



Utilisation de Stub en VB.NET

Méthode pour développeur en Visual
Basic

Apprendre ce qu'est un stub et l'utiliser dans ses
application Visual Basic

H4X0R666
04/04/2010

Utilisation de Stub en VB.NET

Sommaire

Présentation	2
Fonctionnement d'un Stub.....	2
Mise en place d'un Stub	4
Etape1 La Partie Soft	4
Etape 2 La Partie Stub	6
Conclusion	7

Présentation

Bonjour, mon nom est H4X0R666 et aujourd'hui nous allons nous consacrer à l'utilisation des stubs en VB.net ,je serai votre guide durant tout ce tutoriel.

Tout d'abord qu'est ce qu'un Stub ?

Beaucoup d'entre vous doivent sans doute ce posez la question, un Stub pour faire simple est un « morceau » de programme qui va nous servir à plusieurs applications comme par exemple :

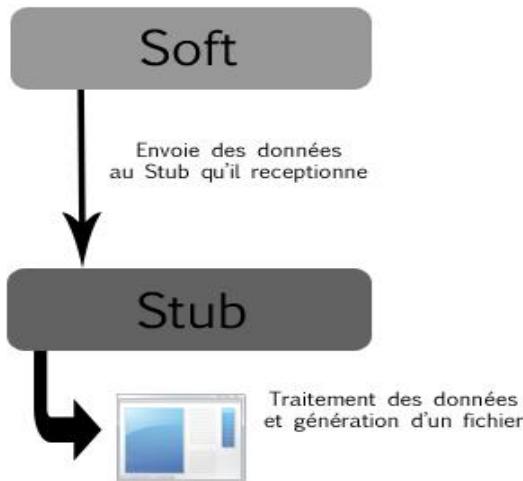
- Crypter un soft afin de le rendre indétectable par certains antivirus
- Binder des fichiers
- Générer des exécutables

Et d'autres applications qui n'a de limite que votre imagination.

Fonctionnement d'un Stub

Un Stub est par définition un morceau de programme, qui dit morceau dit incomplet, un Stub seul ne sert à rien il vient compléter un autre soft qui va envoyer des données au stub et le stub fera ce pourquoi il a été conçu (génération, cryptage,...)

Voici un petit schémas explicatif :



Fonctionnement d'un Stub

Mise en place d'un Stub

Maintenant que vous avez compris le principe du Stub on va s'occuper de la partie développement comme dit précédemment il faut un soft et un stub nous allons nous occuper tout d'abord de la partie Soft.

Etape1 La Partie Soft

La première Etape consiste à réaliser votre soft (Binder , Crypter, Générateur, ...). Ainsi qu'un bouton de construction (un bouton normal qui une fois cliqué générera notre .exe).

Le code du bouton à mettre sur votre bouton de construction ici « button_1 »

```
Private Sub Button1_Click() Handles Button1.Click
End Sub
```

Stockez dans vos variables les données que vous souhaitez envoyer au stub (Je prendrai pour exemple un user contenu dans une textBox1 et un Mdp contenu dans une textBox2)

Ce qui nous donne dans le code :

```
Dim User, Mdp As String
User = TextBox1.Text
Mdp = TextBox2.Text
```

Dans nos déclarations maintenant il faut créer un « FileSplit » un séparateur qui va séparer les données pour ensuite les envoyer au stub dans une chaîne du style :

Donnée1-FileSplit-Donnée2

Le séparateur (FileSplit) est une variable de type String qui doit être **unique** et **commune** au Soft et au Stub.

```
Dim FileSplit As String = "-(SEPARATOR)-"
```

Vous pouvez bien entendu appeler votre variable n'importe comment et y ajouter le contenu que vous voulez du moment qu'il est unique.

Voilà pour ce qui est de la création du FileSplit on va s'en servir plus tard . On va maintenant créer un SaveFileDialog qui va sauvegarder notre fichier généré.

Toujours à la suite du code :

```
'on créer un nouveau "Save File Dialog"
    Dim Sfd As New SaveFileDialog
        'on déclare ici une chaîne pour récupérer le nom du fichier que
        nous allons générer.
        Dim SaveFileName As String
        'On ajoute un filtre ".EXE"
        Sfd.Filter = "*.* | *.exe"
        'On vérifie si l'utilisateur clique sur Ok et on stocke le nom du
        fichier dans la variable sinon on ferme la fenêtre.
        If Sfd.ShowDialog = Windows.Forms.DialogResult.OK Then SaveFileName
= Sfd.FileName
        Else Exit Sub
```

Jusque là rien de bien compliqué nous avons bien instancié notre SaveFileDialog qui va créer notre fichier de sortie ici un exécutable.

Maintenant on va réaliser une instance d'essai autrement dis un « TRY » comme sa si il ya un problème on sait d'où ça vient .Ceci viendra finaliser notre parti Soft voici le code :

Try

```
'Tout d'abord on ouvre le Stub.exe en Mode Binaire et lecture
FileOpen(1, Application.StartupPath & "\Stub.exe",
OpenMode.Binary, _
    OpenAccess.Read, OpenShare.Default)
'On déclare une Chaine Stub pour récupérer les données du Stub
Dim Stub As String = Space(LOF(1))
'On récupère les données du Stub
FileGet(1, Stub)
'On Ferme le Stub
FileClose(1)
'On Ouvre notre exécutable final en Mode Binaire cette fois ci
en Ecriture car on va y placer les données du Stub
FileOpen(2, SaveFileName, OpenMode.Binary, OpenAccess.Write,
OpenShare.Default)
    'Bien, Maintenant on va écrire notre stub dans le fichier de
sortie.
    'On va ajouter les infos que nous avons récupéré du Soft et les
séparer par des « FileSplit »
        'Souvenez vous de l'ordre dans lequel vous introduisez les
        infos c'est vraiment important pour le codage du Stub ! ici Stub,User,Mdp
        FilePut(2, Stub & FileSplit & User & FileSplit & Mdp)
        'On Ferme et Crée notre Fichier de Sortie
        FileClose(2)

        'on affiche un message pour dire que tout c'est bien passé
        MessageBox.Show("Le fichier a été créé!")

        'sinon on récupère les erreurs et on les affiche
        Catch ex As Exception
            'on affiche un message d'erreur en cas de Blème
```

```

        MessageBox.Show("Une erreur est survenue!" & vbCrLf &
ex.ToString)
    'close our "Try - Catch" block
End Try

```

Nous en avons fini avec la partie Soft nous allons maintenant développer la partie Stub.

Etape 2 La Partie Stub

Une bonne chose de faite le soft est terminé on va maintenant réaliser le Stub, pour on va créer un nouveau projet Visual basic dans lequel vous aller créer le modèle de l'application à générer (si vous faites un générateur) sinon pour un crypteur laissez un formulaire vide . dans mon cas j'ai créé une form (Form1) avec deux textbox (TextBox1 et TextBox2) qui affichera le user et mdp envoyé par le Soft.

Dans L'événement « Load » du formulaire principal du Stub on va récupérer les données « envoyés » par le Soft. Voici le Code du Stub :

```

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    'On Instancie le Même FileSplit Que le soft le nom de la variable
    peut changer mais pas le contenu
    Dim FileSplit As String = "-(SEPARATOR)-"

    'On Déclare plusieurs variables de type String le () permet de créer
    plusieurs variables de la forme Option(0), Option(1),... on y stockera les
    données envoyé du soft
    Dim Options() As String

    'On Ouvre le Stub auquel on en envoier des Données par le Soft
    FileOpen(1, Application.ExecutablePath, OpenMode.Binary,
    OpenAccess.Read, OpenShare.Shared)
    'On déclare une Chaine pour récupérer l'ensemble du Fichier
    Dim F As String = Space(LOF(1))
    'On le Stub Dans la Chaine
    FileGet(1, F)
    'une fois les données récupérer on le ferme , on n'a plus besoin de
    lui
    FileClose(1)

    'On Passe maintenant à une étape clef du Stub on va séparer les
    données contenu dans la chaine F qui contient toute les informations
    envoyées au Stub de la forme Stub-FileSplit-User-FileSplit-Mdp on va pour
    cela procéder à une découpe de la chaîne à chaque FileSplit de manière à
    récupérer les données dans les variables Option
    Options = Split(F, FileSplit)

    'Vous avez bien retenu l'ordre dans lequel on a inséré nos infos ?
    'Le Stub est stocké dans Option(0) (on n'en a pas besoin ici)
    Option(1) contient le User et Option(2) Le Mdp alors on les affiche !
    TextBox1.Text = Options(1)
    TextBox2.Text = Options(2)
    'Et Voilà !
End Sub
End Class

```

La Partie Stub est fonctionnelle et terminée la classe hein ?

Conclusion

La mise en place d'un stub peut ne pas être évidente pour certains du à la gestion des fichiers et à la manipulation des données que celui-ci comporte, mais rassurez-vous son utilisation, une fois qu'on a compris, reste la plus simpliste qui soit le seul hic et malheureusement il y a n'a un... c'est que certains Antivirus le détecte (Avira notamment) comme étant un générateur de serveur Rat ... Il existe pour les plus expérimenté des moyens pour contourner ce problème avec des injections en mémoire (Utilisation de RunPE et autre artifice de cryptage) Mais ce n'est pas le but du tuto.

En espérant vous avoir aidé mais surtout fait comprendre la mise en place d'un Stub.