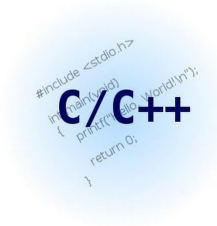
/



Formation C C++ Secure Coding

Dr. Ing. Chiheb Ameur ABID

Contact: chiheb.abid@gmail.com

Aôut 2019

Démarche pédagogique

Démarche pédagogique

Présentation des concepts, démonstrations et ateliers

- Buffer overflow, Stack smashing, Injection SQL, Exécution d'un code arbitraire, Hardening, etc.
- Mise en œuvre des bonnes pratiques

Outils

- Système d'exploitation : Linux (distribution Debian)
- Compilateur : GCC (version > 8.0)
- Environnement de développement : KDevelop / Eclipse / Cevelop
- Autres outils : objdump, GDB, Valgrind

Plan

Motivations

Programmer avec C et C++

- Performance
- Empreinte mémoire faible
- Portabilité
- Bibliothèques existantes (C depuis 1978, C++ depuis 1983)

Motivations

Programmer avec C et C++

- Performance
- Empreinte mémoire faible
- Portabilité
- Bibliothèques existantes (C depuis 1978, C++ depuis 1983)

Esprit de codage avec C/C++

- Trust the programmer
- Don’t prevent the programmer from doing what needs to be done
- Make it fast even it is not guaranteed to be portable

Motivations

Programmer avec C et C++

- Performance
- Empreinte mémoire faible
- Portabilité
- Bibliothèques existantes (C depuis 1978, C++ depuis 1983)

Esprit de codage avec C/C++

- Trust the programmer
- Don’t prevent the programmer from doing what needs to be done
- Make it fast even it is not guaranteed to be portable

Les problèmes avec C++ ?

- C et C++ ne protègent pas les programmeurs
- Standardisation (1983)
- Différentes implémentations des compilateurs

Introduction

Objectifs

Connaître les notions préliminaires liées au codage sécurisé d'une application

Sensibiliser les développeur sur la sécurité du code

Programme du module

- Motivations
- C et C++ des langages peu sécurisés ?
- Connaître les risques liés à la programmation
- Les traces laissées par les développeurs

Classification des risques

Objectifs

Connaitre les principaux acteurs dans le domaine de sécurité

Classification des risques selon CERT

Connaitre les principales références documentaires et guides pour le codage sécurisé en C et C++

Programme du module

- Les différents acteurs : CERT, PCI, CWE, OWASP, etc.
- Codage sécurisé d'une application
- Classification des risques selon CERT
- Guides pour le codage sécurisé

Codage sécurisé avec C et C++

Objectifs

Étudier les différentes vulnérabilités pouvant être introduites dans un programme C/C++ à travers des exemples

Découvrir les techniques et les solutions pour remédier à chaque type de vulnérabilité

Programme du module

- Modèle mémoire
- Compilation
- Appels des fonctions
- Les tableaux et les chaînes de caractères
- Les pointeurs
- Gestion de la mémoire dynamique
- Sécurité des entiers
- Sorties formatées
- Les fichiers

Les bonnes pratiques

Objectifs

Les bonnes pratiques à mettre en oeuvre.

Techniques de vérification d'un programme afin d'identifier les vulnérabilités

Programme du module

- Macro et inline
- Gestion de la mémoire
- Gestion des erreurs et des exceptions
- Structure des classes
- Passer à C++14 et C++17
  - Généralités : nullptr, enum, deleted fonctions, etc.
  - Utilisation des pointeurs intelligents (smart pointers)
- Vérification du code : valgrind, gdb

Durcissement de la sécurité (Hardening)

Objectifs

Connaitre et évaluer les différents mécanismes et options offerts par le compilateur et les systèmes d'exploitation (OS) permettant d'améliorer la protection des applications.

Programme du module

- Mécanismes de protection offerts par GCC
- Les options de sécurité de GCC (formatting strings, stack protection, read-only relocation, fortify source protection)
- Tests des configurations de GCC
- Validation du "Hardening"

MERCI POUR VOTRE ATTENTION

Questions ?