Présentation d'un cours de programmation sécurisée

Olivier Levillain

Agence nationale de la sécurité de systèmes d'information

15 décembre 2017

Plan

Présentation des modules

Écriture d'un parser MiniPNG

Présentation des modules

Écriture d'un parser MiniPNG

Modules enseignés

Cours sur la programmation sécurisée

- Programmation orientée sécurité (POS) à l'université Paris Sud
  - cours optionnel dans un M2
  - 18 heures
  - depuis 2014

- Programmation sécurisée à l'INSA de Rennes
  - option en 2^{e} année du cycle d'ingénieur
  - 26 heures
  - depuis 2017

Présentation des contenus (1/2)

- Cours introductif sur les enjeux du développement
  - en insistant sur la responsabilité des développeurs !

Présentation des contenus (1/2)

- Cours introductif sur les enjeux du développement
  - en insistant sur la responsabilité des développeurs !
- Cours et TP sur les vulnérabilités classiques
  - injections en tout genre
  - problèmes de logique
  - débordements de tampon
  - *race conditions*
  - *timing attacks*
  - *string format attacks*
  - *directory traversal*
  - (problèmes liés à la cryptographie)
  - ...

Présentation des contenus (2/2)

- Cours/conférence *Mind your languages*

Présentation des contenus (2/2)

- Cours/conférence *Mind your languages*
- Cours sur les bonnes pratiques de développement
  - prise en compte de la sécurité
  - nettoyage des entrées utilisateur
  - qualité du code (lisibilité, maintenabilité)
  - tests (y.c. négatifs / non régression)
  - prise en compte du déploiement et de l'environnement d'exécution
  - ...

Validation des modules

Projet bibliographique (seulement à l'INSA)

- un mois pour étudier une faille et sa correction
  - *Lucky 13*
  - *Ticketbleed*
  - *DirtyCOW*
  - ...
- travail en binôme
- présentation de 20 minutes + questions

Validation des modules

Projet bibliographique (seulement à l'INSA)

- un mois pour étudier une faille et sa correction
  - *Lucky 13*
  - *Ticketbleed*
  - *DirtyCOW*
  - ...
- travail en binôme
- présentation de 20 minutes + questions

Examen écrit

Validation des modules

Projet bibliographique (seulement à l'INSA)

- un mois pour étudier une faille et sa correction
  - *Lucky 13*
  - *Ticketbleed*
  - *DirtyCOW*
  - ...
- travail en binôme
- présentation de 20 minutes + questions

Examen écrit

TP noté sur l'écriture d'un *parser*

- format d'image inspiré de PNG
- spécification mal écrite
- une semaine pour terminer le TP

Présentation des modules

Écriture d'un parser MiniPNG

Présentation générale du TP noté

Objectif : écrire un *parser* d'images au format MiniPNG

Présentation générale du TP noté

Objectif : écrire un *parser* d'images au format MiniPNG

Découpage du TP

1. écriture du *parser* de la structure d'images en noir et blanc
2. afficher les images lues
3. extension du *parser* à des images en couleurs

Présentation générale du TP noté

Objectif : écrire un *parser* d'images au format MiniPNG

Découpage du TP

1. écriture du *parser* de la structure d'images en noir et blanc
2. afficher les images lues
3. extension du *parser* à des images en couleurs

Compléments

- langage de développement assez libre (mais à valider)
- bien sûr, il faut réfléchir à la sécurité !
- il faut expliciter les hypothèses qu'ils font par rapport à la spécification

Spécification de MiniPNG (1/3)

Une image au format MiniPNG est constituée

- d'un marqueur `Mini-PNG`
- d'une liste de blocs

Spécification de MiniPNG (1/3)

Une image au format MiniPNG est constituée

- d'un marqueur `Mini-PNG`
- d'une liste de blocs

Format d'un bloc

Offset	**Nom du champ**	**Taille du champ**
0	Type de bloc	1 octet
1	Longueur du bloc *l*	4 octets
5	Contenu du bloc	*l* octets

Spécification de MiniPNG (2/3)

Une image en noir et blanc contient

- un bloc d'en-tête `H`
- d'éventuels blocs de commentaires `C`
- un ou plusieurs blocs de données `D`

Spécification de MiniPNG (2/3)

Une image en noir et blanc contient

- un bloc d'en-tête `H`
- d'éventuels blocs de commentaires `C`
- un ou plusieurs blocs de données `D`

Bloc `H`

Offset	**Nom du champ**	**Taille du champ**
0	Largeur de l'image	4 octets
4	Hauteur de l'image	4 octets
8	Type de pixels (0 = N&B, 1 = gris...)	1 octet

Spécification de MiniPNG (3/3)

Bloc `C`

- une simple chaîne de caractères ASCII affichables

Spécification de MiniPNG (3/3)

Bloc `C`

- une simple chaîne de caractères ASCII affichables

Bloc `D`

- le contenu des blocs `D` doivent être concaténés pour obtenir un *bitmap*
- en N&B, chaque pixel doit être codé sur un bit
- en niveau de gris, chaque pixel est codé sur un octet
- en couleurs, chaque pixel est codé sur 3 octets (RVB)
- avec une palette, jusqu'à 256 couleurs peuvent être définies

Fourniture de fichiers

Des fichiers *a priori* bons

- des fichiers avec 1 ou 2 blocs D (et éventuellement des commentaires)
- un fichier avec un bloc D de plus de 256 octets
- un fichier N&B de taille 13x7
- un fichier avec un bloc H à la fin
- des fichiers en niveau de gris ou en couleur (avec ou sans palette)

Fourniture de fichiers

Des fichiers *a priori* bons

- des fichiers avec 1 ou 2 blocs D (et éventuellement des commentaires)
- un fichier avec un bloc D de plus de 256 octets
- un fichier N&B de taille 13x7
- un fichier avec un bloc H à la fin
- des fichiers en niveau de gris ou en couleur (avec ou sans palette)

Des fichiers *a priori* mauvais

- marqueur invalide
- absence d'en-tête
- absence de données
- trop de données

Discussion / correction

- Discussion sur les hypothèses manquantes
- Proposition d'amélioration de la spécification

Discussion / correction

- Discussion sur les hypothèses manquantes
- Proposition d'amélioration de la spécification

- Description au tableau des fonctions à implémenter

Discussion / correction

- Discussion sur les hypothèses manquantes
- Proposition d'amélioration de la spécification

- Description au tableau des fonctions à implémenter
- Écriture du code de manière simple et lisible

Discussion / correction

- Discussion sur les hypothèses manquantes
- Proposition d'amélioration de la spécification

- Description au tableau des fonctions à implémenter
- Écriture du code de manière simple et lisible
- Description d'implémentations vulnérables

Discussion / correction

- Discussion sur les hypothèses manquantes
- Proposition d'amélioration de la spécification

- Description au tableau des fonctions à implémenter
- Écriture du code de manière simple et lisible
- Description d'implémentations vulnérables
- En fonction du temps, digression sur PDF...

Quelques points croustillants (1/2)

Échauffement

- quelle *endianness* ?
- quid d'un bloc de plus de 256 octets ?

Quelques points croustillants (1/2)

Échauffement

- quelle *endianness* ?
- quid d'un bloc de plus de 256 octets ?

Sur l'ordre des blocs. Que faire avec...

Quelques points croustillants (1/2)

Échauffement

- quelle *endianness* ?
- quid d'un bloc de plus de 256 octets ?

Sur l'ordre des blocs. Que faire avec...

- un fichier sans bloc `H`
  - un élève avait prévu des valeur par défaut ? !

Quelques points croustillants (1/2)

Échauffement

- quelle *endianness* ?
- quid d'un bloc de plus de 256 octets ?

Sur l'ordre des blocs. Que faire avec...

- un fichier sans bloc `H`
  - un élève avait prévu des valeur par défaut ? !
- un fichier avec plusieurs blocs `H`

Quelques points croustillants (1/2)

Échauffement

- quelle *endianness* ?
- quid d'un bloc de plus de 256 octets ?

Sur l'ordre des blocs. Que faire avec...

- un fichier sans bloc `H`
  - un élève avait prévu des valeur par défaut ? !
- un fichier avec plusieurs blocs `H`
- un fichier avec un bloc `H` après un bloc `D`

Quelques points croustillants (1/2)

Échauffement

- quelle *endianness* ?
- quid d'un bloc de plus de 256 octets ?

Sur l'ordre des blocs. Que faire avec...

- un fichier sans bloc `H`
  - un élève avait prévu des valeur par défaut ? !
- un fichier avec plusieurs blocs `H`
- un fichier avec un bloc `H` après un bloc `D`
- un fichier avec un bloc de type inconnu

Quelques points croustillants (1/2)

Et le contenu ?

Quelques points croustillants (1/2)

Et le contenu ?

- un bloc `H` fait toujours 9 octets. Pourquoi le vérifier ?

Quelques points croustillants (1/2)

Et le contenu ?

- un bloc `H` fait toujours 9 octets. Pourquoi le vérifier ?
- quid d'un fichier auquel il manque des données
  - un élève proposait des pixels à 0 par défaut ? !

Quelques points croustillants (1/2)

Et le contenu ?

- un bloc `H` fait toujours 9 octets. Pourquoi le vérifier ?
- quid d'un fichier auquel il manque des données
  - un élève proposait des pixels à 0 par défaut ? !
- quid d'une longueur négative ?

Quelques points croustillants (1/2)

Et le contenu ?

- un bloc `H` fait toujours 9 octets. Pourquoi le vérifier ?
- quid d'un fichier auquel il manque des données
  - un élève proposait des pixels à 0 par défaut ? !
- quid d'une longueur négative ?
- que faire d'un commentaire non ASCII ?

Quelques points croustillants (1/2)

Et le contenu ?

- un bloc `H` fait toujours 9 octets. Pourquoi le vérifier ?
- quid d'un fichier auquel il manque des données
  - un élève proposait des pixels à 0 par défaut ? !
- quid d'une longueur négative ?
- que faire d'un commentaire non ASCII ?
- quid d'une palette absente ou incomplète ?

Quelques points croustillants (1/2)

Et le contenu ?

- un bloc `H` fait toujours 9 octets. Pourquoi le vérifier ?
- quid d'un fichier auquel il manque des données
  - un élève proposait des pixels à 0 par défaut ? !
- quid d'une longueur négative ?
- que faire d'un commentaire non ASCII ?
- quid d'une palette absente ou incomplète ?
- quid d'une image noir et blanc 13 par 7 ?

Quelques implémentations

- `minipng.py`
- `minipng-oups.py`
- `minipng-funambule.c`

Questions ?

Merci de votre attention

olivier.levillain@ssi.gouv.fr

Les supports de cours, les exemples et les implémentations sont à votre disposition sur simple demande