UNIVERSITE KASDI MERBAH OUARGLA
Faculté des Nouvelles Technologies de l'Information et de la Communication
Département d'informatique et technologie de l'information

Mémoire

MASTER ACADEMIQUE
Domaine : Mathématiques et Informatique
Filière : Informatique
Spécialité : Informatique Fondamentale

Présenté par :
Babelhadj Yamina

Thème

Sécurisation des sites web
Cas Université Kasdi Merbah Ouargla

Soutenu publiquement
Le : 29/06/2017

Devant le jury

M. HERROUZ Abdelhakim	Maitre A	UKM Ouargla	Président
M. MAHDJOUBE Mohamed Bachir	Maitre A	UKM Ouargla	Rapporteur
M. BENKHROUROU Chafika	Maitre A	UKM Ouargla	Examinateur

Année universitaire : 2016 /2017

Dédicaces

Je dédie ce modeste travail à celle qui m'a donné la vie, le symbole de tendresse, qui s'est sacrifiée pour mon bonheur et ma réussite, à ma **mère** …

A toute mes frères et mes sœurs.

A toute mes amis.

A tous ceux qui sont proches de mon cœur.

Je dédie ce modeste travail.

Yamina

REMERCIEMENTS

Tout d'abord, Je voudrais remercier ALLAH, tout puissant de m'avoir donné la force et le courage pour terminer ce travail.

Nous exprimons nos profonds remerciements à notre encadreur, M. Mahdjoub Mohammed Bachir qui a dirigé ce travail, pour son soutien, pour son aide et ses conseils précieux et critiques et pour la liberté de recherche qu'il a bien voulu nous laisser.

Je remercie également toutes les personnes qui nous ont aidés de près ou de loin pour la réalisation de ce travail.

Je remercie aussi les membres du jury pour l'honneur qu'ils ont fait en acceptant de juger notre travail.

Sommaire

Liste des Figures

Acronymes et abréviations

API: Application Programming Interface

ASA : Adaptive Security Appliance

ACL : Access Contrôle Liste

Bootstrap: collection d'outils utile à la création du design

CIGREF : Club Informatique des Grandes Entreprises Françaises

CNIL : Commission nationale de l'informatique et des libertés

CSRF: Cross-Site Request Forgery

CSS : Cascading Style Sheets

CMS : Content Management System

CERISTE: Centre de Recherche sur l'Information Scientifique et Technique.

DMZ: Zone démilitarisée

DDOS: Distributed Denial of Service

INFOSEC: Information Security

ISO: International Organization for Standardization

IDS: Intrusion Detection System

Joomla: CMS open source gratuit

LAMP: Linux, Apache, MySQL, PHP

LDAP: Lightweight Directory Access Protocol

OWASP: Open Web Application Security Project

PuTTY : Client SSH et Telnet gratuit pour Windows

SSH : Secure Shell

SMSI: Système de gestion de la sécurité de l'information

UKMO : UniversitéKasdiMerbahOuargla

URL: Uniform Resource Locator

WAF: Web Applications Firewall

WAS: Web Application Security Consortium

WSTC: Washington State Tax Consultants

XSS: Cross-site scripting

Résumé

La fiabilité d'un site web est un enjeu colossal de la compétitivité sur internet. En effet un organisme dans le domaine éducative tel que l'université Kasdi Merbah Ouargla doit assurer une protection efficace de ses sites web permettant aux différents internautes de naviguer dans le site d'une manière confidentielle.

Cette confiance ne peut être gagnée qu'au prix d'une politique de sécurité stricte avec un contrôle permanent des risques d'attaques de piratage. Ces sites sont la cible des attaques diverses : injection de code, défacement, détournement de session, etc.

Le but de ce travail est de faire l'audit d'un site web pour déterminer les différentes failles de sécurité et les vulnérabilités et on donnant des recommandations pour garantir :

- l'intégrité des données
- la confidentialité des données
- la disponibilité des données

Mots clés :

Intégrité, confidentialité, disponibilité, sécurité, audit, vulnérabilités, site web, piratage.

ملخص

يعتبر تحقيق مصداقية موقع ويب تحد كبير للمنافسة على شبكة الانترنت.

وعليه كان لزاما على المؤسسات التربوية أن تعمل على ضمان الحماية الفعالة ومصداقية مواقع الويب مما يسمح للمستخدمين بتصفح الموقع في محيط آمن.

هذه المصداقية لا يمكن كسبها الا باتباع استراتيجية أمنية صارمة لمواجهة مختلف التهديدات وهجمات القرصنة والسيطرة عليها.

هذه المواقع تعد عرضة لهجمات مختلفة: حقن SQL، تشويه البيانات المخزنة، سرقة الجلسة ...الخ

والهدف من هذا العمل هو القيام بدراسة معمقة وفحص دقيق للموقع لتحديد الثغرات الأمنية المختلفة وكذانقاط الضعف وتقديم توصيات لضمان ما يلي:

- سلامة البيانات
- خصوصية بيانات
- توافر البيانات

الكلمات المفتاحية:

سلامة البيانات، خصوصية بيانات، توافر البيانات، الأمن، المراجعة، الثغرات الأمنية، موقع، القرصنة.

Abstract

The reliability of a website is a huge challenge of competitiveness on the internet. Indeed an organization in the educational field such as the Kasdi Merbah Ouargla University must ensure effective protection of its websites allowing the different Internet users to navigate the site in a confidential way.

This confidence can only be gained through a strict security policy with permanent control over the risks of piracy. These sites are the target of various attacks: code injection, defacement, hijacking…

The purpose of this work is to audit a web site to determine the various security failures and vulnerabilities and give recommendations to ensure:

- data integrity
- confidentiality of data
- availability of data

Key words:

Integrity, confidentiality, availability, security, audit, vulnerabilities, website, hacking.

Introduction générale

Introduction Générale

L'internet est devenu aujourd'hui un moyen de communication incontournable de la société moderne. La démocratisation de l'usage de l'internet a entrainé une prolifération des sites web parmi lesquels des sites web professionnels, propriétés des entreprises, les administrations publiques.

Un site web constitue généralement une véritable plateforme d'accès à l'information.

Les entreprises ont été amenées à concevoir et à mettre en œuvre des sites web, les attributions de ces derniers étant :

- un rôle de vitrine informative, présentation de produits et services ;
- une fonction de communication et d'information, mise à disposition d'informations utiles ;
- une plate-forme d'échange commerciale, commerce en ligne et vente par correspondance.
- conception d'un site web multimédia interactif, destiné à la formation pédagogique des étudiants dans le cadre de leur cursus universitaire.

Cependant, les vulnérabilités de ces applications Web ou de sites web sont désormais le vecteur le plus important des attaques dirigées contre la sécurité des systèmes d'information de ces administrations et de ces entreprises. En effet, D'après les différents rapports publiés par les observatoires et sociétés de sécurité informatiques, les attaques web sont en constante augmentation. Les conséquences peuvent être très lourdes pour les administrations victimes de cette situation :

- Atteinte à l'image de l'administration,
- Une défiguration du site pour relayer un message politique, pour dénigrer ou pour revendiquer son attaque,
- Mise en danger de l'intégrité du système d'information,
- Récupérer, des données et d'information sensibles.

A cet effet, nous ne pouvons plus nous permettre de tolérer les problèmes les plus simples comme ceux présentés dans le Top 10 OWASP qui sont dus principalement à un développement et un déploiement non sécurisé. Ainsi, la mise en place de méthodes et d'outils pour gérer le développement et le contrôle qualité des applications s'avère plus que nécessaire pour réduire leur vulnérabilité.

Notre mémoire présente nos travaux selon le plan suivant. Dans un premier chapitre, nous rappelons les grands concepts liés aux domaines de la sûreté de fonctionnement et de la sécurité informatique liés au site web afin de préciser la terminologie et les concepts utilisés dans ce mémoire. Ce chapitre présente également un panorama des travaux existants dans le

domaine de l'évaluation de la sécurité afin de placer notre approche dans le paysage des approches quantitatives pour la mesure de la sécurité.

Le second chapitre de notre mémoire présente plus précisément le cadre de nos travaux et la démarche que nous adoptons pour la production des mesures. Nous y spécifions les concepts spécifiques que nous avons utilisés et présentons les facteurs environnementaux que nous avons identifiés :

1) le cycle de vie de la vulnérabilité ;

2) le comportement de la population des attaquants ;

3) le comportement de l'administrateur ainsi que leur évolution comportementale dans le temps. Nous en déduisons les scénarios d'exploitation de vulnérabilité qu'il nous est indispensable de distinguer afin d'élaborer le processus de mesure.

Dans le troisième et dernier chapitre, nous présentons des meilleures pratiques permettant d'éviter les failles les plus connus dans le développement des sites web.

La seconde partie de notre chapitre porte sur les bonnes pratiques à respecter lors du déploiement et la mise en production d'un site dans les cas de site de l'université Kasdi Merbah Ouargla. Aussi, elle explique le processus de détection des incidents.

Enfin, dans une conclusion, nous présentons un bilan de nos travaux. Nous essayons d'avoir un recul critique sur notre approche et détaillons les améliorations possibles ainsi que les perspectives que nous envisageons à court et moyen terme.

Chapitre 1

Les Failles de sécurité des sites web

Chapitre 1 : Les Failles de sécurité des sites web

1.1. Introduction

La sécurité informatique c'est l'ensemble des moyens mis en œuvre pour réduire la vulnérabilité d'un système contre les menaces accidentelles ou intentionnelles. Il convient d'identifier les exigences fondamentales en sécurité informatique.

Les sites web sont par nature des éléments très exposés du système d'information. Leur sécurisation revêt une grande importance.

Ce chapitre propose un état de l'art concernant les vulnérabilités dont souffrent aujourd'hui les sites web ainsi que les mécanismes permettant d'assurer la protection face aux attaques ciblant ces vulnérabilités.

Nous présentons d'abord le domaine de la sûreté de fonctionnement de sécurité informatique des sites web, qui réunit trois des principaux attributs de la sûreté de fonctionnement : confidentialité, intégrité et disponibilité .Ensuite, nous décrions les principales vulnérabilités des applications Web et les attaques correspondantes, en focalisant sur les plus répandues. Enfin, nous présentons Un aperçu global des mécanismes permettant d'assurer la protection des sites web publiées à ce jour.

1.2. Le site web, un composant des systèmes d'information

Le début des années quatre-vingt-dix, outre l'accélération de la montée en puissance de l'informatique personnel, a vu une interconnexion grandissante entre le monde de la micro-informatique et celui des grands systèmes. L'implémentation des architectures client-serveur et la naissance du World Wide Web (WWW) y ont grandement contribué.

Le concept client-serveur correspond à une philosophie d'approche de l'informatique en rupture forte par rapport aux stades précédents. Le client est vu comme l'utilisateur qui derrière son poste de travail, généralement un PC, demande des services à plusieurs serveurs distants, qui peuvent être d'anciens mainframes ou non. Ceux-ci, généralement caractérisés par des gammes de puissances, plus importantes que le poste de travail de l'utilisateur, rendent le service demandé. Il se peut que dans ce modèle, pour répondre aux besoins du client, qu'ils aient eux-mêmes à demander certaines informations à d'autres systèmes. Nous voyons alors qu'un serveur d'un client donné peut, afin de rendre un service demandé, se

retrouve lui-même temporairement à l'état de client. Cet état de choses à fait naitre un grand réseau, le réseau des réseaux, l'internet.

Internet a été par la suite utilisé pour développer un service : le web. De nos jours, il est devenu presque anodin de se connecter à internet et de bénéficier des services du web grâce aux sites web.

1.2.1. Définitions historique et rôles des sites web

Un site ou site web de l'anglais website, qui se traduit littéralement en français par site de la toile, est un ensemble de pages web hyper liées entre elles et accessible à une adresse web. Un site web est composé d'un ensemble de documents structurés, nommés pages web, stockés (hébergés) sur un ordinateur (serveur) connecté au réseau mondial (internet).[1]

Une page web est l'unité de consultation du Word Wide Web. C'est un document informatique qui peut contenir du texte, des images, des formulaires à remplir et divers autres éléments multimédias et interactifs.

Il conviendra de distinguer un site web d'une application web. En effet, Une application web2 est une application manipulable grâce à un navigateur web. De la même manière qu'un site web, une application web est généralement placée sur un serveur et se manipule en actionnant des onglets directement à partir du navigateur web via un réseau informatique.

Ainsi, à la différence des sites web standards, une application web est tout site web qui permet à ses utilisateurs d'accomplir des tâches spécifiques (gérer des mails, créé du contenu etc).

1.2.2. Rôles d'un site web

L'évolution actuelle du monde a rendu l'utilisation quotidienne de l'internet aussi indispensable que la télévision, le téléphone etc. Lorsqu'un internaute se connecte à un site web, c'est dans un but précis : trouver des réponses à ses questions, communiquer, s'amuser.

L'ère de la numérisation implique l'accès à l'information en direct, répondant au besoin du tout et tout de suite. Une page web contient donc à cet effet des informations, généralement pour

[1)] Web 2.0 origine et évolution

informer ou faire connaître. Le rôle d'un site web devra donc être abordé sous l'angle de celui qui l'utilise (usage particulier, usage d'entreprise, usage pour une organisation etc.)

1.2.3. La typologie des sites web

Il existe deux grandes catégories de site web à savoir : les sites web statiques et les sites web dynamiques.

1. **Les sites web statiques :**

Un site web statique est un site web dont les pages sont statiques. On entend par page statiques, non pas une page sans mouvement ou sans animations mais une page visible telle qu'elle a été conçue.

Le contenu des sites web statiques ne peut pas être mis à jour automatiquement. La mise à jour nécessite l'intervention du webmaster. Ce dernier doit modifier le code source pour y ajouter des nouveautés. Ainsi les sites web statiques sont caractérisés par une très faible fréquence de mise à jour. Ils sont adaptés pour construire des sites web « vitrine ».

2. **Les sites web dynamiques**

Un site web dynamique est un site dont le contenu peut être généré dynamiquement, c'est à dire que ce contenu peut s'afficher en fonction de l'utilisateur qui le consulte ou d'autres paramètres. Les sites web dynamiques incluent l'utilisation de base de données, ce qui offre plus de possibilités de développement. Ainsi le contenu du site web dynamique peut changer sans l'intervention du webmaster. Les sites web dynamiques sont donc caractérisés par un niveau de mise à jour régulier et fréquent.

1.2.4. L'importance d'un Site web dans le système d'information de l'entreprise

La mise en place d'un site web dans une entreprise revêt une importance stratégique pour l'entreprise. Il y va du choix du type de site web, du rôle à lui attribuer et des objectifs stratégiques à atteindre. La prise en compte du site web dans le dispositif des systèmes d'information de l'entreprise permet de se rendre compte que les sites web constituent un support majeur de l'information et de sa divulgation. Ainsi nous pouvons affirmer que le site web est « une partie visible de l'iceberg » qu'est le système d'information. Il convient donc

de s'assurer que sa conception garantit la sécurité de l'information qu'il contient et que ces dernières soient fiables.

Le site web constitue l'épine dorsale du système d'information de l'entreprise, vitale au bon fonctionnement de cette dernière.

1.2.5. Langages de programmation et fonctionnement d'un site web

Le langage est la base de toute communication. Pour communiquer et se comprendre il faut pouvoir parler le même langage.

1. **Les langages de programmation**

Dans le domaine de l'informatique, l'homme à du trouver un langage afin de communiquer avec la machine par l'intermédiaire de codes. Il s'agit du langage de programmation. Un langage de programmation est un vocabulaire et un ensemble de règles d'écriture utilisées pour instruire un ordinateur d'effectuer certaines tâches.

De nos jours il existe plusieurs langages de programmation web. Les plus utilisés sont le XHTML, le CSS, le PHP/MySQL et JavaScript.

2. **Fonctionnement des sites web**

Pour la plupart profanes de l'informatique, le site web a besoin d'être décrypté. Pour se faire on utilise des navigateurs web.

Les navigateurs web : le navigateur est devenu probablement le programme le plus utilisé sur un ordinateur. Et pour cause c'est grâce au navigateur que de nombreux internautes ont accès aux sites internet. Le rôle du navigateur est d'analyser le code XHTML et CSS des pages web et d'en produire un résultat visuel, facile à lire pour un humain.

Les URL : (Uniform Resource Locator) sont une invention du World Wide Web et sont utilisées pour identifier les pages et les sites web. Elles sont aussi appelées adresses web. Les URL ont été inventées pour pouvoir indiquer avec une notation (d'où l'adjectif «uniforme») aux navigateurs web comment accéder à toutes les ressources d'internet. Chaque navigateur web dispose d'une «barre d'adresse» affichant l'URL de la ressource consultée.

Le protocole http : L'HyperText Transfer Protocol, plus connu sous l'abréviation HTTP, littéralement « protocole de transfert hypertexte », est un protocole de communication client-serveur développé pour le World Wide Web. Les clients HTTP les plus connus sont les navigateurs Web permettant à un utilisateur d'accéder à un serveur contenant les données.

Figure 1.1 : Fonctionnement des sites web

1.3. La sécurité des sites web en tant que système d'information

L'importance que revêt le site web dans le système d'information de l'entreprise mérite qu'on accorde une importance toute aussi particulière à sa sécurité.

La sécurité peut être définie comme l'ensemble des moyens mis en œuvre et dont le rôle est d'assurer une protection contre tout danger clairement défini. La sécurité d'un système d'information fait souvent l'objet de métaphore car on la compare souvent à une chaine en expliquant que le niveau de sécurité d'un système est caractérisé par le niveau de sécurité du maillon le plus faible. Ainsi une porte blindée est inutile dans un bâtiment si les fenêtres sont ouvertes sur la rue.

1.3.1. Définitions

La sécurité des systèmes d'information (SSI) est l'ensemble des moyens techniques, organisationnels, juridiques et humains nécessaires et mis en place pour conserver, rétablir, et garantir la sécurité du système d'information. Assurer la sécurité du système d'information est une activité du management des systèmes d'information.[2]

La sécurité des systèmes d'information doit être abordée de façon à assurer la confiance des utilisateurs du système d'information. C'est la raison pour laquelle il est nécessaire d'élaborer une politique de sécurité, c'est à dire :

[2)] Mise en place d'un site sécurisé, Mémoire Online

- élaborer des règles et des procédures, installer des outils techniques dans les différents services de l'organisation (autour de l'informatique) ;

- définir les actions à entreprendre et les personnes à contacter en cas de détection d'une intrusion ;

- sensibiliser les utilisateurs aux problèmes liés à la sécurité des systèmes d'information ;

- préciser les rôles et responsabilités.

De ce fait donc la définition que nous retiendrons de la sécurité des systèmes d'information est que la sécurité des systèmes d'information est un ensemble de politiques et procédures qui permettent d'éviter les intrusions (confidentialité), les incohérences (intégrité) et les pannes (disponibilité) des systèmes d'information, et qui définissent les règles d'authentification.

1.3.2. Les attributs de la sécurité

La sécurité doit être considérée d'abord au niveau de la sécurité générale et du maintien d'une information fiable et cohérente.

Le rôle commun à tous les sites web quel que soit leur typologie est la gestion de l'information : grâce aux sites web, les entreprises collectent des informations (coordonnées, préférences des visiteurs etc.), divulgue de l'information (publicité sur leurs produits, présentation d'équipes, situation géographique etc.). La sécurité des sites web fait alors intervenir la sécurité de l'information.

La sécurité de l'information vise à protéger l'information contre une large gamme de menaces, de manière à garantir la continuité des transactions, à réduire le plus possible le risque et à optimiser le retour sur investissement ainsi que les opportunités en termes d'activité pour l'organisme.

Par conséquent on appelle sécurité de l'information, tous les moyens techniques, organisationnels, juridiques, et humains mis en place pour faire face aux risques identifiés, afin d'assurer la confidentialité, l'intégrité, la disponibilité (variables du triangle infosec) et la traçabilité de l'information traitée.

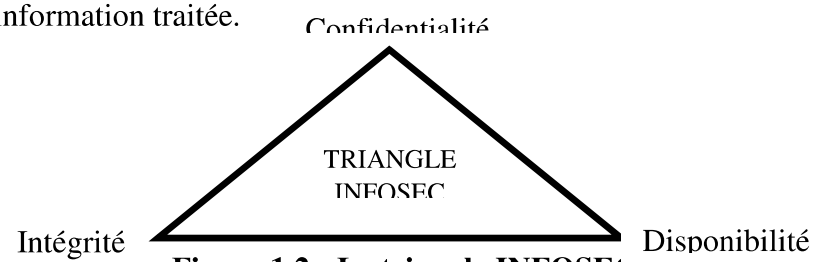

Figure 1.2 : Le triangle INFOSEC

La sécurité est l'association de trois des attributs principaux qui sont la confidentialité, l'intégrité et la disponibilité.

1. **La confidentialité** : définit l'absence de divulgation non autorisée de l'information.

Une attaque contre la confidentialité par une personne malveillante consiste à tenter de récupérer des informations pour lesquelles elle ne possède pas d'autorisation, soit en tentant d'y accéder sur le système, soit en écoutant les communications réseaux, soit de toute autre façon possible.

L'information ne doit pas être divulguée à toutes personnes, entité ou processus non autorisé. En clair, ça signifie que l'information n'est consultable que par ceux qui ont le droit d'y accéder.

2. **L'intégrité** : définit l'absence d'altération inappropriée de l'information.

Une attaque contre l'intégrité vise à introduire de fausses informations, ou à modifier ou détruire l'information existante.

Le caractère complet et correct des actifs doit être préservé. En clair cela signifie que l'information ne peut être modifiée que par ceux qui en ont le droit.

3. **La disponibilité :** définit le fait que le système soit prêt à délivrer son service.

Une attaque contre la disponibilité peut avoir deux origines. La première consiste à déjouer les politiques de sécurité et à exploiter une faute pour qu'elle produise une erreur affectant la délivrance du service. La seconde méthode consiste à engorger le système de demandes de service valides afin d'occuper le système et rendre sa disponibilité faible ou inexistante pour l'utilisateur légitime.

L'information doit être rendue accessible et utilisable sur demande par une entité autorisée. Cela veut dire que l'information doit être disponible dans des conditions convenues à l'avance.

En plus de ces trois attributs, la sécurité compte des attributs dits secondaires, quenous détaillons ici :

• Le **non répudiation** : regroupe la disponibilité etl'intégrité de l'identité de l'émetteur d'un message (non réfutation de l'origine) ou dudestinataire d'un message (non réfutation de la destination).

• L'**authenticité** : regroupe l'intégrité du contenu et de l'origined'un message, et éventuellement d'autres informations, comme l'instant d'émission.

• La **responsabilité** : regroupe la disponibilité et l'intégrité del'identité de la personne qui a effectué une opération.

• La **traçabilité** : garanti que les accès et tentatives d'accès aux éléments considérés sont tracées et que ses traces sont conservées et exploitables.

1.3.3. Terminologie dans le domaine de la sécurité web

Le domaine de la sécurité possède également ses propres termes, Ce sont ces termes que nous évoquons dans ce paragraphe.

Le premier terme spécifique à la sécurité est la **vulnérabilité**. La première définition rattache la vulnérabilité au concept de faute : une vulnérabilité est une faute interne qui permet à une faute externe d'endommager le système. Cette définition peut être complétée par la deuxième : une vulnérabilité est une faute accidentelle, ou intentionnelle, malveillante ou non, dans les spécifications, la conception ou la configuration du système, ou dans la manière dont il est utilisé, qui peut être exploitée pour créer une intrusion. Cette définition rejoint la précédente et définit la vulnérabilité comme une entrave à la sécurité. On retrouve dans la norme ISO : 17799 devenue ISO : 27002 – la définition suivante : une vulnérabilité est une faiblesse d'un bien (quelque chose ayant de la valeur pour l'organisation, ses opérations et leur continuité) ou groupe de biens qui peut être exploitée par un attaquant.
Enfin, l'entreprise Microsoft possède une définition plus énumérative en mettant en avant les différentes possibilités ainsi offertes à un attaquant : une vulnérabilité est une faille dans un produit qui offre la possibilité à un attaquant d'usurper des privilèges d'un utilisateur, effectuer des opérations, compromettre des données, accéder à des données confidentielles. Malgré cela, cette liste non exhaustive de définitions semble cohérente et définit le même concept global.

Dans certaines des définitions que nous avons citées apparaît la notion d'attaque ou celle d'attaquant.

Une **attaque** est définie comme faute d'interaction malveillante visant à violer une ou plusieurs propriétés de sécurité. C'est une faute externe créée avec l'intention de nuire, y compris les attaques lancées par des outils automatiques : vers, virus, zombies, etc. La notion d'attaque ne doit pas être confondue avec la notion d'intrusion.

Une **intrusion** est donc définie comme une faute malveillante interne d'origine externe, résultant d'une attaque qui a réussi à exploiter une vulnérabilité. Elle est susceptible de produire des erreurs pouvant provoquer une défaillance vis-à-vis de la sécurité, c'est-à-dire une violation de la politique de sécurité du système.

La multiplication des vulnérabilités et des attaques sur des sites web sur Internet ont poussé de nombreuses organisations à poser un regard critique sur la qualité de la sécurité de leurs applications web.
Ainsi, plusieurs communautés ont vu le jour, dans le but d'améliorer la sécurisation des applications web. Les travaux dans ce contexte se sont traduits aussi par la proposition de taxonomies et de classifications pour les vulnérabilités et les attaques web les plus répandues. Parmi ces communautés, nous citons **OWASP** (**O**pen **W**eb **A**pplication **S**ecurity **P**roject) et **WASC** (**W**eb **A**pplication **S**ecurity **C**onsortium).

Les membres du "Web Application Security Consortium" ont créé ce projet pour développer et promouvoir une terminologie standard décrivant les problèmes de sécurité des applications Web et permettant aux développeurs d'applications, experts en sécurité, développeurs de logiciels et les consultants en sécurité, d'utiliser un langage commun pour interagir entre eux. Une première version pour la classification des vulnérabilités composée de six classes a été proposée dans le document "Web Application Security Consortium :Threat Classification" :

- **Authentification** : il s'agit de vulnérabilités qui concernent les fonctions du site Web permettant d'identifier un utilisateur, un service ou une application.
- **Autorisation** : cette classe regroupe les vulnérabilités liées aux fonctions destinées à vérifier les droits attachés à un utilisateur, un service ou une application.
- **Attaques côté client** (Client-sideAttacks) : il s'agit de vulnérabilités permettant aux attaquants de cibler directement les utilisateurs du site Web en leur délivrant par exemple des contenus illicites tout en faisant croire qu'il s'agit d'informations provenant du site original.
- **Exécution de commandes** (Command Execution) : cette classe regroupe les vulnérabilités permettant l'exécution à distance de commandes sur le site Web.
- **Divulgation d'information sensible** (Information Disclosure) : cette classe inclut les vulnérabilités dont l'exploitation permet l'obtention d'informations sur le système (système d'exploitation, version, etc...).

- **Erreurs logiques et bug logiciel** (LogicalAttacks) : les vulnérabilités appartenant à cette classe peuvent conduire à des attaques permettant de détourner la logique d'implémentation de l'application pour réaliser des actions illicites.

Cette classification mélange parfois les faiblesses et les attaques permettant de les exploiter. Une nouvelle version a été développée par la suite pour pallier à ce problème en distinguant ces deux dimensions et en fournissant une liste plus riche des menaces (WASC WSTC v2) .
D'un autre côté, l'Open Web Application Security Project (OWASP) a défini dans l'un de ses projets nommé "TOP 10" les dix classes de vulnérabilités Web les plus critiques. L'objectif principal du Top 10 de l'OWASP est d'informer les développeurs, concepteurs, architectes, managers, et les entreprises au sujet des conséquences des faiblesses les plus importantes inhérentes à la sécurité des applications Web. Le Top 10 fournit des techniques de base pour se protéger contre ces vulnérabilités.

Nous les détaillons dans les sous-sections suivantes.

1.3.4. Notions de faute, erreur et défaillance

Ces trois notions sont appelées les entraves à la sûreté de fonctionnement. Une défaillance du service survient lorsque le service délivré par le système dévie du service correct. Rares sont les systèmes qui ne défaillent pas, d'où cette définition alternative de la sûreté de fonctionnement : la sûreté de fonctionnement est l'aptitude à éviter des défaillances du service plus fréquentes ou plus graves qu'acceptables.

La partie de l'état du système pouvant entraîner une défaillance est une erreur. La cause adjugée ou supposée d'une erreur est une faute. Dans ce paragraphe, nous présentons ces trois notions et les liens entre elles :
Les fautes, regroupées en trois familles : les fautes de développement, les fautes physiques et les fautes d'interaction.

Une faute est active lorsqu'elle produit une erreur. Une faute active est soit une faute interne qui était préalablement dormante et qui a été activée par le processus de traitement, soit une faute externe. Une faute interne peut alterner entre les états dormant et actif.
Une erreur peut être latente tant qu'elle n'a pas été reconnue en tant qu'erreur, ou détectée par un algorithme ou mécanisme de détection car elle produit une défaillance. Une erreur latente peut être corrigée avant d'être détectée.

Une défaillance survient lorsqu'une erreur affecte le service délivré par le système, donc lorsqu'elle traverse l'interface système-utilisateur. La conséquence de la défaillance d'un composant est une faute interne pour le système et une faute externe pour les composants avec lesquels il interagit, comme résumé sur la figure.

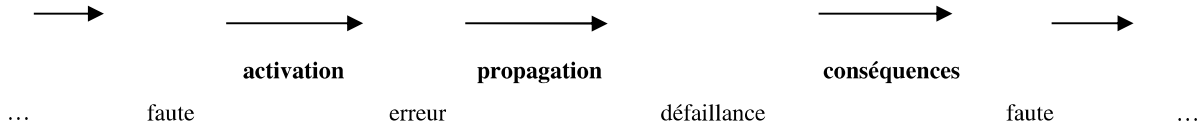

Figure 1.3 : Les entraves de la sûreté de fonctionnement

1.3.5. Contre mesures et moyens de protection

Différentes méthodes et techniques peuvent être mises en oeuvre par les développeurs et les administrateurs en charge de la sécurité informatique pour faire face aux diverses menaces qui visent les applications Web. Dans la suite nous présentons quelques exemples de moyens permettant d'assurer la sécurité, en considérant les objectifs suivants :

- **Prévention de vulnérabilités** : empêcher l'introduction de vulnérabilités par l'application de méthodes de développement rigoureuses.
- **Élimination des** vulnérabilités : identifier les vulnérabilités et les éliminer en utilisant des techniques de vérification et de test.
- **Prévention, détection et tolérance des intrusions** : protéger le système pendant l'exploitation vis-à-vis des attaques et des intrusions en mettant en œuvre des barrières de défense (pare-feu, systèmes de détection, prévention ou de tolérance aux intrusions) permettant à l'application de fournir un service correct en dépit des attaques.
- **Évaluation** : estimer par évaluation l'impact des vulnérabilités et des attaques ainsi que l'efficacité des mécanismes de protection mis en œuvre.

1.3.6. Risques, vulnérabilités et mesures sécuritaires liées aux sites web

Comme les systèmes informatiques, les sites web sont exposés à des risques et vulnérabilités.

1. Les risques liés aux sites web

Etymologiquement, le mot risque vient de l'italien risco, mot dérivé du latin resecum, (« ce qui coupe »), désignant le rocher qui menace les navires marchands, autrement dit le danger en mer. Le web est de nos jours ce qu'était la mer au 18ème siècle, c'est à dire un moyen

d'échange en tous genres entre les continents et de décloisonnement des frontières. A cet effet le risque est alors omniprésent sur le web.

La notion de risque fait alors intervenir deux éléments essentiels : l'évènement dont l'occurrence est probabilisable, qui va entrainer des conséquences négatives : un dommage, une perte.

les risques inhérents non exhaustifs aux données et les systèmes informatiques sont : incendie, inondation, explosion, panne des installations, malveillance, erreur humaine, pirate, défaillance fournisseur, environnemental, cyber-attaque, virus, rupture d'approvisionnement, externalisation défaillante, panne d'énergie, etc.. »

L'usage de site web par les entreprises les a fait rentrer dans le monde de l'information numérique. Ainsi toutes les informations traitées par les entreprises à travers leurs sites web sont des informations à caractère numérique.

- **Les risques liés au contrôle des systèmes d'information**

- **Vol/altération/modification** de données de l'entreprise par l'utilisation du site web par des employés (attaque interne) :

Nous sommes ici dans le cadre d'une malveillance interne. L'employé qui a accès au réseau interne de l'entreprise peut potentiellement l'utiliser pour lui faire du mal. La gravité dépend de l'information à laquelle il accède. Ce risque n'est pas à négliger et les employés ont généralement une bonne connaissance des failles de sécurité de l'entreprise (CIGREF, 2011).

- **Vol/altération/modification** de données de l'entreprise par l'utilisation du site web par des pirates (attaques externes).

Les attaques essuyées par les sites web d'entreprises du fait de hackers ne sont pas rares. Le pirate doit en effet avoir une raison particulière pour s'attaquer à l'entreprise. Si le phénomène «hacker» est très médiatisé, il n'en reste pas moins que les auteurs des actions les plus dommageables reste les employés de l'entreprise.

- **Vol/altération/modification** de données de l'entreprise par l'utilisation du système réseau par des programmes malveillants (virus).

Ce genre de dommage peut également être causé par tout type de programme malveillant. Si la menace est quotidienne, le dommage n'est généralement pas très important. La réponse à apporter est généralement assez simple.

- La négligence des salariés.

Au-delà des risques dus à la malveillance, on retrouve également des risques liés au comportement négligent des employés. Ce comportement est produit le plus souvent par une méconnaissance des enjeux de sécurité de l'entreprise pour le salarié ou par des usages de travail (partage des sessions, des mots de passe, etc.). Cette négligence peut entraîner vol/altération/modification des données de l'entreprise.

- Déni de service entrainé par la saturation du serveur web.

Utiliser le numérique entraine une utilisation croissante du réseau de l'entreprise, une sollicitation plus importante des processus numériques mis en place et cela peut entraîner une saturation. Cette saturation se traduit par un ralentissement (parfois une immobilisation) tel que l'entreprise n'est plus en mesure de produire, de communiquer ou de fournir son client. Ce risque de déni de service peut avoir une source interne (ex: saturation du logiciel de facturation) ou une source externe (ex: problème technique chez le sous-traitant en cloudcomputing).

- **Les risques éthiques et juridiques**

- Respect de la vie privée et confidentialité des données.

Lorsqu'une entreprise décide de numériser des données personnelles (relatives à des clients ou des collaborateurs...), elle a l'obligation de déclarer ces données à la CNIL, ainsi que d'en assurer la sécurité (contrairement à des données non numériques). L'entreprise fait donc face à des risques juridiques liés à cette obligation légale de sécurisation de ces données (CIGREF).

- **Les risques marketing**

- Risque de réputation.

Un site internet est un outil vulnérable. C'est un outil de communication et de marketing de premier plan qui peut être la cible de personnes malveillantes à l'égard de l'entreprise. Ceci est d'autant plus facile lorsque le site propose une plateforme de communication bottom-up. Que ce soit une campagne organisée ou de réels mécontentements, l'entreprise court un risque réel en termes d'image en offrant un espace de liberté au cœur de son outil de communication

Ces risques sont accrus par des vulnérabilités qui naissent lors du développement, de l'implémentation et de l'utilisation des sites web (CIGREF).

2. Les vulnérabilités liées aux sites web

La fondation OWASP (Open Web Application Security Project) dans son document « les dix vulnérabilités de sécurité applicatives web les plus critiques » présente quelques une des vulnérabilités des applications web notamment des sites web.[3]

Au nombre de ces vulnérabilités nous pouvons citer :

- **Les failles d'injection**

Les failles d'injection, en particulier l'injection SQL, sont communes dans les applications web. L'injection se produit quand des données écrites par l'utilisateur sont envoyées à un interpréteur en tant qu'élément faisant partie d'une commande ou d'une requête. Les données hostiles de l'attaquant dupent l'interpréteur afin de l'amener à exécuter des commandes fortuites ou changer des données.

Figure 1.4 : Principe d'attaque d'injection

- **Le cross site Scripting (XSS)**

Les failles XSS se produisent à chaque fois qu'une application prend des données écrites par l'utilisateur et les envoie à un navigateur web sans avoir au préalable validé ou codé ce contenu. XSS permet à des attaquants d'exécuter un script dans le navigateur de la victime afin de détourner des sessions utilisateur, défigurer des sites web, potentiellement introduire des vers, etc.

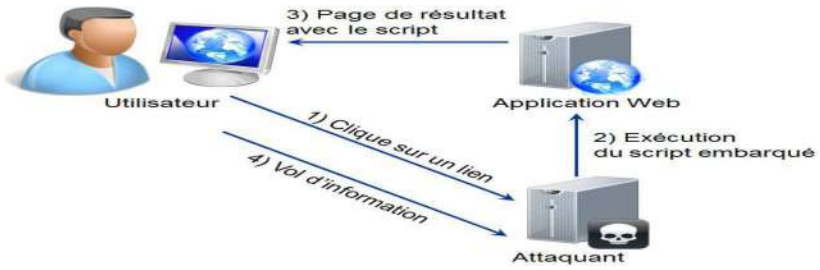

Figure 1.5 : Principe d'une attaque XSS par réflexion

[3]) Les dix vulnérabilités de sécurité applicatives web les plus critiques

- **Violation de gestion d'authentification et de sessions**

Les droits d'accès aux comptes et les jetons de session sont souvent incorrectement protégés. Les attaquants compromettent les mots de passe, les clefs, ou les jetons d'authentification identités pour s'approprier les identités d'autres utilisateurs.

Figure 1.6 : Principe de détournement de session

- **Référence directe non sécurisée à un objet**

Se produit quand un développeur expose une référence à une variable interne, tel un nom de fichier, de dossier, un enregistrement de base de données, ou une clé de base de données. Sans un contrôle d'accès ou autre protection, les attaquants peuvent manipuler ces références pour accéder à des données non autorisées.

- **Falsification de requêtes inter-sites (CSRF)**

Force le navigateur d'une victime authentifiée à envoyer une requête http, comprenant le cookie de session de la victime ainsi que toute autre information automatiquement inclue, à une application web vulnérable. Ceci permet à l'attaquant de forcer le navigateur de la victime à générer des requêtes, l'application vulnérable considérant alors qu'elles émanent légitimement de la victime.

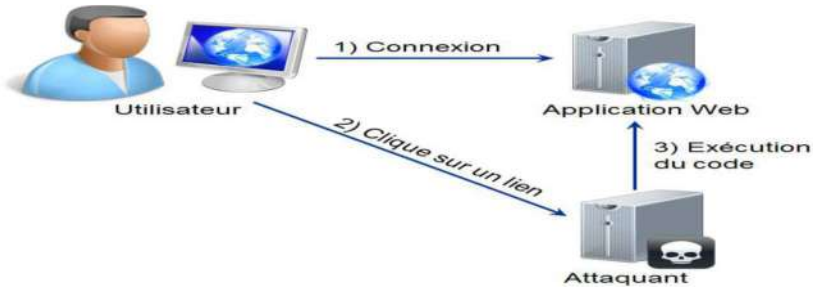

Figure 1.7 : Principe d'une attaque CSRF par réflexion

- **Exécution de fichier malicieux**

L'exécution de fichiers malicieux existe dans beaucoup d'applications. Les développeurs utilisent souvent la possibilité de concaténer des données d'entrée dans des fonctions de gestion de fichiers ou de flux, ou font confiance à des fichiers donnés en entrée. Lorsque les

données ne sont pas correctement vérifiées, cela peut conduire à l'exécution de code distant ou à l'affichage de contenu non voulu lorsde l'exécution par le serveur Web.

- **Stockage cryptographique non sécurisé**

Les applications web utilisent rarement correctement les fonctions cryptographiques pour protéger les données et les droits d'accès. Les attaquants utilisent des données faiblement protégées pour perpétrer un vol d'identité et d'autres crimes, tels que la fraude à la carte de crédit.

- **Manque de restriction d'accès d'URL**

Se produit quand une application web ne protège pas l'accès aux URL. Les applications doivent effectuer des contrôles d'accès similaires chaque fois que ces pages sont accédées, sinon les attaquants seront en mesure de forger des URL pour accéder à ces pages cachées.

- **Fuite d'information et traitement d'erreur incorrect**

Les applications peuvent involontairement dévoiler des informations sur leur configuration, fonctionnement interne, ou violer la vie privée par toute sorte de problèmes applicatifs. Les applications peuvent également dévoiler des informations sur leur état interne par l'intermédiaire du temps qui leur est nécessaire au traitement de certaines opérations ou via différentes réponses à des entrées différentes, par exemple afficher le même message d'erreur avec des numéros d'erreurs différents. Les applications web dévoileront souvent des informations sur leur état interne à cause de messages d'erreurs détaillés ou de débogage. Ces informations peuvent souvent être utilisées pour déclencher, voire même automatiser des attaques plus puissantes.

- **Communications non sécurisées**

Les applications ont souvent des défaillances lors du chiffrement du trafic réseau quand il est nécessaire de protéger des communications sensibles. Le chiffrement (habituellement SSL) doit être utilisé pour toutes les connexions authentifiées, et plus spécifiquement lors d'accès aux pages de sites WEB sans oublier les connexions vers les serveurs back-office. Sinon, l'application expose en clair les éléments d'authentification ou de session. En plus, le chiffrement doit être utilisé à chaque fois qu'il y a des informations sensibles, comme des informations de cartes de crédit, de cartes de santé ou lorsque des informations de santé sont transférées. Les applications qui permettent le retour arrière ou qui peuvent être contraintes de sortir d'un mode de chiffrement peuvent être attaquées par des pirates.

1.3.7. Différentes étapes d'une attaque :

Une attaque est l'exploitation d'une faille d'un système informatique connecté à un réseau. Pour réussir leur exploit, les attaquants tentent d'appliquer un plan d'attaque bien précis pour aboutir à des objectifs distincts.

La plupart des attaques, de la plus simple à la plus complexe fonctionnent suivant le même schéma :

Identification de la cible : cette étape est indispensable à toute attaque organisée, elle permet de récolter un maximum de renseignements sur la cible en utilisant des informations publiques et sans engager d'actions hostiles. On peut citer par exemple l'interrogation des serveurs DNS.

Scanning : l'objectif est de compléter les informations réunies sur une cible visée. Il est ainsi possible d'obtenir les adresses IP utilisés, les services accessibles de même qu'un grand nombre d'informations de topologie détaillée.

Exploitation : cette étape permet à partir des informations recueillies d'exploiter les failles identifiées sur les éléments de la cible, que ce soit au niveau protocolaire, des services et applications ou des systèmes d'exploitation présents sur le réseau.

Progression : il est temps pour l'attaquant de réaliser son objectif. Le but ultime étant d'obtenir les droits de l'utilisateur root sur un système afin de pouvoir y faire tout ce qu'il souhaite.

1.3.8. Différents types d'attaques

De nombreux types d'attaques du réseau ont été identifiés. Ces attaques sont généralement classées en trois principales catégories :

- Les attaques dans le but de découvrir des informations ;
- Les attaques par intrusions sont menées afin d'exploiter les faiblesses de certaines zones du réseau telles que les services d'authentification ;
- Les attaques d'interruption de service (ou déni de service) saturent l'accès à une partie ou à l'intégralité d'un système. Les attaques d'interruption de servicedistribué (DDOS : DistributedDenial of Service) qui consistent à saturer ainsiplusieurs machines ou hôtes, sont encore plus nuisibles.
- **Sniffing :** grâce à un logiciel appelé "sniffer", il est possible d'intercepter toutes les trames que notre carte réseau reçoit et qui ne nous sont pas destinées.

Si quelqu'un se connecte par Telnet par exemple à ce moment là, son mot de passe transitant en clair sur le net, il sera aisé de le lire. De même, il est facile de savoir à tout moment quelles pages Web regardent les personnes connectées au réseau, les sessions ftp en cours, les mails en envoi ou réception. Un inconvénient de cette technique est de se situer sur le même réseau que la machine ciblée.

- **IP spoofing:** cette attaque est difficile à mettre en œuvre et nécessite unebonne connaissance du protocole TCP. Elle consiste, le plus souvent, à se faire passer pour une autre machine en falsifiant son adresse IP de manière à accéderà un serveur ayant une "relation de confiance" avec la machine "spoofée".

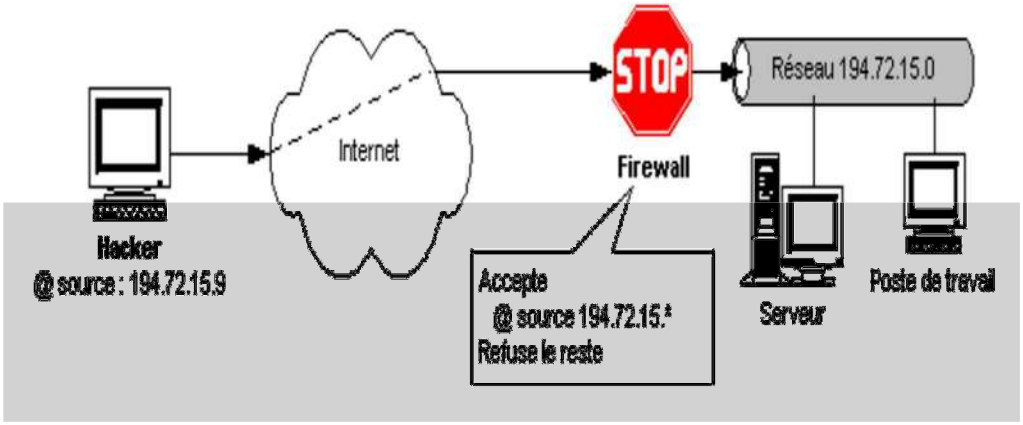

Figure 1.8 : Principe du spoofing

- **Scanners :** un scanner est un programme qui permet de savoir quels portssont ouverts sur une machine donnée. Les Hackers utilisent les scanners poursavoir comment ils vont procéder pour attaquer une machine. Leur utilisationn'est heureusement pas seulement malsaine, car les scanners peuvent aussipermettre de prévenir une attaque.
- **Attaques passives :** les attaques passives sont la capture du contenu d'unmessage et l'analyse de trafic. Elles sont très difficiles à détecter car elles necausent aucune altération des données. Le but de l'adversaire est d'obtenir uneinformation qui a été transmise.
- **Attaques actives :** ces attaques impliquent certaines modifications du flot dedonnées ou la création d'un flot frauduleux ; elles peuvent être subdivisées en 4catégories : mascarade, rejeu, modification de messages et déni de service.
  - Une mascarade a lieu lorsqu'une entité prétend être une autre entité. Une attaque de ce type inclut habituellement une des autres formes d'attaque active.
  - Le rejeu implique la capture passive de données et leur retransmission ultérieure en vue de produire un effet non autorisé.

- La modification du message (man in the middle) signifie que certaines portions d'un message légitime sont altérées ou que les messages sont réorganisés.
- Dénis de services [2]: d'une manière générale, l'attaque par déni de service (Denial of Service DoS) vise à rendre une application informatique incapable de répondre aux requêtes de ses utilisateurs par saturation de ses ressources.

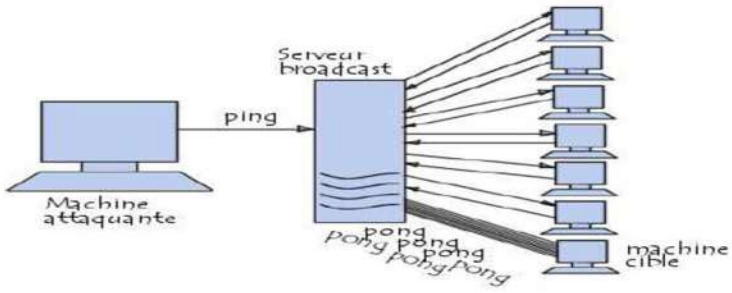

Figure 1.9. Exemple de DoS

1.3.9. Les mesures sécuritaires liées aux sites web

Il existe plusieurs techniques permettant d'assurer la sécurité des sites web et les informations numériques qu'ils contiennent. Ces mesures peuvent être classées par catégorie.

1. Les mesures de sécurité physique

Ces mesures protègent les supports de stockage et de traitement des données ou informations de façon physique. Si le site est hébergé sur des serveurs appartenant à l'entreprise elle-même, la solidité des bâtiments logeant les serveurs est obligatoire. La souscription d'un contrat d'assurance est importante.

Mais dans le cas où le site et hébergé sur des serveurs de prestataires de service, il convient de s'assurer que l'hébergeur garanti un ensemble de mesures de sécurité tant physiques que logiques.

2. Les mesures de sécurité logique

Pour la sécurité logique, les solutions des grands éditeurs sont :

- les codes malveillants : La solution de sécurité est l'usage d'un progiciel antivirus sur les postes clients ainsi que sur les serveurs, couplé à un dispositif pare-feu ou firewall.
- les facteurs humains : La sensibilisation et une grande vigilance du personnel permettent de se prémunir de l'ingénierie sociale et du phising. L'usage de proxy, d'un pare-feu, de sonde réseaux réduisent l'accès et détecte les intrusions.

- la compromission de l'information (défiguration du site web etc.) souligne que la limitation des accès aux utilisateurs est une mesure contre l'usurpation d'identité, «les contrôles d'accès utilisateurs limitent l'accessibilité aux données ou informations et privilégient une catégorie du personnel vue la confidentialité des données ou de l'information».

Le chiffrement ou cryptographie constitue un autre moyen de sécuriser l'accès logique aux informations. Grâce à l'usage de ses algorithmes, la cryptographie permet :

- l'authentification ;
- l'intégrité ;
- la non répudiation.

1.4. Conclusion

Ce premier chapitre nous a permis d'avoir des notions élémentaires par rapport aux sites web, à leur utilisation, développement, leurs importances dans le système d'information de l'entreprise, mais aussi et surtout à la sécurité qu'ils requièrent.

Il est donc important pour une entreprise qui possède un ou plusieurs sites web de s'assurer de leurs sécurités, il y va de la survie du système d'information de l'entreprise toute entière car le site web est « une partie visible de l'iceberg » qu'est le système d'information.

La négligence ou le manque de sécurité au niveau des sites web d'une entreprise peut lui porter de nombreux préjudices tes que :

- des lourdes pertes financières
- la perte d'informations sensibles
- la perte de notoriété

Chapitre 2

Recommandation et bonne pratique sur la sécurité site web

2.1. Introduction

Afin d'éliminer les vulnérabilités, contrer les attaques et garantir un niveau élevé de protection du site web, on peut utiliser des services, des mécanismes, des outils et des procédures que l'on nomme de façon générale des solutions ou des mesures de sécurité. Par exemple, les politiques de sécurité décrivent la manière dont les informations sensibles et les autres ressources sont gérées, protégées et distribuées à l'administrateur d'un site.

Dans ce cadre, le présent Chapitre se propose d'aider les responsables de la sécurité des systèmes d'information, à travers la présentation des règles de sécurité devant être respectées pendant les différentes phases du cycle de vie d'une application, à mieux sécuriser leurs applications web. Ainsi, le présent document est organisé en quatre parties :

- La première partie présente les recommandations de base à respecter, notamment les clauses de sécurité à intégrer dans le cahier des spécifications spéciales ainsi que la formation.
- La deuxième partie est consacrée aux meilleures pratiques permettant d'éviter les failles les plus connus dans le développement des applications web.
- Enfin, La troisième partie porte sur les bonnes pratiques à respecter lors du déploiement et la mise en production d'une application web.

2.2. Les normes de la famille ISO 2700X

Les normes de la famille ISO/IEC 2700x constituent un ensemble de méthodes, mesures et bonnes pratiques reconnues au niveau international dans le domaine de la sécurité de l'information. Elles sont destinées à tout type de société, quelle que soit sa taille, son secteur d'activité ou son pays d'origine. Ces normes ont pour but de décrire les objectifs à atteindre en matière de sécurité informatique.

ISO/CEI 27001 :

La norme ISO 27001 porte sur la politique du management de la sécurité des systèmes d'information dans les entreprises. Elle définit les contrôles de sécurité dont la mise en œuvre est exigée.

ISO/CEI 27002 :

La norme propose sur onze chapitres, une liste de 133 mesures de sécurité accompagnées chacune de points à aborder pour la mise en place d'un SMSI. Parmi ces chapitres, on a par exemple, la gestion des actifs, la sécurité physique, la sécurité des ressources humaines, la gestion des incidents, la continuité d'activité, la conformité etc.

ISO/CEI 27005:

Dans le cadre d'un audit de la sécurité d'un site web, il peut servirde base pour l'évaluation des risques. La norme est constituée de douze chapitres. Encomplément de ces chapitres, viennent s'ajouter six annexes proposant des explications plusdétaillées qui présentent des listes de menaces et vulnérabilités.

Le tableau 1 ci-dessous donne un aperçu des domaines qui seront couverts par les normes de la famille ISO/CEI 270xx.

PROJET	SECTEUR D'ACTIVITE
ISO/CEI 27010	Gestion de la communication inter secteu
ISO/CEI 27031	Continuité d'activité
ISO/CEI 27032	Cyber sécurité
ISO/CEI 27033	Sécurité réseau
ISO/CEI 27034	Sécurité des applications
ISO/CEI 27035	Gestion des incidents
ISO/CEI 27036	Audit des mesures de sécurité du SMSI
ISO/CEI 27037	Gestion des preuves numériques

Tableau 2.1 : Normes ISO/CEI 270xx en préparation[4]

2.3. Mécanismes de sécurité :

La sécurité vise à garantir la confidentialité, l'intégrité et la disponibilité des services. Il faut mettre en place des mécanismes pour s'assurer que seules les personnes autorisées ont accès à l'information et que le service est rendu correctement. Parmi ces mécanismes, on peut citer :

2.3.1. Cryptage :

Cryptographie est une science mathématique dans laquelle on fait les études des méthodes permettant de transmettre des données de manière confidentielle.

Afin de protéger un message, on lui applique une transformation qui le rend incompréhensible ; c'est ce quon appelle le chiffrement, qui, à partir dun texte clair, donne un texte chiffré ou cryptogramme. Inversement, le déchiffrement est l'action qui permet de reconstruire le texte en clair à partir du texte chiffré. Dans la cryptographie moderne, les

[4]) Mise en œuvre de la SSI (Sécurité du Système D'information) de SUSS microOptics par l'approche Processus ISO/CEI 27001

transformations en question sont des fonctions mathématiques, appelées algorithmes cryptographiques, qui dépendent d'un paramètre appelé clef.

La Figure montre le fonctionnement de chiffrement.

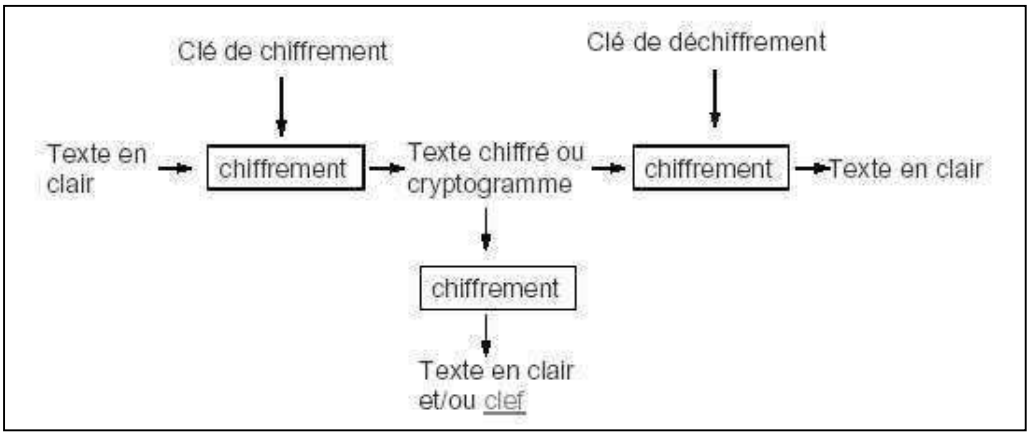

Figure 2.1 : Chiffrement

2.3.2. Pare-feu :

Un pare-feu (firewall) est une solution matérielle ou logicielle mise en place au sein de l'infrastructure du réseau afin de filtrer l'accès à des ressources réseau définies. Il ne laisse entrer que les utilisateurs autorisés, disposant d'une clef ou d'un badge, et crée une couche protectrice entre le réseau et le monde extérieur. Il est doté de filtres intégrés qui peuvent empêcher des documents non autorisés ou potentiellement dangereux d'accéder au système. Il enregistre également les tentatives d'intrusions dans un journal transmis aux administrateurs du réseau. Il permet également de contrôler l'accès aux applications et d'empêcher le détournement d'usage.

Le pare-feu permet à laisser passer tout ou partie des paquets qu'ils sont autorisés, et à bloquer et journaliser les échanges qui sont interdits.

La Figure 2.2 schématise le fonctionnement d'un pare-feu.

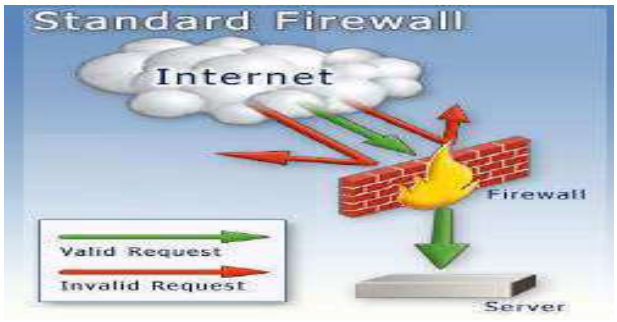

Figure 2.2 : Pare-feu.

2.3.2 Antivirus :

Un antivirus est un logiciel qui protège une machine contre les virus. Les antivirus se fondent sur des fichiers de signatures et comparent alors les signatures génétiques duvirus aux codes à vérifier. Certains programmes appliquent également la méthode heuristique tendant à découvrir un code malveillant par son comportement.

2.4. Politique de sécurité :

La fonction de gestion des politiques de sécurité doit donc être confiée à des personnes particulièrement dignes de confiance et disposant des compétences techniques nécessaires.

2.4.1. Mots de passe : le moyen le plus simple et le plus classique de s'assurer que seules les personnes autorisées peuvent accéder à une certaine partie du réseau est de protéger certaines zones du réseau par un mot de passe. Cependant, les plus puissantes infrastructures de sécurité sont inefficaces si les mots de passe ne sont pas eux-mêmes protégés.

Les règles d'or ou politiques à suivre, en matière de mots de passe sont les suivants :

- Changer régulièrement les mots de passe.
- Choisir des mots de passe aussi dénués de sens que possible.
- Ne jamais divulguer les mots de passe.

2.4.2. Certificat SSL : La technologie SSL a été conçue pour éviter que des données sensibles, soient transmises « en clair » sur Internet, et les sites commerciaux arborent aujourd'hui tous un logo « site sécurisé par un certificat SSL ».

Un certificat SSL est un fichier de données qui lie une clé cryptographique aux informations d'une organisation ou d'un individu. Installé sur un serveur, le certificat active le cadenas et le protocole HTTPS via le port 443 dans les navigateurs, afin d'assurer une connexion sécurisée entre le serveur web et le navigateur.

Figure 2.3 : Logo mode sécurisé par SSL

2.4.2. Le fichier .htaccess : Les fichiers .htaccess sont des fichiers de configuration d'Apache, permettant de définir des règles dans un répertoire et dans tous ses sous-répertoires (qui n'ont pas de tel fichier à l'intérieur). On peut les utiliser pour protéger un répertoire par mot de passe, ou pour changer le nom ou l'extension de la page index, ou encore pour interdire l'accès au répertoire.

2.4.3. Le Captcha : Un CAPTCHA (CompletyAutomated Public Turing test to tell Computers and HumansApart) est un système de sécurité et un mécanisme mis en place sur une page web présentant un formulaire se présente sous forme de question/réponse pour s'assurer que c'est bien un être humain qui valide les champs.

ette méthode met en place une image sur laquelle sont inscrit des chiffres et lettres présentant une distorsion et un fond de couleurs dégradées, seul un humain peut donc déchiffrer cette image.

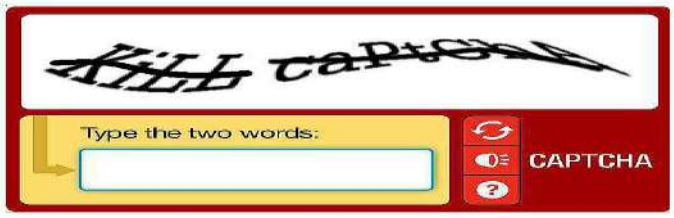

Figure 2.4 : modele de Captcha

2.4.4. La sécurité englobe la sécurité des systèmes information (Le périmètre et la segmentation)

Sécuriser un système impose de maîtriser son cycle de vie et son utilisation à bon escient, de contrôler son fonctionnement, de pérenniser son évolution et consiste à s'assurer que les ressources matérielles et logicielles ainsi que les informations d'une personne ou d'une organisation sont strictement utilisées dans le cadre général qui est prévu. La sécurité est une propriété qui contribue à l'intégrité d'un système dans son acception la plus large.

La sécurité englobe la sécurité des systèmes d'exploitation (OS), des logiciels, des communications interpersonnelles, de la messagerie, des données elles-mêmes, du partage des connaissances et de la propriété intellectuelle. La sécurité de l'informatique et des télécoms conjugue la liberté et la volonté de protéger les valeurs matérielles ou intangibles et leur image de marque, avec la correction des logiciels, la robustesse des architectures, l'immunité des applications, la résilience des systèmes, l'instillation et le maintien de la confiance dans les édifices numériques.

La sécurité se décompose en plusieurs volets :

- la sécurité physique des lieux, des personnes et des biens, des infrastructures et des ressources matérielles, relative à des accidents (dégâts des eaux, sinistres) ou des sabotages.
- la sécurité logique, la sûreté de fonctionnement, la fiabilité des systèmes embarqués, relative à la bonne marche, à la robustesse ou à la survie d'un système, suite à des dysfonctionnements internes ou externes ou des perturbations accidentelles ou intentionnelles de l'environnement.
- la sécurité des infrastructures, des systèmes de télécommunication, des réseaux et des systèmes répartis, utilisant une informatique prépondérante en réseau, relative à la perturbation par des attaques ou des propagations d'erreurs via le réseau .
- la sécurité des SI de nature personnelle, technique, bureautique ou administrative, relative à la divulgation d'informations confidentielles ou à la corruption de base de données.

2.4.5. Les mesures de protection contre les attaques les plus connus :

Les attaques d'un système d'information visent :

L'intégrité des données :

- Modification des données publiées sur le site (défiguration).
- Modification ou suppression d'informations confidentielles.

La confidentialité : obtention de données sur **:**

- Le client (n° sécurité sociale, carte bancaire, coordonnées, …).
- Les visiteurs du serveur web (logs).
- L'organisation (accès à des données sensibles, au réseau local de l'organisation).
- Le serveur (accès aux mots de passes, aux fichiers de configuration en vue d'une attaque).

La disponibilité des données :

- Bloquer l'accès au site web (DoS).
- Ou à un utilisateur en particulier.

L'attaque d'un serveur web peut également être destinée à prendre le contrôle du serveur pour attaquer d'autres sites ou installer des services.

2.4.6. Protection contre les dix vulnérabilités de sécurité applicatives web les plus critiques déclarés par l'OWASP

1. Protection contre les failles d'injection

Meilleure méthode pour éliminer les failles d'injections est l'utilisation d'API sûrs, telles les requêtes fortement typées et les librairies d'objets de cartographie relationnels (ORM). Ces interfaces gèrent toute évasion de donnée, ou ne requièrent pas d'échappements. Notez que bien que les interfaces sûres résolvent le problème, la validation est toutefois recommandée afin de détecter les attaques.

- Validation des données d'entrée : utilisez un mécanisme standard de validation des entrées pour valider toutes les données d'entrée pour la longueur, le type, la syntaxe et les règles métiers avant d'accepter l'affichage ou le stockage de donnée.
- Utilisez des APIs de requêtes fortement typées : avec des marqueurs de substitution dédiés, même lors d'appels de procédures stockées.
- Utilisez des procédures SQL stockées : car elles généralement non vulnérables à l'injection SQL.
- N'utilisez pas les fonctions d'échappements simples : comme la fonction PHP addslashes() ou les fonctions de remplacement de caractères comme str_replace ("'", "''"). Elles sont faillibles et ont déjà été utilisées avec succès par des attaquants.

2. Protection contre le cross site Scripting (XSS)

La meilleure protection contre Cross Site Scripting XSS est une combinaison de validation de type « whitelist » de toutes les données entrantes et du codage approprié de toutes les données produites en sortie. La validation permet la détection d'attaques, et l'encodage empêche toute injection de script de se produire dans le navigateur.

La protection d'une application complète contre XSS exige une approche architecturale cohérente :

- Validation d'entrée. Utiliser un mécanisme standard de validation d'entrée pour valider la longueur, le type et la syntaxe de toute entrée saisie, ainsi que les règles de gestion avant d'accepter l'affichage ou le stockage des données.
- Chiffrement robuste des données en sortie. Assurez-vous que toutes les données écrites par l'utilisateur sont convenablement codées par entité (soit HTML ou XML selon le mécanisme de présentation) avant le rendu, prenant le parti de coder tous les

caractères plutôt qu'un sous-ensemble très limité. C'est l'approche de la bibliothèque Anti-XSS de Microsoft, et la prochaine bibliothèque Anti-XSS PHP de l'OWASP.

3. Protection contre violation de gestion d'authentifications et de sessions

L'authentification repose sur la sécurité des communications et du stockage des informations d'identification.

- Assurez-vous d'abord qu'SSL est la seule option pour toutes les parties authentifiées de l'application.
- N'acceptez pas des identifiants de sessions, nouveaux, préréglés ou invalides à partir de l'URL ou dans la requitte
- Assurez-vous que toutes les pages disposent d'un lien de déconnexion. La déconnexion doit totalement détruire tout état de session côté serveur et cookies côté client. Considérez aussi le facteur humain :demandez pas confirmation car les utilisateurs fermeront juste l'onglet ou la fenêtre plutôt que déconnecter proprement.
- Vérifier l'ancien mot de passe lorsque l'utilisateur change pour un nouveau mot de passe.

4. Protection contre référence directe no sécurisée a un objet

La meilleure protection est d'éviter d'exposer une référence directe à un objet à l'utilisateur, grâce à l'utilisation d'un index, une équivalence par référence indirecte ou une autre méthode indirecte facile à valider. Si une référence directe à un objet doit être utilisée, vérifiez que l'utilisateur est autorisé avant de l'utiliser.

La mise en place d'une méthode de référence aux objets d'application est importante :

- Evitez d'exposer des références d'objet privé aux utilisateurs, chaque fois que possible, tels les clés primaires ou noms de fichiers.
- Validez sans retenue toutes les références aux objets privés, via la méthode d'acceptation des bonnes valeurs.
- Vérifiez l'autorisation à tous les objets référencés

5. Protection contre falsification de requêtes inter-sites

La solution est d'utiliser un token personnalisé que le navigateur ne pourra «mémoriser» et donc pas envoyer automatiquement par une attaque CSRF.

Les stratégies suivantes devraient être inhérentes à toutes les applications web **:**

- S'assurer qu'il n'y a pas de vulnérabilité de type XSS dans l'application.

- N'utilisez pas de requêtes GET (URLs) pour les données sensibles ou pour n'effectuer des transactions de valeur. Utiliser uniquement la méthode POST lorsque vous recevez des données de l'utilisateur.

6. Mauvaise configuration de sécurité

En général, une application bien écrite ne doit pas utiliser une donnée entrée par un utilisateur comme nom de fichier pour une ressource du serveur (tel que les images, les documents XML et XSL, ou l'inclusion de scripts), et il est nécessaire de mettre en place des règles de filtrage permettant de limiter les connexions sortantes vers l'Internet ou vers d'autres serveurs internes.

- Validez fortement les données d'entrées en utilisant la stratégie d'acceptation uniquement des bonnes valeurs.
- Ajoutez des règles à vos Firewalls pour empêcher les serveurs web d'effectuer de nouvelles connexions vers des sites web externes ou internes.
- Vérifiez que les fichiers ou noms de fichiers fournis par l'utilisateur ne puissent se soustraire à d'autres contrôles, tels que altération de données dans l'objet de session, avatars et images, rapports PDF, fichiers temporaires, et ainsi de suite.

7. Protection contre stockage de données cryptographiques non sécurisé

Les recommandations suivantes doivent être considérées comme une partie de votre stratégie de test pour s'assurer que les données cryptographiques sont manipulées de manière sécurisée :

- Ne pas créer d'algorithmes de chiffrement. Utiliser seulement des algorithmes reconnus publiquement.
- Assurez-vous que les données chiffrées sur disque ne sont pas faciles à décrypter

8. Protection contre défaillance dans la restriction des accès url

Prendre le temps de planifier les autorisations en créant une matrice pour mettre en correspondance les rôles et les fonctions des applications est une étape clé pour assurer la protection des accès URL non restreints. Les applications WEB doivent renforcer le contrôle d'accès à chaque URL et chaque fonction métier.

- Assurez-vous que toutes les URLs et les fonctions métier sont protégées par un mécanisme de contrôle d'accès efficace qui vérifie le rôle de l'utilisateur et les droits associés avant chaque traitement

- Effectuez un test de pénétration avant chaque déploiement ou livraison de code pour vous assurer que l'application ne peut pas être utilisée à mauvais escient par des attaquants mal intentionnés.
- Ne supposez pas que les utilisateurs ne connaissent pas les URLs ou les APIs spéciales ou cachées. Assurez-vous toujours que les actions d'administration ou nécessitant des privilèges importants sont protégées.
- Bloquez l'accès à tous les types de fichier que votre application n'utilisera jamais.
- Être à jour à niveau concernant la protection antivirus et des patches.

9. Fuite d'information et traitement d'erreur incorrect

Protection contre les développeurs devraient utiliser des outils comme WebScarab (OWASP) pour faire générer des erreurs à leur application. Les applications qui n'ont pas été testées de cette façon généreront certainement des erreurs inattendues. Les applications devraient aussi intégrer une architecture standard de traitement d'exceptions afin d'empêcher toute fuite d'information non désirée à destination des attaquants.

- désactivez ou limiter le traitement d'erreur détaillé. En particulier, ne pas afficher les informations de débogage aux utilisateurs, l'état de la pile ou des informations relatives aux chemins.
- Différentes couches peuvent retourner des erreurs fatales ou une exception, telle la couche base de données. Il est crucial que les erreurs venant de chaque couche soient dûment vérifiées et configurées pour empêcher les messages d'erreur d'être exploités par des intrus.

10. Communications non sécurisées

La protection la plus adaptée est l'utilisation de SSL pour toutes les connexions authentifiées et à chaque fois qu'une donnée sensible est transmise. La configuration de SSL pour les applications WEB nécessitent de maîtriser pas mal de détails, donc il est important de bien comprendre et analyser votre environnement.

- Utiliser SSL pour toutes les connexions qui sont authentifiées ou qui transmettent des données sensibles, comme des éléments d'authentification, les détails d'une carte de crédit, d'une carte de santé ou autres informations privées.
- Assurer que les communications entre les composants d'une infrastructure, comme entre des serveurs WEB et des serveurs de bases de données, sont protégées de façon appropriée

grâce à l'utilisation d'une couche de transport sécurisée ou d'un chiffrement au niveau protocole pour les éléments d'authentification et les données intrinsèques.

2.4.7. Protection contre des types de menaces définie par le WASC

Nous reprenons dans cette partie la classification des types de menaces définie par le WASC (*Web Application Security Consortium*) et les bonnes pratiques afin de les évitées :

1. Attaques liées à l'authentification

Le but de ces attaques est l'accès à une application web protégée.

a) Force brute

Emploi d'un **processus automatique pour trouver les informations protégeant un système** (login, mot de passe, clé cryptographique). Généralement le pirate cherche un mot de passe pour un login fixé.

Protection :

- Limiter le nombre d'essais lors de l'authentification (bloquer l'accès pendant 15mn après 3 échecs) et répondre après un délai de 3 secondes lors du 1er essai, 15 lors du 2d et 30 pour le 3ème.
- Conserver des traces de toutes les tentatives de connexion.
- Imposer une taille minimale pour le login et le mot de passe.
- Imposer une complexité minimale (nombre de chiffres, caractères spéciaux, …).
- Ne jamais indiquer si c'est le login ou le mot de passe qui est erroné ;
- Ne jamais conserver les comptes avec un login et mot de passe par défaut.
- utiliser le module mod_evasive qui protège contre des attaques de force brute et de déni de service.

b) Authentification insuffisante

Certaines applications insuffisamment protégées permettent l'**accès à des ressources à des personnes non authentifiées** :

Il est possible d'accéder à l'intranet :

- depuis un listing de répertoire s'il n'y a pas de fichier index.html ;
- par une attaque de force brute recherchant les noms de répertoires d'administration les plus courants (/admin/administrateur /intranet / …) ;
- en interne :
    - depuis un bookmark ou l'historique de navigation sur un ordinateur en accès libre.

- en notant l'URL affichée dans le navigateur lorsque l'administrateur est connecté.

Protection : utiliser un .htaccess (protection du répertoire et de ses sous-répertoires) ou des sessions.

c) Mauvais traitement des recouvrements de mot de passe :

Une application web doit gérer le recouvrement des mots de passe des utilisateurs. La méthode la plus simple et la plus sure serait de réaliser une nouvelle inscription mais les données de l'ancien login seraient perdues. Pour recouvrer le mot de passe plusieurs méthodes sont utilisées :

- Partage de secret : lors de la création du compte l'application pose plusieurs questions personnelles à l'utilisateur, lors de la procédure de recouvrement l'utilisateur doit répondre à une ou plusieurs des questions posées lors de l'inscription. Ces questions ne doivent pas porter sur des données qui peuvent être obtenues par un pirate (adresse mail, adresse personnelle, numéro de tel …). Plus le pirate connaît personnellement la victime plus il a de chance de pouvoir répondre aux questions.

- Envoi par mail d'un nouveau mot de passe à l'adresse mail fournie lors de l'inscription. Le procédé ne nécessite aucune intervention humaine. Le seul inconvénient est que le mot de passe peut être obtenu par une écoute du réseau ou lu sur l'ordinateur de l'utilisateur.

Bonnes pratiques :

- toujours fournir un nouveau mot de passe quand il a été perdu
- conserver toutes les demandes de recouvrement de mot de passe ;
- limiter le mot de passe à une utilisation unique, l'utilisateur devra obligatoirement le changer lorsqu'il se connectera avec son nouveau mot de passe.

2. Attaques liées aux autorisations

d) Expiration de session

Plus la **durée de validité d'une session** est courte, plus il sera difficile pour un pirate de détourner la session.

Beaucoup d'attaques sont possibles car les données de sessions restent présentes sur le serveur web (pas supprimées après la session). La session d'un utilisateur ne devrait plus être valide au bout d'une durée fixée selon le type d'application.

3. Attaques côté client

a) Usurpation de contenu (content spoofing)

Attaque consistant à faire croire à un utilisateur que le contenu apparaissant sur le site web est légitime et ne vient pas d'une source extérieure. Utilisation de frames, iframe, XSS, click jacking (utiliser CSS et des iframes pour placer un contenu invisible sur un contenu visible sur lequel l'utilisateur cliquera (but détourner le clic).

b) *XSS*

Attaque qui a pour but de faire **exécuter** un **code** malveillant par le **navigateur** du client

Bonne pratique :

- filtrer les entrées de l'application ($_GET, $_POST, $_COOKIE) en utilisant une approche par liste blanche : ce qui est autorisé et non pas ce qui est interdit (moins de liberté mais plus sur) : par exemple, un nom comporte uniquement des caractères de l'alphabet et des espaces (s'il le faut on ajoutera plus tard des caractères).
- protéger les données envoyées vers le navigateur.
- utiliser les fonctions prédéfinies utf8_decode et strip_tags (suppression des balises dans la chaîne).
- utiliser des noms d'identificateurs spécifiques pour repérer les données pas encore vérifiées.

4. Attaques par exécution de commandes ou de requêtes

a) Débordement de tampon

Attaque visant à placer dans la mémoire un code arbitraire par débordement de tampon (difficile à réaliser provoque plus souvent un déni de service par *segmentation fault*).

b) Injection LDAP

Attaque concernant les applications qui construisent dynamiquement des requêtes LDAP.

c) Injection de commandes

Attaque concernant les applications qui exécutent des commandes dans un shell ou qui exécutent des fichiers inclus.

d) Injection SQL

Attaque concernant les applications qui construisent dynamiquement des requêtes SQL

Bonne Pratique :

- Préciser le ou les répertoires d'inclusion autorisés pour le site avec la directive open_basedir.
- Effectuer les mises à jour des briques logicielles et les correctifs de sécurité.

- Mettre en place des règles de filtrage sur le pare-feu du serveur web pour limiter les connexions vers d'autres sites web.

3. Attaques logiques :

Ces attaques concernent l'abusl'exploitation du flux logique de l'application web (procédure de récupération d'un mot de passe oublié, enregistrement de comptes, …)

a) Abus de fonctionnalité

Attaque qui utilise les caractéristiques et fonctionnalités du site web. Voici quelques exemples :

- utiliser une fonction de recherche du site web pour accéder à des fichiers en dehors du répertoire.
- remplacer un fichier de configuration du système en faisant un fichier uploadé.
- déni de service en envoyant plusieurs mots de passe faux pour des utilisateurs existants afin de bloquer leurs comptes.

b) Déni de service (DoS)

Attaque qui a pour but d'empêcher le serveur de répondre aux clients. Le déni de service est provoqué par la **consommation excessive de ressource** (CPU, mémoire, bande passante, espace disque, …). . L'attaque peut viser :

- un utilisateur en particulier (invalidation du mot de passe) ;
- le serveur de base de données (injection SQL pour amener le serveur à une charge maximale, grand nombre de requêtes à un site web qui utilise un SGBD pour produire les pages, …).

Bonne pratique :

Utiliser un outil pour mesurer les performances lors de la montée en charge permet de donner une idée du nombre de requêtes qu'un pirate aura à générer pour obtenir un déni de service.

Régler les directives du httpd.conf qui limitent les ressources utilisées et le nombre de clients et de connexions persistantes simultanés (MaxClients, MaxRequestsPerChild, Timeout, KeepAliveTimout, RLimitMEM, RLimitCPU, LimitRequestBody, LimitRequestFields, LimitRequestFieldSize, LimitRequestLine).

Régler les directives du php.ini memory_limit, post_max_size, max_input_time, max_execution_time.

En résumé, la plupart des attaques sont rendues possibles par :

- un contrôle des données inexistant ou insuffisant.
- une mise à disposition de données sensibles (traitement des erreurs inadapté).
- des contrôles d'autorisation ou d'authentification inexistants ou insuffisants.

2.5. Exemples de mesures de sécurité courantes

Voici ci-après, quelques pratiques générales en matière de sécurité informatique qui sont fréquemment mises en place dans la sécurisation du SI de nos unités de recherche.

2.5.1. Sécurité physique des locaux

L'objectif est d'empêcher tout accès physique non autorisé, tout dommage ou intrusion dans les locaux dans lesquels résident les informations de l'unité. Les locaux contenant des informations sensibles et des moyens de traitement de l'information (salles serveurs, secrétariat de direction ou d'enseignement…) doivent donc être protégés physiquement des accès incontrôlés ou malveillants (contrôle d'accès par carte ou code).

2.5.2. Sécurité du matériel et du câblage

On protégera les matériels sensibles (routeurs, serveurs…) des pertes d'alimentation électrique par un système de secours bien dimensionné, ainsi que d'éventuelles surchauffes par des moyens de climatisation adéquats et bien dimensionnés.

Afin de garantir une disponibilité permanente et un bon fonctionnement en cas de panne, le matériel sensible qui nécessite un fonctionnement continu doit être placé sous contrat de maintenance.

Les accès aux câbles réseaux transportant des données doivent être protégés contre toute possibilité d'interception de l'information, ou de dommage. Les câbles ou concentrateurs réseaux doivent être hors de portée immédiate et donc protégés dans des gaines ou des armoires de répartition.

2.5.3. Procédures de sécurité informatique liées à l'exploitation

1. Protection contre les codes malveillants : virus et autres « malwares »

La plupart des attaques via le réseau tentent d'utiliser les failles du système d'exploitation ou applications. Les attaques recherchent les ordinateurs dont les logiciels ou des applications n'ont pas été mis à jour afin d'utiliser la faille non corrigée et ainsi parvenir à s'y introduire. C'est pourquoi il est fondamental que les administrateurs mettent à jour les logiciels des serveurs et des postes clients afin de corriger ces failles.

2. Sauvegarde des informations

La sauvegarde des informations est un processus essentiel permettant de garantir la disponibilité des données et la continuité de l'activité du site en cas d'incident.

Une sauvegarde régulière des données des utilisateurs ainsi qu'un processus de restauration, testés au préalable, doivent être mis en place. Les droits d'accès à ces sauvegardes doivent faire l'objet d'une attention particulière.

Des copies de ces sauvegardes doivent être réalisées sur des supports externes (robot de bandes, disques externes…) et placées dans des locaux (ou coffres) sécurisés et distants. Ces copies de sauvegardes doivent aussi être testées régulièrement conformément à la politique de sauvegarde convenue.

3. Journaux systèmes – les logs

Les journaux systèmes produits par nos serveurs informatiques permettent la surveillance du contrôle d'accès à nos systèmes et réseaux.

Les journaux systèmes qui enregistrent les activités des utilisateurs, les exceptions et les événements liés à la sécurité doivent être produits et conservés pendant la période légale pour surveiller l'exploitation du système.

Il est important de protéger les serveurs qui conservent les informations journalisées contre des accès non autorisés ou des actes de malveillance qui pourraient s'opposer au maintien de la preuve.

4. Synchronisation des horloges

En cas d'analyse des journaux informatiques, pour retracer la chronologie d'un événement ou d'une anomalie, il est essentiel que les horloges des différents systèmes de traitement de l'information (serveurs, routeurs, PC utilisateurs…) de nos unités de recherche soient synchronisées à l'aide d'une source de temps précise et préalablement définie.

5. Protection des transferts de données : chiffrement

L'objectif des mesures cryptographiques est de protéger la confidentialité, l'authenticité ou l'intégrité de l'information par des algorithmes utilisant des clés de chiffrement. Aussi, il faut les utiliser pour protéger les flux d'information liés à des services sensibles. Par exemple, la messagerie électronique ou les accès intranet ou tout autre service demandant une identification doivent être protégés de manière adéquate par des protocoles sécurisés reposant sur SSL, comme IMAPS, SSTPS, SASL pour la messagerie ou HTTPS pour le web.

2.5. Conclusion

La sécurité d'un site n'est pas à prendre à la légère, c'est quelque chose de très compliqué qui requiert des connaissances techniques approfondies afin de pouvoir identifier les vulnérabilités et mettre en place les mesures de protection nécessaires.

L'objectif de ce chapitre était de vous faire prendre conscience des problèmes de sécurité que nous pouvons rencontrer sur le web.

Après qu'il faut filtrer tout ce qui vient de client et ne pas négliger la configuration du serveur. On a donné quelques simples conseils sous forme d'une liste de bonnes pratiques qu'un professionnel doit appliquer à la rigueur pour se défendre des attaques automatiques et empêcher ceux qui visent votre site web.

Chapitre 3

Audit de la sécurité du site web de l'UKMO

3.1. Introduction

Dans ce chapitre on va aborder le cas de site web de **l'Université Kasdi Merbah Ouargla** et on va citer les différentes failles de sécurité et les types des attaques qui ont été enregistrés par les administrateurs systèmes et réseaux au seine du Centres des Réseaux et Systèmes.

Ce chapitre sera l'occasion donc de mettre en pratique les aspects théoriques développés dans le premier chapitre. Il consistera dans un premier temps à présenter l'établissement UKMO ; le site web qui fera l'objet de l'étude ; ensuite nous allons mettre en pratique l'audit de la sécurité du site web et en formulant des recommandations pour l'amélioration de la sécurité d'un site web Joomla.

3.2. Présentation de l'Université Kasdi Merbah Ouargla

L'université Kasdi Merbah Ouargla tire ses origines de l'Ecole Normale Supérieure (E.N.S.) érigée par le Décret n° 65-88 du 22 mars 1988. En 1997, par le Décret n°159-97 du 10 mars 1997 est créé le Centre Universitaire de Ouargla qui regroupe désormais sous son autorité l'Institut national de Formation Supérieure en Agronomie Saharienne (INAFSAS – Décret n°337-97 du 10 septembre 1997) au côté des 05 instituts fondateurs:

- Institut d'Agronomie saharienne
- Institut des sciences exactes
- Institut de Droit et sciences politiques
- Institut des Lettres et Langues
- Institut des Sciences Economiques et Sciences Sociales

L'Université kasdi Merbah Ouargla se compose actuellement de 10 facultés ; chacune d'elles compte plusieurs départements conformément aux textes en vigueur, notamment le Décret exécutif n° 13-100 du 14 mars 2013 modifiant et complétant le Décret exécutif n° 01-210 du 23 juillet 2001 portant création de l'université de Ouargla.. Le nombre et la vocation des facultés composant l'Université Kasdi Merbah Ouargla sont fixés comme suit:

- Faculté des Mathématiques et des Sciences de la Matière
- Faculté des Nouvelles Technologies de l'Information et de la Communication
- Faculté des Sciences Appliquées
- Faculté des Hydrocarbures, des Energies Renouvelables, des Sciences de la Terre et de l'Univers
- Faculté des Sciences de la Nature et de la Vie

- Faculté des Sciences Economiques, Sciences Commerciales et des Sciences de Gestion
- Faculté Droit et Sciences Politiques
- Faculté des Lettres et des Langues
- Faculté des Sciences Humaines et Sociales
- Faculté de Médecine
- et deux instituts à savoir:
- Institut des Sciences et Techniques des Activités Physiques et Sportives
- Institut de Technologie

Le rectorat de l'Université Kasdi Merbah Ouargla se situe sur la route de Ghardaïa .

Trois grands campus composent l'architecture principale de l'Université. Depuis le 05 septembre 2005, l'Université porte désormais le nom de "**Université Kasdi Merbah Ouargla**".

3.3. Présentation générale du site web de l'Université Kasdi Merbah Ouargla

Le site web de L'Université KasdiMerbah Ouargla est un site dynamique basé sur le CMS Joomla!et hébergé localement sur un serveur de type LAMP ; dans ce qui suit on va présenter on détail la plateforme de développement de ce site.

Après les interlocutions avec les administrateurs systèmes et réseaux de CRI de l'UKMO et le webmaster de l'université on a arrivé à recueillir les informations nécessaire pour faire l'audit en question.

La présentation du présent site est basée sur les informations recueillies suite à l'administration d'un questionnaire de prise de connaissance au directeur technique, en charge de l'administration du site suivis d'entretiens.

La présentation générale se fera autour de critères tels que :

- Les objectifs du site ;
- Historique de site ;
- La navigation ;
- La structure ;
- La technologie utilisée ;
- L'hébergeur.

3.3.1. Objectifs du site

La connaissance des objectifs du site web de L'Université Kasdi Merbah est primordiale dans l'étude que nous menons. En effet elle nous permettra de nous enquérir de la stratégie que sous-tend le site web et nous permettra de formuler des recommandations en adéquation avec les moyens et les objectifs de la direction générale. Le site web de L'Université Kasdi Merbah **(https:// www.univ-ouargla.dz)** a pour objectif de :

- présenter l'établissement, sont services, son savoir-faire et son organisation ;
- faire connaître l'université au-delà des frontières ;
- Fournir des informations pertinentes et à jour à un large public ;
- Informer les étudiant, les enseignants, chercheurs à l'occasion d'un événement particulier, des formations et des actualités ;
- Informer les étudiants sur les annonces de l'inscription, commencement des cours, les notes d'examen et les moyens,
- Consulter et télécharger les Cours, les thèses, et les différents documents de l'université.
- fournir l'information d'une manière accessible et utilisable.

3.3.2. Historique du site

Année	**OS**	**Joomla!**
2009	Windows XP sur un PC local	1.5
2013	RedHat 6 dans le Datacenter	2.5
2014	CentOS 6 dans le Datacenter	2.5
2016	CentOS 7 dans le Datacenter	2.5
2017	CentOS7 dans le Datacenter	3.6

Tableau 3.1 : Evolution site de l'Université Kasdi Merbah Ouargla

3.3.3. La navigation

Le site web est composé de plusieurs zones de navigation, c'est-à-dire d'emplacements spécifiques dont le but est de proposer un mode de navigation. Le mode de navigation privilégié est la visite guidée. Dans ce sens, le site présente les menus suivants :

Accueil : il s'agit d'une page d'accueil qui indique aux visiteurs qu'il s'agit du site d'UKMO.

Services : il s'agit d'une page qui présente tous les services proposés par UKMO.

Organes de l'établissement : les différents organes de l'université comme le rectorat, le Conseil Pédagogique de l'Université et le Conseil scientifique de l'université

Facultés et instituts : les liens vers les sites des facultés et instituts de l'université

Services : le centre d'examen et le Centre d'Enseignement Intensif des Langues CEIL

Vie estudiantine : un espace pour les étudiants

Contact : une page est entièrement dédiée aux contacts de l'établissement avec la possibilité de laisser des messages écrit.

3.3.4. La structure du site

La capture d'écran ci- dessous illustre la page d'accueil du site web de l'établissement. Cette dernière présente l'activité en cour.

Figure 3.1 : La page d'accueil du site web de l'Université Kasdi Merbah Ouargla

3.3.4. Le type des pages

Le site contient des pages statiques représente chacune d'elle un contenu pertinent de l'établissement et des mises à jour sont faites en fonction de l'évolution de l'activité.

3.4. La technologie utilisée pour la création de site web

Le site web de l'UKMO a été créé par le CMS (Content Management System) **Joomla**!, qui estbasée sur le langage de programmation PHP, le JavaScript et le Framework Bootstrap.

3.4.1. Le CMS Joomla!:

Joomla!est un système de gestion de contenu (CMS - *content managment system*), qui permet de créer des sites internet de qualité professionnelle. De nombreux aspects, notamment sa facilité d'utilisation et l'extensibilité, ont fait de Joomla!le logiciel le plus populaire, voire le meilleur de tous. Joomla!est une solution open source et gratuite accessible à tout le monde.

3.5. L'hébergeur

Le site web est hébergé localement par le Centre des Systèmes et Réseaux Informatique et Télé-enseignement et Visioconférence situé au niveau du rectorat de l'université sur un serveur de type LAMP (Linux, Apache, PHP, MySQL):

- **L**inux: le Systèmes d'exploitation GNU Linux CentOS 7.
- **A**pache: le Serveurs Web: Apache 2.4
- **P**HP: le langage de développement PHP5.6
- **M**ySQL : la Bases de données : MYSQL 5.x / MariaDB

3.6. Les dispositifs de sécurité du site web de l'UKMO

Il se trouve qu'il n'y a pas de manuel de procédures, précisant des mesures à prendre en matière de sécurité informatique dans son ensemble, et de sécurité du site web de l'établissement en particulier. Dans ce sens, la description du dispositif de sécurité du site web de l'université est basée sur les différentes observations effectuées au cours de notre travail et les différentes recommandations qui se trouvent dans le référentiel de sécurité Informatique dans sa version 2016 du Ministère de l'Enseignement Supérieur et de la Recherche Scientifique MESRES.

3.7. La gestion et l'évaluation des risques

Nous avons constaté qu'au niveau de l'université, qu'il n'y a pas un processus de gestion et d'évaluation des risques informatiques. Cet état de chose expose l'établissement à des risques tels que :

- Réponse aux risques non efficaces ;
- confiance excessive dans les contrôles techniques existants ;

- non détection de l'impact d'un risque sur l'établissement.

3.7.1. La sécurité du système

Il s'agit pour nous d'identifier les mesures de sécuritéappliquées pour garantir la sécurité du site web de l'UKMO.

3.7.2. La gestion des identités et des comptes administrateurs

Grâce à l'ACL (Access Contrôle Liste) de Joomla les administrateurs systèmes et réseaux ainsi que le webmaster del'université définissent les mises à jour des informations accessibles sur le site. Il octroie des droits en fonction des responsabilités de chaque utilisateur. Un login et un mot de passe sont exigés avant l'accès à l'administration du site.

3.7.3. Prévention, détection neutralisation d'attaques

Il n'existe pas un poste unique dédié à l'administration du site web. Mais le webmaster exige aux administrateurs de site web d'utiliser un poste protégé contre les attaques grâce à l'antivirus KAPERSKY avec une licence valide et mis à jour.

En utilisant le l'outil PuTTY qui permet l'accédé au serveur d'une manière sécurisé grâce au protocole SSH en élimant pas mal de tentatives de sniffer les données transmissent entre le poste utilisé dans l'administration de site et le serveur de site.

3.7.4. La sauvegarde et l'archivage des données

Toutes les tentatives de connexion à l'interface d'administration, de connexion au site web, sont journalisés. Des droits d'accès sont attribués aux fichiers pour des mesures de sécurité.
Des sauvegarde sont faite régulièrement dans le serveur (dossier web et la base de donnée) ainsi qu'une sauvegarde globale de la machine virtuelle qui contient le serveur.

3.7.5. La gestion du code source

Les administrateurs systèmes suivre régulièrement le site officiel de Joomla ainsi que les sites dédiés à la sécurité pour voire les différentes failles et vulnérabilités ; donc des mises à jour sont faite régulièrement dès qu'une nouvelle version arrive.
Le code source est régulièrement parcouru à la recherche d'anomalies et est régulièrement mis à niveau.

3.8. Défaillance de sécurité rencontrée dans l'UKMO

Selon les recommandations de l'ISO 2700x on peut diffuser les problèmes de sécurité en trois catégories qui sont :

1. La Disponibilité (DoS/DDoS)
2. La Confidentialité (Bruteforce attaque, SQL Injection)
3. L'Intégrité

3.8.1. Problèmes de Disponibilité

Au cours de nos études on a arrivé à énumérer les problèmes suivants qui peuvent être rencontré dans le cas de l'ukmo

a. Réseaux
b. Saturation de serveur
c. Défaillance des Services (DoS)
d. Défaillance des Configurations et applications
e. Défaillances matérielles
f. Problème de coupure d'électricité
g. Désastre (incendie, …)

Selon chaque cas on va donner la solution adéquate pour remédier le problème.

a. Problème de réseau :

Pour résoudre les problèmes qui peuvent apparaitre dans le réseau on propose d'avoir une autre connexion de secoure d'un autre ISP (Fournisseur d'accès Internet); ce qui est le cas dans l'ukmo (CERISTE + Algérie Télécom).

Sachant que le serveur web est situé dans une zone DMZ grace au firwalle ASA.

b. Problème Déni de Service DoS:

L'outil de test des performances du serveur HTTP Apache

ApacheBench **ab**est un utilitaire qui vous permet de tester les performances de votre serveur HTTP Apache. Il a été conçu pour vous donner une idée du degré de performances de votre installation d'Apache. Il vous permet en particulier de déterminer le nombre de requêtes que votre installation d'Apache est capable de servir par seconde.

ab -n 100 -c 10 https://www.univ-ouargla.dz/

Ceci va envoyer 100 requêtes GET HTTP, avec jusqu'à 10 requêtes parallèles, à l'URL spécifiée, ici "https://www.univ-ouargla.dz/".

L'Apache contient un Anti DoS qui doit être activé dans le fichier de configuration httpd.conf comme suit :

Fichier httpd.con

KeepAlive On

MaxKeepAliveRequest 1024 remplaceer par 300

KeepAliveTimeout 5 remplaceer par 2s

c. Problème Défaillance des Configurations et applications et Défaillances matérielles :

Un clusterà deux nœuds au plus sera la solution optimale dans ces cas.

Un cluster serveur c'est un groupe de systèmes informatiques indépendant, appelés nœuds qui exécutent un système d'exploitation et fonctionnent ensemble comme s'il s'agissait d'un seul système pour garantir que des applications et des ressources restent disponible pour le client. Les clusters permettent aux utilisateurs et aux administrateurs d'accéder aux nœuds et de les gérer comme un seul système.

Un cluster à deux nœuds a pour objectif d'assurer la haute disponibilité et fonctionne selon le principe suivant :

Quand un des nœuds tombe en panne, un autre nœud du cluster prend le relais dans le but d'assurer la disponibilité des données.

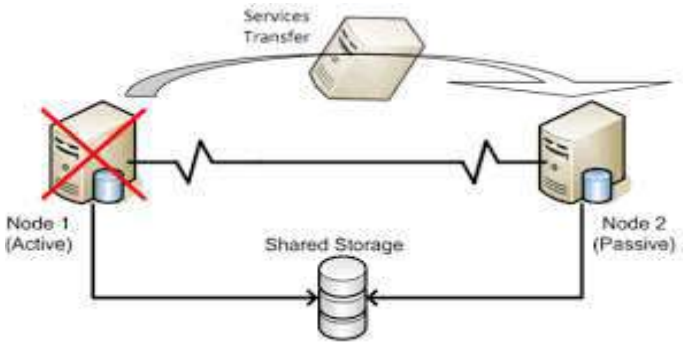

Figure 3.2 : cluster à deux nœuds

d. Ondulation :

On recommande d'avoir une ondulation de 2Heures minimum de marque connue comme APC par exemple que ce soit pour les serveurs ou les PCs administrations.

e. Désastre :

Pour éviter un éventuel désastre tel un incendie à titre d'exemple l'université doit avoir un autre site de secours sur un autre emplacement géographique différent soit dans le même siège de l'universités ou bien ailleurs.

Dans ce cas deux figures se présentent

- Réplication ou Basculement synchrone (il peut engendrer un trafic énorme dans le réseau)
- Réplication ou Basculement asynchrone

Réplication

La réplication repose sur un ensemble de technologies qui permettent de copier et de distribuer des données et des objets de base de données d'une base de données vers une autre, puis de synchroniser ces bases de données afin de préserver leur cohérence. Avec la réplication, vous pouvez distribuer des données en différents emplacements et à des utilisateurs distants ou mobiles sur des réseaux locaux et étendus, des connexions d'accès à distance, des connexions sans fil, et Interne.

Réplication Synchrone

Définition

- garantit que, lorsqu'une transaction met à jour une copie primaire (ex: EMP), toutes ses copies secondaires (ex: Emp) sont mises à jour dans la même transaction.
- implémenté par un protocole two-phase-commit (2PC).

Critique

- les copies primaires et secondaires sont toujour égales entre elles.
- peut bloquer en cas de panne de site.
- pas performant si beaucoup de copies secondaires.

Réplication Asynchrone

Définition

- lorsqu' une transaction met à jour une copie primaire, elle est validée sur le site master puis, chaque copie secondaire est mise à jour dans une transaction séparée.

Critique

- les copies primaires et secondaires peuvent ne pas être égales entre elles.
- plus performant que 2PC à mesure que le nombre de sites augmente.
- ne bloque pas en cas de panne de site.
- largement utilisé en pratique.

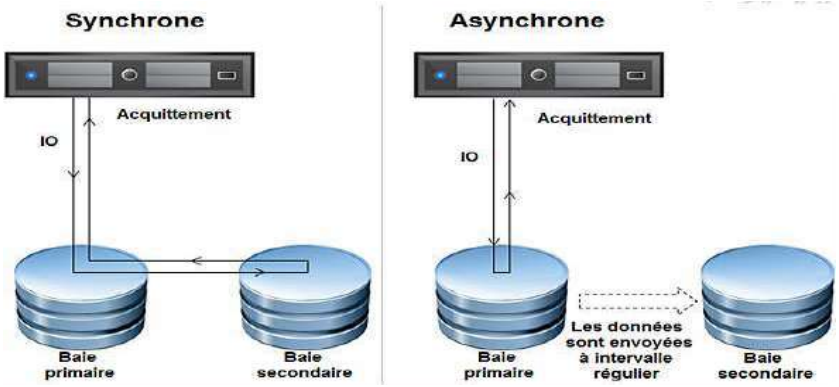

Figure 3.3 : Réplication synchrone et asynchrone

Pour éviter le problème de volume important on peut faire les choses suivantes :

1. **La compression** : il est courant de réduire la taille des données en exploitant la puissance des processeurs plutôt qu'en augmentant les capacités de stockage et de transmission des données.
2. **Différentielle** : Une sauvegarde différentielle est basée sur la sauvegarde complète des données précédente la plus récente. Une sauvegarde différentielle enregistre uniquement les modifications effectuées depuis la toute dernière sauvegarde complète. La sauvegarde complète sur laquelle une sauvegarde différentielle est basée s'appelle la *base* de la différentielle.
3. **Connexion dédié pour la réplication** : permet de configurer les paramètres du serveur de réplication : adresse du serveur, nom de l'utilisateur et mot de passe.

3.8.2. Problèmes de confidentialité

Les problèmes de confidentialités rencontrés dans le cas de site de l'ukmo sont :

1. Brute force attaque : Contourner l'authentification (récupération des mots de passe)
2. Récupération des fichiers
3. Injection des contenus

Les solutions qu'on propose sont comme suit :

- Activer dans **Joomla**! Le module anti brute force, et le captcha
- Activer l'anti brute force dans un LDAP
- Eviter les mots de passes simples
- Activer le mode SSL

Procédure d'obtention d'un certificat SSL

1. Générer une signature (clé publique + nom)

2. Envoyer à un CA (Certificat Autority)
3. Installer le certificat
4. Configurer la redirection http/https

Pour obtenir un certificat ssl l'établissement Il y a deux choix de certificat SSL

a. Pour un seul URL

b. Wildcard certificat *.univ-ouargla.dz: Wildcard SSL va permettre à un nombre illimité de sous-domaines d'être sécurisés, et ce avec un seul et unique certificat. C'est la solution idéale pour toute personne qui héberge ou gère plusieurs sites ou pages qui existent sur le même domaine. Le coût unique du certificat va inclure tous les sous-domaines ou serveurs que vous ajouterez plus tard.

- La solution proposée pour résoudre le problème de l'injection SQL est le suivant
- Installation d'un reverse proxy : Apache plusMod_Security
- Les mots de passes doivent être stockés chiffrés

Un proxy classique se place entre un client et les serveurs auxquels il peut accéder. Il transfère les requêtes HTTP du client aux serveurs externes et lui renvoie les réponses correspondantes.

Un proxy inversé se place entre un serveur et tous ses clients. Plus généralement, on va utiliser ce type de proxy afin d'obtenir un seul point d'entrée vers un ou plusieurs serveurs de manière transparente pour l'utilisateur. Le reverse proxy récupère les requêtes HTTP des clients et se charge de les transmettre aux serveurs internes designés par les requêtes correspondantes. Il existe de nombreuses solutions logicielles qui implémentent ce mécanisme (Apache, Squid, Pound, Nginx etc...).

Un proxy inversé sécurisé notre objectif est d'obtenir un reverse proxy basé sur Apache qui accepte uniquement des requêtes HTTPS en entrée et qui les redirige en clair vers un serveur. Les requêtes une fois déchiffrées doivent être analysées et filtrées par un pare-feu applicatif afin de limiter les risques d'attaques pouvant viser le serveur. Il s'agit d'un module Apache appelé mod_security qui permet de réaliser ce genre d'opérations de filtrage des requêtes HTTP.

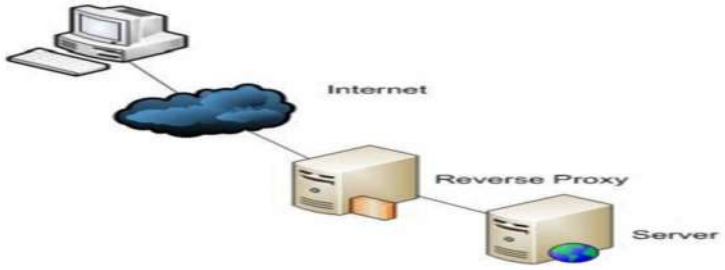

Figure 3.4 : Reverse proxy

3.8.3. Fonctionnement d'Apache appliqué à notre cas

La mise en place d'un reverse proxy avec Apache a besoin du module proxy qui est disponible nativement. Cependant, dans notre cadre qui nécessite une certaine sécurité nous devons utiliser des modules supplémentaires comme le module ssl (natif) ainsi que le module security. On peut éventuellement ajouter le module evasive afin de pouvoir limiter dans une certaine mesure les attaques de type DDOS (Distributed Denial of Service).

mod_evasive : Ce module de sécurité pour Apache a pour objectif de détecter les accès massifs à certaines ressources qui sont révélateurs d'attaques de type DDOS et de les stopper (dans une certaine mesure) en bloquant temporairement les IP néfastes. Il peut être intéressant d'installer ce type d'outil sur un reverse proxy afin de limiter les risques de défaillances en cas de DDOS de faible ou moyenne envergure.

mod_ssl : Il s'agit du module dédié aux échanges chiffrés grâce aux protocoles SSL/TLS. Dans notre configuration, il va s'occuper de vérifier si le client utilise bien une connexion sécurisée. Si ce n'est pas le cas, le client qui utilise une requête non sécurisée se verra éconduit.

mod_security: Ce module est un pare-feu applicatif permettant de filtrer les flux HTTPS de manière très précise parce qu'il a accès aux données de la transaction avant le chiffrement et après le déchiffrement. Il agit un peu à la manière d'un IDS (Intrusion Detection System) mais en analysant le traffic réseau au niveau HTTP.

Lorsqu'il est correctement configuré, il permet d'empêcher de manière très efficace un nombre impressionnant d'attaques (injections sql, tentatives d'attaques type xss etc..).

mod_proxy : Son rôle est de récupérer les flux SSL/TLS qu'il reçoit et de les renvoyer sous forme de requêtes HTTP simples vers le/les serveurs protégés derrière le proxy.

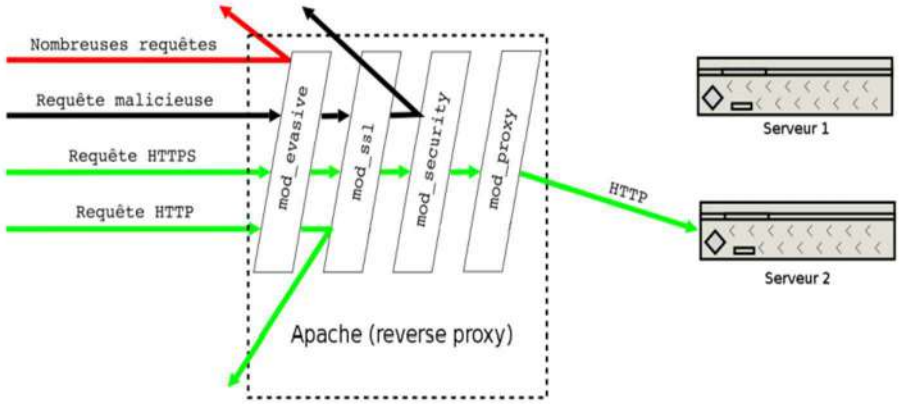

Figure 3.5 : Fonction Apache Reverse proxy

3.9. Quelles bonnes pratiques pour mettre en œuvre un site Joomla Sécurisé

3.9.1. Conseils liée à l'application :

En tant qu'application Open Source, Joomla est sujet à toute sorte d'attaques pirates. C'est pourquoi il est important de prendre le plus de mesure possible pour sécuriser votre site Joomla. En suivant les quelques conseils ci-dessous, vous minimiserez les chances d'être piraté.

Cette liste n'est bien sûr pas exhaustive et des attaques peuvent toujours survenir.

1. Garder Joomla et ses extensions à jour :

La partie la plus importante de la sécurisation de votre site Joomla est**la maintenance**, il vous faut le tenir à jour régulièrement. Dans presque toutes les mises à jour (nouvelles versions), des failles de sécurité sont comblées.

Gardez également vos extensions Joomla à jour, on dit toujours que **le niveau de sécurité d'un Joomla est équivalent à l'extension la plus vulnérable**(le maillon faible). Notez qu'il y a de plus en plus d'attaques qui utilisent les failles de sécurités des extensions Joomla..

2. Utiliser des identifiants de connexion complexes :

N'utilisez pas des noms d'utilisateur du type "admin" ou "administrator". Ceux-ci sont les premiers identifiants ciblés par les hackers.

Il est important d'avoir **un mot de passe fort et complexe** pour votre site web. De nombreux attaquants tentent une attaque frontale (brute-force) pour récupérer votre mot de passe. Ils utilisent en fait une liste de mot de passe couramment utilisé afin de trouver le vôtre. Il y existe plusieurs astuces qui vous aideront à vous protéger contre de telles attaques :

- Ne pas utiliser des mots en guise de mot de passe comme : love, dieu, pass, admin, admin123…
- Évitez les renseignements personnels dans les mots de passe comme votre nom, prénom…
- Utiliser des caractères spéciaux (*@{#), des chiffres et des lettres majuscules

3.9.2. Une bonne utilisation des droits et permissions de fichiers

Les droits et permissions sur vos fichiers et dossiers Joomla jouent un rôle important.

- Il est recommandé d'utiliser ceux-ci pour vos sites :
  - Définissez une permission 755 pour vos dossiers Joomla
  - Définissez une permission 644 pour vos fichiers Joomla
  - Définissez une permission 444 pour votre fichier configuration.php.
- Utiliser les extensions de sécurité de Joomla

L'utilisation d'extension est un moyen facile d'améliorer la sécurité de votre Joomla. Vous trouverez ci-dessous une liste de meilleures extensions Joomla de sécurité :

- jHackGuard
- AeSecure
- Admin Tools
- jomDefender
- jSecure

- Protéger l'accès à l'administration

Vous pouvez améliorer considérablement la sécurité de votre site Joomla si vous limitez l'accès à la page de connexion Admin. Tout d'abord, vous pouvez empêcher l'accès au répertoire **/administrator** grâce à AdminExil. Ensuite, vous pouvez restreindre l'accès à ce répertoire via un fichier nommé « .htaccess », à mettre dans le répertoire **/administrator**, avec les lignes suivante à la fin du fichier :

```
Deny                                from                                ALL
Allow from x.x.x.x
```

Notez que vous devez remplacer **x.x.x.x** avec votre adresse IP publique réelle. Pour connaitre votre adresse, vous pouvez vous rendre sur What Is My IP par exemple. Pour ajouter

plusieurs IPs, il faut simplement dupliquer à la ligne **Allowfromx.x.x.x.** Faites bien sûr attention à ne pas avoir une IP dynamique, qui change par exemple toutes les 24 heures.

- Scannez votre site

Parce qu'il vaut mieux prévenir que guérir, il existe de nombreux outils en ligne (souvent gratuits) qui permettent de détecter si votre site souffre d'une infection virale et/ou d'une faille de sécurité.
Si vous avez le moindre doute, procédez sans délai à un examen de votre site Joomla.
Quelle que liens utiles :

- SUCURI
- Web Inspector
- Quttera

- Utilisez l'authentification en deux étapes

Depuis Joomla! 3.2, vous avez la possibilité de renforcer significativement la sécurité du panneau d'administration grâce à deux nouveaux plugins natifs. Dans les deux cas, la protection peut s'appliquer sur le backend, sur le frontend ou sur les deux. Il convient également de modifier le profil de l'utilisateur pour cela.

- **YubiKey** : ce plugin permet aux utilisateurs de votre site d'utiliser l'authentification en deux étapes en se servant d'une clé USB de sécurité YubiKey. Les utilisateurs doivent d'abord se procurer leur propre YubiKey sur le site http://www.yubico.com/.
- **Google Authenticator** : ce plugin vous permet d'utiliser l'authentification en deux étapes en vous servant de Google Authenticator qui est un générateur de mots de passe à usage unique basé sur l'heure.

- Passez au .HTACCESS

Afin d'ajouter une couche supplémentaire de sécurité, vous pouvez utiliser le fichier .htaccess afin de protéger certains répertoires sensibles par mot de passe. Par contre, il faut savoir que la protection par mot de passe .htaccess seule est pas une méthode parfaitement sécurisée. Elle doit être utilisée sur un serveur SSL pour atteindre une protection maximale. Un serveur SSL est requis pour protéger votre site contre les attaques plus sophistiquées.

3.10. Les recommandations proposées pour le site de l'université UKMO

- Changer l'url d'administration
- Changer le login d'administrateur avec un mot de passe compliqué
- Installer un module Anti Brute Force
- Programmer un script de backup automatique
- Configurer le module mod-security avec apache pour countourner le SQL Injection
- Chiffrer les mots de passes stockés dans la base de données joomla
- Suivez les sites dédiés aux problèmes de sécurité et notamment les vulnérabilités de Joomla !

3.11. Résultats de l'audit de la sécurité du site web de UKMO

Dans ce chapitre, nous présenterons le déroulement de nos travaux d'audit, nous ferons une synthèse des forces et faiblesses constatées et formulerons des recommandations afin de corriger les faiblesses et de renforcer les dispositifs existants.

En Informatique le terme « Audit » apparu dans les années 70 a été utilisé de manière relativement aléatoire. Nous considérons par la suite un « audit de sécurité informatique » comme une mission d'évaluation de conformité sécurité par rapport àun ensemble de règles de sécurité.

3.11.1. Définition d'audit de sécurité

Un audit de sécurité consiste à s'appuyer sur un tiers de confiance afin de valider lesmoyens de protection mis en œuvre.

Un audit de sécurité permet de s'assurer que l'ensemble des dispositions prises par l'établissement sont réputées sures.

3.11.2. La phase de planification de la mission d'audit

Cette phase a commencé par une prise de connaissance générale de l'entité et particulièrement de l'élément en étude, en l'occurrence le site web.

Elle nous a permis d'effectuer la présentation de l'entité, de décrire le site web objet de notre étude, et de présenter les mesures de sécurisation du site web déployées par l'entité.

3.11.3. Objectifs de l'audit du site web d'UKMO

Dans le cadre de notre étude, l'objectif général identifié est de : S'assurer que les mesures de sécurité existantes protègent efficacement le site web pour lui procurer disponibilité, confidentialité et intégrité.

- s'assurer que des processus et procédures sont mis en place pour garantir la sécurité de l'information.
- S'assurer de l'existence d'un dispositif d'évaluation des risques et de son fonctionnement adéquat.
- S'assurer que tous les utilisateurs (internes, externes et temporaires) et de leurs activités sur les systèmes informatiques qui ont un lien direct avec le site web (applications d'entreprise, de l'environnement informatique. etc.) fait l'objet d'une traçabilité.
- S'assurer que seules les personnes autorisées accèdent aux informations sensibles.
- S'assurer que les politiques et procédures sont en place pour organiser la production, le changement, la révocation, la destruction, la distribution, la certification, le stockage, la saisie, l'utilisation et l'archivage des clés cryptographiques pour assurer la protection des clés contre la modification et la divulgation non autorisée.
- S'assurer que les mesures de prévention, de détection et de correction des menaces et vulnérabilités sont en place à travers l'organisation de la protection des SI.
- S'assurer de la mise en œuvre des procédures de sécurisation des réseaux afin de contrôler les flux en provenance de réseaux.

3.11.4. Audit de site de l'Université Kasdi Merbah Ouargla

Il existe plusieurs sites qui permettent de faire l'audit d'un site web online parmi lesquels sucuri, detectify, quttera, Acunetix,…etc. Certains sites sont gratuites d'autres sont payantes.

Pour simplifier notre étude on a choisie de faire le test sur deux sites différents sur le site Sucuri qui permet d faire un audit gratuitement et sur le site Detectify qui permet de faire l'audit pour une période de test.

1. Sucuri

SUCURI est le logiciel malveillant et le scanner de sécurité du site Web le plus populaire. Il permet d'effectuer un test rapide pour les logiciels malveillants, les listes noires du site Web, les SPAM injectés et les défauts. SUCURI nettoie et protège le site web des menaces en ligne

et fonctionne sur toutes les plateformes de site Web, y compris WordPress, Joomla, Magento, Drupal, phpPP, etc.

La figure suivante montre le scan fait sur le domaine www.univ-ouargla.dz; on remarque qu'il n y a pas des risques sauf que la notification Website Firewall Not Found (Par-feu de site web non trouvé).

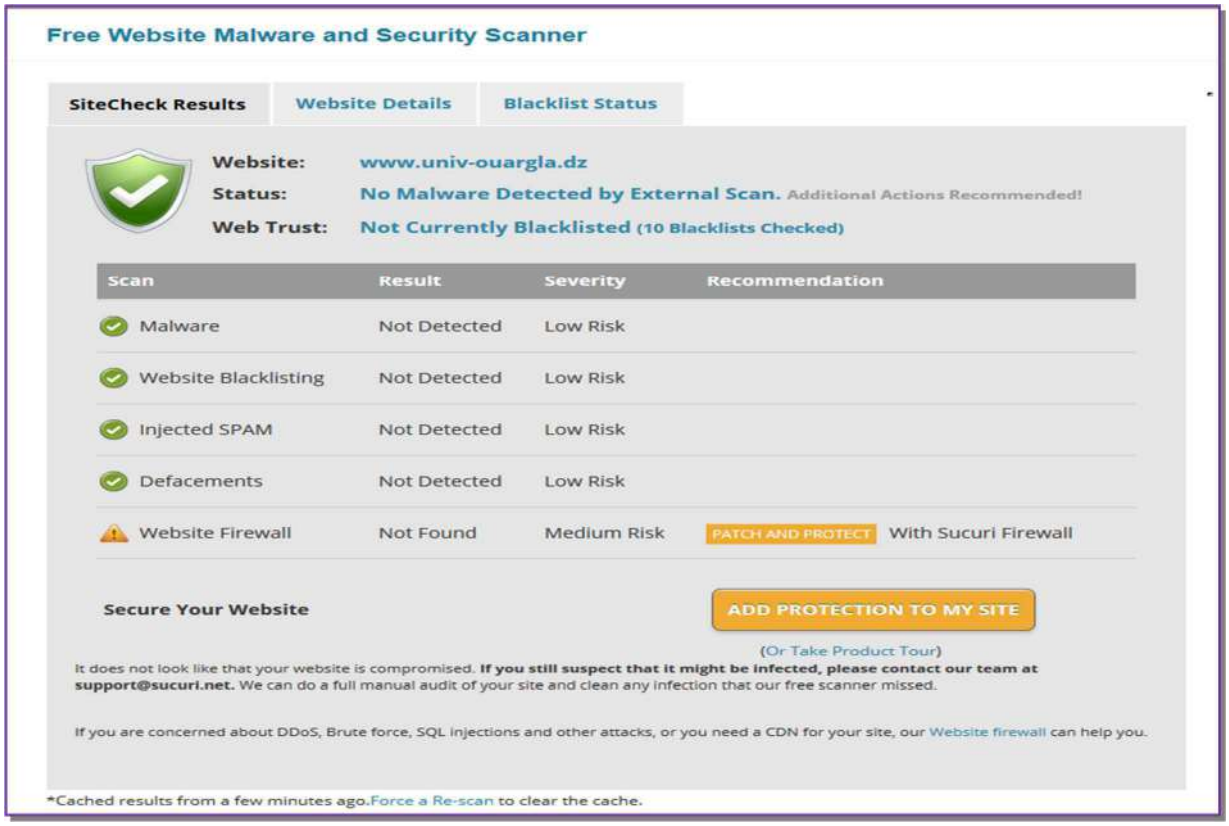

Figure 3.6 : Résultat de scan du site par SUCURI

2. Pare-feu de site web WAF

Le pare-feu d'application Web (WAF) est l'un des meilleurs moyens de protéger un site web contre les menaces en ligne.

Les solutions commerciales WAF peuvent être coûteux et si on cherche une solution gratuite pour protéger notre site Web en utilisant WAF, le Firewall d'application Web open source suivant peut être utile.

La liste suivante donne des exemples de WAF opensource

- ModSecurity
- IronBee
- NAXSI

- WebKnight
- Shadow Daemon

La solution simple et optimale pour le cas de l'université est d'utiliser le modsecurity qu'on va présenter dans ce qui suit :

ModSecurity par TrustWave est l'un des pare-feu d'applications Web les plus populaires et il prend en charge Apache HTTP, Microsoft IIS &Nginx.

Les règles gratuites de ModSecurity vous seront utiles si vous recherchez la protection suivante.

- Cross-site scripting
- Trojan
- Information leakage
- SQL injection
- Common web attacks
- Malicious activity

L'installation de mod_security dans LunixCentOS est comme suit :
 # yum install mod_security
 # /etc/init.d/httpd restart

3.12. Detectify

Detectify est un scanner de sécurité de site Web basé sur le **SaaS**. Cela a obtenu plus de 100 tests de sécurité automatisés, y compris **OWASP Top 10**, logiciels malveillants et bien plus encore.

Après l'inscription dans le site pour une période de test de 14 jours et la validation de domaine www.univ-ouargla.dz. On a fait un scanne total de site et le score total est illustré dans la figure ci-dessous

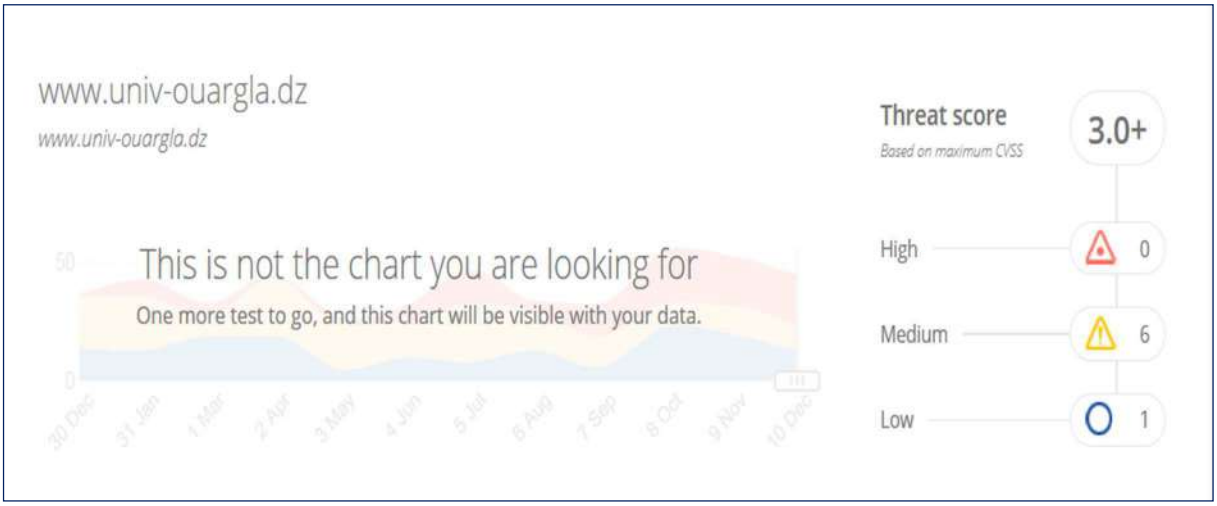

Figure 3.7 : Threat score

Le score du domaine www.univ-ouargla.dz est de 3.0+ ce qui est acceptable ; On remarque qu'il n y a pas des failles marquées comme failles majeurs dans le site; 06 failles de type moyenne et une seule failles de type mineur.

3.12.1. Les failles de type moyennes

1. Cross Site Request Forgery (CSRF/XSRF)
2. Side Channel AuthenticationTokenLeakage
3. PHP Easter Egg
4. Lacking Transport Security
5. Cross-Site Tracing (XST)
6. Email Spoofing / Missing SPF Record

3.12.2. Les failles de type mineurs

1. Technology Disclosure

Sachant que le site detectify permet d'exporter le résultat dans un fichier **pdf** dans notre cas le fichier contient plus de 500 pages pour résumer les résultats de l'audit on va expliquer seulement le OWSAP TOP 10.

3.12.3.OWSAP TOP 10 Score

Detectify permet de filtrer les résultats de l'audit. Les figures ci-dessous montrent l'audit vis-à-vis les OWSAP TOP 10.

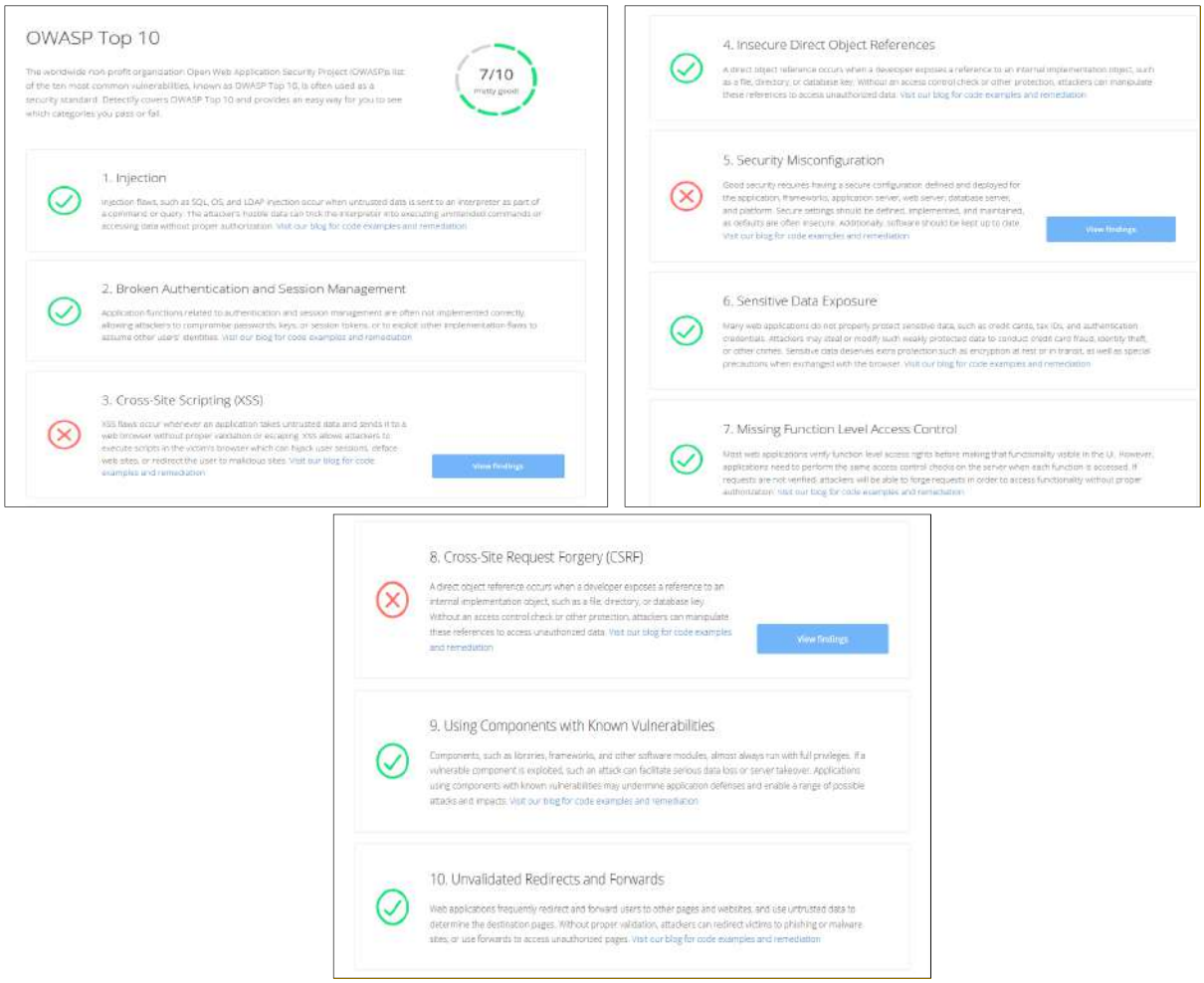

Figure 3.8 : OWSAP TOP 10 Score de L'UKMO par Detectify

3.12.4. Description des résultats

Le score de site de l'université Kasdi Merbah Ouargla est de **7/10**; un travail supplémentaire est nécessaire pour corriger les failles de **XSS,** le **CSRF** et le **Security Misconfiguration.**

Pour chaque faille on va essayer de donner la cause, l'explication, les problèmes et on propose la solution adéquate.

a- La faille XSS (Cross-Site Scripting)

Cause:

Un attaquant peut l'utiliser pour voler des cookies (par exemple, des cookies de session) de la victime qui, à son tour, peut entraîner des sessions d'utilisateurs détournées

Explication:

Il est possible d'envoyer une requête HTTP TRACE et le serveur répondra avec toutes les données envoyées avec la demande.

Commande

curl -XTRACE -H "TRACE-XSS: Vulnerable" "http://www.univ-ouargla.dz/"

- **Le XSS réfléchi (non permanent)**

Cette faille est la plus simple des deux. Elle est appelée non permanente car elle n'est pas enregistrée dans un fichier ou dans une base de données.

- **Le XSS stocké (permanent)**

La faille permanente est la faille XSS la plus sérieuse car le script est sauvegardé dans un fichier ou une base de données. Il sera donc affiché à chaque ouverture du site.

Comment s'en protéger ?

La solution la plus adaptée contre cette faille est d'utiliser la fonction **htmlspecialchars()**. Cette fonction permet de filtrer les symboles du type <, & ou encore ", en les remplaçant par leur équivalent en HTML. Par exemple :

- Le symbole&devient&
- Le symbole " devient"
- Le symbole ' devient'

b- La faille CSRF (Cross site requestforgery)

Explication

La faille CSRF ("Cross Site RequestForgery") est très souvent assimilée à la XSS alors que ces deux failles sont diamétralement opposées. Quand la XSS cherche à dérober des informations personnelles de l'utilisateur, la CSRF cherche à lui faire exécuter des actions à son insu directement sur son ordinateur. Dans la section précédente, on a vu comment le pirate utilisait la XSS pour voler les cookies de l'administrateur, pour pouvoir prendre le contrôle du site. Si le hacker décidait d'utiliser la CSRF, il ferait exécuter l'action directement sur l'ordinateur de la victime.

D'après l'audit qu'on a fait deux failles sont trouvées

Une faille Cross Site RequestForgery (CSRF/XSRF) trouvé sur le fichier:

http://www.univ-ouargla.dz/media/k2/assets/js/elfinder.min.js

Ce fichier de type javascript est utilisé par l'extension K2 installé sur la version Joomla! En cours sur le site de l'université.

Pour éviter un éventuel attaque une mise a jours de l'extension k2 est indispensable.

Comment s'en protéger ?

Authentification pat jeton (token):

Il n'existe malheureusement pas de protection parfaite contre la CSRF. La façon la plus rependue étant l'utilisation d'un jeton unique qui sera vérifié à chaque modification. Beaucoup d'internautes préconisent d'utiliser la fonction uniqid(), mais c'est en fait une erreur. Elle génère un identifiant unique basé sur le temps en microsecondes, et contrairement à ce que l'on pourrait penser, sa valeur n'est pas impossible à deviner. Il est donc, à mon gout, plus sécurisé d'utiliser ceci :

<?php$token=bin2hex(mcrypt_create_iv(32, MCRYPT_DEV_URANDOM)); ?>

D'après la documentation PHP :

mcrypt_create_iv() crée un IV (vecteur d'initialisation) à partir d'une source aléatoire.

Le vecteur d'initialisation est le seul moyen de fournir une initialisation de remplacement aux méthodes d'initialisation. Ce vecteur n'a pas besoin d'être particulièrement secret, même si c'est mieux. Vous pouvez l'envoyer avec vos documents chiffrés sans perdre en sécurité.

c- Security Misconfiguration (Moyenne)

Cause Email Spoofing / Missing SPF Record

Ce problème est lié principalement au mail de l'université

Explication

Le domaine www.univ-ouargla.dz manque d'un enregistrement de politique DNS SPF. Les politiques SPF doivent être appliquées sur tous les domaines (y compris les sous-domaines) ayant un enregistrement A, AAAA ou MX.

Que peut-il arriver ?

Un attaquant sera en mesure de falsifier les courriels provenant du domaine, permettant des attaques de phishing ou d'autres escroqueries.

Comment s'en protéger ?

La solution de ce problème consiste à ajouter une entrée SPF dans le DNS qui gère le domaine univ-ouargla.dz.

3.13. Conclusion

En conclusion, nous avons vu que le nombre de types d'attaques qu'il est possible de mener contre un site Web est considérable. Il est donc extrêmement important de prendre en compte la sécurité le plus en amont possible lors du développement d'une application Web. L'idéal est de considérer les contraintes sécuritaires. Les recommandations exposées au cours de cette présentation permettront alors de bloquer la plupart des attaques. Cependant, il reste indispensable de faire procéder à un test d'audit applicatif par une établissement spécialisée.

Nous avons passés en revue sur un certain nombre d'éléments pour avoir un aperçu sur la gestion de la sécurité informatique globale à l'**UKMO** en en particulier sur la gestion de la sécurité du site web en étude.

Conclusion Générale

Conclusion Générale

Au terme de notre étude nous pouvons affirmer que la majorité des établissements intègrent les technologies d'information et tous particulièrement leurs sites web dans la gestion courante de leurs activités afin d'être plus performantes.

L'usage des sites web, bien que bénéfique expose les organisations à de nombreux risques inhérents, auxquels l'administrateur doit faire face. L'audit est l'un des outils les plus fiables dont disposent les dirigeants pour réduire ces risques afin d'atteindre une assurance raisonnable.

Ces raisons nous ont poussés à retenir comme thème de notre étude «sécurité des sites web cas du site web de l'université «Kasdi Merbah Ouargla» afin d'améliorer la sécurité en assurant la confidentialité, l'intégrité et la disponibilité des informations qu'il contient et des technologies qui le sous-tendent.

Ce thème a été traité en deux grandes parties :

- La première dite théorique concerne les aspects théoriques de sécurité des site web qui comprend deux (02) chapitres à savoir : le site web, un composant des systèmes d'information, ensuite les concepts liées au sécurité, les vulnérabilités et les type d'attaque des sites web et enfin la méthodologie de l'étude, et les recommandations et les mesures des sécurités pour développer un site web bien sécurisé.
- La seconde partie dite pratique a abordé l'aspect pratique qui se compose de dernier chapitre : la présentation de l'Univérsité Kasdi Merbah Ouargla, la description du site web et des bonnes pratiques liées et enfin la présentation des résultats de l'audit.

Ainsi au cours de notre étude, nous avons mené des investigations grâces aux méthodes de l'audit des systèmes d'information afin de porter un jugement sur le niveau de sécurité du site web sujet de notre étude.

L'aspect de la sécurité physique étant majoritairement du ressort de l'hébergeur, et étant donné que nous avons restreint le champ d'action de notre mission à l'UnivérsitéKasdiMerbah Ouargla seul, nous n'avons pas suffisamment insisté sur ce paramètre.

A la fin de ce travail, nous pouvons dire que nous avons acquis une visibilitéconcrète sur un domaine bien spécifique qui est la sécurité informatique.

En plus, ce travail nous a été profitable en terme d'acquérir une bonne expérience professionnelle, à travers laquelle nous avons eu l'occasion d'appliquer nos connaissances scientifiques et de confronter la notion théorique à la pratique.

En fin on doit signaler que tous les composants de système nécessitent une succession des études de sécurités pour chaque composant.

Bibliographie

Références Bibliographiques

[1] Jean-Luc Archimbaud, Robert Longeon, Guide de la sécurité des systèmes d'information a l'usage des directeurs (de laboratoires de recherche), 1 Fév 2011.

[2] Kenneth Graf, LES DÉFIS DE LA SÉCURITÉ DES APPLICATIONS : DES SOLUTIONS , livre blanc d'IBM , 2007.

[3]Les principes de la sécurité Critères fondamentaux, Master 2 Professionnel Informatique, Université de REMIS CHAMPAGNE-ARDENNE

[4]RiadhAbdelli, Audit et Sécurité Informatique d'un Réseau LocalD'entreprise, Mémoire de License appliqué, entechnologie de l'information et communication, UNIVERSITE VIRTUELLE DE TUNIS, 2011.

[5] Géraldine Vache Marconato , Evaluation quantitative de la sécurité informatique : approche par les vulnérabilités, 10 Mar 2010.

[6] Patrick CHAMBET, Eric Larcher, Vulnérabilités et solutions de sécurisation

des applications Web.

[7] Guillaume HARRY, FAILLES DE SECURITE DES APPLICATIONS WEBPRINCIPES, PARADES ET BONNESPRATIQUES DE DEVELOPPEMENT, 03/04/2012.

[8] GUIDE DE SECURITE DES APPLICATIONS WEB, ROYAUME DU MAROC, 12/12/2014.

[9] AMOUZOUN Mériadec, Audit de la sécurité du site web de TALISMEAS, Sénégal : www.talismeas.com, Octobre 2013.

[10] Nicolas Mayer, Jean-Philippe Humbert, La gestion des risques pour les systèmes d'information.

[11] Note technique Recommandations pour la sécurisation des sites web, Agence nationale de la sécurité Nombre de pages du document des systèmes d'information, N° DAT-NT 009/ANSSI/SDE/NP, Paris, le 13 août 2013.

[12] LES DIX VULNERABILITES DE SECURITE APPLICATIVES WEB LES PLUS CRITIQUES, OWASP TOP 10, EDITION 2007

[13] Industrie des cartes de paiement (PCI) Norme de sécurité des données, Attestation de conformité Evaluations sur site – Prestataires de services, Version 3.2, Avril 2016.

[14]Rim Akrout, Analyse de vulnérabilités et évaluation de systèmes détection d'intrusions pour les applications Web, Informatique et langage [cs.CL]. INSA de Toulouse, 2012.

[15]*Michel Riguidel*,La sécurité des réseaux et des systèmes, ENST PARIS, 2007.

[16] Guillaume HARRY, Principales failles de sécurité des applications Web Principes, parades et bonnes pratiques de développement, Direction des systèmes d'information.

[17] Sécurité des applications web comment maitriser les risques lies à la sécurité des applications web ?, Club de la sécurité de l'information Français , Septembre 2009.

[18] LA SÉCURITÉ INFORMATIQUE, LESCOP Yves [V1.6]

[19]Magali Contensin, Sécurité des applications Web – PHP/MySQL, janvier 2009.

[20] Asma CHIKH, Amina DJENNANE, Sécurité d'une application Web à l'aide d'un système de détection d'intrusions comportementale, Mémoire de fin d'études en Informatique, Université Abou BakrBelkaid– Tlemcen**,** Juillet 2012.

[21]RichCannings, HimanshuDwivedi, Zane Lackey, Hacking sur le Web 2.0 Vulnérabilité du Web 2.0 et sécurisation, PEARSON, 2008.

[22]Référentiel de Sécurité Informatique, Ministère de la Poste et des Technologies de l'Information et de la communication, Juin 2016.

[23] ACISSI, Sécurité Informatique Ethical Hacking, Apprendre l'attaque pour mieux se défendre, Editions ENI, Février 2015.

Les liens hypertextes :

[24]Guillaume HARRY, FirePrawn, Failles de sécurité des applications Web. URL:https://web.developpez.com/tutoriels/web/failles-securite-application-web/ (consulté le 27/03/2017).

[25] Audit de sécurité d'un site internet, Groupe GTS, URL:https://www.groupe-gts.fr/conseil-audit-web/audit-securite-site-internet (consulté le 04/03/2017).

[26] Formation Sécurité des applications Web, URL:http://www.orsys.com/formation-securite-application-web.asp (consulté le 06/03/2017).

[27] La sécurité des sites Internet, URL:http://www.xp-internet.com/xp-infos/securite-sites-internet.php(consulté le 04/03/2017).

[28] Mouhamadou GAYE, Mise en place d'un site sécurisé, Université Cheikh Anta Diop de Dakar - Licence professionnelle 2010, URL:http://www.memoireonline.com/02/12/5265/Mise-en-place-d-un-site-securise.html (consulté le 04/03/2017).

[29] Rodrigue Mpyana, Mise en place d'un système de sécurité basé sur l'authentification dans un réseau IP, 2011, URLhttp://www.memoireonline.com/04/12/5679/m_Mise-en-place-dun-systeme-de-securite-base-sur-lauthentification-dans-un-reseau-IP-Cas-d0.html (consulté le 04/03/2017).