|  |  |  |
| --- | --- | --- |
| **Conséquences** perte financière, perte de réputation, perte d'image de marque (cas : Loveletter, 2000), perte d'avantage compétitif, perte de temps (cas : Blaster,2003), perte de fonctionnalité (cas : SQL Slammer, 2003)  **Modèle en oignon**  Physique, réseau, protocole, hôte, application, information. De la couche réseau à la couche applications c'est la sécurité technique.  **CIA**   * Confidentialité : S'assurer que l'information est accessible seulement à ceux qui sont autorisés à y avoir accès. * Intégrité : protéger l'exatitude et la complétude de l'information et des méthodes de traitement. * Disponibilité : s'assurer que les utilisateurs autorisés ont accès à l'information et aux ressources associées au moment et au lieu exigés.   **AAA**   * Authentification : s'assurer que la personne est bien celle qu’elle prétend être ; déterminer son identité et éventuellement son rôle ; * Autorisation : détermine en fonction de l’identité (ou rôle), que cela soit une personne ou un système, si l’accès (ou le traitement) est autorisé ; * Comptabilisation : s’assurer qu’il soit possible de suivre les accès/traitement qui ont été effectués ; journalisation (logging)   **Domaines de la SSI**   * Sécurité physique : protection, prévention et récupération physique. Tout ce qui touche aux infrastructures. * Sécurité organisationnelle : couvre les aspects sociaux, légaux, organisationnels et humains liés à la SSI. Comme les procédures, contrats, cahier des charges et formation * Sécurité technique : La sécurité technique concerne le traitement, le stockage et la communication de l'information numérique (ou analogique).   **5 principes fondamentaux**   * La sécurité globale est aussi forte que le maillon le plus faible * La sécurité parfaite est impossible * La sécurité est un processus (pas un produit) * La sécurité est inversement proportionnelle à la complexité * Participation des utilisateurs   **9 règles fondamentales**   * Interdiction par défaut * Moindre privilège * Défense en profondeur * Séparation des fonctions * Segmentation * Economie de mécanismes * Goulet d’étranglement * Interruption/erreur sûre * Eviter la sécurité par obscurité   **Menaces**  accidentelles, environnementales, délibérées  **Vulnérabilités**  Matériel, Logiciel, Réseau, personnel, site physique, organisation. Conceptuelles, d'implémentation et de configuration.    **Codes Malveillants description et classification**  Les dégâts qui en découlent sont la perte de données, la perte de ressources et de temps, le manque à gagner ou la perte de crédibilité.  **Un virus** : Un virus est un code malveillant qui se propage à l'aide d'un autre programme ou fichier. Il ne peut pas se propager sans cet hôte  **Fichiers Exécutable**: Infection de fichier exécutable par le virus qui continue à exister même si l'hôte est inactif.  **Secteur d'amorçage** : Les virus qui s'installent sur le secteur d'amorçage peuvent s'introduire dans la mémoire avant que l'OS ne démarre. Il est aussi possible que le secteur d'amorçage d'une clé USB soit infecté. Normalement avant de démarrer l'OS fait une vérification du secteur d'amorçage.  **Macros**: Les macros des fichiers texte peuvent être utilisé pour créer des virus.  **Vers**: Un vers peut se propager par ses propres moyens en faisant usage des réseaux informatiques. Ils passent d'un ordinateur à un autre en exploitant des failles logicielles ou envoyant des copies d'eux-mêmes a d'autre machine via mail.  **Chevaux de Troie**: Les chevaux de Troie sont des programmes à l'apparence utile ou ludique alors qu'il contient des fonctionnalités malveillantes. Elles sont exécutées lors du démarrage du programme. La fonction malveillante de ces derniers est souvent une backdoor, un rootkit ou un spyware.  **Backdoors**: Programme qui permet de prendre le contrôle de la machine à distance. Elles permettent de copier des fichiers, de lancer des programmes à distance.  **Spywares et adwares**: Le spyware sont des logiciels espions qui recueille des informations sur la machine infectée et qui les transmet à un serveur central et cela dans le but de lui adresser des pubs ciblés. Les adwares sont les logiciels qui affichent cette publicité.  **Rootkits**: Ensemble d'outils pour cacher la présence d'un logiciel malveillant sur une machine. À la base ce sont des commandes UNIX. L’objectif d'un rootkit est de cacher la présence d'une backdoor ou d'un spyware.  **Rançongiciels et cryptovirus**: Le ransomware est un logiciel qui essaie de forcer sa victime à verser une rançon en l'empêchant de lancer sa session. Les versions plus avancées cryptent les données de l'utilisateur.  **Polymorphisme**: Le polymorphisme permet aux virus de changer leurs caractéristiques à chaque infection. Le but étant de changer l'apparence du code sans changer sa sémantique.  **Antivirus**: Ils utilisent la liste de tous les codes malveillants connus et cherchent leurs signatures. Dans un deuxième temps ils peuvent l'analyser statiquement pour voir si le programme fait des opérations douteuses. Et dans un troisième temps, ils peuvent simuler un démarrage pour analyser le comportement. Scanner tous les fichiers entrants sur le serveur. Mise à jour de base de données de signature. Utiliser plusieurs antivirus.  **Protections sur les 4 niveaux**   * Tous les postes clients * Serveurs de fichiers * Serveurs de messagerie * Proxies Internet   **Intrusion de systèmes**   * Parcours d’arborescence * Protections Client-side * Session hijacking * Cross-site scripting XSS faille web permettant d’injecter du contenu dans une page * XSS réfléchi : données fournies par le client web (ex : moteur de recherche) XSS stocké : données fournies par le client sont stockées sur srv (ex : forum, livre d’or) * Cross-site request forgeries CSRF * injection SQL * crypto-locker   **Freiner un attaquant en mode hors-ligne**  -Plusieurs hach pour chaque mot de passe -fonction de hash qui dure plus longtemps (hachage qui prend du temps) -mot de passe plus complexe  **Freiner un attaquant en mode en ligne**  -Délai d’attente entre chaque tentative -bloquer l’user qui fait plusieurs tentatives -ajouter un Captcha  **Injection SQL toujours vraie**  'OR'1==1 => chaine de caractère qui retourne VRAI lorsqu’elle se trouve dans le WHERE  **XSS VS CSRF (Cross-Site Request Forgeries)**  La différence est que la cible, l’abus de confiance et les actions effectuées se font sur le client pour le XSS et sur le serveur pour CSRF  Il est basé sur une architecture client-serveur. C'est un protocole de la couche application. Il est sans état (chaque requête est indépendante). 8 méthodes dont les 2 plus utilisées, GET et POST. En cas de succès de la requête, la réponse peut simplement retourner un fichier, faire une génération de réponse dynamique ou retour des métadonnées sans corps de message.  URI uniform resource identifier. URN uniform name. uniform resource locator. Une page web est consituée d'objets.  Pour rendre HTTP stateful, il faut utiliser des URL personnalisées et des cookies. Les cookies utilisent 2 nouveaux en-têtes HTTP.  **EBP et ESP :** désignent la stack  **DES** a une sécurité insuffisante **TDES** taille de bloc de 64 bits  Pour détecter des modifications dans un message à clés symétriques il faudrait un tag d’intégrité.  La crypto symétrique est plus rapide que la crypto asymétrique. | **Spams**  Filtrage : par mot clé, blacklist, greylist, par corrélation. Définition : envoi de masse et falsification du courrier électronique. Objectifs directs :   * Publicité, Fraude, intrusion de malware, phishing Dégâts indirects : * Surcharge de serveur * Admin de la messagerie est inondé de message d'erreur pour des adresses invalides * Le fournisseur internet peut couper la ligne, le serveur se fait placer sur liste noire Prévention du spam * Ne pas l'afficher, ne pas la transmettre, Filtre sur les serveurs sur la base des liste noires * Configurer les serveurs correctement   **Messagerie électronique**   * SMTP : sert à envoyer des messages, POP et IMAP servent à rapatrier des messages pour leur lecture * POP : récupère le courrier en local, IMAP : permet de consulter les mails sur un serveur, laisse les messages sur le serveur.   **Attaque par e-mails forgés**  Falsification du courrier électronique en exploitant une faille conceptuelle de SMTP qui ne vérifie pas les sources. Informations fiables dans un e-mail :   * Fiables : champ Received, champ Return-Path, champ Delivered-to * Normalement fiable : peut-être forgé   **Attaques logicielles** une fois qu'un programme est distribué, il peut être soumis à toutes sorte d'attaque   * Analyse : extraction d'algorithme, architecture * Modifie ou réutilise le logiciel * Distribue un programme modifié la protection logicielle inclut : * Développement sécurisé * Éviter de pouvoir insérer du code malveillant * Éviter de contourner les protections * Éviter les vols de code   **Manipulation mémoire**   * Écriture ou lecture en mémoire à un endroit non autorisé, buffer overflow * Un entier mal interprété, une chaîne est mal formée et interprétée de manière inattendue   Le buffer overflow est un bug qui permet d'écrire à l'extérieur de l'espace alloué au tampon en écrasant les données écrites à l'emplacement. Un shellcode est une suite d'instruction destiné à être injecté puis exécutées.  **Organisation mémoire**   * Stack stockage dynamique, Heap allocation de mémoire, .bss données globales initialisées * .data données globales non-initialisées * .text code exec L'ESP (stack pointer) pointe l'adresse basse mémoire qui le top de la pile. L'EBP est l'endroit où est stocké la base de la pile L'EIP pointeur d'instruction. Lors d'un saut dans une fonction, un saut est effectué. Dès lors on doit sauver l'environnement. Appelant : * Pousse les paramètres (de droite à gauche). * Appel la fonction : sauve l'adresse de retour dans EIP, sauter à l'adresse de la fonction * Appelé : sauve et initialise EBP, exécute son code, recharge l'adresse de retour * Appelant : restaure le stack pointer     **Différentes attaques**   * Stack smash: les variables sont écrasées * Stack off-by-one: dépassement d'un seul caractère * heap overflow: les variables sur le heap sont manipulées * integer overflow: entiers dépassant la taille * Format string bug: mauvaise interprétation d'un string   **Protection**  Rendre la stack et le heap non exécutable, utilisation de canaris, randomisation des adresse mémoire librairies sécurisées  **Manière de casser un mot de passe, recherche**   * Naïve = on entre tous les mdp possible * Exhaustive (force brute) = test avec a, aa, aaa, aaa0… ok pour le sel * Dictionnaire = mots courants, habituels, existants * Heuristique = variation des mots du dictionnaire (maison32) ok pour le sel * Pré-calcul d’empreintes = table avec pleins de mdp et leur empreinte * Rainbow table = on prend des mdp, on effectue des hash puis réduction plusieurs fois, On cherche une empreinte, si elle n’est pas dans la table, on effectue des hash et réduction jusqu’à trouver l’empreinte dans la table correspondant à un mot de passe ne va pas pour le sel * Compromis temps mémoire ne va pas pour le sel   **Introduction à la cryptographie** La cryptographie apporte confidentialité, authenticité et intégrité. De même que la non-répudiation, l'anti-clonage. La cryptographie étudie les codes secrets qui permet d'envoyer de l'information de manière confidentielle.   * Un Cipher est un algorithme permettant de rendre l'information confidentielle en général à clé secrète. * Un Pk est algorithme de chiffrement et déchiffrement à clé publique. La théorie du codage est la science qui permet d'envoyer de l'information de manière fiable et efficace. * Le code est un système de symboles représentant de l'information. La stéganographie est l'art de dissimuler une information en la cachant, mais en ne transformant pas son contenu. La cryptographie symétrique utilise une clé secrète pour chiffrer et déchiffrer l'information. Elle utilise la même clé est requise pour le chiffrement et le déchiffrement. Pas de notion d'authentification de l'origine. * Exemple : DES (Block ciphers), AES (Block ciphers), RC4 La cryptographie asymétrique à clé publique génère une paire de clé. Une clé publique pour chiffrer le message et une clé privée pour déchiffrer le message. Chaque utilisateur possède une paire. Pas de secret partagé entre les deux utilisateurs. * Exemple : RSA(Stream ciphers)   Le scytale pratique le chiffrement par transposition. Le chiffre de César remplace chaque lettre K lettre après elle. Le chiffrement de Vigenère propose l'idée de chiffrer via plusieurs caractères. Mais ces solutions sont rapidement cassées par la force brute ou l'études des fréquences des lettres.  **Les principes de Kerckhoffs**   * La confidentialité du message doit reposer sur la confidentialité de la clé uniquement * On ne fait pas reposer la sécurité sur la confidentialité de l'algorithme Vernam cipher utilise une clé aléatoire uniformément distribuée, pour ne pas pouvoir déduire des informations du texte, et de même taille que le message. On effectue un ou exclusif entre les deux pour chiffrer et déchiffrer. Il offre un chiffrement parfait mais il est inutilisable. La confidentialité parfaite signifie qu'on ne peut pas apprendre quelque chose sur le cipher text si on connait le plain text est vice versa.   fig:  **DES** Le Data Encryption standard est basé sur Lucifer et influencé par la NSA. C'est un block cipher qui chiffre par block de 64 bits. Il utilise des clés de 56 bits voir 64 bits. Avec cela on peut chiffrer de long message L’algorithme fonctionne dans un premier temps par une permutation des blocs. Puis vient le chiffrement par ronde. Seize tours sont effectuées avec chaque fois une clé différente. Lors de chaque tour le block est divisé en deux et permuter selon un schéma feistel. Pour finir une permutation inverse à celle effectuée dans le premier temps est faite. DES possède des faiblesses connues. Et la recherche de clé est possible. Des Variantes TDES sont possibles mais trop lente.  **Mode de chiffrement**  L'Electronic code book chiffre chaque block indépendemment. Le Cipher block chaining utilise la partie chiffrée du block précédent pour chiffrer l'actuel block. ECB : Electronic Code Book (sans mode de chiffrement) On sépare le texte clair en blocs et on chiffre le bloc avec une clé et on recompose chaque bloc chiffré CBC : Cipher Block Chaining (le plus utilisé) On sépare le texte clair en blocs et on fait un XNOR avec un vecteur d’initialisation puis on prend le texte chiffré et on fait un XNOR avec le prochain bloc du texte clair etc… • Il existe des modes de chiffrement pour l'authentification (CBC-MAC, ...) • Il existe des modes de chiffrement avec authentification (EAX, ...)  **Diffie-Hellman**  La distribution de clé est toujours un problème. Pour résoudre ce problème, On utilise des fonctions difficilement réversibles.  **RSA** La paire de clé privée/publique dépendent mathématiquement l'une de l'autre. Tout le monde peut accéder à la clé publique mais personne ne peut accéder à la clé privée. RSA est sûr car la factorisation est un problème difficile. **Attaque « man-in-the-middle »** Un générateur de clé publique et privée. Un message est encrypté avec une clé publique par A puis l’attaquant parvient à décrypter le message et le ré-encrypte pour l’envoyer à B et B décrypte le message sans se rendre compte de rien avec sa clé privée. Ne permet pas d’envoyer de long message plus grand que le modulus.  La difficulté de factoriser un grand nombre composite (composite = pas premier)  **Fonction de hachage** Un message qui a passé par une fonction de hachage est appelé digest. Elles sont résistantes aux collisions, car il est infaisable de trouver deux messages avec le même haché. Elles sont résistantes aux préimages. Utilisations :   * Engagement / preuve pour caractériser un message de manière unique. * Extension de domaine en longueur fixe * Génération de nombre pseudo aléatoire à partir d'une graine * Exemple MD (Message Digest) = md2, md4, md5 et SHA (Secure Hash Algorithme) = SHA-0, SHA-1, SHA | **Modèle de Merkle-Damgard** Les fonctions de compression sont un mélange de transpositions, de décalages, de substitutions, xor. Lors du découpage en bloc, il y a toujours un padding.  **Authentification de données MAC et signature**  MAC est un code de taille fixe qui est envoyer avec un message pour prouver son intégrité et une origine. Le tag a taille fixe de 160 bits. Exemple : pour authentifier les messages Diffie-Hellman. ISO/IEC 9797 est une meilleure variante. Une signature évite toute répudiation de la source. Elle permet de s'assurer de la provenance, de le prouver, de prouver la livraison et de garantir l'intégrité des données. RSA et DSA permettent de signer.  **MAC : Fonctionnement**  A envoie un message à B. Le message est passé dans la fonction MAC avec une clé secrète et on obtient le tag. B utilise la fonction MAC avec la même clé secrète et on obtient le tag. Il compare et déduit si c’est authentique ou non (faux message ou non, notamment s’il y a man in the middle par exemple)  MAC : Constructions -> HMAC SÉCURISÉ -> On prend une clé secrète, on fait un XNOR avec ipad (constante) et on concatène au Message. On passe le tout dans la fonction de hash. On reprend la clé secrète et on fait Un XNOR avec ipad (constante) et on concatène ça au hash fait précédemment. Puis on Passe le tout dans une fonction de hash pour en sortir le tag. • Constructions basées sur des algorithmes de chiffrement -> ISO/IEC 9797 SÉCURISÉ -> On prend X1, on le chiffre. Puis on fait un XNOR avec X2 etc…A la fin, on prend la Première fonction de chiffrement utilisée et on chiffre avec la 2ème fonction de Chiffrement puis on passe dans une fonction « trunc » et on en sort le MAC  **Authentification**  Facteurs d’authentification : l'authentification forte utilise au moins deux types de facteurs.   * Ce que l'on sait, ce que l'on possède, ce que l'on est * Authentification forte : utilisation d'au moins 2 types de facteurs ci-dessus. Types d’authentification : * Par jeton passif fixe dans le temps : mot de passe, badge magnétique * Par jeton actif changeant dans le temps : liste à biffer, SecureID. * Par question réponse : fonctionnement asynchrone * Par la biométrie * Par un autre canal   **Signature vs chiffrement** Chiffrement : A envoie un message à B. A encrypte le message avec la clé publique de B et B décrypte avec sa clé privée. Signature : A signe avec sa clé privée et B vérifie le message avec la clé publique de A  **Signature vs MAC** Signature : expliqué en haut MAC : A passe le message dans la fonction MAC avec une clé secrète. Le tag est envoyé avec le message. B utilise la fonction MAC avec la même clé et vérifie avec le tag envoyé.  **Hash and sign** A hash le message, le signe avec sa clé privée. B hash le message reçu et le vérifie avec la signature sortie après le hash par A et avec la clé publique de A.  **Algorithmes de signature** •RSA (avec p. ex. MD5 ou SHA-1). • DSA : Digital Signature Algorithme (1993)  **Authentification à clé publique** • Question/réponse avec signature électronique d'un nonce • Nonce : « Number used ONCE » (aléatoire de taille suffisante ou compteur)  **PKI : infrastructure à clé publique**   * Ensemble d'éléments matériels et logiciels, protocoles et services. * Rôle : gestion des clés publiques à grande échelle. Il n'y a plus besoin de connaître son interlocuteur ni de lui faire confiance. * CA « Certificate Authority » : émettre, renouveler et maintenir les certificats * RA « Registration Authority »: gérer les demandes * VA « Validation Authority » : service de contrôle des certificats * CPS : « Certification Practice Statement » * CP : « Certificate Policy »   Trouver la longueur d’une clé = autocorrélation, une fois le fini, il repère les répétitions Trouver une clé = analyse de fréquence, répétitions de caractères comme a et e en fr Dissimuler u message dans un autre = stéganographie (ne fait pas partie de la crypto)  **Résoudre une faille XSS** Utiliser des fonctions / commandes de sanitarisations afin de supprimer toutes les balises qui pourraient modifier le fonctionnement normal de l’application WEB.  **Authentification à l’aide de certificat** On reprend le concept d’authentification expliqué en haut et on envoie simplement un certificat de la clé publique à utiliser avec la signature  **Certificat**  Ce lien est certifié par une autorité tierce : -> souvent l'autorité de certification (Certificate Authority ou CA) ou entre users • Concrètement : -> un certificat est un fichier texte d'environ 1 KB qui contient surtout le nom de l'entité, la clé publique associée et la signature de cette clé publique (signée par la CA)  **Sécurité réseaux** Whois:  Interrogation des registres internet, informations à usages varié. info sur adresse IP et nom de domaine. nslookup : interrogation DNS host, dig: information sur hôtes non-accessibles publiquement traceroute:reconnaissance réseau  **Autres types d’attaques**  Phishing = hameçonnage, pour vol d’identité Scams = arnaque, financière (1'000'000 à gagner !) FPharming = dévoiement, comme l’hameçonnage mais la destination des données est autre ClickJacking = détournement de click, force l’utilisateur à cliquer Man-in-the-Browser (SpyExe, Zeus)  **scanning**  Déterminer les machines vivantes, quels services tournent ou écoutent, déterminer les protocoles réseaux. Protections contre le scanning : Toujours utiliser la dernière version, Filtrage des messages ICMP, Utilisation de pare-feu / IDS, Utilisation de proxys inverses, Bannières minimales  **Protection contre scanning**  Toujours utiliser la dernière version, filtrage message ICMP, utilisation pare-feu, utilisation de proxys inverses, bannières minimales. sniffing: écoute du traffic réseau spoofing: falsification d'identité source man-in-the-middle: ARP spoofing et DNS cache poisoning Session hijacking: vol de session par les protocoles TCP et HTTP DoS: déni de service, nuire à la disponibilité d'un système d'information  **Pare-feu**  Équipement situé entre deux réseaux pour faire un contrôle d'accès. Filtrage statique (obsolète) ou Filtrage dynamique. Pare-feu réseau : équipement réseau, filtrage entre 2 réseaux Pare-feu personnel : logiciel sur pc, filtrage entre pc et réseau  **TLS**  Soit création d'un nouveau protocole à partir d'un protocole existant (HTTPS). Le serveur doit avoir un certificat. Cela garantie l'authenticité du serveur et de la confidentialité des communications. Soit extension du protocole pour négocier TLS (ESMTP). TLS contient des cipher suites.  **PGP**  Utilisé pour sécuriser des emails, des textes, fichiers. garantit la confidentialité et l'authentification. Utilise la cryptographie hybride. Fonction de hachage MD5, SHA-1. Chiffrement symétrique : 3DES, IDEA,AES. Algorithme asymétrique RSA, DSA. il est nécessaire d'authentifier les clés publiques. permet plusieurs identités comparer à X.509. Un des principes : A envoie un message signé avec sa clé privée, concatène le message de base + le signé. Si y’a chiffrement => La concaténation est encryptée avec une clé random secrète, puis la clé secrète est encryptée avec des clés publiques de diverses personnes et est concaténé au message encrypté de base par la clé secrète.  **RSA**  Signature    Pour forger un autre message authentique    Pour avoir une signature valide :    Il ne faut pas utiliser RSA tel quel, il faut faire un pré-traitement sur le message avec une fonction d’hachage et après seulement on le signe avec RSA.  Pour utiliser chiffrer de long message avec RSA il faudrait chiffre la clé symétrique avec RSA.  **Un Certificat X.509v3**  Une clé publique, une signature, une période de validité ainsi qu’une identité. |