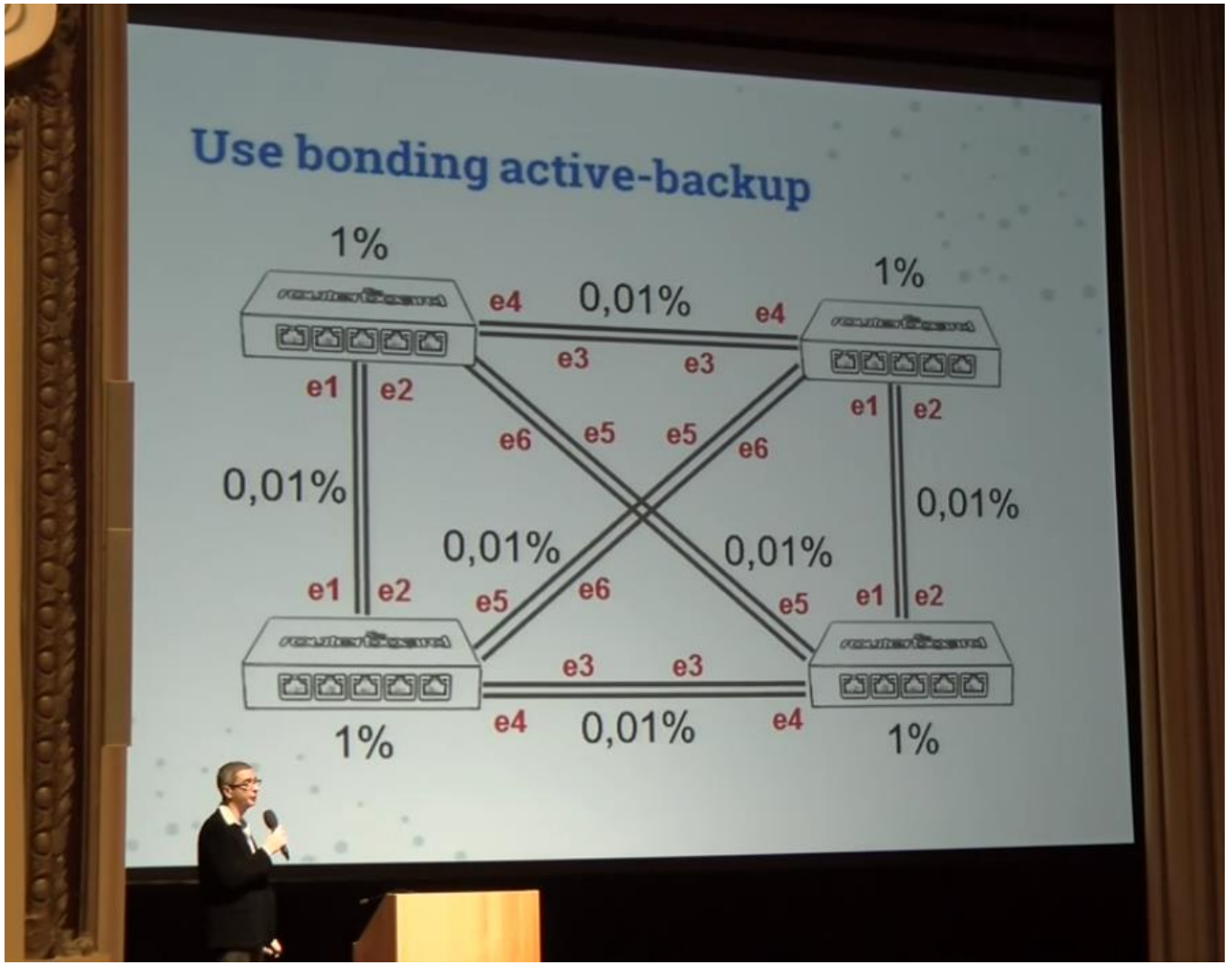
Security by harnessing the power of RouterOS

Mihai Săftoiu - MUM România

29 October 2018

Mihai Săftoiu

- MikroTik Certified Trainer
- MikroTik Consultant
- TIER Data Center CEO
- STARNET NOC Manager
- Italian ISP CTO
- Security auditor for military and energy contractors



Why this presentation?

What this presentation is not.

What this presentation is.

Who is in the audience ?

What is security?

Security is the consequence of the following situation:

When an unauthorized person:

- does not have the key
- cannot find/copy the key
- if the key gets found, it cannot be used

Conclusion:

Security is the applied logic (algorithm, way of doing etc.) which leads to that consequence.

Defining the goal:

To have a system which even if compromised (**revealed password**) will remain secure from a functional authentication perspective.

Is that even possible?

Discussion topics:

General communication priciples

RouterOS security mechanisms

Applying authentication logic on different layers

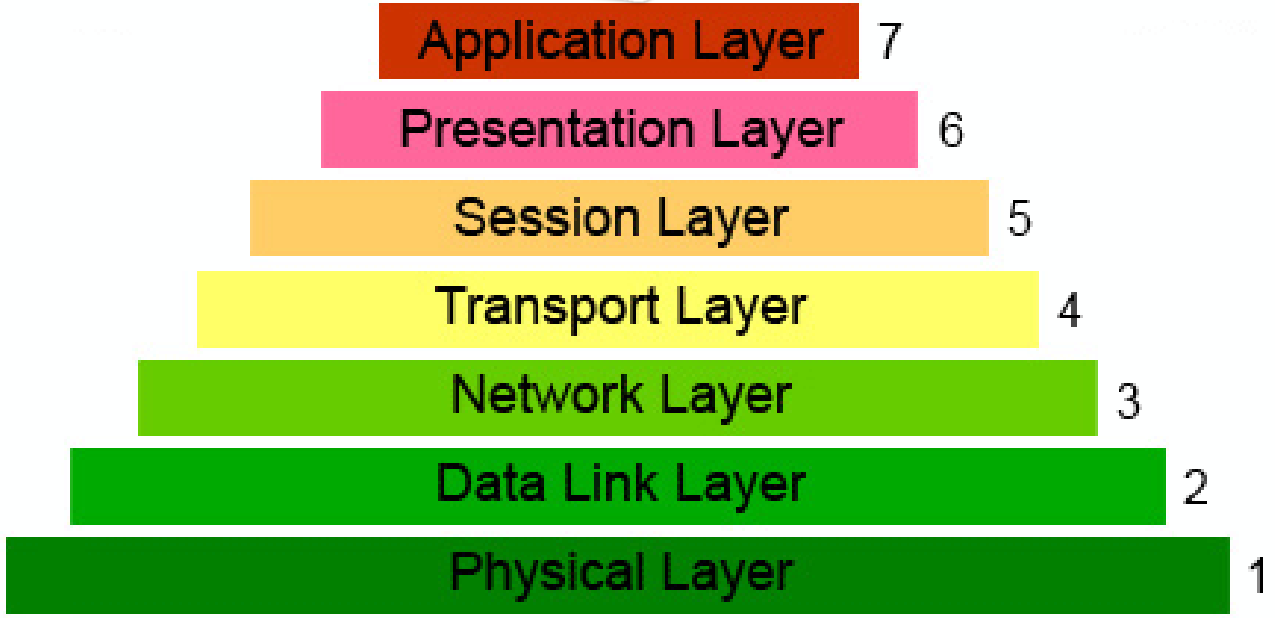
Putting it all together

General principles

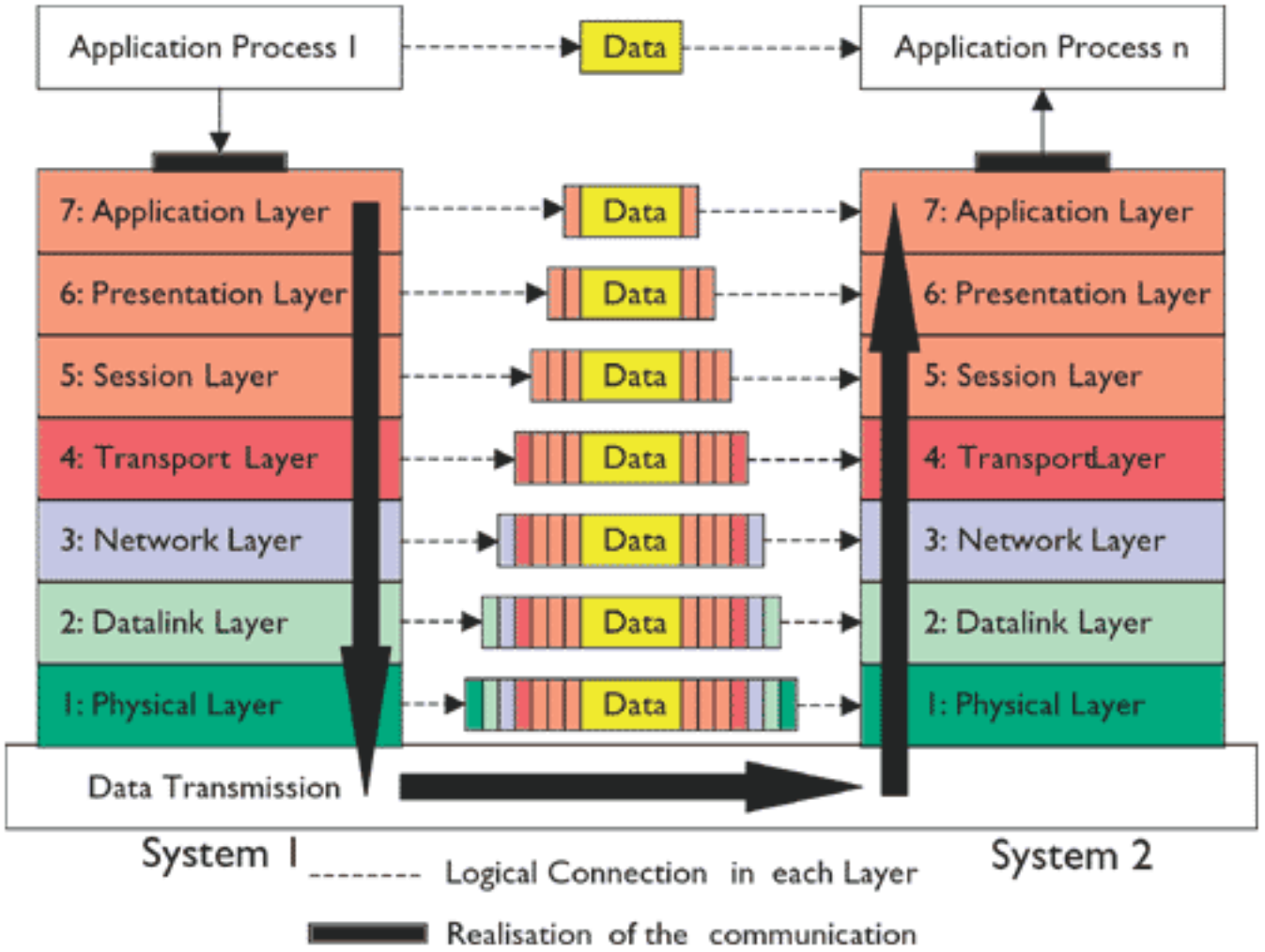
The OSI model is the basis of all inter-system communication.

Understanding the OSI model is the first step in understanding where to apply security concepts.

General principles - OSI



General principles - OSI



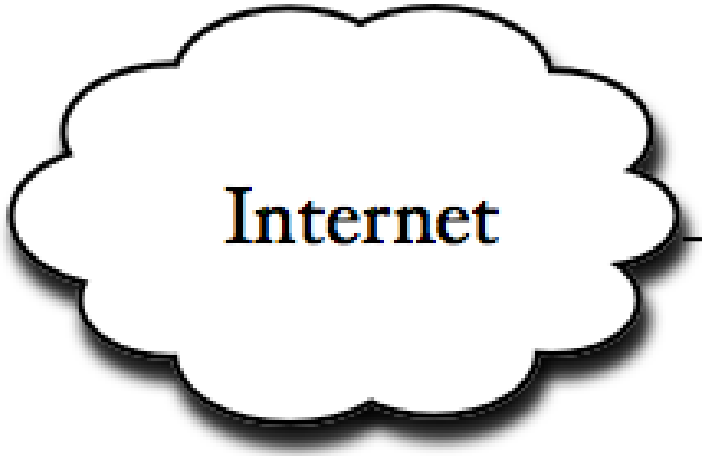
General principles - OSI

Application level authentication security	Application Layer	7
These layers are geared towards security, we can actually make use of them	Presentation Layer	6
	Session Layer	5
TCP/IP security	Transport Layer	4
	Network Layer	3
MAC authentication security	Data Link Layer	2
Physical security	Physical Layer	1

The setup

192.168.88.250/24

192.168.88.1/24
ether2



ether1

RouterOS security mechanisms

Physical security

MAC authentication security

TCP/IP security

Application level security

Physical security

1. Using secure enclosures/racks
2. Using centralized logging
3. Disabling unused interfaces
4. Protecting from factory reset

Physical security

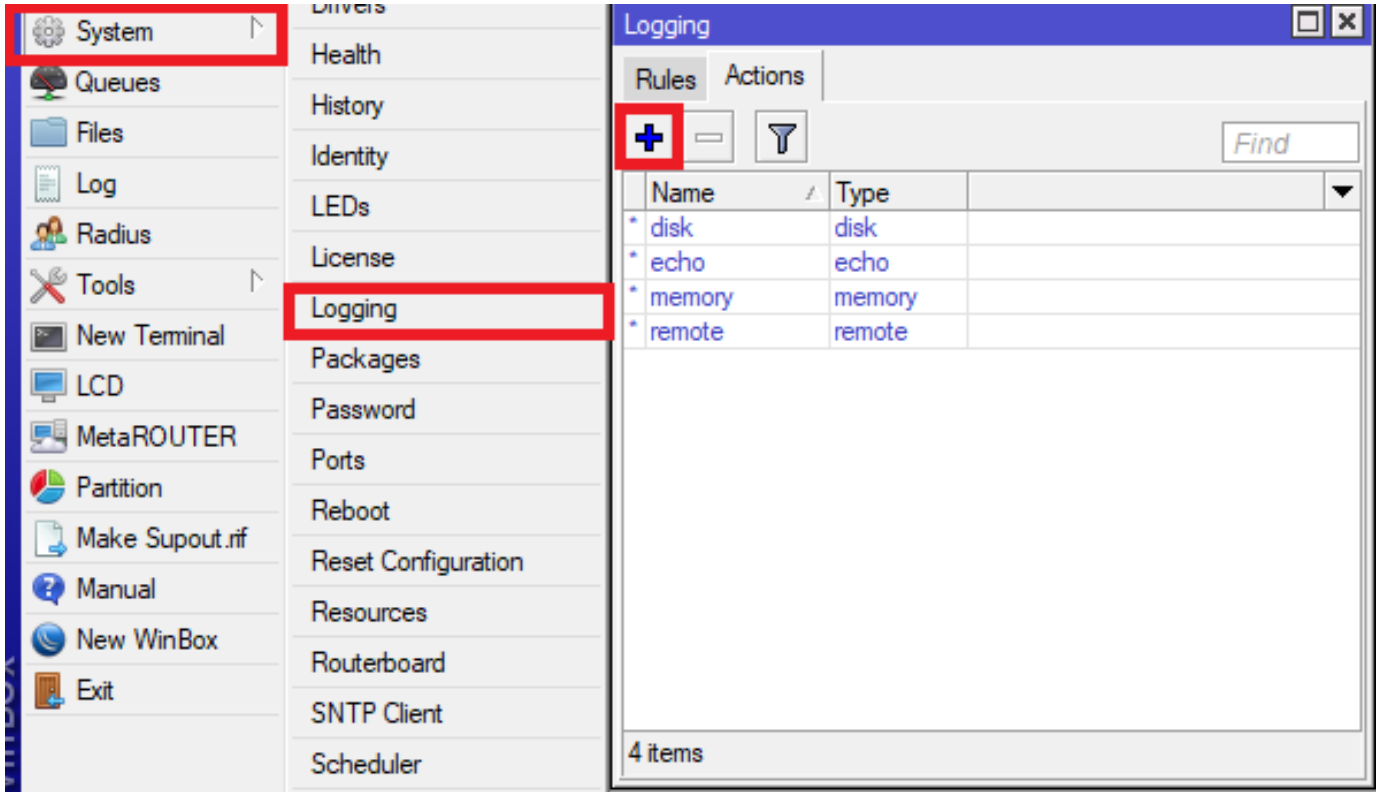
2. Using centralized logging

Centralized logging is possible for free using The Dude

https://wiki.mikrotik.com/wiki/Manual:The_Dude_v6/Syslog

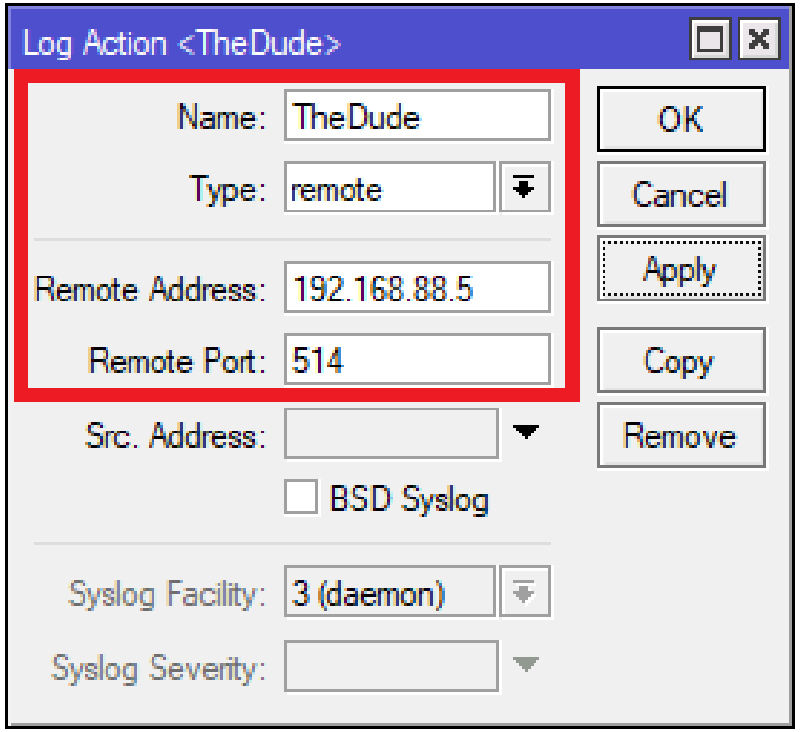
Physical security

2. Using centralized logging



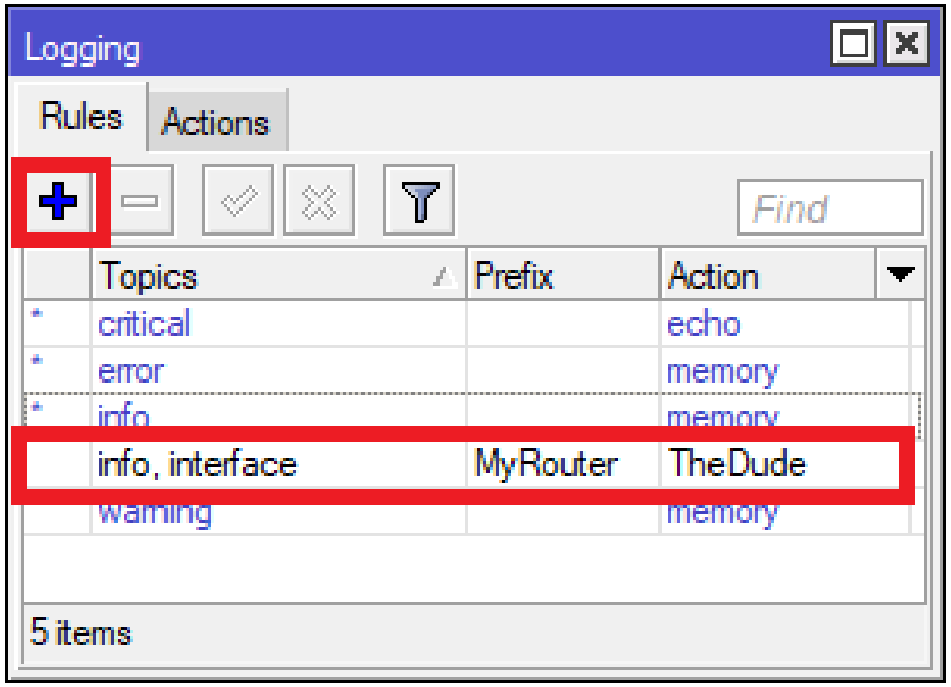
Physical security

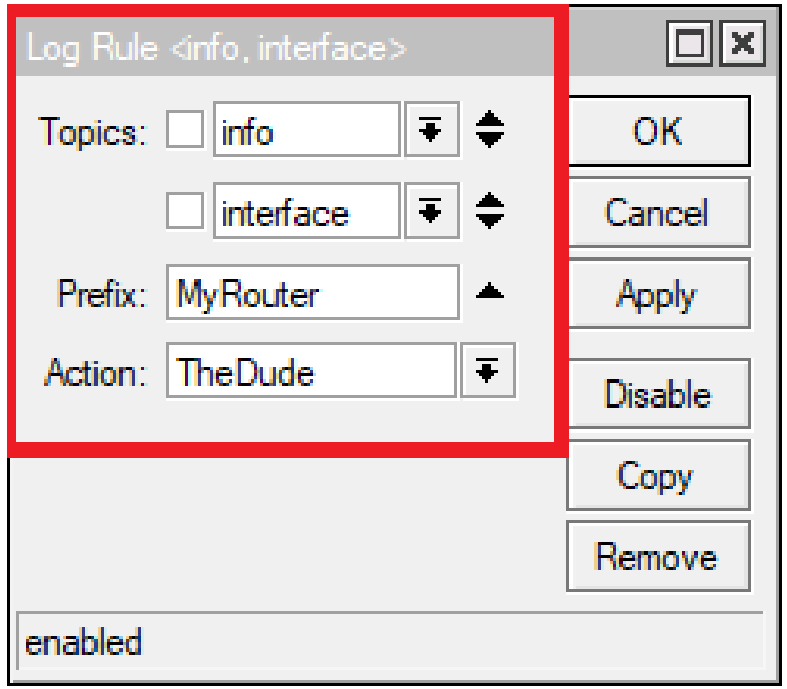
2. Using centralized logging



Physical security

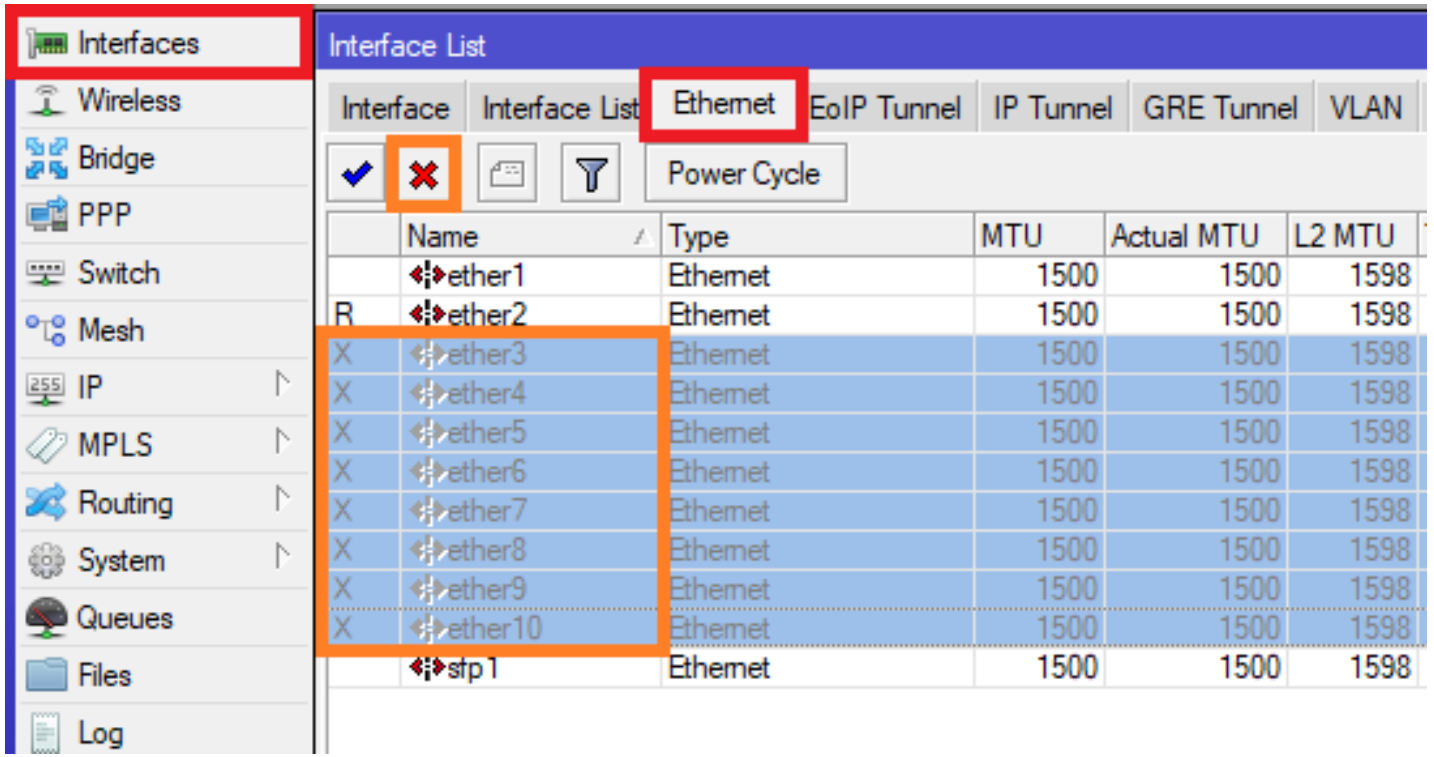
2. Using centralized logging





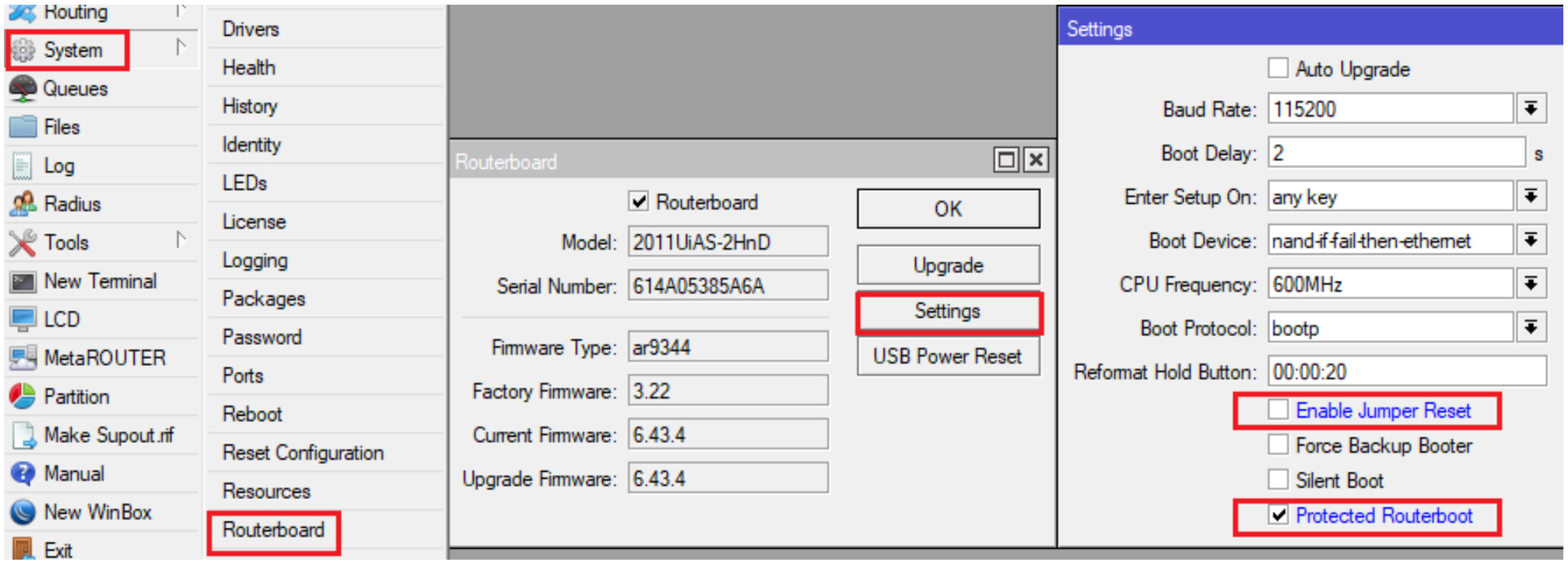
Physical security

3. Disabling unused interfaces



Physical security

4. Protection from hardware reset



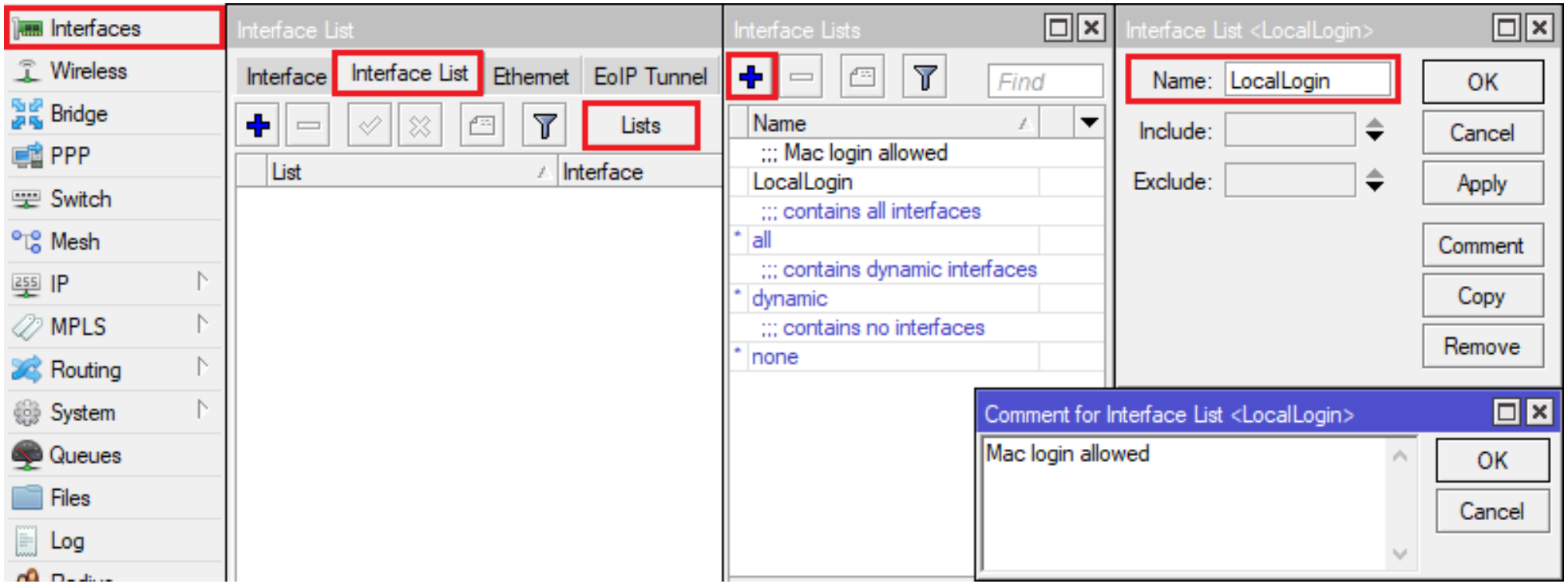
MAC authentication security

1. Disabling unwanted local login

2. Allowing only specific devices to access the physical ports

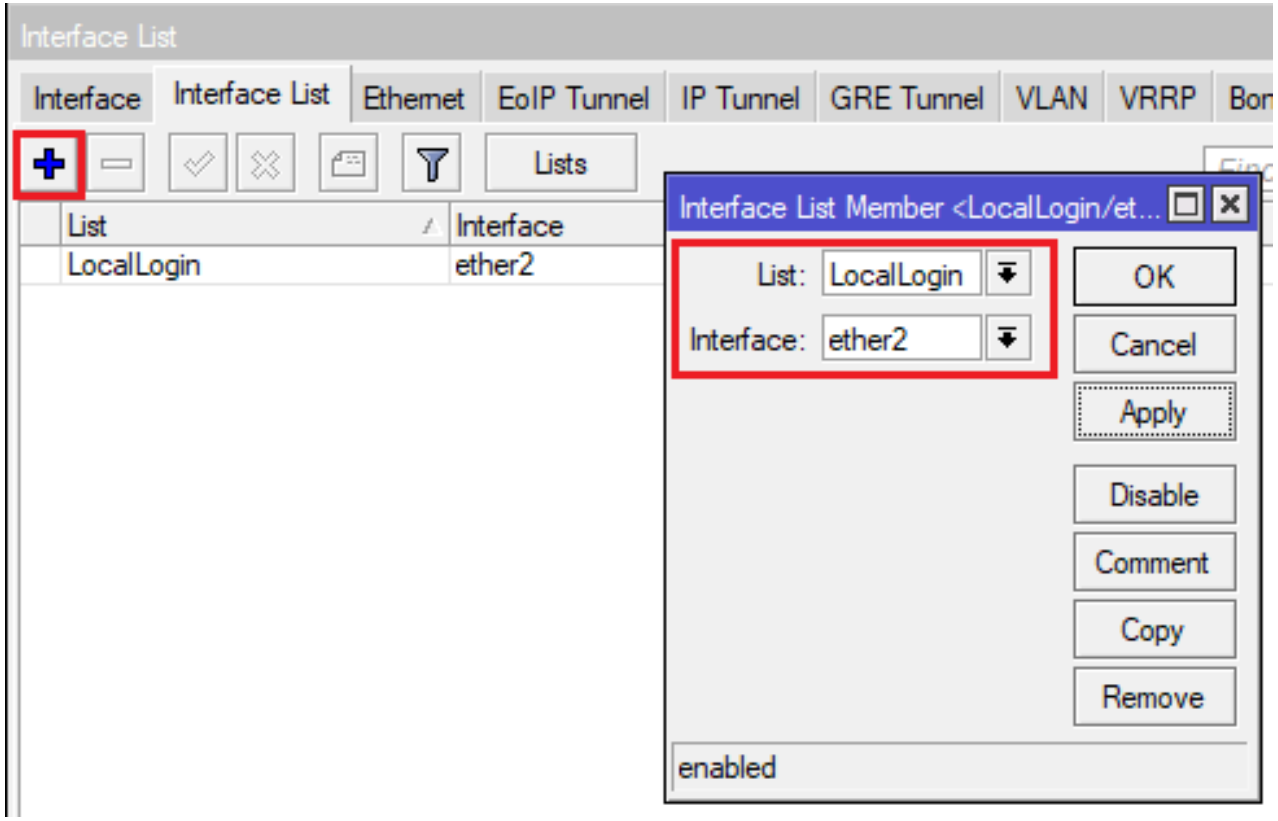
MAC authentication security

1. Disabling unwanted local login



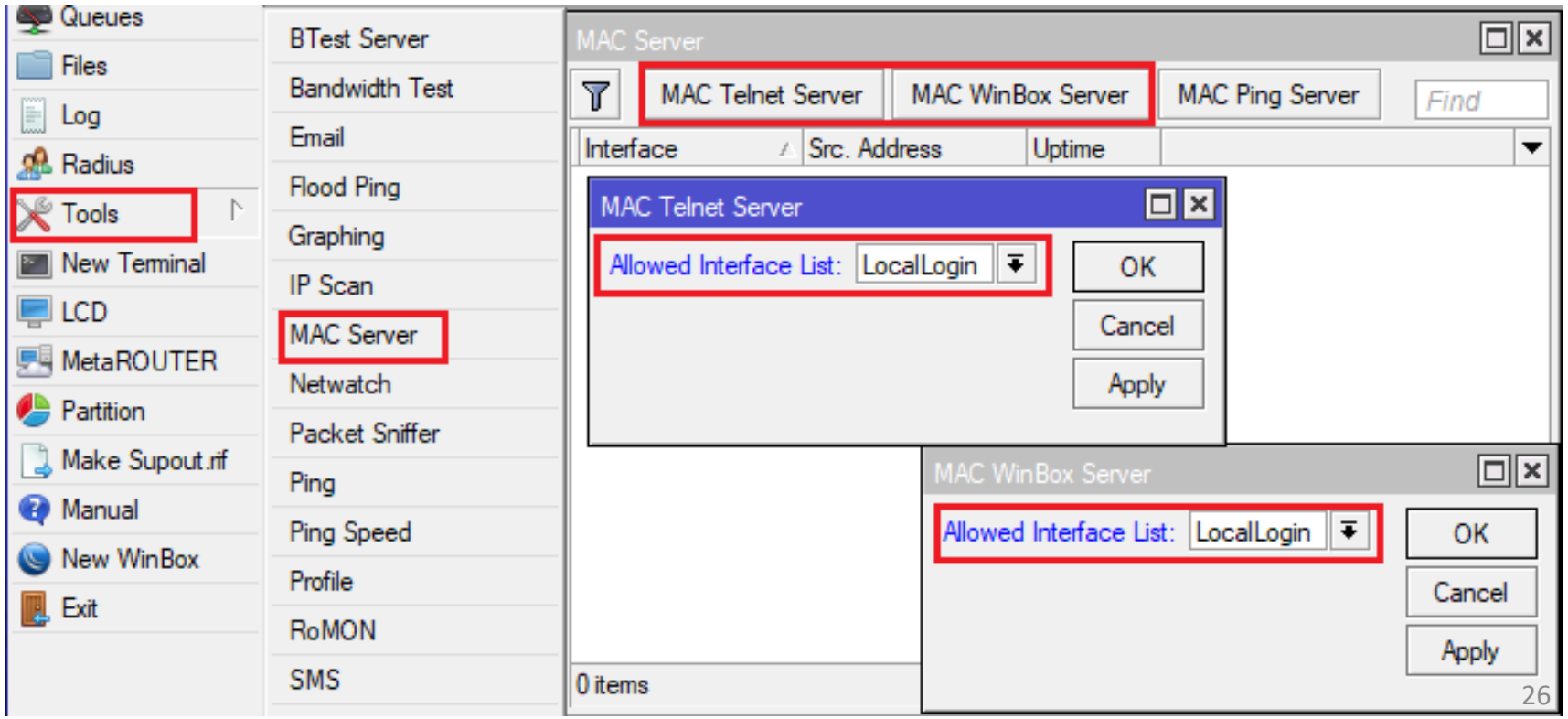
MAC authentication security

1. Disabling unwanted local login



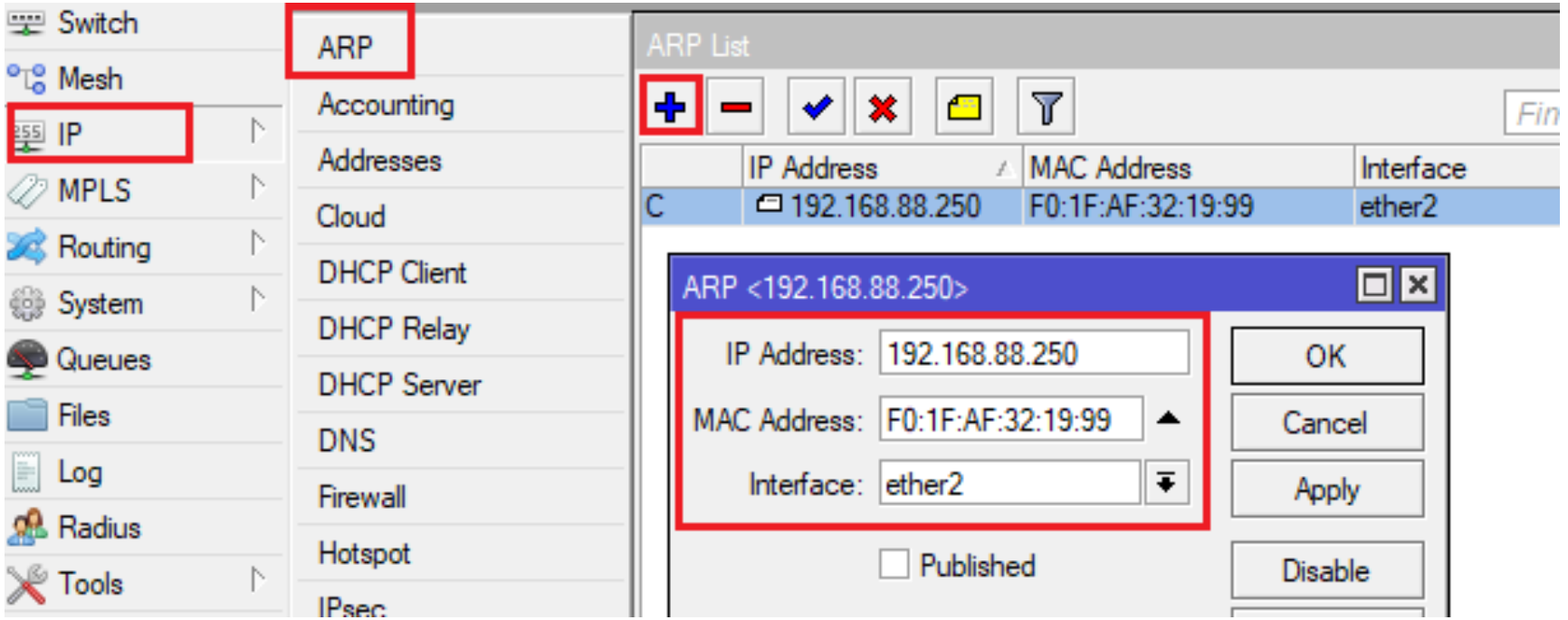
MAC authentication security

1. Disabling unwanted local login



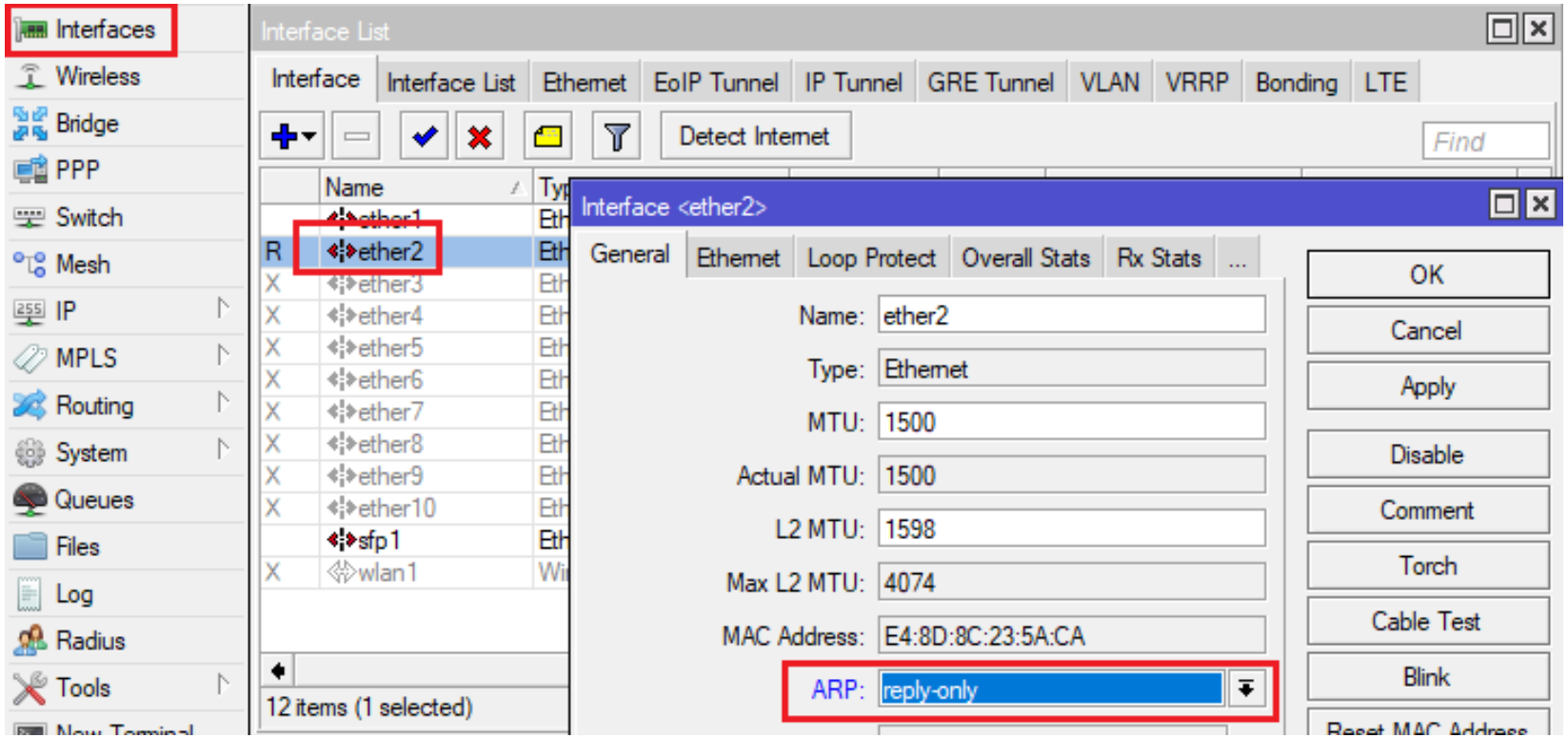
MAC authentication security

2. Allowing only trusted devices



MAC authentication security

2. Allowing only trusted devices



TCP/IP Security

Practices found on the Internet usually employ techniques such as:

- IP ACL trust relationship (firewall)
- filtering invalid packet sources
- some form of port knocking technique

TCP/IP Security

IP ACL trust relationship (firewall)

- the most common layer 3 method
- easy to configure
- useful when used with static IPs
- easy to bypass using spoofing
- what to do when in another city?

TCP/IP Security

Filtering invalid packet sources

- also common in modern firewalls
- a skilled attacker will not send invalid packets and will not get blacklisted

TCP/IP Security

Some form of port knocking

- employs a set of ports which are “knocked” which then enables the login to the device
- the ports used can also be sniffed or discovered by specialized tools

Application layer security

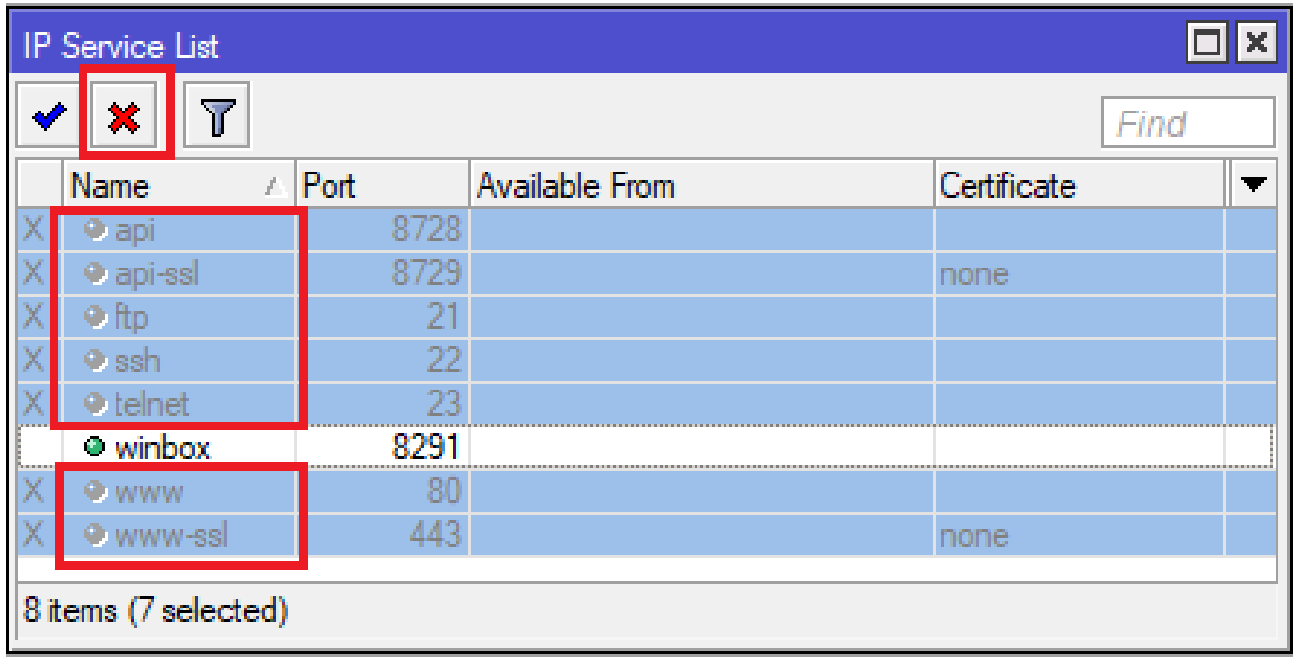
1. Disabling unwanted services

2. Using built-in ACL mechanism and changing default ports

P.S. We'll get back to TCP/IP soon

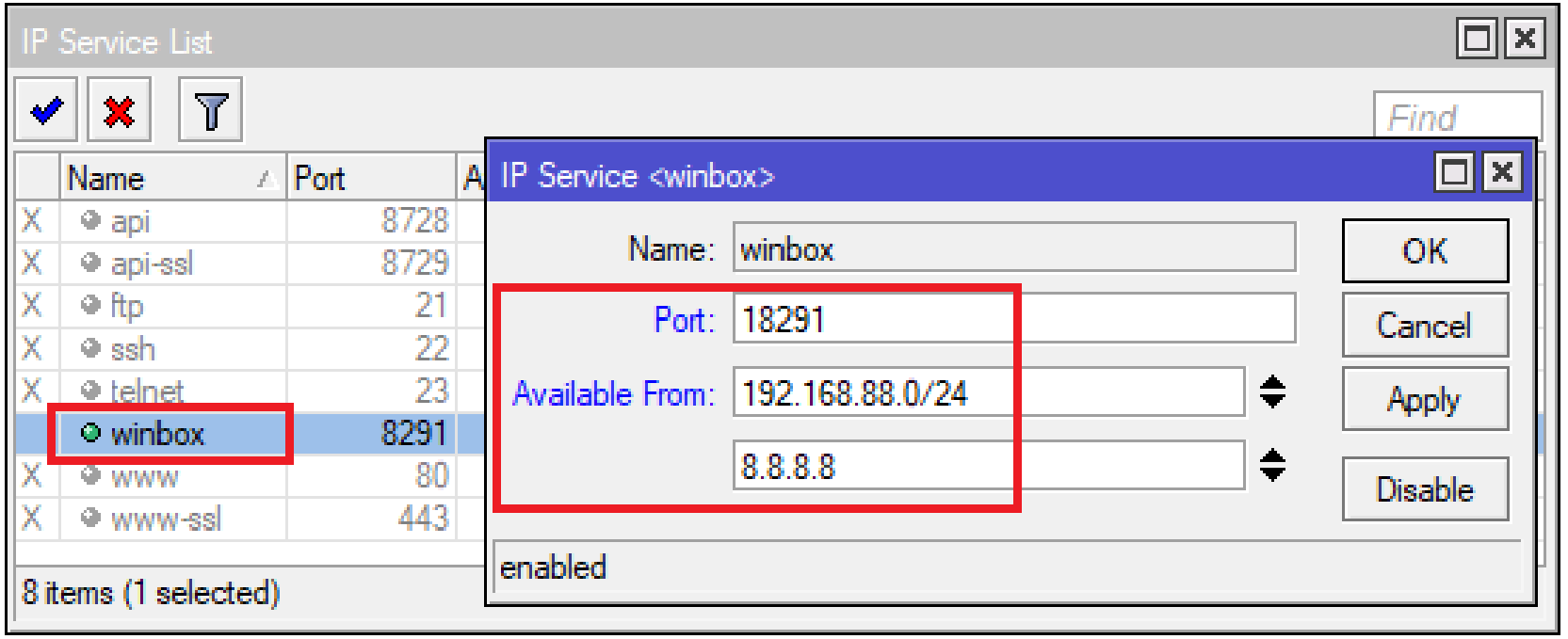
Application layer security

1. Disabling services (IP -> Services)



Application layer security

2. ACL and default port change



Reiterating the goal:

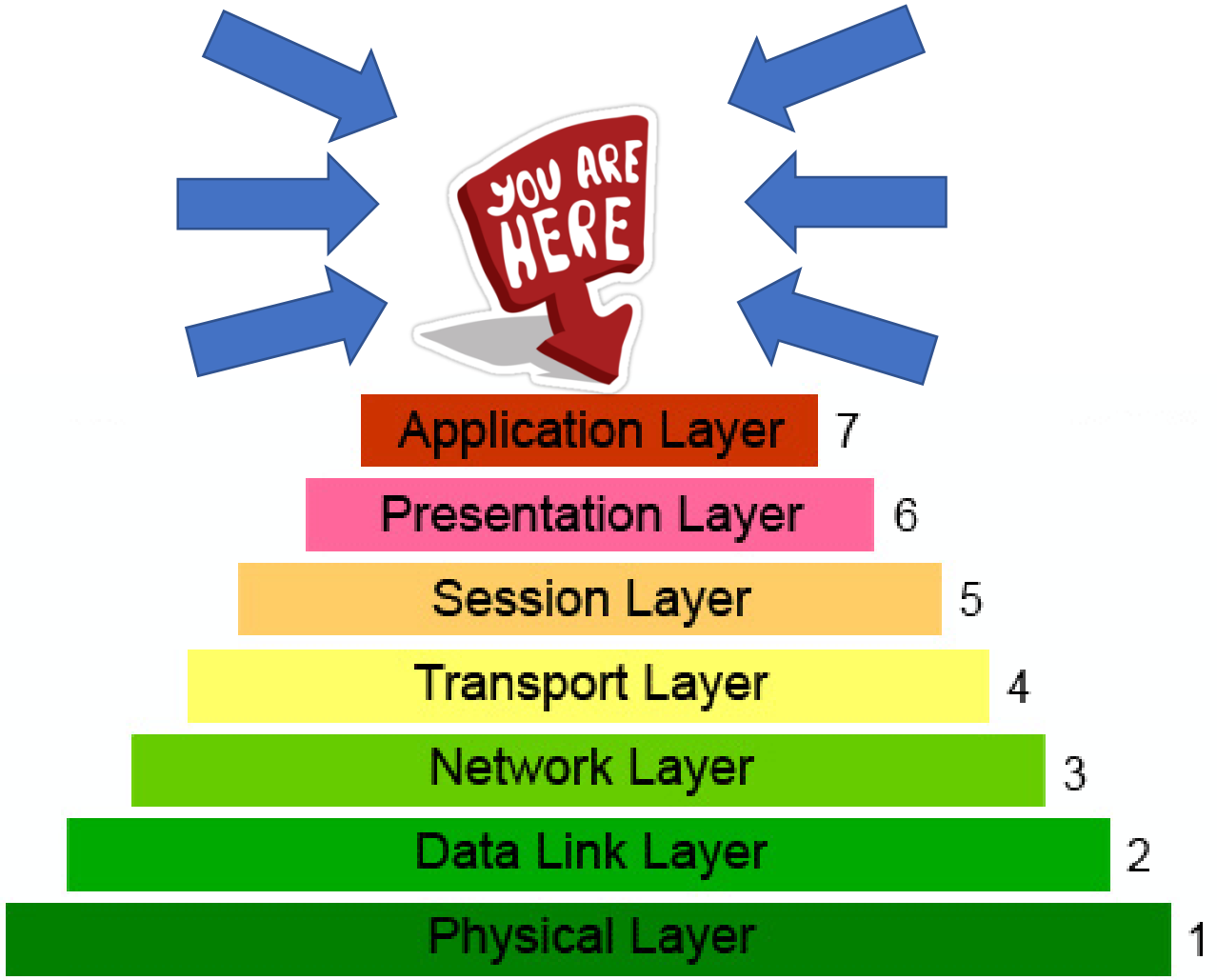
To have a system which even if compromised (**revealed password**) will remain secure from a functional authentication perspective.

Is that even possible?

What is the problem with these setups?

Are there any improvements we can make?

The most important security layer



How can this 8-th layer actually come into play ?

Steganography is the practice of concealing a file, message, image, or video within another file, message, image, or video.

The word steganography combines the Greek words **steganos** (στεγανός), meaning "covered, concealed, or protected", and **graphein** (γράφειν) meaning "writing".

source: Wikipedia

My birthday is: 01 February 1983

Let’s change the date format: 01.02.1983

Let’s turn it into a numerical code:

01021983

To express the same idea into a different format we could say: *“At 01 hours and 02 minutes you should ping the equipment 47 times in order to access it and then 111 times.”*

01021983

IP header 20B, ICMP header 8B
47 - 20 - 8 = **19**, 111 - 20 - 8 = **83**

01021983

Let’s read the true logic decoded:

- protocol 1
- 2-nd packet relevant
- length 19, encrypted length 83

Protocol 1: ICMP, echo request (ping)

2-nd packet relevant: the MikroTik router will only process the odd packets

length 19, encrypted length 83: the MikroTik router will respond in a receptive manner if two sets of data are involved: unencrypted 19 bytes, the second sent over encryption 83 bytes

So for a highly secured device the algorithm is as follows:

- drop icmp echo request of size 19 (True size: 19 + 20 + 8 = 47) and add source to trusted1
- allow encrypted connections from trusted1
- drop icmp echo request of size 83 over encryption (True size: 83 + 20 + 8 = 111) and add source to trusted2
- allow to port 8291 encrypted from trusted2
- allow discovery over encryption (optional)
- drop everything else

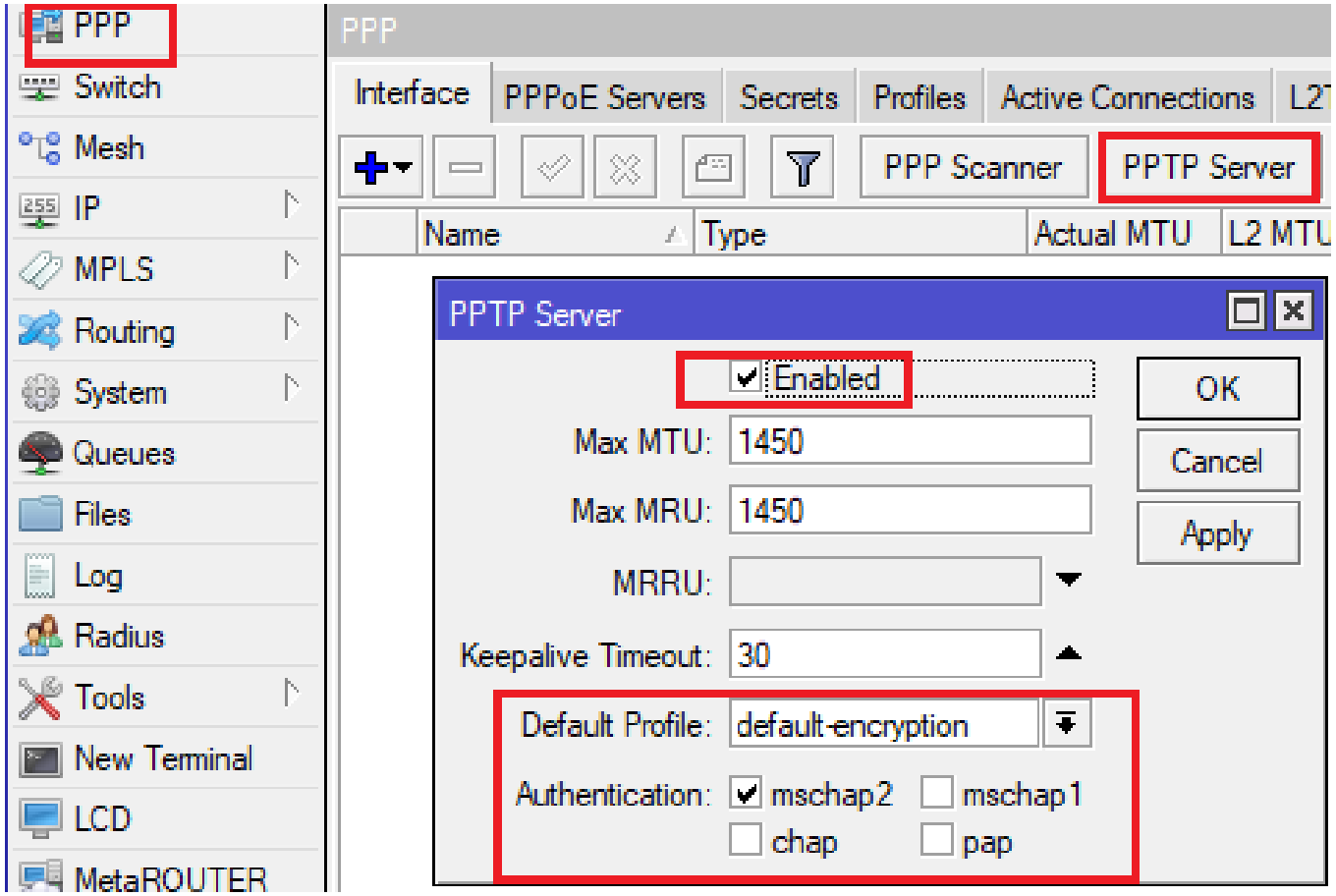
Setting up the authentication system:

1. Set up a PPTP server (you can use any type of more advanced tunneling server SSTP with SSL, L2TP over IPSec etc.)

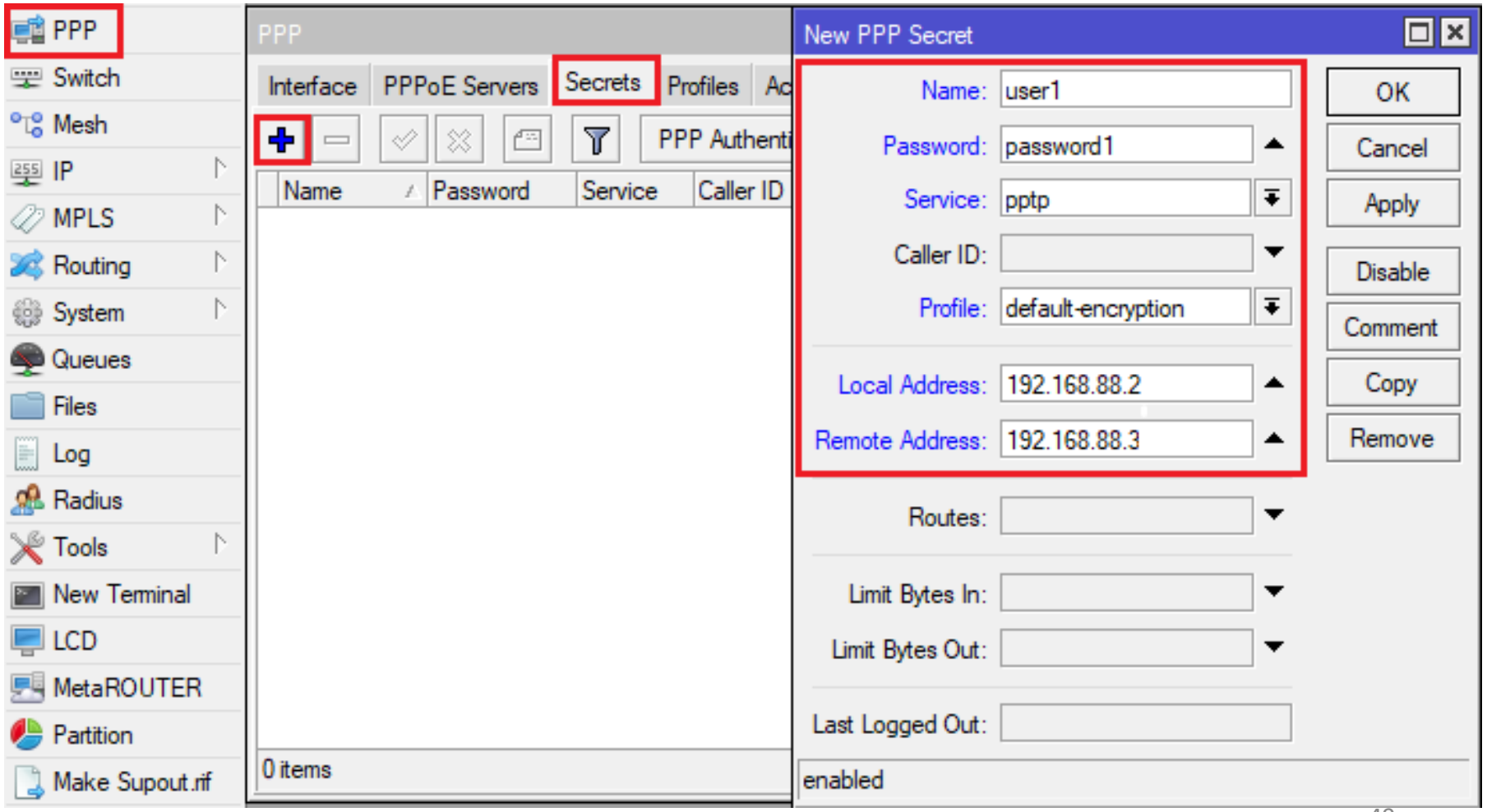
2. Set up a PPTP user and password

3. Set up the firewall

PPTP server


PPP
Switch
Mesh
IP
MPLS
Routing
System
Queues
Files
Log
Radius
Tools
New Terminal
LCD
MetaROUTER
PPP
Interface
PPPoE Servers
Secrets
Profiles
Active Connections
PPP Scanner
PPTP Server
Name
Type
Actual MTU
L2 MTU
PPTP Server
Enabled
Max MTU: 1450
Max MRU: 1450
MRRU:
Keepalive Timeout: 30
Default Profile: default-encryption
Authentication: mschap2 mschap1 chap pap
OK
Cancel
Apply

PPTP user and password



Firewall rules

Terminal

```
[admin@MikroTik] > ip firewall filter export
# jan/02/1970 05:47:52 by RouterOS 6.43.4
# software id = N53Q-VEGT
#
# model = 2011UiAS-2HnD
# serial number = 614A05385A6A
/ip firewall filter
add action=add-src-to-address-list address-list=Trusted1 address-list-timeout=\
    1h chain=input connection-bytes=47 icmp-options=8:0-255 nth=2,1 protocol=\
    icmp
add action=accept chain=input dst-port=1723 protocol=tcp src-address-list=\
    Trusted1
add action=accept chain=input protocol=gre src-address-list=Trusted1
add action=add-src-to-address-list address-list=Trusted2 address-list-timeout=\
    1h chain=input connection-bytes=111 icmp-options=8:0-255 in-interface=\
    all-ppp nth=2,1 protocol=icmp
add action=accept chain=input dst-port=8291 protocol=tcp src-address-list=\
    Trusted2
add action=accept chain=input dst-port=20561 protocol=udp src-address-list=\
    Trusted2
add action=drop chain=input
[admin@MikroTik] >
```

Firewall rules applied

Terminal

```
[admin@MikroTik] > ip firewall filter export
# jan/02/1970 05:47:52 by RouterOS 6.43.4
# software id = N53Q-VEGT
#
# model = 2011UiAS-2HnD
# serial number = 614A05385A6A
/ip firewall filter
add action=add-src-to-address-list address-list=Trusted1 address-list-timeout=\
    1h chain=input connection-bytes=47 icmp-options=8:0-255 nth=2,1 protocol=\
    icmp
add action=accept chain=input dst-port=1723 protocol=tcp src-address-list=\
    Trusted1
add action=accept chain=input protocol=gre src-address-list=Trusted1
add action=add-src-to-address-list address-list=Trusted2 address-list-timeout=\
    1h chain=input connection-bytes=111 icmp-options=8:0-255 in-interface=\
    all-ppp nth=2,1 protocol=icmp
add action=accept chain=input dst-port=8291 protocol=tcp src-address-list=\
    Trusted2
add action=accept chain=input dst-port=20561 protocol=udp src-address-list=\
    Trusted2
add action=drop chain=input
[admin@MikroTik] >
```

Firewall

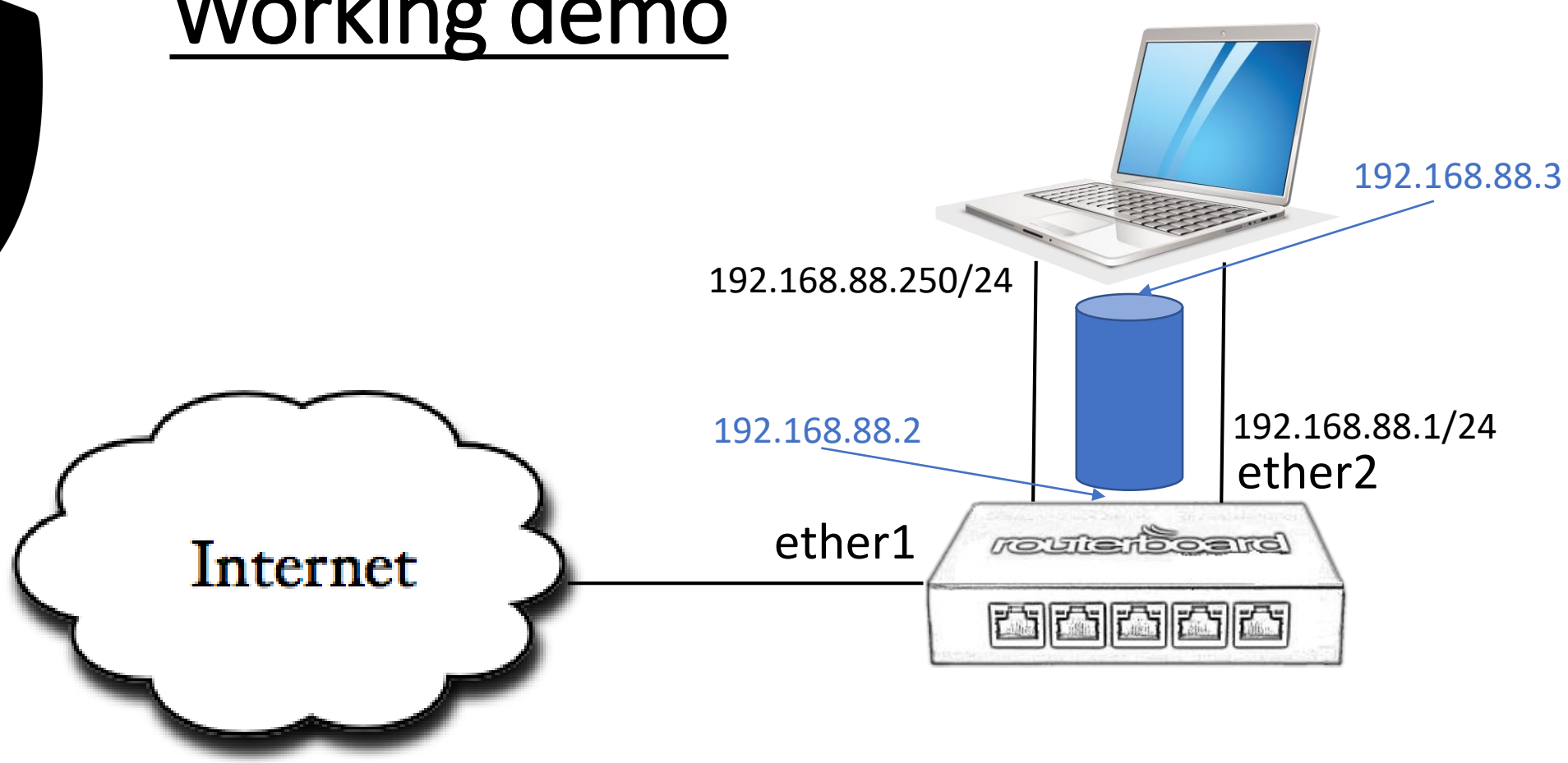
Filter Rules | NAT | Mangle | Raw | Service Ports | Connections | Address Lists | Layer7 Protocols

oo Reset Counters | oo Reset All Counters | Find | all

#	Action	Chain	Protocol	Dst. Port	In. Interface	Src. Address List	Connection Bytes	ICMP Options/...	Nth/Every	Nth/Packet	Address List	Timeout	Bytes	Packets
0	add src to address list	input	1 (icmp)				47	8 (echo request)	2	1	Trusted1	01:00:00	0 B	0
1	accept	input	6 (tcp)	1723		Trusted1							0 B	0
2	accept	input	47 (gre)			Trusted1							0 B	0
3	add src to address list	input	1 (icmp)		all ppp		111	8 (echo request)	2	1	Trusted2	01:00:00	0 B	0
4	accept	input	6 (tcp)	8291		Trusted2							0 B	0
5	accept	input	17 (udp)	20561		Trusted2							0 B	0
6	drop	input											1367.2 KiB	19 359

7 items

Working demo



Final conclusions

A subtle but important difference in logic makes the difference in security.

By making use of this technique, a system which has a compromised admin password will remain unaccessible by attackers.

Questions?