Call for side quest

« NetCore »

Sommaire

Abstraction

Nous avons dans le but de “*upskill*” dans ce qu’on peut considérer comme fondamentale nos compétence réseau bas niveau. L'idée sera de reconstruir Wireshark à partir de zéro en langage C ou Rust et d'embarquer notre version personnalisée de Wireshark sur un Raspberry Pi où Arch Linux en sera le système d'exploitation. Ce projet est pour nous l'occasion de "monter en compétences" dans les couches 2, 3 et 4 du modèle OSI, voire même la couche 1 et de parfaitement maîtriser l'ensemble.

NetCore

Afin de réaliser cette Side Quest, elle sera constituée de <n> étapes distinctes. Le tout couvrant ainsi une expertise approfondie des couches 1, 2, 3 et 4 du modèle OSI.

Step 1: Etat de l'art (1 mois)

Nous commencerons par faire un "état de l'art" des différents composants servant à la réalisation de notre version personnalisée de Wireshark, à savoir les points suivants :

- Compréhension d'une carte réseau (NIC)
- Compréhension des protocoles réseau majeurs (TCP, IP, ARP, DHCP, etc.), tous dans l'idéal
- Review code de la lib pcap
- Review code du driver EBPF
- Compréhension de l'OS "Arch Linux"
- Review de toutes les RFC concernant les points cités ci-dessus

Des composants important dans l'architecture d'un OS où l'ont veut vraiment comprendre parfaitement tous les protocoles majeurs, utilisés dans l'inter communication quotidienne des machines (TCP/IP, ARP, DHCP, etc.).

Mais aussi comprendre parfaitement comment l'OS s'interface et gère le fonctionnement avec la carte réseau. Viens aussi tout les protocoles de communications sans fil (WEP, WPA(x), etc...)

Je pense qu'ils nous faudra lire tout les RFCs concernant le réseau et reconstruire à la main la lecture des protocoles de communication.

Step 2: Preuve de concept (4 mois)

Une fois les points clés identifiés par l'étape précédente, nous pourrons commencer à réaliser des POC (Proof of Concept) des éléments fondamentaux du projet, pour ensuite envisager d'ajouter des fonctionnalités supplémentaires potentielles.

Exemples : Network Interface Card (NIC) :

- Recoder un driver NIC
- Recoder libpcap

Protocoles :

- Recoder une capture TCP
- Recoder une capture ARP

Step 3: Documentation et masterclass (1 mois)

Chaque composant se vera une documentation distinct avec l'élaboration d'une masterclass (15-30 min) propre au composant lui même.

Extra

Ci-dessous des idées supplémentaire pour nourrir cette side-quest et aller plus loin dans l'expertise !

- *POC d'ancienne CVE avec nos logiciel custom*
    1. **CVE-2022-2383:** A vulnerability in Wireshark that could allow remote code execution when processing malformed packets.
    2. **CVE-2021-2207:** A vulnerability in libpcap that could lead to a denial of service or remote code execution.
    3. **CVE-2020-10736:** A vulnerability in Wireshark that could allow remote code execution when handling certain protocols.
    4. **CVE-2019-14041:** A vulnerability in Wireshark that could lead to a denial of service or remote code execution when processing malformed packets.
    5. **CVE-2018-16043:** A vulnerability in libpcap that could allow remote code execution when processing malformed packets.
    6. **CVE-2017-1000254:** A vulnerability in Wireshark that could allow remote code execution when handling certain protocols.
    7. **CVE-2016-2388:** A vulnerability in Wireshark that could lead to a denial of service or remote code execution when processing malformed packets.
    8. **CVE-2015-7578:** A vulnerability in Wireshark that could allow remote code execution when handling certain protocols.
    9. **CVE-2014-2297:** A vulnerability in Wireshark that could lead to a denial of service or remote code execution when processing malformed packets.
    10. **CVE-2013-4164:** A vulnerability in Wireshark that could allow remote code execution when handling certain protocols.
- *Création/réalisation de chall CTF lié à cette side quest*
- *« Ton incroyable idée ! »*