République Tunisienne.Ministère de l'Enseignement Supérieur et de la Recherche Scientifique.Université de Sousse.Institut Supérieur des Sciences Appliquées et de Technologie de Sousse.	Sécurité des Logiciels et des Systèmes d'Information 7403(FI-GL2) DS (14-03-2023) Nombre de pages : 4 Enseignant : MTIBAA Riadh

FEUILLE DE QUESTIONS

I. NE PAS REPONDRE SUR CETTE FEUILLE DE QUESTIONS!!
II. REPONDRE SUR LA FEUILLE DE REPONSES!!!

1. Les mesures de sécurité des modules de COMPUTING des systèmes IoT concernent:
A. Le Domaine FOG.
B. Les interfaces matérielles entre les domaines FOG et SENSING.
C. Le domaine CLOUD.
D. Les interfaces logicielles entre les domaines CLOUD et FOG.
E. Le Domaine SENSING.

2. Le critère de sécurité FORWARD SECRECY pour le COMPUTING des systèmes IoT exige que:
A. Chaque client IoT se charge de la validation de l'intégrité des trames reçues avant de procéder au FORWARDING de ces dernières aux autres entités IoT.
B. Les routeurs de la couche FOG implémentent une cryptographie adéquate à l'opération de FORWARDING des trames reçues.
C. Chaque équipement IoT est responsable de la confidentialité des trames qu'il retransmet aux autres noeuds IoT.
D. Les interfaces logicielles avec les CAPTORS et ACTUATORS du domaine SENSING doivent se charger du FORWARDING confidentiel des trames à échanger avec les autres entités CLOUD.
E. Une fois une entité IoT se déconnecte du réseau IoT, elle ne pourrait plus comprendre les communications entre les autres entités.

3. L'origine des vulnérabilités les plus graves qui visent les applications IoT est:
A. Le service TCP/IP lancé par les noyaux des serveurs des applications IoT.
B. Le HYPERVISOR qui interface les VMs dédiées aux applications IoT par rapport aux ressources matérielles des serveurs physiques qui les déploient.
C. Les composantes matérielles des serveurs qui hébergent les VMs des applications IoT.
D. Le HYPERVISOR qui gère le FORWARD des trames destinées aux applications IoT.
E. La couche FOG qui passe les trames transmises par les capteurs IoT aux serveurs des applications IoT.

4. L'attaque de type HIDDEN-CHANNEL est causée par:
A. Le partage des ressources physiques, telles que mémoire et cache, des serveurs CLOUD qui hébergent les VMs du COMPUTING IoT.
B. Le partage des routeurs spécialisés du domaine FOG lors de la collecte des trames transmises par les capteurs IoT.
C. Les failles de sécurité des pilotes qui contrôlent les composantes électroniques du domaine Sensing.
D. Les sessions TCP établies entre les VMs du COMPUTING IoT pour faire du TUNNELING.
E. La configuration du CLUSTERING utilisée par le fournisseur du service CLOUD pour le compte des applications IoT.

5. Durant HIDDEN-CHANNEL ATTACK, l'attaquant essaie de réussir le MAPPING TARGET VM pour:
A. Saboter tous les serveurs du POD de la VM cible.
B. Deviner l'@IP interne du CLUSTER de la VM cible.
C. Usurper l'@IP externe du POD de la VM cible.
D. Déterminer surtout le CLUSTER de la VM cible.

E. Exploiter une vulnérabilité du serveur CLOUD COMPUTING d'une application IoT particulière.

6. Dans la deuxième étape de HIDDEN-CHANNEL ATTACK, l'attaquant:

A. Garantit que sa VM malicieuse est dans le même CLUSTER que la VM de l'application IoT victime.

B. Tente, avant tout, de s'assurer que sa VM malicieuse est affectée par le CLOUD PROVIDER SCHEDULING ALGORITHM au même serveur CLOUD COMPUTING de la VM de l'application IoT visée.

C. S'assure que sa VM malicieuse est dans la même ZONE que la VM de l'application IoT victime.

D. Doit saboter l'algorithme CLOUD PROVIDER SCHEDULING ALGORITHM pour obtenir une VM malicieuse dans le même CLUSTER que la VM victime, mais jamais dans le même POD.

E. Pourrait utiliser la commande TRACEROUTE pour déterminer le nombre de HOPs entre son serveur de VM malicieuse et le serveur d'application IoT victime, à fin d'ajuster la valeur TTL des paquets transmis.

7. Pour sécuriser un CLUSTER de CLOUD COMPUTING contre HIDDEN-CHANNEL ATTACK on pourrait:

A. Utiliser la stratégie NOISY DATA ACCESS TIME, qui altère les valeurs de FETCH DATA TIME, pour entraver l'analyse malicieuse des accés mémoire et cache des VMs victimes.

B. Implémenter la stratégie de NOISY DATA ACCESS TIME pour déranger l'attaquant lors de l'altération des accés mémoire et cache des VMs cibles et ceci en déplaçant les octets du FETCH DATA TIME.

C. Utilisant une technique de LIMITING CACHE SWITCHING RATE qui permet de protéger les VMs attaquées en suivant une procédure de FLUSH de la mémoire et cache partagés.

D. Exécuter une procédure de FALSE RESSOURCE ADVERTISING pour déclarer de faux états et configurations d'exploitation de ressources à fin d'inciter l'HYPERVISOR à déplacer les VMs malicieuses vers des serveurs CLOUD HONEYPOTS utilisés pour piéger les attaquants.

E. Contre-attaquer les serveurs CLOUD COMPUTING malicieux par VM MIGRATION ATTACK pour les planter.

8. l'HYPERVISOR XEN est vulnérable à THEFT-OF-SERVICE-ATTACK car:

A. Au moment du SAMPLING, la VM malicieuse avait épuisé toute la durée de 10 ms allouée normalement à la VM victime.

B. L'attaquant corrompe les cumuls des durées d'exploitation des ressources du CLOUD COMPUTING SERVER par les différentes VMs, respectivement, pour que la VM malicieuse soit favorisée lors de l'allocation des ressources.

C. Au moment du SAMPLING, XEN suppose que la VM en cours d'exécution avait épuisé toute la durée de 10 ms allouée.

D. La VM malicieuse vole les données lues et écrites par la VM victime à partir du cache et mémoire partagés.

E. La VM malicieuse remplace la VM ciblée en obligeant l'HYPERVISOR à faire migrer cette dernière vers des serveurs CLOUD COMPUTING d'applications Iot vulnérables.

9. Soit $e_b(x)$ la fonction de chiffrement RSA ; b est l'exposant publique, m le module, et $x \in \mathbb{Z}_m$. Le RSA est un cryptosystème homomorphique car:

A. $\exists$ une loi de composition $\otimes$ sur $\mathbb{Z}_m$ t.q. $\forall x_1, x_2 \in \mathbb{Z}_m$ on a $e_b(x_1 \otimes x_2) = e_b(x_1) \otimes e_b(x_2)$.

B. $\forall x_1, x_2 \in \mathbb{Z}_m$ on a $e_b(x_1 x_2) = e_b(x_1)\, e_b(x_2)$.

C. $\forall x_1, x_2 \in \mathbb{Z}_m$, $\forall$ loi de composition $\otimes$ sur $\mathbb{Z}_m$ on a $e_b(x_1 \otimes x_2) = e_b(x_1) \otimes e_b(x_2)$.

D. $\exists$deux lois de compositions $\otimes$ et Σ sur $\mathbb{Z}_m$ t.q. $\forall x_1, x_2 \in \mathbb{Z}_m$ on a $e_b(x_1 \otimes x_2) = e_b(x_1)\, \Sigma e_b(x_2)$.

E. $\forall x_1, x_2 \in \mathbb{Z}_m$ on a $x_1 x_2 = e_b(e_b(x_1)\, e_b(x_2))$.

10. Soient e_{pb} et d_{pr} les algorithmes de chiffrement et de déchiffrement d'un cryptosystème homomorphique, respectivement. Les lois de composition sur les codes clairs et chiffrés du cryptosystème sont notées Σ et $\oplus$, respectivement. On considère un algorithme A qui

doit s'appliquer, sur le CLOUD, aux chiffrements y_1 et y_2 des deux codes clairs x_1 et x_2, respectivement (seuls y_1 et y_2 sont stockés sur le CLOUD!). Sachant que $A(y_1) = z_1$ et $A(y_2) = z_2$, la valeur de $A(e_{pk}(x_1 \Sigma x_2))$ est calculée par le serveur CLOUD COMPUTING comme suit:

A. $A\left(e_{pk}(d_{pr}\left(y_1\right) \Sigma d_{pr}\left(y_2\right))\right)$. | B. $e_{pk}(A\left(d_{pr}\left(y_1\right) \Sigma d_{pr}\left(y_2\right)\right))$. | C. $A\left(d_{pr}\left(y_1\right) \Sigma d_{pr}\left(y_2\right)\right)$.

D. $z_1 \Sigma z_2$. | E. $z_1 \otimes z_2$.

11. On considère le cryptosystème homomorphique RSA de paramètres publiques $b = 5$ et $m = 221$, déjà connus par le serveur de CLOUD COMPUTING. Maintenant on considère deux codes clairs $x_1 = 8$ et $x_2 = 13$. La différence, calculée par le serveur CLOUD COMPUTING, entre les chiffrements de x_1 et x_2 par RSA est égale à:

A. 91. | B. 47. | C. -5. | D. 216. | E. 190.

12. L'algorithme FHE QUICK SORT est moins sécurisé que celui FHE COMPARISON SORTING car:

A. Il utilise des opérations de SWAP.

B. Sa complexité est relativement petite.

C. Il est récursif.

D. Il est plus vulnérable au CHOSEN-PLAINTEXT ATTACK.

E. Le PARTITION SORTING SERVER ne permet pas toujours de faire le tri d'un ensemble de messages d'attaques lors de la CPA INDISTINGUISHABILITY EXPERIMENT.

13. Dans une BLOCKCHAIN, les FORKs correspodent à:

A. Des transactions en boucles infinies qui vident les comptes de certains NODES de leur CRYPTOCURRENCY suite à l'exécution exécessive de fonctions de WITHDRAWAL.

B. La validation de plusieurs BLOCKS qui jouent le rôle de BLOCK GENESIS chacun, suite à un conflit dans la résolution de consensus PoW.

C. Lancement de nouveaux processus de consensus sans attendre la fin de consensus antérieurs inachevés.

D. L'apparition de plusieurs sous-chaines reliées au BLOCK GENESIS suite à l'inconsistance de la condition d'intégrité entre les anciens et les nouveaux BLOCKS.

E. Des bugs qui permettent à un attaquant d'installer des SMART CONTRACTS utilisant des TOKENs à durée de validité infinie.

14. Un BLOCK risque de devenir STALE lorsque:

A. Il est relativement très proche du BLOCK GENESIS, et donc il est tellement ancien que son empreinte numérique n'est plus utile pour vérifier l'intégrité des nouveaux BLOCKS.

B. Il est généré par un NODE peu actif et qui ne gagne presque jamais dans les consensus PoW.

C. Son traitement de création et validation est entravé par d'autres BLOCKS générés presque au même moment que lui ; il ne pourrait être rajouté à la chaine principale.

D. Il y a une majorité de NODES dont les ressources sont épuisées par des agissement malicieux et donc le temps nécessaire pour valider des BLOCKS est pratiquement infini.

E. Un attaquant tente d'exécuter plusieurs transactions au même moment pour réaliser DOUBLE-SPENDING par exemple.

15. Le consensus PoW pourrait être vulnérable puisque:

A. Plusieurs NODES peuvent gagner dans la compétition en trouvant la solution du puzzle.

B. Le hashage répétitif pour résoudre le puzzle est sanctionné par un ALLOCATED TIME PROCESSING réduit, ainsi l'interruption du consensus au niveau d'un NODE implique la perte du REWARD au moins.

C. Il consomme une grande partie des ressources de COMPUTING du NODE, ainsi il est possible que ce dernier voit son STAKE réduit à zéro.

D. Il occupe tellement les processeurs des NODES que ces derniers ne peuvent plus indiquer un NETWORK TIME raisonnable, ce qui augmenterait le nombre des BLOCKS STALE.

E. Il est lourd à maintenir vu que le puzzle à résoudre est modifié à chaque fois qu'il y ait un nouveau consensus lancé par l'apparition de nouveaux BLOCKS à rajouter à la chaine.

16. Dans une FINNEY ATTACK:

A. La variabilité du NETWORK TIME est accentuée pour éclipser la majorité des NODES qui refusent de communiquer des BLOCKs de transactions.

B. Les NODES sont contrôlés par des attaquants qui épuisent leurs ressources en les obligeants à participer à de faux consensus PoW.

C. La balance de 51% des NODES du réseau attaqué est modifiée illicitement à l'aide de SMART CONTRACTS utilisant des TOKENS ERC20 avec bug. Ainsi les STAKEs ne sont plus corrects et donc les vrais WINNERS ne peuvent plus exécuter leurs transactions.

D. L'attaquant diffuse tout le temps des BLOCKS de transactions fictives pour entraver l'ajout des BLOCKS des autres NODES à la chaine principale. Par conséquent une majorité de NODES perd les REWARDS des consensus et les transactions. Ainsi ces NODES victimes sont isolés dans la BLOCKCHAIN.

E. L'attaquant diffuse un BLOCK de transaction de versement de CRYPTOCURRENCY dans l'un de ses comptes. Ce BLOCK sert à invalider un BLOCK précédemment transmis contenant une transaction de transfert de CRYPTOCURRENCY vers un fournisseur de marchandises ; ainsi l'attaquant récupère sa monnaie tout avec la marchandise.

17. Dans une TIMEJACKING ATTACK, les BLOCKS sont:

A. Rejetés car le MINIMUM des NETWORK TIMEs des voisins dépasse 70 minutes.

B. Rejetés car le MAXIMUM des NETWORK TIMEs des voisins dépasse 70 minutes.

C. Rejetés car la moyenne des NETWORK TIMEs des voisins dépasse 70 minutes.

D. Retardés car le MEDIAN des NETWORK TIMEs des voisins dépasse 70 minutes.

E. Rejetés car le MEDIAN des NETWORK TIMEs des voisins dépasse 70 minutes.

18. Le GAS est épuisé malicieusement en:

A. Augmentant le nombre des adresses des NODES exigeants REFUNDS.

B. Diminuant le nombre des NODES capables de gagner des REWARDS.

C. Isolant la majorité des NODES qui ont d'important STAKES.

D. Créant beaucoup de fausses adresses de NODES participants dans les consensus.

E. Epuisant les ressources d'une majorité de NODES par des transactions malicieuses.

19. Une OVERFLOW ATTACK consiste à:

A. Exécuter des boucles infinies de hachage et de vérification d'intégrité des entêtes des BLOCKS.

B. Altérer les adresses mémoires qui pointent vers les BLOCKS de la chaine.

C. Epuiser la totalité de la mémoire d'un NODE.

D. Dépasser malicieusement le montant maximal d'ETHER d'un compte (2^{256}) pour que le système de la BLOCKCHAIN le remet à zéro.

E. Toujours dépasser le NETWORK TIME pour achever la validation d'un BLOCKS reçus.

20. Dans une ECLIPSE ATTACK:

A. Le NODE cible est isolé du réseau des NODES légitimes en le connectant à un groupe de NODES de CRYPTOJACKING. Ces derniers épuisent les ressources du NODE victime qui se plante puis redémarre pour se connecter à d'autres NODES malicieux.

B. Le NODE compromis sert à isoler tout un réseau de NODES légitimes en les connectant à d'autres NODES malicieux. Ces derniers bombardent le NODE compromis par GARBAGE DATA qu'il retransmet aux NODES victimes qui peuvent se planter et redémarrer.

C. Le NODE cible est isolé du réseau des NODES légitimes en le connectant à un grand ensemble de NODES compromis. Ces derniers bombardent le NODE VICTIME par GARBAGE DATA pour l'obliger à redémarrer. Lors du redémarrage de nouvelles connexions vers d'autres NODES malicieux peuvent êtres établies.

D. Le NODE cible est isolé du réseau des NODES légitimes en le connectant à plusieurs NODES infectés par des malwares qui épuisent toute la bande passante.

E. Le NODE cible est isolé du réseau des NODES légitimes en le bombardant par un flux infini de BLOCKS fictifs.

FIN