UNIVERSITÉ DE RENNES 1

Direction de la Recherche et de l'Innovation

Appel à projets - Année 2022

Contrats doctoraux «Ordinaires»

ECOLE DOCTORALE : MathSTIC **POLE DE RECHERCHE : Numérique**

Unité de recherche d'accueil

Unité de Recherche : IRISA/UMR 6074

Equipe : PACAP

Encadrement

Directeur de thèse : Isabelle Puaut **Unité de recherche** : IRISA

Employeur/Établissement d'origine : Université de Rennes 1

HDR : ☑ oui ☐ non **Nb de thèses en cours** : 3

Mail : isabelle.puaut@irisa.fr **Tél**. : 02 99 84 73 10

Co-directeur de thèse : Erven Rohou **Unité de recherche** : Inria

Employeur/Établissement d'origine : Inria

HDR : ☑ oui ☐ non **Nb de thèses en cours** : 3 (dont 1 soutenance au printemps 2022)

Mail : erven.rohou@inria.fr **Tél.** : 02 99 84 74 93

Sujet

Titre : Approches par modification de code machine pour la défense contre les attaques ROP et JOP

Mots clés : Analyse de risques ; Environnement de conception sécurisé ; Outils et techniques d'évaluation ; Évaluation logicielle ; Outils d'investigation ; OS et applications ; Terminaux et objets connectés ; Sécurité de l'Internet des objets, Informatique de confiance ; Composants et hardware

Cotutelle internationale (oui/non) : non **Si oui, préciser l'établissement et le pays de rattachement** :

Descriptif (15-30 lignes) :
Contexte socioéconomique et scientifique ; Les hypothèses et questions posées ; Les grandes étapes de la thèse et démarche ; Approches méthodologiques et techniques envisagées

Objectifs de la thèse et originalité
Le contexte de la thèse est celui de la sécurité des systèmes informatiques en présence d'attaques d'usagers malveillants. Les premières attaques par débordement de tampon, permettant d'exécuter du code situé dans la pile du processus attaqué, ont motivé l'introduction de contre-mesures (protection des piles contre l'exécution, protection W⊕X, randomisation des adresses de base de certains segments). Les attaques les plus élaborées recherchent du code (gadgets) composant les attaques dans le code existant (bibliothèque, section de code des processus, système), et exécutent ce code en positionnant des adresses de retour vers ce code dans la pile (ROP, *Return-Oriented Programming*) ou en se branchant directement sur le code des gadgets (JOP, *Jump-Oriented Programming*). L'utilisation grandissante des randomisations d'adresse pour non seulement les bibliothèques mais aussi le code ou la pile (ASLR, *Address Space Layout Randomization*) rendent les attaques ROP plus compliquées puisqu'il faut deviner l'adresse des gadgets pour pouvoir les exécuter.

L'objectif de la thèse est de développer des techniques de compilation permettent de se défendre contre des attaques de type ROP et JOP. Les méthodes de compilation développées seront basées sur les technologies de transformation de binaires développées par l'EPI PACAP (outil Padrone, et son successeur Damas [1, 2]). L'objectif des techniques de compilation proposées sera d'éviter la présence de gadgets dans le code machine ou d'ajouter des protections supplémentaires dans le code machine pour éviter le branchement aux gadgets (utilisation de *shadow stack*, protection des branchements, etc.)

Nous privilégierons l'application de techniques procédant par modification du code binaire (une fois la génération de code par le compilateur effectuée). Ces modifications seront effectuées soit lors du chargement des programmes, soit lors de leur exécution. L'originalité de ce type d'approche est de pouvoir travailler sur du code pré-existant, non sécurisé a priori, et dont le code source n'est pas disponible.

Verrous et moyens de les lever

Le principal verrou lié à l'introduction de techniques de sécurité par modification de code binaire est celui du surcoût temporel engendré. Ce surcoût sera géré par l'utilisation de méthodes de compilation appliquées au chargement du programme, mais également si les transformations sont appliquées à l'exécution par des techniques de maitrise des coûts, l'équipe PACAP étant experte dans l'amélioration des performances des logiciels. Un second verrou est d'ordre technologique : les techniques de transformation de binaires sont très pointues. PACAP possède une expertise reconnue dans ce type d'outil : Padrone, infrastructure générique de modification de binaires [1], et Damas [2], environnement de réécriture de binaire dédié au tissage de mécanismes de sécurité dans un programme au format binaire non protégé.

Références

[1] Emmanuel Riou, Erven Rohou, Philippe Clauss, Nabil Hallou, and Alain Ketterlin. PADRONE : a Platform for Online Profiling, Analysis, and Optimization. In DCE 2014 - International workshop on Dynamic Compilation Everywhere, Vienne, Austria, January 2014.

[2] Camille Le Bon, Erven Rohou, Frédéric Tronel, and Guillaume Hiet. DAMAS : Control-Data Isolation at Runtime through Dynamic Binary Modification. In SILM 2021 - Workshop on the Security of Software / Hardware Interfaces, pages 1–10, digital event, Austria, September 2021.

[3] Etienne Louboutin. Sensibilité de logiciels au détournement de flot de contrôle. Theses, Ecole nationale supérieure Mines-Télécom Atlantique, January 2021.

[4] Henry Massalin. Superoptimizer: A look at the smallest program. In Proceedings of the Second International Conference on Architectual Support for Programming Languages and Operating Systems, ASPLOS II, page 122–126, Washington, DC, USA, 1987. IEEE Computer Society Press.

[5] Manasij Mukherjee, Pranav Kant, Zhengyang Liu, and John Regehr. Dataflow-based pruning for speeding up superoptimization. Proc. ACM Program. Lang., 4(OOPSLA), November 2020.

[6] Eric Schkufza, Rahul Sharma, and Alex Aiken. Stochastic superoptimization. In Proceedings of the Eighteenth International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS '13, page 305–316, New York, NY, USA, 2013. Association for Computing Machinery.

[7] Zheng Wang and Michael O'Boyle. Machine learning in compiler optimization. Proceedings of the IEEE, 106(11) :1879–1901, 2018.

[8] Edward J Schwartz, Thanassis Avgerinos, and David Brumley. Q: Exploit hardening made easy. In USENIX Security Symposium, volume 10, 2011.

[9] Vasilis Pappas, Michalis Polychronakis, and Angelos D Keromytis. Smashing the gadgets : Hindering return-oriented programming using in-place code randomization. In 2012 IEEE Symposium on Security and Privacy, pages 601–615. IEEE, 2012.

[10] Stephen Checkoway, Lucas Davi, Alexandra Dmitrienko, Ahmad-Reza Sadeghi, Hovav Shacham, and Marcel Winandy. Return-oriented programming without returns. In Proceedings of the 17th ACM conference on Computer and communications security, pages 559–572, 2010.

[11] Kaan Onarlioglu, Leyla Bilge, Andrea Lanzi, Davide Balzarotti, and Engin Kirda. G-free : defeating return-oriented programming through gadget-less binaries. In Proceedings of the 26th Annual Computer Security Applications Conference, pages 49–58, 2010.