Asterisk SOAP How-to

January 13, 2006

Abstract

This How-To shows how a simple CGI-file can be used as a SOAP-interface to initiate calls with Asterisk. Because of its standardized interface, it can easily be used to initiate calls from remote machines. Although this is no *enterprise*-solution, it is a portable solution for private networks when limited resources (memory / speed) are an issue.

1 Introduction

This is the second article in the series of mini How-to articles on configuring Asterisk on limited resource devices such as OpenWRT-routers. Running on a device with limited resources (cpu-speed / memory), part of the problem is to use only the standard tools available, especially that are part of the commands provided by the *busybox*. Focusing on these basic commands also ensures that the solution will most likely run on other systems as well.

This article shows how a simple CGI script can be used to initiate calls. It has been tested on a Linksys WRT54GS v. 1.1 running OpenWRT *white russian rc4* and the *asterisk-cvs* package. This solution cannot be considered to be *enterprise-grade*, because of the lack of authentication methods and security checks of the request. When you need help implementing such a more advanced solution, feel free to contact the author.

2 CGI-file

SOAP *(Simple Object Access Protocol)*-requests are standardized XML messages. Although these messages could be transfered to another server in various ways , HTTP POSTs are used most often. Our CGI-script shown here does neither check the syntax of the request nor authenticate the initiator, but simply tries to extract the data it needs to place the call. The call is placed by a *.call*-file in the `/var/spool/asterisk/outgoing` directory, which Asterisk constantly monitors.[1] This script assumes that a context named `default` exists in the `extensions.conf` and is appropriate for originating calls.[2]

```
#!/bin/ash
SERVER_NAME="192.168.1.1"
if [ "$REQUEST_METHOD" = "POST" ] ; then
  SOAP_INPUT=`dd count=$CONTENT_LENGTH bs=1 2> /dev/null`"&"
  acc=`echo "$SOAP_INPUT"|grep acc|sed -e 's/^.*<acc.*>\(.*\)<\/acc>.*$/\1/g'`
  num=`echo "$SOAP_INPUT"|grep num|sed -e 's/^.*<num.*>\(.*\)<\/num>.*$/\1/g'`
  echo -e "Channel: SIP/$acc\nMaxRetries: 1\nRetryTime: 60\nWaitTime: 30\n\
Context: default\nExtension: $num\nPriority: 1" > /tmp/FileToMove.call
```

[1] This feature is enabled by the `spool`-module in Asterisk.

[2] If Asterisk is running not as root, you must also `chown` the created .call file to the user Asterisk runs as.

```
# chown asterisk:asterisk /tmp/FileToMove.call
  mv /tmp/FileToMove.call /var/spool/asterisk/outgoing/.
  resp=`cat <<EOF
<?xml version="1.0"?>
<SOAP-ENV:Envelope
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body><ns1:dialResponse xmlns:ns1="$SERVER_NAME$SCRIPT_NAME">
<return xsi:type="xsd:string"> SUCCESS </return></ns1:dialResponse>
</SOAP-ENV:Body></SOAP-ENV:Envelope>
EOF
`
  len=`echo -n "$resp"|wc -c|awk '{print $1}'`
  echo -e "HTTP/1.1 200 OK\nContent-Length: $len\nContent-Type: text/xml\n"
  echo -n "$resp"
 fi
```

Because this shell script is sensitive to trailing *whitespace*, ensure that the lines end right after the *backspace* (\) and the `EOF` term. Setting the variable `SERVER_NAME` to the IP of the server (in our case 192.168.1.1) is only necessary on those systems (like the *busybox*) where the CGI-variable is unset. The file has to be executable (i.e. use `chmod +x` *filename*) and be located in the *cgi-bin*-directory (which is by default `/www/cgi-bin` on OpenWRT) to be serveable by the web server.

3 Access-Example

There are numerous ways[3] to make SOAP-calls. On a Macintosh you can simply use AppleScript:[4]

```
set myPhone to "10"
set numberToCall to "1234567890"
tell application "http://192.168.1.1/cgi-bin/soapcall.cgi"
  set paras to {acc:myPhone as string, num:numberToCall  as string}
  call soap {method name:"dial", parameters:{{paras}}, SOAPAction:"/dial"}
end tell
```

This script assumes that the former CGI-script can be accessed under the URL `http://192.168.1.1/cgi-bin/soapcall.cgi`. Your phone should be accessed under the SIP extension 10.

Disclaimer

Please contact the author in case you find any errors or think of any ways on how to improve this paper.

Although the information and code has been edited carefully and seems to work as expected on our machine with our data, please be advised that any use of it is at your own risk and without any warranty. Using the information / code given in this paper may result in loss of data as well as damage and (costly) malfunction of your equipment. Since the script has no access control, make sure it can only be accessed by authorized persons.

[3] If you want to make SOAP calls with PHP, you may want to consider using the *nusoap* library.

[4] If you experience problems on making your SOAP requests, feel free to contact the author for help.