ISP's black box

provisioning behind the scenes

Luka Perkov

December 27, 2012

Outline

About the speaker

- Linux enthusiast
- hacking embedded devices
- OpenWrt developer

General

Broadband forum (originally named DSL Forum) was founded in 1994. Today it has around 300 members, mostly equipment vendors and ISPs.

This talk is focused on CPE WAN Management Protocol (*CWMP*) which originates from Technical Report 069 (*TR-069*). It defines an application layer protocol for remote management of end-user devices.

Terminology

- *CPE* – Customer Premises Equipment
- *CWMP* – CPE WAN Management Protocol
- *ACS* – Auto Configuration Server
- *provisioning* – the process of CPE configuration

Why

- How to effectively manage *10k*, *100k*, *1 million* or more CPEs?
- How to handle situation when you have different CPE vendors?
- How to define fine grained access to certain information?

Questions for audience

- Can you replace your CPE? Do you have credentials for the services you are using?
- When replacing user's CPE or connecting a new user, do they need to configure the CPE?

CWMP

- bidirectional SOAP/HTTP based protocol
- communication is in XML
- ~20 TR-* schemas and data model definitions
- *a lot* of objects and parameters

CWMP

Object

CWMP objects contain one or more parameters. For example in TR-181 ACS related parameters are defined in object:

- `Device.ManagementServer.`

Parameter

Values can be stored or read from a parameter. Few parameters from `Device.ManagementServer.` object:

- `Device.ManagementServer.URL`
- `Device.ManagementServer.Username`
- `Device.ManagementServer.Password`

CWMP

- reboot, factory reset, flash firmware
- save and restore configuration
- create or delete objects
- get or set parameter values
- get or set parameter attributes

CWMP

But what can you *actually* do? You can view and/or change:

- credentials for PPP, SIP and other services
- configuration for services like DNS, DHCP
- wireless settings
- routing
- firewall
- QoS

CPE

CPE *always* connects to the configured ACS URL. It connects on events like reboot, factory reset, periodic interval or after receiving connection request.

- can connect via *http* or *https* to the ACS
- listens for connection requests from ACS
- object `*Device.DeviceInfo.`

ACS

ACS can configure or get status from CPE by setting or reading appropriate parameter values.

- waits for inform messages
- object `*Device.ManagementServer.`

ACS

There are several methods you could use to find out URL for your ACS. *Disclaimer: Do this at your own risk.*

By the book

Take a look at TR-069 specification on page 13. You can see that ACS URL might be provided in DHCP option 43.

CPE hacking

If you can get console access to the CPE or reverse engineer the firmware search for `*Device.ManagementServer.URL` parameter value. Note that CPE might be ISP's property.

Overview

If ACS wants to change parameter value on CPE...

1. ACS initiates connection request
2. CPE sends inform message
3. ACS replies with inform response
4. CPE sends empty POST message
5. ACS replies with *SetParameterValues request* message
6. CPE sends *SetParameterValues response* message
7. ACS replies without content

Inform message

```
<soap_env:Body>
 <cwmp:Inform>
  <DeviceId>
   <Manufacturer>freecwmp</Manufacturer>
   <OUI>FFFFFF</OUI>
   <ProductClass>freecwmp</ProductClass>
   <SerialNumber>FFFFFF123456</SerialNumber>
  </DeviceId>
  <Event soap_enc:arrayType="cwmp:EventStruct[1]">
   <EventStruct>
    <EventCode>0 BOOTSTRAP</EventCode>
    <CommandKey/>
   </EventStruct>
  </Event>
```

Inform message

```
<CurrentTime>2012-12-27T12:45:00+01:00
</CurrentTime>
<RetryCount>0</RetryCount>
<ParameterList
 soap_enc:arrayType="cwmp:ParameterValueStruct[11]"
>
 <ParameterValueStruct>
  <Name>Device.DeviceInfo.SpecVersion</Name>
  <Value xsi:type="xsd:string">1.0</Value>
 </ParameterValueStruct>
 <ParameterValueStruct>
  <Name>Device.DeviceInfo.Manufacturer</Name>
  <Value xsi:type="xsd:string">freecwmp</Value>
```

Inform message

- IGD.DeviceInfo.Manufacturer
- IGD.DeviceInfo.ManufacturerOUI
- IGD.DeviceInfo.ProductClass
- IGD.DeviceInfo.SerialNumber
- IGD.DeviceInfo.HardwareVersion
- IGD.DeviceInfo.SoftwareVersion
- IGD.DeviceInfo.ProvisioningCode
- IGD.ManagementServer.ParameterKey
- IGD.ManagementServer.ConnectionRequestURL
- IGD.WANDevice.{i}. ... .ExternalIPAddress

SetParameterValues request

```
<soap_env:Body>
 <cwmp:SetParameterValues>
  <ParameterList
  soap_enc:arrayType="cwmp:ParameterValueStruct[1]"
  >
   <ParameterValueStruct>
    <Name></Name>
    <Value></Value>
   </ParameterValueStruct>
  </ParameterList>
  <ParameterKey></ParameterKey>
 </cwmp:SetParameterValues>
```

SetParameterValues response

```
<soap_env:Body>
 <cwmp:SetParameterValuesResponse>
  <Status>0</Status>
 </cwmp:SetParameterValuesResponse>
</soap_env:Body>
```

GetParameterValues request

```
<soap_env:Body>
 <cwmp:GetParameterValues>
  <ParameterNames
  soap_enc:arrayType="xsd:string[1]">
   <string></string>
  </ParameterNames>
 </cwmp:GetParameterValues>
```

GetParameterValues response

```
<soap_env:Body>
 <cwmp:GetParameterValuesResponse>
  <ParameterList
  soap_enc:arrayType="cwmp:ParameterValueStruct[1]"
  >
   <ParameterValueStruct>
    <Name></Name>
    <Value></Value>
   </ParameterValueStruct>
  </ParameterList>
 </cwmp:GetParameterValuesResponse>
</soap_env:Body>
```

Reboot request

```
<SOAP-ENV:Body>
 <cwmp:Reboot>
  <CommandKey />
 </cwmp:Reboot>
</SOAP-ENV:Body>
```

Factory reset request

```
<soap_env:Body>
  <cwmp:FactoryReset />
</soap_env:Body>
```

General

libfreecwmp, *freecwmp*, *mod_cwmp* and *freeacs-ng* projects are GPLv2 licensed software. Source code can be obtained via git:

```
git clone git://dev.libfreecwmp.org/libfreecwmp/
git clone git://dev.freecwmp.org/freecwmp/
git clone git://dev.freeacs-ng.org/mod_cwmp/
git clone git://dev.freeacs-ng.org/freeacs-ng/
```

libfreecwmp

- shared code ends up in this library
- at the moment ~200 lines of C

freecwmp

- CWMP client for (but not limited to) OpenWrt

Dependencies

- *uci*
- *libubox*
- *ubus*
- *microxml* (*Mini-XML* fork)
- *curl* or *zstream*
- *libfreecwmp*
- *shflags*

freecwmp

freecwmp core

The core part is coded in C and it is in charge of communication with ACS.

- offloads actual parameter handling to *freecwmp scripts*
- internally stores only a few parameters

freecwmp scripts

Number of scripts that integrate TR-* parameters.

- compatible with default OpenWrt busybox shell
- modular and extensible

mod_cwmp

- *nginx* proxy module for CWMP

Dependencies

- *nginx*
- *libxml2*

mod_cwmp

Standard *nginx* features

- *load balancing*
- *caching proxy*

CWMP specific features

- *protocol optimizer*
- *message inspector* (feature not publicly available)

mod_cwmp

- define *approved* and *rejected* location
- configure *approved* location so traffic is redirected to *upstream* ACS
- configure *rejected* location to execute appropriate action; for example to send factory reset command and log CPE's message

If message is flagged as malicious it will be internally redirected to *rejected* location, otherwise it will be redirected to *approved* location.

freeacs-ng

- Auto Configuration Server (ACS)
- under development

Dependencies

- *cscgi*
- *cnetstring*
- *libevent-2.0*
- *libxml2*
- *rabbitmq-c*
- *uci*
- *libfreecwmp*

freeacs-ng

- SCGI server
- Advanced Message Queuing Protocol (AMQP) client

Workflow

1. upon receiving initial connection from CPE, *freeacs-ng* informs *provisioning back-end* using AMQP
2. *provisioning back-end* will send a message to the appropriate AMQP exchange once it has collected the data
3. CPE connects to the *freeacs-ng* after receiving request from *connection requester*
4. *freeacs-ng* provisions the CPE

freeacs-ng

At the moment if there is anything in the AMQP queue it will send CWMP reboot command, otherwise it will reply with HTTP "204 No Content" status.

TODO

- define AMQP message content and transfer rules
- code basic *provisioning back-end* in a few languages
- code *connection requester*

Further Reading

- http://www.broadband-forum.org/cwmp.php
- http://freecwmp.org
- http://freeacs-ng.org
- http://libfreecwmp.org

Contact Information

- IRC channel on Freenode: `#freecwmp`
- Mailing list: `freecwmp@linux-mips.org`
- `luka@openwrt.org`