

Authentication for E-commerce

Replace the static password login on your website through a secure One-Time-Password login. There are only a few minor changes to the existing environment.

The goal is to keep the implementation time as low as possible.

Installation of SAVERNOVA authentication web application or web service

Install the SAVERNOVA authentication web application or web service. The application can be installed either on the same web server as the customer's application (no need to secure the communication channel because all communication is done locally on the server) or on another server in the network (SSL, HTTPS, VPN or some other secure channel or encryption should be used for the communication). This web application/web service is used to create OTP (one time password) challenges for users and to check their passwords (OTPs). The used communication protocol is XML-RPC, a simple remote procedure call protocol which uses XML to encode its calls and HTTP as a transport mechanism. XML-RPC is supported in all major programming languages and platforms (PHP, Java, ASP.NET, ...) but in case it was necessary the whole communication can be done using only HTTP protocol.

Definition of user registration process

Change/extend your user registration process. The card activation has to be included as part of the registration, either by replacing the current password definition step or in addition to it. The card activation process is simplified because the SAVERNOVA card can be assigned automatically (in this case only reading method definition and confirmation is needed). If manual assigning of the cards has been decided (e.g. allowing users to import and use their own same cards on multiple web sites) additional card importation step would be necessary.

SAVERNOVA SDK

Database Definition

Decide whether you want to extend your current database in order to store the SAVERNOVA cards, user reading methods and other information necessary or use a separate database for this information. Depending on your needs the database structure can look like this:

tblSAVERNOVA_SECURITY_GROUPS

- SG_ID (PK)
- SG_name
- SG_logins
- SG_password_age
- SG_password_times
- SG_show_card
- SG_use_secure_code
- SG_secure_code_position

tblSAVERNOVA_USERS

- USR_ID (PK)
- SG_ID (FK)
- USR_reading_method

tblSAVERNOVA_CARDS

- CRD_ID (PK)
- USR_ID (FK)
- CRD_content
- CRD_secure_code
- CRD_flags
- CRD_status
- CRD_imported
- CRD_assigned
- CRD_activated
- CRD_expired
- CRD_starting_positions

tblSAVERNOVA_SESSIONS

- session_ID
- USR_ID
- CRD_ID
- CRD_starting_position

User login process

Change your login web page and script to communicate with the SAVERNOVA web application/web service, to show the OTP challenge and if necessary the SAVERNOVA card on screen

Example

Let's assume you are using login/password authentication on your website. You have a login web form called login.html (login.php, login.aspx, ...) with two labels and two textboxes - login and password. This web form is sent back to the customer's server where the server side script (login.pgp, login.aspx,...) takes web form's parameters and compares them with the database. If the password is correct users can access the protected content or their user account. If the password is wrong his access is denied.

Changes to be considered

We will divide the login process into two steps. In the first step the user provides his login name only. This login name and session ID is sent to the server side code (same as before by posting the web form or by using AJAX technology) where an XML-RPC request is created and sent to SAVERNOVA web application/web service (this is new). In return, SAVERNOVA application/service creates a new OTP challenge and sends it back to customer's web application server side code.

The second step is to show the challenge information to the user (starting position, card on screen, secure-code, etc.) together with a password field where the user writes his/her one-time-password (OTP). This OTP is then sent again to the customer's server side code and a new XML-RPC request is created and sent to the SAVERNOVA web application/web service. SAVERNOVA application/service compares the OTP with the database and an Ok/Error response is sent back to the customer's web application which then according to this information continues the same way as before by either granting or denying the access for the user.

Contact

Savernova Ltd
info@savernova.com

Before SAVERNOVA implementation:

```
login.html (simplified)
...
<form action="login.php" method="post">
  Login: <input id="login" name="login" type="text" />
  Password: <input id="password"
            name="password" type="password" />
</form>
...

login.php (simplified)
<?php
...
$login = $_POST["login"];
$password = $_POST["password"];

if (check_password($login, $password))
  access OK
else
  access denied
...
?>
```

After SAVERNOVA implementation:

```
login.html (simplified)
...
<form action="login.php" method="post">
  <div id="step1">
    Login: <input id="login" type="text" />
  </div>
  <div id="step2">
    Starting position: <div id="starting_position"></div>
    Password: <input id="password" type="password" />
  </div>
</form>
...

login.php (simplified)
<?php
...
//step2
$password = $_POST["password"];
if (XMLRPC->CheckPassword(sessionID, $password))
  access OK
else
  access denied
...
?>
```