TP 1 – Authentification HTTP et ses faiblesses

Les usagers d'Internet sont confrontés à un nombre croissant d'applications web nécessitant de s'authentifier. Cette authentification permet de s'assurer que l'utilisateur est bien la personne qu'il prétend être mais aussi de lui fournir des informations personnalisées en fonction de son profil et des données personnelles qu'il transmet. Elle est donc sujette à de nombreuses attaques de la part de personnes malveillantes qui souhaitent en exploiter toutes ses vulnérabilités. L'exploitation de ces failles peut être utilisée pour compromettre les services d'authentification web.

Authentification HTTP L'authentification HTTP permet de s'identifier auprès d'un serveur HTTP à l'aide d'un nom d'utilisateur et d'un mot de passe. Il existe deux méthodes : la méthode Basic et la méthode Digest (dont on verra deux différentes variantes).

Préparation

Nous allons avoir besoin d'un serveur web, auquel nous allons envoyer des requêtes. Pour simuler cette situation, nous allons utiliser une machine virtuelle qui communique avec notre machine comme deux machines distantes. Une machine sera utilisée pour lancer un serveur web ; la seconde pour se connecter à ce serveur et étudier les communications entre la page et le serveur. Afin de pouvoir étudier les communications entre notre nos deux machines, nous allons utiliser Wireshark.
Vous trouverez une guide pour installer votre MV à ce lien : `http://pageperso.lif.univ-mrs.fr/~emmanuel.godard/ens/reseaux/01_decouverte/`
L'adresse du fichier .box que vous devez télécharger est la suivante : `http://pageperso.lif.univ-mrs.fr/~emmanuel.godard/tpsir/apache-gui.box`
L'adresse du fichier Vagrantfile que vous devez télécharger est la suivante : `http://pageperso.lif.univ-mrs.fr/~emmanuel.godard/tpsir/Vagrantfile`
Enfin, plutôt que d'appeller votre box "debian-tp" dans la commande $vagrant\ box\ add$ comme dans le tutoriel, entrez le nom "apache".
Une fois installé votre MV, lancez là avec $vagrant\ up$ et lancez Wireshark à l'intérieur de la box, cliquez sur eth0 et cliquez sur Start. Puis ouvrez votre navigateur internet à l'extérieur de la box en navigation privée et entrez l'adresse `localhost:8080`. Vérifiez à l'aide de Wireshark, en allant sur votre MV, un premier flux de paquets entre le serveur et votre explorateur internet.
Sur le site web affiché vous allez trouver trois liens avec différent modes d'authentification. Dans tout les cas le login est *albert* et le mot de passe est *einstein*. On etudiera et on commentera les différences et les faiblesses dans chacun de ces trois cas.

Exercice 1. *HTTP sans authentification*
Dans cet exercise préliminaire on décrira et on commentera l'access sans authentification, essayons de mettre en evidence dans ce cas ce que vous connaissez du protocole HTTP. En vous connectant à l'host locale vous allez trouver trois repertoires sans authentification : (i) un répertoire "nopass" (listé dans l'index.html) ; (ii) un répertoire "secret" (non listé dans l'index.html) et en fin un répertoire "P7cG3ssXmRH4CYfN" (non listé dans l'index.html).
Commentez ce que vous voyez sur Wireshark.

1. Quelles différences de protection ont-ils ces trois répertoires ? Quelle consideration pourriez-vous faire sur leur niveau de securité pour chacun de ces cas ?

2. À la lumière de vos different essaies, quel attaque pouvez vous imaginer dans les deux derniers cas ?

Exercice 2. *Authentification HTTP Basic*

L'utilisateur doit fournir un nom d'utilisateur et un mot de passe pour s'authentifier. Le nom d'utilisateur et le mot de passe sont concaténés avec deux points et le tout est encodé en base 64. Il est donc très facile de décoder les données et d'obtenir les informations d'identification. Accédez avec votre identifiant et mot de passe. Commentez ce que vous voyez sur Wireshark.

1. Qu'est-ce qui se passe si vous utilisez un mot de passe erronée ?
2. Imaginez un type d'attaque. Quelles sont à votre avis les faiblesses de ce système d'authenfication ?

Exercice 3. *Authentification HTTP Digest simplifié*

L'authentification Digest a été conçue comme une amélioration de l'authentification HTTP de base. L'une des principales améliorations est que les données ne sont pas transmises en clair mais sont transmises à l'aide d'un message digeste chiffré. Si cette méthode est moins vulnérable aux attaques par écoute, elle le reste encore aux attaques par rejeu. En effet, si un attaquant est en mesure de rejouer le message digeste chiffré alors le serveur lui donnera accés. On supposera ici que le nonce fournit par le serveur ne contienne pas un timestamp et reste toujours valide.
Toutefois, il arrive que le nonce fournit par le serveur contienne un timestamp. Ceci permet au serveur d'en vérifier la valeur lors de l'authentification : si la valeur du nonce est dépassée, alors la demande du client est rejetée.
Accédez avec votre identifiant et mot de passe. Commentez ce que vous voyez sur Wireshark.

1. Imaginez un type d'attaque en utilisant le nounce que vous pouvez arriver à lire sur Wireshark. Quelles sont à votre avis les faiblesses de ce système d'authenfication ? Indiquez des solutions possibles pour maitriser ces faiblesse et optimiser le niveau de securité du SI.

Exercice 4. *Authentification HTTP Digest avec timestamp*

Imaginez maintenant que le nonce fournit par le serveur contienne un timestamp. Ceci permet au serveur d'en vérifier la valeur lors de l'authentification : si la valeur du nonce est dépassée, alors la demande du client est rejetée.

1. Pouvez vous utiliser le meme attaque que dans l'exercise precedent ? Pourquoi ? Immaginez un type d'attaque possible.