Rapport de cours
Jour 3

Victoria SAUTEREAU

Cybersécurité

E S T I A M

École d'informatique
et du numérique

2ème année
2022/2023

Sommaire

Partie 1 Les types de contrôle de l'intégrité des données

. Algorithmes de hashage

La cryptographie est une méthode permettant de stocker et de transmettre des données, de sorte que seul le destinataire désigné puisse les lire ou les traiter. La cryptographie moderne utilise des algorithmes sécurisés pour s'assurer que les cybercriminels ne puissent pas facilement compromettre des informations protégées.

La confidentialité des données garantit que seul le destinataire pourra lire le message. Pour ce faire, les deux parties ont recours au chiffrement. Il s'agit d'un processus qui consiste à brouiller les données afin de rendre leur lecture difficile par une partie non autorisée.

1. Qu'est-ce que le hash ?

Les utilisateurs doivent avoir la garantie que leurs données ne subiront aucune modification, qu'elles soient au repos ou en transit. Le hash est un outil qui assure l'intégrité des données en prenant des données binaires (le message) et en générant une représentation de longueur fixe, appelée valeur de hash ou condensé de message, comme illustré sur cette figure.

L'outil de hash utilise une fonction de hash cryptographique pour vérifier et garantir l'intégrité des données. Il peut également vérifier l'authentification. Les fonctions de hash remplacent les clés de cryptage ou le mot de passe en clair, car il s'agit de fonctions unidirectionnelles. Cela signifie que si un mot de passe est hashé avec un algorithme de hash spécifique, le condensé de hash obtenu sera toujours le même. Le qualificatif « unidirectionnel » est utilisé dans la mesure où, avec les fonctions de hash, il est impossible, sur le plan de traitement, que deux ensembles de données différents génèrent une sortie ou un condensé de hash identique.

Chaque fois que les données sont modifiées ou altérées, la valeur de hash change également. C'est la raison pour laquelle les valeurs de hash cryptographiques sont souvent désignées sous le nom d'empreintes numériques. Elles peuvent détecter les fichiers de données en double, les changements de version du fichier et les applications similaires. Ces valeurs constituent une protection contre toute modification accidentelle ou intentionnelle des données, et contre leur corruption accidentelle. Le hash s'avère également très efficace. Un fichier volumineux ou le contenu d'un disque entier donne comme résultat une valeur de hash de même taille.

2. Propriété du hash

Le hash est une fonction mathématique unidirectionnelle relativement simple à calculer, mais extrêmement difficile à inverser.
U ne fonction de hash cryptographique possède les propriétés suivantes :

- Il n'y a pas de limite de longueur pour le texte saisi.

- La longueur du résultat est fixe.
- La fonction de hash est unidirectionnelle et irréversible.
- Deux valeurs d'entrée différentes donneront pratiquement toujours deux valeurs de hash identiques.

3. Algorithmes de hashage

Les fonctions de hash se révèlent particulièrement utiles pour s'assurer que les données ne sont pas modifiées accidentellement par une erreur de communication ou par un utilisateur. Prenons l'exemple d'un expéditeur qui souhaite s'assurer de l'intégrité de son message jusqu'à ce qu'il parvienne au destinataire. Dans ce cas, l'appareil émetteur entre le message dans un algorithme de hash et calcule son empreinte ou condensé de longueur fixe.

Algorithme de hachage simple (somme de contrôle 8 bits)

La somme de contrôle 8 bits fut l'un des premiers algorithmes de hash. Il s'agit de la forme la plus simple d'une fonction de hash. Une somme de contrôle 8 bits calcule le hash en convertissant le message en nombres binaires, puis en organisant la chaîne de nombres binaires en blocs de 8 bits. L'algorithme additionne ensuite les valeurs de 8 bits. La dernière étape consiste à convertir le résultat à l'aide d'un processus appelé « Complément à 2 ». Le complément à 2 inverse un nombre binaire, puis y ajoute 1. Cela signifie que 0 est converti en 1 et inversement. La dernière étape consiste à ajouter 1, ce qui donne une valeur de hash de 8 bits.

4. Algorithmes de hash modernes

De nombreux algorithmes de hash modernes sont largement utilisés de nos jours. Les plus populaires sont MD5 et SHA.

Algorithme MD5 (Message Digest 5)

C'est à Ron Rivest que l'on doit le développement de MD5, un algorithme de hachage utilisé aujourd'hui par plusieurs applications Internet. MD5 est une fonction unidirectionnelle qui permet de calculer facilement un hash à partir des données d'entrée spécifiées. En revanche, calculer les données d'entrée en connaissant uniquement une valeur de hash s'avère très difficile.
L'algorithme MD5 génère une valeur de hash de 128 bits. Le malware Flame a compromis la sécurité de MD5 en 2012. Les créateurs du malware Flame ont utilisé une collision MD5 pour falsifier un certificat de signature de code Windows. Cliquez ici pour lire un article consacré à l'attaque par collision du malware Flame.

Secure Hash Algorithm (SHA)

L'Institut national des normes et de la technologie (NIST) des États-Unis a développé SHA, l'algorithme spécifié dans la norme SHS (Secure Hash Standard). La publication de l'algorithme SHA-1 date de 1994. SHA-2 a remplacé SHA-1 en ajoutant quatre fonctions de hash qui composent la famille SHA :

SHA-224 (224 bits)
SHA-256 (256 bits)
SHA-384 (384 bits)
SHA-512 (512 bits)
SHA-2 est un algorithme plus puissant qui remplace MD5. SHA-256, SHA-384 et SHA-512 sont des algorithmes de nouvelle génération.

5. Hash de fichiers et de supports numériques

L'intégrité garantit que les données et informations sont complètes et ne subissent aucune altération au moment de leur acquisition. Il s'agit d'un élément important lorsqu'un utilisateur télécharge un fichier sur Internet ou qu'un expert judiciaire recherche des preuves sur des supports numériques.

Pour vérifier l'intégrité de toutes les images IOS, Cisco propose des sommes de contrôle MD5 et SHA sur son site web de téléchargement de logiciels. L'utilisateur peut comparer ce condensé MD5 à celui d'une image IOS installée sur un appareil, comme illustré sur cette figure. Il peut alors être sûr que personne n'a modifié ni falsifié le fichier image IOS.

Remarque : la commande verify /md5, illustrée ici, n'entre pas dans le cadre de ce cours.

Dans le domaine de l'expertise judiciaire en informatique, le hash est utilisé pour vérifier tous les supports numériques qui contiennent des fichiers. Par exemple, l'enquêteur crée un hash et une copie bit à bit du support contenant les fichiers pour produire un clone numérique. Il compare ensuite le hash du support d'origine avec la copie. Si les deux valeurs correspondent, les copies sont identiques. Le fait qu'un ensemble de bits soit identique à l'ensemble de bits initial établit la fixité. Cette fixité permet de répondre à plusieurs questions :

L'enquêteur dispose-t-il des fichiers auxquels il s'attendait ?
Les données ont-elles été altérées ou modifiées ?
L'enquêteur peut-il prouver que les fichiers n'ont pas été altérés ?
L'expert scientifique peut, à présent, examiner la copie à la recherche de preuves numériques, en laissant intacte la version d'origine.

6. Hash de mots de passe

Les algorithmes de hash transforment n'importe quelle quantité de données en hash numérique ou empreinte de longueur fixe. Il est impossible pour un hacker d'inverser un hash numérique pour découvrir l'entrée d'origine. Si la saisie subit une quelconque modification, cela se traduit par un hash différent. Cela fonctionne pour la protection des mots de passe. Un système doit stocker un mot de passe sous une forme qui le protège, tout en conservant la possibilité de vérifier qu'il est correct.

Cette figure illustre le workflow d'enregistrement et d'authentification d'un compte utilisateur à l'aide d'un système avec hash. Le système n'écrit jamais le mot de passe sur le disque dur ; il enregistre uniquement le hash numérique.

7. Applications

Vous pouvez utiliser des fonctions de hash cryptographique dans les situations suivantes :

- Pour fournir une preuve d'authenticité en cas d'utilisation avec une clé d'authentification secrète symétrique, comme l'authentification par protocole de routage ou IPsec (IP Security).

- Pour fournir une authentification en générant des réponses uniques et unidirectionnelles aux challenges des protocoles d'authentification.

- Pour fournir des preuves du contrôle d'intégrité d'un message, comme celles utilisées dans les contrats à signature numérique, et des certificats des infrastructures à clé publique (PKI), comme ceux acceptés lors de l'accès à un site sécurisé à l'aide d'un navigateur.

Lors de la sélection d'un algorithme de hash, utilisez SHA-256 ou un algorithme plus puissant, car il s'agit, pour l'heure, des formes les plus sécurisées. Évitez MD5 et SHA-1 en raison des failles de sécurité qui ont été découvertes. Dans les réseaux de production, optez pour l'algorithme SHA-256 ou pour une version plus puissante.

Bien que le hash puisse détecter des modifications accidentelles, il ne peut pas assurer de protection contre les actes délibérés. La procédure de hash ne comporte aucune information d'identification unique provenant de l'expéditeur. Cela signifie que n'importe qui peut calculer un hash pour n'importe quelle donnée, à condition de disposer de la fonction de hash correcte. Par exemple, lorsqu'un message transite par le réseau, un hacker peut l'intercepter, le modifier, recalculer le hash et ajouter ce dernier au message. L'appareil récepteur n'effectuera la validation que par rapport au hash ajouté et ce, quel qu'il soit. Par conséquent, le hash est vulnérable aux attaques man-in-the-middle (MitM) et ne garantit pas la sécurité des données transmises.

8. Piratage du Hash

Pour pirater un hash, le hacker doit deviner le mot de passe. Les deux méthodes les plus utilisées pour trouver des mots de passe sont les attaques par force brute et les attaques par dictionnaire.

Une attaque par dictionnaire utilise un fichier contenant des mots, des expressions et des mots de passe courants. Le fichier demande le calcul des hashs. Une attaque par dictionnaire compare les hashs du fichier à ceux du mot de passe. Si un hash correspond, le hacker connaît un groupe de mots de passe potentiellement corrects.

Une attaque par force brute essaie toutes les combinaisons de caractères possibles jusqu'à une longueur donnée. Ce type d'attaque consomme beaucoup de temps machine, mais la découverte du mot de passe n'est qu'une question de temps. Les mots de passe doivent être suffisamment longs pour que l'exécution d'une attaque par force brute prenne trop de temps pour en valoir la peine. Pour le hacker potentiel, le hash rend plus difficile la tâche de récupération de ces mots de passe.

Le salage

Le salage est une méthode permettant de renforcer la sécurité des mots de passe. Si deux utilisateurs possèdent le même mot de passe, ils auront également les mêmes hashs de mot de passe. Une valeur de salage, qui correspond à une chaîne aléatoire de caractères, est une entrée supplémentaire du mot de passe avant le hash. Elle crée un résultat de hash différent pour les deux mots de passe, comme illustré sur la figure. Une base de données stocke le hash et la valeur de salage.

Le salage empêche les hackers de lancer une attaque par dictionnaire pour essayer de deviner un mot de passe. Le salage rend également impossible l'utilisation de tables de correspondance et de rainbow tables pour pirater un hash.

Tables de correspondance

Une table de correspondance stocke les hashs de mots de passe précalculés dans un dictionnaire de mots de passe, accompagnés du mot de passe correspondant. Cette table est une structure de données qui traite des centaines de recherches de hashs par seconde. Cliquez ici pour voir à quelle vitesse une table de correspondance peut casser un hash.

Tables de correspondance inversées

Ce type d'attaque permet à un cybercriminel de lancer une attaque par force brute ou par dictionnaire sur de nombreux hashs sans la table de correspondance précalculée. Le cybercriminel crée une table de correspondance qui mappe chaque hash de mot de passe de la base de données des comptes compromise sur une liste d'utilisateurs. Le cybercriminel hashe chaque mot de passe supposé et utilise la table de correspondance pour obtenir une liste d'utilisateurs dont le mot de passe correspond à son hypothèse, comme illustré sur cette figure. Étant donné que de nombreux utilisateurs ont le même mot de passe, l'attaque s'avère fructueuse.

Rainbow tables

Les rainbow tables privilégient la réduction de la taille des tables de correspondance au détriment de la vitesse de piratage des hashs. Une table plus petite permet, en effet, de stocker les solutions d'un plus grand nombre de hashs dans un espace réduit. Cliquez ici pour accéder à une rainbow table capable de pirater n'importe quel hash MD5.

Voici quelques recommandations pour la mise en œuvre réussie du salage :

- La valeur salt doit être unique pour chaque mot de passe utilisateur.
- Ne jamais recycler une valeur salt.
- La longueur de la valeur salt doit être égale à celle de la sortie de la fonction de hash.
- Toujours effectuer le hash sur le serveur dans une application web.

L'utilisation d'une technique connue sous le nom d'étirement de clé permet de se protéger contre les attaques. L'étirement de clé ralentit sensiblement la fonction de hash. Cela rend moins efficace le matériel haut de gamme capable de calculer des milliards de hashs par seconde.

Le HMAC

Pour empêcher un cybercriminel de lancer une attaque par force brute ou par dictionnaire sur un hash consiste à ajouter une clé secrète à ce dernier. Seule la personne qui connaît le hash peut valider un mot de passe. Pour y parvenir, une méthode consiste à inclure la clé secrète dans le hash à l'aide d'un algorithme de hash appelé HMAC ou KHMAC (Keyed-Hash Message Authentication Code). Les HMAC utilisent une clé secrète supplémentaire en entrée de la fonction hash. HMAC ne se contente pas d'assurer l'intégrité des données, il ajoute une étape d'authentification. HMAC utilise un algorithme spécifique qui combine une fonction de hash cryptographique et une clé secrete.

Seuls l'expéditeur et le récepteur connaissent la clé secrète, et le résultat de la fonction de hash dépend à présent des données d'entrée et de la clé secrète. Seules les parties qui ont accès à cette clé secrète peuvent calculer le condensé d'une fonction HMAC. Cette caractéristique bloque les attaques MitM et fournit une authentification de l'origine des données.

Les codes HMAC peuvent également authentifier un utilisateur web. De nombreux services web utilisent l'authentification de base, laquelle ne chiffre pas le nom d'utilisateur et le mot de passe lors de la transmission. Avec la fonction HMAC, l'utilisateur envoie un identificateur de clé privée et un code HMAC. Le serveur recherche la clé privée de l'utilisateur et crée un code HMAC. La valeur HMAC de l'utilisateur doit correspondre à celle calculée par le serveur.

Les VPN qui utilisent le protocole IPsec dépendent des fonctions HMAC pour authentifier l'origine de chaque paquet et pour fournir le contrôle d'intégrité des données.

Partie 2 : Signatures Numériques

- ## Signatures numériques et la loi

1. Signatures numériques ?

Les signatures manuscrites et les sceaux prouvent la paternité du contenu d'un document. Les signatures numériques peuvent fournir la même fonctionnalité que les signatures manuscrites.

Un document numérique non protégé peut être modifié très facilement. Une signature numérique peut déterminer si quelqu'un modifie un document après qu'il a été signé par l'utilisateur. Une signature numérique est une méthode mathématique utilisée pour vérifier l'authenticité et l'intégrité d'un message, d'un document numérique ou d'un logiciel.

Dans de nombreux pays, les signatures numériques ont la même valeur légale qu'un document signé manuellement. Les signatures électroniques ont un caractère contraignant pour les contrats, les négociations ou tout autre document nécessitant une signature manuscrite. Une liste d'audit retrace l'historique du document électronique à des fins de réglementation et de défense juridique.

L'authenticité, l'intégrité et la non-répudiation sont établies grâce à une signature numérique. Les signatures numériques possèdent des propriétés spécifiques qui assurent l'authentification des entités et l'intégrité des données, comme illustré sur cette figure.

Elles constituent une alternative à HMAC.

2. Non-répudiation

Dans le jargon juridique, la répudiation est synonyme de renonciation. La non-répudiation est le fait de s'assurer que l'expéditeur d'un message ou document ne peut nier le fait qu'il l'a envoyé et que le destinataire ne peut nier le fait qu'il l'a reçu.

Une signature numérique garantit que l'expéditeur a signé électroniquement le message ou document. Une signature numérique étant propre à la personne qui l'a créée, cette dernière ne peut pas nier l'avoir appliquée.

- ## Fonctionnement de la technologie de signature numérique

1. Processus de création d'une signature numérique

La cryptographie asymétrique est à la base des signatures numériques. Un algorithme de clé publique comme RSA génère deux clés : une privée et l'autre publique. Ces clés sont associées mathématiquement.

2. Utilisation des signatures numériques

Signer un hash plutôt que l'intégralité du document garantit efficacité, compatibilité et intégrité. Les entreprises souhaitent remplacer les documents papier et les signatures à l'encre par une solution qui assure au document électronique le respect de toutes les exigences légales.

Voici deux situations qui illustrent l'utilisation de signatures numériques :

- Signature de code : permet de vérifier l'intégrité des fichiers exécutables téléchargés à partir du site web d'un fournisseur. La signature de code utilise également des certificats numériques pour authentifier et vérifier l'identité du site.

- Certificats numériques : utilisés pour vérifier l'identité d'une entreprise ou d'une personne afin d'authentifier le site web d'un fournisseur et d'établir une connexion chiffrée pour l'échange de données confidentielles.

3. Comparaison des algorithmes de signature numérique

Les trois algorithmes de signature numérique les plus courants sont DSA (Digital Signature Algorithm), RSA (Rivest-Shamir-Adleman) et ECDSA (Elliptic Curve Digital Signature Algorithm) Tous trois génèrent et vérifient les signatures numériques. Ces algorithmes dépendent des techniques de chiffrement asymétrique et de clé publique. Dans le cas des signatures numériques, deux opérations sont requises :

1. Génération de clé

2. Vérification de la clé

Ces deux opérations nécessitent le chiffrement et le déchiffrement de la clé.

L'algorithme DSA utilise une factorisation de grands nombres. Il est utilisé par les autorités pour créer des signatures numériques. Cet algorithme ne s'étend pas au-delà du message proprement dit.

RSA est l'algorithme de cryptographie à clé publique le plus utilisé de nos jours. Créé en 1977, cet algorithme tire son nom des initiales de ses trois inventeurs, à savoir Ron Rivest, Adi Shamir et Leonard Adleman. RSA dépend du chiffrement asymétrique. Outre la signature, cet algorithme chiffre le contenu du message.

DSA s'avère plus rapide que RSA pour la signature d'un document numérique. RSA, en revanche, convient mieux pour la signature et la vérification de documents électroniques et le chiffrement de messages.

Comme c'est généralement le cas dans le domaine de la cryptographie, l'algorithme RSA repose sur deux principes mathématiques, à savoir la factorisation de nombres premiers et de modules. Cliquez ici pour savoir comment l'algorithme RSA utilise la factorisation de nombres premiers et de modules.

ECDSA est l'algorithme de signature numérique le plus récent. Il remplace progressivement l'algorithme RSA. L'avantage de ce nouvel algorithme est de pouvoir utiliser des clés bien plus petites pour le même niveau de sécurité et de nécessiter moins de puissance de calcul que RSA.

Partie 3 : Certificats

• Principes de base des certificats numériques

Un certificat numérique est l'équivalent d'un passeport électronique. Il permet à des utilisateurs, des hôtes et des entreprises d'échanger des informations de manière sécurisée sur Internet. Plus précisément, un certificat numérique authentifie et vérifie que l'expéditeur d'un message est bien celui qu'il prétend être. Les certificats numériques peuvent également assurer la confidentialité du destinataire en lui permettant de chiffrer sa réponse.

Sur Internet, échanger continuellement des informations d'identification entre toutes les parties s'avérerait peu pratique. Les parties concernées acceptent donc l'intervention d'un tiers neutre. Il est probable que ce tiers effectue un examen approfondi avant d'émettre les informations d'identification. Ces informations d'identification sont particulièrement difficiles à falsifier. À partir de ce moment-là, toutes les personnes qui font confiance au tiers acceptent simplement les informations d'identification qu'il émet. Une autorité de certification (AC) fonctionne de la même manière que le bureau de délivrance des permis. Elle émet des certificats numériques qui authentifient l'identité des entreprises et des utilisateurs. Ces certificats signent également les messages pour s'assurer que personne ne les a falsifiés.

• Création d'un certificat numériques

Tant qu'un certificat numérique respecte une structure standard, toute entité peut le lire et le comprendre, quel que soit l'émetteur. X.509 est un standard permettant à une infrastructure de clé publique (PKI) de gérer des certificats numériques. L'infrastructure à clé publique (PKI) correspond aux politiques, aux rôles et aux procédures nécessaires pour créer, gérer, distribuer, utiliser, stocker et révoquer des certificats numériques. La norme X.509 spécifie que les certificats numériques contiennent les informations standard.

Les navigateurs et applications effectuent un contrôle de validation avant d'approuver les

certificats afin de s'assurer de leur validité. Les trois processus mis en œuvre sont les suivants :

- La détection de certificats valide le chemin de certification en vérifiant que chaque certificat commence par le certificat de l'autorité de certification racine.
- La validation de chemin sélectionne un certificat de l'autorité de certification émettrice pour chaque certificat de la chaîne.
- La révocation détermine si le certificat a été révoqué, ainsi que la raison de la révocation.

Un utilisateur reçoit un certificat pour une clé publique d'une autorité de certification commerciale. Ce certificat appartient à une chaîne de certificats désignée sous le nom de chaîne de confiance. Le nombre de certificats de la chaîne dépend de la structure hiérarchique de l'autorité de certification.

Partie 4 : Protection de l'intégrité des bases de données.

- ## Intégrité des données

Les bases de données permettent de stocker, de récupérer et d'analyser efficacement des données. À mesure que le volume des données collectées augmente et que les données deviennent plus sensibles, il est important que les professionnels de la cybersécurité protègent les bases de données toujours plus nombreuses. Une base de données est un système de classement électronique. L'intégrité des données fait référence à l'exactitude, à la cohérence et à la fiabilité des données stockées dans une base de données. La responsabilité de l'intégrité des données revient aux ingénieurs, aux développeurs et à la direction de l'entreprise.

Les quatre contraintes ou règles d'intégrité des données sont les suivantes :

- Intégrité de l'entité : toutes les lignes doivent être associées à un identifiant unique appelé clé primaire.
- Intégrité du domaine : toutes les données stockées dans une colonne doivent respecter un format et une définition identiques.
- Intégrité référentielle : les relations entre les tables doivent rester cohérentes. Un utilisateur ne peut donc pas supprimer un enregistrement lié à un autre.
- Intégrité définie par l'utilisateur : ensemble de règles définies par un utilisateur et n'appartenant à aucune des autres catégories. Par exemple, un client passe une nouvelle commande.

L'utilisateur vérifie d'abord s'il s'agit d'un nouveau client. Si c'est le cas, il l'ajoute dans la table des clients.
Pour s'assurer que les utilisateurs saisissent des données correctes dans un système, plusieurs contrôles peuvent être mis en œuvre. Il faut établir des règles pour les vérifications de base, notamment :

- La saisie obligatoire garantit qu'un champ spécifique contient des données.
- Les masques de saisie empêchent les utilisateurs de saisir des données non valides ou permettent de garantir une saisie cohérente (un numéro de téléphone, par exemple).
- Montants positifs en euro.
- Avec un contrôle des plages de données, vous avez la garantie que l'utilisateur saisit des données dans une plage bien précise (une date de naissance saisie sous la forme 18-01-1820, par exemple).
- Approbation obligatoire par une deuxième personne (si un employé de banque reçoit une demande de dépôt ou de retrait supérieure à une valeur spécifiée, l'approbation par une deuxième ou une troisième personne est requise).
- Déclencheur du nombre maximum d'enregistrements modifiés (si le nombre d'enregistrements modifiés dépasse une valeur prédéfinie sur une période donnée, l'utilisateur est bloqué jusqu'à ce qu'un supérieur détermine la légitimité des transactions).
- Déclencheur d'activités inhabituelles (un système se verrouille lorsqu'il identifie une activité inhabituelle).

Une règle de validation vérifie que les données respectent les paramètres définis par le concepteur de la base de données. Une règle de validation permet de s'assurer de l'exhaustivité, de l'exactitude et de la cohérence des données. Voici quelques critères utilisés dans une règle de validation :

- La taille : vérifie le nombre de caractères dans un élément de données.
- Le format : vérifie que les données respectent un format spécifié.
- La cohérence : s'assure de la cohérence des codes contenus dans les éléments de données.
- La plage : vérifie que les données sont comprises entre une valeur minimale et une valeur maximale.
- Le chiffre de contrôle : fournit un calcul supplémentaire pour générer un chiffre de contrôle en vue de la détection d'erreurs.

- **Validation des bases de données**

La validation du type de données est la forme la plus simple de validation des données. Elle vérifie qu'un utilisateur saisit les données conformément au type des caractères attendus. Un numéro de téléphone, par exemple, ne doit pas contenir de caractères alphabétiques. Les bases de données acceptent trois types de données : nombres entiers, chaînes et nombres décimaux.

Le contrôle du processus de saisie de données est l'un des aspects les plus vulnérables de la gestion de l'intégrité des bases de données. De nombreuses attaques communes ciblent une base de données et y insèrent des données sous un format incorrect. Cela peut se traduire par une déstabilisation de l'application, une panne ou la divulgation d'un nombre trop important d'informations au hacker. Les hackers lancent des attaques de saisie automatisées.

La détection des anomalies consiste à identifier des schémas de données qui ne respectent pas le comportement attendu. a vérification et la détection des anomalies constituent une contre-mesure ou un moyen de protection important pour identifier les fraudes. La détection des anomalies de base de données permet d'identifier les fraudes à l'assurance et à la carte bancaire. Cela permet également de protéger les données contre une modification ou une destruction massive. La vérification des anomalies nécessite des modifications ou des demandes de données de vérification lorsqu'un système détecte des schémas inhabituels ou inattendus. Il peut s'agir, par exemple, de deux transactions effectuées en peu de temps avec une même carte de crédit, mais à des endroits très éloignés l'un de l'autre.

• Exigences en matière d'intégrité des bases de données

Une base de données est un système de classement électronique. Il est fondamental de bien classer les données dans la base de données pour préserver leur fiabilité et leur utilité. Une base de données se compose de tables, d'enregistrements, de champs et de données. Pour préserver l'intégrité du système de classement de la base de données, les utilisateurs doivent respecter certaines règles. L'intégrité de l'entité est une règle d'intégrité qui stipule que chaque table doit posséder une clé principale et que la ou les colonnes qui correspondent à la clé principale doivent être uniques et non nulles. Dans une base de données, « nul » désigne des valeurs manquantes ou inconnues. L'intégrité de l'entité permet une organisation correcte des données de cet enregistrement.

La relation entre des tables ou systèmes de classement différents est une autre notion importante. Les clés étrangères constituent la base de l'intégrité référentielle. Une clé étrangère dans une table fait référence à une clé principale dans une autre table. La clé principale identifie de manière unique des entités (lignes) dans la table. L'intégrité référentielle préserve l'intégrité des clés étrangères.