Cible de Sécurité rWeb4

Certification Sécurité de Premier Niveau

Version 1.3

26 Février 2013

Table des Matières

1. Identification

1.1 Identification de la cible de sécurité

La cible de sécurité CSPN du logiciel **rWeb4** a été rédigée par Deny All dans le cadre d'une démarche de certification auprès de l'ANSSI. Cette cible de sécurité a été élaborée en vue d'une évaluation Certification Sécurité de Premier Niveau (CSPN).

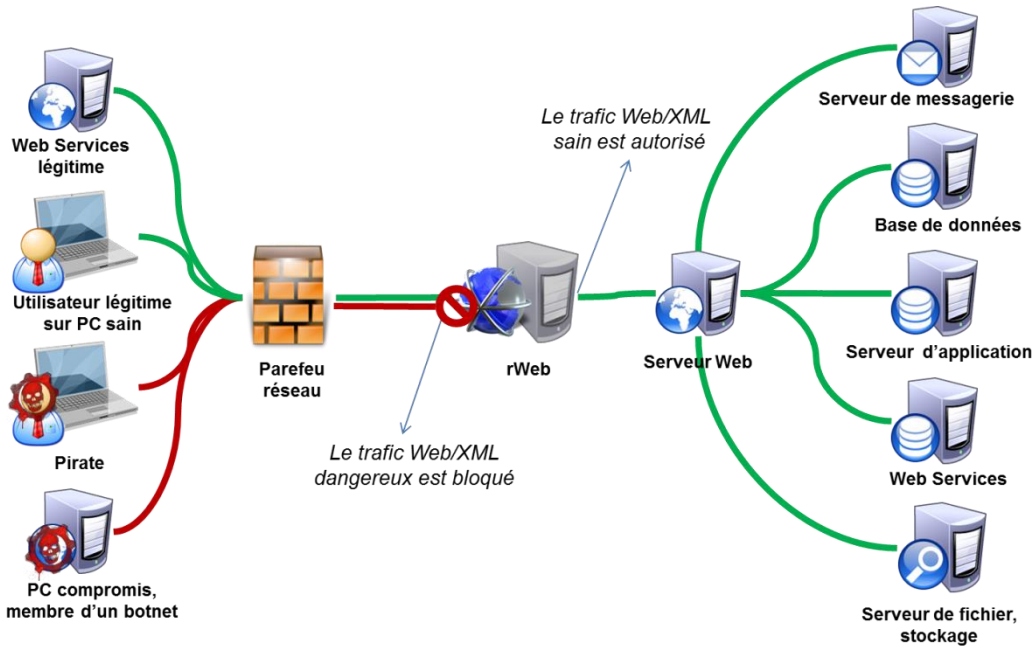
1.2 Identification du produit

Catégorie	Identification
Organisation éditrice	**Deny All**
Lien vers l'organisation	**http://www.denyall.com**
Nom commercial du produit	**rWeb**
Numéro de la version évaluée	**4.1 FP1**
Catégorie de produit	**Pare-feu applicatif Web**

2. Description du produit

2.1 Description générale du produit

rWeb est un serveur HTTP/HTTPS qui fournit des fonctionnalités de sécurité, d'authentification et d'accélération aux applications Web et Web Services. Complémentaire des contrôles de sécurité réseau traditionnels, rWeb est une solution intégrée qui protège les applications Web et XML et, par extension, toute l'infrastructure applicative d'une organisation :



Simple à installer, éprouvé depuis de nombreuses années sur les infrastructures les plus critiques, rWeb est capable de sécuriser et d'accélérer des dizaines d'applications avec un niveau de sécurité et de performance sans équivalent. rWeb est basé sur une technologie de reverse proxy optimisé et filtre la totalité des flux applicatifs HTTP/HTTPS, SOAP et XML.

rWeb est disponible en version logicielle, installable sous Linux, mais aussi sous la forme d'une image VMware ou encore en « bundle » avec une appliance standard HP. Les facteurs de forme appliance (virtuelle et physique) inclus un système d'exploitation Linux sécurisé, appelé DAOS.

2.2 Description des composants inclus dans le produit

rWeb est composé des cinq composants suivants :

- DAOS : ce système d'exploitation est basé sur Linux, ses performances ont été optimisées, sa sécurité renforcée. Il prend en charge toutes les fonctionnalités système. Il prend également en charge les mécanismes de haute disponibilité au travers de VRRP et la partie partage de charge entre les reverse-proxies.
- La GUI : fournit l'interface de management graphique ;

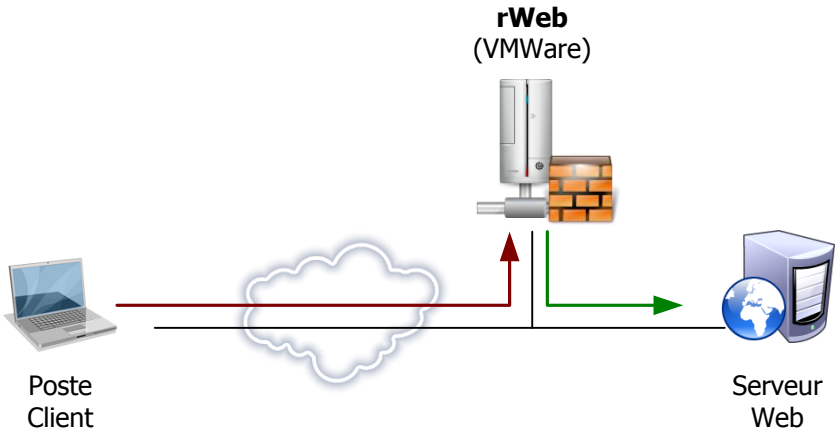
- L'API SOAP : est une interface standardisée pour tous les accès au "core manager" qui proviennent de la CLI, de la GUI, de la console de management, ou de tout autre logiciel personnalisé ;
- Le "Core Manager" : gère les différentes configurations et fournit la plupart des interfaces de communication ;
- La couche « Services » : prend en charge les requêtes clientes et réalise diverses opérations à travers ses modules pour fournir la sécurité, l'accélération et l'authentification. Les principaux composants de la couche service sont :
    - Apache : basé sur le reverse proxy Apache 2.2, mais a subi des modifications pour améliorer la sécurité et les performances (par exemple, protection contre l'attaque Slowloris) ;
    - Modules Apaches : toutes les fonctionnalités de sécurité sont fournies à travers des modules propriétaire. Des modules supplémentaires assurent les fonctions de reporting et de statistiques ;
    - HAProxy : c'est le composant qui assure le partage de charge pour les serveurs Web ;

2.3 Description de l'utilisation du produit

rWeb est un reverse-proxy (relai inverse), c'est-à-dire qu'il est la destination de l'ensemble des flux initiés par les postes clients. Les requêtes sont alors analysées et seules celles ne présentant pas de menace pour le serveur protégé sont transférées.

Ainsi les adresses des serveurs protégés ne sont plus publiées et rWeb devient le seul point d'accès à l'ensemble des applications Web.

Dans ce contexte, rWeb est utilisé pour la sécurisation de l'intégralité des flux HTTP et HTTPS à destination des serveurs et est par conséquent en coupure au niveau de la couche réseau. Il effectue une rupture protocolaire.



2.4 Description de l'environnement d'utilisation prévu

rWeb peut être installé dans les environnements suivants :

- Sur une appliance, livrée par DenyAll ;
- Dans un environnement VMWare ;
- Sur un système Linux CentOS ou RedHat.

Quel que soit l'environnement d'utilisation, le spectre fonctionnel reste identique.

Dans le cadre de la certification l'environnement d'utilisation sera l'environnement VMWare. rWeb sera donc installé dans une machine virtuelle. Cette machine virtuelle disposera d'une adresse IP, destination du trafic initié par les postes clients.

Une fois le trafic analysé, les requêtes saines sont transmises au serveur protégé. L'environnement d'utilisation est schématisé ci-dessous.

2.5 Description des hypothèses sur l'environnement

Environnement Logique:

- rWeb doit être installé sur un système sain, correctement mis à jour, en particulier au niveau des correctifs liés à la sécurité. Il convient également de sécuriser le système, par désactivation des services et partages inutiles, ou d'y installer DAOS, le système d'exploitation sécurisé fournit par Deny All.
- Le serveur rWeb est correctement configuré et administré.
- L'administrateur dispose des moyens de contrôler la configuration de rWeb par rapport à un état de référence, ou de la régénérer dans un état sûr.

Environnement physique :

- Les équipements contenant le serveur rWeb doivent se trouver dans des locaux sécurisés dont l'accès est contrôlé et restreint aux administrateurs.

Mesures organisationnelles :

- Les administrateurs sont des personnes considérées comme non hostiles. Ils sont formés pour exécuter les opérations dont ils ont la responsabilité et suivent les manuels et procédures d'administration.
- Les administrateurs sont sensibilisés à l'analyse régulière des événements d'audit générés par rWeb.
- Les procédures de gestion et traitement des alarmes sont formalisées, documentées et connues par les administrateurs de sécurité de rWeb.

2.6 Description des dépendances

C'est la version VMware de rWeb qui sera utilisée pour la certification. La seule dépendance est donc l'utilisation d'un hyperviseur (au choix du testeur) avec au minimum 4Go de RAM.

2.7 Description des utilisateurs typiques

L'utilisation de rWeb est transparente pour l'utilisateur final.

Le contexte d'utilisation de rWeb fait intervenir l'administrateur qui réalise les activités suivantes :

- L'installation du serveur ;
- La configuration des règles de sécurité ;
- La récupération et l'exploitation des journaux (alertes, actions) générés par rWeb.

2.8 Définition du périmètre de l'évaluation

L'évaluation porte sur la capacité du logiciel rWeb à protéger des applications Web et leurs données avec le jeu de règles standards sur l'environnement cible décrit dans la section 2.3. Les règles seront adaptées afin que l'application protégée puisse s'exécuter normalement.

Deny All ne fournit pas d'application de tests au CESTI.

La politique de sécurité évaluée est la politique « Maximum Security Policy » livrée avec rWeb 4.1. Cette politique applique l'ensemble des mécanismes de sécurité avancés avec une configuration générique, c'est-à-dire applicable sans configuration spécifique à l'application protégée.

3. L'environnement technique de fonctionnement

3.1 Matériel compatible ou dédié

Les matériels utilisables sont ceux adaptés pour les systèmes d'exploitation retenus.

3.2 Environnement système retenu

Dans le cadre de cette évaluation CSPN, l'environnement d'utilisation prévu est :

- rWeb 4.1 FP1
- DAOS10

4. Les biens sensibles que le produit doit protéger

Le produit contribue à protéger des biens utilisateurs de type informations et services de l'application Web, protégés par le filtrage des flux, susceptibles d'être accédés ou modifiés.

- Protection : Confidentialité, Intégrité et Disponibilité;

Nota : Les biens sensibles du logiciel rWeb (règles, journaux) doivent être protégés par le système d'exploitation et l'environnement d'exploitation sous lequel s'exécute l'application (hors périmètre).

- Protection : Confidentialité, Intégrité et disponibilité.

5. Description des menaces

Etant donné que les administrateurs ne sont pas considérés comme des attaquants potentiels, l'agent de menace est une entité qui transmet un flux vers le serveur Web :

- Un attaquant transmet des requêtes HTTP anormales au serveur Web. Cette attaque vise à modifier les informations gérées par l'application web.
- Un attaquant tente une attaque de type débordement de tampon ou de manipulation de paramètres. Cette attaque vise à modifier les informations gérées par l'application web. Celle-ci par effet de bord peut entrainer une indisponibilité du service offert par l'application web.
- Un programme malveillant consomme de la bande passante pour rendre le site inactif. Cette attaque vise à rendre indisponible le service offert par l'application web.
- Un programme malveillant recueille de manière systématique des informations sur le site Web (scan de vulnérabilités, fingerprinting, etc). L'objectif de cette démarche entreprise par l'attaquant est de récupérer des informations sur l'application web afin de pouvoir lancer ultérieurement une attaque ciblée.
- Un attaquant cherche à corrompre l'application Web par des attaques applicatives de type : injections SQL, Cross Site Scripting (XSS), injections de commandes, injections de code ColdFusion, PHP et ASP, injections d'E-mail, HTTP Response Splitting, modification du contenu XML etc.). Ces attaques visent à modifier les informations gérées par l'application web ou prendre le contrôle de celle-ci.
- Un attaquant tente d'établir des connexions avec des chevaux de Troie ou des portes dérobées. Cette possibilité est réalisable suite à une attaque, précédemment réussie, qui aurait permis d'injecter un cheval de Troie ou une porte dérobées au sein de l'application web. L'application étant corrompu, l'attaquant peut modifier le contenu de l'application web avec du contenu malveillant. L'objectif étant d'attaquer les utilisateurs par rebond depuis l'application web. L'attaquant peut en parallèle récupérer des informations auxquelles il ne devrait pas avoir accès.

6. Description des fonctions de sécurité du produit

La fonctionnalité de sécurité principale de rWeb est de protéger en disponibilité et en intégrité les applications et les données hébergées par un serveur Web contre les attaques. Cette protection est assurée par les règles configurées dans le logiciel.

Les fonctions de sécurité de rWeb4 sont donc :

- Mettre en œuvre les règles de sécurité telles que configurées dans l'application :
    - Détecter les événements selon le paramétrage du produit :
        - Analyse de la conformité protocolaire :
            - Détection des anomalies dans le protocole HTTP ;
            - Vérification du respect des contraintes de l'application (longueur des paramètres envoyés à l'application, etc.).
        - Détection d'attaques :
            - Détection d'outils de collecte d'information : scanneurs, robot d'indexation, robots divers… ;
            - Détection des tentatives de connexions des « chevaux de Troie » ou des « portes dérobées » déjà déployés au sein du système d'information ;
            - Détection générique des attaques applicatives connues :
                - Injections SQL diverses ;
                - Cross Site Scripting (XSS) ;
                - Injection de commandes ;
                - Injection de code ColdFusion, PHP et ASP ;
                - Injection d'E-mail ;
                - HTTP Response Splitting;
    - Réaliser des actions préventives (en vue de prévenir une attaque) :
        - Protection du contenu XML ;
        - Surveillance des accès aux sites Web ;
        - Réécriture des messages d'erreur renvoyés par le serveur Web ;
- Bloquer les attaques suite à leur détection,
- Journaliser les événements et les actions.

7. Améliorations et nouvelles fonctions du produit

Les améliorations apportées au produit depuis la dernière évaluation sont les suivantes :

- Canonisation Base64

  Identification des éléments encodés en base64 (longueur multiple de 4 caractères, padding, dictionnaire base64). Décodage puis passage aux moteurs de sécurité. Applicable à :

  - url
  - paramètres
  - headers

- Directory traversal

  Gestion des « \ » et d'un nombre de « . » supérieur à 2. Comptage des niveaux de navigation pour identifier une « remontée » au-dela de la racine

- HPP (HTTP Parameter Pollution)

  Concaténation des valeurs des paramètres ayant un nom identique avec un caractère de séparation (défaut : « , »). Passage aux moteurs de sécurité.

- HTTP Response Splitting

  Principe : blockage des headers dans les données postées et les query string. Liste de headers par défaut, peut être enrichie de headers « custom », des headers de la requête.

Les nouveaux modules de sécurité ajoutés au produit dans cette version sont les suivants :

- HTMLsec

  Blocage de certains tags, attributs et des event handlers HTML4 et HTML5. Repose sur une version modifiée de libxml2 pour gérer les malformations volontaires

- NestedSec

  Objectif : blocage des injections de code. Identification des délimiteurs de blocs () {} [], y compris les commentaires /* */ etc. Identification de chaines suspectes. Calcul de poids. Support de Java, JavaScript, SSI et PHP

- CalcSec

  Identification des opérations (1+5, -5, 3%2) etc. via analyse grammaticale générée par ANTLR

- SQLIsec

  Identification des requêtes SQL via analyse grammaticale générée par ANTLR

- Canonisation JSON

Les données au format JSON sont canonisées dans un format http standard (parametre=valeur[<&parametre=valeur>,…]). Elles sont ensuite envoyées aux moteurs de sécurité pour une analyse similaire à celle effectuée sur des données au format http.

- Injections de commandes

Utilise les grammaires de bash et cmd.exe. Si une injections (';', '|', ...) est détectée, que la grammaire correspond et que la commande shell est reconnue, on bloque. Possibilité de se baser sur les commandes présentes sur l'appliance (à la 'which') – mode « dynamique. »

Note : les nouveaux modules viennent en complément des modules existants. Par exemple SQLIsec tout seul sera très peu efficace, en revanche il permet de bloquer pas mal de requêtes SQL obfusquées…