Commencé le	mardi 19 janvier 2021, 14:02
État	Terminé
Terminé le	mardi 19 janvier 2021, 15:00
Temps mis	58 min 26 s
Note	Pas encore évalué

Question **1**

Terminer

Noté sur 6,00

Expliquez comment fonctionnent les attaques par buffer overflow. Par ailleurs, quels changements sont introduits dans cette famille d'attaques par l'approche ROP (Return-Oriented Programming) ?

Les attaques par *buffer overflow* consistent à envoyer énormément d'informations dans un buffer pour le faire déborder et corrompre les zones mémoires adjacentes, qui se font réécrire par le surplus d'informations qui n'ont pas pu être écrites dans le buffer.

L'approche ROP est un cas particulier de l'attaque Return-to-libc, et cherche à provoquer un stack buffer overflow. L'attaquant va utiliser des gadgets (morceaux de code) de la librairie C, et va écrire les adresses des gadgets qui l'intéressent dans la stack. Ces adresses doivent être écrites dans un ordre précis afin d'obtenir le résultat voulu, en remplaçant les valeurs des "return" de la stack par les adresses des gadgets. Cette attaque implique de se reposer entièrement sur la librairie C (pas d'injection de code), et de la connaître. De plus, ce type d'attaque reposant sur les adresses est facilement vaincu par de l'ASLR. Elle permet cependant de contourner les zones mémoires protégées NX : Writable or Executable.

Question **2**

Terminer

Noté sur 3,00

Le fuzzing peut-il servir à tester des protocoles cryptographiques ? Justifiez.

Le fuzzing sert à tester la défense de son code et existe en 2 méthodes :

- Fuzzing par mutation : il s'agit de la méthode "simple" de fuzzing, qui est assez simple à implémenter. Le code va injecter plusieurs entrées (inputs) aléatoires dans le programme à tester jusqu'à ce qu'il en trouve une qui soit valide, puis il va créer des variantes/mutations de ce résultat et les réinjecter à nouveau.

- Fuzzing par génération : il s'agit de la méthode "intelligente" de fuzzing, plus difficile à mettre en place. Cette fois, le code va générer des entrées (inputs) qui correspondent au format d'entrée valable, mais des anomalies seront créées pour une entrée à chaque endroit possible.

Question **3**

Terminer

Noté sur 3,00

Expliquez ce qu'est un chiffre à substitution polyalphabétique. Donnez un exemple. Quelle est la motivation derrière la conception de telles techniques ?

Le chiffre à substitution polyalphabétique consiste à chiffrer un message donné en appliquant des règles différentes pour chaque lettre :

- à chaque lettre va correspondre un alphabet particulier

- et chaque lettre va disposer de son propre principe de permutation.

Un exemple de tel chiffre est celui de Vigenere.

La motivation derrière les différentes méthodes de chiffrement qui existent est de sécuriser des messages/informations en les chiffrant. L'objectif est que l'information soit la plus difficile possible à déchiffrer. Seul le destinataire doit être capable de déchiffrer le message, grâce à une clé.

Question **4**

Terminer

Noté sur 3,00

Le principe du moindre privilège est central dans les architectures logicielles de sécurité. Que spécifie-t-il ? Illustrez avec des exemples.

Le principe du moindre privilège consiste à attribuer le moins de privilèges possibles à un utilisateur. Ainsi, il ne disposera que des droits qui lui sont absolument nécessaires pour utiliser le logiciel.

Pour illustrer, prenons un forum disposant de deux types de rôles : modérateur et utilisateur. Un utilisateur pourra voir les messages postés par les autres utilisateurs sans pouvoir les modifier, tandis qu'un modérateur va pouvoir supprimer les messages ne respectant pas les règles établies. Il est indispensable qu'un utilisateur de dispose pas du privilège de suppression de message, car sinon n'importe quel utilisateur pourrait venir et supprimer des messages importants.

Question **5**

Non répondue

Noté sur 3,00

Quel est l'intérêt du reverse engineering par rapport à d'autres techniques de test de sécurité ? Expliquez et justifiez.

Question **6**

Terminer

Noté sur 2,00

Expliquez ce qu'est l'ASLR et son intérêt.

L'*Adress Space Layout Randomization* consiste à faire changer à chaque exécution du code l'adresse des éléments (pointeurs, variables...). Ceci offre une défense contre les types d'attaques ciblant la mémoire comme ROP (*Return Oriented Programming*) ou JOP (*Jump Oriented Programming*). En effet, les adresses changeant tout le temps, il devient extrêmement difficile pour l'attaquant de savoir où se trouvent les éléments qu'il souhaite corrompre.
Cependant, ceci peut devenir contraignant au niveau du code car il faut faire attention à ce qu'il ne soit pas dépendant des adresses.

Question **7**

Partiellement correct

Note de 0,50 sur 2,00

Indiquez quelles affirmations sont vraies concernant le fuzzing :

Veuillez choisir au moins une réponse :

- [x] a. *je peux découvrir des erreurs différentes si j'exécute plusieurs fuzzers sur un même programme* ✔
- [ ] b. *deux exécutions d'un même fuzzer sur un même programme peuvent découvrir des erreurs différentes*
- [ ] c. *on peut mesurer l'efficacité du fuzzing effectué en lignes de code testées*
- [ ] d. *si un fuzzer a tourné pendant 5 heures sur un programme, il n'y a plus aucune chance qu'il trouve de nouvelle vulnérabilité*
- [ ] e. *on peut mesurer l'efficacité du fuzzing effectué en nombre de branchements empruntés*
- [ ] f. *le fuzzing détecte exhaustivement toutes les vulnérabilités d'un programme*

Votre réponse est partiellement correcte.

Vous en avez sélectionné correctement 1.
Les réponses correctes sont : *je peux découvrir des erreurs différentes si j'exécute plusieurs fuzzers sur un même programme, deux exécutions d'un même fuzzer sur un même programme peuvent découvrir des erreurs différentes, on peut mesurer l'efficacité du fuzzing effectué en lignes de code testées, on peut mesurer l'efficacité du fuzzing effectué en nombre de branchements empruntés*

Question **8**

Non répondue

Noté sur 2,00

Indiquez quelles affirmations sont vraies concernant les attaques par buffer overflow :

Veuillez choisir au moins une réponse :

- [ ] a. *l'attaque fonctionne en modifiant l'adresse stockée au sommet du tas*
- [ ] b. *un buffer overflow ne fonctionne que si le shellcode est plus petit que le buffer où il est injecté*
- [ ] c. *l'attaque fonctionne en modifiant l'adresse stockée à l'adresse désignée par le stack pointer*
- [ ] d. *l'attaque fonctionne en modifiant l'adresse stockée à l'adresse désignée par un décalage connu à partir du base pointer de la trame courante*
- [ ] e. *une attaque par buffer overflow peut se dérouler avec succès même si on a introduit un mécanisme Writable XOR Executable (NX)*
- [ ] f. *le mécanisme de canari vise à empêcher une attaque par buffer overflow mais peut être contourné dans certains cas*

Votre réponse est incorrecte.

Les réponses correctes sont : *l'attaque fonctionne en modifiant l'adresse stockée à l'adresse désignée par un décalage connu à partir du base pointer de la trame courante, le mécanisme de canari vise à empêcher une attaque par buffer overflow mais peut être contourné dans certains cas*

Question **9**

Partiellement correct

Note de 1,00 sur 2,00

Quelles affirmations sur les attaques par concurrence (race conditions) sont-elles vraies ?

Veuillez choisir au moins une réponse :

- a. *une attaque par concurrence est systématiquement reproductible lors des tests*
- b. *une attaque par concurrence peut permettre de contourner une règle de contrôle d'accès à une ressource*
- c. *une attaque par concurrence ne peut arriver que si une ressource est partagée par deux processus* ✔
- d. *les attaques par concurrence durent au maximum quelques dixièmes de seconde, ce qui les rend quasiment inexploitables en pratique*
- e. *une attaque par concurrence ne peut se dérouler qu'en cas de conflit entre une opération de lecture et une opération d'écriture*
- f. *une attaque par concurrence ne fonctionnera que sur un processeur à cœurs multiples supportant des threads véritables*

Votre réponse est partiellement correcte.

Vous en avez sélectionné correctement 1.
Les réponses correctes sont : *une attaque par concurrence ne peut arriver que si une ressource est partagée par deux processus, une attaque par concurrence peut permettre de contourner une règle de contrôle d'accès à une ressource*

Question **10**

Partiellement correct

Note de 0,67 sur 2,00

Quelles affirmations suivantes pouvez-vous valider concernant le chiffrement symétrique par bloc ?

Veuillez choisir au moins une réponse :

- a. *le chaînage n'est pas nécessaire si le texte clair n'est constitué que d'un octet*
- b. *l'utilisation d'un tel algorithme nécessite deux clés, une publique pour chiffrer les données, une privée pour déchiffrer les messages envoyés par les correspondants*
- c. *un algorithme de chiffrement par bloc combine permutations et substitutions* ✔
- d. *l'utilisation d'un tel algorithme nécessite deux clés, une privée pour chiffrer les données, une publique pour déchiffrer les messages envoyés par les correspondants* ✖
- e. *le mode de chaînage ECB offre toujours une protection suffisante*
- f. *un algorithme de chiffrement par bloc vise à assurer confusion et diffusion* ✔

Votre réponse est partiellement correcte.

Vous en avez sélectionné correctement 2.
Les réponses correctes sont : *un algorithme de chiffrement par bloc combine permutations et substitutions, un algorithme de chiffrement par bloc vise à assurer confusion et diffusion, le chaînage n'est pas nécessaire si le texte clair n'est constitué que d'un octet*

Question **11**

Correct

Note de 2,00 sur 2,00

Qu'appelle-t-on un labyrinthe dans un système de fichiers (file maze) et quelle est son utilité ?

Veuillez choisir au moins une réponse :

- a. *il s'agit d'un mécanisme de protection pour interdire la modification des chaînes de blocs du tas*
- b. *il s'agit d'un mécanisme d'attaque visant à compromettre l'intégrité du système de fichiers*
- c. *il s'agit d'un mécanisme d'attaque visant à compromettre l'intégrité de la pile*
- d. *il s'agit d'un mécanisme d'attaque par concurrence visant à en augmenter la fenêtre d'opportunité* ✔
- e. *il s'agit d'un mécanisme pour protéger l'intégrité du système de fichier*
- f. *il s'agit d'un mécanisme de chiffrement visant à obtenir la propriété de confusion*

Votre réponse est correcte.

La réponse correcte est : *il s'agit d'un mécanisme d'attaque par concurrence visant à en augmenter la fenêtre d'opportunité*

Question **12**

Partiellement correct

Note de 0,50 sur 2,00

Quelles affirmations suivantes sont vraies concernant l'utilisation de la cryptographie ?

Veuillez choisir au moins une réponse :

- a. *la signature électronique assure la non-répudiation d'origine et l'intégrité des données signées* ✔
- b. *la signature électronique s'appuie indifféremment sur la cryptographie symétrique ou asymétrique*
- c. *la propriété de résistance à la pré-image d'une fonction de hachage est le fait qu'il soit difficile, étant donné un entier K, de retrouver M tel que h(M)=K*
- d. *la propriété de résistance à la collision d'une fonction de hachage est le fait qu'il soit difficile, étant donné un entier K, de retrouver M tel que h(M)=K*
- e. *la sécurité d'un mécanisme de chiffrement repose entièrement sur l'ignorance de la clé et des détails de l'algorithme employé qui doivent rester inconnus des utilisateurs* ✖
- f. *un algorithme de hachage cryptographique est un mécanisme de chiffrement avec des propriétés d'intégrité supplémentaires*

Votre réponse est partiellement correcte.

Vous en avez sélectionné correctement 1.
Les réponses correctes sont : *la signature électronique assure la non-répudiation d'origine et l'intégrité des données signées, la propriété de résistance à la pré-image d'une fonction de hachage est le fait qu'il soit difficile, étant donné un entier K, de retrouver M tel que h(M)=K*