Advanced Call Control

Using cURL to implement DONOTCALL with

SIParator® / Firewall®

(Illustrated case using DNC Lookup API from RealPhoneValidation.com)

For the Ingate SIParators using software release 6.2.1 or later

Revision 1

May 10th, 2019

Ingate SIParator®/Firewall®

Table of Contents

1 Do Not Call Compliance explain

There are numerous compliance laws and regulations at the state and federal level that any company or call center could be in violation of and not even know it. There are tools and providers in the market that can help your company to be in compliance.

Cold calling is something regulators have been trying to control and make the voice of the citizen to rule their way.

There are facts that you should be aware of before you make a decision on how to implement a compliance strategy.

All these regulations and rules are commonly known as "Do-Not-Call" and violating them may cost you tens of thousands of dollars.

In some states for example, the US Government prohibits telemarketers from calling wireless numbers, regardless of if you use an automated dialer or not.

Another example is Louisiana prohibits solicitations during a state of emergency when the Public Service Commission has been required to notify the Office of Homeland Security and Emergency Preparedness.

Under TCPA, telemarketers are restricted from calling consumers before 8 a.m. and after 9 p.m. Call curfew requirements apply based on the local times of the called phone number, not on the location of the telemarketer.

But states have their own rules too on calling curfews. For example, Kentucky prohibits calls before 10 a.m.

If your organization utilizes independent contractors or agents to sell your services, you are liable for their telemarketing violations.

Unlike criminal law where an individual is assumed innocent until proven guilty, telemarketers always bear the burden to prove he or she is not liable for a DNC violation.

There are much more additional details on regulations at Federal and State level, including consumers and B2B.

2 Our use case explained.

In this document just to be able to explain how to take advantage of Advanced Call Control in our SIParator, we have chosen and specific scenario that has not the intention to resolve all Do-Not-Call Challenges, but illustrates how to develop an implement an external application that validates compliance in the middle of an outbound call setup.

Even there are several other considerations that can make the application even more complex, we will keep our focus on using an external service provider vial RestAPI calls to validate a number to conclude if it belongs to any national or state level Do-Not-Call list. If so is the case, we will abort the call and therefore avoid a Do-Not-Call violation.

In order to illustrate our case we will use https://RealPhoneValidation.com, and specifically we will interface our SIParator Call Control RestAPI with their RestAPI interface for the service they call DNC Lookup.

For every outbound call to the PSTN via a SIP Trunk provider, we will check during call setup with DNC Lookup and decide if the call is finally routed to the Trunk or just dropped.

For information on how RealValidation API works you can review this document:
https://drive.google.com/file/d/0B4Pn-GVvGTbYOFR5aHhINjA3Zm8/view

NOTE: Telemarketers wanted to use RealValidation are required be the FTC to have a SAN - you can sign up for one here: https://telemarketing.donotcall.gov/

Typical cURL request will look like:

"https://api.realvalidation.com/rpvWebService/DNCLookup.php?phone=number&token=xxxxxx"

Where **Phone=** will be used to pass the **number** we want to validate and **Token=** is the **security key** provided to you associated to your account in realvalidate.com

RealValidation can provide a response either in JSON format or XML format. Similar to this examples:

JSON:

```
{
   "RESPONSECODE":"OK",
   "RESPONSEMSG":{},
   "national_dnc":"Y",
   "state_dnc":"N",
   "dma":"N",
   "litigator":"N",
   "iscell":"N",
   "id":"4"
  }
```

XML:

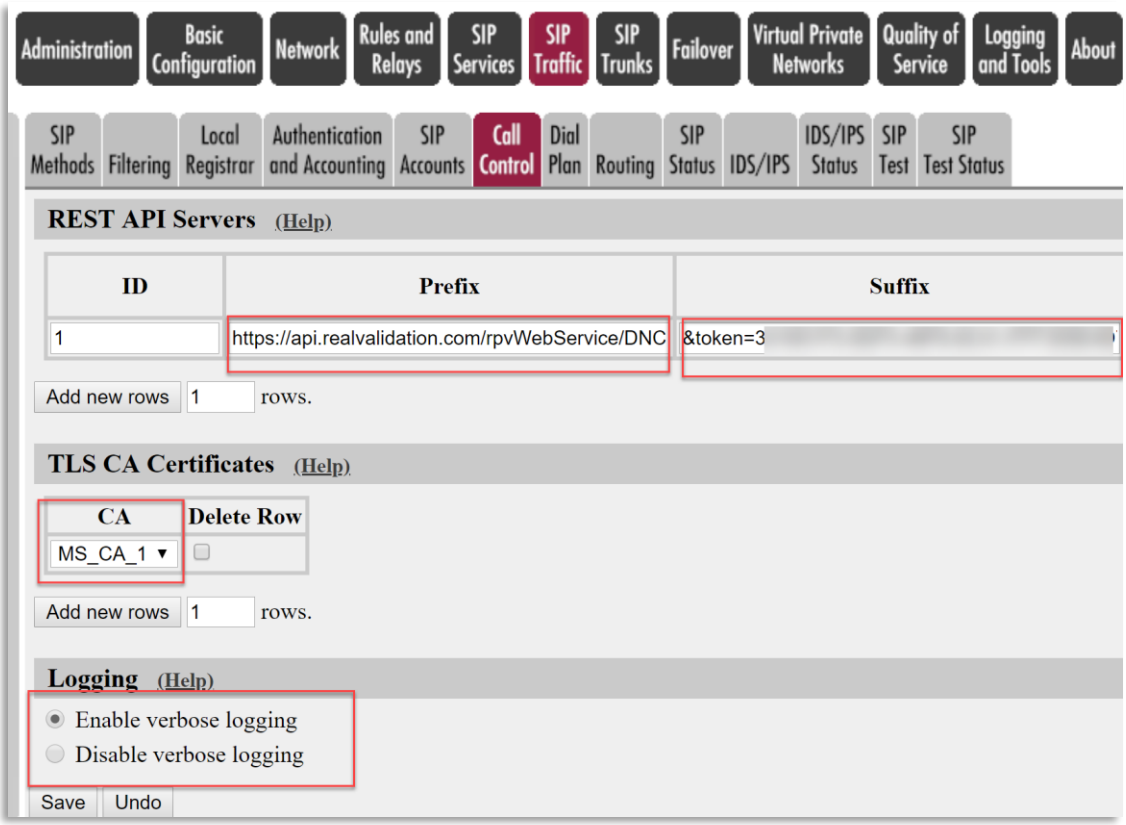
```
<?xml version= "1.0 " encoding= "UTF-8 "?>
 <response>
  <RESPONSECODE>OK</RESPONSECODE>
  <RESPONSEMSG></RESPONSEMSG>
  <national_dnc>Y</national_dnc>
  <state_dnc>N</state_dnc>
  <dma>N</dma>
  <litigator>N</litigator>
  <iscell>N</iscell>
```

<id>4</id>
</response>

3 SIParator configuration

There are 3 places where we will need to configure what is required to implement Advanced call control. The first location in under “SIP Traffic”, a section called “Call Control”, where we will define each RestAPI Servers needed. In our case it will be only one (RealValidation). Second, we will use the Dial Plan, under SIP Traffic match outbound calls and apply cURL as part of the regular expressions used in the “Forward To:” section.

3.1 Configuring Call Control



Here you can define REST API servers for call control features in the Dial Plan.

The servers are accessed from the Dial Plan using the ID value. REST API invocations in the Dial Plan start with $curl, then the ID value, and then in parenthesis the URL for the REST API call which is a HTTP(S) GET command. The table of REST API servers is intended to shorten the URL written in the Dial Plan, by allowing to define a static prefix and suffix for the URL. For example, specifying server ID '5', a prefix which is "http://my-call-control-server/rest/" and a suffix which is "?auth=myauthtoken", allows you to write

"sip:$curl5($1/fwd)" in the Reg Expr field in the Forward To table, the 5 after $curl is the server ID. If you have "sip:(.*)@(.*)" in the Reg Expr field in the Dial Plan's Matching Request-URI table, then for an incoming INVITE message to 99@my-sip-server.com the $1 will resolve to 99 so that the whole URI for the REST API GET command will be "http://my-call-control-server/rest/99/fwd?auth=myauthtoken".

Your REST API server should send back a response with a body that is either XML or JSON or text. If the response is text, the whole text will be used when resolving the REST API invocation. So, if your server in the example above returns a body with content-type text and content "new_dest@new_host.com", then the incoming INVITE will be rewritten and forwarded to new_dest@new_host.com.

If the response body is XML, for example

"<?xml version="1.0" encoding="UTF-8"?><root><fwd>@new_host.com</fwd></root>",

you have to specify exactly which part of this response you want to extract and use. This is done by using the XPATH language standard.

In this example, to get the new destination out of that document, you can use the XPATH expression "//fwd/text()". To do this you need this expression in the Dial Plan:

"sip:$curl5($1_XPATH//fwd/text())".

So you have to add the XPATH expression after the curl arguments, preceded by _XPATH.

If the response body is JSON, we distinguish two cases.

In the most simple case that the body consists only of a value surrounded by double quotes (which is a valid JSON expression), the double quotes are removed and the remainder is used like in the case of text content described above (For example if the JSON body is "new_dest@new_host.com"). For more complex JSON content, the JSON content is converted to XML content, and then an XPATH expression can be used like described above.

For example, the JSON content {"fwd":"@new_host.com"} will be converted to the XML content <?xml version="1.0" encoding="UTF-8"?><root><fwd>@new_host.com</fwd></root>. To extract the new destination you can use the XPATH expression "//fwd/text()", so in the Reg Expr field in the Dial Plan's Forward To table you put this: sip:$curl5($1_XPATH//fwd/text())

Hint: Set logging to verbose. You can look at the result of the conversion from JSON to XML in the log, search for "JSON to XML conversion result".

Hint: You can specify everything in the Dial Plan without using the table of REST API servers, in this case you have to put the full URL (including prefix and suffix) directly into the Dial Plan, and use only $curl(...) instead (without any ID value).

As for the CA certificate, select the trusted certificates which are needed to verify the server certificate for the https requests from this box. If no certificate is selected, then the server certificate will not be verified, which is not secure!

3.2 Configuring Dial Plan

First of all, we have to explain how we are going to decide is a call is routable to the destination or not. In our example the criteria will be:

We won't route the call if one or more of the following responses happen:

1) Dialed Number is in the National DNC list (national_dnc is "Y")
2) Dialed Number is in the State DNC list (state_dnc is "Y")
3) Dialed Number is in the DMA list (DMA is "Y")
4) Dialed Number belongs to a Serial TCPA Litigator (litigator is "Y")
5) Dialed Number is a Cellphone (iscell is "Y")

If any of these circumstances happens, we will terminate the call. Of course, this is just an example. It can be more complex as for instance, in some state's cold calls to a cell phone are not banned. In other states there is also rules that applies in certain times of the day for the time zone of the destination.

In order to effectively process all outbound calls and taking advantage of SIParator full SIP Proxy capabilities we will process every call in two steps.

1) First step will be to qualify/validate the call, tag it and then send the call again to itself to be processed based on the tag.
2) Second will identify tagged calls and be processed depending on the tag.

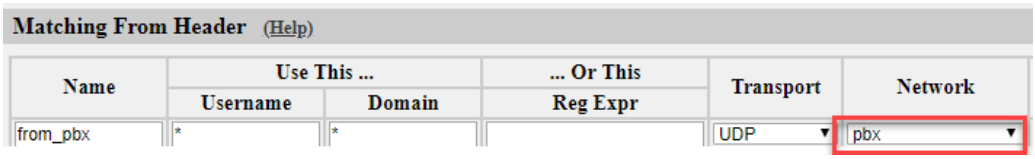
In order to tag the call, we will just add a prefix to the call.

1) "Y" tag means call should be terminated
2) "N" tag means call should be untagged and routed to destination.

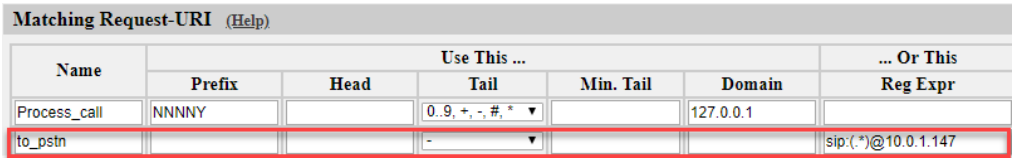
In other words, if we find a way to concatenate all responses for each one of the 5 values indicated before, we can later parse such stream and look for any "Y" match. If the match happens, then the call should be terminated (or routed someplace else), otherwise it can be routed to final original destination.

Our example dial plan will look like this:

1) Lets match From header and R-URI (from where and to where is the call INVITE):

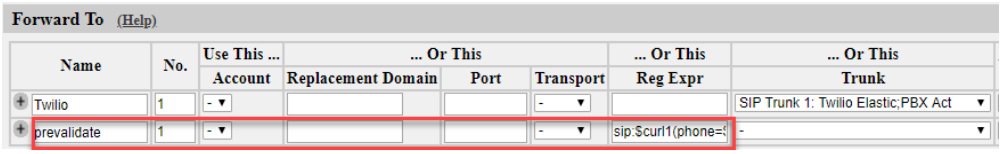


Matching the call originated from the PBX



Check the call was received in the SBC and capture the dialed number in $r1

2) Lets define potential destinations. Here it is important to explain that we will pass the call twice thru the Proxy. In the first pass we will invoke the Validation Service to Verify 5 points about the destination to qualify if we can send or not the call to the final destination.



To prevalidate the call we will send a request to the cURL Server 1 using this expression:

sip:$curl1(phone=$r1_XPATH//national_dnc)$curl1(phone=$r1_XPATH//state_dnc)$curl1(phone=$r1_XPATH//dma)$curl1(phone=$r1_XPATH//litigator)$curl1(phone=$r1_XPATH//iscell)$r1@127.0.0.1

This will return a sip r-uri like this:

Sip:XXXXX1234567890@127.0.0.1

XXXXX is a 5 characters represent each one the value for each element in the response coming from the dnc service provider, and will be added as a prefix in front of the dialed number:

national_dnc: Y, N or ?

state_dnc: Y, N or ?

dma: Y, N or ?

litigator: Y, N or ?

iscell: Y or N

assuming here 1234567890 is the dialed number (Destination)

As the call host destination is 127.0.0.1, the call will be processed again by the Dial Plan.

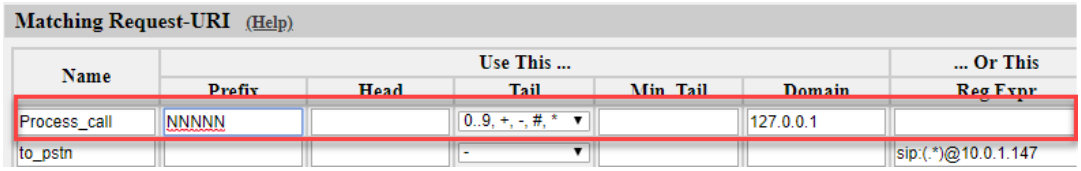
3) This is the second pass of the call which now has a prefix added as a result of the pre-validation step.

To catch calls in this second stage we will match (one or more matches) depending in our criteria to process the call to final destination.

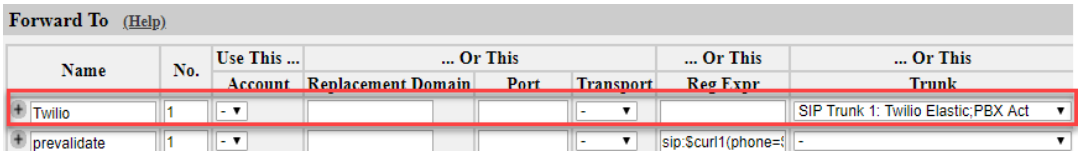
Let's say for example that we will only process calls with a prefix of NNNNN, which means in this case:

This destination number is not in the national dn0c list, not in the state dnc list, not in the dma list, not a litigator, not a cellphone.

In any other case, the call will be dropped.



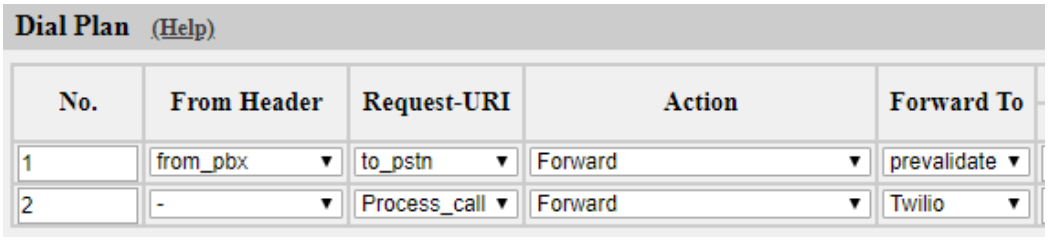
Match and strip out the prefix, then send the call to itself again to finally route the call to the SIP Trunk Provider.



Forward destination defined to be able to route the valid call to the SIP Trunk Provider.

4) Final routing to destination. After the second pass, if the call matched the prefix (pass criteria to send the call to destination),

   So, we really have two lines in the actual dial plan:



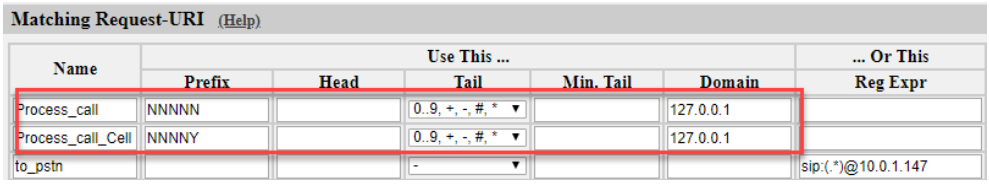
Line 1, used to send the call to itself to do the pre-validation. This line is executed if From Header shows the call is coming from the PBX, and it was sent from outside to the SBC IP address. Here is force the call to be routed to pre-validate destination (itself using the regular expression that invokes the DNC service provider)

Line 2, used to route calls to final destination if Request URI match the Process_call criteria (Selected prefix to authorize the outbound).

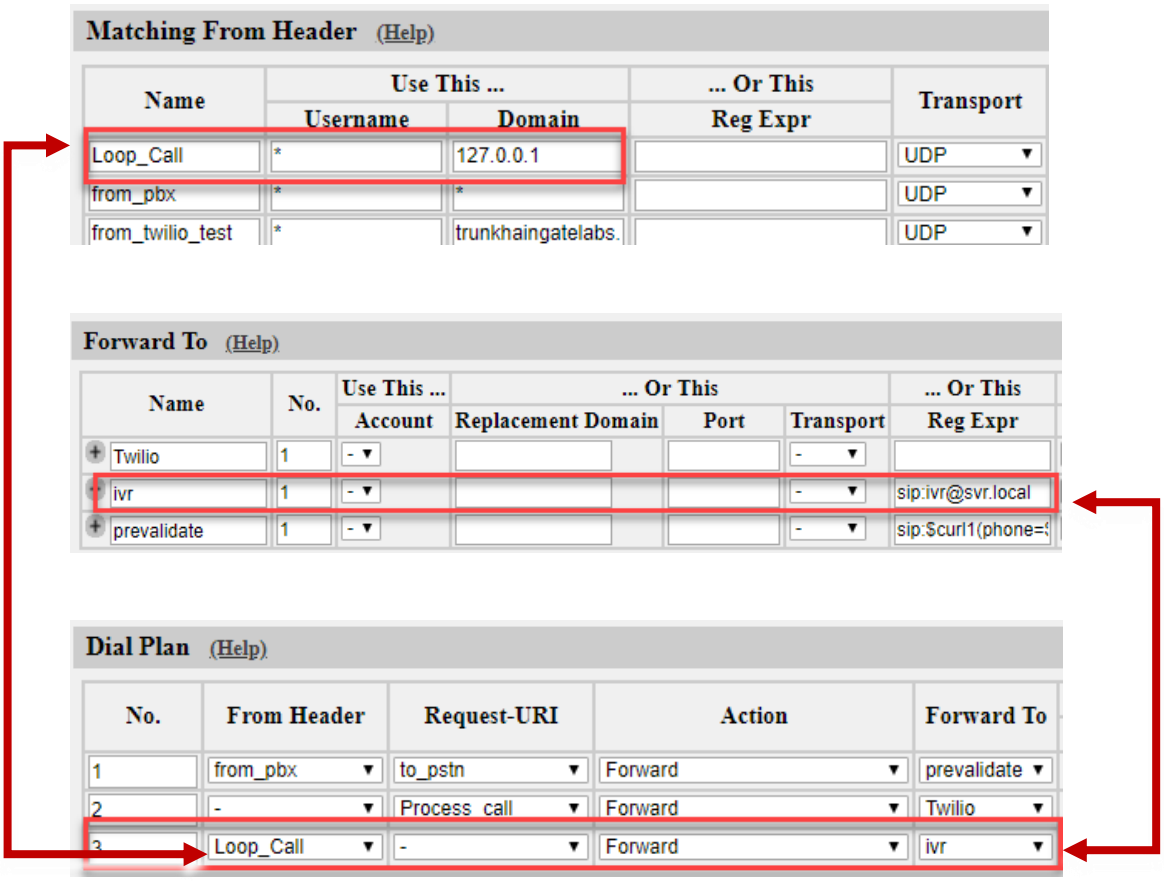
3.3 Expanding criteria in the dial plan.

Two enhancements can be done here:

1) Add more matching cases which define allowed outbound calls. For instance, with the existing prefix, we are not allowing call to cellphones to be processed. Let's assume we want to provide also termination to cellphones:



2) When a call is not allowed to be terminated, send the call to an IVR that will play back to the caller some message.



With these 3 additions, any call that is not allowed to be terminated will be sent to an ivr located at irv@srv.local

4 Additional help or support

If you have questions, suggestions and any other concern feel free to contact Educronix LLC

Web: www.educronix.com

Email: support@educronix.com

Toll-Free: +1 855 866 8854

Ph: +1 954 866 8884

We also provide consulting services as well as remote hands troubleshooting and configuration.