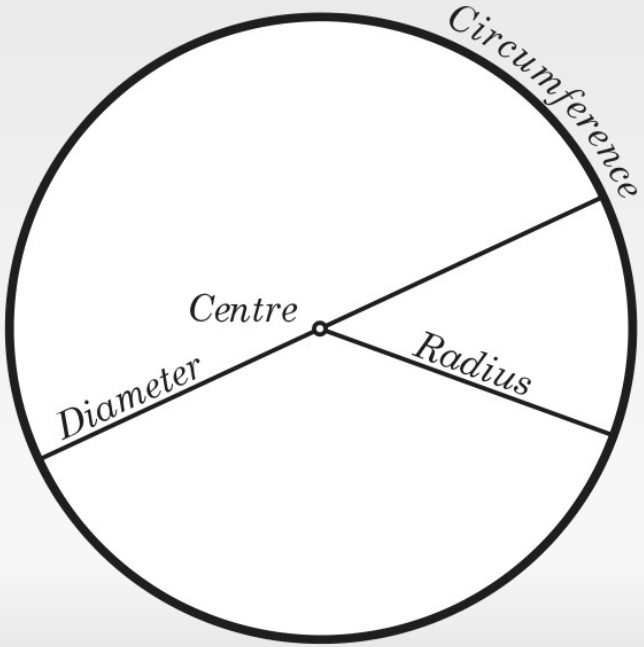
RADIUS

- make life easier



by Daniel Starnowski

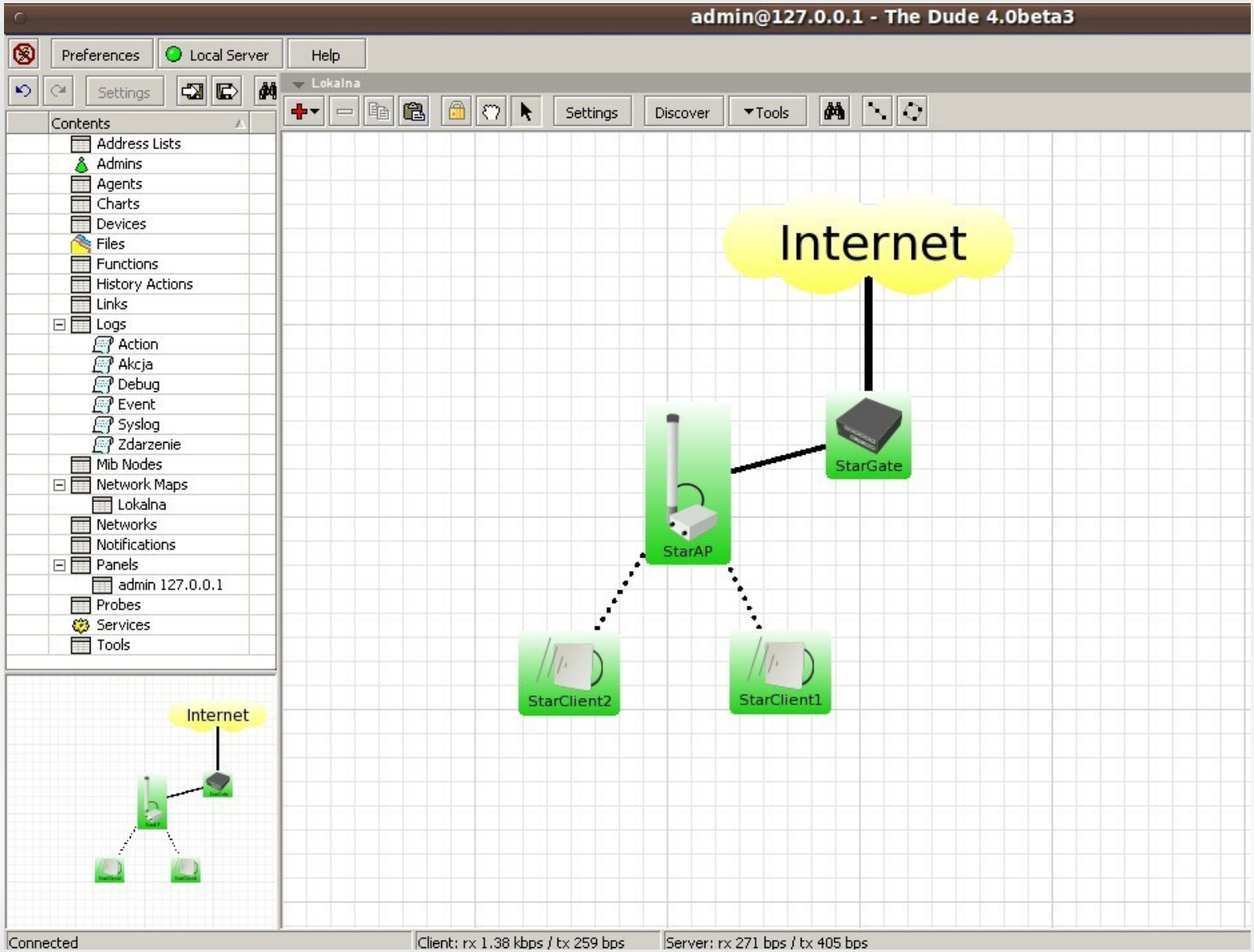
About me

- Daniel Starnowski
- Network administrator since 2000
- MikroTik user since 2008
- **MikroTik Trainer** since 2011
- From Kraków, Poland
  - 1038-1596 capital of Poland
  - 2007 Mikrotik User Meeting
- http://startik.net

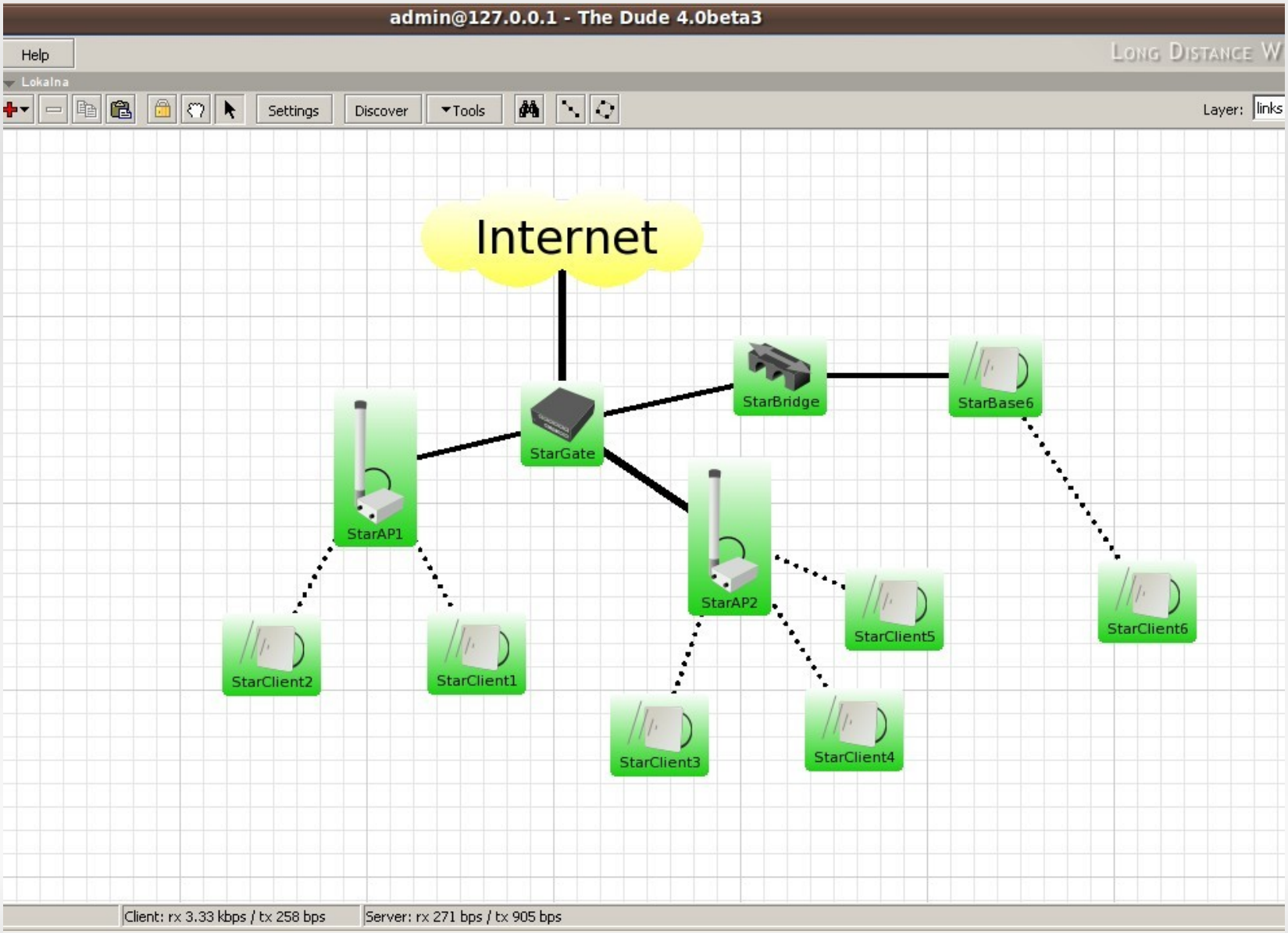
Outline

- Introduction
- FreeRADIUS – quick install
- Example: login management
- Connecting do SQL database
- Short example: wireless
- Example: DHCP (and modifying SQL query)
- Hotspot: MAC authorization & HTML redirection
- How to create a management platform in PHP

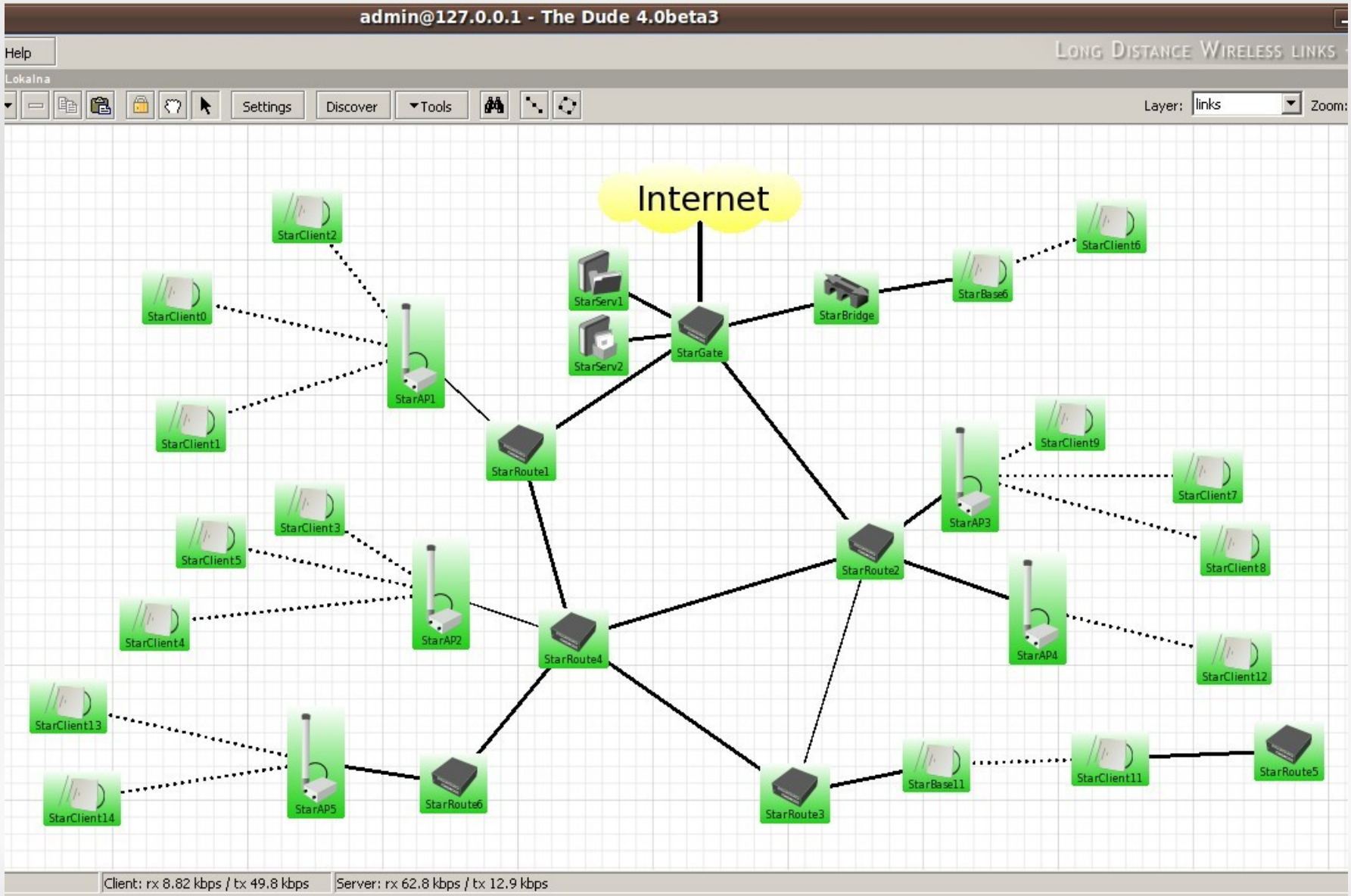
Introduction



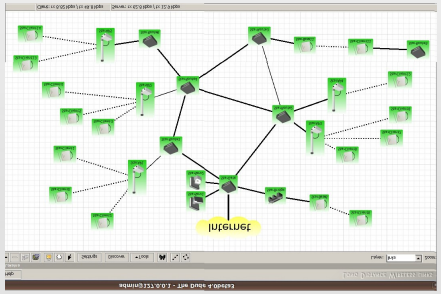
Introduction



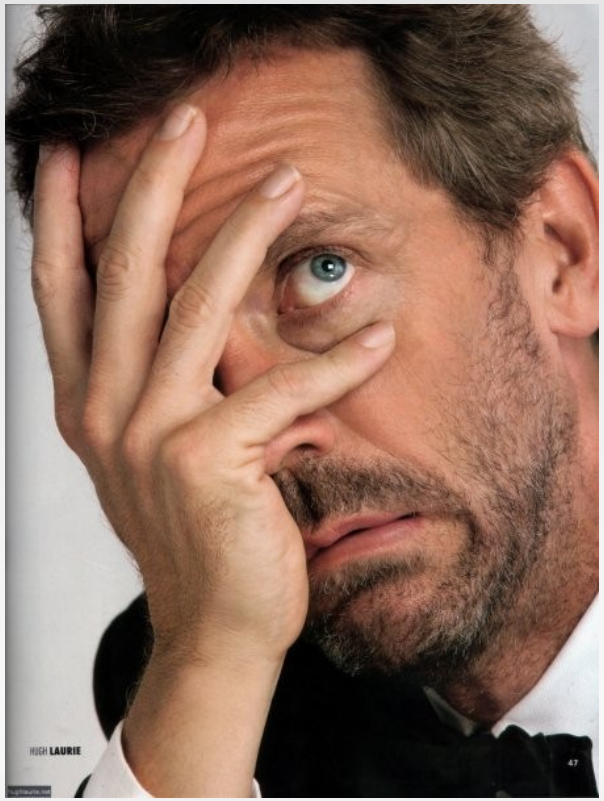
Introduction



Introduction



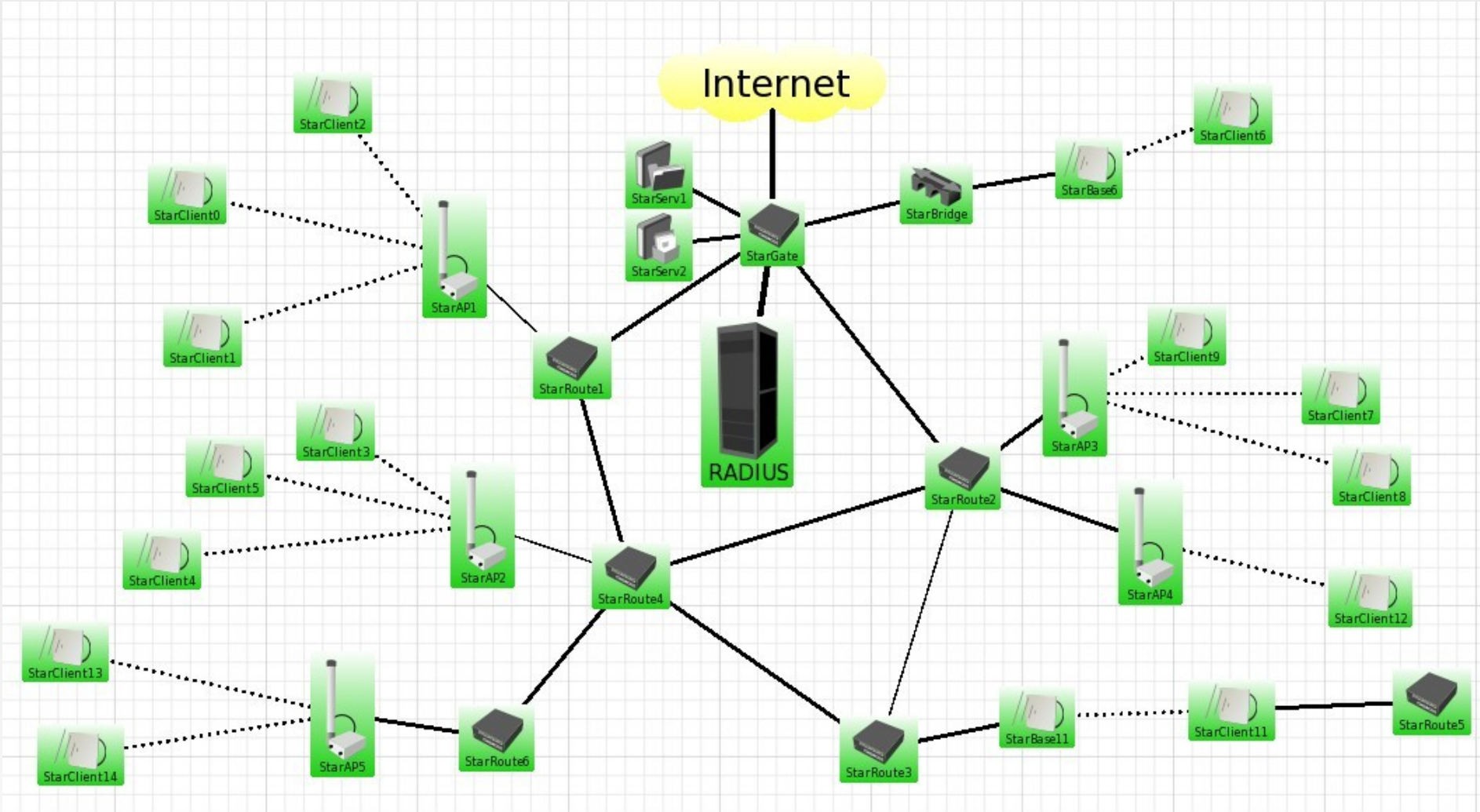
- More devices = more problems
- Inconsistent login configuration
- Authorization and queueing for customers on the nearest router – very problematic and hard to manage



RADIUS – the protocol

- Remote Authentication Dial In User Service
- RFC 2865
- uses UDP ports 1812 and 1813
- AAA concept
  - Authentication
  - Authorization
  - Accounting

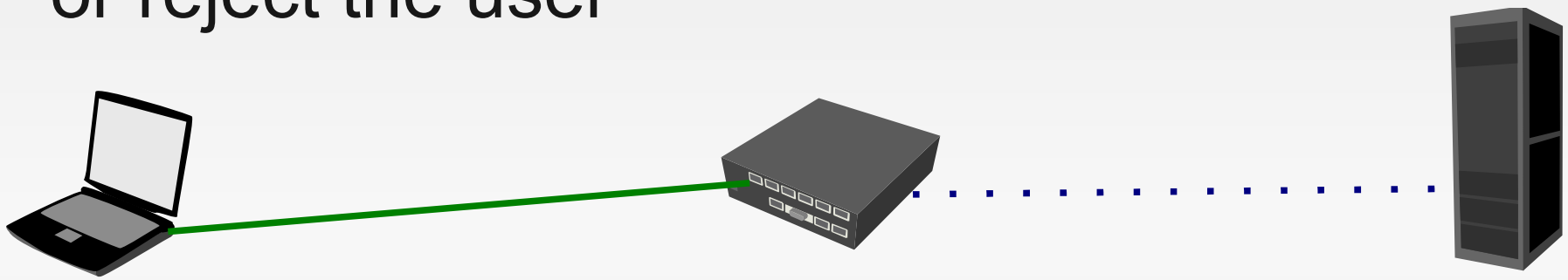
RADIUS



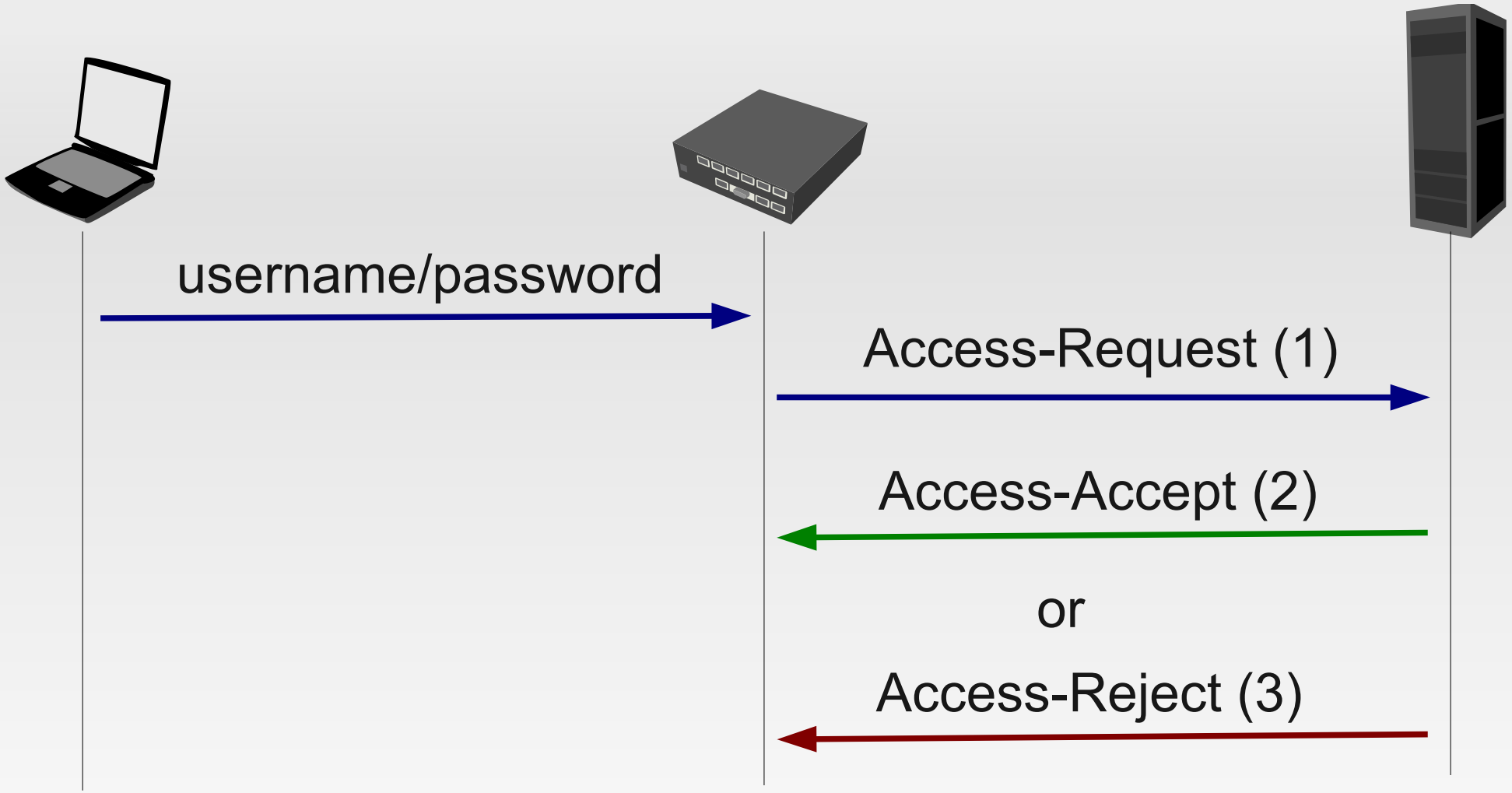
- One server can centralize all **user** accounts

RADIUS – server, client, user

- **User** (a computer) tries to connect to the gateway (ppp, hotspot, etc.) using username and password
- **Client** (MikroTik) looks for the user in local database and if it fails – asks RADIUS server
- **Server** – tell the client whether it should accept or reject the user

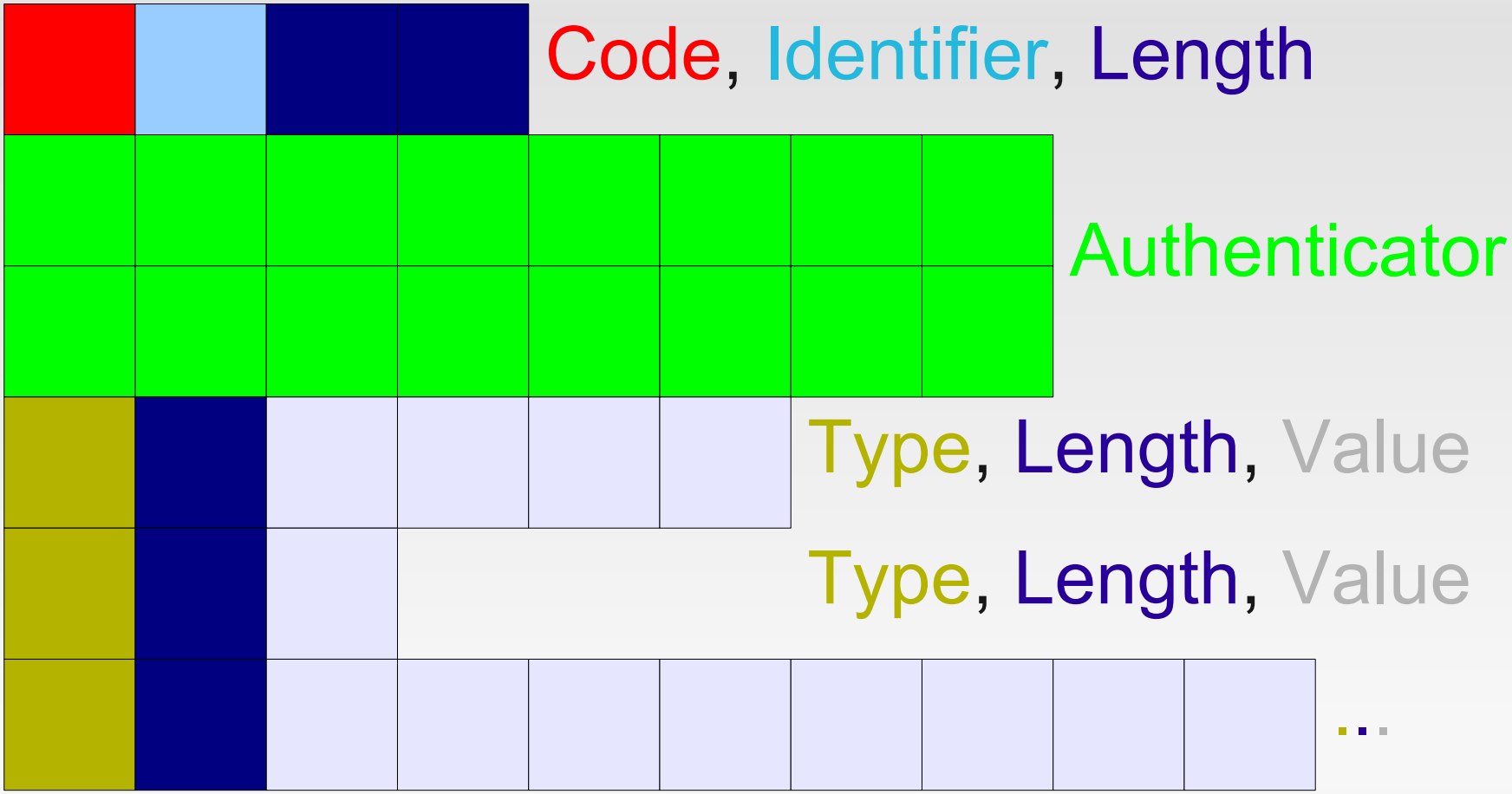


RADIUS – request and response



- Request and response – single UDP packets

Radius – the packet



FreeRADIUS – quick install

- Installation of FreeRADIUS is really easy!
- Ubuntu: **sudo apt-get install freeradius**
- /etc/freeradius – directory with the settings
- clients.conf – the only file we **need** to edit:

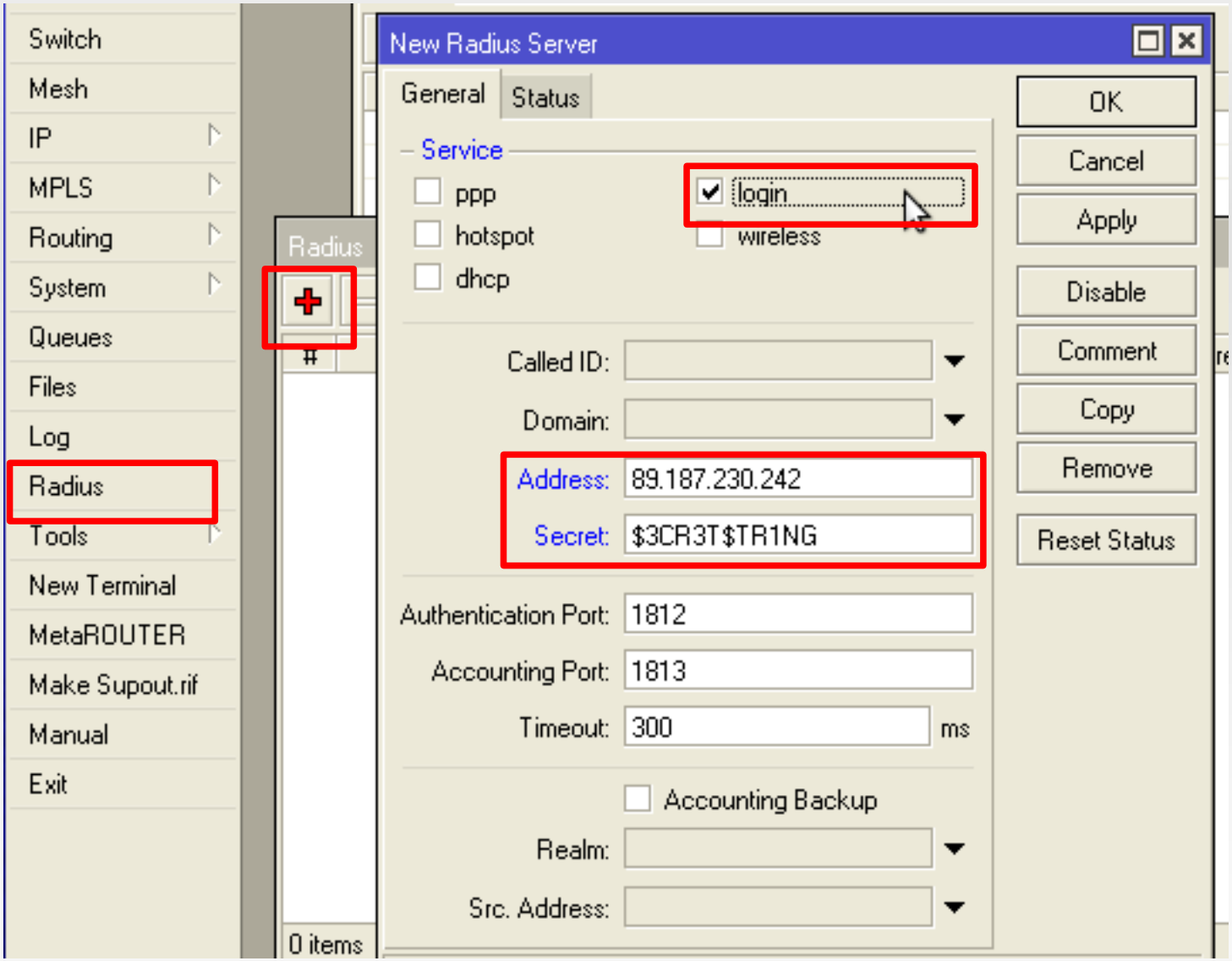
```
client 192.168.255.1/32 {
        secret          = $3CR3T$TR1NG
        shortname       = MikroTik }
```

- We specify addresses accepted by the server

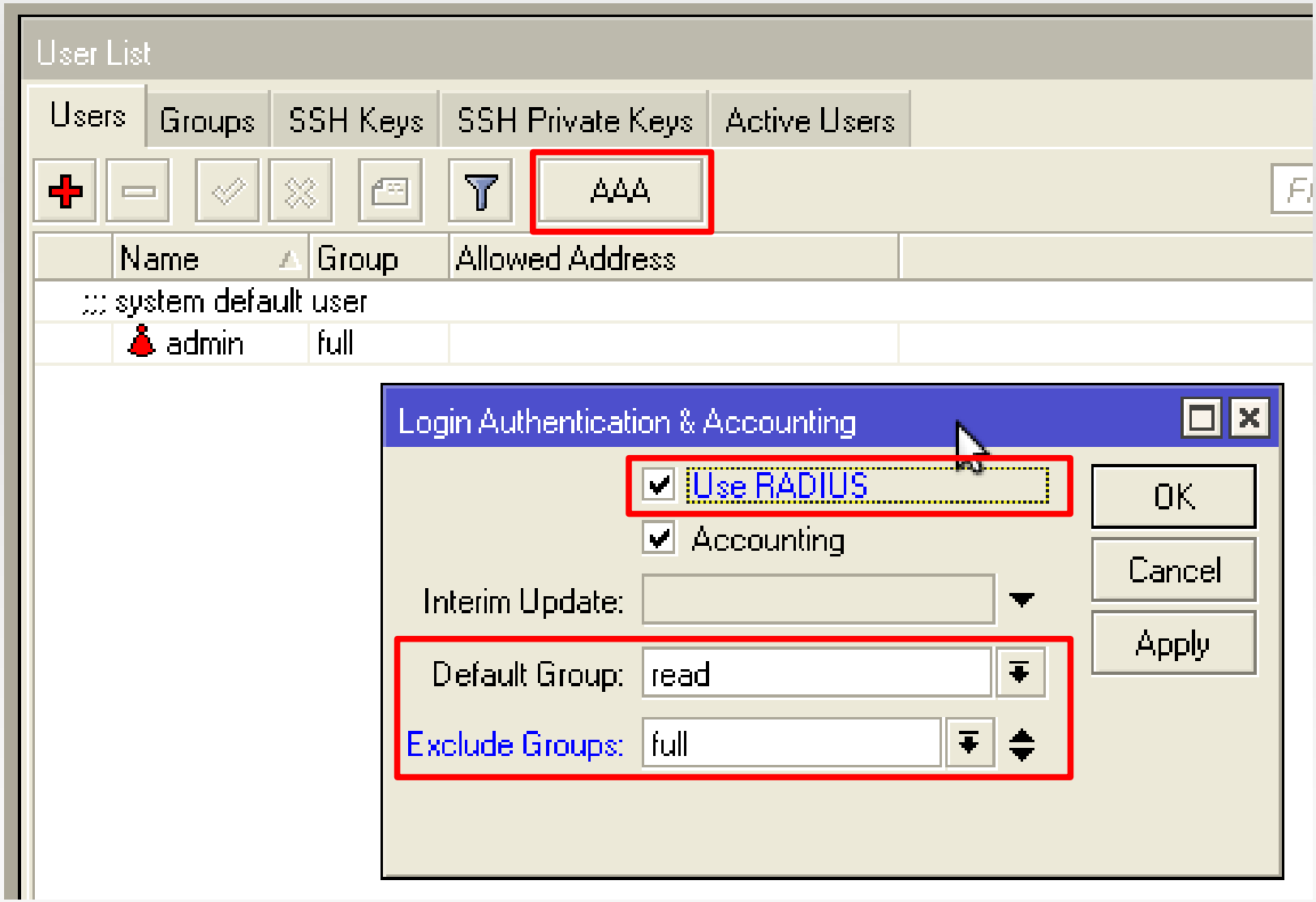
RADIUS – dictionaries

- /usr/share/freeradius/ - dictionary files
- dictionary.rfc2865:
- ```
  ATTRIBUTE User-Name             1  string
  ATTRIBUTE User-Password         2  string
                                    encrypt=1
  ATTRIBUTE CHAP-Password         3  octets
  ATTRIBUTE NAS-IP-Address        4  ipaddr
  ATTRIBUTE NAS-Port              5  integer
  ATTRIBUTE Service-Type          6  integer
  ATTRIBUTE Framed-Protocol       7  integer
  ATTRIBUTE Framed-IP-Address     8  ipaddr
  ATTRIBUTE Framed-IP-Netmask     9  ipaddr
  ```
  ```

FreeRADIUS – quick install



Example: login management



Example: login management

- File users in /etc/freeradius
- `username Cleartext-Password := ”password”`
- User ”username” with password ”password” will be accepted by the router, with default group
- `username Cleartext-Password := ”password”`
  `Mikrotik-Group := ”write”,`
  `Another-Attr := ”a_value”`
- We can specify, what attributes the RADIUS server will give in the response

Example: login management

- Access-Request:
  - Service-Type = Login-User
  - User-Name = (name entered by user)
  - User-Password = (encrypted password)
  - Calling-Station-Id = (IP address of the **user)**
  - NAS-Identifier = (system identity of client)
  - NAS-IP-Address = (IP address of the **client**)

Example: login management

- Access-Accept
- If there was no configured parameters, the accept packet has no ”attribute-value” fields
- example: Mikrotik-Group = ”write”

```
[startik@StarTik] > user active print detail
Flags: R - radius
 0   when=jan/02/1970 00:05:07 name="admin" address=192.168.133.112 via=winbox group=full

 1 R when=mar/10/2012 13:00:26 name="startik" address=192.168.133.113 via=telnet group=read
[startik@StarTik] >
```

Connecting to SQL database

- `sudo apt-get install mysql-server-5.1`
- `sudo apt-get install freeradius-mysql`
- /etc/freeradius/sql/mysql/ - here are configuration files for Radius to work with SQL
- `mysql> CREATE DATABASE radius;`
- We import schema.sql (or just simply paste the commands from the file) to MySQL database

Connecting to SQL database

- Back to radiusd.conf – in the ”modules” section we enable (uncomment) the SQL module:

```
#         $INCLUDE sql.conf
```

- In the sql.conf file:

```
database = "mysql"
server = "localhost"
login = "db_user"
password = "his_password"
radius_db = "radius"
```

Creating SQL entries

- Instead of the users file - two tables:
  - radcheck
  - radreply
- They look exactly the same!
- In **radcheck** – the conditions to be checked
- In **radreply** – the attributes sent with the reply packet

Creating SQL entries

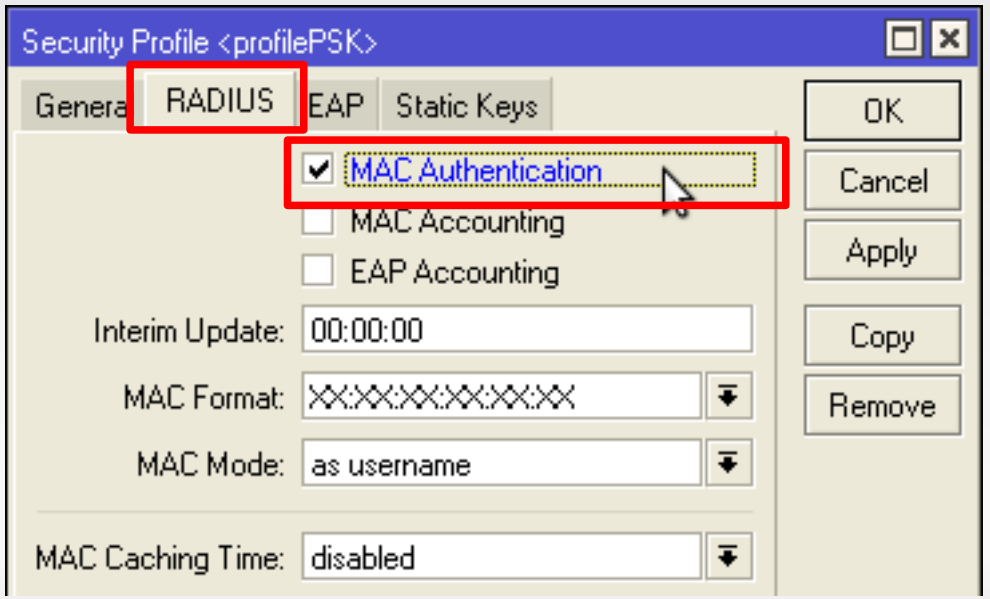
```
mysql> show fields from radcheck;
+-----------+-------------------+
| Field     | Type              |
+-----------+-------------------+
| id        | int(11) unsigned  |
| username  | varchar(64)       |
| attribute | varchar(64)       |
| op        | char(2)           |
| value     | varchar(253)      |
+-----------+-------------------+
5 rows in set (0.00 sec)
```

Creating SQL entries

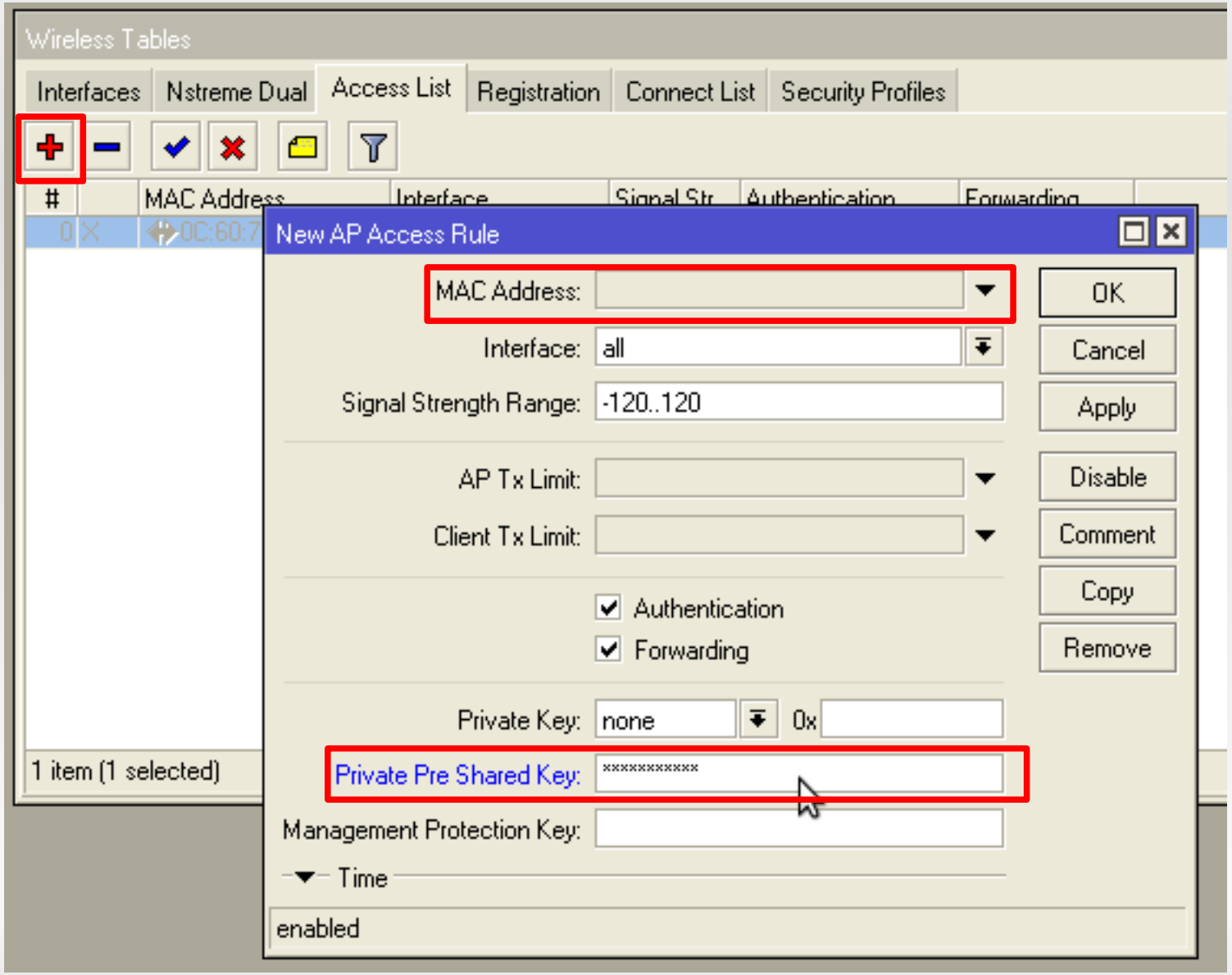
- ```
  INSERT INTO radcheck
  (username, attribute, op, value)
  VALUES
  ('user','Cleartext-Password',':=','pass');
  ```
- ```
  INSERT INTO radreply
  (username, attribute, op, value)
  VALUES
  ('user','Mikrotik-Group',':=','write');
  ```
- Exactly like in the users file:
- ```
  user Cleartext-Password := "pass"
       Mikrotik-Group  := "write"
  ```
  ```

Short example: wireless

- For wireless – RADIUS works similar to ”Access List” and ”Connect List” - decides, which stations can get to the registration table
- Configured in the Security Profile
- ”Default Authenticate” stops working!



Short example: wireless

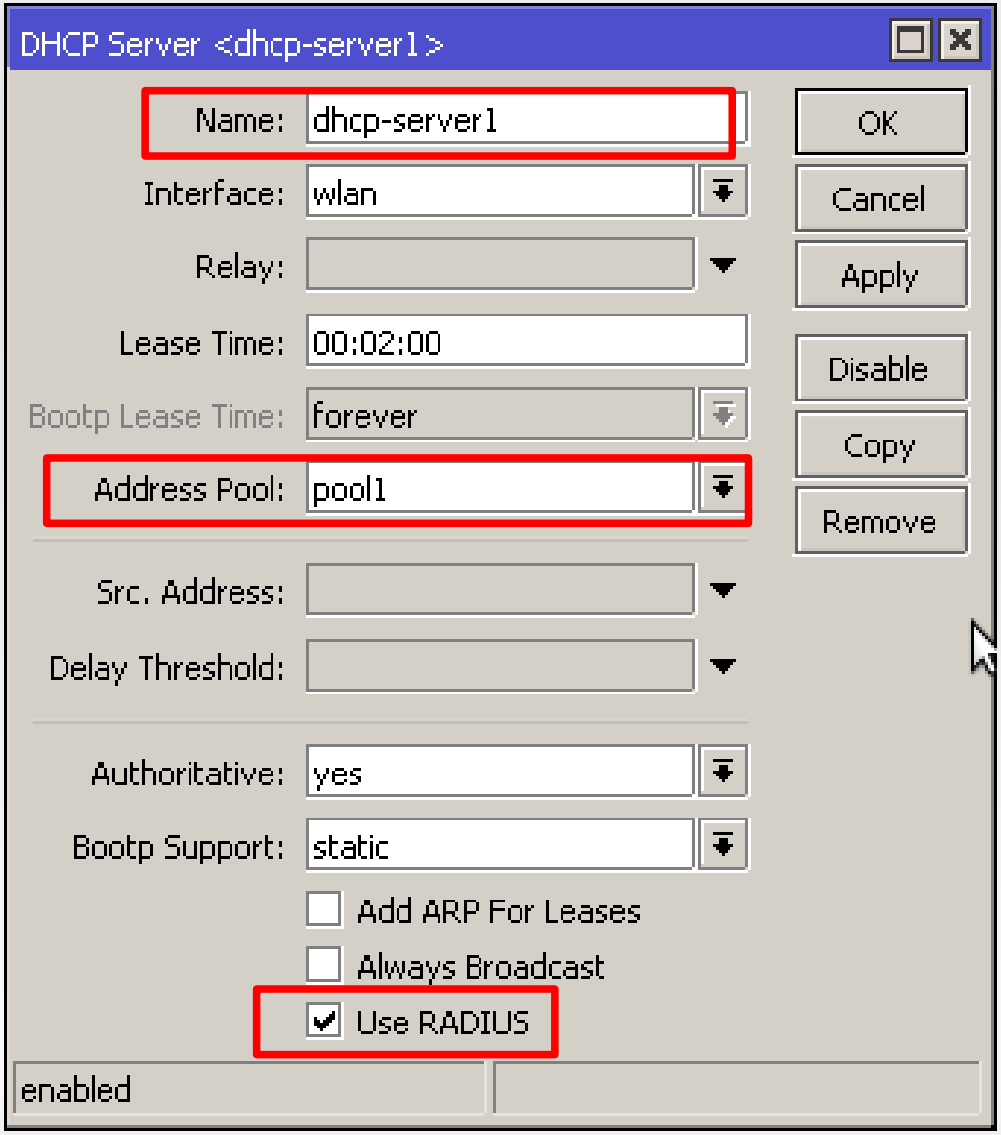


Short example: wireless

- ```
  INSERT INTO radcheck
  (username, attribute, op, value)
  VALUES
  ('00:0C:42:01:02:03',
   'Auth-Type',':=','Accept');
  ```
- ```
  INSERT INTO radreply
  (username, attribute, op, value)
  VALUES
  ('00:0C:42:01:02:03',
   'Mikrotik-Wireless-PSK',':=','PSKstring');
  ```

Example: DHCP

- MAC authorized and has ”Framed-IP-Address” in the reply: it will get the specific address
- MAC is authorized but without reserved IP: it will get it from the pool
- MAC not authorized: won't get any address!



Example: DHCP

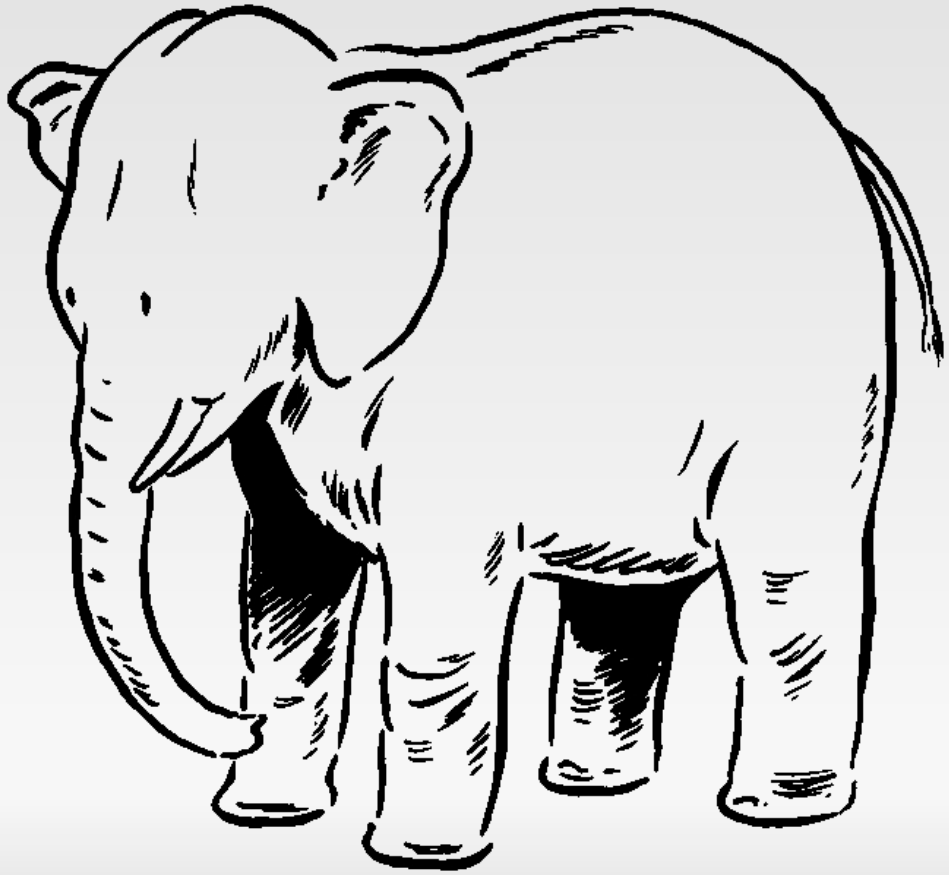
- ```
  INSERT INTO radcheck
  (username, attribute, op, value)
  VALUES
  ('00:0C:42:01:02:03',
   'Auth-Type',':=','Accept');
  ```
- Wait... we already have this one!
- ```
  INSERT INTO radreply
  (username, attribute, op, value)
  VALUES
  ('00:0C:42:01:02:03',
   'Framed-IP-Address',':=','172.17.2.2');
  ```

Example: DHCP

- We have the same MAC address for wireless and for DHCP services!
- RADIUS will reply with all attributes to every service
- Wireless will get **Mikrotik-Wireless-PSK**, but ignore **Framed-IP-Address**
- DHCP will get **Framed-IP-Address**, but ignore **Mikrotik-Wireless-PSK**

Example: DHCP

- If a MAC address is not in the RADIUS database (it is not authorized) – it will not get a DHCP lease!!
- What can we do to prevent it?



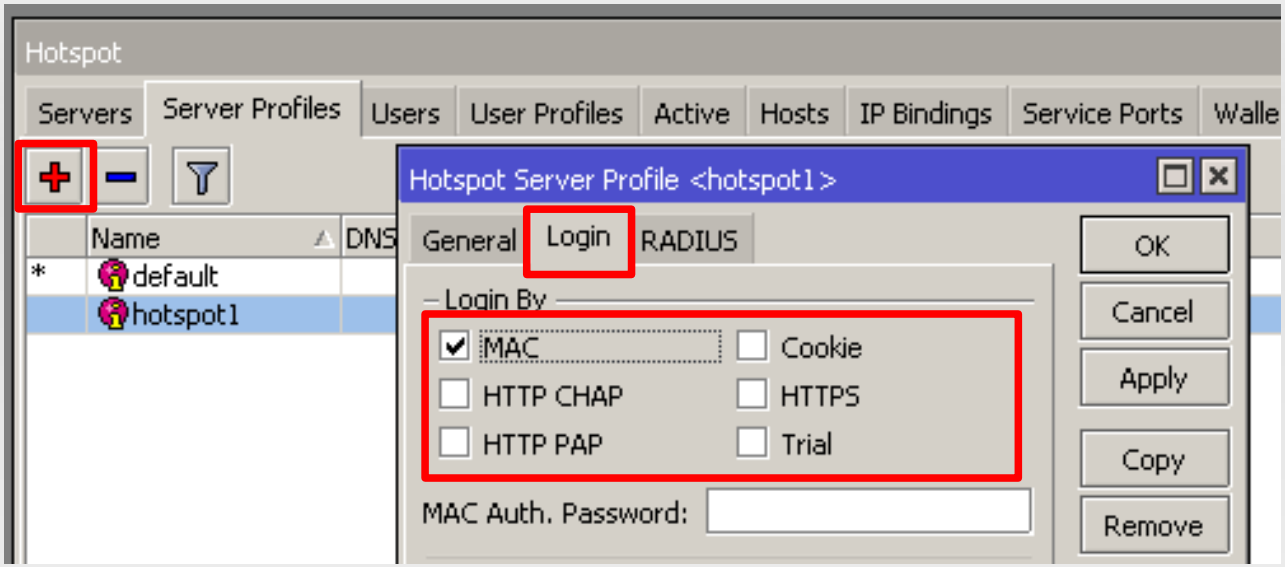
Modifying SQL query

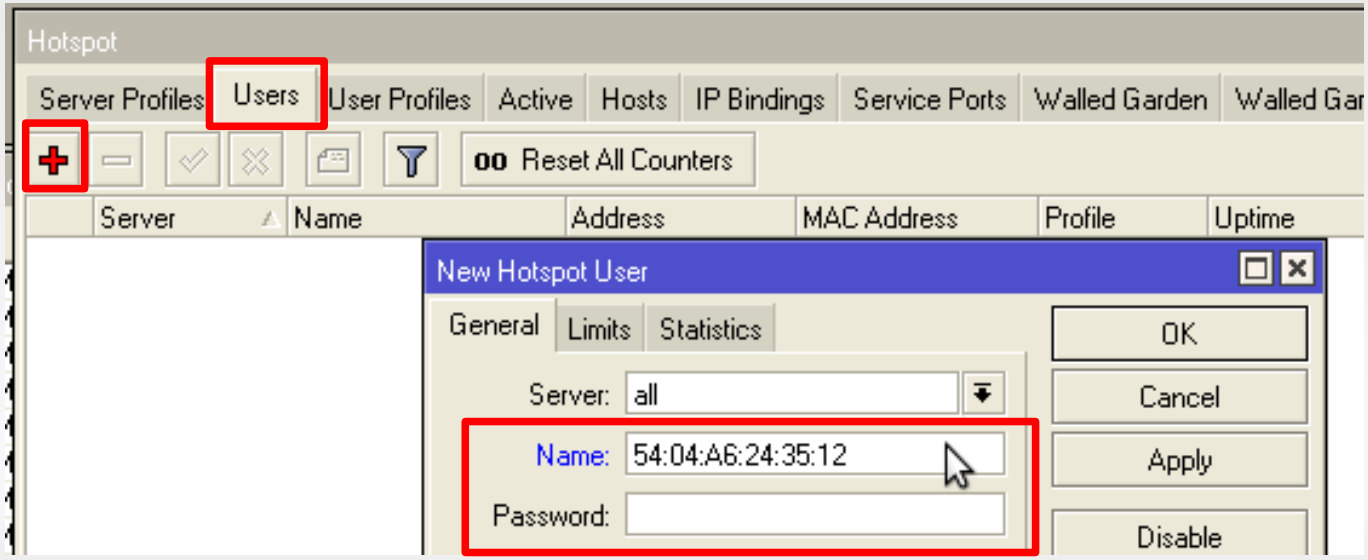
- In dialup.conf file – we have the exact SQL query used to get the data from database:
- ```
  authorize_check_query =
  "SELECT
  id, username, attribute, value, op \
  FROM ${authcheck_table} \
  WHERE username = '%{SQL-User-Name}' \
  ORDER BY id"
  ```
- We can modify it, so that for every request from DHCP server it will give Auth-Type := Accept
  ```

Modified SQL query

- ```
  authorize_check_query =
  "SELECT
  id, username, attribute, value, op \
  FROM ${authcheck_table} \
  WHERE username = '%{SQL-User-Name}' \
  UNION \
  SELECT DISTINCT 0, '%{SQL-User-Name}',
  'Auth-Type', 'Accept', ':=' \
  FROM ${authcheck_table} \
  WHERE '%{Called-Station-Id}' like 'dhcp%' \
  ORDER BY id"
  ```
- Now **every MAC** will get an IP address from the DHCP!
- `0,'54:04:A6:24:35:12','Auth-Type',':=','Accept'`
  ```

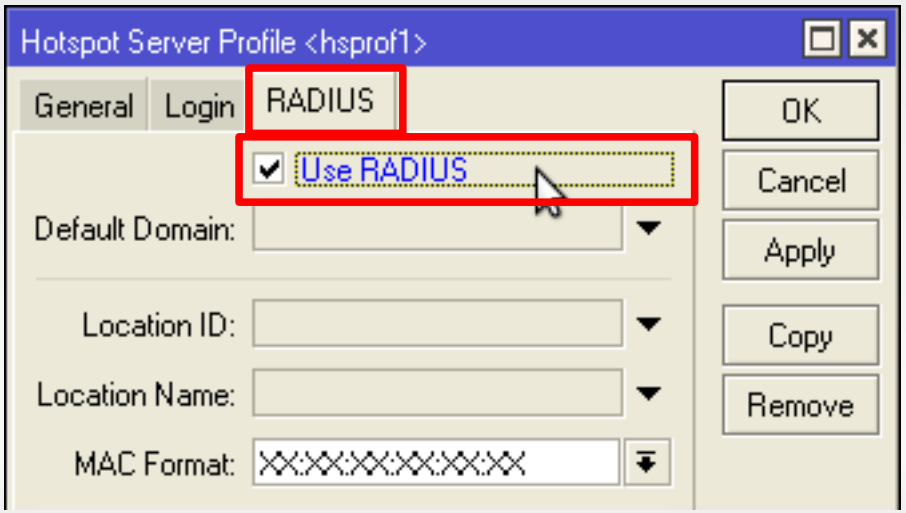
Hotspot: MAC authorization





Hotspot: MAC authorization

- If a user (MAC address) is not present in the Users list of the hotspot, it will be checked in the RADIUS database
- Only authorized users will access the network, unauthorized will get the login.html page

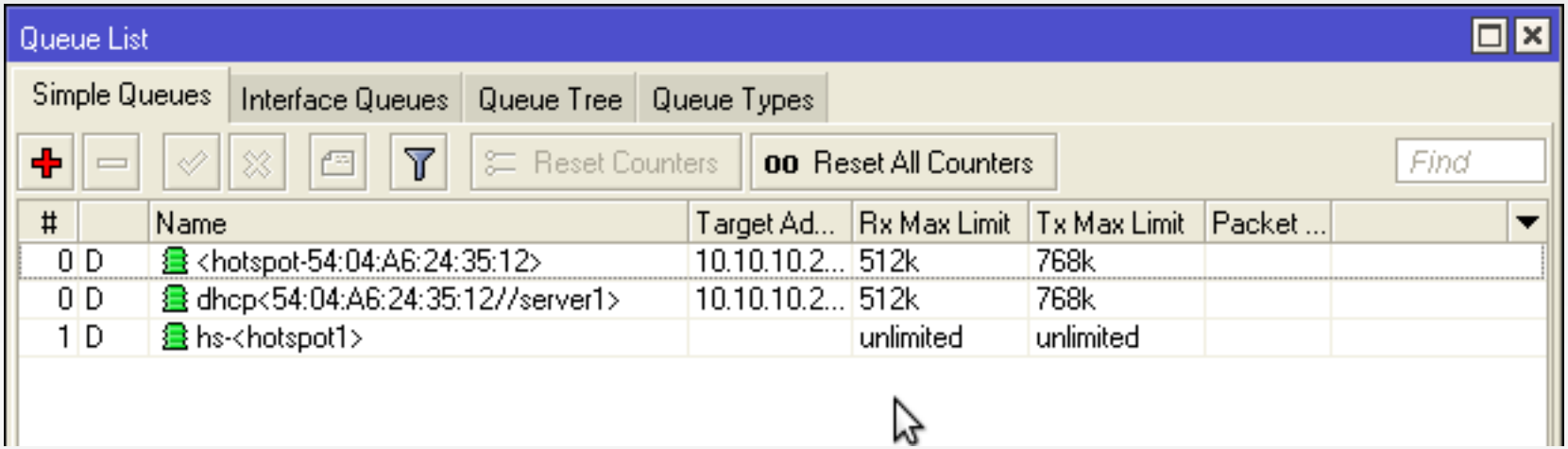


Hotspot: MAC authorization

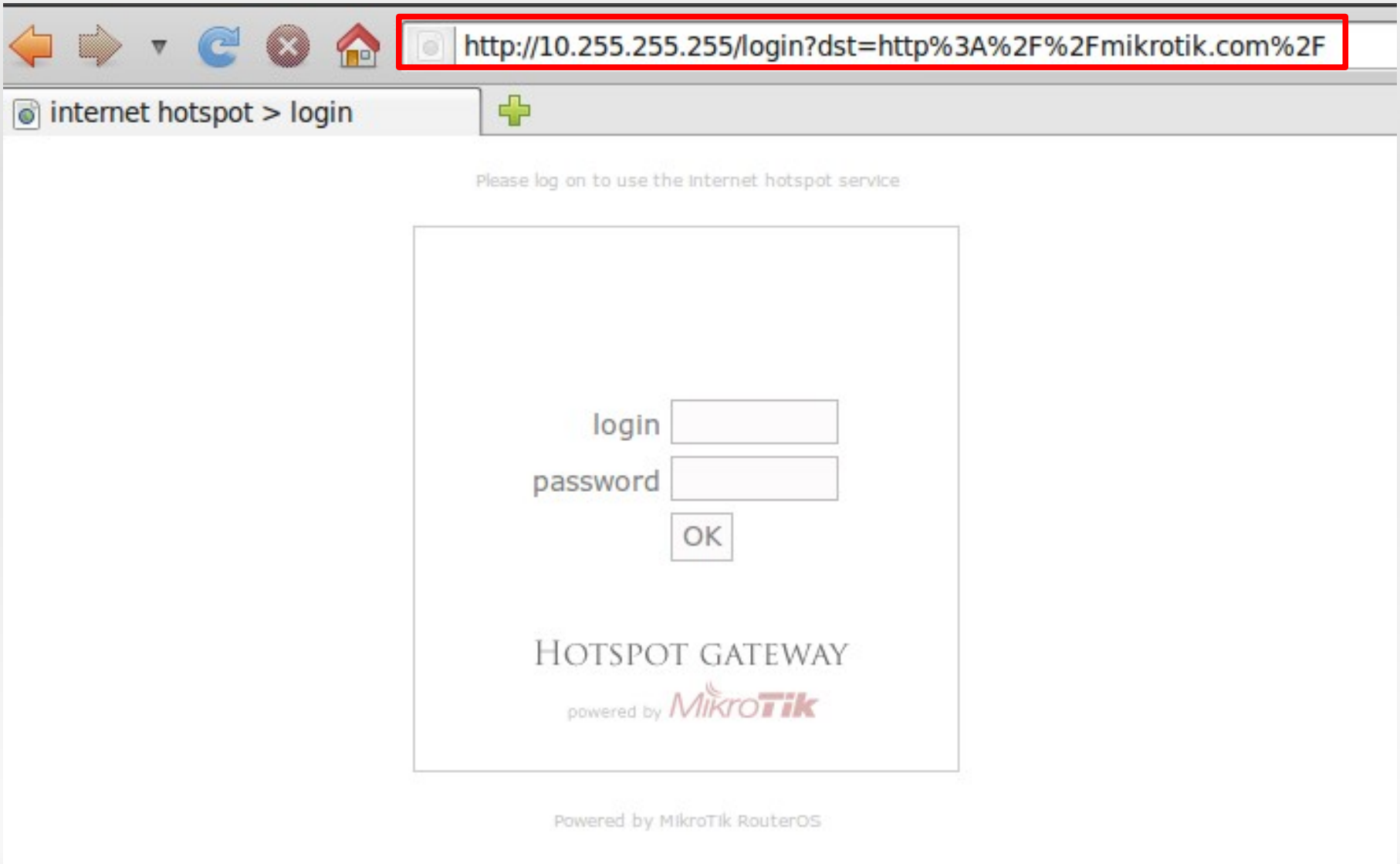
- The MAC address will be authorized, if it will pass the radcheck query (i.e. will be present as username in the radcheck table)
- Additional reply attributes possible, like limits for the up/down/total bytes or connection time
- `Mikrotik-Rate-Limit := ”256k/512k”`
- Rate Limit will create a dynamic simple queue with the max-limit restrictions.

Hotspot: MAC authorization

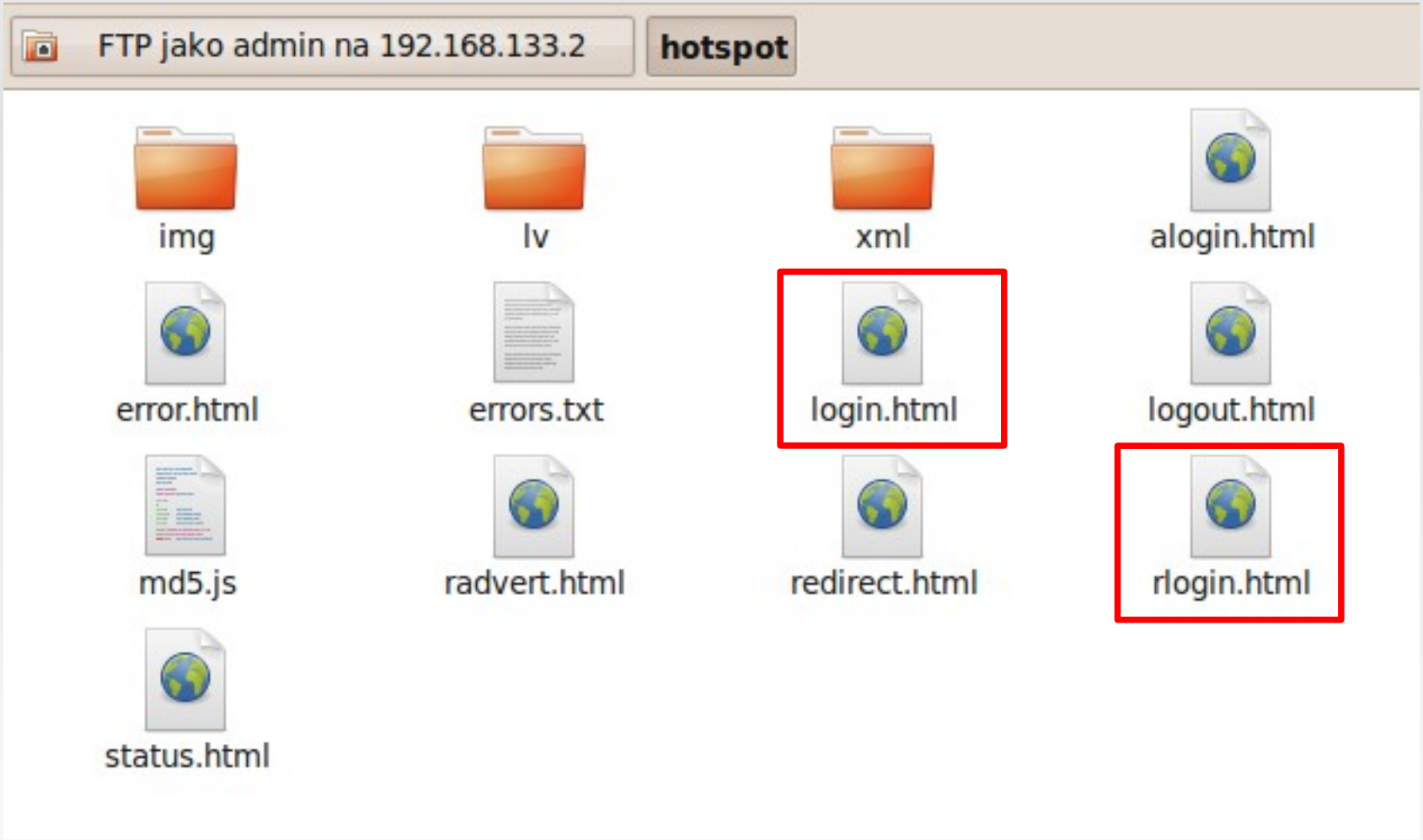
- If both DHCP and Hotspot services get data from the same RADIUS database – the queue will be created twice!
- It can be avoided by modifying the reply SQL query



Hotspot: HTML redirection



Hotspot: HTML files



Hotspot: HTML files (rlogin)

```
$(if http-status == 302)Hotspot login required$(endif)
$(if http-header == "Location")$(link-redirect)$(endif)
<html>
<!--
<?xml version="1.0" encoding="UTF-8"?>
  <WISPAccessGatewayParam
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://$(hostname)/xml/WISPAccessGatewayParam.xsd">
    <Redirect>
        <AccessProcedure>1.0</AccessProcedure>
        <AccessLocation>$(location-id)</AccessLocation>
        <LocationName>$(location-name)</LocationName>
        <LoginURL>$(link-login-only)?target=xml</LoginURL>
        <MessageType>100</MessageType>
        <ResponseCode>0</ResponseCode>
    </Redirect>
  </WISPAccessGatewayParam>
-->
<head>
<title>...</title>
<meta http-equiv="refresh" content="0; url=$(link-redirect)">
<meta http-equiv="pragma" content="no-cache">
<meta http-equiv="expires" content="-1">
</head>
<body>
</body>
</html>
```

Hotspot: HTML files

- For **$(link-redirect)** hotspot puts: http://10.255.255.255/login.html?dst=OLD_URL
- We modify the **rlogin.html** page
- Instead of $(link-redirect) we put: http://192.168.255.2/register.php?mac=$(mac)
- 192.168.255.2 – our PHP/MySQL server
- For **$(mac)** hotspot will put user's MAC address
- The http server needs to be added to Hotspot's **Walled Garden**

Management platform

- New SQL table **customers**:
- ```
  +----------+--------------------+
  | Field    | Type               |
  +----------+--------------------+
  | id       | int(11) unsigned   |
  | username | varchar(64)        |
  | password | varchar(64)        |
  +----------+--------------------+
  ```
- Tables **radcheck** and **radreply** get additional field ”customer” (integer)
  ```

Management platform – live demo

- You can connect to the live demo platform!
- SSID = **StarTik**
- All the settings from DHCP server

- Try to open any webpage

RADIUS – make life easier

Any questions?

Thank you!