

CPE WAN Management Protocol  
&  
Auto Configuration Server

# Who I am

Mi name is Jorge Castellet  
I'm Mikrotik certified Trainer.



More than 15 years of experience in networking.

...



# Hello, I do not have internet

- The customer communicates with the call center to complain that nothing is working.
- After some checks ... It is determined that the CPE has been restored to factory defaults.

(Of course, he has not touched anything)

You have to go to the client's house to reconfigure the CPE



# Hello, I do not have internet

Everything would be wonderful if my devices were configured by themselves only by plugging them.

If this were possible, I would have more free time.



# Wish list ...

- I need a system with which my devices can communicate.
- That is able to identify the devices and send the configuration to each of them.
- I can know if it is configured or not.



# Wish granted !

What you need is....

- ➔ CPE WAN Management Protocol (CWMP)
- ➔ Auto Configuration Server (ACS)



# CWMP

- TR-069 (Technical Report 069) is a technical specification that defines the application layer for remote management of CPE connected to an IP network.
- The TR-069 standard was developed for automatic configuration and management of devices by Auto Configuration Servers (ACS)



# CWMP

- It's a SOAP/HTTP based protocol.
- Includes:
  - ✓ Safe autoconfiguration
  - ✓ Control of management functions
- Integrated framework





# CWMP

- Session is all the messages exchanged during a communication.
- Only CPE can initiate a Session.
- CPE starts a Session on different events.
  - For example: first boot, reboot, periodic interval. Transfer complete, diagnostic complete, etc.
- ACS can request to the CPE to start a Session.
- During a session each part call RPC's to be executed on the other side.
- CPE always start a Session with an Inform RPC, containing connection reason, device info and some extra parameters values.



# CWMP

- Parameters are simple name=value pairs
- Each vendor can decide which parameters to support in its devices.
- Data Model is the combination of all supported parameters.
- There are three root Data Models: TR-098, TR-181:1, TR-181:2
- Vendor must base their supported parameters on root Data Models



# What about security ...

TR-069 provides several mechanisms:

- Authentication:
  - ✓ Username/password
  - ✓ SSL client certificate
- Communication:
  - ✓ SSL
- And you can also implement security on device (like firewall address list)



# Building my own CWMP & ACS

I need to configure:

- CPE:
  - ✓ To survive the customer/user
  - ✓ To schedule events
- Server:
  - ✓ Receive device information
  - ✓ Display data
  - ✓ Device adoption



# CPE

- We need our device to execute a script when it restarts .... easy .. schedule event
- But .. If the client performs a device reset, then all the configuration is deleted, including our scheduled event. How do I solve it?
- I need netinstall to overwrite the default startup script.



# CPE

- We overwrite the default startup script to:
- Perform the initial contact with the server, indicating that it is a new device.
- install our script, and schedule it.
- What does our script do?
  - ✓ Send keepalive
  - ✓ Receive config file and install it
  - ✓ Send device status



# Server

- I need to build a system with which my device can communicate.
- In order to achieve it, I will need:
  - ➔Apache
  - ➔PHP
  - ➔Mysql



# Server

Communicating with CPE:

- ✓ knock.php
- ✓ keepalive.php
- ✓ data.php
- ✓ rpc.php





# Knock.php

- It's called from CPE after an reset or on first boot.
- Insert the new device in pending devices table.
- It's called periodically until CPE is adopted.
- Available responses:
  - ➔Wait.  
Waiting for human.
  - ➔Rpc.  
Sends a command to CPE.
  - ➔Nothing.  
No business.



# Keepalive.php

- This script is called periodically after an adoption.
- It's called also after a reboot.
- Only send commands to CPE.
- Available response:
  - ➔No  
Unknown device. Wrong place
  - ➔Knock  
Must perform a knock to finish adoption
  - ➔Rpc  
Send a command to CPE
  - ➔Nothing  
No business



# Data.php

- This script is called by the CPE to receive the file with the commands it must execute.
- StartAdoption command has a different approach.
- The file that it returns contains CLI commands that must be executed in the mikrotik.



# Rpc.php

- This script is called by the CPE to indicate that it has completed the command that was requested.
- Update the status of the command with the result codes:

✓ OK

The script has been executed correctly.

✓ ERROR

There have been errors in the execution.

Maybe Syntax error ??



# Server

We also need a frontend, in order to:

- ✓ Accept device (adopt)
- ✓ View device status
- ✓ Change device configuration



# Discovered.php

- Shows a list of devices that are knocking on the door to enter.

Pending devices [Managed devices](#)

Device ID	RouterBoard	RouterOS	Status	Last Seen	Cause
00:01:02:03:04:06	x86	6.37.4 (bugfix)	Adopting	2018-03-02 03:36:13	
00:0C:29:7B:F0:1B	x86	6.37.4 (bugfix)	Pending	2018-03-16 05:16:29	first boot <a href="#">Adopt</a>
FA:BA:DA:FA:BA:00	no	6.37.4	Adopting	2018-03-01 01:53:58	

Name:

Pending devices [Managed devices](#)

adoption of device 00:0C:29:7B:F0:1B initiated

Device ID	RouterBoard	RouterOS	Status	Last Seen	Cause
00:01:02:03:04:06	x86	6.37.4 (bugfix)	Adopting	2018-03-02 03:36:13	
00:0C:29:7B:F0:1B	x86	6.37.4 (bugfix)	Adopting	2018-03-16 05:16:29	first boot
FA:BA:DA:FA:BA:00	no	6.37.4	Adopting	2018-03-01 01:53:58	



# Managed.php

- Shows a list of the adopted devices.
- It also allows us to send commands to the CPE.

[Pending devices](#) **Managed devices**

Device ID	RouterBoard	RouterOS	Status	Last Seen	Cause
00:01:02:03:04:05	x86	6.37.3	Ok	2018-03-15 13:39:20	<a href="#">Send Command</a>
00:01:02:03:04:07	x86	6.37.4 (bugfix)	Ok	2018-03-02 03:59:51	<a href="#">Send Command</a>
00:01:02:03:04:08	x86	6.37.4 (bugfix)	Ok	2018-03-02 04:03:52	<a href="#">Send Command</a>
00:0C:29:7B:F0:1B	x86	6.37.4 (bugfix)	Ok	2018-03-15 17:13:38	periodic <a href="#">Send Command</a>
FA:BA:DA:FA:BA:DA	x86	6.37.4	Ok	2018-03-01 01:16:23	<a href="#">Send Command</a>



# CPE: our beloved mikrotik

- We need a script to perform queries to the server.
- As said before, we need to customize the default mikrotik script to survive the customer/child/wife/husband.
- We will use the same script for the initial contact.





# CPE

The script we will use:

- **Knock**

Performs initial contact requesting server for adoption.

- **Keepalive**

Once it has been adopted, this script executes the commands from the server.

It also keeps a heartbeat with the server.

- **Run-keepalive**

check if we have previously downloaded a new version of the keepalive script

Of course, it also runs the keepalive script



# CPE

I have used other people's functions:

- **UrlEncode**

Skot

<https://forum.mikrotik.com/viewtopic.php?t=84705>

- **GetBetween**

CuriousKiwi

<https://forum.mikrotik.com/viewtopic.php?t=96172>



# Knock

```
:local url "http://$username:$password@$acsserver/acs/api/knock.php?mac=$macaddr&routerboard=$rboard&route"
:put $url
do {
/tool fetch url=$url mode=http dst-path="/knock.txt"
:local data [/file get knock.txt contents]
:local status [$getBetween inputString=$data betweenStart="<action>" betweenEnd="</action>"]
if ( $status != "Rpc") do={
:put "nothing to do. I'll check later"
} else={
```

```
if ($cmd="StartAdoption") do={
:local identity [$getBetween inputString=$params betweenStart="<identity>" betweenEnd="</identity>"]
/system identity set name=$identity
:set url "http://$username:$password@$acsserver/acs/api/data.php?mac=$macaddr&id=$id&name=keepalive"
:put $url
/tool fetch url=$url mode=http dst-path="/keepalive.rsc"
:local runstatus
:do {
[:parse [/file get keepalive.rsc contents]]
:set runstatus "Ok"
/system scheduler disable schedule1
} on-error={
:set runstatus "Error"
}
}
```



# Keepalive

```
if ($cmd="RunScript") do={
:set url "http://$username:$password@$acsserver/acs/api/data.php\?mac=$macaddr&id=$id&name=keepalive"
:put $url
/tool fetch url=$url mode=http dst-path="/mywork.rsc"
:local runstatus
:do {
  [:parse [/file get mywork.rsc contents]]
  :set runstatus "Ok"
} on-error={
  :set runstatus "Error"
}
:set url "http://$username:$password@$acsserver/acs/api/rpc.php\?mac=$macaddr&id=$id&status=$runstatus"
:put $url
/tool fetch url=$url mode=http dst-path="/rpc.txt"
}
```



# OOOOPS

After all the work we have done, ... Mikrotik has implemented the TR-069 client.

So let's use it.



# Mikrotik TR-069

- RouterOS implementations supports HTTP or HTTPS on URL of ACS
- Username / password HTTP authentication to “login” into ACS
- Periodic inform
- Client certificate for extra security
- RouterOS Data Model is based on **TR-181 Issue 2 Amendment 11**



# Mikrotik TR-069

The screenshot shows the 'TR069 Client' configuration window. It features several sections of settings:

- Enabled:** A checkbox that is checked, with the text 'Enabled' in a dotted box next to it.
- ACS URL:** A text field containing 'http://192.168.64.143:754'.
- Username:** An empty text field with a dropdown arrow on the right.
- Password:** An empty text field with a dropdown arrow on the right.
- Periodic Inform Enabled:** A checkbox that is checked.
- Periodic Inform Interval:** A text field containing '1d 00:00:00'.
- Connection Request Username:** An empty text field with a dropdown arrow on the right.
- Connection Request Password:** An empty text field with a dropdown arrow on the right.
- Client Certificate:** A dropdown menu currently set to 'none'.
- Last Session Error:** An empty text field.
- Retry Count:** A text field containing '0'.

On the right side of the window, there are three buttons: 'OK', 'Cancel', and 'Apply'. At the bottom left of the window, the status 'running' is displayed.



# Mikrotik TR-069

- Unfortunately, after a reset, the tr-069 configuration is also lost
- We still need to use netinstall to survive the customer. ;)
- Our default script, only need to:
  - 1) Install the ACS certificate (in case of HTTPS communication)
  - 2) Assign an IP address to a network interface
  - 3) Configure the client TR-069
- Sample script in:  
<https://wiki.mikrotik.com/wiki/Tr069-best-practices>





# ACS

Mikrotik is compatible with several ACS, for example:

- ✓ AVSystem
- ✓ Axiros
- ✓ Friendly Tech
- ✓ GenieACS (open source)



# Genie ACS

- Fast, lightweight TR-069 ACS
- GenieACS is an open source TR-069 remote management solution with advanced device provisioning capabilities.
- Is built with Node.js and uses MongoDB as its database



# Installation

- We need:
  - ✓ Node.js: 6.x and 8.x (8.x recommended)
  - ✓ MongoDB: 2.6 through 3.4
- To install Node.js on a Debian/Ubuntu distro you need to download the installation script:
  - ➔ `setup_8.x`
- From:
  - ➔ <https://deb.nodesource.com>



# Configuration

- By default GenieACS uses Data model TR-098.
- that means that our devices do not appear correctly in genieacs

Showing 1 devices

Serial number	Product class	Software version	MAC	IP	WLAN SSID	Last inform
						17 minutes ago

[Download](#)



# Configuration

To solve this, we can modify the files

- ✓ genieacs-gui/config/index\_parameters.yml
- ✓ summary\_parameters.yml

Changing:

```
Serial number: InternetGatewayDevice.DeviceInfo.SerialNumber
```

```
Product class: InternetGatewayDevice.DeviceInfo.ProductClass
```

```
Software version: InternetGatewayDevice.DeviceInfo.SoftwareVersion
```

To:

```
MK Serial number: DeviceInfo.SerialNumber
```

```
MK Product class: DeviceInfo.ProductClass
```

```
MK Software version: DeviceInfo.SoftwareVersion
```



# Configuration

- The best way would be to use a function that returns the undesired values whether it is a mikrotik or another device that follows the TR-098.
- We must use Virtual parameters
  - Are user-defined parameters whose values are generated using custom Javascript code.



# More information

- [https://www.broadband-forum.org/technical/download/TR-069\\_Amendment-5.pdf](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://mum.mikrotik.com/presentations/ZA17/presentation_4990_1512109593.pdf)
- <https://genieacs.com>



# More information

If you want the source code, or more information, you can send me an email to:

[j.castellet@yatuaprendes.com](mailto:j.castellet@yatuaprendes.com)





