Une faille Java met en danger JBoss, Jenkins, Weblogic et Websphere

Une société de sécurité, Foxglove Security (spécialisée dans les tests de pénétration), publie plusieurs codes (exploit) permettant de mettre à profit une faille critique dans une bibliothèque Java largement utilisée, issue des Apache Commons. Cette vulnérabilité, qui autorise l'exécution de code à distance, touche par ricochet de nombreux produits commerciaux, dont Weblogic, Websphere, JBoss, Jenkins ou OpenNMS.

« *La faille la plus sous-estimée de 2015, la moins relayée dans les médias a récemment attiré mon attention*, écrit **Stephen Breen**, l'un des dirigeants de Foxglove Security dans un billet de blog. *Personne ne lui a donné un joli nom de baptême, il n'y a pas eu de communiqué de presse, ni de recours à Mandiant pour éteindre l'incendie. En fait, même si un prototype de code d'exploitation de cette faille a été dévoilé il y a plus de 9 mois, aucun des produits mentionnés n'a été patché, ni tous les autres qui sont également concernés d'ailleurs. En plus des produits commerciaux faillibles, la vulnérabilité affecte aussi de nombreux développements spécifiques.* » Pour tenter de secouer le marché, Foxglove publie le code d'exploits pour JBoss, Jenkins, Weblogic et Websphere sur GitHub.

Java : pas d'ordre dans les librairies

La faille affectant la librairie Java repose sur la dé-sérialisation (processus permettant d'intégrer des données binaires dans une application). Stephen Breen raconte qu'en association avec un autre chercheur, il a commencé voici deux ans à chercher des failles zero day dans Websphere, avant de s'intéresser aux librairies communes à l'ensemble du monde Java. Et particulièrement à la dé-sérialisation. « *Pourquoi ? Parce que tout dans le monde Java utilise la sérialisation d'objets, et presque tout peut être contraint à accepter des données sérialisées non sécurisées et fournies par l'utilisateur* », écrit encore Stephen Breen.

Mais ce sont en réalité deux autres chercheurs, Gabriel Lawrence et Chris Frohoff, qui ont mis au jour, fin janvier, la faille affectant la librairie, très populaire dans le monde Java. Les codes produits par Foxgrove permettent de prendre pleinement conscience de sa nocivité. D'autant que, comme l'explique Stephen Breen, la situation est très différente de celle rencontrée avec d'autres failles affectant des librairies massivement déployées : « *Le recours aux librairies Java est très désordonné. Chaque serveur d'application possède son propre lot de librairies ; pire, chaque application que vous déployez sur le serveur est souvent liée à son propre set de librairies. Pour régler le problème, vous devez donc trouver et mettre à jour chaque librairie individuellement* », indique Stephen Breen. Comme Java n'exploite pas de librairies partagées, méthode qui facilite la correction de bug comme ce fut le cas pour Heartbleed, la vulnérabilité est là pour durer, assure le chercheur. Qui propose une méthode – qu'il qualifie lui-même d'horrible – pour limiter les effets de la faille : identifier manuellement tous les fichiers (jar ou class) faisant appel à la librairie et retirer la classe 'InvokerTransformer' utilisée dans l'exploit.

Apache a également livré un patch dans l'urgence permettant de désactiver la dé-sérialisation sur la classe en question. De son côté, Jenkins propose une méthode de contournement, désactivant le

service (CLI) détourné par l'exploit de Foxglove.

A lire aussi :

OpenWorld 2015 : Oracle trace la feuille de route de Java
Oracle : un nettoyage chez les évangélistes Java ?
Le méchant Android a cassé mon Java, pleurniche Oracle

crédit photo © Pavel Ignatov – shutterstock