La gestion de la protection dans des applications WinDev

par Emmanuel Lecoester Francis Morel

Date de publication : 09 Septembre 2009

Dernière mise à jour : 09 Mai 2009

I - Les différents niveaux de protection

I-A - Confidentialité des applications et des données

Pour s'assurer de la confidentialité des données, il faut :

- qu'intrinsèquement le contenu des fichiers HyperFileSQL soit inaccessible, par quelque moyen (WDMAP, Editeur hexa...) que se soit,
- que l'application soient suffisamment protégée pour empêcher la découverte du mot de passe utilisé en interne.

I-B - Protection contre les utilisations illicites

Une application protégée contre des utilisations illicites est caractérisée par une utilisation limitée:

- à un mode particulier (démonstration par exemple)
- à un ou plusieurs utilisateurs donnés
- à un nombre d'exécutions spécifiques
- à une date limite d'utilisation
- à une version particulière
- ...

I-C - Intégrité des données

L'intégrité de la base de données, des liaisons... et le respect de ces règles par le développeur, l'utilisateur et l'application elle-même, bien que liée à la sécurité des données, ne fait pas partie de ce document.

II - Confidentialité des applications et des données

Pour constater les évolutions de Windev 7 et suivant un petit retour en arrière sur Windev 5.5 s'impose.

II-A - Historique - Protection des fichiers HF sous Windev 5.5

Bien que l'aide recommande d'utiliser 4 caractères minimum pour les mots de passes, le stockage dans le fichier HF est fait sur 2 octets, et encore sur chacun des ces octets seuls les codes 0x00 à 0x7f (127) sont utilisés. Soit au mieux 16129 possibilités.

Dans ces conditions un algorithme simple en recherche par force brute est quasi-instantané.

La protection des fichiers HyperFileSQL sous Windev 5.5 est donc totalement inopérante. L'utilisation de ces fichiers au format HF5.5 est donc à proscrire pour les données sensibles.

II-B - Historique - Protection des Exécutables

Toujours dans Windev 5.5, un simple éditeur hexadécimal permet de visualiser quantité d'informations textuelles de la bibliothèque. Tous les objets contenus dans la WDL sont clairement identifiables et pour chacun, tous les constituants sont aussi visibles.

C'est d'ailleurs ce qui a permis la création d'outils comme WDL.EXE, qui à partir d'une bibliothèque WDL, reconstituait toutes les fenêtres, images, classes, procédures....

Pour peu que "Supprimer le code source à la compilation" n'ai pas été coché lors de la création, les objets reconstitués contenaient aussi le code source de l'application, sinon seul le Pcode créé était présent.

II-B-1 - Windev 7 et suivantes

Heureusement depuis Windev 7 les choses ont bien changés.

- L'entête des fichiers HF n'est plus limitée en taille et les informations de protection (mots de passe et cryptage), sont stockées cryptées en quasi totalité.
- La bibliothèque WDL est maintenant cryptée/compactée et même s'il était possible de la décrypter, le code source n'est plus jamais inclus. Seules quelques images restent accessibles en natif.Toutefois, dans certaines versions de Windev, l'inclusion des codes nécessaires à l'utilisation de "Etat et Requêtes" laissait, de façon incompréhensible, quelques informations textuelles dans le code de la WDL, voire même une partie de code source de l'initialisation du projet.Un exécutable windev est ainsi comparable à un code obfusqué (impénétrable), même l'utilisation d'un désassembleur ne permet pas la compréhension du code.En fait comme pour un programme obfusqué, la partie principale du code doit d'abord passer par une phase de décompression, de décryptage et d'interprétation avant d'être exécutée. Seule la partie lanceur de Windev est "compréhensible" (en assembleur) dans un débogueur.

II-B-2 - Depuis WinDev 11

On peut ainsi, en première approche, raisonnablement penser que les applications Windev et les fichiers HyperFileSQL 11, 12 14 sont protégés efficacement, à condition de respecter certaines règles, objet de ce document.

II-C - Protection des fichiers HF avec WinDev

Actuellement il est possible de définir 3 types de protection dans HyperFileSQL

- Mot de passe sur l'analyse
- Mot de passe sur les fichiers HyperfileSQL
- Cryptage des fichiers HyperFileSQL

II-C-1 - Mot de passe sur l'analyse

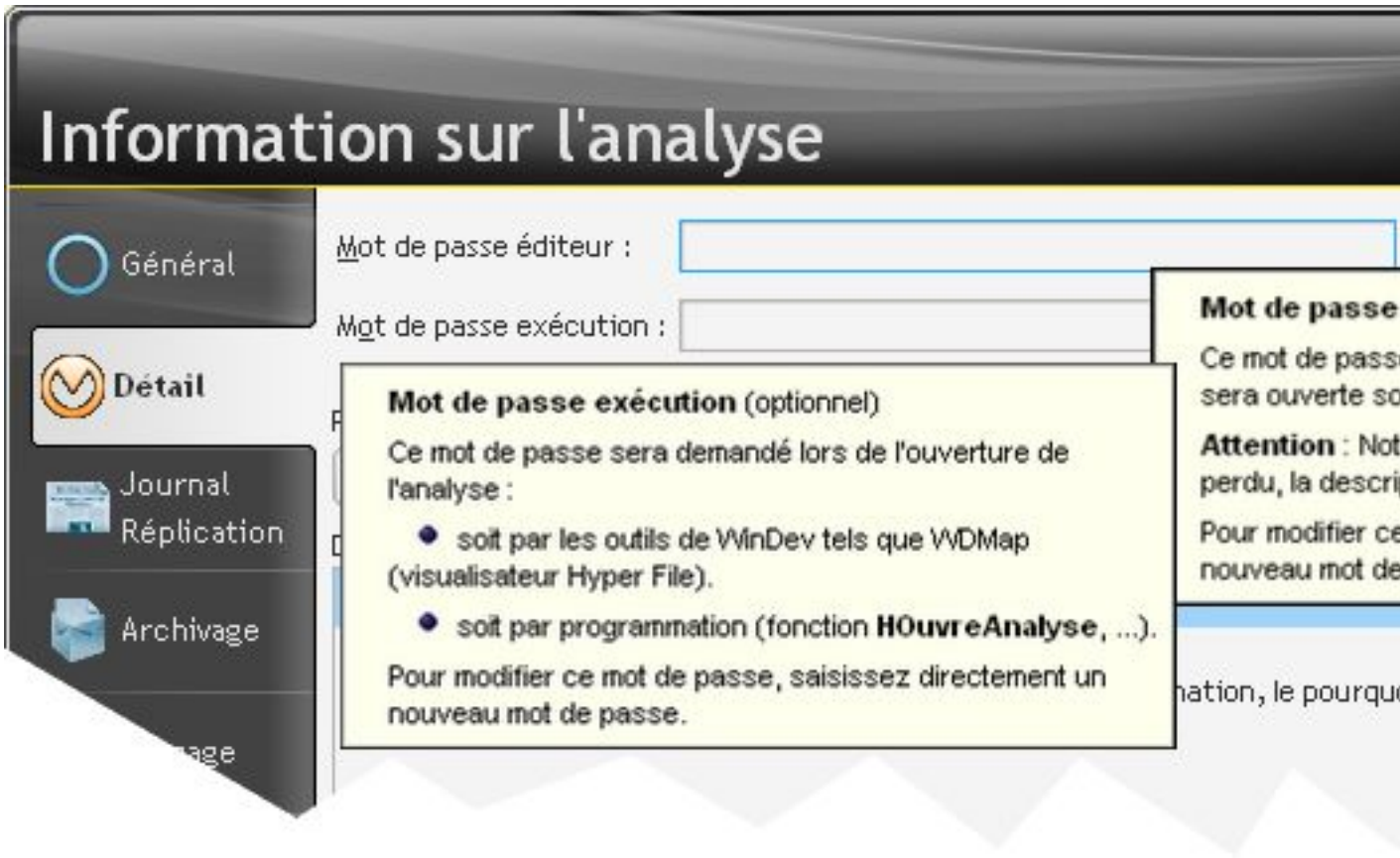
Cette protection à un double intérêt :

- Eviter toute modification de l'analyse par une personne non autorisée (mot de passe en édition)
- Empêcher son utilisation dans une application non autorisée (mot de passe en exécution)

Par contre du fait de **l'ouverture automatique** de l'analyse associée à une application, elle n'a aucun intérêt pour assurer la confidentialité des données sans précaution supplémentaire.

Il faudrait ainsi utiliser HfermeAnalyse() ou équivalent puis HouvreAnalyse() dans le projet.

La protection de l'analyse par un mot de passe en exécution peut être fait directement lors de la création (étape 2) ou après création dans l'onglet "Détail" de sa description.



Pour éviter les demandes systématiques à chaque lancement du mode test ou lors de la génération de l'exécutable, le mot de passe en exécution peut être renseigné dans l'onglet "Analyse" de la description du projet.

En remplissant ce champ le mot de passe de l'analyse ne sera plus demandé ni en mode test, ni EN MODE EXECUTION. Il est écrit crypté dans l'exécutable.

II-C-1-a - Synthèse : Mettre un mot de passe sur l'analyse

- Projet > Charger l'analyse
- Analyse > Description de l'analyse... > Onglet Détail
- Mot de passe éditeur (ouverture de l'analyse sous éditeur)
- Mot de passe exécution (ouverture de l'analyse en exécution avec HOuvreAnalyse ou WDMAP)
- Projet > Description du projet > Onglet Analyse
- Mot de passe (aucun mot de passe ne sera alors demandé lors de l'exécution)

II-C-2 - Mot de passe sur les fichiers HyperFileSQL

Le mot de passe d'un fichier HyperFileSQL est défini dans le code par programmation avec les fonctions HPasse(), HCréation() et HcréationSiInexistant().Il n'est pas possible de définir directement un mot de passe dans l'éditeur pour un nouveau fichier.

Il n'est pas modifiable par programmation, pour un fichier existant

Le mot de passe est demandé lors de la modification automatique (par WdModfic) ou lors de l'ouverture par WDMAP.

Utilisé seul (sans cryptage) le mot de passe d'un fichier n'est d'aucun intérêt vis-à-vis d'une protection des données. En effet le contenu en clair est visualisable avec un simple éditeur hexadécimal.

II-C-3 - Cryptage des fichiers HF

- Le cryptage des fichiers HyperFileSQL est défini dans l'éditeur d'analyse, pour chaque fichier dans l'onglet détail de la description du fichier lui-même.

Ce cryptage peut être spécifié sur les données elles-mêmes, sur les index et sur les fichiers de mémo.

Trois algorithmes en 128 bits sont proposés :

- Cryptage rapide optimisé sur 128 bits
- Cryptage RC5 sur 12 boucles
- Cryptage RC5 sur 1- boucles

Le cryptage obtenu est natif, indépendant du mot de passe (voir aide HPasse).

Aucune clé n'est demandée pour effectuée ce cryptage (probablement inscrite dans l'entête).

Comme pour le mot de passe, utilisé seul ce cryptage n'est que d'un intérêt limité, le fichier est effectivement crypté mais visible dans WDMAP (ou tout appli WD pouvant ouvrir le fichier).

Il est donc indispensable pour assurer la confidentialité des données **d'activer le cryptage** des données, des index et mémo **ET** d'ouvrir le fichier à l'aide d'un **mot de passe**.

Pour permettre la modification automatique des fichiers ainsi protégés, il faut aussi cocher "Activer la sécurité renforcée".

II-C-3-a - Synthèse : Protéger un fichier HyperFileSQL

1 Projet > Charger l'analyse
2 Structure de fichiers > Description > Onglet Détail
3 Sélectionner le fichier à protéger
4 Protection des données, activer le Cryptage (128 bits, RC5 12 boucles ou RC5 16 boucles)
5 Cocher activer la sécurité renforcée
6 Dans le code (de préférence pas le code du projet) ouvrir les fichiers HyperFileSQL par
7 HcréationSiInexistant("*","motdepasse") ou Hcréation("*","motdepasse") ouutiliser Hpasse("*","motdepasse")
8 Analyse > Génération pour régénérer l'analyse si nécessaire

II-C-3-b - Conseil : Choix des mots de passe

- Utiliser au minimum 8 caractères.
- Utiliser des lettres (mots de passe insensibles à la casse) ET chiffres ET caractères spéciaux.Le mieux consiste même à utiliser des codes non imprimables.
- Eviter les mots courants issus des dictionnaires, préférer une mini phrase.
- Construire le mot de passe, plutôt que d'utiliser une chaine constante.En effet après l'interprétation du Pcode les chaines sont visibles clairement à l'aide d'outils d'exploration de la mémoire.

II-C-4 - Modification des mots de passe HyperFileSQL

Il n'existe pas de possibilité directe de modification du mot de passe d'un fichier HF que celui-ci ait été mis dans le code ou modifié préalablement lors de la modification automatique du fichier.

Dès que le fichier est crée, pour modifier ou supprimer le mot de passe il faut :

- soit changer le mot de passe lors de la modification automatique des fichiers (comme indiqué dans l'aide à HPasse)
- soit utiliser le composant de la LST 64 "WD ChangeMotDePasse"
- soit écrire une procédure interne ou un composant, qui à l'aide des fonctions HAlias et HCopieEnregistrement effectue le transfert d'un fichier à un autre

II-C-4-a - Synthèse : Modification du mot de passe par WDOUTILS

1 Projet > Charger l'analyse
2 Structure de fichiers > Rubriques
3 Modifier la taille d'une rubrique existante ou ajouter une rubrique bidon
4 Analyse > Génération pour régénérer l'analyse
5 Lors de la modification Automatique HyperFileSQL (par WDOUTILS)
- Cocher (ou ajouter si besoin) l'emplacement de recherche des fichiers à modifier
- Cocher les fichiers à modifier dans la liste proposée
- Valider éventuellement la sauvegarde proposée
- Indiquer pour chaque fichier l'ancien mot de passe
- Cocher "Je veux saisir ou changer les mots de passe"
- Cocher chaque fichier et indiquer le nouveau mot de passe à utiliser
6 ValiderPenser à changer le mot de passe dans votre code

III - Vulnérabilité des données et applications

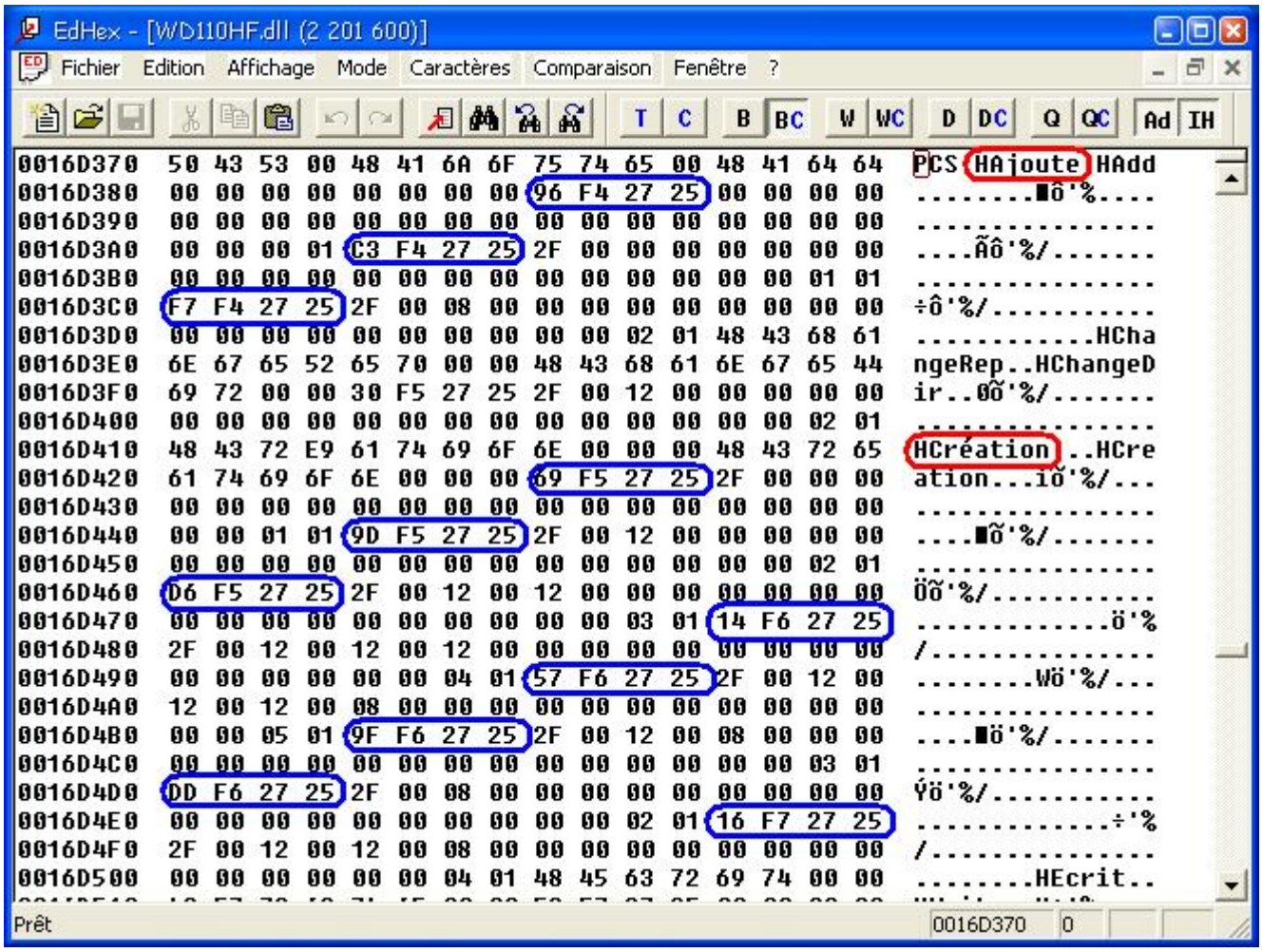
Intrinsèquement les fichiers HyperFileSQL peuvent résister correctement à une attaque par force brute, et disposent ainsi à première vue, d'une sécurité satisfaisante.

Le code généré constitué par la bibliothèque WDL compressé et crypté est difficilement interprétable, même avec un désassembleur-débogueur.

Le point faible reste cependant les DLL associées indispensables au bon fonctionnement de l'application.

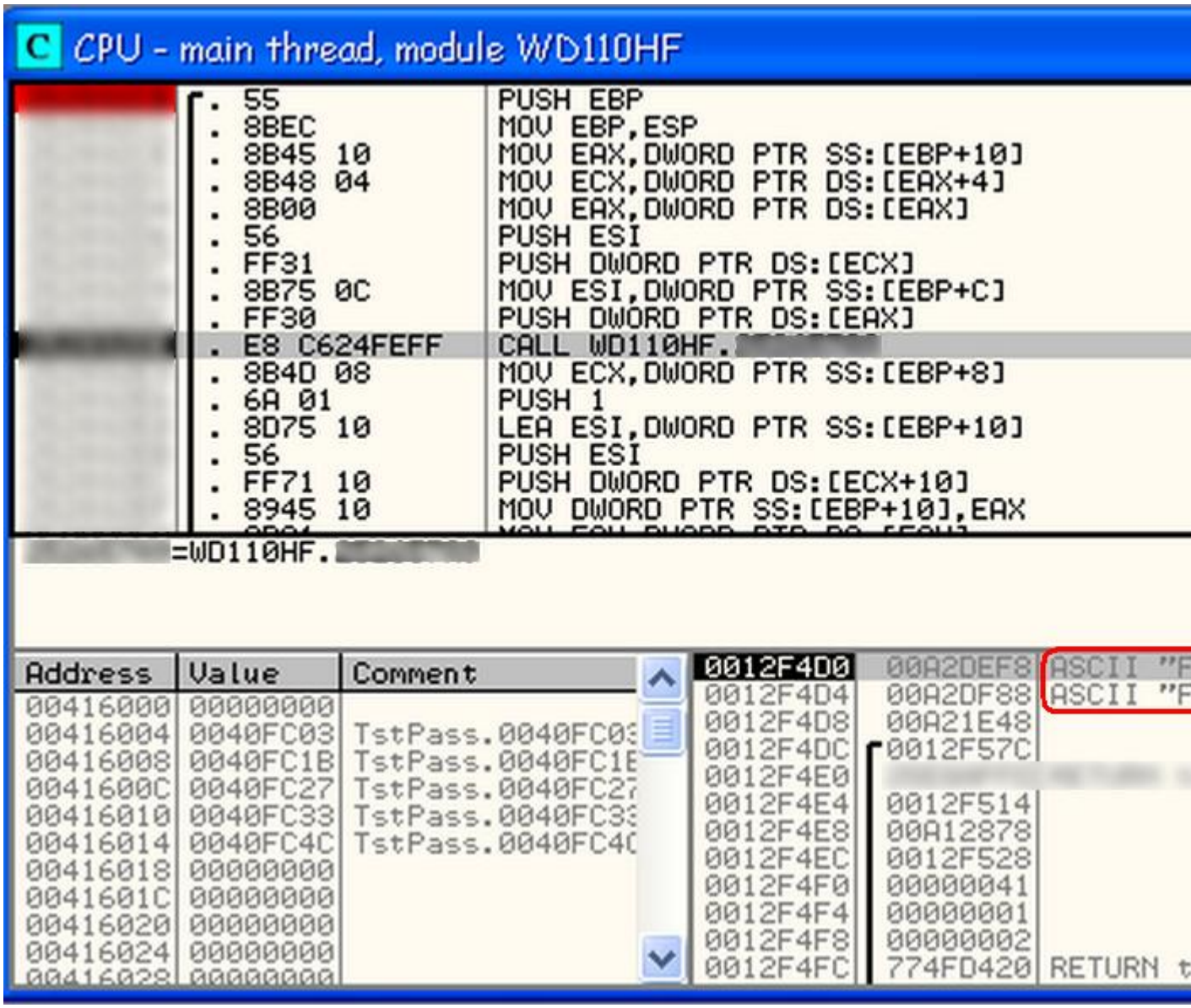
III-A - Les DLL de Windev

L'examen de ces dll, avec un simple éditeur hexadécimal révèle quantités d'informations textuelles interprétables susceptibles d'être utilisées pour attaquer l'application.



En particulier, comme l'illustre l'image précédente, l'identification des fonctions et des points d'entrée est rendue aisée par la proximité des noms des fonctions et des adresses mémoires correspondantes.

Muni de ces informations et d'un simple débogueur il est relativement aisé d'obtenir les paramètres des fonctions de composante (les fonctions du WLangage). En particulier, illustré ici sur la fonction HPasse() un point d'arrêt placé au bon endroit donne accès immédiatement au mot de passe du fichier (ici "FMLPASSWD" pour le fichier "FichPass3") quel qu'en soit la complexité.



Il existe même aujourd'hui, au moins une application "Hyper File Password Recovery Tool" (construite avec Windev) qui automatise tout ce processus et permet de retrouver les mots de passe perdus à condition de disposer de l'application.

III-B - Le point faible : les DLL de Windev

Comme constat de cette analyse, on s'aperçoit que quelle que soit la base de données utilisée, pour casser la protection (mot de passe, cryptage...) d'une application Windev et accéder ainsi aux données sensibles il suffit :

- d'avoir accès aux fichiers de la base de données
- d'avoir le fichier EXE ou WDL qui réalise le traitement à intercepter
- que le mot de passe (ou code de cryptage) soit statique (indépendant du Pc, de l'utilisateur)
- que l'exécutable ne soit pas protégé du débogage
- d'avoir la possibilité d'utiliser une session de débogage

Les Dll de Windev 14 semblent déjà un peu mieux protégées et ne permettent pas, simplement, le fonctionnement en mode pas à pas des décompilateurs / débogueurs.

Toutes les solutions qui permettent de supprimer une ou plusieurs de ces conditions contribueront à la protection des données sensibles de l'application.

III-C - La suppression des points faibles

Ainsi parmi les solutions disponibles pour assurer la confidentialité des données on pourra donc :

- utiliser une application client/serveur ou les fichiers HyperFileSQL sont dans des dossiers inaccessibles, protégés en lecture,
- protéger l'exécutable contre les utilisations non autorisées,
- utiliser des mots de passe dynamiques construits à partir d'informations sûres comme les caractéristiques du PC, de Windows, des disques, de l'adresse MAC...
- intégrer un dispositif anti-débogage dans l'exécutable, pour empêcher l'utilisation conjointe d'un débogueur et le lancement par d'autres modules que Windows lui-même.

Pour être efficaces ces solutions doivent faire l'objet d'une attention toute particulière et ne peuvent aisément être décrites ici.

Aussi le chapitre suivant présente la solution anti-débogage DataProtect.

III-D - Composant DataProtect

Ce composant, intégrable simplement dans les applications Windev 11, 12 ou 14 a pour principal but de restreindre (ou rendre plus difficile) les sessions de débogage, interdisant ainsi :

- le lancement de l'application protégée par un autre module exécutable que Windows
- la pose de point d'arrêt
- la consultation de la mémoire durant le fonctionnement de l'application

Il a été écrit pour ne consommer que des ressources CPU limitées

- 30 ms en mode rapide
- 300 ms en mode complet

Il dispose aussi d'un générateur de mots de passe dynamiques pour construire une chaine à partir des caractéristiques du serveur, de l'utilisateur, des disques...

Il est disponible en téléchargement sur **http://www.SoftProtect.fr**

III-D-1 - II-D-1. Synthèse : Mise en oeuvre de DataProtect 1.14

1 Télécharger DataProtect (composant, documentation et exemple) et dezipper
2 Dans le projet Windev à protéger : Atelier > Composant > Importer un composant dans le projet > A partir d'un fichier...Sélectionner DataProtect.wdi et valider
3 Dans l'éditeur, Entourer chaque opération critique par le code suivant :

```
SI iSecure(8)=0 ALORS
// Code critique à protéger

FIN
```

4 Au besoin créer vos mots de passe dynamiques par le code suivant :

```
sPass est une chaîne = sGetPassword()
```

5 Créer l'exécutable en intégrant le composant Déployer
6 D'autres codes plus complets sont disponibles dans l'aide du composant