3IF5050 - Techniques Avancées d'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) : **40**

Nombre d'heures présentielles d'enseignement (HPE) : **18**

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.
Nous tentons de couvrir l'ensemble des techniques mises au point par les attaquants afin d'exploiter ces vulnérabilités.
Nous faisons aussi le tour de l'ensemble des contre-mesures à l'état de l'art en matière de lutte contre ces vulnérabilités.

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

SM10

Prérequis :

Afin de suivre ce cours avec profit, il est obligatoire d'avoir suivi la première partie du cours consacrée aux attaques de base contre la mémoire.
Il est par ailleurs utile d'avoir suivi un cours de compilation afin de comprendre les contre-mesures implémentées au niveau du compilateur.

Plan détaillé du cours (contenu) :

Ce cours commence par passer en revue les différentes méthodes d'exploitation des vulnérabilités liées aux erreurs de manipulation de la mémoire.
Pour chacune des classes de vulnérabilité, on illustre la faute de programmation à son origine, ainsi que la manière dont l'erreur qu'elle peut provoquer à l'exécution du code fautif se propage jusqu'à permettre à un attaquant d'exploiter la vulnérabilité:

- Erreurs liées aux chaînes de format (format string error).
- Erreurs liées aux débordement de capacité des entiers et leur lien avec les erreurs de gestion de la mémoire (integer overflow).
- Débordement de tampon dans le tas (heap overflow):
  - Étude de la structure du tas sous Linux: algorithmes dlmalloc et ptmalloc; exploitation des erreurs de programmation des allocations dans le tas.
  - Étude de la structure du tas sous Windows; exploitation des erreurs de programmation des allocations dans le tas.

- Contre-mesures les plus répandues dans les systèmes d'exploitation: randomisation des espaces d'adressage et application d'une politique de droits sur les zones mémoire (ASLR et bit NX).
- Contournement des contre-mesures classiques:
  - Retour vers la librairie C (Ret-to-libc): introduction du concept et exemple d'exploitation.
  - Return Oriented Programming (ROP): introduction du concept et exemple d'exploitation.
- Outils de découverte des gadgets pour le ROP.
- Détournement de l'exécution vers les zones de code compilées à la volée: JIT et Heap spraying.

Confusion de types dans les langages "sécurisés":

  - table de pointeurs dans les langages orienté objet ou orienté prototype
  - principe des confusion de type dans les langages typés (dynamiquement ou statiquement)
- Étude des formats de binaires:
  - Format de fichiers binaire ELF:
    - Entête, rôle des sections et des segments.
    - Traitement des relocalisations (relocations).
    - Fonctionnement du chargeur (ld.so). Lien avec la mise en place de la randomisation de l'espace d'adressage et de l'application d'une politique de droits stricts sur les différentes zones de mémoire.
    - Résolution dynamique des symboles par l'éditeur de liens dynamique (ld.so). Lien avec la mise en place de
    - la randomisation de l'espace d'adressage. Fonctionnement des sections .got et .plt.
    - Appels système et VDSO.
    - Détournement du flot d'exécution via la GOT et la PLT.
    - Contre-mesures à la compilation.
  - Format de fichiers binaire PE:
    - Entête et rôle des tables d'import et d'export
    - Fonctionnement du chargeur et l'éditeur de lien dynamique
    - Randomisation de l'espace d'adressage
- Contre-mesures :
  - Protection du flot de contrôle (Control Flow Integrity) via le compilateur.
  - Protection du flot de données (Data Flow Integrity) via le compilateur.
  - Protection des pointeurs (Cheri par exemple) via le matériel.
  - Intel CET

Déroulement, organisation du cours :

CM 9h
TP 15h

Organisation de l'évaluation :

Ce module sera évalué via une mise en situation conduisant à analyser l'attaque d'un système d'information via l'exploitation à distance d'un service vulnérable. La charge utile utilisée par l'attaquant sera capable d'outrepasser plusieurs contre-mesures de sécurité mises en place sur le système. Son analyse permettra d'illustrer les différents concepts vus durant le cours. Cette mise en situation se déroulera durant des séances de TP. Le TP débouchera sur une présentation orale.

CF: présentation orale
La présentation orale permettra de valider la compétence C2 (partie technique) et C7 (présentation orale).

Moyens :

Enseignants: Frédéric Tronel et Pierre Wilke

Acquis d'apprentissage visés dans le cours :

- Auditer un code source écrit en langage C ou C++ afin d'y trouver des vulnérabilités relevant des différentes classes d'attaques liées à la manipulation manuelle de la mémoire.
- Auditer une charge malveillante évoluée.
- Auditer la configuration d'un système d'exploitation, et de sa chaîne de compilation afin d'en qualifier la robustesse.

Description des compétences acquises à l'issue du cours :

C2.3
C7.1

Bibliographie :

- Solar Designer, "Return-into-lib(c) exploits" sur seclists.org, août 1997
- Nergal, "The advanced return-into-lib(c) exploits", Phrack, n°58, 2001
- Sebastian Krahmer, "x86-64 buffer overflow exploits and the borrowed code chunks exploitation technique", septembre 2005
- Jonathan Salwan et Allan Wirth, "ROPgadget, Gadgets finder and auto-roper"
- Laszlo Szekeres, Mathias Payer, Tao Wei, Dawn Song, "SoK: Eternal War in Memory", Proc. of the 2013 IEEE Symposium on Security and Privacy