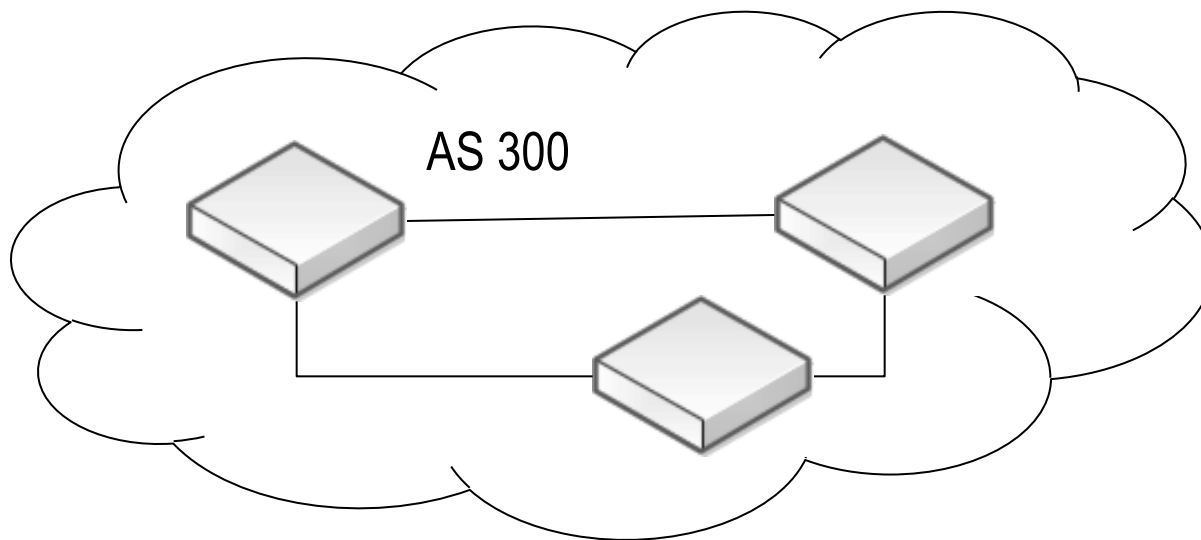
→ BGP Overview

→ BGP threats and countermeasures

4) Conclusions.

Autonomous System (AS) and the Internet

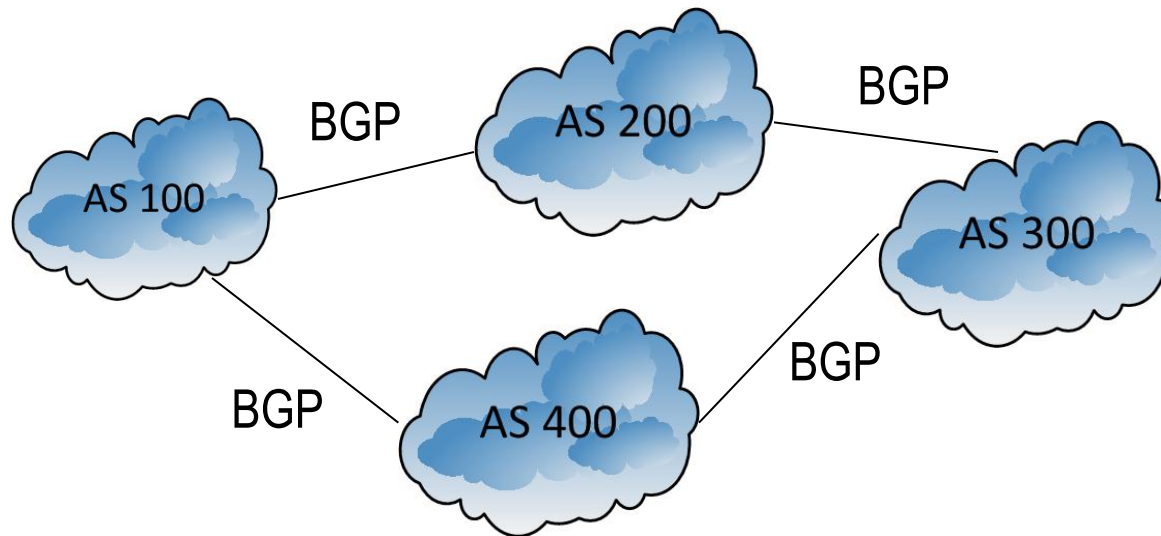
According to RFC 1930, an autonomous system (AS) is a collection of connected Internet Protocol (IP) routing prefixes under the control of one or more network operators that presents a common, clearly defined routing policy to the Internet.



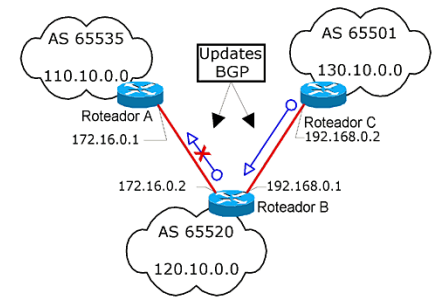
Each AS has a exclusive number that is designated by IANA and Regional Registry entities (RIPE for Europe, LACNIC for Latin America and Caribbean, etc). AS numbers from 64512 through 65535 are reserved for private AS's.

Autonomous System (AS), the Internet and BGP protocol

The Internet is nothing more than a set of interconnected AS's, each one under a distinct technical administration.



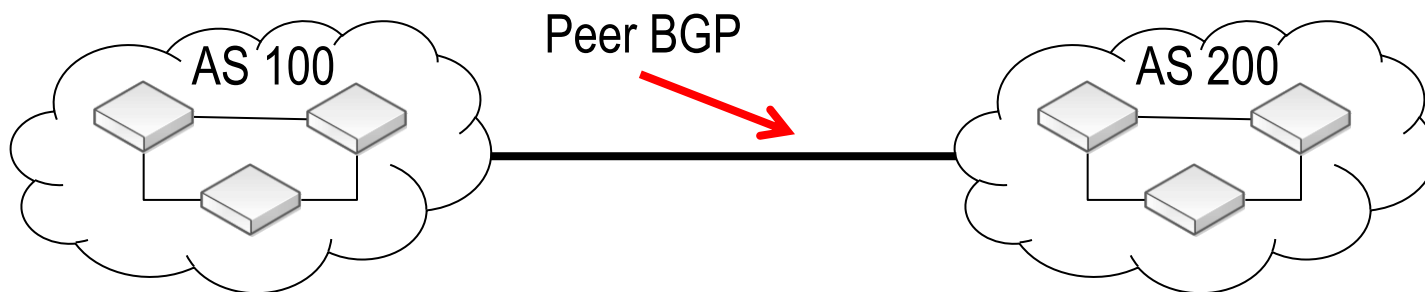
BGP is the protocol that glues all those AS's forming a huge net that should work well even under actions of thousands of administrators from allover the world.



BGP characteristics:

- BGP is a “distance vector” protocol .
- Current version is v4, according to RFC 1771.
- Network prefixes are announced with a list of the AS's that are in the path to reach such prefixes.
- Internal topology of the AS doesn't matter, but only information on how to reach the prefixes (AS path and next hop)

Peering BGP

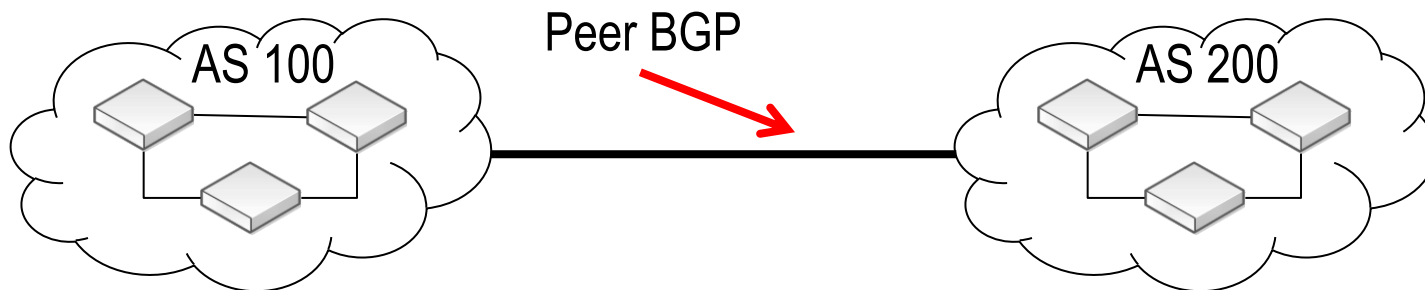


BGP peerings are configured statically by both AS administrators.

To ensure a reliable communication, between the peers BGP protocol relies on TCP protocol, port 179.

The first message is an **OPEN** and once a the peering is established the AS's exchange routes information.

Peering BGP

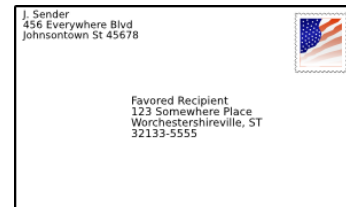


Once the database is complete on both sides, only modifications on BGP routes (new routes and withdrawn routes) are informed to the other side via **UPDATE** messages.

To ensure that the neighbor is “alive” periodically BGP peers send **KEEPALIVE** messages, waiting for reply.

In case of failure, BGP will send a **NOTIFICATION** message.

BGP messages



Common header

| Field | Length (bytes) |
|--------|----------------|
| Marker | 16 |
| Length | 2 |
| Type | 1 |

OPEN message

| Field | Length (bytes) |
|---------|----------------|
| Marker | 16 |
| Length | 2 |
| Type | 1 |
| Message | 0..4077 bytes |

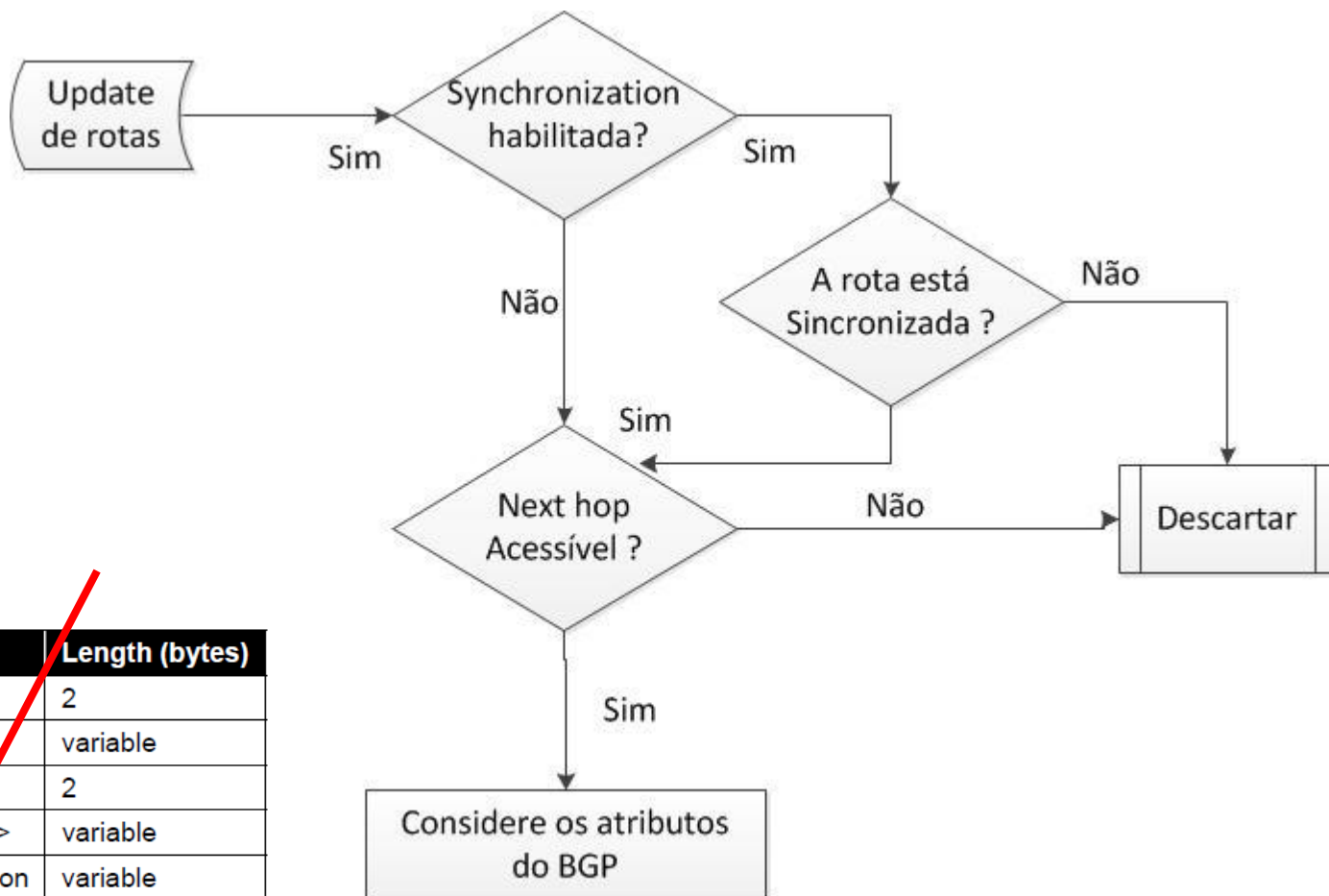
UPDATE message

| Field | Length (bytes) |
|--|----------------|
| Withdrawn routes length | 2 |
| Withdrawn routes | variable |
| Path attributes length | 2 |
| Path attributes <type, length, value> | variable |
| Network layer reachability information | variable |

NOTIFICATION message

| Field | Length (bytes) |
|---------------|----------------|
| Error code | 1 |
| Error subcode | 1 |
| Length | 2 |
| Data | Variable |

How does BGP select the best path ?



UPDATE message

| Field | Length (bytes) |
|--|----------------|
| Withdrawn routes length | 2 |
| Withdrawn routes | variable |
| Path attributes length | 2 |
| Path attributes <type, length, value> | variable |
| Network layer reachability information | variable |

BGP attributes for best path selection on Mikrotik RouterOS

- 1) Prefer the path with highest WEIGHT (default = 0)
- 2) Prefer the path with highest LOCAL_PREF (default = 100)
- 3) Prefer the path with Shortest AS_PATH
- 4) Prefer the path that was locally originated via [aggregate](#) or [BGP network](#)
- 5) Prefer the path with lowest Origin (IGP, EGP, incomplete)
- 6) Lower MED (default = 0)
- 7) Prefer eBGP over iBGP paths
- 8)

BGP Attributes

BGP attributes are an array of information carried by UPDATE messages. The attributes could be:

Well-known (must be recognized by all implementations)

- Mandatory (should be present on all UPDATE messages)
- Discretionary

Optionals

- Transitive (should be passed to other routers, even if unrecognized)
- Intransitive

BGP attributes and security

For the purposes of this presentation, we'll focus only on the below well-known attributes:

→ **AS_Path**

Sequence of AS numbers that should be passed to access some network.

→ **Next_Hop**

IP address of the router that information should be forwarded.

And an optional, transitive attribute:

→ **BGP Communities**

Attributes AS_Path and Next_Hop



AS_Path

AS-Path attribute is empty when a local route is inserted on BGP table.

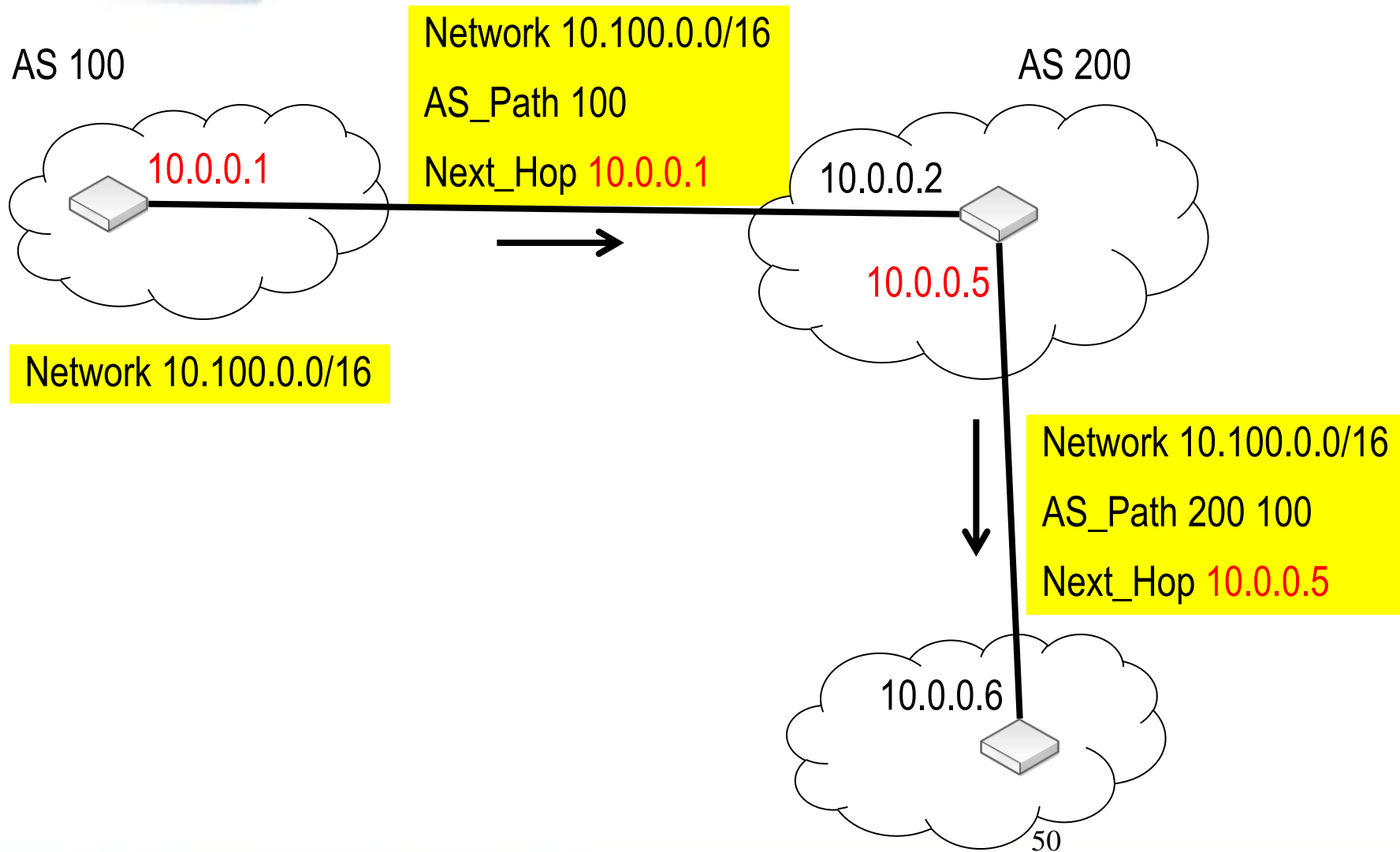
Sender's AS number is prepended to the AS_Path, every time an update crosses the boundary of a AS.

Looking at the contents of AS_Path attribute, a recipient knows the list of AS's the update has passed.

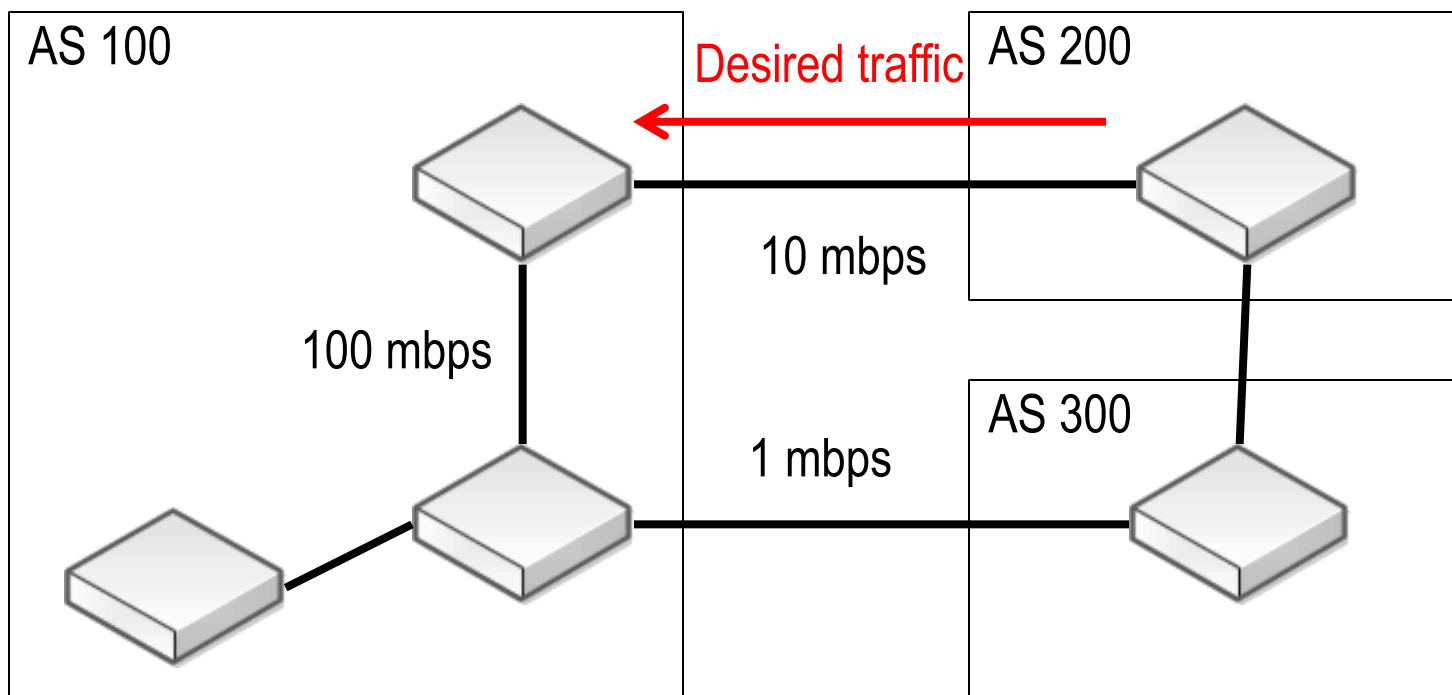
Next_Hop

In external BGP the sender inform the recipient your IP address, as the next hop for delivery packets.

AS_Path and Next_Hop attributes

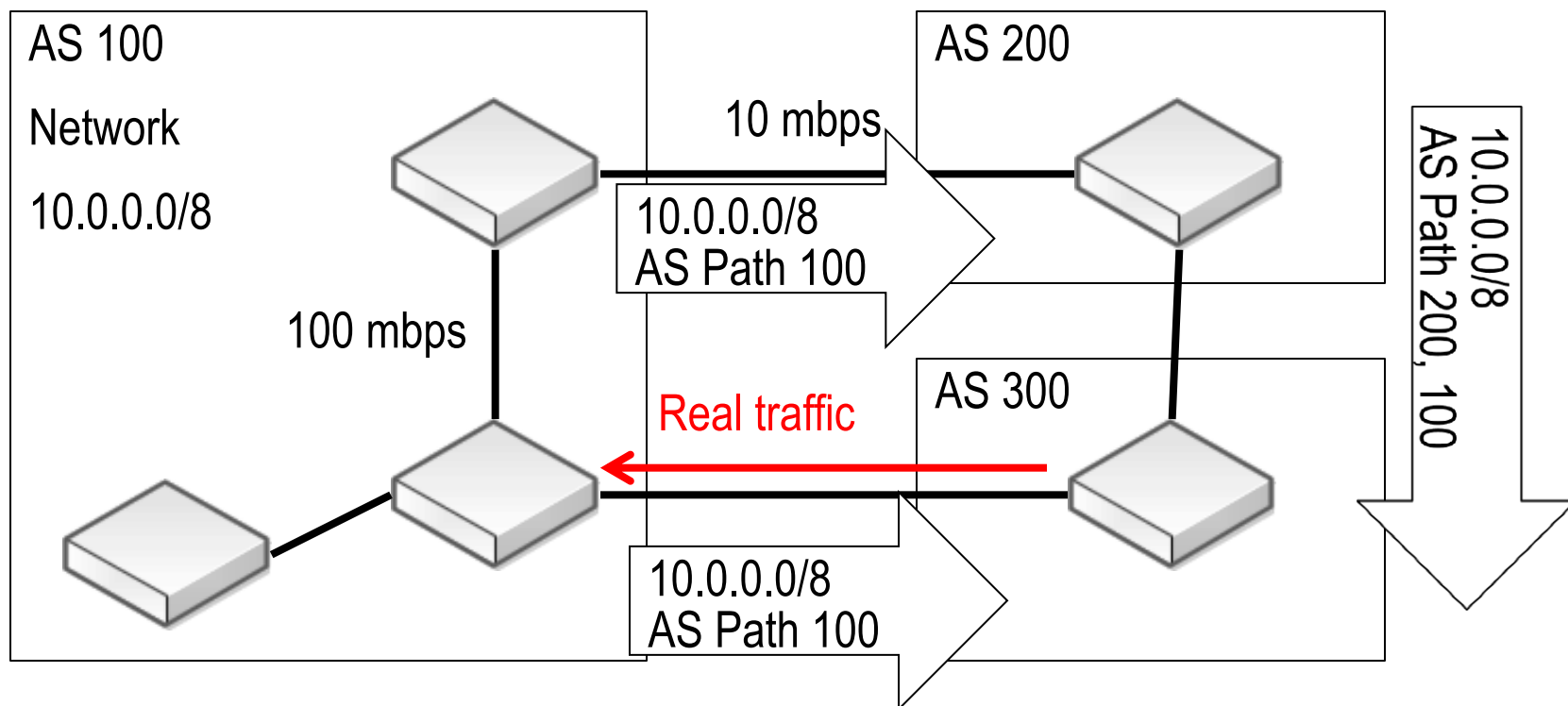


AS-Path attribute



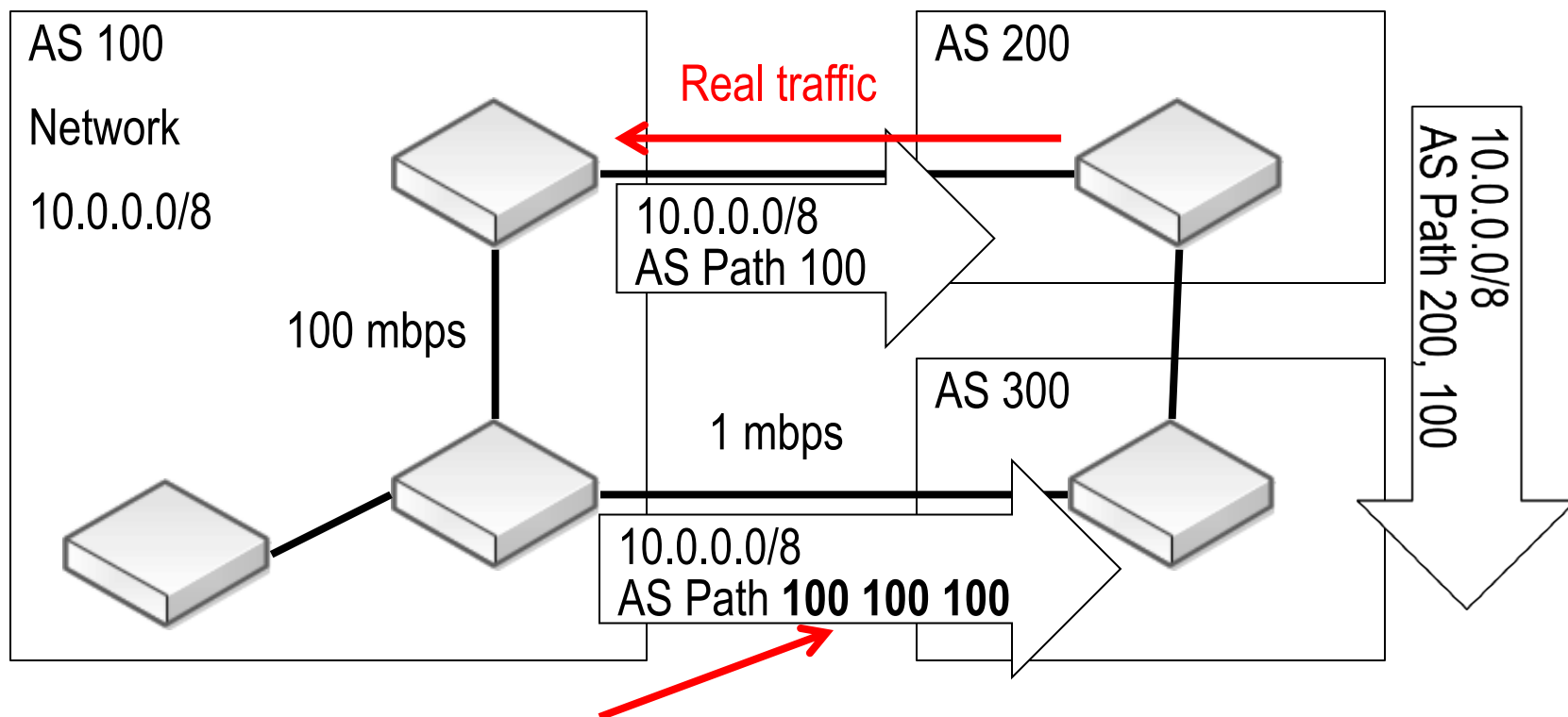
Suppose the above situation.

AS-Path attribute



AS 300 sees two paths to network 10.0.0.0/8, the “shortest” is directly through AS 100 because there is only one AS in the path. Via AS 200 there are 2 AS’s.

AS-Path prepending



AS 100 prepends two times its own AS number.

Now AS 300 sees the shortest path (2 hops) through AS200.

BGP Communities

Community is a powerful attribute widely used to implement routing policy. Administrators can manipulate BGP communities based on their own network policies.

Communities are described in RFC 1997 as they are a transitive & optional attribute. Therefore, they can travel to different Autonomous Systems.

A Community is a 32 bit integer represented as two 16 bit numbers. There are some well-known communities, like:

- **no-export** 65535:65281 (do not advertise to any eBGP peers)
- **no-advertise** 65535:65282 (do not advertise to any BGP peer)
- **no-peer** 65535:65284 (do not advertise to bi-lateral peers (RFC3765))

BGP Communities

AS administrators can however, define a set own communities to advertise to the external world their local policies.

→ **XXXXX:YYYYY**, where

→ XXXXX is the AS number that is defining the community

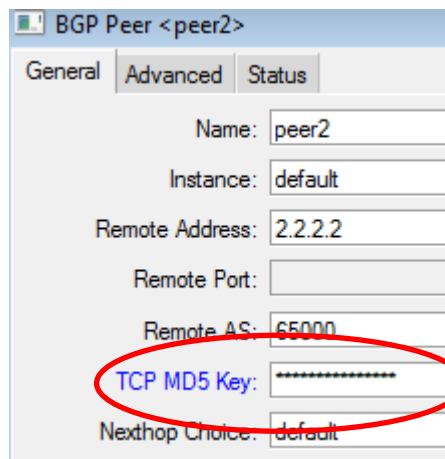
→ YYYYY: an arbitrary number that define some local policy.

→ e.g. The administrator of AS 28657 advertises to his peers that if he receives the community 28657:8888, he will put the receiving prefix as blackhole. Someone that wants to put some prefix as blackhole in AS 28657, advertises the prefix appending the community 28657:8888

BGP security features on Mikrotik RouterOS

Authentication:

Mikrotik RouterOS provides authentication by means of a pre shared MD5 key, configured on both peers



BGP Peer <peer2>

General Advanced Status

Name: peer2

Instance: default

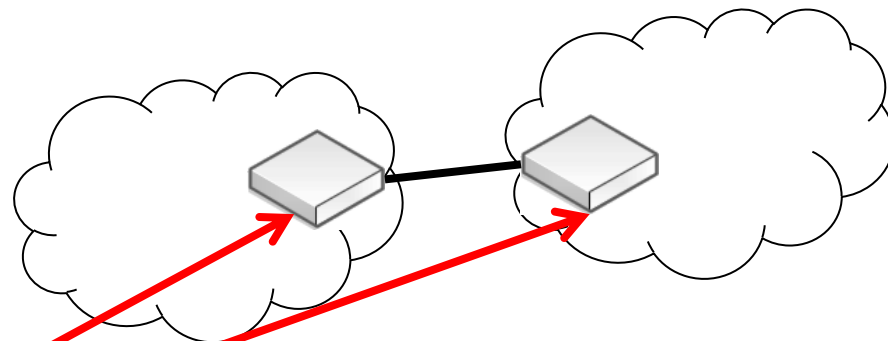
Remote Address: 2.2.2.2

Remote Port:

Remote AS: 65000

TCP MD5 Key: *****

Next hop Choice: default



BGP security features on Mikrotik RouterOS

BGP TTL “hack”

This simple configuration tries to ensure that one Router will communicate only with other that is n hops away.

BGP Peer <peer1>

General Advanced Status

Name: peer1

Instance: default

Remote Address: 1.1.1.1

Remote Port:

Remote AS: 65530

TCP MD5 Key:

Nexthop Choice: default

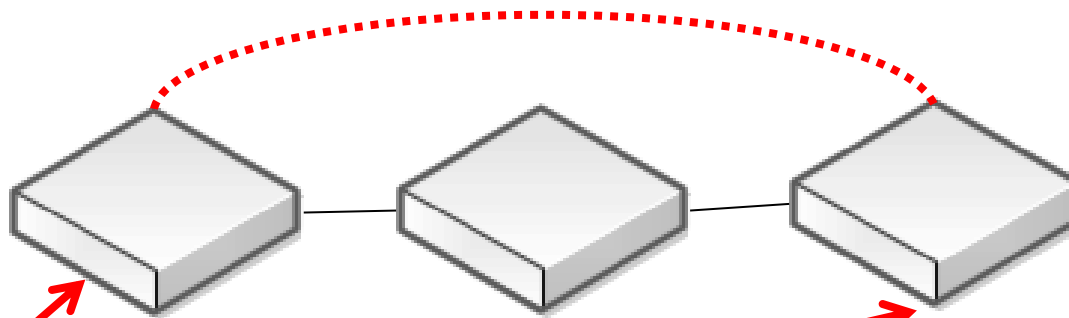
☒ Multihop

☐ Route Reflect

Hold Time: 180

TTL: 2

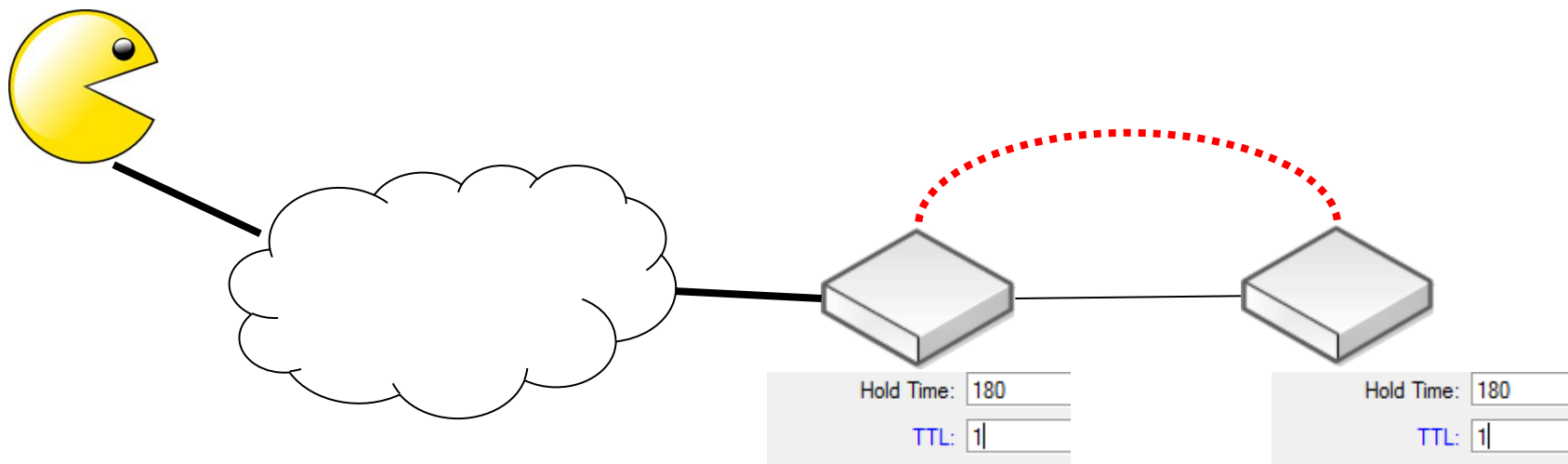
BGP Session



BGP security features on Mikrotik RouteOS

BGP TTL “hack”

However, with this implementation an attacker can guess how many hops he/she is away from the penultimate router and forge a packet that will arrive with the appropriate TTL.



Suggestion to Mikrotik guys:

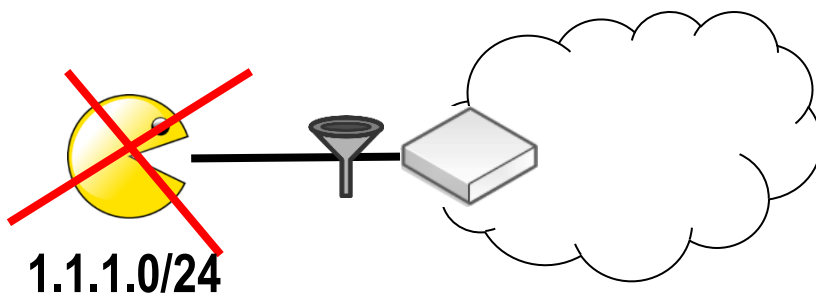
A more secure implementation could be as stated in RFC 3682 – TTL expected = 255



Routing Filters

- Routing Filters are not related only for security, but they are the main tool to manipulate BGP attributes and thus establishing a routing policies.
- Routing filters are used to prevent undesirable announces to enter/leave the network
- Filters are organized in channels, like the Firewall.
- Filters are applied to peers for incoming and or outgoing BGP routing updates.

Example – Filtering Prefixes



New Route Filter

| Matchers | BGP | Actions | BGP Actions |
|----------------|--------------------------|------------|-------------|
| Chain: | undesirable_prefix | | |
| Prefix: | <input type="checkbox"/> | 1.1.1.0/24 | |
| Prefix Length: | <input type="checkbox"/> | 24-32 | |

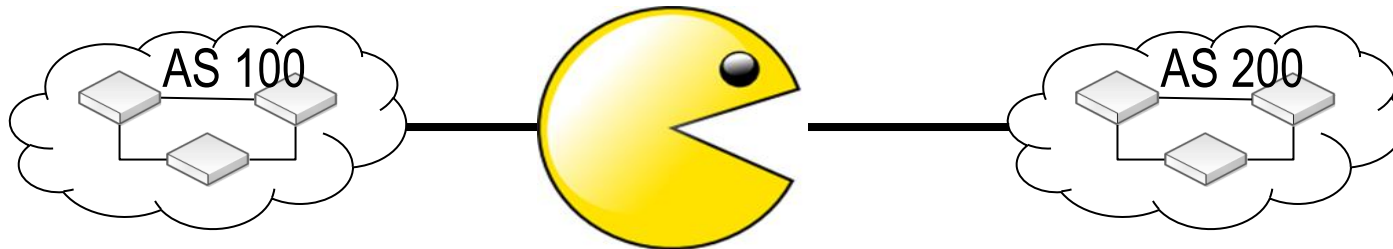
New Route Filter

| Matchers | BGP | Actions | BGP Actions |
|----------|---------|---------|-------------|
| Action: | discard | | |

BGP Peer <peer1>

| General | Advanced | Status |
|--|--|--------|
| Name: | peer1 | |
| Instance: | default | |
| Remote Address: | 1.1.1.1 | |
| Remote Port: | | |
| Remote AS: | 65530 | |
| TCP MD5 Key: | | |
| Nexthop Choice: | default | |
| <input checked="" type="checkbox"/> Multihop | <input type="checkbox"/> Route Reflect | |
| Hold Time: | 180 | |
| TTL: | 2 | |
| Max Prefix Limit: | | |
| Max Prefix Restart Time: | | |
| In Filter: | undesirable_prefix | |

Attacks against BGP



Attacking BGP session

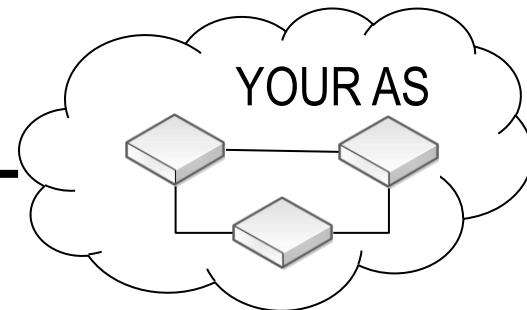


200.1.0.0/20 189.1.0.0/19 170.1.0.0/16

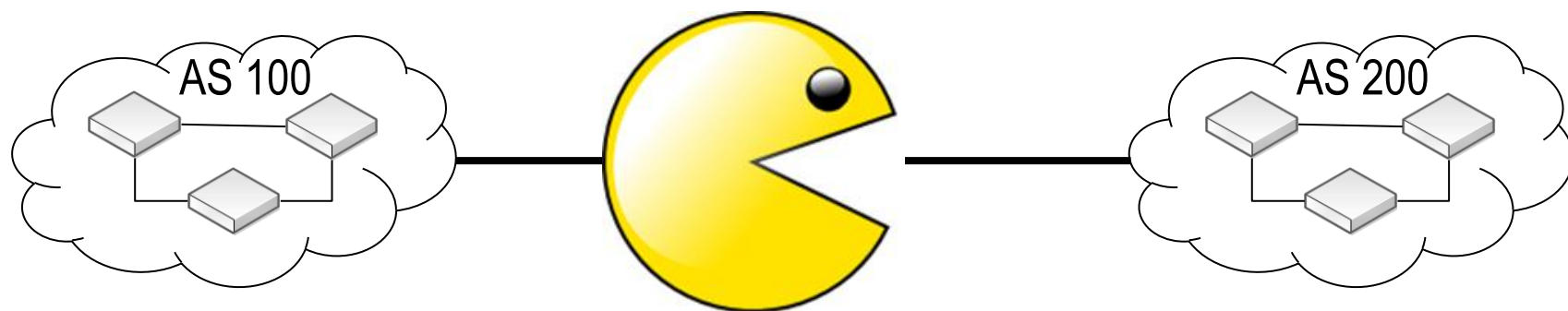
Hijacking prefixes



**Misconfigurations
and garbage**



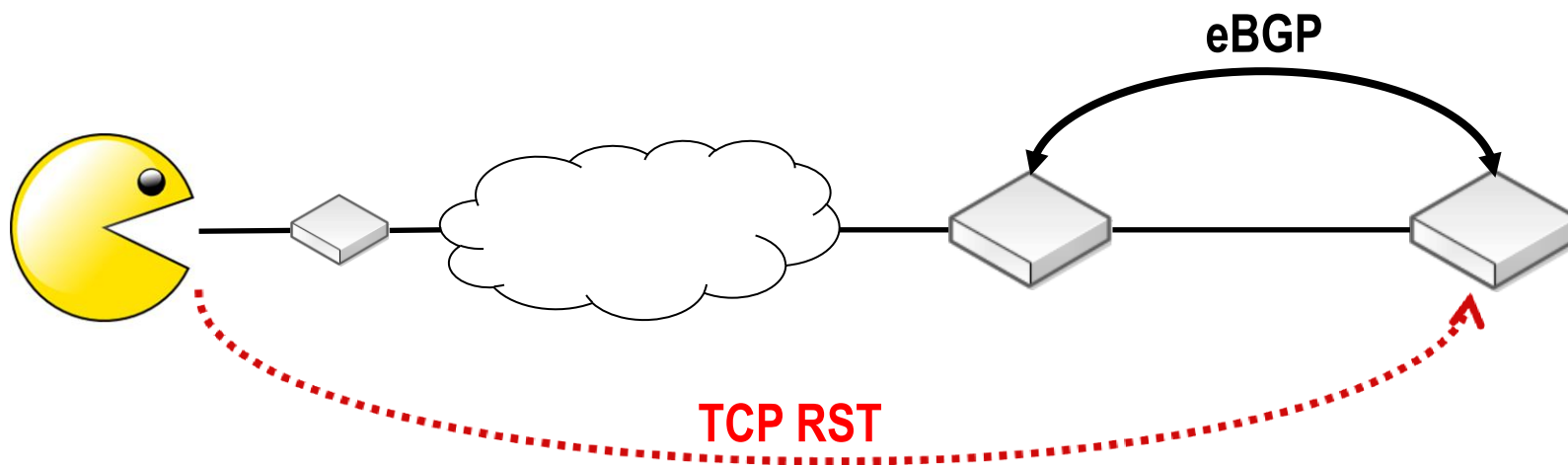
Attacking the BGP session



Attacks against BGP

1) Peer spoofing and TCP resets

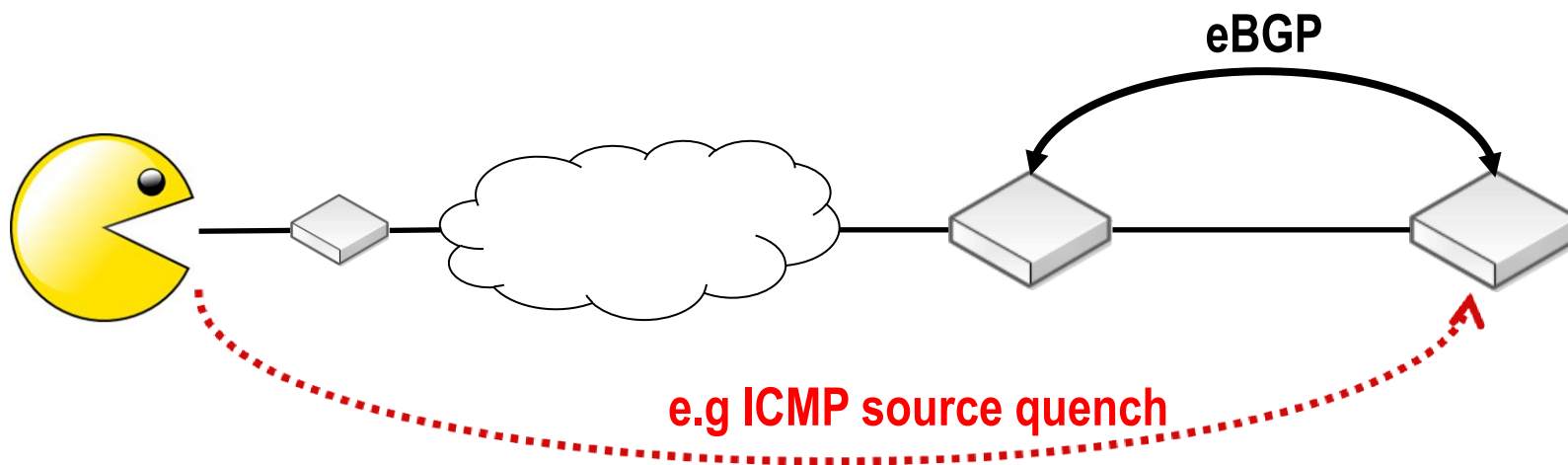
- Consists on injecting TCP RESET message on an ingoing BGP session
- Requires previous monitoring of the TCP session to gain enough information (peer's IP addresses, sequence numbers, etc)



Attacks against BGP

2) TCP resets using ICMP

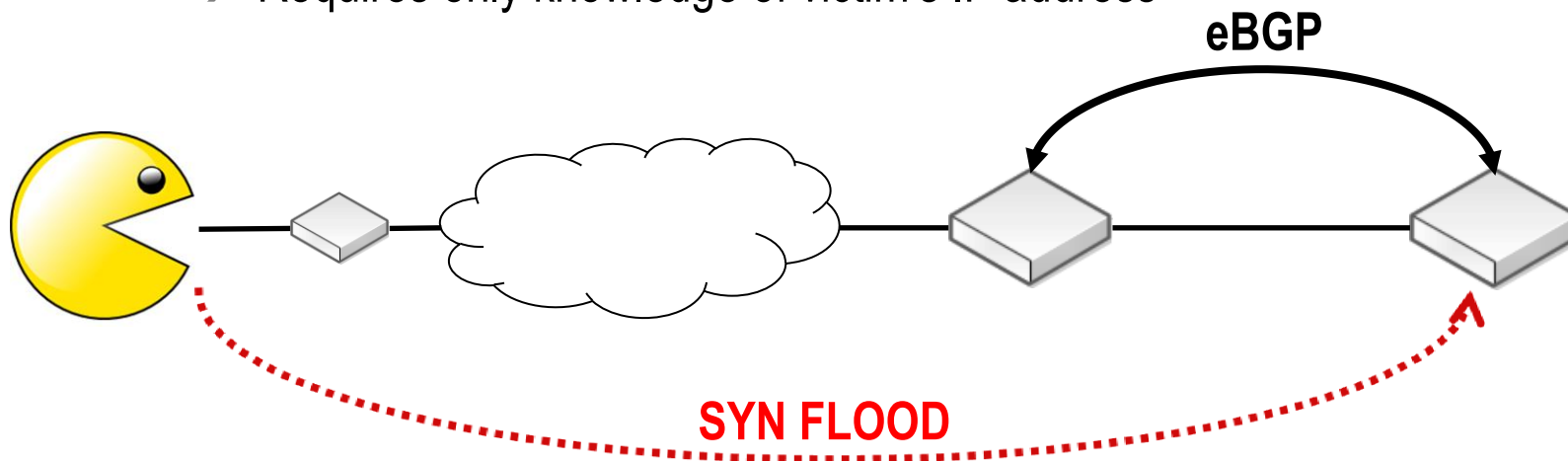
- ICMP messages can be used to produce resets without the knowledge of TCP session, but only victim's IP address and port number
- ICMP messages of hard error will cause resets while soft error will produce throughput degradation



Attacks against BGP

3) Syn flood attack

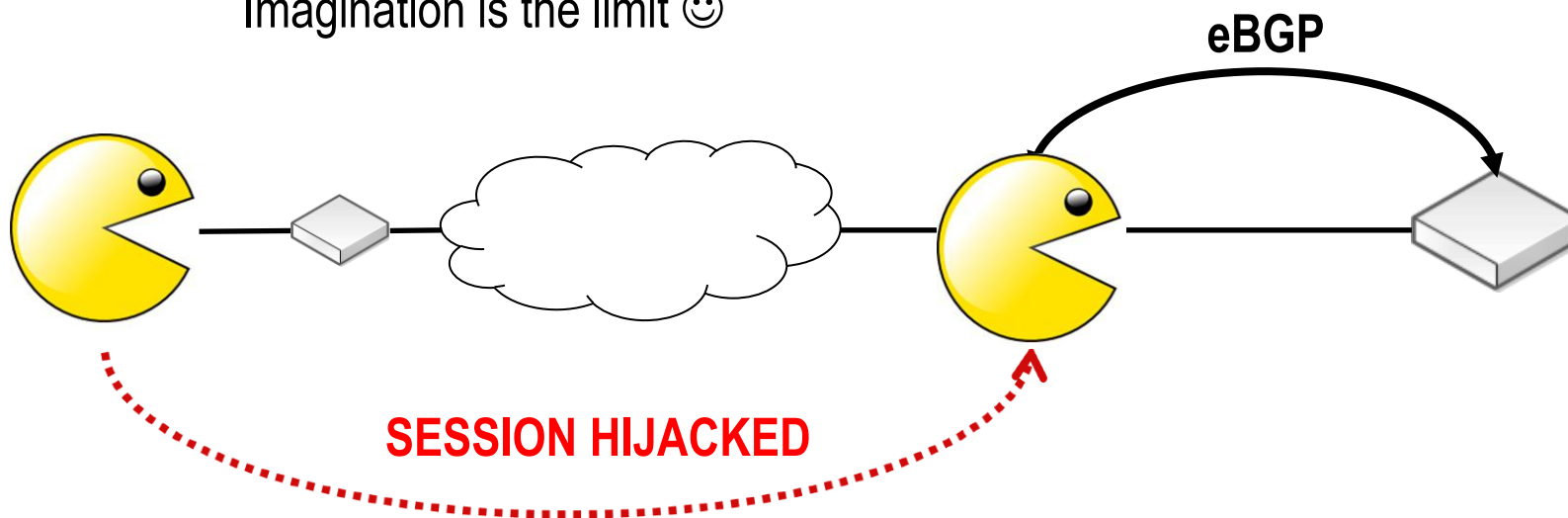
- Consists on sending tons of SYN packets by an attacker pretending to start a TCP session.
- Victim will reply with SYN ACK + its own SYN waiting for SYN ACK from attacker and resources can be exhausted.
- Requires only knowledge of victim's IP address



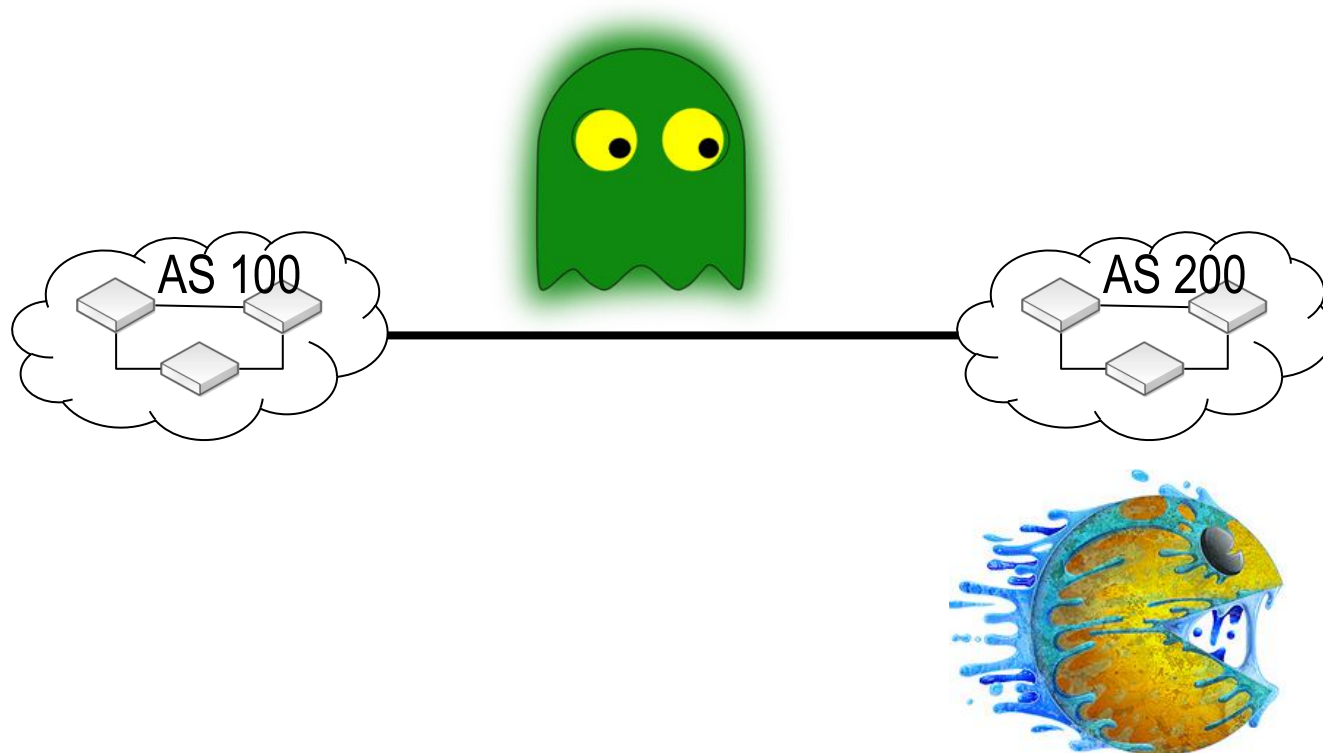
Attacks against BGP

4) Session Hijacking

- Like on TCP resets, attacker should gather information of ongoing BGP sessions.
- With enough information he/she can impersonate a peer, sending prefix updates or any BGP message.
- In this situation he/she can cause eavesdropping, blackholing, etc. Imagination is the limit 😊



Protecting BGP session



Protecting BGP session

There is not only one measure to ensure security of the BGP session, but a “cocktail” of them.

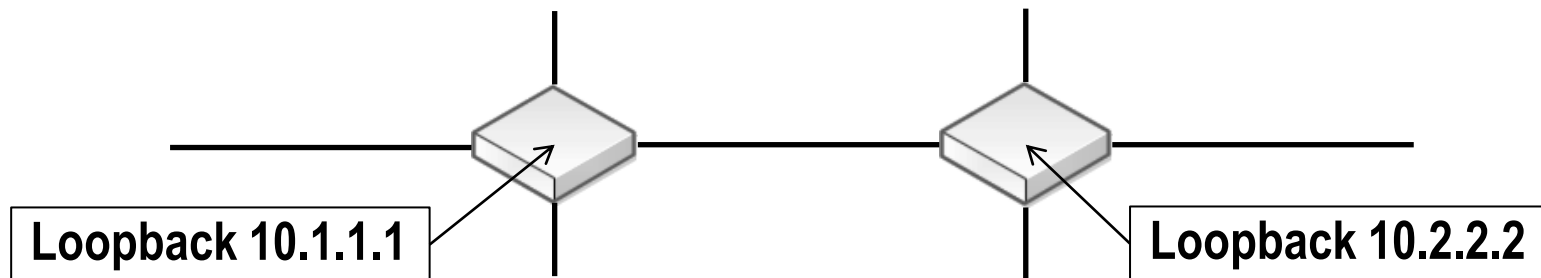
- 1) Use authentication with a strong password
- 2) Use TTL hack
- 3) Use loopback interfaces for BGP peering (Why ? – see next 2 slides)

Think about other measures:

- 4) In case of SYN flood attack, enable SYN cookies on the firewall
- 5) Ensure a bandwidth to your TCP connection with some QoS technique
- 6) If you (and your neighbor AS) are paranoid use IPSec ☺

Loopback addresses

Loopback addresses eliminate the dependency of physical interfaces ensuring that even when one interface goes down, the router could be reachable by other one. Using loopback interfaces is mandatory for a good iBGP or OSPF setup.



eBGP does not rely on loopback interfaces to work properly.

Why should be then the use of loopback interfaces for eBGP considered a good practice ?

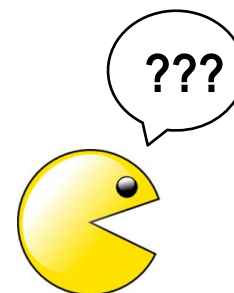
Why use loopback addresses even for eBGP connections ?

1) For balancing purposes (when you have more than one physical link)

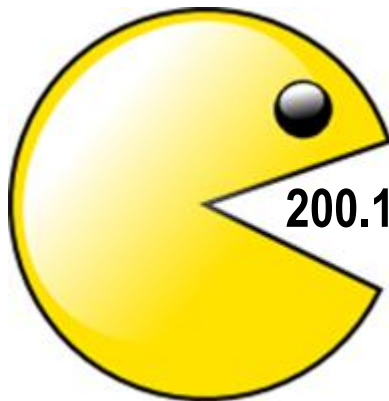


2) For security reasons:

- TTL hack and authentication are not enough
- Attacks against BGP session need IP's and ports
- One port is well known (179) and interfaces IP's can easily be discovered
- Using loopback addresses with arbitrary one will turn things more difficult for attacker.



Prefixes Hijacking



200.1.0.0/20

189.1.0.0/19

170.1.0.0/16

Attacks against BGP

Unallocated Prefix announcements

- Anyone that owns an AS (or has gained control over one by compromising a BGP router), can announce any prefix !
- Yes, in this moment someone could be announcing your IP blocks and there is nothing you can do to avoid this.
- Announcing the same prefix will lead to a partial route hijacking

Attacks against BGP

Prefix hijacking by route de-aggregation

- To completely hijack the prefix, attackers will announce more specific prefixes (longer bitmasks)
- More specific routes mean optimal paths and will be chosen. BGP will widespread them to another peers, all over the Internet.

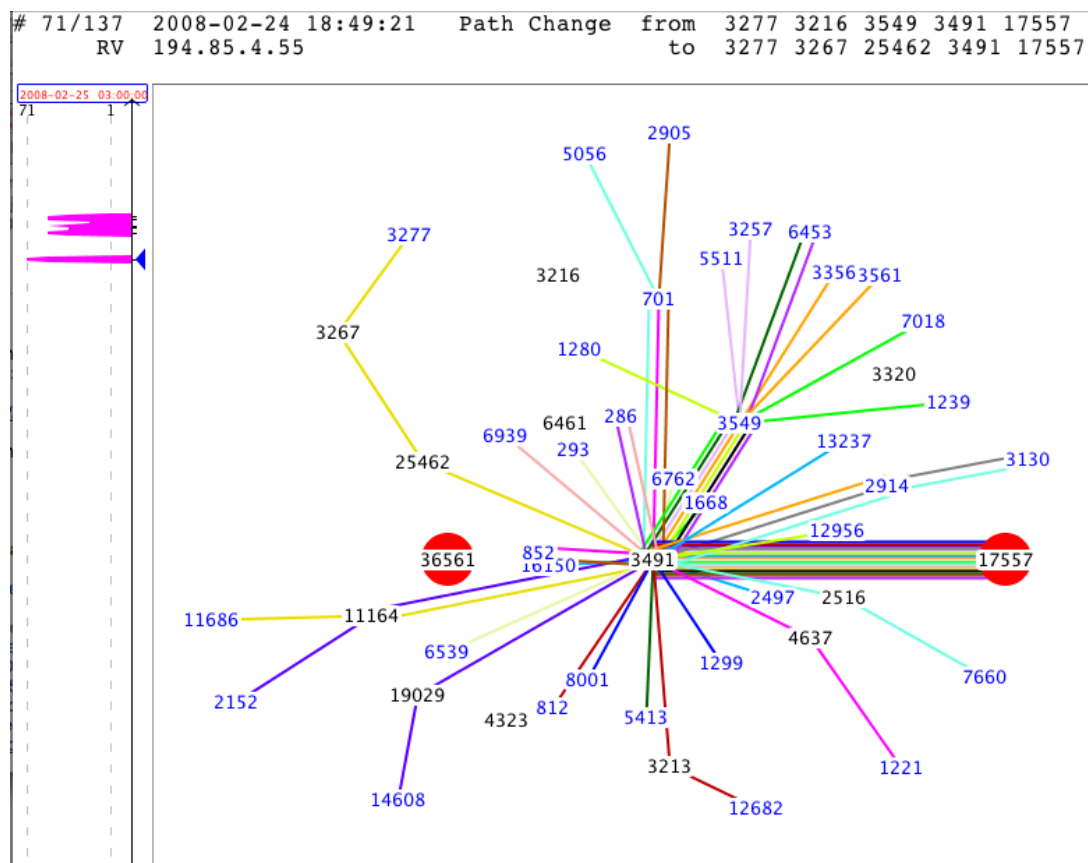
This issue is not new .

- 1997 – The first public problem officially reported
- 2008 - YOUTUBE x Pakistan Telecom

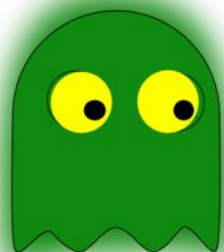


Attacks against BGP

Prefix hijacking by route de-aggregation



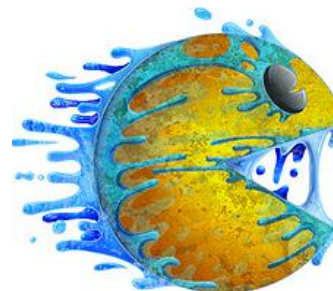
Can we do anything about prefix hijacking ?



200.1.0.0/20

189.1.0.0/19

170.1.0.0/16



Can we do anything about
prefix hijacking ?

Not much ☹

What we can really do is to have some good practices, like:

1) **Subscribe your AS to IRR:**

“The Internet Routing Registry (IRR) is a distributed routing database development effort. Data from the Internet Routing Registry may be used by anyone worldwide to help debug, configure, and engineer Internet routing and addressing. The IRR provides a mechanism for validating the contents of BGP announcement messages or mapping an origin AS number to a list of networks.



**List of Routing
Registries**


**How to Register
a NEW Registry**

OverView RPSL FAQ Home

<http://www.irr.net/>

Can we do anything about prefix hijacking ?

2) Monitor your prefixes (and much more) with BGPMon



maia@mdbrasil.com.br

- My BGPMon
- My Alerts
- My Prefixes
- My ASn
- My Ignore list

Bogon Analyses

- Bogon AS Announcements
- IPv4 Bogon prefixes
- IPv6 Bogon prefixes

- IPv4 BGP weathermap
- IPv6 BGP weathermap
- Statistics

- BGP
- 6to4
- Tere
- Long

My Alerts

Include filter

Select AS: AS28657: MD Brasil Tecnologia da Informação Ltda (0 alerts) ▼

Alarm type: All alarm types (0) ▼ Activity: All ▼

Show my Updates

Delete

Delete all AS28657 updates older then: 30 days Delete

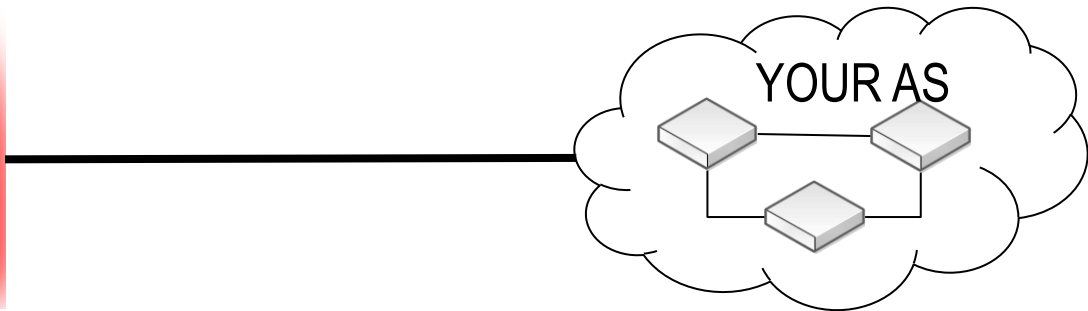
Delete all AS28657 updates with: code: 22 Delete

Delete selected alerts

| <input type="checkbox"/> | View Details | Mark as False Positive | Alert Type | seen by #peers | update time (UTC) | monitored network | Announced prefix | Origin AS | transit AS | Regex ASpath mismatch | Active |
|--------------------------|--------------|------------------------|------------|----------------|-------------------|-------------------|------------------|-----------|------------|-----------------------|--------|
| --Page 1 of 1-- | | | | | | | | | | | |
| ---- | | | | | | | | | | | |

<http://www.bgpmon.com/>

Misconfigurations from other administrators and garbage in general

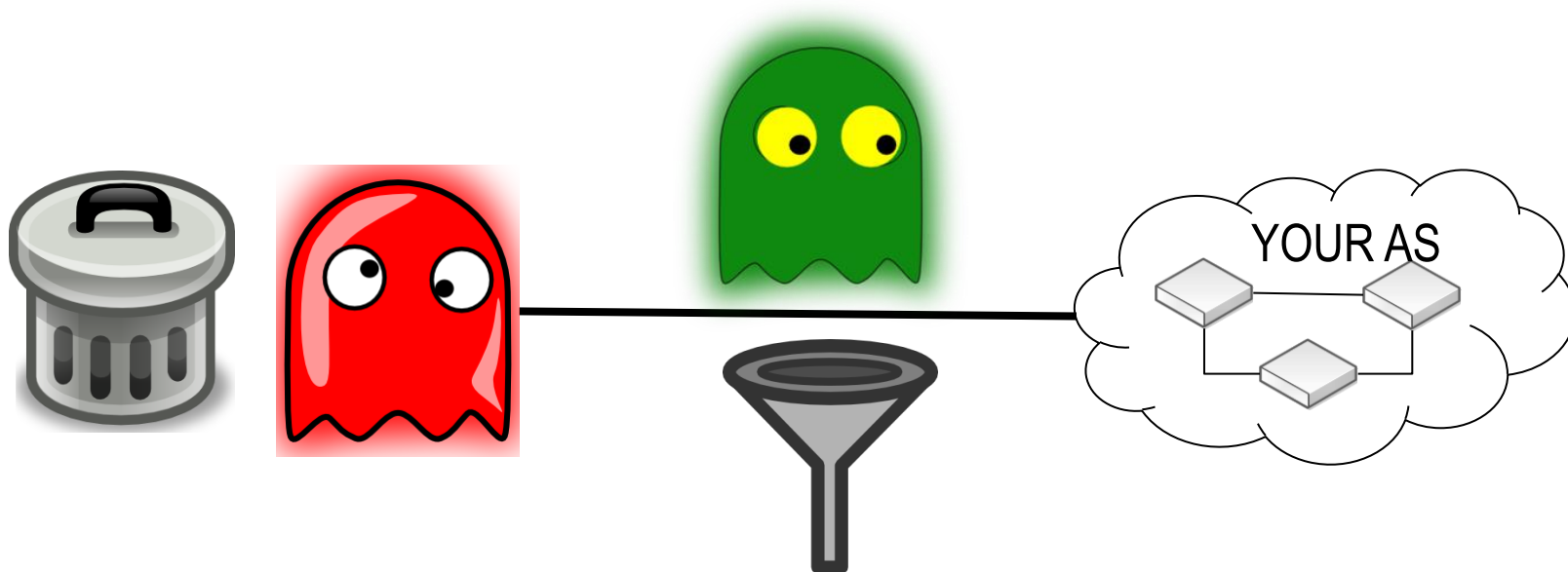


Common misconfigurations and garbage

Common misconfigurations and garbage that can affect you:

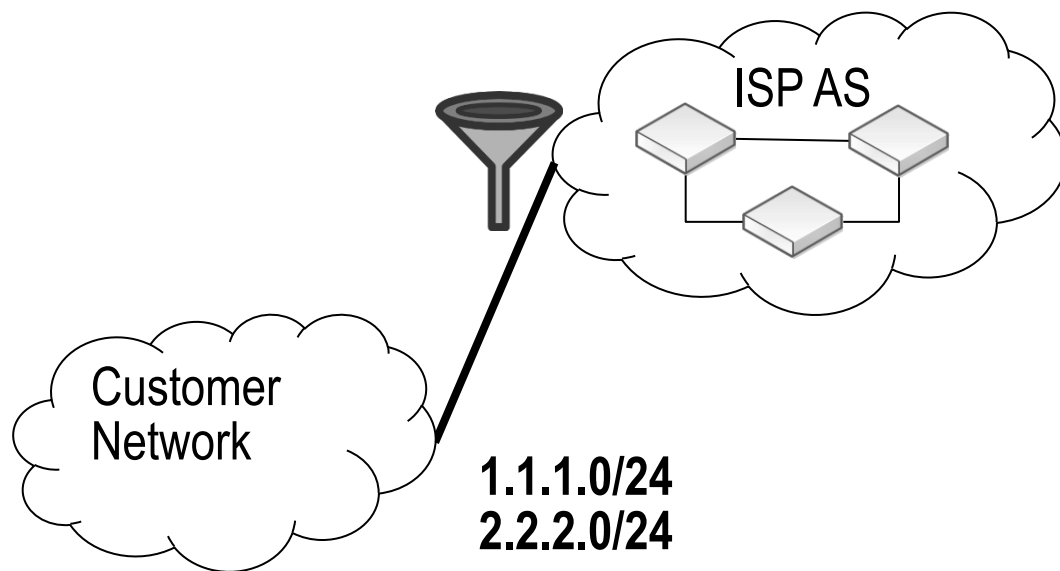
- Someone , anywhere is announcing to you your own prefix
- Someone , anywhere is announcing to you prefixes owned by or allocated to your Customers
- Someone, anywhere is sending too long AS_Path's
- Your peer is starving you sending tons of prefixes
- Your upstream provider is sending you private/reserved prefixes
- Your upstream provider is sending you BOGON prefixes

Preventing misconfigurations from other administrators and getting rid of garbage



Receiving Prefixes from Customers

- ISPs should only accept prefixes which have been assigned or allocated to their downstream customer
- If the ISP has NOT assigned address space to its customer, then check in the RIR databases to see if this address space really has been assigned to the customer



| Matchers | BGP | Actions | BGP Actions |
|---|-----|---------|-------------|
| Chain: in-filters | | | |
| Prefix: <input type="checkbox"/> 1.1.1.0/24 | | | |
| Prefix Length: <input type="checkbox"/> 24-32 | | | |

| Matchers | BGP | Actions | BGP Actions |
|---|-----|---------|-------------|
| Chain: in-filters | | | |
| Prefix: <input type="checkbox"/> 2.2.2.0/24 | | | |
| Prefix Length: <input type="checkbox"/> 24-32 | | | |

| Matchers | BGP | Actions | BGP Actions |
|----------------|-----|---------|-------------|
| Action: accept | | | |

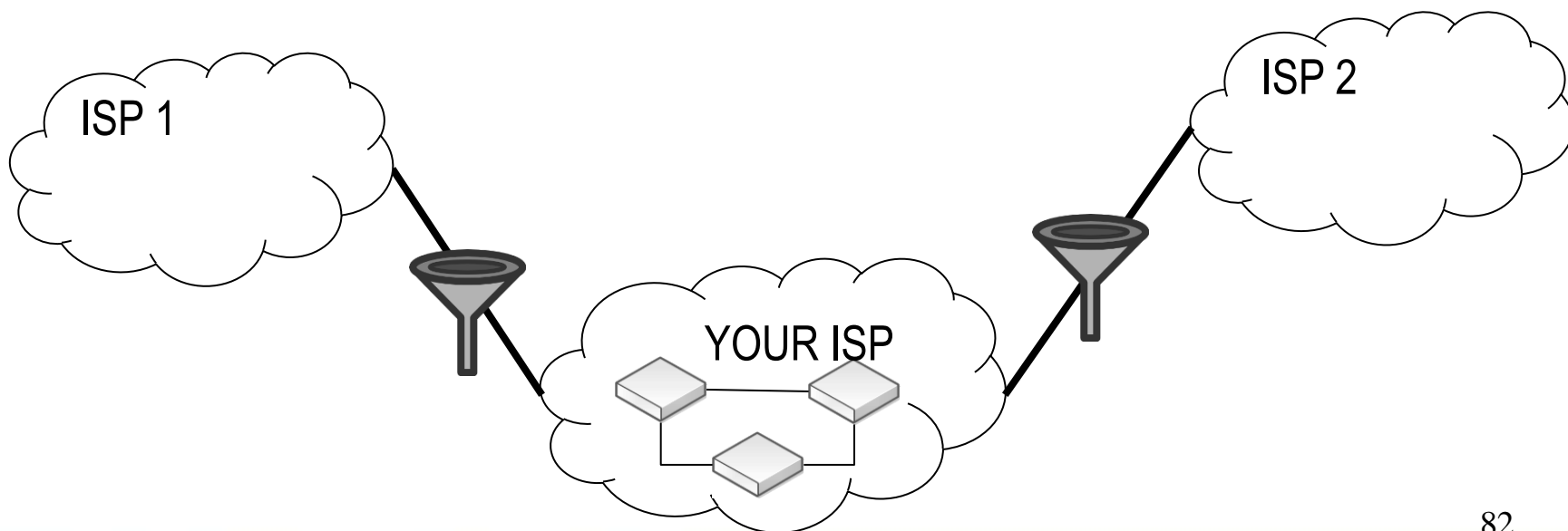
| Matchers | BGP | Actions | BGP Actions |
|-------------------|-----|---------|-------------|
| Chain: in-filters | | | |

| Matchers | BGP | Actions | BGP Actions |
|-----------------|-----|---------|-------------|
| Action: discard | | | |

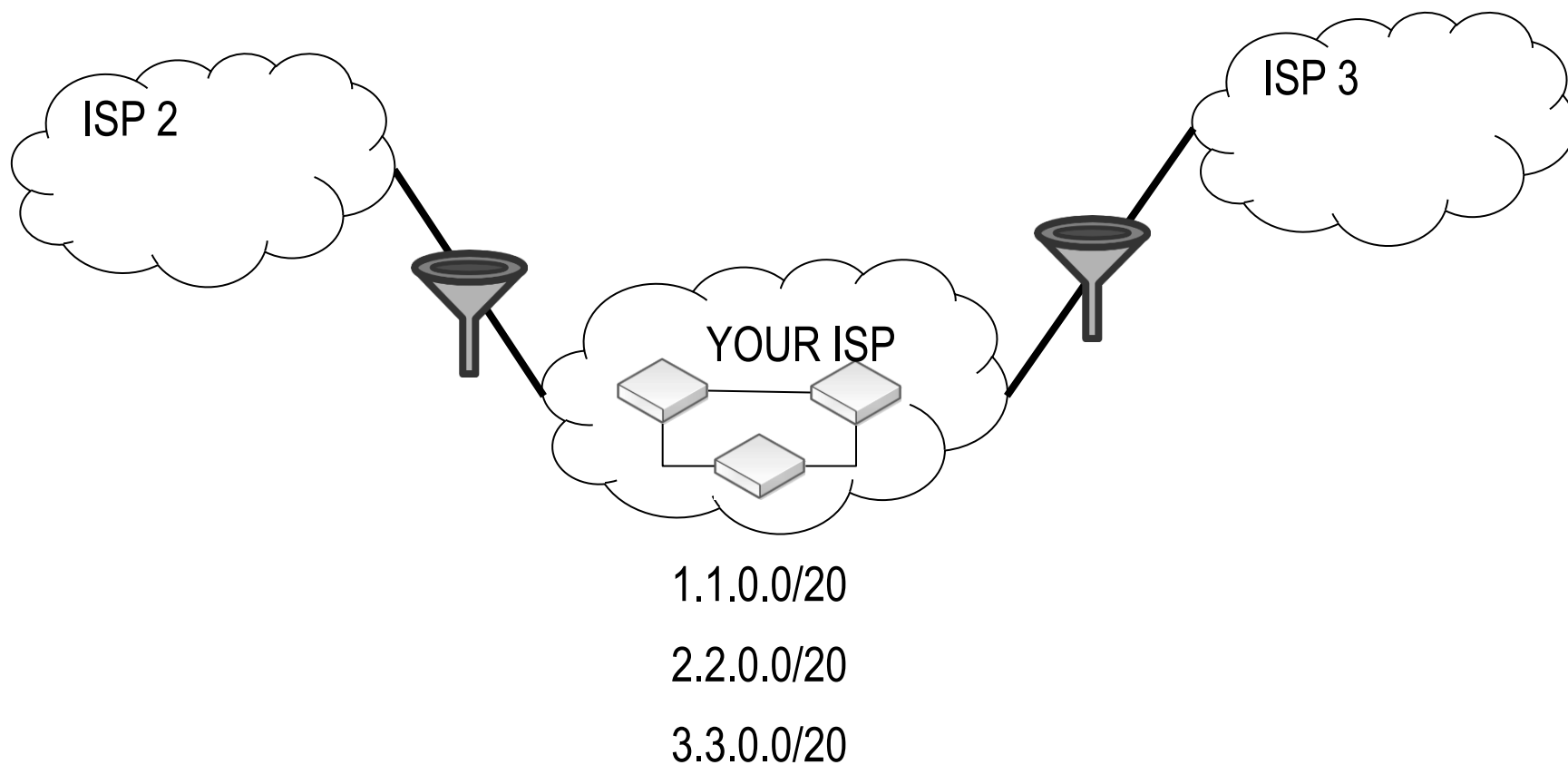
Receiving Prefixes from Peers

- A peer is an ISP with whom you agree to exchange some prefixes.
- Prefixes you accept from a peer are only those they have indicated they will announce
- Prefixes you announce to your peer are only those you have indicated you will announce

If you are not a transit provider, take care to no become one !

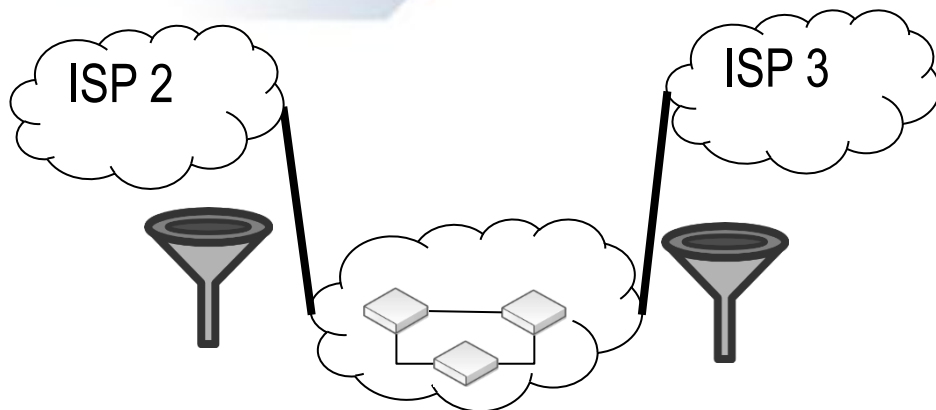


Receiving Prefixes from Peers



Best Common Practices

Filtering examples



YOUR ISP

Owned Prefixes:

1.1.0.0/20

2.2.0.0/20

3.3.0.0/20

In Filters

- Don't accept your own prefixes
- Don't accept RFC 1918 private address and other reserved ones (RFC 5735)
- Don't accept default (unless you need it)
- Don't accept prefixes longer than /24
- Don't accept BOGONS prefixes
- Limit your Max Prefix
- Limit AS_Path







Out Filters

- Announce only owned prefixes (in case you do not provide transit to other AS's)







Best Common Practices

Filtering examples

Discard receiving own prefixes

|       | | | | | | |
|---|--------------------|------------|---------------|----------|-------------|---------|
| # | Chain | Prefix | Prefix Length | Protocol | BGP AS Path | Action |
| ... All prefixes owned by the provider should be listed here | | | | | | |
| 0 | own_prefix_discard | 1.1.0.0/20 | 20-32 | | | discard |
| 1 | own_prefix_discard | 2.2.0.0/20 | 20-32 | | | discard |
| 2 | own_prefix_discard | 3.3.0.0/20 | 20-32 | | | discard |

Discard default route

|       | | | | | | |
|--|-----------------------|-----------|---------------|----------|-------------|---------|
| # | Chain | Prefix | Prefix Length | Protocol | BGP AS Path | Action |
| ... Reject_Default_Route | | | | | | |
| 22 | default_route_disc... | 0.0.0.0/0 | | | | discard |

Best Common Practices

Filtering examples

Longer Bitmask discard

| # | Chain | Prefix | Prefix Length | Protocol | BGP AS Path | Action | |
|----|---------------------|--------|---------------|----------|-------------|---------|--|
| 23 | Longer_Bitmask_d... | | 25-32 | | | discard | |

Limiting prefixes received

| | | |
|--------------------------|---|-----|
| Max Prefix Limit: | <input type="text" value="3700000"/> | ▲ |
| Max Prefix Restart Time: | <input type="text" value="0 (infinity)"/> | ▼ ▲ |

NB: Not a filter, but a configuration on peers

Best Common Practices

Filtering examples

Announcing only owned prefixes

| # | Chain | Prefix | Prefix Length | Protocol | BGP AS Path | Action |
|----|---------------------|------------|---------------|----------|-------------|---------|
| 25 | announcing_only_... | 1.1.0.0/20 | | | | accept |
| 26 | announcing_only_... | 2.2.0.0/20 | | | | accept |
| 27 | announcing_only_... | 3.3.0.0/20 | | | | accept |
| 28 | announcing_only_... | | | | | discard |

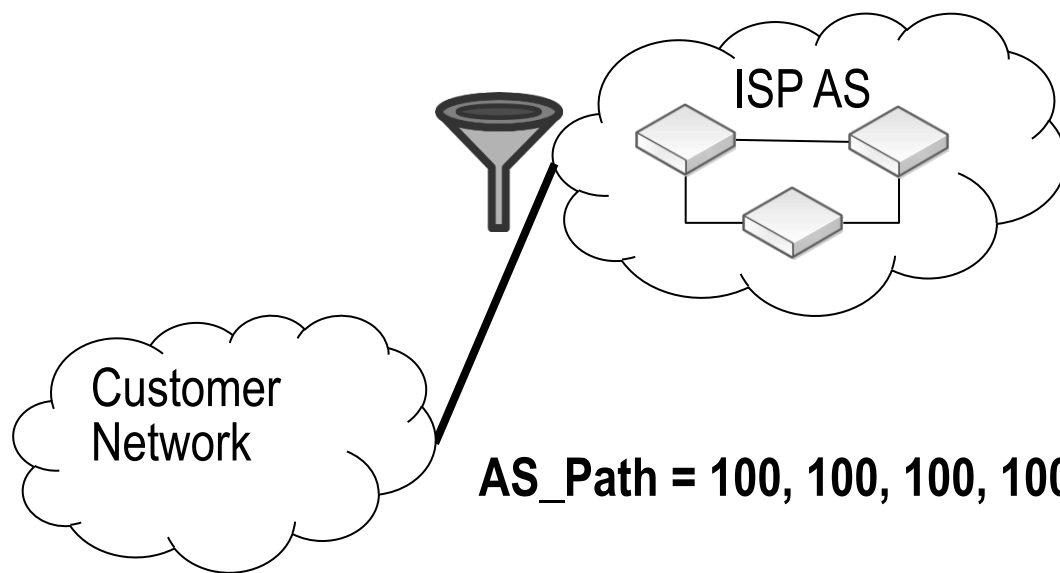
Long AS_Path discard

| # | Chain | Prefix | Prefix Length | Protocol | BGP AS Path | Action |
|----|---------------------|--------|---------------|----------|-------------|---------|
| 29 | long_as_path_dis... | | | | | discard |

Long AS_Path issue

Typically on the net you can reach most networks with only 4 ~ 5 AS's. Much longer AS_Paths should be suspected

Mikrotik x Cisco BUG: In February, 2009 the Internet suffers instability problems due to a misconfiguration on a Mikrotik device, causing a lot of Cisco's to crash (<http://www.renesys.com/blog/2009/02/the-flap-heard-around-the-world.shtml>)



| Matchers | BGP | Actions | BGP Actions |
|----------|-----|---------|--|
| | | | Set BGP Weight: <input type="text"/> |
| | | | Set BGP Local Pref.: <input type="text"/> |
| | | | Set BGP Prepend: <input type="text" value="16"/> |
| | | | Set BGP Prepend Path: <input type="text" value="200"/> |
| | | | <input type="text" value="300"/> |
| | | | <input type="text" value="400"/> |

Special Use IP Addresses (RFC 5735)

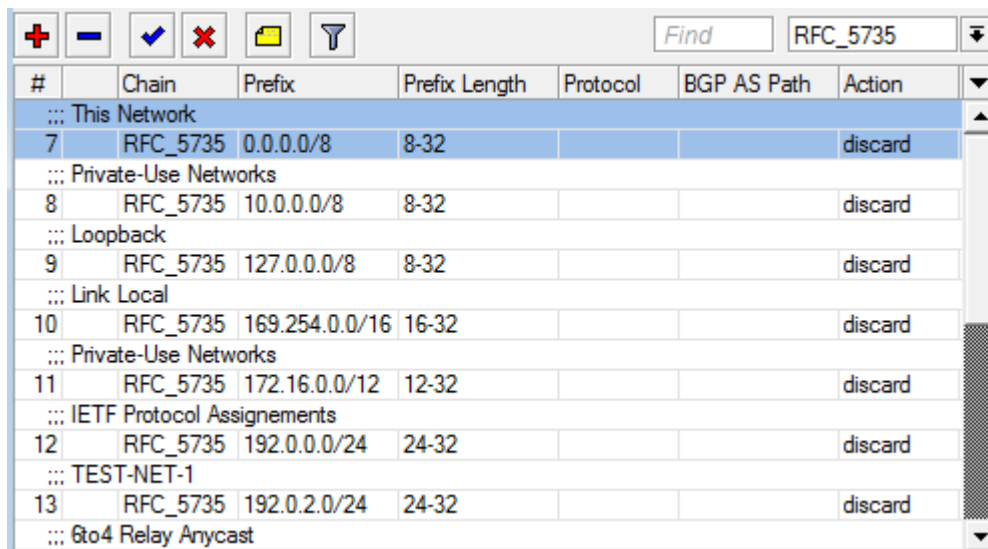
4. Summary Table

| Address Block | Present Use | Reference |
|--------------------|---------------------------|--|
| 0.0.0.0/8 | "This" Network | RFC 1122, Section 3.2.1.3 |
| 10.0.0.0/8 | Private-Use Networks | RFC 1918 |
| 127.0.0.0/8 | Loopback | RFC 1122, Section 3.2.1.3 |
| 169.254.0.0/16 | Link Local | RFC 3927 |
| 172.16.0.0/12 | Private-Use Networks | RFC 1918 |
| 192.0.0.0/24 | IETF Protocol Assignments | RFC 5736 |
| 192.0.2.0/24 | TEST-NET-1 | RFC 5737 |
| 192.88.99.0/24 | 6to4 Relay Anycast | RFC 3068 |
| 192.168.0.0/16 | Private-Use Networks | RFC 1918 |
| 198.18.0.0/15 | Network Interconnect | |
| | Device Benchmark Testing | RFC 2544 |
| 198.51.100.0/24 | TEST-NET-2 | RFC 5737 |
| 203.0.113.0/24 | TEST-NET-3 | RFC 5737 |
| 224.0.0.0/4 | Multicast | RFC 3171 |
| 240.0.0.0/4 | Reserved for Future Use | RFC 1112, Section 4 |
| 255.255.255.255/32 | Limited Broadcast | RFC 919, Section 7 RFC 922, Section 7 |

Best Common Practices

Filtering examples

Discarding RFC 5735 IP's



The screenshot shows the Mikrotik WinBox interface for configuring an IP address list. At the top, there are icons for adding, removing, enabling, disabling, saving, and applying, along with a search bar containing 'RFC_5735'. The main table lists various IP ranges with the action 'discard'.

| # | Chain | Prefix | Prefix Length | Protocol | BGP AS Path | Action |
|-------------------------------|----------|----------------|---------------|----------|-------------|---------|
| ::: This Network | | | | | | |
| 7 | RFC_5735 | 0.0.0.0/8 | 8-32 | | | discard |
| ::: Private-Use Networks | | | | | | |
| 8 | RFC_5735 | 10.0.0.0/8 | 8-32 | | | discard |
| ::: Loopback | | | | | | |
| 9 | RFC_5735 | 127.0.0.0/8 | 8-32 | | | discard |
| ::: Link Local | | | | | | |
| 10 | RFC_5735 | 169.254.0.0/16 | 16-32 | | | discard |
| ::: Private-Use Networks | | | | | | |
| 11 | RFC_5735 | 172.16.0.0/12 | 12-32 | | | discard |
| ::: IETF Protocol Assignments | | | | | | |
| 12 | RFC_5735 | 192.0.0.0/24 | 24-32 | | | discard |
| ::: TEST-NET-1 | | | | | | |
| 13 | RFC_5735 | 192.0.2.0/24 | 24-32 | | | discard |
| ::: 6to4 Relay Anycast | | | | | | |

Best Common Practices

Filtering examples

Discarding Bogon's

- You can manually set filtering to specific bogon's lists
- You can do it automatically

Automatic BOGON filter



**TEAM CYMRU
COMMUNITY
SERVICES**

HOW DO I OBTAIN A PEERING SESSION?

To peer with the bogon route servers, contact bogonrs@cymru.com. When requesting a peering session, please include the following information in your e-mail:

1. Which bogon types you wish to receive (traditional IPv4 bogons, IPv4 fullbogons, and/or IPv6 fullbogons)
2. Your AS number
3. The IP address(es) you want us to peer with
4. Does your equipment support MD5 passwords for BGP sessions?
5. Optional: your GPG/PGP public key

We will typically provide multiple peering sessions (at least 2) per remote peer for redundancy. If you would like more or less than 2 sessions please note that in your request. We try to respond to new peering requests within one to two business days, but, again, can provide no guarantees for this **free** service.

Remember that you must be able to accomodate up to **100 prefixes** for *traditional bogons*, and up to **50,000 prefixes** for *fullbogons*, and be capable of multihop peering with a private ASN. If you improperly configure your peering and route all packets destined for bogon addresses to the bogon route-servers, your peering session will be dropped.

Automatic BOGON's filter

Marking incoming routes from Cymru as blackhole

Route Filter <>

Matchers BGP Actions BGP Actions

Chain: cymru-in

Route Filter <>

Matchers BGP Actions BGP Actions

BGP AS Path:

BGP AS Path Length:

BGP Weight:

BGP Local Pref.:

BGP MED:

BGP Atomic Aggregate:

BGP Origin:

Locally Originated BGP:

▲ BGP Communities

BGP Communities: 65332:888

Route Filter <>

Matchers BGP Actions BGP Actions

Action: accept

Jump Target:

Set Distance:

Set Scope:

Set Target Scope:

Set Pref. Source:

Set In Nexthop:

Set In Nexthop Direct:

Set Out Nexthop:

Set Routing Mark:

Set Route Comment:

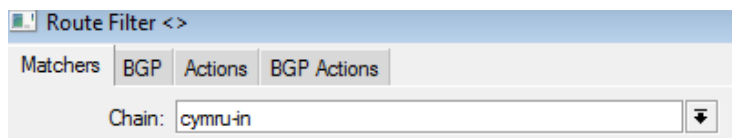
Set Check Gateway:

Set Disabled:

Set Type: blackhole

Automatic BOGON's filter

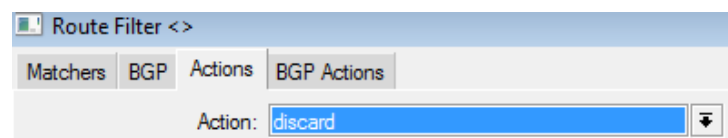
Discarding other prefixes



Route Filter <>

Matchers BGP Actions BGP Actions

Chain: cymru-in

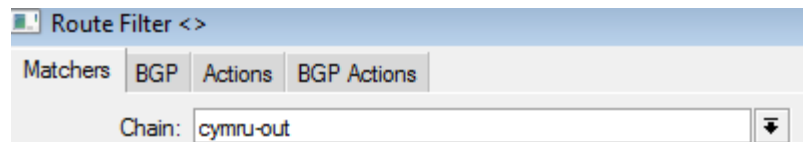


Route Filter <>

Matchers BGP Actions BGP Actions

Action: discard

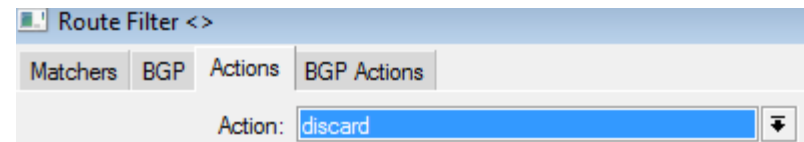
To prevent sending prefixes to Cymru



Route Filter <>

Matchers BGP Actions BGP Actions

Chain: cymru-out



Route Filter <>

Matchers BGP Actions BGP Actions

Action: discard

Best Common Practices

Filtering examples

Putting all together

Matchers BGP Actions BGP Actions

Chain: ▼

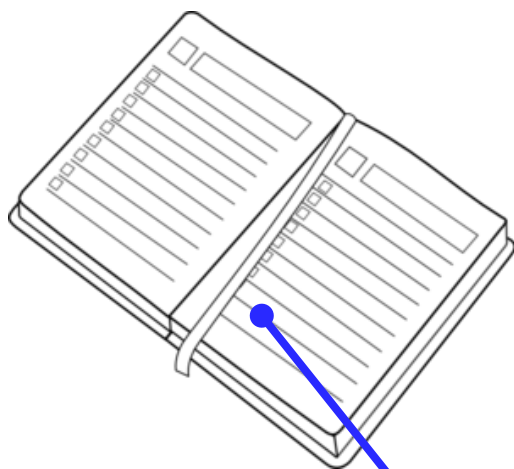
Matchers BGP Actions BGP Actions

Action: ▼

Jump Target: ▼

| <div> + - ✓ ✗ 📁 🔍 </div> | | | | | | | |
|--|-----------------|--------|---------------|----------|-------------|--------|--|
| # | Chain | Prefix | Prefix Length | Protocol | BGP AS Path | Action | |
| 3 | in_filter_ISP_1 | | | | | jump | |
| 30 | in_filter_ISP_1 | | | | | jump | |
| 31 | in_filter_ISP_1 | | | | | jump | |
| 32 | in_filter_ISP_1 | | | | | jump | |
| 33 | in_filter_ISP_1 | | | | | jump | |

Agenda



1) Dynamic routing essentials ✓

2) OSPF

→ OSPF Overview ✓

→ OSPF threats and countermeasures ✓

3) BGP

→ BGP Overview ✓

→ BGP threats and countermeasures ✓

4) Conclusions.

Final considerations and conclusions



Default implementations of Routing systems can be exploited easily if no protective measure is taken.

OSPF can be well protected if some protective measures are used. Special care about topology should be watched.

When it comes to BGP, there is no definitive measure to ensure an absolutely security.

There are some drafts for secure external routing systems, like sBGP, soBGP, RPKI, etc

While such new protocols variants are not available, all we can do is to apply best practices to minimize the risks.

References



A Survey of BGP Security - Kevin Butler, Toni Farley, Patrick McDaniel, Jennifer Rexford

Beware of BGP Attacks (Nordstrom, et. al.)

BGP Security Vulnerabilities Analysis (draft-ietf-idr-bgp-vuln-01.txt, Murphy)

Best Practices for securing Routing Protocols – Cisco

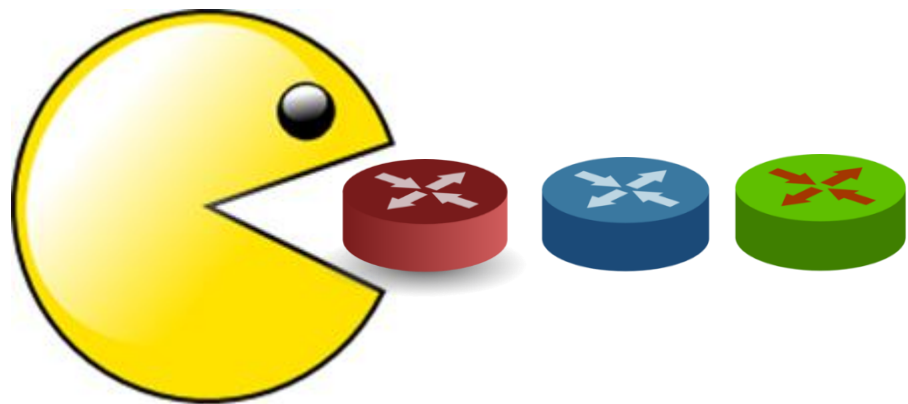
Border Gateway Protocol Security - Recommendations of the National Institute of Standards and Technology NSIT – (Rick Kuhn, Kotikalapudi Sriram, Doug Montgomery)

BGP Techniques for Internet Service Providers – Cisco – (Philip Smith)

Burning Asgard – An Introduction to the Tool *Loki* (Rene Graf, Daniel Mende, Enno Rey)

Mikrotik Wiki

Extra Slides



Routing Filters example

/routing filter

```
add action=discard chain=own_prefix_discard comment="All prefixes owned by the  
provider should be listed here" disabled=no invert-match=no prefix=1.1.0.0/20 prefix-  
length=20-32
```

```
add action=discard chain=own_prefix_discard comment="" disabled=no invert-  
match=no prefix=2.2.0.0/20 prefix-length=20-32
```

```
add action=discard chain=own_prefix_discard comment="" disabled=no invert-  
match=no prefix=3.3.0.0/20 prefix-length=20-32
```

```
add action=jump chain=in_filter_ISP_1 comment="" disabled=no invert-match=no  
jump-target=own_prefix_discard set-type=unicast
```

```
add action=jump chain=in_filter_ISP_2 comment="" disabled=no invert-match=no  
jump-target=own_prefix_discard
```

Routing Filters example

```
add action=discard chain=RFC_5735 comment="This Network" disabled=no invert-match=no prefix=0.0.0.0/8 prefix-length=8-32
```

```
add action=discard chain=RFC_5735 comment="Private-Use Networks" disabled=no invert-match=no prefix=10.0.0.0/8 prefix-length=8-32
```

```
add action=discard chain=RFC_5735 comment=Loopback disabled=no invert-match= no prefix=127.0.0.0/8 prefix-length=8-32
```

```
add action=discard chain=RFC_5735 comment="Link Local" disabled=no invert-match=no prefix=169.254.0.0/16 prefix-length=16-32
```

```
add action=discard chain=RFC_5735 comment="Private-Use Networks" disabled=no invert-match=no prefix=172.16.0.0/12 prefix-length=12-32
```

```
add action=discard chain=RFC_5735 comment="IETF Protocol Assignments" disabled=no invert-match=no prefix=192.0.0.0/24 prefix-length=24-32
```

Routing Filters example

```
add action=discard chain=RFC_5735 comment=TEST-NET-1 disabled=no invert-match=no prefix=192.0.2.0/24 prefix-length=24-32
```

```
add action=discard chain=RFC_5735 comment="6to4 Relay Anycast" disabled=no invert-match=no prefix=192.88.99.0/24 prefix-length=24-32
```

```
add action=discard chain=RFC_5735 comment="Private-Use Networks" disabled=no invert-match=no prefix=192.168.0.0/16 prefix-length=16-32
```

```
add action=discard chain=RFC_5735 comment="Network Interconnect Device Benchmark test" disabled=no invert-match=no prefix=192.18.0.0/15 prefix-length=15-32
```

```
add action=discard chain=RFC_5735 comment=TEST-NET-2 disabled=no invert-match=no prefix=198.51.100.0/24 prefix-length=24-32
```

Routing Filters example

```
add action=discard chain=RFC_5735 comment=TEST-NET-3 disabled=no invert-  
match=no prefix=203.0.113.0/24 prefix-length=24-32
```

```
add action=discard chain=RFC_5735 comment=Multicast disabled=no invert-  
match=\
```

```
no prefix=224.0.0.0/4 prefix-length=4-32
```

```
add action=discard chain=RFC_5735 comment="Reserved for future use"  
disabled=\
```

```
no invert-match=no prefix=240.0.0.0/4 prefix-length=4-32
```

```
add action=discard chain=RFC_5735 comment="Limited Broadcast" disabled=no \
```

Routing Filters example

```
add action=discard chain=default_route_discard comment=Reject_Default_Route  
disabled=no invert-match=no prefix=0.0.0.0/0
```

```
add action=discard chain=Longer_Bitmask_discard comment="" disabled=no invert-  
match=no prefix-length=25-32
```

```
add action=passthrough bgp-as-path-length=22 chain="" comment="" disabled=no  
invert-match=no
```

```
add action=accept chain=announcing_only_owned_prefixes comment=""  
disabled=no invert-match=no prefix=1.1.0.0/20
```

```
add action=accept chain=announcing_only_owned_prefixes comment=""  
disabled=no invert-match=no prefix=2.2.0.0/20
```

```
add action=accept chain=announcing_only_owned_prefixes comment=""  
disabled=no invert-match=no prefix=3.3.0.0/20
```

```
add action=discard chain=announcing_only_owned_prefixes comment="" disabled=  
no invert-match=no
```


OSPF built in security features

OSPF “Fight back” feature

“Every LSA that is circulating in the OSPF network with wrong information will be corrected by its owner.”

Common perception could suggest that:

- Fight Back corrects the damage of most attacks
- Many theoretical attacks are not worth the effort just to cause a brief topology change

Is such perception absolutely true ?

OSPF attacks

Forcing topology changes 2/2

Even, having the authentication key in hands, won't be the attack frustrated by Fight Back feature ?

- When a legitimate owner receives a malicious copy of its own LSAs:
 - Since the malicious LSA has higher sequence number, and a copy of the LSA is already present in the LSDB and this copy was not received by flooding but installed by the router itself,
 - Then Flood the malicious LSA and **AFTER** check ownership.
 - After checking, router will try to update the malicious LSA
 - RFC 2328 specifies a MinLSInterval of 5 seconds which routers cannot inject two same LSA's, but will flood immediately any LSA received.

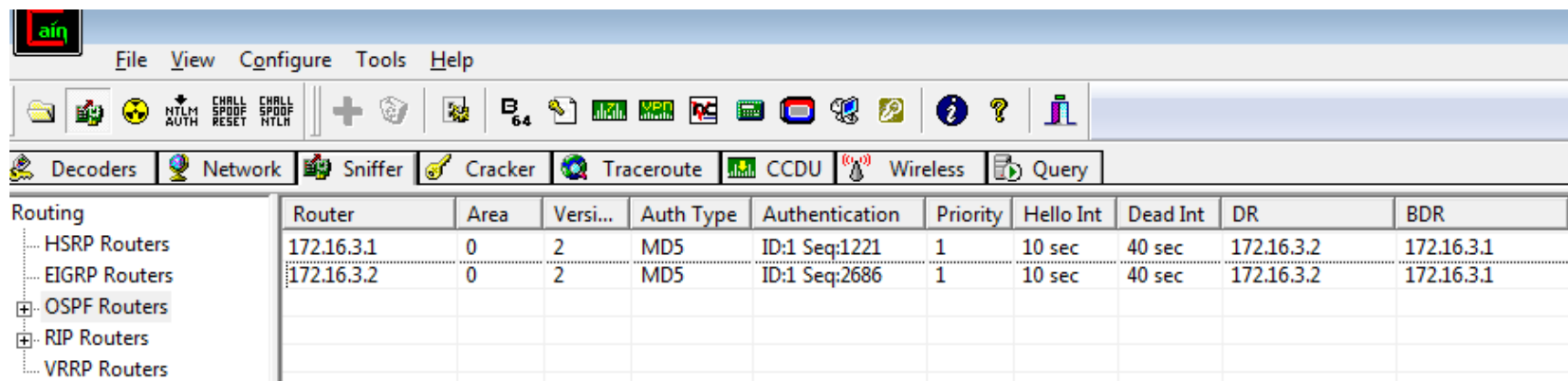
So, If the malicious LSAs are injected with a rate higher than MinLSInterval, fight back won't work !

From RFC 3682 (suggests a TTL “hack” of 255, instead of 1)

5.1. TTL (Hop Limit) Spoofing

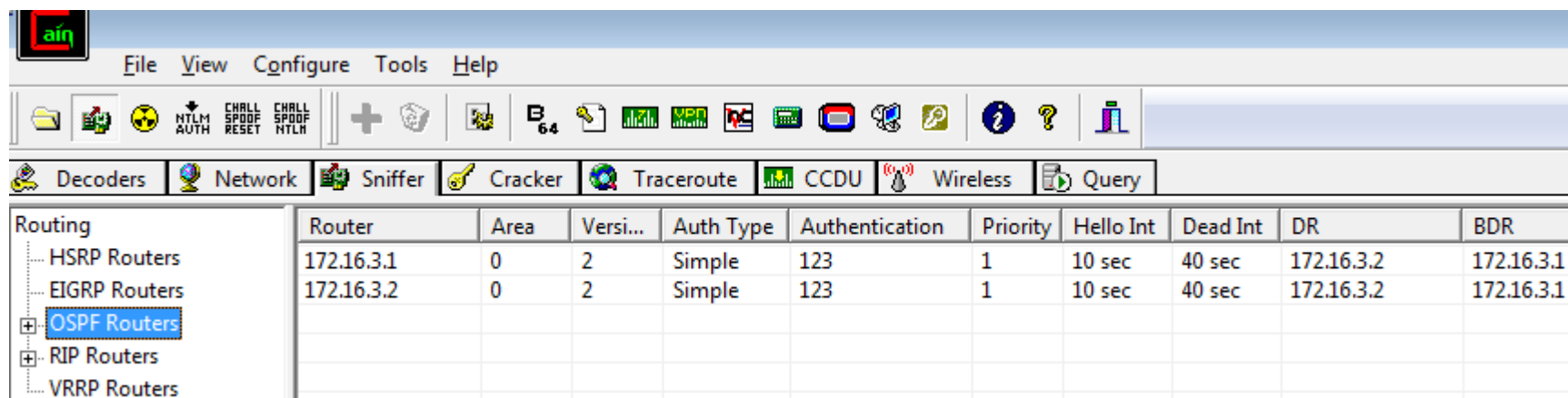
“The approach described here is based on the observation that a TTL (or Hop Limit) value of 255 is non-trivial to spoof, since as the packet passes through routers Towards the destination, the TTL is decremented by one. As a result, when a router receives a packet, it may not be able to determine if the packet's IP address is valid, but it can determine how many router hops away it is (again, assuming none of the routers in the path are compromised in such a way that they would reset the packet's TTL). Note, however, that while engineering a packet's TTL such that it has a particular value when sourced from an arbitrary location is difficult (but not impossible), engineering a TTL value of 255 from non-directly connected locations is not possible (again, assuming none of the directly connected neighbors are compromised, the packet hasn't been tunneled to the decapsulator, and the intervening routers are operating in accordance with RFC 791 [RFC791]).

Windows tool for hacking routing systems



The screenshot shows the Lain application window with the 'Network' tab selected. The interface includes a menu bar (File, View, Configure, Tools, Help), a toolbar with various icons, and a sidebar on the left with a tree view under 'Routing' containing HSRP, EIGRP, OSPF, RIP, and VRRP routers. The main area displays a table of network data.

| Router | Area | Versi... | Auth Type | Authentication | Priority | Hello Int | Dead Int | DR | BDR |
|------------|------|----------|-----------|----------------|----------|-----------|----------|------------|------------|
| 172.16.3.1 | 0 | 2 | MD5 | ID:1 Seq:1221 | 1 | 10 sec | 40 sec | 172.16.3.2 | 172.16.3.1 |
| 172.16.3.2 | 0 | 2 | MD5 | ID:1 Seq:2686 | 1 | 10 sec | 40 sec | 172.16.3.2 | 172.16.3.1 |



This screenshot is similar to the one above but highlights the 'OSPF Routers' entry in the left sidebar. The table in the main area shows the same network data, but with 'Simple' authentication types and '123' as the authentication values.

| Router | Area | Versi... | Auth Type | Authentication | Priority | Hello Int | Dead Int | DR | BDR |
|------------|------|----------|-----------|----------------|----------|-----------|----------|------------|------------|
| 172.16.3.1 | 0 | 2 | Simple | 123 | 1 | 10 sec | 40 sec | 172.16.3.2 | 172.16.3.1 |
| 172.16.3.2 | 0 | 2 | Simple | 123 | 1 | 10 sec | 40 sec | 172.16.3.2 | 172.16.3.1 |

Windows tool for hacking routing systems

The screenshot shows the 'Sniffer' tab in the 'Network' section. The left sidebar lists 'Routing' protocols, with 'HSRP Routers' expanded. The main table displays the following data:

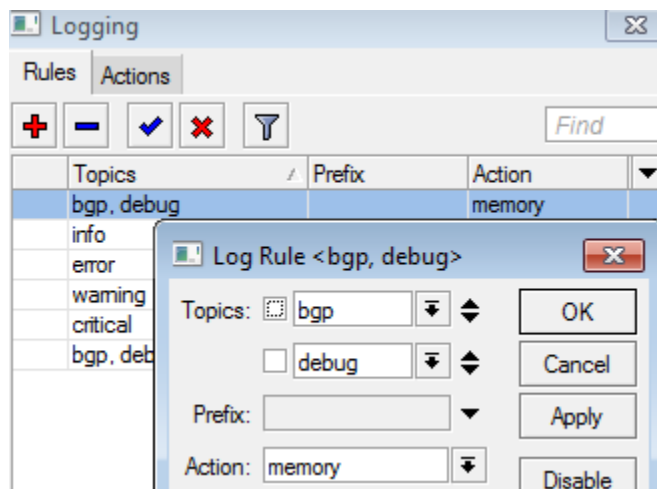
| Router | Area | Versi... | Auth Type | Authentication | Priority | Hello Int | Dead Int | DR | BDR |
|------------|------|----------|-----------|----------------|----------|-----------|----------|------------|------------|
| 172.16.3.1 | 0 | 2 | MD5 | ID:1 Seq:1221 | 1 | 10 sec | 40 sec | 172.16.3.2 | 172.16.3.1 |
| 172.16.3.2 | 0 | 2 | MD5 | ID:1 Seq:2686 | 1 | 10 sec | 40 sec | 172.16.3.2 | 172.16.3.1 |

The screenshot shows the 'Sniffer' tab in the 'Network' section. The left sidebar lists 'Routing' protocols, with 'OSPF Routers' expanded. The main table displays the following data:

| Router | Area | Versi... | Auth Type | Authentication | Priority | Hello Int | Dead Int | DR | BDR |
|------------|------|----------|-----------|----------------|----------|-----------|----------|------------|------------|
| 172.16.3.1 | 0 | 2 | Simple | 123 | 1 | 10 sec | 40 sec | 172.16.3.2 | 172.16.3.1 |
| 172.16.3.2 | 0 | 2 | Simple | 123 | 1 | 10 sec | 40 sec | 172.16.3.2 | 172.16.3.1 |

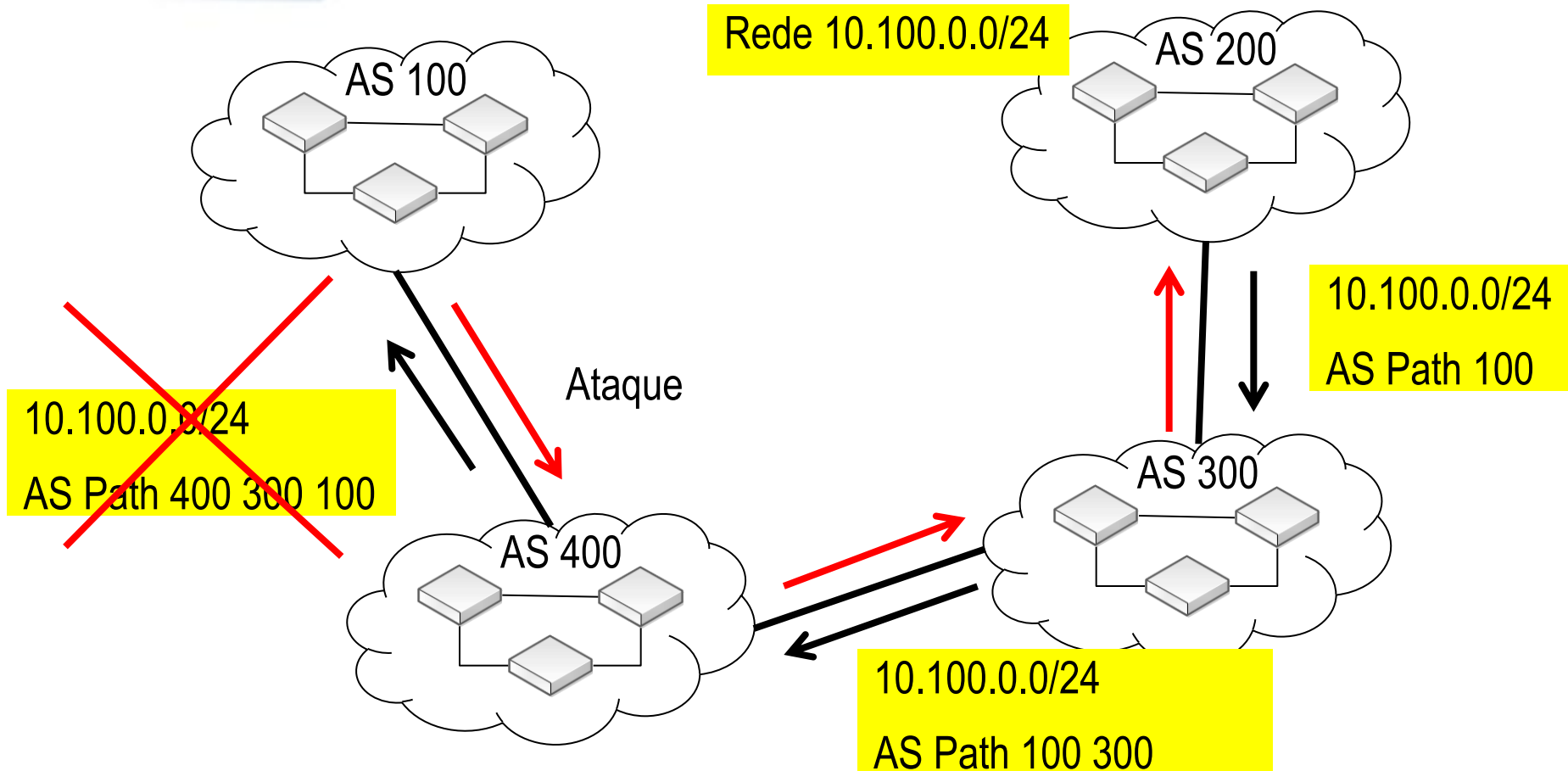
Debugging BGP

Ativate BGP log + debug in /system logging



| Log | | |
|----------------------|-----------------------|--|
| Jan/10/1970 04:53:14 | route bgp debug p... | RemoteAddress=172.16.0.11 |
| Jan/10/1970 04:53:14 | route bgp debug p... | MessageLength=19 |
| Jan/10/1970 04:53:14 | route bgp debug p... | Received KEEPALIVE packet |
| Jan/10/1970 04:53:14 | route bgp debug p... | RemoteAddress=172.16.0.11 |
| Jan/10/1970 04:53:14 | route bgp debug p... | Length=19 |
| Jan/10/1970 04:53:14 | route bgp debug p... | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF |
| Jan/10/1970 04:53:14 | route bgp debug p... | 00 13 04 |
| Jan/10/1970 04:53:14 | route bgp debug p... | KEEPALIVE Message |
| Jan/10/1970 04:53:14 | route bgp debug p... | RemoteAddress=172.16.0.21 |
| Jan/10/1970 04:53:14 | route bgp debug p... | MessageLength=19 |
| Jan/10/1970 04:53:14 | route bgp debug p... | Received KEEPALIVE packet |
| Jan/10/1970 04:53:14 | route bgp debug p... | RemoteAddress=172.16.0.21 |
| Jan/10/1970 04:53:14 | route bgp debug p... | Length=19 |
| Jan/10/1970 04:53:14 | route bgp debug p... | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF |
| Jan/10/1970 04:53:14 | route bgp debug p... | 00 13 04 |
| Jan/10/1970 04:53:19 | route bgp debug ti... | ConnectRetryTimer expired |
| Jan/10/1970 04:53:19 | route bgp debug ti... | RemoteAddress=2001:470:13:9e::1 |
| Jan/10/1970 04:53:19 | route bgp debug | Update source |

Avoiding DoS attacks by generating same AS from the attacker



The looping avoidance feature of BGP could be used to block attacks from an arbitrary AS. Just advertise the attacked prefix appending the AS attacker. (Filters on upstream providers could frustrate such technique)

Real Case Scenario - Americana Digital

| # | Chain | Prefix | Prefix Length | Protocol | BGP AS Path | Action |
|----|-----------------|-----------------|---------------|----------|-------------|-------------|
| 0 | rfc1918 | 0.0.0.0/8 | 0-32 | | | discard |
| 1 | rfc1918 | 10.0.0.0/8 | 0-32 | | | discard |
| 2 | rfc1918 | 127.0.0.0/8 | 0-32 | | | discard |
| 3 | rfc1918 | 169.254.0.0/16 | 0-32 | | | discard |
| 4 | rfc1918 | 172.16.0.0/12 | 0-32 | | | discard |
| 5 | rfc1918 | 192.168.0.0/16 | 0-32 | | | discard |
| 6 | rfc1918 | 224.0.0.0/3 | 0-32 | | | discard |
| 7 | rfc1918 | | | | | return |
| 8 | cymru-in | | | | | discard |
| 9 | cymru-in | | | | | accept |
| 10 | cymru-in | | | | | discard |
| 11 | cymru-out | | | | | discard |
| 12 | network-in | | | | | jump |
| 13 | network-in | 189.36.224.0... | 20-32 | | | discard |
| 14 | network-out | 189.36.224.0... | 20-21 | | | discard |
| 15 | network-out | 189.36.224.0... | 21 | | | passthrough |
| 16 | ptt-cas-rs1-in | | | | | jump |
| 17 | ptt-cas-rs1-in | 189.36.224.0... | 20-32 | | | discard |
| 18 | ptt-cas-rs1-in | | | | | passthrough |
| 19 | ptt-cas-rs1-out | | | | | discard |
| 20 | ptt-cas-rs1-out | | | | | accept |
| 21 | ptt-cas-rs1-out | 189.36.224.0... | 20-21 | | | discard |
| 22 | ptt-cas-rs1-out | 189.36.224.0... | 20 | | | passthrough |
| 23 | ptt-cas-rs2-in | | | | | jump |
| 24 | ptt-cas-rs2-in | 189.36.224.0... | 20-32 | | | discard |
| 25 | ptt-cas-rs2-in | | | | | passthrough |
| 26 | ptt-cas-rs2-out | | | | | discard |

141 items

Real Case Scenario - Americana Digital

The screenshot displays two Mikrotik WinBox windows. The 'IPv6 Route List' window on the left shows a list of IPv6 routes with columns for Dst. Address, Gateway, and Distance. The 'Route List' window on the right shows a list of routes with columns for Dst. Address, Gateway, Distance, Routing Mark, and Pref. Source. Both windows have a search bar and a filter button. The 'Route List' window also has tabs for Routes, Nexthops, Rules, and VRF.

IPv6 Route List

| | Dst. Address | Gateway | Distance |
|-----|--------------------|---|----------|
| XS | 2000::/3 | 2001:470:1f0d:be::1 | 1 |
| DAb | 2001::/32 | fe80::290:6902:cc65:4c03%ptt-sp-nicbr:716 reachable | 20 |
| DAb | 2001:200::/32 | fe80::290:6902:cc65:4c03%ptt-sp-nicbr:716 reachable | 20 |
| DAb | 2001:200:900::/40 | | |
| DAb | 2001:200:c000::/35 | | |
| DAb | 2001:200:e000::/35 | | |
| DAb | 2001:208::/32 | | |
| DAb | 2001:218::/32 | | |
| DAb | 2001:218:400::/40 | | |
| DAb | 2001:218:6002::/48 | | |
| DAb | 2001:220::/35 | | |
| DAb | 2001:240::/32 | | |
| DAb | 2001:250::/32 | | |
| DAb | 2001:250:204::/48 | | |
| DAb | 2001:250:e000::/36 | | |
| DAb | 2001:250:f000::/36 | | |

3885 items

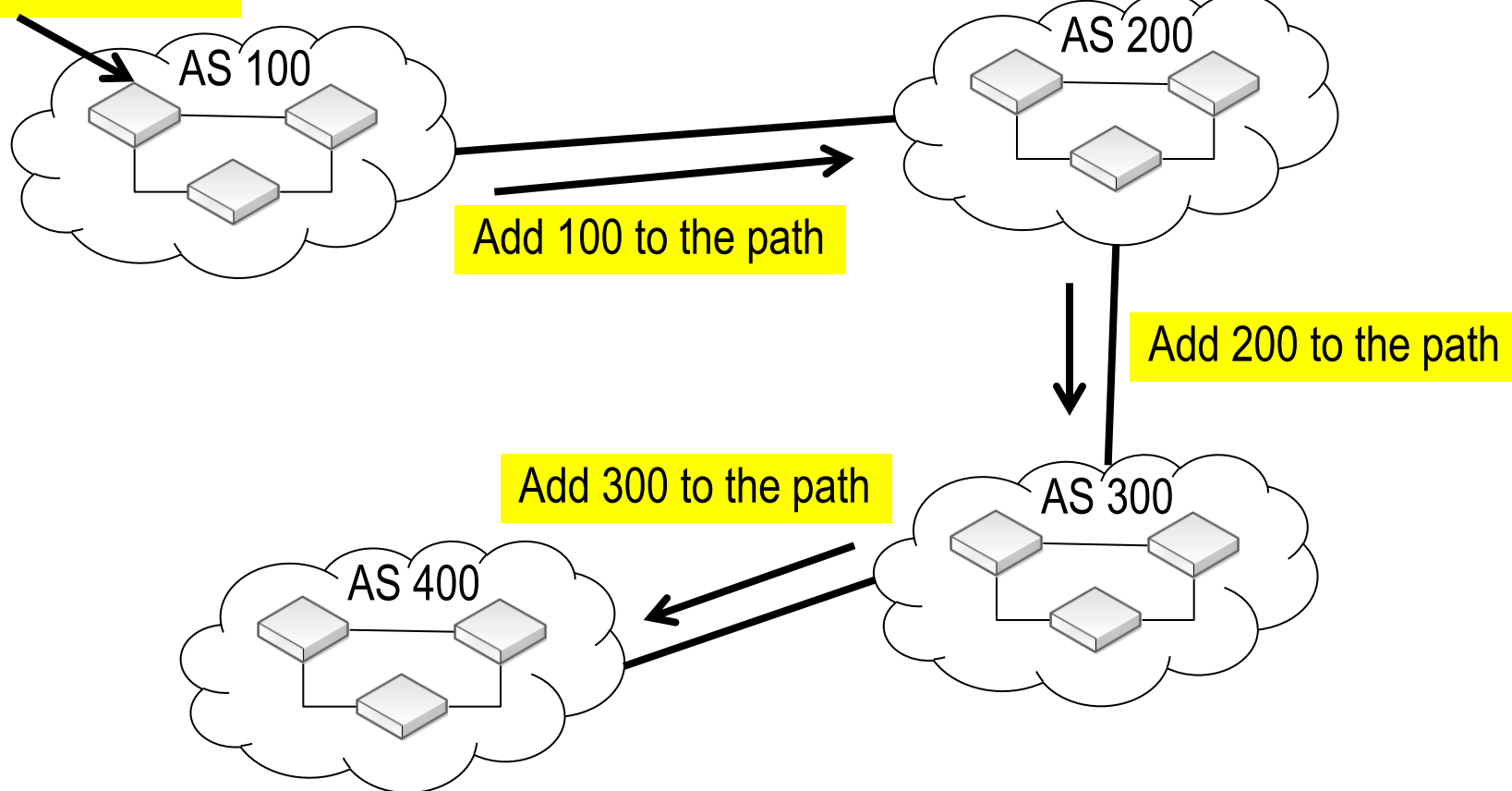
Route List

| | Dst. Address | Gateway | Distance | Routing Mark | Pref. Source |
|-----|--------------|---------|----------|--------------|--------------|
| DAb | 0.0.0.0/8 | | 20 | | |
| DAb | 0.0.0.0/8 | | 20 | | |
| DAb | 1.0.1.0/24 | | 20 | | |
| DAb | 1.0.1.0/24 | | 20 | | |
| DAb | 1.0.2.0/23 | | 20 | | |
| DAb | 1.0.2.0/23 | | 20 | | |
| DAb | 1.0.4.0/22 | | 20 | | |
| DAb | 1.0.4.0/22 | | 20 | | |
| DAb | 1.0.8.0/21 | | 20 | | |
| DAb | 1.0.8.0/21 | | 20 | | |
| DAb | 1.0.16.0/20 | | 20 | | |
| DAb | 1.0.16.0/20 | | 20 | | |
| DAb | 1.0.32.0/19 | | 20 | | |
| DAb | 1.0.32.0/19 | | 20 | | |
| DAb | 1.0.64.0/18 | | 20 | | |

5881 items out of 355720

Path Vector implementation

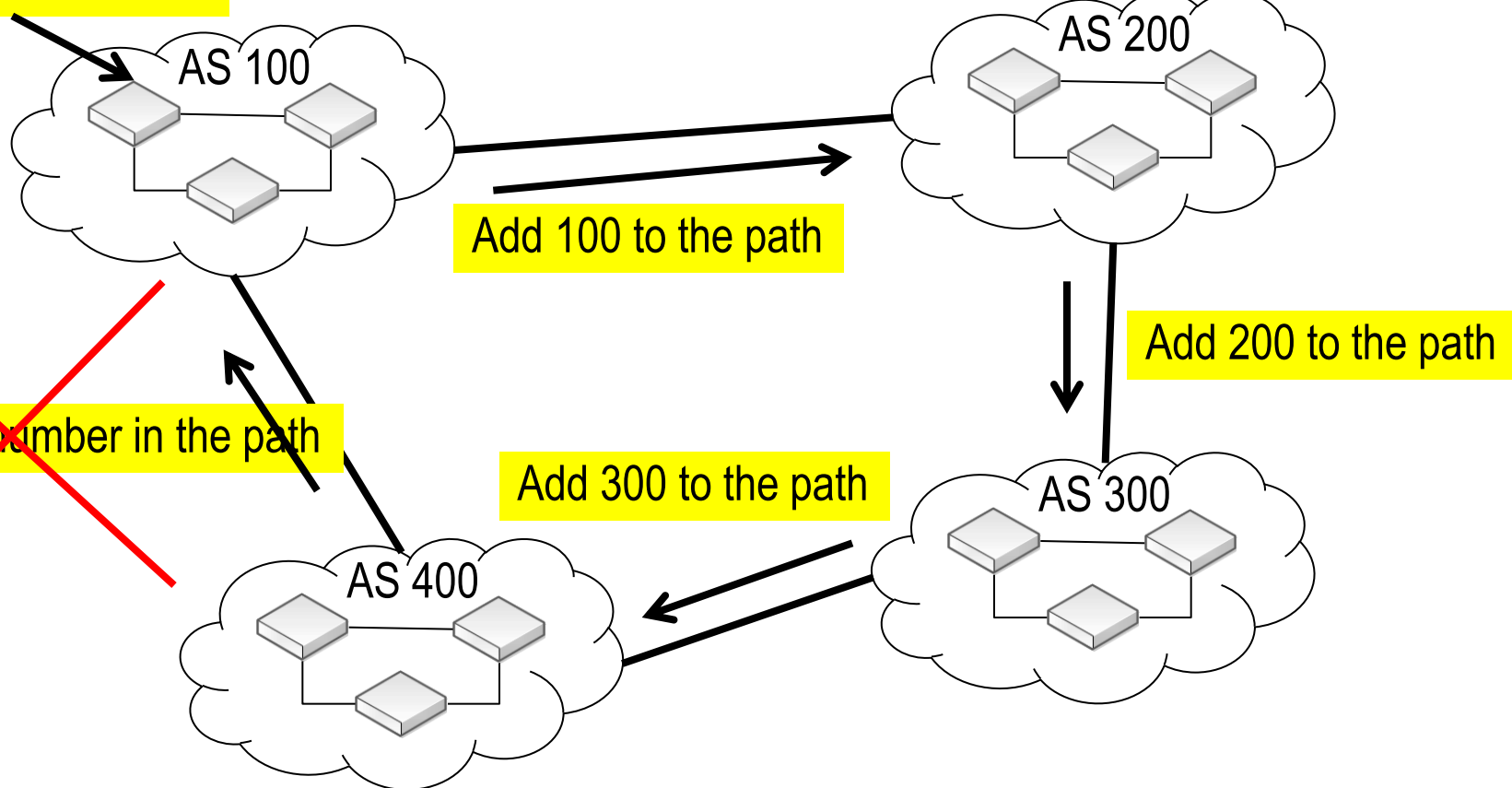
Network 10.100.0.0/24



AS 400 knows that, to reach network 10.100.0.0/24, the path is through 300 e 200

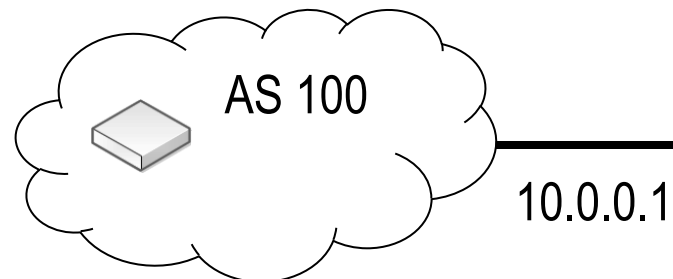
Path Vector implementation looping avoidance

Network 10.100.0.0/24



Next hop on shared media (e.g. a IXP)

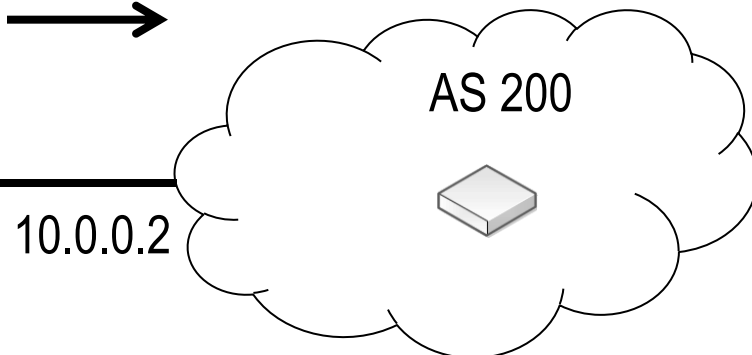
Network 10.100.0.0/16



Network 10.100.0.0/16

AS_Path 100

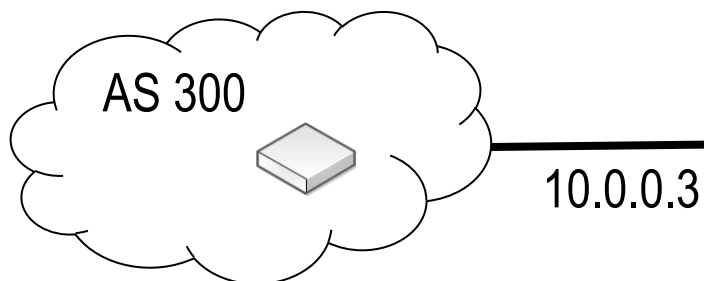
Next_Hop 10.0.0.1



Network 10.100.0.0/16

AS_Path 200 100

Next_Hop 10.0.0.1



If the receiving router is in the same subnet of the prior Next_Hop router, this remains intact to optimize packet forwarding.

Questions ?

Wardner Maia – maia@mikrotikbrasil.com.br

Köszönöm Egészségére !



Wardner Maia – maia@mikrotikbrasil.com.br

