3IF5020 - Introduction aux attaques en mémoire

Responsables : **Frederic TRONEL**

Langues d'enseignement : **FRANCAIS**

Campus où le cours est proposé : **CAMPUS DE RENNES**

Nombre d'heures d'études élèves (HEE) : **45**

Nombre d'heures présentielles d'enseignement (HPE) : **24**

Année académique : **2024-2025**

Niveau avancé : **non**

Présentation, objectifs généraux du cours :

Dans ce cours, nous nous intéressons à une large classe d'attaques à savoir celles liées à des corruptions de la mémoire.
Ces attaques sont en grande partie dues à des erreurs de programmation lors de la gestion manuelle de la mémoire dans des langages de programmation tels que C et C++.
Elles sont une des principales sources de vulnérabilités exploitées par des attaquants et ce depuis au moins 30 ans.
L'absence de mécanisme de sûreté autour de la gestion de la mémoire laisse le champ libre aux attaquants pour détourner à leur profit l'exécution de programmes vulnérables.
Un très grand nombre de palliatifs à ces problèmes ont été proposés; pour autant aucun d'entre eux n'a encore permis d'arrêter complètement le flot d'attaque reposant sur cette classe de vulnérabilités.

Période(s) du cours (n° de séquence ou hors séquence) :

SM10

Prérequis :

Pour suivre ce cours avec profit, il faut avoir suivi au préalable un cours d'assembleur (idéalement celui du processeur Intel X86 mais tout autre processeur peut suffire pour suivre, notamment le processeur RiscV étudié en InfoSec1), un cours de langage C et un cours de fonctionnement des systèmes d'exploitation.
Idéalement un cours de compilation permet de mieux comprendre les problèmes de sécurité liés à la manipulation manuelle de la mémoire, ainsi que les contre-mesures possibles au niveau du compilateur.

Plan détaillé du cours (contenu) :

Ce cours est organisé de la manière suivante:
- Introduction.
- Bref rappels des connaissances supposées acquises (C, assembleur, systèmes d'exploitation)
- Étude d'un code présentant une erreur de programmation grossière (débordement dans la pile):
  - Étude du code binaire
  - Analyse dynamique au débogueur.
- Étude d'un code modifiant lui-même son adresse de retour:

- Étude du code binaire.
- Analyse dynamique au débogueur.

- Principe général d'une attaque par débordement de mémoire sur la pile.
- Conception d'une charge offensive pour une attaque par débordement de mémoire sur la pile.
- Revue des principales contre-mesures:
  - Protection stricte des différentes zones de mémoire (code, tas, pile, bibliothèques).
  - Randomisation des espaces d'adressage virtuels.
  - Utilisation d'un canari.
  - Placement intelligent des variables dans la pile.
- Pour aller plus loin (quelques éléments sur des attaques plus élaborées).

Les séances de travaux pratiques sont consacrées à une mise en situation permettant de mettre en pratique ces connaissances sur une attaque visant un serveur accessible par le réseau.

Déroulement, organisation du cours :

CM 9 h
TP 15 h

Organisation de l'évaluation :

Le module sera évalué par une mise en situation conduisant à analyser l'attaque d'un système d'information via l'exploitation à distance d'un service vulnérable.
Cette mise en situation se déroulera durant des séances de TP.
Le TP fera l'objet d'une restitution orale notée.

CF: Présentation orale en binôme.

L'examen oral permettra de valider les compétences C2 (pour la partie technique, travail réalisé) et C7 (pour la qualité de la présentation).

Moyens :

Enseignants: Frédéric Tronel et Pierre Wilke
Logiciels: Virtualbox, chaîne de compilation (gcc et gdb).

Acquis d'apprentissage visés dans le cours :

- Auditer un code source en langage C dans le but de trouver des erreurs de manipulation de la mémoire (dépassement de tampon ou buffer overflow, double libération, utilisation après libération, chaînes de format, ...).
- Déboguer un code binaire à l'aide d'un débogueur (ou dévermineur) afin de traquer des erreurs de manipulation de la mémoire (afficher le code source, le code assembleur, afficher les registres du processeur, faire fonctionner le code en mode pas à pas, savoir poser un point d'arrêt, savoir poser un point d'observation, ...).
- Auditer le fonctionnement d'un code binaire inconnu à l'aide d'outils spécialisé (comme Valgrind par exemple)
- Collecter des indicateurs de compromission sur un système d'information inconnu (trace réseau, exploration des journaux systèmes, audit des fichiers de configuration).
- Analyser une charge offensive à partir de son code binaire.

Description des compétences acquises à l'issue du cours

:

C2.5
C7.4

Bibliographie :

One, Aleph. "Smashing the Stack for Fun and Profit." Phrack 7 , no. 49 (1996)