Webh4ck partie III

Ce document est la troisième partie de l'étude sur les mécanismes de traversée automatique de sites. On a vu quels étaient les composants fondamentaux sous-jacents à la navigation sur le web, les moyens de récupérer des paramètres invisibles, puis d'automatiser d'une manière ou d'une autre leur utilisation à des fins diverses.

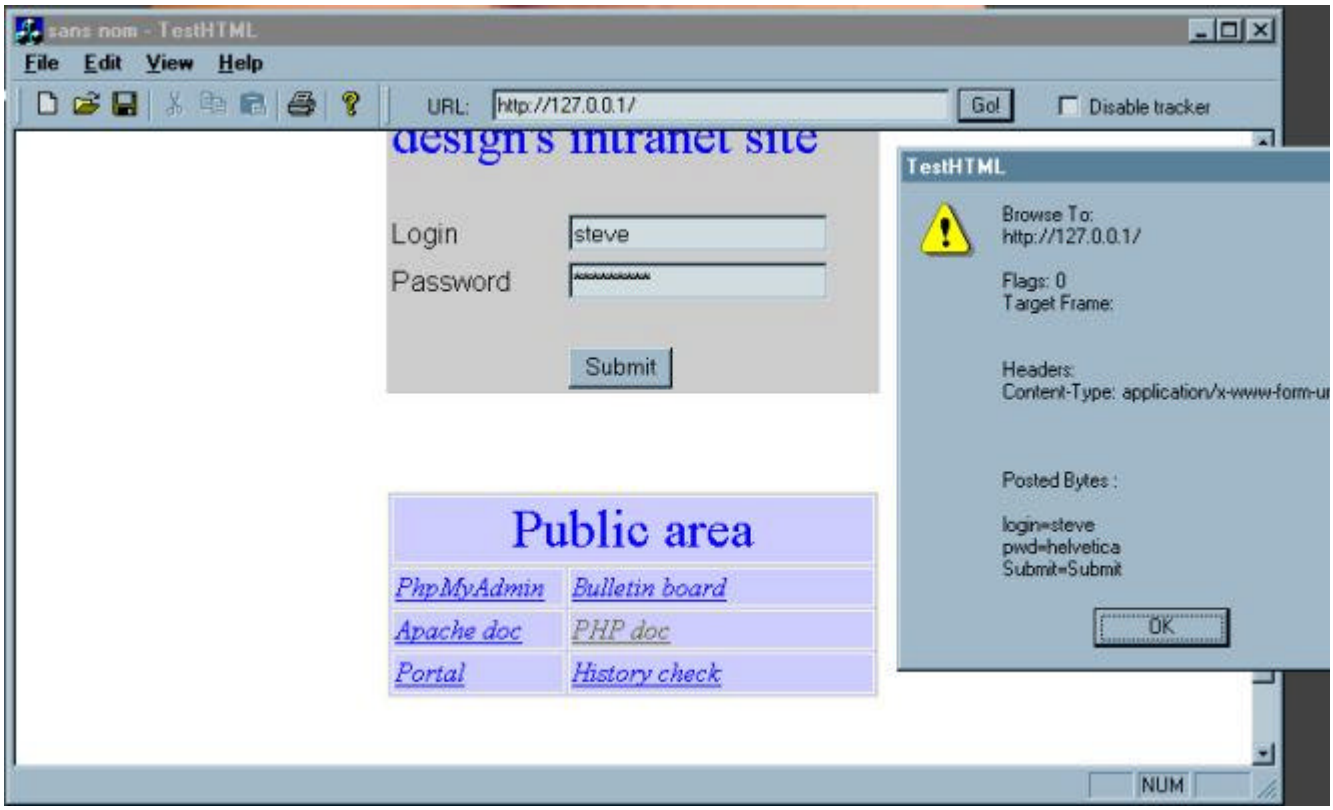
Cette partie détaille la mise en œuvre d'un outil d'analyse très puissant. Fourni gratuitement, cet outil permet de récupérer à chaque clic souris l'ensemble des entêtes visibles ou invisibles qui passent sur la ligne entre le navigateur web et le serveur web. A l'aide d'un tel outil il devient possible de comprendre la structure complète d'un site, sans avoir à déployer des techniques complexes.

Cet outil est basé sur le fait qu'il est possible de créer un **hook** sur le navigateur Internet Explorer. De manière logicielle, l'outil s'enregistre pour un événement particulier, en fait le clic souris sur un lien hypertexte ou la validation d'un formulaire. Et le navigateur web, qui maintient une liste d'applications s'ayant enregistré, notifie chacune d'entre elles au moment opportun, en transmettant en paramètre les fameuses entêtes GET et POST.

Bref, à l'aide d'un tel outil, il devient trivial de savoir de manière parfaitement fiable ce qui passe sur la ligne. Et cet outil est parfaitement transparent, il ne génère aucun biais ou aucune différence de navigation, ce qui n'aurait probablement pas été le cas de filtres par exemple.

1. Un grand coup d'œil sur ce logiciel

Une capture d'écran vaut mieux qu'un long discours :



L'outil de tracking d'entêtes GET et POST lors de son utilisation

A côté d'une page web principale (en l'occurrence une page locale fournie sur un serveur personnel Apache/PHP), on distingue une boite de dialogue modale. Il suffit de cliquer sur OK pour la quitter.

La boite de dialogue modale présente les informations de la façon suivante :

- BrowseTo : URL complète (avec entêtes GET en particulier)
- Target : cadre dans lequel la page va s'afficher (pertinent pour un site comportant plusieurs cadres)
- Headers : certaines entêtes invisibles
- Posted Bytes : entêtes POST

Dans le cas de figure, j'ai tapé mon login et mon mot de passe dans la bannière d'accès privé au site. Dans la page web, pour des raisons de sécurité, le navigateur affiche des signes * en lieu et place du mot de passe. Mais on voit bien dans les paramètres POST que le mot de passe n'est pas crypté. Et d'ailleurs on le récupère en clair, ici helvetica.

L'utilisation de cette technique permet de tester un site avant mise en ligne. Et s'assurer que les données de formulaire ne sont pas traitées ou filtrées.

Mais l'utilisation principale est la récupération des noms des champs du formulaire. Ici je sais que le formulaire est composé de trois champs : **login**, **pwd** et **Submit**.

(En réalité, Submit n'est autre que le bouton d'envoi des données du formulaire. Le mécanisme des formulaires sur le web fait que celui-ci est envoyé au même titre que les vrais champs de contenu. Comme des sites sont susceptibles de vérifier l'existence du champ Submit, et même de la valeur associée, il est absolument nécessaire de traiter ce champ comme si c'était un champ soumis par l'utilisateur, même si en l'occurrence il s'agit d'un champ à valeur statique.)

Comme on le voit, sans effort, sans afficher le fichier source et décortiquer toutes les arcanes de HTML, y compris le passage par des pré-traitements en Javascript (de plus en plus fréquents), l'outil m'a révélé l'essentiel. En vertu de la modélisation de données vue dans la première partie de cette étude, je peux ajouter trois nouveaux enregistrements dans la base de données et dorénavant passer à travers sans plus jamais taper mon login et mon mot de passe.

De plus, comme ces données sont hébergées sur une base de données potentiellement mise sur Internet, je ne suis pas contraint à utiliser ma machine. Je peux partir en mission avec un ordinateur portable, vide avec juste une connection Internet, et accéder aux fonctionnalités Webh4ck sans problème. C'est un des intérêts du Web : la collaboration et le travail en mode nomade.

2. Comment ça marche ?

Internet Explorer est une application construite autour d'une API web, une interface de programmation capable de se connecter sur un site, et d'aller chercher des pages web. En fait, une API web est très simple. Un intérêt primordial pour le développeur est que l'API web génère des évènements particuliers lors d'un clic. Si tel est le cas, il est possible d'une manière ou d'une autre de récupérer ces évènements et de déclencher notre propre application.

L'événement pertinent ici est *OnBeforeNavigate()*. Cet événement est déclenché avant tout chargement de nouvelle page web. D'ailleurs ce point de passage permet d'interdire le chargement de la dite page, si nécessaire.

L'API web à ceci d'élégant qu'elle est intégrée dans une classe MFC Windows (cf MSDN Library) et qu'il suffit de surcharger la méthode *OnBeforeNavigate()* pour se "plugger" dans la navigation web.

Pour des raisons d'évolution, l'API web a évolué et est passée de IWebBrowser à IWebBrowser2, ce qui fait qu'aujourd'hui avec les MFC en version 6.0 on implémente en fait plutôt *OnBeforeNavigate2()*. Mais le principe est inchangé.

Très techniquement, les évènements sont des points de connection dispatch définis dans DWebBrowser (cf MSDN Library).

3. Je le veux !

J'ai trouvé la base du logiciel sur un site de développeurs. Le code source associé, moins de 100 lignes de code, permet de valider l'entrée de données dans des formulaires. Mais j'ai pensé qu'il convenait parfaitement à d'autres utilisations.

Aussi ai-je rajouté une barre de navigation URL pour simuler un navigateur web à proprement parler. Le bouton Go! permet d'aller sur le site spécifié. La case à cocher Enable/Disable permet de désactiver le mode tracking. Lorsque le mode tracking est désactivé finalement tout se passe comme si ce logiciel était votre navigateur web.

Pour aller en arrière, je n'ai pas mis de bouton avec une flèche vers la gauche, car il suffit d'appuyer sur la touche Backspace.

Le logiciel suppose qu'Internet Explorer est installé, en version 4 au minimum. Ce qui n'est pas une contrainte en pratique.

La page de démarrage sur le logiciel n'est autre qu'une page d'exemple de la personne à l'origine de celui-ci : Ted Crow. Il revient à cette personne l'essentiel de la paternité de celui-ci, sachant qu'il s'appuit sur IWebBrowser, une interface de programmation fournie par Microsoft.

Pour télécharger le logiciel, cliquer ici.