ya tu aprendes

MikroTik

CERTIFIED TRAINER

TR-069
Survive Customer Reset

WE INSPIRE KNOWLEDGE

Who am I?

I’m Jorge Castellet

I’m a Mikrotik Certified Trainer

MTCNA, MTCIPv6E, MTCRE, MTCTCE, MTCWE

MTCUME, MTCINE, MTCSE

I’m freelance

j.castellet@yatuaprendes.com


– HELLO?
– HELLO?
– INTERNET
DOES NOT WORK!
ya tu aprendes
MikroTik
CERTIFIED TRAINER

Once upon a time…

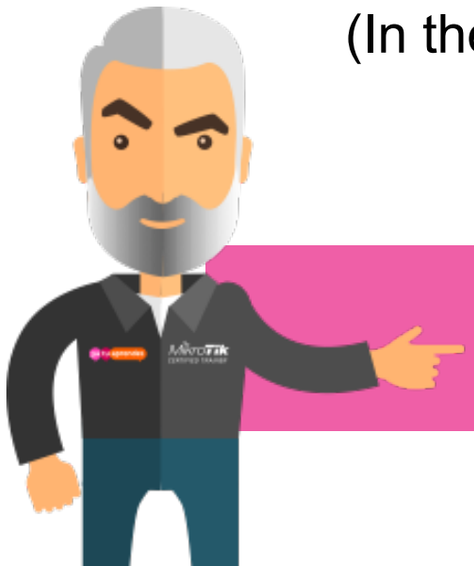
The customer calls the "call center" to complain (kindly) that he does not have access to the internet.

After a series of tests (and a lot of patience), we determine that the incident is caused because the client's router has been restored to factory values.

We check the weather part of the day and see that it is a very sunny day, so it could not be a thunderstorm.

We were perplexed to see that it was the client who has reborn the router.

It's Friday afternoon, and… we'll have to go to the customer's house to configure the device.
(In the distance you hear the client say that he has not touched anything, nothing)



This story is fiction and only serves to expose the problem

I wish…

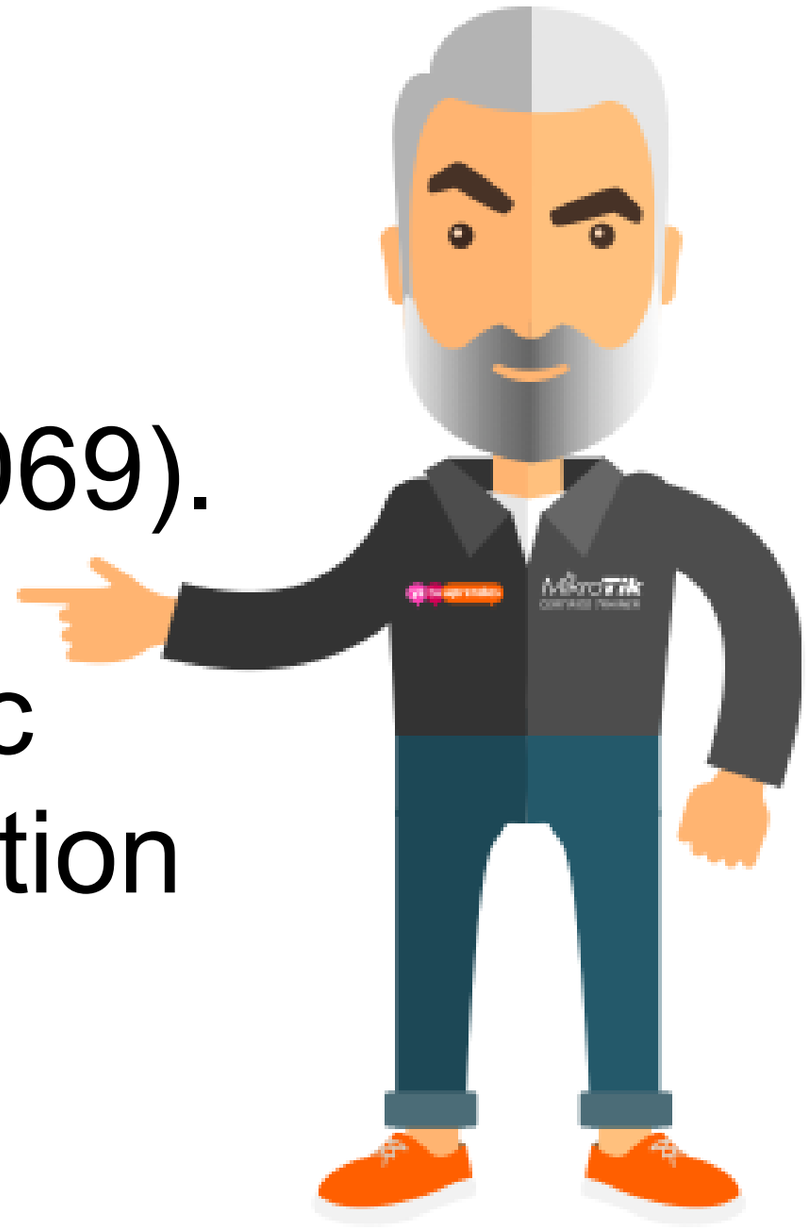
- Send the configuration to my devices.
- To know if they are configured or not.

What I need is…

- **C**PE **W**an **M**anagement **P**rotocol
- **A**uto **C**onfiguration **S**erver

CWMP

- TR-069 (Technical Report 069).
- Developed for the automatic management and configuration of the devices.



CWMP

- Based on SOAP / HTTP.
- Secure self-configuration.
- Functions for management control.
- Integrated environment.

CWMP

- A Session is a message exchange.
- The CPE starts a Session in response to different events.
- Only CPE starts a Session

CWMP

- The ACS may request a Session.
- Execute RPC’s
- CPE always stars a sesión with an “Inform” RPC.

CWMP

- Parameters: key = value.
- The set of supported parameters is called a data model.

CWMP

- There are 3 predefined data models

  TR-098, TR-181:1 y TR181:2.

- The manufacturer must base on the predefined data models.

And what about security?

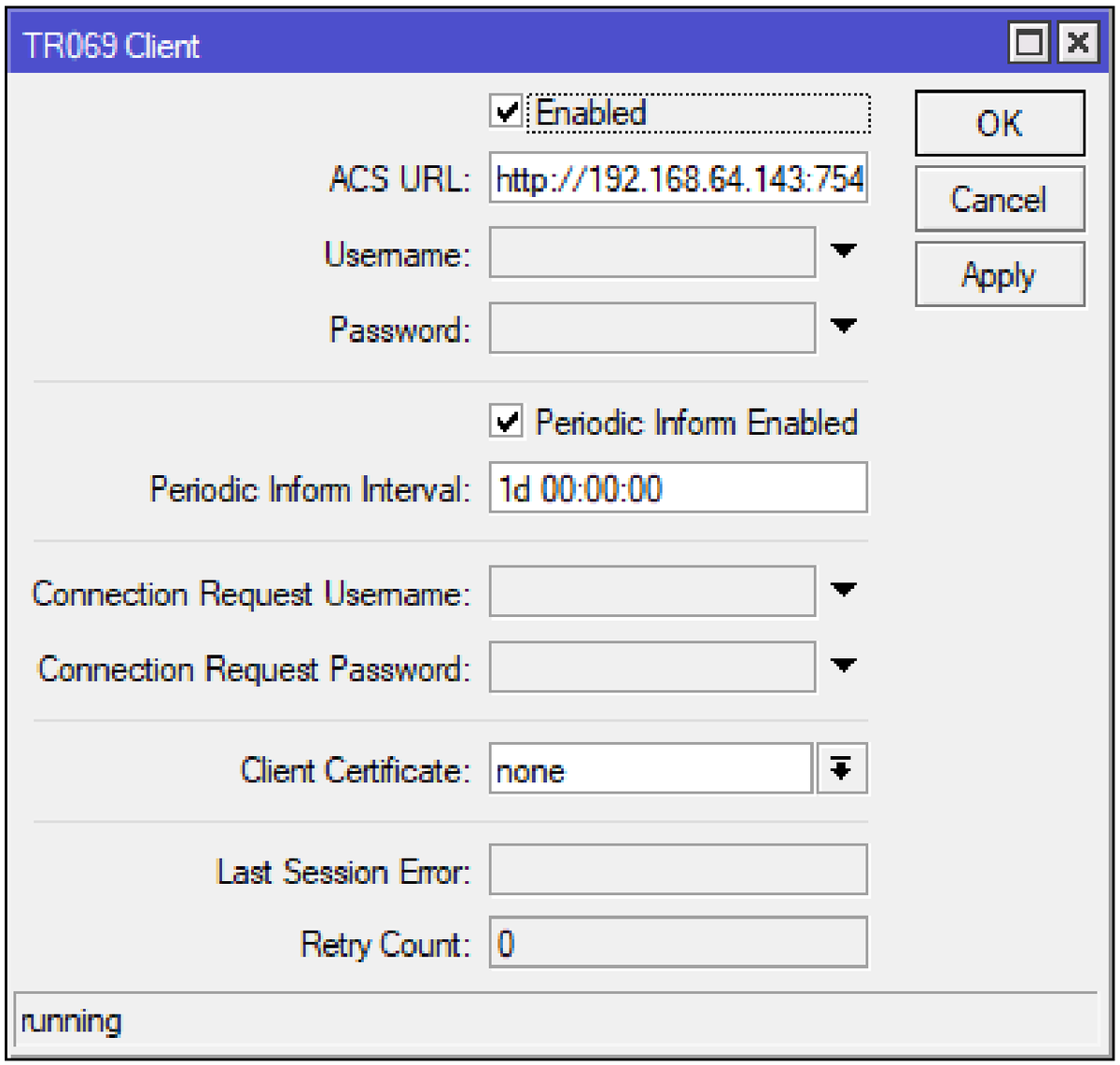
- Authentication.
- User password.
- SSL.
- Certificates for clients.

Mikrotik TR-069

- Supports HTTP and HTTPS.
- HTTP authentication.
- Inform.
- Client certificates.
- Data Mode based on the **TR-181 Issue2 Amendment 11.**

Mikrotik TR-069

Minimum configuration:



Mikrotik TR-069

- Unfortunately, after a reset, the configuration of the TR-069 is lost.

(and with that, all of our dreams)

- We have to use netinstall.

We have an example script in the wiki:
http://wiki.mikrotik.com/wiki/Tr069-best-practices

ACS

Mikrotik is compatible with different ACS:

- ✓ AVSystems.
- ✓ Axiros.
- ✓ Friendly Tech.
- ✓ GenieACS (open source).

GenieACS

- Fast and Light Autoconfiguration System.
- Open source.
- TR-069 solution for remote management and provisioning.
- Built on Node.js and MongoDB.

Installation

We will need:

- Node.js: from version 6.x to 8.x

  (Version 8.x is recommended)

- MongoDB: from version 2.6 to 3.4

- The compilation tools (build tools) and the libxml2 library

  (We will install them from the apt-get)

To install Node.js in a Debian / Ubuntu distribution we will need to download the relevant script.

In this case https://deb.nodesource.org/distributions/deb/setup_8.x

“Once we have made the broth, we go to the soup”

(jedi proverb)

Installation

- We will perform the installation through the npm:

```
npm install -g genieacs
```

If the previous installation process gave us a problem, we can clean the directory and install it from the git repository, for this we will execute the commands:

```
git clone https://github.com/zaidka/genieacs.git
cd genieacs
git checkout $(git tag -l v1.1.* --sort=-v:refname | head -n 1)
npm install
npm run compile
```

Installation

- Once installed, we have 3 executables:

- **Genieacs-cwmp**

  It is the service by which the CPE is communicated. By default it listens on port 7547 TCP.

- **Genieacs-nbi**

  This service exports a REST API for the GUI frontend. By default it listens on port 7557 TCP.

- **Genieacs-fs**

  This service is the file server from which our CPE will download the firmware images. By default listens on port 7567 TCP.

To modify the default values and / or configure the file server name, we will edit the file: config/config.json

Installation

- So far we have installed the genieACS and we have configured our CPE mikrotik that has connected to the ACS and has done little things, but we can not see it in a nice way. We only have the screen output of the genieACS execution and the log of our mikrotik.

Sep/25/2017 21:14:07	memory	tr069, warning	tr069 running in non-secure mode (HTTP)
Sep/25/2017 21:14:07	memory	tr069, debug	starting session, events: [0 BOOTSTRAP, 1 BOOT,]
Sep/25/2017 21:14:07	memory	system, info	tr069-client settings changed by admin
Sep/25/2017 21:14:07	memory	tr069, debug	send: Inform
Sep/25/2017 21:14:07	memory	tr069, debug	rcvd: InformResponse
Sep/25/2017 21:14:07	memory	tr069, debug	send: ""
Sep/25/2017 21:14:07	memory	tr069, debug	session finished ok
Sep/25/2017 21:14:07	memory	tr069, debug	scheduled next Periodic Inform after 86400 seconds

Installation

```
listening; pid=1819 address="0.0.0.0" p
ort=7547
2018-10-05T03:28:29.270Z [INFO] 192.168
.88.131 E48D8C-RB951Ui%2D2HnD-4AC704613
lE4: Inform; cpeRequestId=undefined inf
ormEvent="0 BOOTSTRAP,1 BOOT" informRet
ryCount=0
2018-10-05T03:28:29.454Z [INFO] 192.168
.88.131 E48D8C-RB951Ui%2D2HnD-4AC704613
lE4: New device registered
```

```
on="1.1.2" dependencies="node@8.11.1,lat
er@1.2.0,libxmljs@0.18.7,mongodb@2.2.34,
seedrandom@2.4.3,redis@undefined" config
="DEBUG=true"
2018-10-05T02:33:35.102Z [INFO] Worker l
istening; pid=1833 address="0.0.0.0" por
t=7557
2018-10-05T02:33:35.123Z [INFO] Worker l
istening; pid=1817 address="0.0.0.0" por
t=7557
```

```
ion="1.1.2" dependencies="node@8.11.1,l
ater@1.2.0,libxmljs@0.18.7,mongodb@2.2.
34,seedrandom@2.4.3,redis@undefined" co
nfig="DEBUG=true"
2018-10-05T02:33:35.021Z [INFO] Worker
listening; pid=1818 address="0.0.0.0" p
ort=7567
2018-10-05T02:33:35.053Z [INFO] Worker
listening; pid=1808 address="0.0.0.0" p
ort=7567
```

```
  Rendered devices/_commands.html.erb (1
1.8ms)
  Rendered devices/show.html.erb within
layouts/application (97.0ms)
  Rendered layouts/_menu.html.erb (4.3ms
)
Completed 200 OK in 462ms (Views: 285.3m
s | ActiveRecord: 0.0ms)
```

```
[genieacs]0:GenieACS*                    "ubuntu" 20:38 04-Oct-18
```

Example of running genieACS over tmux

‘You made your bed,
now you have to lie in it’

We still need to install the web frontend

Installation

To install the frontend (genieacs-gui) you need:

- Ruby on Rails

  (A version equal to or greater than 2.2.2 is recommended)

- Bundler

Installation

- We will clone the git repository:

```
git clone https://github.com/zaidka/genieacs-gui.git
```

Installation

Once cloned we will execute:

```
cd genieacs-gui

cp config/graphs-sample.json.erb config/graphs.json.erb

cp config/index_parameters-sample.yml config/index_parameters.yml

cp config/summary_parameters-sample.yml config/summary_parameters.yml

cp config/parameters_edit-sample.yml config/parameters_edit.yml

cp config/parameter_renderers-sample.yml config/parameter_renderers.yml

cp config/roles-sample.yml config/roles.yml

cp config/users-sample.yml config/users.yml

bundle
```

And… It’s done


No es seguro | 192.168.88.130:3000/log_in
genieacs
Home
Log in
Username
Password
Log in
No es seguro | 192.168.88.130:3000/devices
genieacs
Home
Devices
Faults
Presets
Objects
Listing devices
Filters
+
Filter
Clear
Showing 2 devices
Serial number
Product class
Software version
MAC
IP
WLAN SSID
Last inform
6 months ago
36 minutes ago
What happens? After all the “hard work”, nothing comes out. That sucks!

We have to manipulate the genieacs.
We see it better in a demo, right?

Solution

This happens because the keys waiting for the genieacs-gui are not those sent by Mikrotik or any other device based on TR-181.

To correct it, we have to modify the files:

- genieacs-gui/config/index_parameters.yml
- genieacs-gui/config/summary_parameters.yml

Solution

- Being based on TR-098 the genieacs-gui expects the information to be in:

  `InternetGatewayDevice.DeviceInfo.*`

- But our mikrotik (based on TR-181) gives us the information in:

  `Device.DeviceInfo.*`

- But that is not all. Since version 1.1.x, genieacs exports its own object with the "essential" information of the client. This object contains the information:

  ✓ **SerialNumber.** Serial number of the device. In mikrotik it is the interface Mac.

  ✓ **ProductClass.** Family of the product. In mikrotik it is the routerboard model.

  ✓ **OUI**. Manufacturer identifier. It is a unique identifier for each manufacturer.

  ✓ **Manufacturer**. Name of the manufacturer.

Solution

```
Serial number: _deviceId._SerialNumber
Product class: _deviceId._ProductClass
Software version: Device.DeviceInfo.SoftwareVersion
MAC: InternetGatewayDevice.WANDevice.1.WANConnectionDevice.1.WANIPConnection.1.MACAddress
IP: InternetGatewayDevice.WANDevice.1.WANConnectionDevice.1.WANIPConnection.1.ExternalIPAddress
WLAN SSID: Device.WiFi.SSID.1.SSID
```

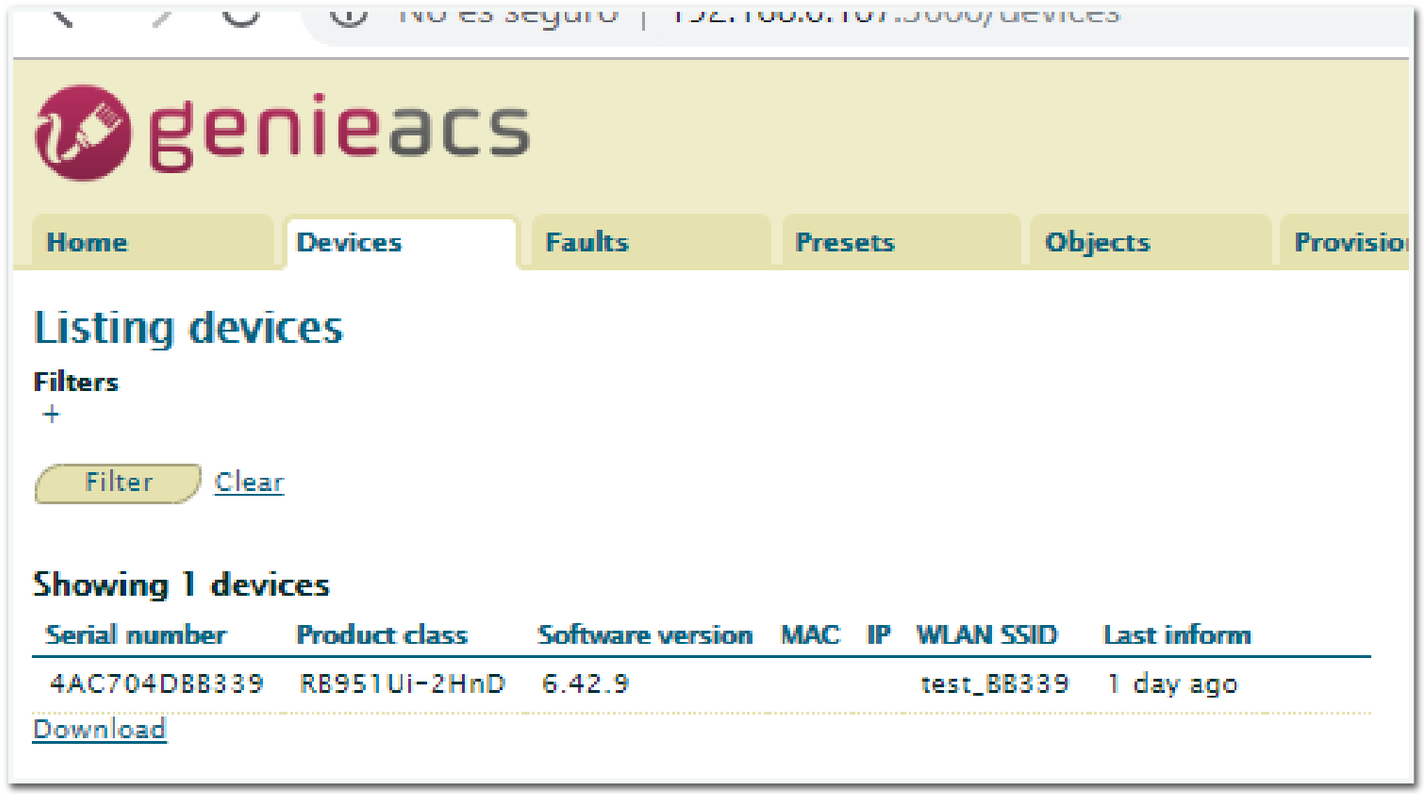
index_parameters.yml

```
Serial number: _deviceId._SerialNumber
Product class: _deviceId._ProductClass
OUI: _deviceId._OUI
Manufacturer: _deviceId._Manufacturer
Hardware version: Device.DeviceInfo.HardwareVersion
Software version: Device.DeviceInfo.SoftwareVersion
MAC: InternetGatewayDevice.WANDevice.1.WANConnectionDevice.1.WANIPConnection.1.MACAddress
IP: InternetGatewayDevice.WANDevice.1.WANConnectionDevice.1.WANIPConnection.1.ExternalIPAddress
WLAN SSID: Device.WiFi.SSID.1.SSID
WLAN passphrase: InternetGatewayDevice.LANDevice.1.WLANConfiguration.1.KeyPassphrase
Hosts:
    _object: Device.Hosts.Host
    Host name: HostName
    IP: IPAddress
    MAC: MACAddress
```

summary_parameters.yml

Solution

Now, it works!



But, Our mikrotik still does not self-configure

Autoprovision

- The genieACS is a software capable of "talking" with any device. Although it is oriented to the devices that use the TR-098, it communicates perfectly with our mikrotik.

- We have already made a modification in the frontend configuration so that it shows us the information of our mikrotik Now it's time for the genieACS.

Presets

- The presets are, as the name suggests, configurations that we will apply to the device.

- That is, in a preset we can assign, for example, the value of SSID, the frequency and the password of the wifi that our device will have.

Presets

- The presets are made up of two parts:
- **Preconditions**: in this part we can control when the preset is executed. They can be, for example:
  - OUI
  - Tag
  - Serial Number
- **Configurations**: in this part we will apply the configurations. We can, for example:
  - Assign values to a parameter
  - Add or remove Tags
  - Refresh a parameter
  - Execute a provision

Presets

Additionally we can specify:

- **A planning**: we can define the periodicity with which the preset will be executed.

- **The events**: pwe can define which CPE events triggered the preset. We can specify one or a series of them. Events can be:

✓ 0 BOOT

✓ 1 BOOTSTRAP

✓ 7 TRANSFER COMPLETE

✓ M DONWLOAD

✓ 2 PERIODIC

Presets

- Los Tags, no son un parámetro del CPE, es un parámetro interno que nos ofrece el genieACS. Mediante el uso de Tags podemos definir un flujo de preset o identificar el estado del CPE.

- A device can contain more than one Tag.

- In our demonstration we have defined the tags:

  - **PENDING**: this tag is associated with the CPEs that have not yet started the process.

  - **UPGRADING**: this tag is associated with the CPE so that it is checked if it needs a firmware update.

  - **UPGRADED**: this tag is associated to the CPE when it is already updated.

  - **PROVISIONING**: this tag is associated with the CPE when it starts the process of provisioning the configuration. Additionally, in this state the following tags are used: INIT, WIFI and OTHER.

  - **PROVISIONED**: this tag is associated with the CPE when it is fully configured.

Presets

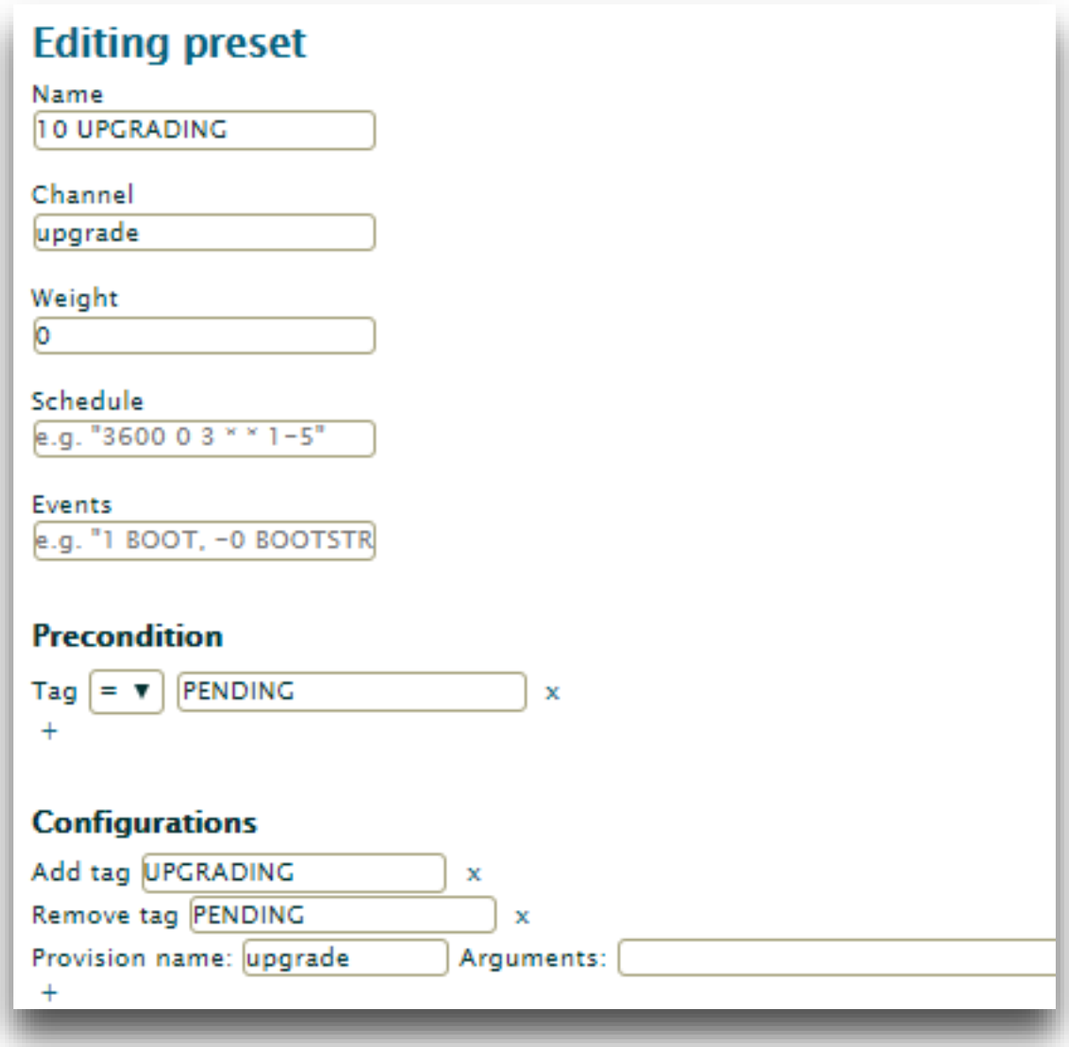
- From the previous Tags we define the presets:

Name	Channel	Weight	Events
00 NEW DEVICE		0	1 BOOT, -0 BOOTSTRAP
19 UPGRADED	upgrade	0	7 TRANSFER COMPLETE, M Download
20 PROVISIONING	provision	0	2 PERIODIC
21 PROVSIONING - WIFI	provision	0	
28 PROVISIONING - OTHER	provision	0	
10 UPGRADING	upgrade	0	
29 PROVISIONED	provision	0	

The weight serves so that in case two presets contain the same element to be configured, the configuration of the one with the highest weight prevails.

Presets

- The preset of 10 UPGRADING would be:



Provision

- In the previous preset you have seen that in Configuration there is the "Provision" option.

- What is a provision?

A provision is a program that we execute to perform configurations based on complex conditions.

Provision

- The Provisions are introduced in version 1.1
- They are javascript programs that run in a sandbox.
- It offers a series of functions defined by genieACS.
- We can pass parameters that will be available through the array args.

Provision

We have defined two provisions:

- **Upgrade**: the version of the routerOS is checked and if it is less than 4.20.9. It updates the mikrotik.

- **Wifi**: set the ssid to the string test_ + the last five digits of the MAC and configure the ap_bridge mode.

Provision

- The upgrade script would be:

```
let version=declare("Device.DeviceInfo.SoftwareVersion",{value:1}).value[0];
log('device version: ' + version);
if (version=="6.42.9"){
  log('No upgrade needed');
  declare("Tags.UPGRADED",null,{value:true});
  declare("Tags.UPGRADING",null,{value:false});
} else {
  log('upgrading firmware');
  declare("Downloads.[FileType:1 Firmware Upgrade Image]",
        {path: 1}, {path: 1});
  declare("Downloads.[FileType:1 Firmware Upgrade Image].FileName",
        {value: 1}, {value: "upgrade-mipsbe-6.42.9.xml"});
  declare("Downloads.[FileType:1 Firmware Upgrade Image].Download",
        {value: 1}, {value: Date.now()});
}
```

Files

- In the script we have defined a Download specifying the file type to "1 Firmware Upgrade Image".

- The file that our mikrotik downloads is an xml file.

- Mikrotik specifies that to update the version of the routerOS from the ACS it is necessary to send an xml file that contains the links to the different npk that we are going to update.

Files

Contents of the xml file:

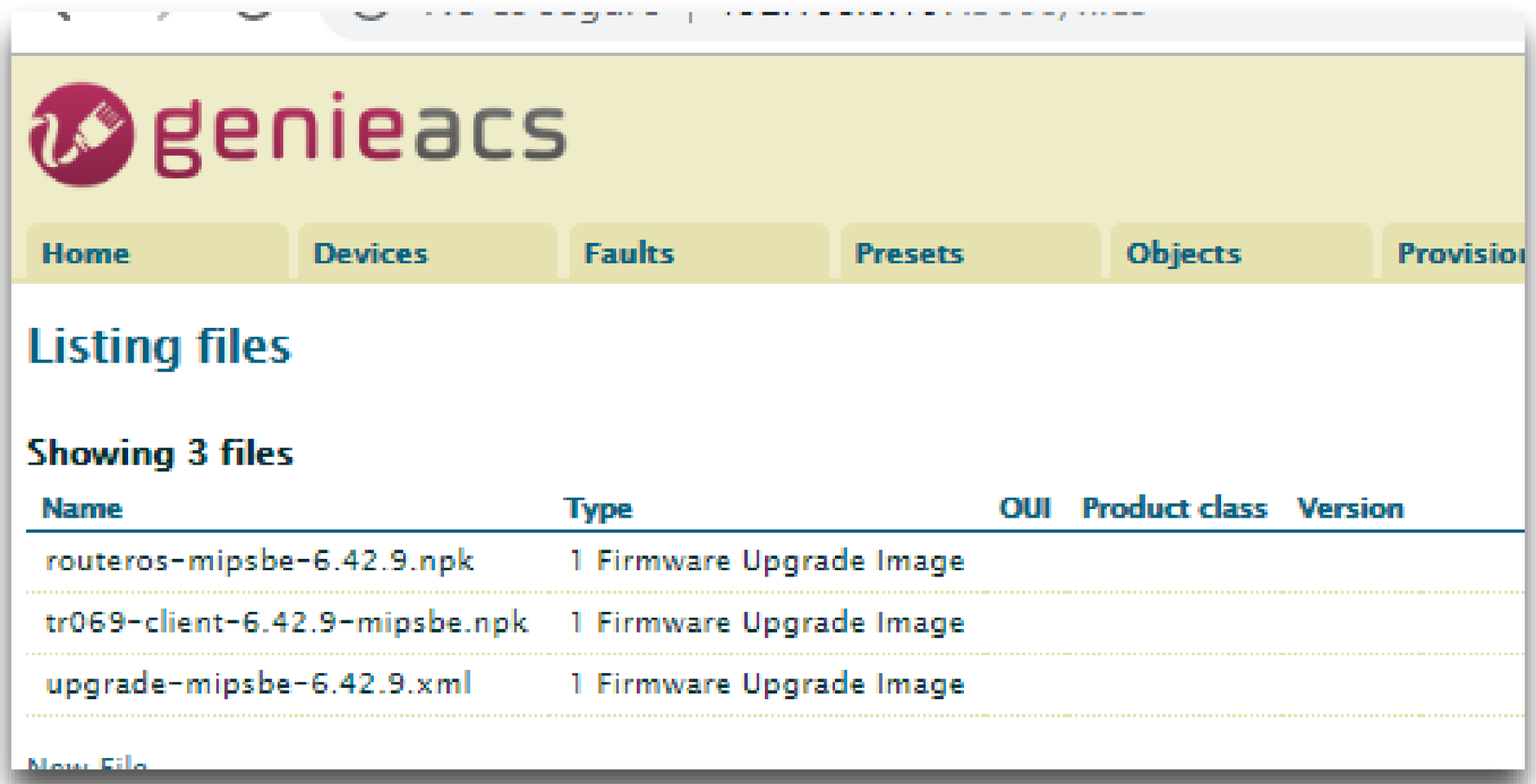
```
<upgrade version="1" type="links">
    <config/>
    <links>
        <link>
            <url>http://192.168.0.107:7567/routeros-mipsbe-6.42.9.npk</url>
        </link>
        <link>
            <url>http://192.168.0.107:7567/tr069-client-6.42.9-mipsbe.npk</url>
        </link>
    </links>
</upgrade>
```

Note that we have also added the tr069 package. Since we do not, after updating the version, the service will no longer be available and our mikrotik will stop communicating with the ACS.

If we serve other additional packages, we must also add them.

Files

So, in our system we will have the following files:



Files

I have done everything you have taught me and it doesn’t upgrade.

It gives me an error of “unresolved".

What I do?

Files

Oh!

- You have not configured the file server name correctly in the config.json configuration file of genieacs.
- The entry of the file in question is:

  FS_HOSTNAME

- In it we will specify the ip or the name of the server where the genieACS is running, more specifically, the genieacs-fs process.

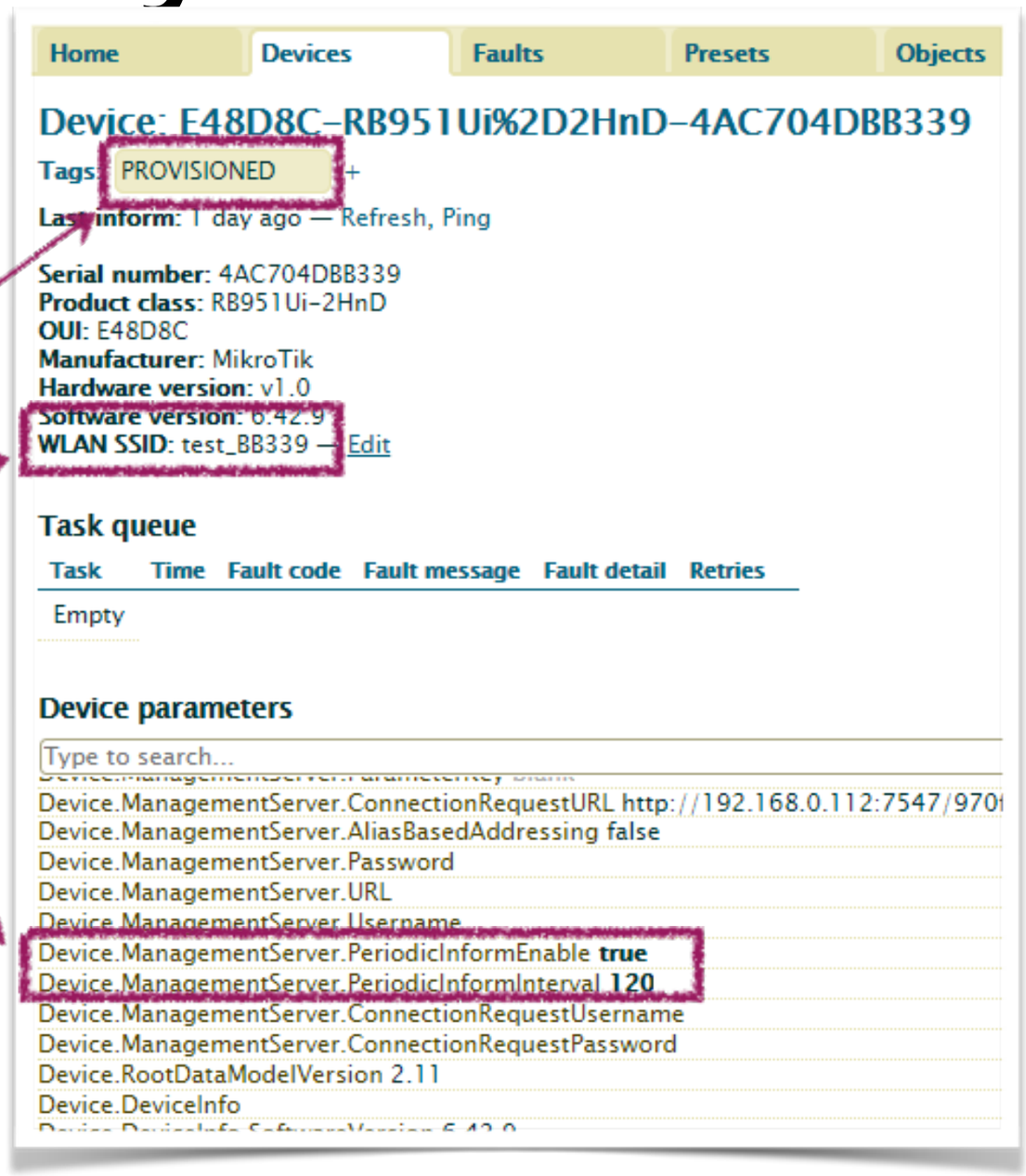
This is to download the xml file. The links inside the xml file can refer to any server (even the mikrotik downloads ;)

Finally

Once the genieACS is configured, we reboot our CPE and ….

Eureka!!

It Works!



If you want to obtain a copy of the virtual machine that has been used for the presentation,

Please, send an email to:

j.castellet@yatuaprendes.com

And we will send you a download link.

Thank you for your attention.

More Information

- https://www.broadband-forum.org/technical/download/TR-069_Amendment-5.pdf
- https://wiki.mikrotik.com/wiki/Manual:TR069-client
- https://wiki.mikrotik.com/wiki/Tr069-best-practices
- https://wiki.mikrotik.com/tr069ref/current.html
- Hannes Willemse presentation at ZA17
- https://mum.mikrotik.com/presentations/ZA17/presentation_4990_1512109593.pdf
- https://genieacs.com