ZIMPERIUM®
ADVANCED MOBILE SECURITY

Top 7 des techniques d'obfuscationde code source

Techniques d'obfuscation de code source— Outils du métier

Jour après jour, le nombre et la portée des méthodologies et programmes malveillants utilisés par les pirates informatiques pour accéder sans autorisation aux applications, appareils et données personnelles augmentent. Parfois, c'est le code source même du logiciel qui fournit le point d'entrée pour l'attaque. En 2020, 35 % des violations peuvent être reliées à l'analyse et l'exploitation de vulnérabilités de code, faisant de cela le vecteur d'infection le plus courant, avant même l'hameçonnage. Heureusement, les professionnels de la sécurité qui ont la charge de protéger le monde en ligne ont leur propre arsenal d'outils pour riposter.

L'une des armes les plus importantes entre les mains des équipes de développeurs et de sécurité dans la guerre contre le piratage d'application, l'infiltration des appareils, l'injection de code et autres actes malveillants est l'obfuscation de code source. Mais qu'est-ce que l'obfuscation de code source et que signifie-t-elle dans le contexte du développement de logiciel ? Nous allons nous y intéresser de près.

Qu'est-ce que l'obfuscation de code source ?

L'obfuscation de code source transforme le code source de l'application pour accroître la difficulté et le temps nécessaire pour le comprendre, sans affecter l'exécution du programme. L'obfuscation de code source utilise diverses techniques pour rendre le code indéchiffrable pour les pirates informatiques, pour qu'ils ne puissent pas découvrir de vulnérabilités, voler les clés, des données ou d'IP ou trouver d'autres moyens de compromettre les applications. Avoir une stratégie claire du mode opératoire de l'obfuscation du code et l'utilisation de techniques de chevauchement aident à atténuer le risque que des décompilateurs et des attaquants essayent de faire de la rétro-ingénierie du programme.
OWASP cite la rétro-ingénierie comme l'une des principales menaces pour la sécurité des applications, puisqu'elle sert de tremplin pour la plupart des types d'attaques..

Pour donner une meilleure idée du fonctionnement de l'obfuscation en programmation, voici certaines des techniques de sécurité les plus couramment utilisées par les développeurs dans le monde.

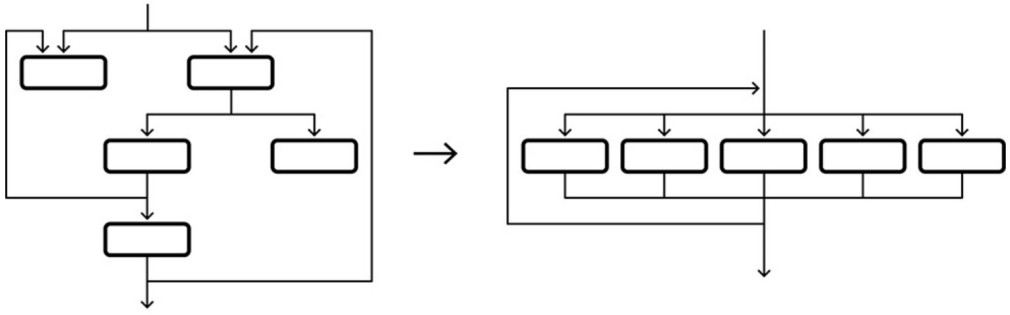
Sept techniques d'obfuscation de code source courantes

1. Transformation du code
L'un des éléments importants de l'obfuscation de code source consiste à transformer les données traitées par le programme en une autre forme, ayant un effet minimal sur la performance du code, mais qui complexifie sa compréhension et la rétro-ingénierie par les pirates informatiques.

Parmi les exemples d'obfuscation de code source basées sur cette technique, on peut citer l'utilisation de la forme binaire des nombres pour rendre le code source plus complexe, la modification de la forme dans laquelle les données sont enregistrées ou le remplacement d'une valeur par une expression.

2. Obfuscation du flux de code
En changeant le flux de contrôle du code, on modifie l'orientation du code. Cela signifie que, même si les résultats finaux sont les mêmes, il faut plus de temps pour comprendre pourquoi le code prend une certaine direction ou quelle est sa destination.

L'obfusccation du flux de contrôle peut se faire en altérant l'ordre des instructions d'exécution du programme, en changeant le graphe de contrôle par insertion d'instructions de saut arbitraire et en convertissant les constructions conditionnelles arborescentes en instructions Switch plates, comme le montre le schéma suivant..



3. Obfuscation d'adresse
Cette technique, utilisée par certains outils de sécurité pour l'obfuscation du code source, altère les adresses des données et le code du programme pour créer de l'imprévisibilité et compliquer le piratage. Après le développement d'une application, l'algorithme d'obfuscation randomise les emplacements absolus de certains codes et données en mémoire et les distances relatives entres les différents éléments de données. Cela réduit non seulement la probabilité de réussite des attaques, mais cela signifie aussi que, même si le pirate a du succès sur une application ou un appareil, il ne sera pas à même de reproduire cela sur d'autres, réduisant l'avantage de la rétro-ingénierie de programme.

4. Renouvellement régulier du code d'obfuscation
Ce système prévient les attaques de manière proactive en publiant régulièrement des mises à jour du logiciel obfusqué, pour contrecarrer les tentatives des pirates d'accéder au système. En remplaçant occasionnellement un logiciel existant avec de nouvelles instances obfusquées, celui qui l'attaque est obligé d'abandonner son analyse en cours. En fin de compte, l'effort pour franchir la barrière de sécurité de l'obfuscation ne vaut pas la valeur du gain.

5. Obfuscation des appels de message et des métadonnées pour Objective-C
Les outils d'obfuscation de code source fonctionnent avec Objective-C de deux manières. Premièrement, ils font de l'obfuscation des appels de message de texte clair contenus dans le code source pour qu'ils soient difficiles à lire et à modifier. Deuxièmement, ils cryptent certaines des métadonnées d'Objective-C pour cacher les informations sensibles des outils d'analyse, comme les noms de catégories, les classes, les méthodes, protocoles, propriétés de classe et variables d'instance, ainsi que les arguments de méthode et leurs types. Les données cryptées sont seulement décryptées lorsque l'application obfusquée est chargée pour le fonctionnement.

6. Obfuscation des instructions du code d'assemblage
La transformation et l'altération du code d'assemblage peuvent rendre plus difficile la rétro-ingénierie. Une telle méthode (aussi connue sous le nom de méthode « jump-in-the-middle ») consiste à utiliser des instructions d'assemblage de chevauchement qui cachent le code dans un autre code, poussant l'outil de désassemblage à produire un résultat incorrect. Le code d'assemblage peut également être renforcé contre la pénétration en incluant des instructions de contrôle non nécessaires et du code mort.

7. Obfuscation des informations de débogage
Les informations de débogage peuvent être utilisées pour faire de la rétro-ingénierie à un programme afin de découvrir son code source, en décompilant et recompilant son code. C'est la raison pour laquelle il est important de bloquer l'accès non autorisé et le débogage. Les outils d'obfuscation de code source accomplissent cela en modifiant les nombres de ligne et les noms de fichier dans les données de débogage ou en supprimant totalement les informations de débogage.

L'obfuscation de code source en programmation permet de lutter contre le pirate

Les attaques dirigées contre les logiciels, les applications et même les appareils IoT domestiques sont constantes. Elles augmentent à mesure que nos vies personnelles, nos données et informations de valeur sont transmises en ligne. Comprendre ou savoir comment faire de l'obfuscation de code est essentiel pour assurer notre défense contre les pirates et les autres mauvais acteurs. Pour les entreprises souhaitant protéger leur propriété intellectuelle et les données de leurs clients, une obfuscation de code source efficace et des outils qui appliquent l'obfuscation de code source et d'autres systèmes intégrés de protection des applications font partie des armes les plus puissantes dans leur arsenal.

En tant que pionnier de la sécurité mobile, Zimperium propose des techniques d'obfuscation de code source de première qualité, comme partie intégrante de sa suite de sécurité pour application mobile. zShield et zKeyBox de Zimperium protègent les informations importantes, telles que les clés cryptographiques, les algorithmes propriétaires et les données d'utilisateur.

Pour en savoir plus sur le fonctionnement de l'obfuscation de code source et comment nous protégeons votre logiciel, contactez-nous et parlez à l'un de nos experts en sécurité.

ZIMPERIUM.

4055 Valley View Dallas, TX 75244
844.601.6760
info@zimperium.com