BOMGAR™

Bomgar Verify
VB.NET Authentication API

TC:5/3/2017

Table of Contents

Name Space SecurEnvoy.Authenticate

Bomgar Verify leverages the SecurEnvoy.authenticate name space to provide two-factor authentication. To use this name space, add a reference to the file authAPI.dll and declare it. For example, add the following line for VB.NET:

```
Dim admin As New securenvoy.authenticate
```

Two additional files, **.CHILKATENCRYPTLib.dll** and **Interop.CAPICOM.dll**, must exist in the same directory as **authAPI.dll** for this API to operate properly and must be on the same server as the SecurEnvoy server software.

Note: *Ensure the compiler targets CPU type x86 and not x64. Configure the project properties to target .NET 3.5.*

Method

Name	Description
authenticate()	The property **sUserID** must include the userid of the user who is authenticating. The property **sPasscode** must be set to the authenticating user's PIN and passcode. If PIN or passcode are incorrect, the system returns **OK, DENIED**. If the PIN or passcode are correct, the system returns **OK, AUTHOK**. If Real-Time, Voice Vall, or Two Step, the system returns **OK, SESSION, sSessionKey**. If a failure occurs, the system returns an error message.

Properties

Name	Type	Description
bDebug	Boolean	If set to True, debugging is enabled.
sDebugFile	String	The filename where debug information is sent and writes to c:\debug.
sInstallDir	String	The directory location of the Verify server. By default, this setting is retrieved from the registry.
sPasscode	String	The PIN and passcode needed to authenticate. Note that the PIN defaults to the user's LDAP password.
bRealTimePostChallenge	Boolean	This item only applies to real-time passcodes, Voice Call, or bUseTwoStepAuthentication. Set to False when authenticating with the PIN and password for the first time. Set to True when authenticating for the second time.
sSessionKey	String	This item only applies to real-time passcodes, Voice Call, or bUseTwoStepAuthentication and is set when authenticating the first time. This must be set with the same value returned in OK, SESSION.
sUserID	String	The userID of the authenticating user.
bUseTwoStepAuthentication	Boolean	If True, all token types operate over two dialogs like with real-time and voice call. The first dialog prompts for sUserID along with a PIN or password (no passcode), which is set in sPasscode. Then it returns sSessionKey. The second dialog prompts for the passcode only, which is also set in sPasscode.

VB.NET Example

This sample assumes the following have been configured:

- Label1 – UserID prompt
- Label2 – Password / Passcode prompt
- TextBoxUserID - UserID Entry
- TextBoxPasscode - password first, then passcode entry

```
authenticate.bDebug = True ' setup debug, should be set to false on final role-out
authenticate.sDebugFile = "test_auth.txt" ' this file is created inc:\debug
```

```
' Setup the display
Label2.Text = "Pin or Password:"
Label1.Visible = True
TextBoxUserID.Visible = True
```

```
' Call authenticate object
authenticate.bUseTwoStepAuthentication = True ' all token types use two steps
authenticate.sUserID = TextBoxUserID.Text ' UserID
authenticate.sPasscode = TextBoxPasscode.Text ' password step 1 then passcode step 2
Dim sReturn as String = authenticate.authenticate ' call auth
```

```
' Check for a second step (required for realtime, Voice Call or UseTwoStep is True)
If Left(sReturn, 10) = "OK,SESSION" Then
Step 2, need to prompt for a passcode
Label2.Text = "Passcode"
TextBoxPasscode.Text = ""
Label1.Visible = False
TextBoxUserID.Visible = False
```

```
 ' Set RealTimePostChallenge to True and return the session key
authenticate.bRealTimePostChallenge = True
authenticate.sSessionKey = Mid(TextBoxReturn.Text, 12)' cut to key
```

```
ElseIf sReturn = "OK,AUTHOK" Then
' Access Accepted
authenticate.bRealTimePostChallenge = False
```

```
ElseIf sReturn = "OK,DENIED" Then
' Access Denied
```

```
    authenticate.bRealTimePostChallenge = False
```

```
    Else
    ' Error
    authenticate.bRealTimePostChallenge = False
```

```
    End If
```

Security Recommendations

The minimum security requirements and practices provided should be followed:

- The security cookie used to prevent re-authentication must be used only through HTTPS.
- The security cookie must be session-based, which means it is maintained in the memory and not written to a disc.
- The security cookie must have the UserID and validity time embedded and must be tamper resistant. The data must be encrypted or hashed with a secret key stored on the server.
- When presenting any user-specific information, the application must use the UserID stored within the security cookie because it is tamper resistant.
- After a set time of inactivity, the session must time-out and force re-authentication. This length of time can be set from one minute to eight hours. The session must use the embedded security cookie time, which is tamper resistant.
- If any external PIN or password is used, it must be able to resist brute force attacks by locking the account after a set number of failed log in attempts. The number of failed log in attempts can be set for three to ten attempts.

Copyright