Security Standards Council ™

Norme : DSS (Data Security Standard)
Condition : 6.6
Date : février 2008

Complément d'informations : Condition 6.6 clarifiant les révisions de code et les pare-feu d'applications

Publication : 15-04-2008

Généralités

La condition 6.6 PCI DSS propose deux options dont l'objectif est de traiter les menaces usuelles pesant sur les données des détenteurs de carte et de garantir que les données destinées aux applications Web provenant d'environnements non certifiés fassent l'objet d'un examen approfondi. Les dispositions détaillées permettant de satisfaire cette condition dépendent de l'implémentation spécifique relative à une application donnée.

L'analyse approfondie des compromissions de données des détenteurs de carte a démontré que les applications Web constituent généralement le point initial d'attaque à l'encontre des données des détenteurs de carte, notamment par l'intermédiaire des vulnérabilités de type « SQL injection ».

La condition 6.6 a pour objectif de garantir la protection des applications Web exposées au grand public contre la plupart des types usuels d'attaque malveillante. De nombreuses informations publiques sont disponibles sur les vulnérabilités des applications Web. Les principales vulnérabilités à prendre en considération sont décrites dans la condition 6.5. (La section Références contient d'autres sources d'informations concernant le test des applications Web.)

L'implémentation correcte des deux options offrira un système de défense multicouche optimal.

Le PCI SSC reconnaît que le déploiement des deux options, du fait de son coût et de sa complexité de mise en œuvre, peut s'avérer impossible. Par ailleurs, le déploiement de l'une ou l'autre option n'est parfois pas possible dans certaines situations (notamment en cas d'absence d'accès au code source). Il est toutefois possible d'appliquer au moins l'une ou l'autre des deux options décrites dans ce document et une implémentation appropriée peut satisfaire l'objectif de cette condition.

Ce document fournit l'assistance requise pour déterminer la meilleure option (cette dernière variant en fonction des produits utilisés), comprendre comment une organisation se procure ou développe ses applications Web, et tenir compte d'autres facteurs au sein de l'environnement.

Condition 6.6, option 1 – Révisions des codes d'application

L'option de révision des codes d'application ne requiert pas nécessairement la révision manuelle du code source. Il est possible d'étudier plusieurs solutions tout en gardant à l'esprit que l'objectif de la condition 6.6 vise à prévenir l'exploitation des vulnérabilités usuelles (telles que celles présentées dans la condition 6.5). Ces solutions, dynamiques et proactives, nécessitent la mise en œuvre spécifique d'un processus manuel ou automatique. Correctement implémentée, une ou plusieurs des quatre solutions suivantes peuvent satisfaire l'objectif de l'option 1 et garantir un niveau minimal de protection contre les menaces usuelles pesant sur les applications Web :

1. Révision manuelle du code source de l'application
2. Utilisation appropriée des outils automatisés (analyse) de l'analyseur du code source de l'application
3. Évaluation manuelle des vulnérabilités au niveau de la sécurité des applications Web

4. Utilisation appropriée des outils d'évaluation (analyse) des vulnérabilités au niveau de la sécurité des applications Web

Ces solutions doivent être conçues pour vérifier l'existence de vulnérabilités au niveau des applications Web, comme indiqué dans la section « Généralités » ci-dessus. L'évaluation des vulnérabilités se contente d'identifier et de signaler les vulnérabilités tandis qu'un test d'intrusion tente d'exploiter les vulnérabilités pour déterminer si un accès non autorisé ou toute autre activité malveillante est possible.

Les révisions/évaluations manuelles peuvent être exécutées par une ressource interne ou une tierce partie qualifiée. Dans tous les cas, la ou les personnes concernées doivent posséder les compétences et l'expérience requises pour comprendre le code source et/ou l'application Web, être en mesure de les évaluer, rechercher les vulnérabilités et interpréter les résultats obtenus. De la même façon, les individus utilisant des outils automatisés doivent posséder les compétences et connaissances requises pour configurer correctement l'outil et l'environnement de test, utiliser l'outil et évaluer les résultats.

Si ce travail est confié à des ressources internes, ces dernières ne doivent pas être impliquées dans la gestion des applications à tester. Par exemple, l'équipe qui a développé le logiciel ne doit pas se charger de la révision ou de l'évaluation finale ni vérifier la fiabilité du code.

Il existe plusieurs façons de réviser le code ou d'évaluer les applications pour parvenir au même niveau (ou à un niveau supérieur) de protection que celui proposé par un pare-feu d'applications (voir l'option 2 ci-après). Voici deux exemples pouvant satisfaire l'objectif de l'option 1 :

1. Une entreprise a mis en place, dans le cadre de son SDLC (cycle de vie du développement logiciel), un processus par lequel le code source des applications Web est soumis à une révision indépendante de sa sécurité. La révision de la sécurité doit consister en un contrôle des applications visant à traiter les vulnérabilités usuelles des applications Web. Ce type de révision du code peut être implémenté suivant un processus manuel ou automatique.
2. Mise en œuvre d'un processus manuel ou déploiement d'outils spécialisés pour rechercher les vulnérabilités ou les défauts lors de l'exécution d'une application Web. Cette approche implique la création et la soumission d'entrées malveillantes ou non standard pour simuler l'attaque d'une application. Les réponses à cette entrée sont examinées pour déterminer si l'application est vulnérable à certaines attaques.

Les révisions ou évaluations doivent être intégrées au SDLC et mises en œuvre avant le déploiement de l'application dans l'environnement de production. Le SDLC doit garantir la sécurité permanente des informations, conformément à la condition 6.3. Les processus de contrôle des modifications doivent garantir que les développeurs logiciels ne sont pas en mesure de contourner l'étape de révision du code et d'évaluation des applications, et de déployer ainsi directement le nouveau logiciel dans l'environnement de production. Les processus de contrôle des modifications doivent également mettre en œuvre les corrections et de nouveaux tests des vulnérabilités avant l'implémentation.

Si l'approbation/validation finale des résultats de la révision/analyse relève d'une organisation indépendante, il est recommandé que les révisions et analyses soient également exécutées le plus tôt possible au cours du processus de développement. Les outils doivent être accessibles aux développeurs logiciels et intégrés à leur suite de développement pour plus de simplicité.

Les fournisseurs peuvent intégrer les fonctions d'analyse du code et d'évaluation des applications aux produits conçus à d'autres fins, comme la validation de la conformité aux meilleures pratiques architecturales liées à un langage spécifique. Lors de l'évaluation de ces outils, il est crucial de vérifier leur capacité à tester la vulnérabilité des applications Web avant de partir du principe qu'ils peuvent être utilisés pour satisfaire la condition 6.6. De plus, pour faire face aux nouvelles menaces et aux menaces émergeantes, les outils doivent pouvoir intégrer de nouvelles règles d'analyse. Les individus réalisant les révisions et évaluations manuelles doivent se tenir informés des tendances de l'industrie pour disposer des compétences requises en matière de test et d'évaluation pour faire face aux nouvelles vulnérabilités.

Condition 6.6, option 2 – Pare-feu d'applications

Dans le cadre de la condition 6.6, un pare-feu d'applications consiste en un pare-feu d'applications Web (ou WAF), c'est-à-dire un point d'exécution de la politique de sécurité installé entre l'application et le client. Cette fonctionnalité peut être implémentée au niveau logiciel ou matériel et exécutée sur un périphérique ou un serveur type utilisant un serveur d'exploitation classique. Le périphérique peut être autonome ou intégré aux composants réseau.

Des pare-feu réseau types sont implémentés au périmètre du réseau ou entre les segments du réseau (zones) et assurent la première ligne de défense contre la plupart des types d'attaques. Ils peuvent toutefois permettre à des messages d'atteindre les applications Web qu'une organisation a choisi d'exposer sur l'Internet public. Les pare-feu réseau ne sont généralement pas conçus pour contrôler, évaluer et réagir aux éléments (paquets) d'un message de protocole Internet utilisés par les applications Web. C'est pourquoi les applications publiques reçoivent fréquemment des entrées non contrôlées. Il en résulte un nouveau périmètre de sécurité logique (l'application Web elle-même) et les meilleures pratiques de sécurité exigent que les messages soient contrôlés lorsqu'ils passent d'un environnement non sécurisé à un environnement sécurisé. Il existe de nombreuses attaques connues contre les applications Web et, nous en avons tous conscience, les applications Web ne sont pas toujours conçues pour se défendre contre ces attaques. À ce risque s'ajoute la disponibilité de ces applications à toute personne ou presque possédant une connexion Internet.

Les pare-feu d'applications Web sont conçus pour contrôler le contenu de la couche d'application d'un paquet IP, ainsi que le contenu de toute autre couche susceptible d'être utilisée pour attaquer une application Web. Il est à noter, toutefois, que la condition 6.6 n'a pas pour objet d'introduire des contrôles redondants. Le contenu du paquet IP correctement contrôlé (c'est-à-dire par un système de protection équivalent) par les pare-

feu réseau, les proxies ou tout autre composant ne doit pas être recontrôlé par un pare-feu d'applications Web.

La structure des paquets IP suit un modèle reposant sur des couches, chaque couche contenant des informations précises exécutées par des nœuds de réseau ou des composants (physiques ou logiciels) spécifiques prenant en charge le flux d'informations circulant sur Internet ou un intranet. La couche contenant le contenu traité par l'application s'appelle la couche d'application.

La technologie WAF est de plus en plus souvent intégrée aux solutions qui comprennent d'autres fonctions comme le filtrage des paquets, l'utilisation de proxies, la terminaison SSL, l'équilibrage de la charge, la mise en cache des objets, etc. Ces périphériques sont proposés sur le marché sous les appellations « pare-feu », « passerelles d'application », « système d'administration d'application », « proxy sécurisé », etc. Il est crucial de comprendre les fonctionnalités d'inspection des données d'un tel produit pour déterminer si ce dernier satisfait l'objectif de la condition 6.6.

Il est à noter que la conformité n'est pas garantie par la simple implémentation d'un produit dont les fonctionnalités sont décrites dans ce document. Un emplacement, une configuration, une gestion et une surveillance appropriés constituent également des aspects fondamentaux d'une solution conforme. L'implémentation d'un pare-feu d'applications Web représente l'une des options permettant de satisfaire la condition 6.6 et n'exclut pas la nécessité d'un processus de développement d'un logiciel sécurisé (condition 6.3).

Fonctionnalités recommandées

Un pare-feu d'applications Web doit intégrer les caractéristiques suivantes :

- Satisfaire toutes les exigences des normes PCI DSS en vigueur concernant les composants du système au sein d'un environnement de données de détenteurs de carte.
- Réagir comme il se doit (conformément aux règles ou à la politique en vigueur) aux menaces pesant sur les vulnérabilités appropriées, telles qu'elles ont été identifiées, au minimum, dans le OWASP Top Ten (document de sensibilisation aux normes industrielles pour la sécurité des applications Internet) et/ou dans la condition 6.5 PCI DSS.
- Contrôler les entrées/sorties des applications Web (autorisation, blocage et/ou alerte) en fonction de la politique et des règles actives et consigner les actions mises en œuvre.
- Prévenir les fuites de données : être en mesure de contrôler les entrées/sorties des applications Web (autorisation, blocage et/ou alerte) en fonction de la politique et des règles actives et consigner les actions mises en œuvre.
- Appliquer des modèles de sécurité positifs et négatifs. Le modèle positif (« liste blanche ») définit notamment le comportement autorisé, les entrées et les plages de données acceptables et refuse tout le reste. Le modèle négatif (« liste noire ») définit ce qui n'est PAS autorisé. Les messages correspondant à ces signatures sont bloqués tandis que le trafic ne correspondant pas aux signatures (les éléments ne figurant pas sur la liste noire) est autorisé.
- Contrôler le contenu des pages Web (Hypertext Markup Language (HTML), Dynamic HTML (DHTML) et Cascading Style Sheets (CSS)) et les protocoles sous-jacents permettant de délivrer ce contenu (Hypertext Transport Protocol (HTTP) et Hypertext Transport Protocol over SSL (HTTPS)). (Outre le protocole SSL, le protocole HTTPS comprend le protocole Hypertext Transport Protocol over TLS.)
- Contrôler les messages des services Web, si ces derniers sont exposés à l'Internet public. En règle générale, ce contrôle couvre les protocoles SOAP (Simple Object Access Protocol) et XML (eXtensible Markup Language), les modèles orientés document et RPC et le protocole HTTP.
- Contrôler les protocoles (propriétaires et normalisés) ou la construction des données (propriétaires ou normalisées) utilisés pour transmettre les données de ou à une application Web, lorsque ces protocoles ou données ne sont pas contrôlés en un autre point du flux de messages.

  Remarque : Les protocoles propriétaires constituent un défi pour les pare-feu d'applications actuels et des modifications personnalisées peuvent s'avérer nécessaires. Si les messages d'une application ne respectent pas les protocoles et constructions de données standard, il peut s'avérer judicieux de configurer le pare-feu pour qu'il contrôle ce flux de messages spécifique. Le cas échéant, l'implémentation de l'option de révision du code/évaluation des vulnérabilités de la condition 6.6 est probablement un meilleur choix.

- Se protéger contre les menaces ciblant le pare-feu d'applications Web lui-même.
- Prendre en charge une terminaison SSL et/ou TLS ou veiller à décrypter les transmissions cryptées avant le contrôle par le pare-feu d'applications Web. Les flux de données cryptées ne peuvent pas être contrôlés, sauf si le protocole SSL se termine en amont du moteur d'inspection.

Fonctionnalités recommandées supplémentaires pour certains environnements

- Se protéger contre et/ou détecter les fraudes aux jetons de session, par exemple, en cryptant les cookies de la session, les champs de formulaire masqués ou d'autres éléments de données utilisés pour la maintenance de l'état de la session.
- Recevoir et appliquer automatiquement les mises à jour de signatures dynamiques d'un fournisseur ou d'une autre source. Si cette fonctionnalité n'est pas disponible, il est nécessaire de mettre en place des procédures garantissant la mise à jour régulière des signatures WAF ou d'autres paramètres de configuration.
- Déploiement de type « fail open » (un périphérique en panne permet au trafic de circuler sans être contrôlé) ou « fail closed » (un périphérique en panne empêche tout trafic) suivant la politique active. Remarque : Il est nécessaire d'évaluer avec soin la possibilité d'autoriser un pare-feu d'applications Web à s'ouvrir en cas de panne du fait du risque d'exposer des applications Web sans protection sur l'Internet public. Il est possible, dans certaines circonstances, d'appliquer un mode de contournement dans lequel aucune modification n'est apportée au trafic. (Même en mode « fail open », certains pare-feu d'applications Web ajoutent des en-têtes de suivi, nettoient le code HTML considéré comme hors norme ou exécutent d'autres actions. Cela peut avoir une incidence négative sur les efforts de dépannage.)
- Dans certains environnements, le pare-feu d'applications Web prend en charge les certificats de clients SSL et l'authentification des clients utilisant des proxies via les certificats. De nombreuses applications Web modernes utilisent les certificats SSL clients pour identifier les utilisateurs finaux. Sans ce support, ces applications ne peuvent pas résider derrière un pare-feu d'applications. La plupart des pare-feu d'applications modernes intègrent le protocole LDAP (Lightweight Directory Access Protocol) ou d'autres répertoires utilisateurs et peuvent même procéder à une authentification initiale au nom de l'application sous-jacente.
- Certaines applications de e-commerce peuvent nécessiter un support de base de clés matérielles FIPS. Si cet élément doit être pris en compte dans votre environnement, il est nécessaire de vérifier que l'éditeur du pare-feu d'applications Web propose un système prenant en charge cette exigence, et de

garder à l'esprit que cette fonctionnalité risque d'augmenter considérablement le coût de la solution.

Considérations supplémentaires

Si les pare-feu d'applications Web peuvent garantir une protection contre la plupart des menaces concernant la sécurité, ils peuvent également entraîner des problèmes techniques au sein d'une infrastructure. Il est essentiel de vérifier les points suivants car ils peuvent entraver la réussite du déploiement :

- Les sites qui prennent en charge des en-têtes, des URL ou des cookies inhabituels peuvent nécessiter un réglage spécial. Les pare-feu d'applications Web imposent généralement une taille maximale à leurs composants. De plus, les signatures qu'ils recherchent peuvent filtrer des chaînes spécifiques qui, bien qu'elles soient considérées comme une exploitation abusive, sont en fait parfaitement valides pour une application donnée.
- Le contenu non conforme aux RFC HTML/HTTP ou tout simplement inhabituel peut également être bloqué sans réglage des filtres par défaut. Il peut tout autant s'agir du téléchargement de fichiers trop volumineux que de la soumission de contenu dans une langue ou un jeu de caractères étranger.
- Les technologies DHTML, AJAX (Asynchronous JavaScript and XML) et autres peuvent nécessiter une prise en charge, des tests et des réglages spéciaux. Ces applications ont parfois accès à un site Web d'une manière considérée comme malveillante par le pare-feu d'applications Web.
- Les applications qui requièrent des informations sur la session réseau sous-jacente, comme l'adresse IP, peuvent nécessiter des modifications si le pare-feu d'applications Web se comporte comme un proxy inverse. En règle générale, ces pare-feu d'applications Web placent les informations côté client dans un en-tête HTTP, ce comportement pouvant être inattendu pour les applications existantes.

Considérations importantes

- Les révisions de code et les évaluations des vulnérabilités des applications décrites dans ce document doivent être réalisées avant l'implémentation de l'application en production.
- Si un pare-feu d'applications Web de type « fail open » ou contournement est envisagé, des procédures spécifiques et des critères définissant l'utilisation des modes les plus risqués doivent être déterminés avant l'implémentation. Les applications Web ne sont pas protégées lorsque ces modes sont actifs et une longue période d'utilisation n'est pas recommandée.
- L'impact des modifications apportées aux pare-feu d'applications Web doit être évalué pour définir les éventuelles répercussions sur les applications Web appropriées et inversement.
- Communiquer le calendrier et la portée des modifications apportées aux pare-feu d'applications Web à toutes les parties concernées au sein de l'organisation.

- Respecter toutes les politiques et procédures, y compris le contrôle des modifications, la continuité de l'activité et la récupération des données en cas de sinistre.
- Les modifications apportées à l'environnement de production doivent être mises en œuvre pendant une phase de maintenance surveillée.

Sources d'informations supplémentaires

Voici une liste non exhaustive de sources d'informations supplémentaires sur la sécurité des applications Web.

- OWASP Top Ten (document de sensibilisation aux normes industrielles pour la sécurité des applications Internet)
- OWASP Countermeasures Reference (Références sur les contre-mesures OWASP)
- FAQ sur la sécurité des applications OWASP
- Build Security In (service de sécurité de Homeland, National Cyber Security Division)
- Scanners de vulnérabilité des applications Web (Institut national des normes et technologies)
- Critères d'évaluation des pare-feu d'applications Web (Consortium pour la sécurité des applications Web)

À propos de PCI Security Standards Council

La mission de PCI Security Standards Council est de renforcer la sécurité des données de paiement via la sensibilisation et la formation aux normes PCI Data Security Standard et autres standards qui augmentent le niveau de sécurité des données de paiement.

Le PCI Security Standards Council a été formé par les grands réseaux de cartes de paiement American Express, Discover Financial Services, JCB International, MasterCard Worldwide et Visa Inc. pour constituer un forum transparent dans lequel les parties intéressées peuvent prendre part au processus constant de développement, d'amélioration et de diffusion des normes de sécurité PCI DSS (PCI Data Security Standard), PED (PIN Entry Device) et PA-DSS (Payment Application Data Security Standard). Commerçants, banques, opérateurs et points de vente sont invités à nous rejoindre en tant qu'organisations participantes.