



*Mikro**Tik***
CERTIFIED TRAINER

web portal for
basic router
config using tr069
and genieacs



WE INSPIRE KNOWLEDGE

Lisbon
20 de Setembro

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



Fazemos cursos e certificamos
em Mikrotik com nossos
parceiros da **Truenet**.



Basically we need



- **CPE Wan Management Protocol**
- **Auto Configuration Server**

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 starts a Session with an "Inform" RPC.

Mikrotik TR-069



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

I'm sad



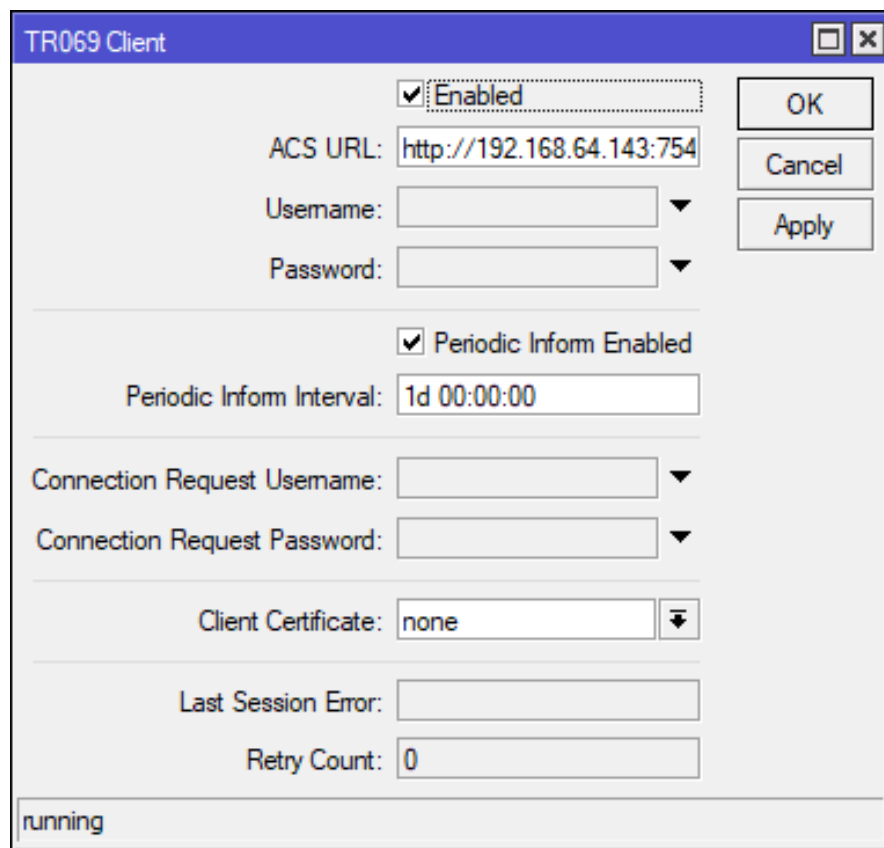
After reading the TR-181 and check the mikrotik wiki....

➤ **NOT ALL CONFIGURATION
OPTIONS ARE AVAILABLE
THROUGH TR-069.**

- We need to use scripting for some options, such as channel width.

Mikrotik TR-069

Minimum configuration:



The screenshot shows the 'TR069 Client' configuration window. The 'Enabled' checkbox is checked. The 'ACS URL' is set to 'http://192.168.64.143:754'. The 'Username' and 'Password' fields are empty. The 'Periodic Inform Enabled' checkbox is checked, and the 'Periodic Inform Interval' is set to '1d 00:00:00'. The 'Connection Request Username' and 'Connection Request Password' fields are empty. The 'Client Certificate' is set to 'none'. The 'Last Session Error' field is empty. The 'Retry Count' is set to '0'. The status bar at the bottom indicates 'running'.

TR069 Client		OK
☒ Enabled		Cancel
ACS URL:	http://192.168.64.143:754	Apply
Username:		
Password:		
☒ Periodic Inform Enabled		
Periodic Inform Interval:	1d 00:00:00	
Connection Request Username:		
Connection Request Password:		
Client Certificate:	none	
Last Session Error:		
Retry Count:	0	
running		

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>

GenieACS



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

API



- GenieACS provides a powerfull API.
- The API is supported by geniacs-nbi.
- I use this API for:
 - ✓ Check the client status.
 - ✓ Send commands to a client.

Customer Portal

- I want customers to be able to change the basic parameters of their WiFi.
- The customer only can access from their place.
- The client only knows the IP of his router: 192.168.88.1.
- As this is very difficult to remember .. I'll use the name "config.me"

➤ <http://config.me>

Previous work.

- To make things flow smoothly, we need to configure our mikoritik.
- I need to:
 - ✓ Add a dns static entry with config.me pointing to 192.168.88.1
 - ✓ A firewall dstnat rule redirecting the <http://192.168.88.1> to our web server <http://172.16.100.33>.

Customer portal

- Client Authentication Page
 - Simple authentication.
 - For sure is not secure 100%.
- Router Status Page.
- Wifi Parameters Edit Page.
 - ✓ Enable/disable interface.
 - ✓ Channel.
 - ✓ Protocol.
 - ✓ SSID.
 - ✓ Password.

Customer Portal

- I made it in Express.js
 - I'm not a senior programmer in Nodejs, so keep it in mind.

Customer Portal

- GenieACS stores in MongoDB a copy of the parameters send by the CPE in the Inform Message.
- We query the genieacs database through the api.
- Genieacs is the only one who query the CPE through cwmp.
- Then ... we need to send a Refresh RPC to genieacs in order to obtain the latest parameters and to know if the CPE is still alive.

Data

- We expect the data in genieacs-gui format

```
Device.IP
```

```
Device.IP.Interface
```

```
Device.IP.Interface.1
```

```
Device.IP.Interface.1.IPv4Address
```

```
Device.IP.Interface.1.IPv4Address.3
```

```
Device.IP.Interface.1.IPv4Address.3.Enable true
```

```
Device.IP.Interface.1.IPv4Address.3.Status Enabled
```

```
Device.IP.Interface.1.IPv4Address.3.IPAddress 192.168.88.73
```

```
Device.IP.Interface.1.IPv4Address.3.SubnetMask 255.255.255.0
```

Data

- But instead we have an object

```
{ _timestamp: '2019-09-20T09:27:53.175Z',  
  IPv4Address:  
    { '4':  
      { _timestamp: '2019-09-20T09:27:53.175Z',  
        _object: true,  
        _writable: true,  
        Enable: [Object],  
        Status: [Object],
```

Data

- And to our misfortune, the references are in the genieacs-gui format

```
Device.IP.Interface.1.IPv4Address.3.IPAddress 192.168.88.73
```

```
Device.IP.Interface.1.IPv4Address.3.SubnetMask 255.255.255.0
```

```
Device.IP.Interface.1.Enable true
```

```
Device.IP.Interface.1.Status Up
```

```
Device.IP.Interface.1.LowerLayers Device.Ethernet.Link.1
```

Data

- The getvalue function comes to help us

```
exports.getvalue= function(obj,x) {  
    x.split(".").forEach(function(v) {  
        if (isNaN(v))  
            obj=eval('obj.'+v);  
        else  
            obj=eval('obj['+v+']');  
    });  
    return obj;  
}
```

Data

```
// Looking for the IP assigned to ether1
ifstack=misc.getvalue(obj,"Device.InterfaceStack");
tosearch='';
for (let [key, value] of Object.entries(ifstack)) {
    if (typeof(value) !== 'object')
        continue;
    if (value.LowerLayer._value=='Device.Ethernet.Interface.1')
        tosearch=value.HigherLayer._value;
};
```

Login Page



Athenea

Sign in to start your session

☐

Remember Me

Sign In

[I forgot my password](#)

[Register a new account](#)

Status Page



Athenea

remote ip 192.168.88.254

hAP ac2
Online

Status

Wireless

Status page

Internet

Interface Name	ether1
IPv4 Address	192.168.88.76
Subnet Mask	255.255.255.0
Gateway	192.168.88.1
MAC Address	4C:5E:0C:59:0F:12
Downloaded Bytes	3623338
Uploaded Bytes	3929018

LAN

Device Name	MAC Address	IPv4 Address
No devices available		

WLAN devices

Device Name	MAC Address	IPv4 Address
-------------	-------------	--------------

Wireles. Basic Settings



Wireless Basic Settings

Enable Wireless: Yes ▼

Network Mode: b/g/n ▼

SSID: test_BB339

Channel: Auto ▼

Update

Wireless. Security





Wireless Security

Security:

Passphrase:

Update

Example – Change wifi password



Wireless Tables

WiFi InterfacesW60G StationNstreme DualAccess ListRegistrationConnect ListSecurity ProfilesChannels

Name	Mode	Authenticatio...	Unicast Ciphers	Group Ciphers	WPA Pre-Shared ...	WPA2 Pre-Shared...
* default	none					
profile 1	dynamic keys	WPA PSK	aes ccm	aes ccm	testingloop	pruebaprueba

Example – Change wifi password

Wireless Security

Security:

Passphrase:



The new passphrase is **LisbonMUM2019**

Example – Change wifi password

Wireless Tables						
+ - 📁 🔍						
Name	Mode	Authentication Types	Unicast Ciphers	Group Ciphers	WPA Pre-Shared ...	WPA2 Pre-Shared...
::: [tr069 auto created]						
ap-security-0	dynamic keys	WPA PSK WPA2 PSK	aes ccm	aes ccm	LisbonMUM2019	LisbonMUM2019
* default	none					
profile1	dynamic keys	WPA PSK	aes ccm	aes ccm	testingloop	pruebaprueba



Test passed !!

If you want to obtain a copy of the virtual machine that I used in this presentation,

Please, send an email to:

j.castellet@yatuaprendes.com

And we will send you a download link.

Thank you for your attention.

