Projet S6

Enoncé

Pour ce projet, deux sujets sont au choix:

- Sujet 1: bypass SMAP/SMEP en partant du TP1
- Sujet 2: Fuzz pour retrouver l'overflow du TP1

Le projet est à faire à deux, et le rendu devra contenir un rapport détaillé ainsi que le matériel nécessaire pour pouvoir reproduire l'intégralité du projet.

Pour toute question, ne pas hésitez à m'envoyer un message sur discord. PS: Je ne check pas mes mails sur ma boite epita.

Rapport

La qualité du rapport sera pris en compte dans la notation. Ce dernier doit être en anglais et contenir une page de garde, un sommaire avec intro/parties/conclusion. Le rapport doit être rendu au format pdf, et de préférence fait en LaTeX.

Matériel

Pour pouvoir reproduire l'intégralité du projet, il est demandé de fournir les fichiers nécessaires, tel que: `initramfs.cpio.gz`, `run.sh`, `exploit.c` et bzImage.

Pour le Sujet 2, d'autres éléments peuvent être apporté, tel que la config ou commande nécessaire pour utilisers le fuzzer.

Sujet 1

Pour le sujet 1, il est demandé de reprendre l'environnement du TP1 et d'ajouter les protections SMEP puis SMAP. Pour cela, il faut reprendre l'environnement du TP1 et ajouter `+smep` à `-cpu` lors du qemu-system dans le `run.sh`, puis `+smap`.

Vous pouvez vous appuyer des liens suivants:

- https://lkmidas.github.io/posts/20210128-linux-kernel-pwn-part-2/
- https://0x434b.dev/dabbling-with-linux-kernel-exploitation-ctf-challenges-to-learn-the-ropes/

Le rendu devra contenir un rapport détaillé, et un zip contenant l'environnement avec l'exploit.

- L'environnement doit contenir: initramfs.cpio.gz (qui contient exploit), bzImage et run.sh
- **Le rapport doit avoir une première partie sur le ret2usr** que nous avons vu en cours.
- Attention, il n'est pas demandé de simplement copier le contenu des liens données, mais de comprendre par vous même et d'essayer de bypass SMEP puis SMAP.

Sujet 2

Pour le sujet 2, il est demandé de fuzz le module du TP1 et d'obtenir le crash causé par le buffer overflow, avec l'aide de syzkaller ou un autre outil de fuzzing. Il n'est pas demandé de dev un fuzzer.

Les sources du module du TP1 sont disponible sur moodle: sources-module-kexpita

Vous pouvez vous appuyer sur le lien suivant: https://slava-moskvin.medium.com/fuzzing-the-kernel-with-syzkaller-part-1-setting-up-on-mac-and-crashing-a-vulnerable-driver-b2a3949ea575

(BONUS): Fuzz le module du TP3 (enlever ligne 116 de kexpita.c: `g_buf = NULL;` pour avoir l'UAF)