Sujet de projet INFO 731

Projet 1- Proxy de sécurité Web

Le communication web peuvent être surveillée et écoutée . Afin de traiter ceci nous souhaitons mettre en place un mécanisme permettant de chiffrer le traffic à l'entrée d'un domaine ou le traffic est susceptible d'être écouté et de déchiffrer celui-ci à la sortie. L'objectif de ce projet est le développement de ce proxy. Le proxy reçoit les requêtes http issu d'un navigateur web qui a été configuré pour utiliser un proxy, et chiffre ces requêtes avant de les envoyer au proxy de sortie. Le proxy de sortie fait la requête http au serveur web à la place du navigateur web, reçoit la réponse la chiffre et la renvoie vers le proxy source, qui renvoie finalement au navigateur qui affiche la page demandée.

1- vous construirez un proxy acceptant les requêtes web et les renvoyant vers un proxy de sortie qui fait la requête à leur place

2- Vous implanterez un mécanisme d 'échange de clé d'encryption par clé publique. Pour implanter RSA vous utiliserez une librairie de cryptographie

3- Vous utiliserez la clé d'encryption échangée par le mécanisme RSA pour chiffrer le traffic

4- vous vérifierez si vous proxy marche sur un grand nombre de sites. Vous mesurerez la performance de votre proxy en terme de débit et de délai.

Nota Bene: je détecterai tous copier/collé de code existant !

Projet 2- calcul homomorphique

Le calcul homomorphique est une approche révolutionnaire permettant d'effectuer des calculs sur des données chiffrer. Le sujet du second projet consiste a utiliser le calcul homomorphe afin de faire du calcul.
1- lire https://medium.com/golden-data/introduction-to-homomorphic-encryption-d903d02d4ce0

2- voir la libraire SEAL https://github.com/Huelse/SEAL-Python/blob/master/README.md

3- Voir le rapport de projet https://courses.csail.mit.edu/6.857/2017/project/9.pdf

4- le but du projet est d'implanter la regression linéaire avec le calcul homomorphique et de le comparer avec le calcul sans encryption

5- vous mesurerez les performance en terme de débit et délai.

Projet 3- Surveillance fine de processus pour la cybersecurité

Afin d'aider un peu plus l'analyse et la compréhension des performances observées, les compteurs matériels de performances (registres spéciaux), si présents dans le CPU,

peuvent être exploités. Ces compteurs enregistrent plein d'événements matériels relatifs aux performances qui sont provoqués par l'exécution de processus: nombre de cycles d'horloge, nombre d'accès mémoire, nombre d'accès au cache L1, nombre de défauts de cache, etc. La nature des événements enregistrés par les compteurs matériels de performances dépend d'une architecture de CPU à une autre. La portabilité n'est donc pas assurée, mais les mesures sont extrêmement précises.

Notons deux façons pour utiliser les compteurs matériels de performances:
Analyser les événements matériels d'un CPU qui exécute plusieurs programmes en concurrence. Par exemple, l'outil Likwid (https://github.com/RRZE-HPC/likwid) récupère les valeurs des compteurs matériels par cœur (et non pas par processus). Il peut agréger les compteurs matériels de tous les cœurs. Cette façon d'utiliser les compteurs matériels de performances permet d'avoir une vue globale du fonctionnement et des performances d'un cœur ou d'un système entier qui exécute plusieurs processus en même temps.

Analyser les événements matériels provoqués par un seul processus qui s'exécute sur une machine. C'est la façon à utiliser pour analyser et éventuellement optimiser les performances d'un programme particulier qui s'exécute sur une architecture matérielle précise.

Là aussi, notons deux façons pour utiliser des compteurs matériels de performances pour un code donné :
Soit l'utilisateur est intéressé par l'analyse des performances d'un bout son programme (par exemple étudier les performances d'une boucle précise ou d'une fonction particulière). Dans ce cas, il faut détenir le code source pour l'instrumenter (le modifier). Le code modifié contiendrait des instructions utilisant des librairies d'accès aux compteurs matériels de performances. La librairie LibPFM ou PAPI permet par exemple à un programme d'accéder aux valeurs des compteurs matériels de performances.
Soit l'utilisateur est intéressé par l'analyse des performances d'un processus entier (pas uniquement un bout du programme). Dans ce cas de figure, l'utilisateur n'a pas besoin d'instrumenter le code source. Il peut utiliser des outils en ligne de commande comme perf pour évaluer les performances du code binaire.
Dans ce projet nous allons étudier en détails ces compteurs matériels de performances. L'étudiant devra se familiariser avec les différents aspects comme ceci:

1-Étudier les événements matériels pouvant être enregistrés sur votre machine de test. Reprendre des codes de micro-benchmarks et les analyser finement avec les compteurs matériels de performances.
2- Réfléchir à comment utiliser ces métriques pour un monitoring de l'activité d'un CPU pour des besoins de cybersecurité.