YDS : Yacc intrusion Detection with Semantics

Éric Alata, INSA-Toulouse
Pierre-François Gimenez, CentraleSupélec
Thomas Robert, Télécom Paris

Localisation: équipe TSF du LAAS-CNRS, Toulouse, France

1 Mots-clés

détection d'intrusion – théorie des langages – *machine learning*

2 Contexte

Les interactions entre un utilisateur et les systèmes d'information reposent sur un motif architectural incontournable : les données de l'utilisateur sont intégrées dans des requêtes dont l'analyse est réalisée par un interpréteur qui pilote l'activité du système. Les attaques visant cette architecture sont très fréquentes et particulièrement sévères, comme en témoigne la récente attaque Log4Shell. Pour couvrir ces attaques, l'usage de systèmes de détection d'intrusion (IDS) est indispensable. Le plus souvent, cette détection se base uniquement sur la syntaxe de ces données, avec une connaissance limitée de leur sémantique. L'extraction automatique de cette sémantique est donc un enjeu de taille, car cela permettrait d'améliorer significativement les performances des détecteurs actuels.

3 Sujet

Les générateurs de parsers (tel que yacc) sont largement utilisés pour construire un interpréteur à partir de la grammaire de langages tels que bash, PHP ou SQL. Notre objectif est de modifier un générateur afin d'enrichir d'une sonde les interpréteurs générés. Le rôle de cette sonde sera de reconstituer la sémantique de la requête tout en l'annotant avec des informations sur le contexte de son analyse. L'utilisation d'un tel générateur aura trois avantages : 1) améliorer la qualité des données pour le détecteur et lui permettre de fournir une explication sémantique sur les alertes levées ; 2) annoter les requêtes permettra au détecteur de reconstruire le flux de données et ainsi identifier les flux anormaux et 3) placer une sonde dans un interpréteur permettra d'interrompre l'analyse d'une requête en levant une erreur de syntaxe.

L'exploitation de la sémantique des requêtes permet de raisonner sur des données épurées et moins bruitées et devrait réduire le nombre de faux positifs du détecteur ; ces faux positifs sont en effet une importante limite au passage à l'échelle des IDS, notamment par anomalies. Le motif architectural considéré dans cette étude est déployé en cascade. En incorporant une sonde au sein des interpréteurs déployés en cascade, nous nous attendons à être en mesure de suivre la propagation des requêtes.

La modification du générateur de parsers est générique et peut être utilisée avec n'importe quelle grammaire sans adaptation. Pour extraire automatiquement des informations sémantiques des grammaires, nous nous appuierons sur l'inférence de la sémantique à partir des étiquettes des non-terminaux de la grammaire d'intérêt. En effet, le concepteur du langage nomme généralement les non-terminaux en fonction de leur sémantique. Dans la grammaire SQL, on retrouve par exemple des non-terminaux nommés "createView", "partitionDefinition" ou encore "dropDatabase". Les progrès de l'analyse automatique du langage via des méthodes de deep learning suggèrent qu'il est possible de prédire si un non-terminal est associé à une certaine sémantique1, comme des actions de lecture, d'écriture ou de suppression.