**APP1 – Programmation sécurisée**

Saïd AARABAT

Guillaume CHANTREL

Mercredi 27 Septembre 2017

Session 8 – Génie informatique

Sommaire

Introduction 3

1. Prise de besoin 3

1.1. Données en transit 3

1.2. Besoins de sécurité 3

2. Conception 3

2.1. Modélisation de la menace 3

2.2. STRIDE 3

3. Réalisation 3

3.1. Outils utilisés 3

3.2. Langage utilisé 4

3.3. Mécanismes de sécurité 4

3.3.1. Https 4

3.3.2. Authentification par Token 4

3.3.3. Offuscation de l’exécutable 4

3.3.4. Autres préventions 4

4. Plan de réponse aux menaces 5

4.1. Plan de réponse 5

4.2. Révision de sécurité 5

4.2.1. Fuzzing 5

4.2.2. Man in the middle 5

4.2.3. Deny of service 5

4.2.4. Reverse engineering 5

4.3. Archivage ??? 5

5. Conclusion 5

# Introduction

Introduire le module de programmation sécurisée

# Prise de besoin

Reformulation de la problématique

## Données en transit

Données json réponses questions

## Besoins de sécurité

Des échanges sécurisés

Authentification de l’utilisateur

Gestion precise des erreur et exceptions

# Conception

## Modélisation de la menace

Sheema

## STRIDE

Tableau

# Réalisation

## Outils utilisés

Options Compilateur pourquoi --> /GS

## Langage utilisé

C# avec .Net Core

## Mécanismes de sécurité

### Https

Expliquer le fonctionnement de https crypto

Génération certificat (clé)

Captures WireShark

### Authentification par Token

Fonctionnement

Transmission client -> serveur / validation au niveau du serveur

Utilisation des bibliothèques

### Offuscation de l’exécutable

Reverse ingeneering , protection de l’application

### Autres préventions

* Limiter le nombre de connections
* Limiter la taille des requêtes
* Gestion des paramètres non valides

# Plan de réponse aux menaces

## Révision de sécurité

### Fuzzing

### Man in the middle

### Deny of service

### Reverse engineering

## Plan de réponse

## Archivage ???

# Conclusion

Compétences acquises, et l’importance de SDL dans le developement applicatif.