SecureMyForm

In this page, you can find an example of a web App written in PHP, which enables the validation of a form via an IVR extension.

Overview

This App is useful for any developer of PHP applications as it contains basic examples of how to integrate third-party applications with VoipNow via VoipNow Call Interactive.

In order to be able to use the SecureMyForm App, you need a webserver with PHP 5.3 or higher installed.

Installation

STEP 1: Retrieve files from our GitHub repository:

```
git clone https://github.com/4psa/app-securemyform.git
```

STEP 2: You need to give proper permissions and ownership rights to the files of the plugin - the plugin needs to be owned by the web server user and group.
For example, if we use `httpd` as a webserver:

```
chmod -R 755 <PATH_TO_CALLPARENT>
chwon -R httpd:httpd <PATH_TO_CALLPARENT>
```

STEP 3: In order to use the App, you must first setup the database by running the SQL script located at `/sql/securemyform.sql`.

Now use those values to run the supplied SQL script:

```
mysql -u<db_user> -p<db_pass> securemyformdb < sql/securemyform.sql
```

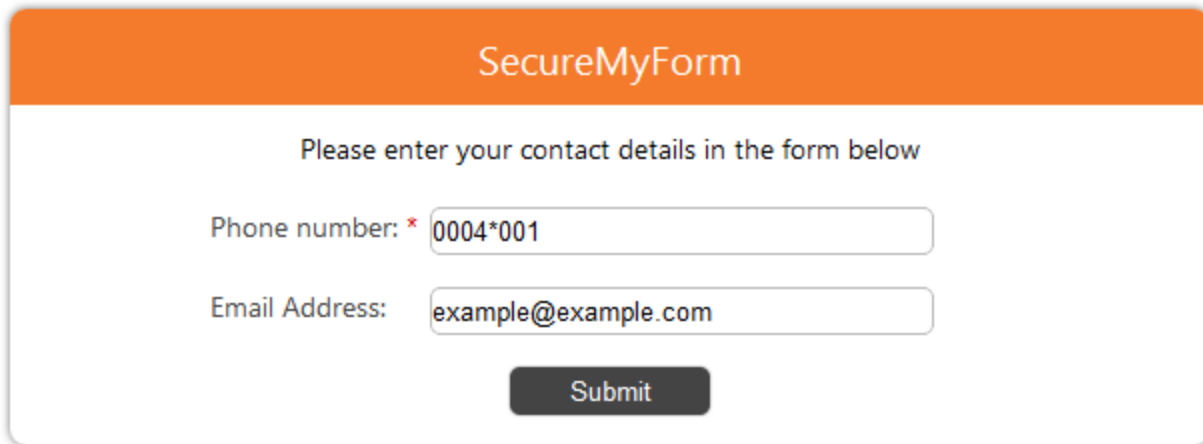You should have your DB admin setup a database for you, e.g. `securemyformdb`.

STEP 4: As soon as the database is setup, you must configure it by opening the file `/config.php` in the `<PATH_TO_SECUREMYFORM>/interface/plib/config.php` and changing the following settings:

Name	Description
VN_SERVER_IP	Variable to the IP or hostname of your VoipNow server;
VN_IVR_EXTENSION	Variable to the IVR extension that will be called by the number entered in the form;
VN_CHARGE_EXTENSION	Variable to the extension that will initate the call, this is the extension that will call the number entered in the form;
RETURN_URL	Variable to the URL the request will be redirected to as soon as processed; You should use the following URL: ```http://<ip|host>/<PATH_TO_SECUREMYFORM>/interface/process.php```
OAUTH_ACCESS_TOKEN	OAuth access token. This token is sent to the Unified API using the **Authorization** HTTP header and has a limited life span of 2 hours, so be sure to renew it
MYSQL_HOST	Variable to the IP/host the database is located at;
MYSQL_USER	Variable to the username used to access the database;
MYSQL_PASS	Variable to the username used to access the database;
MYSQL_DBNAME	Variable to the database name.

Behaviour

User Interface

STEP 1: Go to the URL `https://<PATH_TO_SECUREMYFORM>/index.php` and you will be able to see a form requiring your phone and email address details.
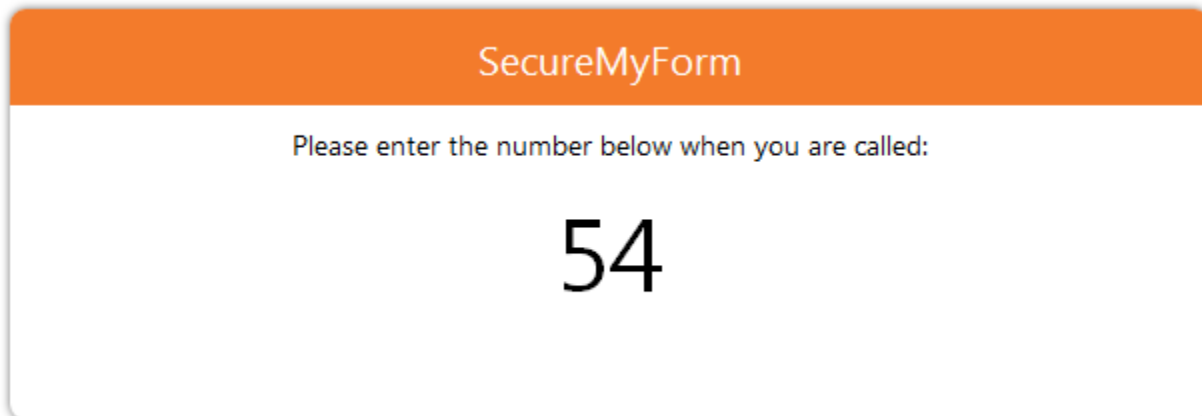

STEP 2: Fill in your phone number and press <u>Submit</u>.

That number is called and once you pickup that number a new call will be placed to the IVR.

A warning about 'CURLOPT_FOLLOWLOCATION' will be generated each time the request is made. Don't worry, this is ok and it won't interfere with the request.

STEP 3: The form will refresh and show two digits that will be required when recording digits to variable.

STEP 4: Once the two digits are validated in the IVR the form will refresh and show that the request is completed.

Internal Flow

As soon as the configuration process is completed, you need to **configure the IVR extension**. This extension registers the number entered by the user and then calls the script ensuring the validation process.

As a result, you need to add an IVR action as described in the Add IVR Context section. The following actions must be added and executed according to the order indicated below:

01. Play Sound

You should add a sound containing the user instructions.

02. Record digits to variable

You must set the IVR to register a number of digits in a variable called `activationCode`. This variable will be transmitted to the script validating the request.

03. Call Interactive

The configuration implies the following:

- Request Method - must be set to the method used to submit the form - by default it is configured to GET.
- Make request to - must be set to an URL of the following form:

```
http://<ip|host>/<PATH_TO_SECUREMYFORM>/ivr_validate.php?activationCode=$activationCode
```

For more details, go to IVR Extension.

When the button is pressed a new call is placed using the UnifiedAPI Create Simple PhoneCalls request:

```
POST /unifiedapi/phoneCalls/@me/simple HTTP/1.1
HOST x.x.x.x
Content Type: application/json
Authorization: <OAUTH_ACCESS_TOKEN>

{
   "extension":"<VN_CHARGE_EXTENSION>",
   "phoneCallView":[
      {
         "source":["phoneNumber"],
         "destination":["<VN_IVR_EXTENSION>"]
         "callerId":"SecureMyForm <8887777>"
      }
   ]
}
```

After the call is made, periodic requests are being made to status.php in order to get the call's status.

When you pick up the call and type in the number, the VoipNow server makes a request to the url found in the "Make request to" field in the action settings of that IVR. It contains information about the call, number pressed and other info. See the Interactive Request Parameters.

The script that the server made the request to responds, giving an XML formatted string which tells it which action to carry. In our example, it hangs up. Please see Interactive Response Structure for more details.

This is the function that makes the call. It returns the ID of the call.

```php
function call($phoneNumber) {

        global $msgArr;

        $config = getConfig();
        /* This is the URL accessed using the REST protocol */
        $reqUrl = 'https://'.$config['VN_SERVER_IP'].'/unifiedapi/phoneCalls/@me/simple';

        $request = new cURLRequest();
        $request->setMethod(cURLRequest::METHOD_POST);
        $request->setHeaders(array(
                'Content-type' => 'application/json',
                'Authorization' => $config['OAUTH_ACCESS_TOKEN']
        ));


        $jsonRequest = array(
                'extension' => $config['VN_CHARGE_EXTENSION'],
                'phoneCallView' => array(array(
                        'source' => array($phoneNumber),
                        'destination' => $config['VN_IVR_EXTENSION']))
        );

        $request->setBody(json_encode($jsonRequest));
        $response = $request->sendRequest($reqUrl);

        $jsonresult = json_decode($response->getBody(true) ,true);

        return $jsonresult[0]['id'];
}
```