Projet Cryptographie

Conception non Sécurisée

Membres du groupe 1 :

Sébastien AHMANY DAN-BAIBE

Khadim FALL

Mouhamadou LEYE

Ndèye Awa ZONGO

Maimouna NDIAYE

Plan

Introduction

Parmi les nombreux aspects de la cybersécurité, la sécurité logicielle représente la résilience des logiciels applications contre les attaques externes. En particulier, les applications Web sont exposées au monde extérieur et constituent donc un point d'attaque privilégié par les pirates. Malgré le risque élevé des attaques contre les applications logicielles, les organisations ne reconnaissent pas toujours l'importance d'évaluer que le logiciel ne contient pas de failles de sécurité. Le problème est exacerbé par le manque d'outils automatiques solides, capables de trouver des vulnérabilités logicielles. La pertinence de telles faiblesses justifie l'intérêt des chercheurs et des entreprises à trouver des moyens d'évaluer que les logiciels ne contiennent pas de menaces. Nous allons nous intéresser à l'OWASP et au travail qu'il a effectué dans ce sens.

I. Qu'est-ce que OWASP ?

1. Définition

L'Open Web Application Security Project (OWASP) est une organisation à but non lucratif fondée en 2004 pour prévenir de manière proactive les attaques sur les applications web. Il s'agit du premier effort de normalisation des pratiques de développement sécurisé.

En 2001, l'OWASP n'était pas une organisation officielle, mais plutôt un collectif qui préconisait des pratiques de développement sécurisé.

Ce collectif a pris de l'ampleur et est devenu l'OWASP fondation en 2004, avec une norme éthique pour maintenir une neutralité et l'absence de pressions commerciales.

L'OWASP n'est réglementée par aucune entreprise. Elle propose un référentiel neutre permettant d'accompagner les entreprises dans le processus de sécurisation ou d'audit de sécurité.

2. Critères de classification du top 10 des vulnérabilités

Le Top Ten de l'OWASP fournit une base de référence avec une liste de contrôles à effectuer pour atténuer les risques les plus courants en matière de sécurité. Cette base de référence est également utilisée pour répondre à des normes

réglementaires plus strictes, tel que le Règlement Général pour la Protection des Données (RGPD) par exemple. Les

critères évalués par L'OWASP pour classer son top ten sont:

- Déclenchement de message d'erreur contenant des informations sensibles (CWE-209)

- Violation de limite de confiance (CWE-501)

- Protection insuffisante d'identifiants (CWE-522)

- Restriction des droits d'accès (CWE-101)

- Incompatibilité avec les contrôles effectués (CWE-401)

- Insécurité des données (CWE-502)

- Incompatibilité avec les normes (CWE-506)

II. Choix du sujet ?

1. Description et justification

Intitule: Missing Encryption of Sensitive Data

(Chiffrement manquant de données sensibles)

Le chiffrement manquant de données sensibles est une notion qui fait référence aux logiciels qui ne cryptent pas les informations sensibles ou critiques avant le stockage ou la transmission. L'absence d'un cryptage approprié des données fait perdre les garanties de confidentialité, d'intégrité et de responsabilité qu'offre un cryptage correctement mis en œuvre.

Comprendre les vulnérabilités courantes des applications Web aide les entreprises à mieux se préparer à protéger leurs données contre de potentielles attaques. Après une brève analyse du top 10 de OWASP, le sous thème de non chiffrement des données sensibles de la catégorie A04 a été retenu. En effet, nombreux sont les logiciels qui ne cryptent pas les informations sensibles ou critiques avant le stockage ou la transmission. Ceci représente un risque de sécurité car une personne sur le réseau d'une entreprise par exemple peut renifler les paquets de la connexion découvrir des données sensibles, mais peut aussi naviguer sur n'importe quel chemin allant de sa machine au serveur final. Il est donc intéressant de rappeler que le but premier de l'utilisation de la cryptographie est de garantir les quatre services fondamentaux de la sécurité des informations.

2. Facteurs

CWE est une liste des types de faiblesses logicielles et matérielles. Il sert d'instrument de mesure pour les outils de sécurité et de référence pour les efforts d'identification, d'atténuation et de prévention des faiblesses. Sa valeur est de 40.

Taux d'incidence maximal 24,19%, une évolution maximale inférieure à la moyenne

Taux d'incidence moyen 3,00% l'évolution moyenne est plutôt minime

Exploitation moyen pondéré : 6,46

Impact moyen pondéré : 6,78

Couverture maximale : 77,25 %

Couverture moyenne : 42,51 %

Nombre total d'occurrences : 262,407

Total des CVE : 2,691

3. Corrélation entre le thème et les concepts cryptographiques

La conception non sécurisée en générale et la sous-catégorie 'chiffrement manquant de données sensibles' se trouvent liées à quelques faiblesses que nous rencontrons couramment dans les logiciels et systèmes d'informations.

D'abord il est important de souligner les défaillances que peuvent véhiculés un mécanisme de protection d'un système d'information (CWE-693). Le mécanisme peut se révéler manquant, insuffisant ou ignoré. L'utilisation de la cryptographie classique illustre bien le problème. Le chiffrement de Cesar par exemple ne donne lieu dans le cas général qu'à 26! permutations possibles et la taille de l'espace des clés est bien suffisante. Ceci montre la non fiabilité du système. Pareillement au chiffrement de Cesar, le chiffrement affine n'est pas fiable car offre un nombre de clés déterminé par la fonction phi d'Euler (n – 1 si n est premier). Les chiffrements de Playfair, de Hill, etc. bien que polygrammiques et rendant plus difficile

l’analyse des fréquences n’offrent pas la fiabilité donc les systèmes ont besoin de nos jours.

Ensuite, la transmission en clair d’informations sensibles (CWE-312) est l’une des faiblesses représentants un risque important au niveau de la sécurité d’un logiciel. N'importe qui peut lire les informations en accédant au canal utilisé pour la communication. Bien que la connexion soit établie avec succès, elle n'est pas chiffrée et il est possible que toutes les données sensibles envoyées au serveur ou reçues de celui-ci soient lues par les mauvaises personnes. Cette faiblesse est due à l'absence d'une tactique de sécurité pendant la phase d'architecture et de conception. La solution au problème est donnée par l’intégration des principes cryptographiques aux processus de conception. Le protocole HTTPS par exemple permet de chiffrer les données échangées entre le navigateur de l’internaute et le site web empêchant l’espionnage (confidentialité) ou la modification (intégrité). HTTPS est une extension du protocole HTTP (le S pour Secured) avec une couche de chiffrement comme SSL ou TSL qui utilisent des options de chiffrement comme RSA qui fait partie de la cryptographie à clef publique. Cette dernière est une méthode de chiffrement qui repose sur l’utilisation d’une clé publique et d’une clé privée qui permettent respectivement de chiffrer et déchiffrer un message. Dans le cas du RSA le niveau de sécurité est relatif à la difficulté de factoriser de grands nombres premiers. Retrouver le texte en clair à partir d’une des clefs et du texte chiffré est supposé équivalent à la factorisation du produit des deux nombres premiers, générés aléatoirement au début de l’exécution de l’algorithme. Il existe bien d’autres algorithme de chiffrement asymétrique comme le Schémas de Rabin et le Schéma ElGamal utilisables à la fois pour les signatures numériques et pour le chiffrement.

Le stockage en texte clair d’informations sensibles (CWE-312) vient s’ajouter à la liste des faiblesses que peuvent présenter le logiciel. Comme les informations sont stockées en clair (non cryptées)

, les attaquants peuvent potentiellement les lire. Même si les informations sont codées d'une manière illisible par l'homme, certaines techniques peuvent déterminer quel codage est utilisé, puis décoder les informations. Une fois arivé sur le serveur, les informations sensibles doivent être hashées avant la sauvegarde en base de données. De nos jours nombreux sont les fonctions de hashage qui assurent les contraintes d'intégrité et de confidentialité. Il est important de noter qu'une fonction de hashage cryptographique se differencie d'une fonction de hashage standard par le fait qu'il est difficile de trouver un antécédant par la première. Une fonction de hashage à sens unique pourrait assurer le niveau la sécurité requise pour une application. Il est tout aussi important de noter que ces fonctions sont classées en deux catégories : les fonctions de hachage sans clé, dont la spécification requiert un seul paramètre d'entrée (un message); et les fonctions de hachage à clé, dont la spécification requiert deux entrées distinctes, un message et une clé secrète

III. Scenario d'attaques et de défenses

Scenario 1 :

Le processus de récupération d'informations d'identification peut inclure des questions et réponses interdit par le NIST 800-63b, l'OWASP ASVS. Les questions et les réponses ne peuvent pas être considérées comme des preuves d'identité car plus d'une personne peut connaître les réponses, c'est pourquoi elles sont interdites. Un tel code doit être supprimé et remplacé par une conception plus sûre.

L'attaque consiste deviner la réponse à la question sécrète posée à l'utilisateur (l'attaquant). Ce dernier se base sur certains concepts liés à la question pour établir une liste de possibles réponses en vue d'obtenir la bonne et accéder aux informations d'identifications.

La défense consiste en l'amélioration du processus de récupération de mot de passe notamment le chiffrement de la réponse à la question sécrète avant le

stockage, l'ajout d'un processus supplémentaire consistant à l'envoie d'un email à identifiant unique à l'adresse de l'utilisateur obligeant ce dernier au changement des informations concernant son identification.

Outils technologiques : php (symfony), mysql

Scenario 2 :

Le site Web de commerce électronique d'une chaîne de magasins n'est pas protégé contre les robots gérés par des revendeurs clandestins qui achètent des cartes vidéo haut de gamme pour les revendre sur des sites Web d'enchères. Cela crée une publicité terrible pour les fabricants de cartes vidéo et les propriétaires de chaînes de magasins et entretient l'animosité des passionnés qui ne peuvent obtenir ces cartes à aucun prix. Une conception anti-bots minutieuse et des règles de logique de domaine, telles que les achats effectués dans les quelques secondes suivant la disponibilité, pourraient permettre d'identifier les achats inauthentiques et de rejeter ces transactions.

L'attaque consiste en la conception d'un bot interagissant automatiquement avec le serveur de l'application attaquée. Le rôle de celui-ci est d'effectuer des achats sur le site de commerce durant les secondes qui suivent la disponibilité d'un produit.

La défense consiste en la configuration de la protection bot pour web application. Un mécanisme sera mis en place pour la gestion des achats qui surviennent quelques secondes après la mise en ligne d'un produit. Il sera nécessaire de confirmer l'achat par le fait de s'assurer que c'est bien une personne et non un programme qui tente d'effectuer un achat. Un Captcha sera mis en place pour la vérification.

Outils technologiques : python, java (spring boot), mysql , php

Scenario 3 :

Attaque : Pour assurer la fluidité du trafic vers un site web, ils utilisent des cookies de session pour se souvenir d'un utilisateur pendant une période

limitée. C'est la seule façon de garantir une utilisation correcte des boutiques en ligne ou d'autres sites. Après tout, leur fonction dépend des différentes activités d'un client. Si le client se déplace de page en page, il enregistre les informations afin d'atteindre la destination souhaitée. Sur les sites de commerce électronique, il s'agit généralement du traitement du paiement et de la confirmation de la commande. Si un attaquant parvient à compromettre l'ordinateur d'un utilisateur, il pourra ainsi grâce aux cookies utilisés accéder aux informations d'identification de cet utilisateur.

Défense : les informations de stockage avant d'être stockées dans les cookies devraient être cryptées à l'aide de mécanisme de chiffrement adéquats. L'utilisateur quant à lui devrait préconiser l'utilisation de pare-feu, éviter d'aller sur certains sites douteux, ou encore de cliquer des liens dont la provenance est douteuse pour une meilleure protection de son ordinateur.

Scenario 4 :

Attaque : De nombreuses opérations sur les fichiers sont destinées à avoir lieu dans un répertoire restreint. En utilisant des éléments spéciaux tels que les séparateurs ".." et "/", les attaquants peuvent s'échapper en dehors de l'emplacement restreint pour accéder à des fichiers ou des répertoires qui se trouvent ailleurs sur le système. L'un des éléments spéciaux les plus courants est la séquence "../", qui dans la plupart des systèmes d'exploitation modernes est interprétée comme le répertoire parent de l'emplacement actuel.

Défense : dans la partie Architecture et conception utilisez une bibliothèque ou un framework approuvé qui ne permet pas à cette faiblesse de se produire ou fournit des constructions qui rendent cette faiblesse plus facile à éviter. Pour la mise en œuvre, utilisez une stratégie de validation des intrants « accepter le bien connu », c'est-à-dire, utilisez une liste d'intrants acceptables strictement conformes aux spécifications. Rejetez toute entrée qui n'est pas strictement conforme aux spécifications ou transformez-la en quelque chose qui le fait.

Scenario 5 :

Attaque : Les écoutes clandestines sont le résultat d'une interception du trafic réseau. Elles permettent à un attaquant d'obtenir des mots de passe, des numéros de carte bancaire et d'autres informations confidentielles qu'un utilisateur envoie sur le réseau. Elles peuvent être passives ou actives :

- Écoute clandestine passive – Un pirate détecte des informations en écoutant la transmission de messages sur le réseau.
- Écoute clandestine active – Un pirate s'empare activement d'informations en se faisant passer pour une unité amie et en envoyant des requêtes aux transmetteurs. On appelle cela sonder, scanner ou saboter.

  Il est souvent plus important de détecter des écoutes passives que des écoutes actives, car ces dernières exigent de l'attaquant qu'il apprenne à connaître les unités amies en effectuant préalablement des écoutes passives.

Défense :

- Utilisation d'un certificat électronique
- Mettre en place un cryptage fort pour protéger les données