LES DOSSIERS TECHNIQUES

Sécurité des applications Web

Comment maîtriser les risques liés à la sécurité des applications Web ?

Septembre 2009

GT Sécurité des applications Web

CLUB DE LA SECURITE DE L'INFORMATION FRANÇAIS

30, rue Pierre Sémard, 75009 PARIS

Tél. : +33 1 53 25 08 80 – Fax : +33 1 53 25 08 88 – e-mail : clusif@clusif.asso.fr

Web : http://www.clusif.asso.fr

REMERCIEMENTS

Le CLUSIF tient à mettre ici à l'honneur les personnes qui ont rendu possible la réalisation de ce document, tout particulièrement :

Michel	**Frenkiel**	*Mobilegov*
Sébastien	**Gioria**	*Groupe Y*
Philippe	**Larue**	*CBP*
Vincent	**Maret**	*Ernst & Young*
Michel	**Maximin**	*Crédit Logement*
Enrick	**Moulin**	*BT CyberNetworks*
Ludovic	**Petit**	*SFR*
Philippe	**Sarafoglou**	*PSA*
Herve	**Schauer**	*HSC*
Arnaud	**Treps**	*Accor*
Patrick	**Vanhessche**	*ADP*

Nous remercions aussi les adhérents du CLUSIF ayant participé à la relecture.

SOMMAIRE

I - Introduction

Ce document s'adresse aux managers, décideurs et responsables métier qui sont en charge de la mise en place et/ou du maintien en condition opérationnelle d'un site Web Internet ou une application Web interne. Son but est de présenter les risques de sécurité propres aux technologies Web, et de décrire les bonnes pratiques en terme de gestion de projet informatique et de gestion de risques qui permettent, durant le cycle de développement et la vie de l'application, de maîtriser ces risques.

Ce document n'a pas pour but de décrire de façon détaillée les concepts, techniques ou outils de sécurisation. Ces éléments seront abordés par un autre groupe de travail du CLUSIF, et le document résultant complétera avantageusement le présent document.

II - Les technologies Web, incontournables, mais porteuses de nouveaux risques

II.1 - Des technologies omniprésentes

L'activité des entreprises et administrations repose de plus en plus sur les technologies et applications Web. Leur facilité de mise en œuvre et de déploiement les a rendues omniprésentes et incontournables, qu'il s'agisse de sites de commerce en ligne, d'applications Intranet ou Extranet, ou de services Internet offerts ou utilisés par les entreprises. Les nouvelles applications sont aujourd'hui presque systématiquement développées avec des technologies Web, et les anciennes applications sont souvent adaptées pour être accédées via un navigateur Web.

II.2 - Des environnements exposés

Des données critiques pour les entreprises et au regard de la loi sont désormais gérées par des applications Web. Or, des vols de données sensibles, des intrusions sur des sites Web ou des incidents liés à la disponibilité des applications Web font la une de l'actualité [1], [2], [3]. Des enquêtes menées par des cabinets d'experts indiquent qu'une grande majorité des sites Web sont vulnérables à des attaques. En outre, deux grandes tendances ont émergé ces dernières années sur le marché de la sécurité :

- L'agresseur n'attaque plus pour des motifs de prestige personnel, mais est mû par l'appât du gain, à dessein de fraude ;
- Les applications Web sont désormais la cible privilégiée des pirates / piratages. Le Gartner Group estime que 75% des attaques ciblent désormais les applications.

Des sondages réalisés aux États-Unis suggèrent que plus de 60 % des clients envisagent de ne plus traiter avec une entreprise en cas de menace sur leurs données personnelles suite à une attaque informatique. L'actualité a montré récemment l'impact que peut avoir l'indisponibilité d'un site de commerce en ligne dans la presse. On imagine donc aisément les éventuelles conséquences (financières, respect de la loi, impact sur l'image, etc.) si une application web n'est pas initialement conçue, développée, installée et exploitée de façon sécurisée.

II.3 - Des réglementations et des responsabilités

Les responsabilités des organisations face à la sécurité de l'information numérique sont croissantes : le législateur, les réglementations et les clients exigent de plus en plus de protection des données et de traçabilité de la part des entreprises.

Face à l'aggravation des violations de sécurité liées aux applications Web, à l'augmentation de leur nombre, et aux difficultés qu'elles posent, la rigueur des réglementations et standards industriels s'est renforcée. De nouveaux référentiels tels que par exemple le standard de sécurité des données de paiement PCI DSS intègrent désormais la sécurité des applications Web. Des réglementations récentes stipulent que les entreprises doivent assurer le développement et la maintenance de systèmes et d'applications sécurisés, et ciblent plus particulièrement les vulnérabilités des applications Web. Aux Etats-Unis par exemple, l'état de Californie a promulgué le « Notice of Security Breach Act 1 » (Code civil de Californie 1798.821), également connu sous le nom de Senate Bill 1386, qui impose une notification, aux partie concernées, mais aussi publique, en cas de compromission, avérée ou suspectée, des données confidentielles d'un client.

C'est pourquoi on ne peut de nos jours évoquer le sujet de la sécurité des applications web sans parler du cadre légal et règlementaire, tant pour une société fournisseur de services, une société éditrice de logiciels, un sous-traitant ou un développeur que pour l'entreprise utilisatrice ou le donneur d'ordre. En effet, la façon dont une application web est conçue et développée peut impacter la disponibilité, l'intégrité et/ou la confidentialité des données. Par voie de conséquence, la mise à disposition d'un service applicatif par une société peut engager la responsabilité [4] - civile et/ou pénale - [5] de cette société, tout comme celle de l'éditeur d'un logiciel ainsi que celle du développeur d'une application.

II.4 - Le rôle du management

La sécurité des applications Web ne peut donc plus être ignorée. C'est aujourd'hui une question de management, qui doit être traitée spécifiquement, au-delà des approches traditionnelles.

Face aux risques liés à des applications Web dont la complexité et l'omniprésence ne cessent de croître, les décideurs doivent initialiser les actions permettant d'assurer un niveau de sécurité suffisant pour les applications Web développées et/ou utilisées par leur entreprise. De même que pour la qualité, le management doit définir les objectifs liés à la sécurité, déléguer les tâches de réalisation et s'assurer de l'atteinte des objectifs. Négliger de telles mesures représenterait une prise de risque conséquente sur le marché actuel.

La sécurité des applications Web doit être prise en compte dès la conception, durant le développement, lors de l'intégration et durant toute la vie de l'application. Elle doit faire partie intégrante de l'application, et non être ajoutée à la fin du cycle de développement. Au-delà de la conception et de l'implémentation, la sécurité d'une application Web doit en outre être testée et vérifiée avant la mise en production et lors de ses évolutions.

Les aspects techniques, les outils, les méthodologies sont nécessaires pour sécuriser une application Web, mais il ne faut pas négliger l'aspect humain. Le management doit donc s'assurer que les différentes parties prenantes, MOA (Maîtrise d'Ouvrage), chef de projet, développeurs, comprennent les enjeux, connaissent et mettent en oeuvre les bonnes pratiques.

III - Sécurité des applications Web : mythes et réalités

Même si le management a globalement conscience des risques associés à la sécurité des applications Web, on constate encore aujourd'hui la persistance d'idées reçues qui sont autant d'obstacles à la compréhension complète des problématiques et entravent la prise des décisions adéquates pour assurer la sécurité d'un environnement Web. En effet, les arguments avancés par certains vendeurs de solutions de sécurité, la volonté des acteurs du web de rassurer leurs clients, ou les efforts de vulgarisation portant sur la sécurité de l'Internet ont contribué à nourrir plusieurs mythes néfastes.

III.1 - Mythe N°1 « Sans demande particulière, le développeur me fournira une solution sécurisée »

La sécurité d'une application web n'est pas innée. Au contraire, si des précautions particulières ne sont pas prises, il est fort probable que des vulnérabilités existent dans la solution mise en place et qu'elles puissent impacter la confidentialité, l'intégrité ou la disponibilité de l'application et des données traitées. Ce qui est vrai pour tout type d'application l'est encore plus pour une technologie à l'origine conçue pour permettre un accès totalement libre aux informations, et donc où les mécanismes d'authentification, de gestion de session et de contrôle d'accès ne sont pas natifs.

Le donneur d'ordre doit donc faire en sorte que les bonnes pratiques de sécurité soient comprises et appliquées par le maître d'œuvre, puis se doter des moyens de contrôler le niveau de sécurité de l'application (voir ci-après le chapitre sur la vérification de la sécurité).

III.2 - Mythe N°2 « Seuls des génies de l'informatique savent exploiter les failles des applications Web »

Au contraire, les failles de sécurité au sein des applications Web sont le plus souvent faciles à exploiter. Un attaquant n'a souvent besoin que d'un simple navigateur Web pour identifier et exploiter des failles de sécurité. Des outils complémentaires, comme des logiciels de proxy locaux, capables d'intercepter les requêtes entre le navigateur et le serveur Web sont disponibles gratuitement sur Internet.

Les technologies Web reposent en outre sur un ensemble de protocoles, langages, et architectures ouverts, dont les spécifications sont librement accessibles. N'importe qui peut donc étudier ces référentiels et les flux échangés entre l'application Web et le client, pour identifier des failles d'implémentation ou de conception.

III.3 - Mythe N°3 « Mon site Web est sécurisé puisqu'il est protégé par SSL »

La technologie SSL a été conçue pour éviter que des données sensibles, comme des données de paiement, soient transmises « en clair » sur Internet, et les sites commerciaux arborent aujourd'hui tous un logo « site sécurisé par un certificat SSL ». Lors de l'explosion du e-commerce, les medias ont expliqué qu'un internaute ne devait se considérer sur un site de confiance que si le fameux cadenas s'affichait dans la barre d'état du navigateur Web. La dérive vers le mythe du « Mon site est sécurisé puisque il est protégé par SSL » s'explique donc facilement.

Or, si SSL est un mécanisme de sécurité nécessaire pour protéger la confidentialité et l'intégrité des données échangées entre le client et le serveur, il n'est pas suffisant pour protéger totalement l'application Web, notamment des attaques contenues dans le flux Web envoyé par le client au serveur, qui passent donc dans le tuyau « SSL ».

III.4 - Mythe N°4 : « Je suis protégé contre les attaques, j'ai un firewall »

Historiquement, les entreprises ont commencé à se connecter à Internet, en mettant en place des composants de sécurité opérant au niveau du réseau, tels que des pare-feu réseau (firewall). S'ils sont bien configurés, ces mécanismes sont et restent efficaces dans la plupart des cas pour empêcher des accès non autorisés à l'infrastructure hébergeant les applications Web, comme les services réseau des systèmes d'exploitation ou des bases de données, et sont donc nécessaires. Malheureusement, ils sont insuffisants pour assurer la sécurité d'une application web, car les failles applicatives sont exploitées via des canaux de communication normaux et nécessaires au bon fonctionnement de l'application, et donc autorisés par le pare-feu.

III.5 - Mythe N°5 : « Une faille sur une application interne n'est pas importante »

Une application interne, non publiée sur Internet, est certes moins exposée qu'un site Internet, accessible 24h sur 24 à des centaines de millions d'internautes. Toutefois, si elle contient des failles, elle est vulnérable aux attaques menées par n'importe quelle personne, ayant accès au réseau interne, et disposant d'un simple navigateur Web.

En outre, l'utilisation de la technologie web pour les applications internes a fait du navigateur Web un outil unique utilisé à la fois pour accéder à des ressources Web externes et internes à l'entreprise. Il peut donc exister des situations où un utilisateur interne accède en même temps et à partir du même navigateur Web, à une application Web interne sensible et à un site Web externe, potentiellement sous le contrôle d'un attaquant. Or les technologies Web peuvent permettre à un site Web « malveillant » de faire réaliser par le navigateur d'un utilisateur qui accède à ce site des requêtes sur un autre site, même interne, et ce à l'insu de l'utilisateur. C'est donc un moyen potentiel pour un attaquant externe d'exploiter des failles sur une application Web intranet, depuis Internet, même si un pare-feu est en place, en utilisant le navigateur Web comme un pivot de rebond.

IV - Les principales failles de sécurité des applications Web

La majorité des failles de sécurité que l'on peut trouver dans les applications Web peut être classée en trois grandes familles :

IV.1 - La gestion incorrecte de l'authentification, des habilitations et du contrôle d'accès

Les paramètres d'authentification, d'habilitation et de contrôle d'accès soumis (par l'utilisateur, ou d'application à application) sont bien souvent incorrectement pris en compte, gérés ou contrôlés. Cette situation peut entraîner des risques d'usurpation d'identité et d'accès à des fonctionnalités ou données illégitimes, et donc d'atteinte à la confidentialité ou à l'intégrité des données.

Le risque est exacerbé par le fait que le fonctionnement par défaut d'un serveur Web est d'autoriser l'accès au contenu publié. Si un contenu spécifique doit être fonctionnellement restreint à certains utilisateurs seulement, le développeur doit mettre explicitement en place des mécanismes d'authentification, d'habilitation et de contrôle d'accès pour protéger chaque contenu (pages Web notamment) devant être protégé.

IV.2 - Les failles de type « injection »

L'injection de données est une technique consistant à insérer (i.e. Injecter) des données spécialement formées en entrée d'une fonction, d'un programme ou d'un script afin de les détourner de leur fonction d'origine (e.g. modification de base de données, récupération d'informations sensibles, etc.). Des failles de type injection peuvent par exemple exister pour les accès en base de données (SQL injection), aux annuaires LDAP (LDAP injection) ou la construction de pages Web dynamiques (Cross Site Scripting)

En injectant des données non prévues par l'application Web, un attaquant peut influer sur le fonctionnement même de l'application, voire compromettre un environnement applicatif complet. C'est donc tout autant la disponibilité, que l'intégrité et la confidentialité des données qui peuvent ne plus être assurées.

Les attaques de type Injection sont facilitées par la possibilité pour un attaquant d'utiliser des outils de type « proxy local » librement disponibles sur Internet. Ces outils lui permettent d'injecter dans les champs de formulaires ou entêtes HTTP des données spécifiquement composées pour permettre d'exploiter les failles, quelles que soient les protections mises en place côté navigateur Web.

IV.3 - Les fuites d'information

Si les fonctionnalités ou composants internes à une application ne sont pas suffisamment « cloisonnés », ou si l'application fournit des pointeurs de référence à des données non sécurisées, des fuites d'informations sensibles peuvent survenir (identifiants et comptes clients, types et versions de composants, requêtes SQL, informations de session, données de saisie dans les formulaires, cookies, voire données applicatives.). Cette situation peut entraîner un risque de perte de confidentialité. Elle peut également fournir à un attaquant des informations lui permettant de faciliter des attaques ultérieures (SQL injection par exemple).

Pour plus de précision sur les failles de sécurité des applications Web, le lecteur pourra se référer au Top Ten de l'OWASP [6].

V - Quelles bonnes pratiques pour mettre en œuvre une application Web sécurisée ?

Face aux risques liés à la sécurité des applications Web, il est primordial pour les organisations de mettre en œuvre les bonnes pratiques permettant d'obtenir des applications disposant d'un niveau de sécurité suffisant par rapport aux risques métier.

Ces bonnes pratiques doivent être mises en œuvre par les différents acteurs du projet, de manière complémentaire et cohérente. La sécurité doit être prise en compte de manière proactive et non réactive, tout au long du cycle de vie du projet, et non ajoutée et testée à la fin du cycle de développement. La prise en compte des aspects sécurité le plus en amont possible permet en outre d'optimiser les coûts et les délais de réalisation. En effet, plus la sécurité est prise en compte tardivement dans les étapes de développement, plus le coût de correction des failles est élevé.

Les bonnes pratiques de sécurité suivantes doivent être mises en œuvre lors des différentes étapes du cycle de développement :

V.1 - Formation et sensibilisation

Les failles dans les applications web sont dues à un manque de respect des bonnes pratiques par les acteurs lors d'une étape du cycle de développement, qu'il s'agisse de la conception, de l'implémentation ou de l'intégration. Tous les membres d'une équipe projet doivent donc être sensibilisés aux enjeux et risques de sécurité, et formés aux mécanismes de sécurité de base. Tous les acteurs sont importants : la maîtrise d'ouvrage doit être en mesure d'identifier les enjeux de sécurité pour exprimer les besoins. La maîtrise d'œuvre, les développeurs doivent être formés afin de pouvoir mettre en place des mécanismes de sécurité appropriés et efficaces. Cette démarche de sensibilisation et de formation doit couvrir les risques et mécanismes de sécurité spécifiques aux applications et technologies Web.

Les mécanismes de sécurité sont en constante évolution et de nouveaux types de vulnérabilités sont découverts chaque année. Il est donc important pour les parties prenantes de suivre régulièrement des formations et d'effectuer une veille sécurité afin de se garder informées des nouvelles techniques d'intrusion et de piratage.

V.2 - Identification des besoins et appréciation des risques

L'identification des besoins de sécurité est fondamentale. C'est durant cette étape que sont prises en compte les caractéristiques fonctionnelles pouvant avoir un impact sur la sécurité ainsi que les besoins de sécurité identifiés par le métier : ouverture sur Internet, sensibilité des données manipulées, accessibilité 24h/24, traçabilité, obligations légales, populations d'utilisateurs, cas d'utilisation,… Durant cette étape, l'aide d'un expert de la sécurité des applications Web peut être utile afin d'aider le métier à identifier les risques et les besoins de sécurité.

Une première évaluation du coût peut être réalisée à ce stade afin de rester cohérent avec les objectifs de la maîtrise d'ouvrage, en utilisant une méthodologie comme OpenSAMM, qui permet d'estimer des coûts pour les différentes étapes du cycle de développement [7].

Une appréciation des risques peut ensuite être réalisée afin de modéliser et d'anticiper les menaces liées au contexte d'utilisation et au fonctionnement de l'application Web. Cette étape permet de s'assurer que l'ensemble des risques a bien été pris en compte, et d'identifier des solutions et mesures de sécurité appropriées en fonction de la probabilité et de l'impact des risques potentiels identifiés. Des méthodes et des outils de modélisation de menaces accessibles existent afin de faciliter cette démarche. [8]

V.3 - Conception et implémentation

Une fois les besoins de sécurité et les menaces identifiés, il convient de concevoir la sécurité de l'application Web, c'est-à-dire de définir précisément les mécanismes de sécurité qui permettront d'assurer l'atteinte des objectifs de sécurité et de répondre aux menaces (authentification, contrôle d'accès, protection contre les injections, etc.). Un document de conception doit décrire formellement ces mécanismes, en fournissant une bonne visibilité sur la manière dont la sécurité sera organisée et les menaces seront gérées. Les mécanismes de sécurité ainsi conçus doivent en outre respecter le principe de la défense en profondeur, qui veut que si une ligne de défense est défaillante ou franchie par l'attaquant, une deuxième voire une troisième ligne soient en place pour protéger les ressources.

Une fois la conception de l'application et de sa sécurité réalisée vient la phase d'implémentation, qui voit les développeurs générer le code source et assembler les composants. Il est alors primordial que les personnes en charge de l'implémentation aient été spécialement formées au développement sécurisé d'applications Web. En outre, des outils doivent être fournis aux équipes :

- un référentiel documentaire qui présente les bonnes pratiques de développement sécurisé,
- une check-list qui permet de s'auto-contrôler et de s'assurer que rien n'a été oublié ;
- des API ou un framework « sécurité » qui évitent d'avoir à recréer des mécanismes de sécurité de base (authentification, filtrage des données, etc.). Une attention particulière doit être portée sur l'utilisation d'API ou de framework internes ou externes qui doivent être validés ou audités afin de s'assurer qu'ils ne portent pas de vulnérabilités ou de code malveillant.

Dans cette phase également, la possibilité pour les équipes de consulter un expert sécurité afin de valider les mécanismes de sécurité développés est un plus. Les équipes peuvent également se référer au Guide de conception et d'implémentation d'applications Web sécurisées de l'OWASP [9].

V.4 - « Recette » de la sécurité

A l'issue de l'implémentation, de même qu'une recette est réalisée par les utilisateurs, la sécurité de l'application doit être vérifiée, afin de valider de manière pragmatique que l'application se comporte de manière sécurisée face aux attaques (voir ci-après le chapitre sur la vérification de la sécurité).

VI - Vérification de la sécurité des applications Web

VI.1 - Pourquoi vérifier ?

Une application Web est le plus souvent un assemblage complexe de briques logicielles (serveur web, serveur d'application, moteur de script, base de données, firewall, reverse proxy) et de code source spécifiquement développé, gérant l'interface utilisateur, les traitements métiers et les accès aux données. Même si la sécurité a été prise en compte dès le début du projet, il n'est pas rare que des erreurs de conception, d'implémentation ou d'intégration existent et permettent au final des atteintes à la sécurité des données et des traitements. La vérification de la sécurité des applications Web est donc primordiale et des travaux de revue doivent être prévus durant le cycle de développement et de vie des applications.

VI.2 - Comment vérifier ?

Plusieurs approches peuvent être utilisées pour vérifier la sécurité d'une application Web, chacune ayant ses avantages et ses inconvénients :

VI.2.1 - Audit des spécifications

Cette approche consiste à considérer des scénarios de menaces et à évaluer comment les architectures techniques et mécanismes de sécurité prévus dans les spécifications sont à même de protéger l'application, ses données et ses traitements. Elle peut être réalisée dès la phase de conception, avant même la phase d'implémentation, indépendamment des technologies utilisées. Elle ne permet toutefois pas de se prémunir contre d'éventuels problèmes d'implémentation.

VI.2.2 - Audit de code

L'audit de code source consiste à analyser le code source de l'application (code Java, JSP, PHP, .Net, C/C++, etc.), afin de vérifier :

- que les mécanismes de sécurité permettant de protéger l'application sont bien présents ;
- que les règles de contrôle interne métier sont bien appliquées ;
- que le code source ne contient pas de bugs pouvant permettre à un attaquant de contourner la sécurité.

L'audit de code source a de nombreux avantages. Le fait d'analyser le code source peut en effet permettre :

- de vérifier le respect des bonnes pratiques et du principe de la défense en profondeur ;
- d'avoir un niveau d'assurance élevé quant à l'absence de failles ;
- de déceler des failles qu'un test d'intrusion n'aurait pas permis d'identifier ;
- d'identifier facilement ce qu'il faut faire pour corriger la faille ;
- de réaliser l'audit dès la fin de l'implémentation de l'application, voire même d'une partie de l'application.

Toutefois :

- il est nécessaire de pouvoir disposer de tout le code source sous forme électronique ce qui n'est parfois pas possible (notamment dans le cas des librairies compilées) ;
- le code mis en production est parfois différent du code audité ;
- Il est difficile pour l'auditeur d'avoir une vision globale de l'application et de sa sécurité à partir du seul code source. Cette approche est donc souvent associée à un test d'intrusion.

L'OWASP a publié un manuel de revue de code des applications Web [10]

VI.2.3 - Test d'intrusion

Le test d'intrusion est une approche déjà utilisée dans d'autres environnements que les applications Web. Il consiste à confronter l'application Web à une situation d'attaque réelle, en simulant les actions d'une personne mal intentionnée. Le test peut être réalisé

- sans connaissance et sans habilitations initiales, afin de simuler un attaquant externe (parfois appelé « tests en boite noire ») ;
- avec les connaissances et habilitations d'un utilisateur de base, pour simuler les attaques d'un utilisateur légitime, mais malveillant (parfois appelé « tests en boite grise ») ;
- avec les connaissances d'un développeur de l'application (mise à disposition de l'attaquant du code source) : (parfois appelé « tests en boite blanche »).

L'intérêt du test d'intrusion est de permettre :

- de déceler d'éventuelles failles sur l'application Web elle-même, mais aussi des failles (patch manquant, problème de configuration) sur la plateforme (OS, serveur Web, base de données, etc.) hébergeant l'application.
- de tester de manière pratique et de bout en bout la sécurité de l'environnement ciblé.

Mais les tests d'intrusion présentent aussi des inconvénients :

- un test d'intrusion peut entraîner des risques d'indisponibilité de l'application Web, ce qui peut être gênant quand un environnement de production est testé ;
- un test d'intrusion ne peut être réalisé qu'à la fin du cycle de développement de l'application. Si une faille de conception est identifiée alors, sa correction peut se révéler très coûteuse ;
- un test d'intrusion ne permet pas d'évaluer les fonctionnalités et mécanismes de sécurité qui ne sont pas accessibles. Il ne permet notamment pas de tester la défense en profondeur ;
- il est parfois difficile, voire impossible, de couvrir de façon exhaustive toutes les formes d'attaques (notamment les différentes formes d'injection).

Pour plus d'information, on pourra consulter le manuel de test de la sécurité des applications Web publié par l'OWASP [11]. Le CLUSIF a également publié un document sur les tests d'intrusion (non spécifique aux applications Web) [12].

VI.2.4 - Revue de l'infrastructure d'hébergement

La sécurité d'une application Web peut dépendre de la configuration des composants de l'infrastructure qui l'héberge. Un paramétrage non adapté dans le serveur Web peut par exemple permettre de s'introduire sur le serveur, et donc de mettre en défaut la sécurité de l'application Web. Aussi il peut être approprié de compléter un audit de code en réalisant une revue du paramétrage des serveurs Web, serveurs d'application, bases de données ou firewall utilisés pour mettre en œuvre l'application, afin de vérifier que les bonnes pratiques sont respectées.

VI.3 - Automatisation

Des outils commerciaux ou open-source peuvent être utilisés pour réaliser des audits de code ou des tests d'intrusion. Ces logiciels sont basés sur des bases de signatures et de modèles de failles. Ils permettent d'automatiser des tâches et de traiter des volumétries dans des temps relativement réduits.

Toutefois, même s'ils sont capables d'un certain degré d'intelligence, ces outils ne sont parfois pas capables d'identifier des failles très spécifiques, notamment au niveau de la logique métier. Ils sont également susceptibles de générer des faux positifs. Il est donc nécessaire qu'ils soient utilisés par un auditeur expérimenté à même de les configurer correctement, d'analyser les résultats produits et de les compléter le cas échéant.

VI.4 - Quand vérifier la sécurité d'une application Web ?

Les vérifications de la sécurité des applications Web sont encore trop souvent réalisées juste avant la mise en production, voire après. Or cette approche peut entraîner des coûts non négligeables. En effet, si la vérification met en évidence des failles de sécurité qui doivent être corrigées, le coût de correction est d'autant plus élevé qu'elle intervient tard.

Il est donc plus indiqué de planifier des vérifications de la sécurité d'une application Web tout au long du cycle de développement. L'approche suivante peut être suivie :

- Revue « papier » dès la conception ;
- audit de code dès l'implémentation ;
- tests d'intrusion au moment des tests utilisateur ;
- revue d'infrastructure avant la mise en production.

En outre, le niveau de sécurité d'une application Web doit être vérifié tout au long de son cycle de vie, notamment lors d'évolutions ou d'ajouts de fonctionnalités.

VII - Prise en compte des développements externalisés

De nombreuses motivations peuvent conduire à externaliser tout ou partie du développement d'une application Web : accès à des compétences externes pointues, gain de temps, réduction des coûts, etc.

Or comme on l'a vu plus tôt, la sécurité d'une application Web doit être traitée en amont. Le donneur d'ordre doit exprimer les besoins de sécurité dans le cahier des charges et contrôler qu'ils ont bien été pris en compte. Ces éléments sont primordiaux dans le cas d'une externalisation. Lors d'appel à de la sous-traitance, les engagements suivants peuvent être demandés au prestataire :

- engagement à suivre un guide de développement sécurisé et à former ses développeurs à la sécurité des applications Web ;
- documentation de l'architecture et des mécanismes de sécurité mis en place ;
- engagement à prendre à son compte les corrections des failles de sécurité qui seraient découvertes durant la vie de l'application, qu'elles soient présentes dans le code écrit, mais aussi dans les éventuels composants, open source ou non, utilisés ;

A la livraison, il est souhaitable que le donneur d'ordre ou l'acheteur fasse procéder à une vérification de la sécurité de l'application Web, selon l'une des approches décrites dans le chapitre « vérification » ci-avant. Cette vérification est d'autant plus nécessaire dans le cas d'une sous-traitance que la pression sur les prix peut entraîner les prestataires à négliger l'aspect sécurité. La sécurité doit faire partie intégrante des aspects que le client d'un sous-traitant doit recetter et valider avant de prononcer l'acceptation et de payer.

VIII - Références

[1] http://www.lemonde.fr/technologies/article/2009/08/20/130-millions-de-cartes-bancaires-piratees-aux-etats-unis_1230199_651865.html

[2] http://www.zdnet.fr/actualites/internet/0,39020774,39703886,00.htm

[3] http://pro.01net.com/editorial/379143/une-attaque-massive-transforme-des-sites-web-en-nids-a-virus/)

[4] https://www.owasp.org/index.php/OWASP_Secure_Software_Contract_Annex

OWASP Secure Software Contract Annex : Cette annexe de contrat est destinée à aider les développeurs de logiciels et leurs clients à négocier d'importantes conditions contractuelles relatives à la sécurité du logiciel à développer ou à livrer. La raison en est que rien n'est prévu dans la plupart des contrats, les parties ayant souvent des points de vue radicalement différents sur ce qui a été initialement effectivement convenu. De fait, la définition claire des responsabilités et limites de chacun est la meilleure façon de s'assurer que les parties puissent prendre des décisions éclairées sur la façon de procéder.

[5] CNIL : Loi n° 2004-801 du 6 août 2004 relative à la protection des personnes physiques à l'égard des traitements de données à caractère personnel et modifiant la loi n° 78-17 du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés et Article 35 - Code Pénal : Article 226-16 et Article 226-17

[6] http://www.owasp.org/index.php/OWASP_Top_Ten_Project

[7] http://www.opensamm.org/

[8] http://www.owasp.org/index.php/Threat_Risk_Modeling

[9] http://www.owasp.org/index.php/Category:OWASP_Guide_Project

[10] http://www.owasp.org/index.php/Category:OWASP_Code_Review_Project

[11] http://www.owasp.org/index.php/Category:OWASP_Testing_Project

[12] https://www.clusif.asso.fr/fr/production/ouvrages/pdf/TestIntrusion.pdf

L'ESPRIT DE L'ÉCHANGE

CLUB DE LA SÉCURITÉ DE L'INFORMATION FRANÇAIS

30, rue Pierre Sémard
75009 Paris
✆ 01 53 25 08 80
clusif@clusif.asso.fr

Téléchargez les productions du CLUSIF sur

www.clusif.asso.fr