Poisoning Attack

* Les attaques par empoisonnement sont des attaques où le but est d’insérer des données ou des mises à jour malveillantes dans le processus d’entraînement ou dans l’agrégation des mises à jour afin de dégrader la qualité du modèle global. À la différence des attaques traditionnelles visant directement le serveur central ou le modèle, ces attaques ciblent les clients participants au Federated Learning (FL).
* On distingue deux types d’attaques :
* **Data Poisoning Attack (DPA)** : l’attaquant modifie les données locales d’un ou plusieurs clients.
* **Model Poisoning Attack (MPA)** : l’attaquant se comporte comme un client et entraîne le modèle reçu du serveur avec ses propres données malveillantes, en renvoyant des mises à jour biaisées.

Pourquoi ces attaques sont utilisées ?

L’attaquant peut viser plusieurs objectifs. Le premier est de nuire à l’apprentissage global en réduisant la précision générale du modèle. Cela peut avoir un impact concret sur des systèmes de production, comme des modèles de reconnaissance vocale ou de recommandation de texte.

Un deuxième objectif possible est de tromper le modèle sur des cas spécifiques, en ciblant par exemple une entrée ou une classe particulière que le modèle apprendra à mal classer. On distingue ainsi deux grandes familles d’attaques :

* **Les attaques ciblées** : elles cherchent à provoquer une erreur bien précise sur un sous-ensemble spécifique d’entrées.
* **Les attaques non ciblées** : elles visent à dégrader la performance du modèle globalement, sans se soucier du type d’erreur générée.

Le document *Back to the Drawing Board* se concentre spécifiquement sur les attaques non ciblées. Celles-ci sont jugées plus dangereuses car elles affectent potentiellement tous les utilisateurs et sont difficiles à détecter : la dégradation des performances peut être perçue comme une variation naturelle du processus d’apprentissage.

1. Attaque par empoisonnement non ciblés.

Les attaques non ciblées présentent l’avantage d’avoir un impact généralisé. Elles sont donc plus pertinentes dans un contexte de production. Elles se déclinent en deux formes principales :

1. Data Poisoning Attack (DPA)

L’attaquant modifie les données locales d’un client, par exemple en inversant les étiquettes (Label Flipping), ce qui perturbe l’apprentissage local du modèle. Le client, sans se douter de l’attaque, effectue un entraînement normal, mais génère des poids biaisés.

Dans le scénario analysé par le document, l’attaque est de type **Nobox Offline DPA** :

* **Nobox** : l’attaquant ne connaît pas la structure ou les poids du modèle global.
* **Offline** : l’attaque est préparée à l’avance sans adaptation dynamique au modèle.

Cette approche est jugée réaliste car elle ne nécessite qu’un contrôle minimal (par exemple l'accès à un appareil compromis) et ne requiert pas de connaissance spécifique du modèle global.

1. Model Poisoning Attack (MPA)

L’attaquant se comporte ici comme un client du système. Il reçoit le modèle global, l'entraîne localement sur des données malveillantes ou calcule directement une mise à jour optimisée pour nuire au modèle. Cette approche suppose un accès plus important (whitebox) et des capacités de calcul pour produire des gradients adaptés, souvent dans un contexte online.

Le document justifie la focalisation sur le DPA pour plusieurs raisons :

* Il correspond à un scénario plus réaliste dans les environnements FL déployés à grande échelle (notamment cross-device).
* Il est peu coûteux à mettre en œuvre : l’attaquant n’a pas besoin d’interagir avec le modèle global.
* Il s’inscrit dans un modèle de menace minimaliste mais crédible : accès local aux données d’un client, sans contrôle sur le processus d’agrégation.

Les résultats du document montrent que même avec un très faible pourcentage de clients compromis (0.1 %), les performances du modèle global peuvent être affectées, bien que des techniques simples de défense (comme le norm-bounding) permettent de contenir ces effets.

Ainsi, le DPA apparaît comme une menace à prendre au sérieux pour l'intégrité des systèmes FL, notamment parce qu’il est discret, scalable et difficile à détecter sans protections adaptées.