sandbox

Une **sandbox** est un environnement isolé et sécurisé dans lequel un programme peut s'exécuter sans affecter le reste du système. Lorsqu'un programme s'exécute dans une **sandbox**, il est restreint dans les actions qu'il peut effectuer, afin de prévenir les problèmes de sécurité.

> Pourquoi ?

Les technologies de **sandboxing** existent depuis plusieurs décennies et ont été utilisées dans divers contextes, notamment pour isoler des applications sur des systèmes d'exploitation en temps partagé.

Cependant, les **sandboxes** modernes utilisent des techniques plus avancées pour créer des environnements de **sandbox** sécurisés et faciles à utiliser. Le concept de **sandboxing** est devenu plus important avec l'augmentation des menaces de sécurité sur les ordinateurs personnels et les serveurs, ainsi qu'avec l'essor des technologies de virtualisation et de conteneurisation.

La popularité des technologies de **sandboxing** a également augmenté avec l'avènement des smartphones et des tablettes, qui ont introduit de nouveaux risques de sécurité. Les **sandboxes** sont maintenant couramment utilisées dans les environnements mobiles pour protéger les données personnelles des utilisateurs et empêcher les applications malveillantes d'accéder à des ressources sensibles, telles que les contacts, les messages ou les photos.

> Principe de fonctionnement

Le mécanisme de **sandbox** repose sur l'utilisation de mécanismes de virtualisation, d'isolation et de contrôle d'accès.

En général, les **sandboxes** sont mises en place en utilisant des techniques de virtualisation pour créer un environnement isolé et sécurisé dans lequel une application peut s'exécuter. Les machines virtuelles ou les conteneurs sont des exemples de technologies de virtualisation qui peuvent être utilisées pour créer des **sandboxes**.

Une fois que l'environnement de **sandbox** est créé, des mécanismes d'isolation sont utilisés pour restreindre l'accès de l'application à certaines ressources système, telles que les fichiers, les entrées/sorties réseau ou les processus. Cela permet d'empêcher l'application d'interagir avec le système hôte et de causer des dommages ou des compromissions de sécurité.

Enfin, des mécanismes de contrôle d'accès sont mis en place pour autoriser ou refuser l'accès de l'application à certaines ressources du système en fonction de son niveau de privilège et des permissions qui lui ont été accordées.

Dans l'ensemble, les mécanismes de virtualisation, d'isolation et de contrôle d'accès permettent aux **sandboxes** de créer des environnements de sécurité isolés pour les applications. Cela réduit les risques de sécurité en limitant les actions qu'une application peut effectuer sur le système hôte.

En général, la mise en place d'une **sandbox** nécessite un certain niveau de configuration pour autoriser l'accès aux ressources du système que l'application a besoin d'utiliser. Lors de la création d'une **sandbox**, il est souvent nécessaire de spécifier les permissions que l'application sera autorisée à utiliser.

Par exemple, si une application a besoin d'accéder à des fichiers sur le système de fichiers local, il faudra autoriser explicitement l'application à accéder à ces fichiers en définissant les bonnes permissions. De même, si l'application a besoin d'accéder à Internet, il faudra spécifier les permissions nécessaires pour permettre à l'application d'utiliser le réseau.

Cependant, la configuration d'une **sandbox** est généralement simplifiée autant que possible pour éviter les erreurs de configuration qui pourraient compromettre la sécurité. Les mécanismes de **sandboxing** modernes, tels que ceux utilisés dans **Flatpak**, sont conçus pour simplifier autant que possible la configuration de la **sandbox** tout en garantissant un niveau élevé de sécurité. Les utilisateurs n'ont souvent pas besoin de configurer manuellement les permissions de la **sandbox**, car celles-ci sont gérées automatiquement par le système de **sandboxing**.

> Technologie

Il existe plusieurs autres technologies de **sandboxing** disponibles, en plus de **Flatpak**, qui sont utilisées pour isoler les applications et renforcer la sécurité sur les systèmes d'exploitation.

Voici quelques exemples de technologies de **sandboxing** populaires :

- Docker : une plateforme de conteneurs qui utilise des mécanismes de virtualisation pour isoler les applications dans des environnements de conteneurs. Les conteneurs Docker offrent une isolation de processus et de réseau, ainsi que des mécanismes de contrôle d'accès pour limiter l'accès aux ressources du système.
- Firejail : un outil de **sandboxing** pour Linux qui utilise des mécanismes d'isolation de processus pour limiter les actions qu'une application peut effectuer sur le système. Firejail est conçu pour être facile à utiliser et propose une interface en ligne de commande simple pour configurer la **sandbox**.
- AppArmor et SELinux : des outils de contrôle d'accès obligatoire (MAC) pour Linux qui permettent de restreindre les actions qu'une application peut effectuer en fonction de ses permissions et de son niveau de privilège. Ces outils sont utilisés pour limiter l'accès aux ressources du système et prévenir les attaques.

Ces technologies de **sandboxing** et bien d'autres sont utilisées pour renforcer la sécurité sur les systèmes d'exploitation en isolant les applications et en restreignant l'accès aux ressources du système.

> Firefox

Firefox, le navigateur web populaire, utilise également un mécanisme de **sandboxing** pour améliorer la sécurité. Le **sandboxing** de Firefox, appelé `Content Process Sandbox`, isole le contenu web dans des processus distincts qui sont séparés du processus principal du navigateur.

Lorsqu'un utilisateur visite un site web, le contenu web (HTML, JavaScript, etc.) est exécuté dans un processus distinct qui est limité dans les actions qu'il peut effectuer sur le système. Si un contenu malveillant ou un code malicieux est exécuté sur le site web, il ne pourra pas affecter le processus

principal du navigateur, ce qui peut protéger l'utilisateur contre les attaques de type `drive-by` ou `drive-by-download` (attaque par téléchargement automatique de logiciel malveillant à l'insu de l'utilisateur).

De plus, **Firefox** utilise également une fonctionnalité appelée `Strict Site Isolation` qui isole les processus de contenu pour chaque site web visité. Cela empêche les sites web de partager des informations entre eux, même si un site malveillant est capable d'exécuter du code dans le processus du navigateur.

En somme, le **sandboxing** de Firefox est un élément important de la sécurité du navigateur, car il permet d'isoler le contenu web dans des processus distincts pour empêcher les attaques de se propager à travers le navigateur ou sur le système.

> ... et bien d'autres

De nombreux programmes et systèmes d'exploitation utilisent le mécanisme de **sandboxing** pour renforcer la sécurité. Voici quelques exemples d'autres programmes et systèmes d'exploitation qui utilisent le mécanisme de **sandboxing** :

- Google Chrome : le navigateur web de Google utilise également un mécanisme de sandboxing pour isoler les onglets dans des processus distincts et limiter les actions qu'ils peuvent effectuer sur le système.
- Microsoft Office : les versions les plus récentes de Microsoft Office utilisent un mécanisme de **sandboxing** appelé “Protected View” pour ouvrir les fichiers téléchargés à partir d'Internet ou d'autres sources non fiables dans un environnement isolé et sécurisé.
- Adobe Reader : le lecteur de PDF d'Adobe utilise un mécanisme de **sandboxing** pour isoler le processus de lecture de PDF dans un environnement de **sandbox** distinct et limiter les actions qu'il peut effectuer sur le système.
- Android : le système d'exploitation mobile d'Android utilise un mécanisme de **sandboxing** pour isoler les applications dans des environnements distincts et limiter l'accès des applications aux ressources du système.
- iOS : le système d'exploitation mobile d'Apple utilise également un mécanisme de **sandboxing** pour isoler les applications dans des environnements de **sandbox** distincts et empêcher les applications de partager des informations entre elles.

En somme, de nombreux programmes et systèmes d'exploitation utilisent le mécanisme de **sandboxing** pour renforcer la sécurité en isolant les processus dans des environnements de **sandbox** distincts et en limitant l'accès des processus aux ressources du système.