Notes de cours de Sécurité des applications 2

Yann Miguel

19 novembre 2021

Table des matières

1 Introduction

Il est important de comprendre ce que l'on fait pour éviter des risques inutiles. Il faut toujours imaginer le pire scénario possible.

2 Cours 1

Découverte de différents systèmes pour trouver des vulnérabilitées, tels que:

- Les **Seven Pernicious Kingdoms**
- Les **24 Deadly Sins of Software Security**, qui sont séparés en plusieurs catégories:
  - Les **Web Application Sins**
  - Les **Implementation Sins**
  - Les **Cryptographic Sins**
  - Les **Networking Sins**
- Les **CWE**
- Les **OWASP Top 10**, qui se déclinent en différents tops:
  - Mobile
  - Docker
  - API
  - etc...

3 Cours 2

La gravité et l'importance d'une vulnérabilité sont déterminées par différents groupes de métriques. Microsoft DREAD(Damage, Reproductibility, Exploitability, Affected users, Discoverability) est un modèle de calcul d risques.

Liste de sites(non exhaustive) pour trouver des vulnérabilités:

- MITRE
- NIST
- Exploit-db
- etc...

Un site pour trouver des outils de reconnaissance open-source: OSINT Framework

4 Cours 3

Durant les opérations de pentesting, il est important de définir des objectifs clairs, et les moyens autorisés pour y parvenir. Si les contraintes ne sont pas respectées, vous vous exposez aux sanctions définies par l'article 323-1 du code pénal.

Le pentesting peûx coûter cher à une entreprise, il faut donc que l'analyse soit la plus poussée et clair possible, afin que cela vaille le coup pour elle. Il ne faut pas oublier d'être rationnel et raisonnable en testant les menaces, afin déviter les dépenses superflues.

Faire des revues de code permet de trouver des problèmes d'implémentation. Idéalement, cela commence dè que l'on commence à coder, mais ça doit être pécisé par les demandes du projet. L'analyse de code se doit aussi d'être ciblée; En effet, il est inutile de vérifier des problèmes de gestion de mémoire manuelle dans un langage dans lequel la gestion de mémoire est automatisée.