LIVRE BLANC

CRITÈRES CLÉS DANS LE CHOIX D'UNE SOLUTION DE WEB APPLICATION ET API PROTECTION (WAAP)

UBIKA


UBIKA

SOMMAIRE

AVANT-PROPOS

Au sein d'un univers corporate de plus en plus digital, les applications web jouent un rôle toujours plus important. Dans chaque grande entreprise, ce sont plus de 100 applications web qui sont utilisées quotidiennement. Actuellement, les attaques ont fait de l'infrastructure applicative une cible privilégiée car considérée comme plus vulnérable. Il en résulte que les applications web sont la première cause des piratages de données.

Les cybercriminels exploitent délibérément les points faibles potentiels des applications web et ne sont que rarement détectés par les systèmes classiques de sécurité réseau et de prévention des intrusions. Qui plus est, même les pare-feu de nouvelle génération ne suffisent plus à juguler la menace.

Afin de prévenir ces attaques, les entreprises doivent identifier les failles applicatives au prisme de tests de type SAST et DAST. Parallèlement, elles doivent déployer une solution de type pare-feu applicatif web (WAAP pour Web Application Firewall et API Protection). Pour autant, sur le terrain, les entreprises qui déploient un WAAP sont confrontées à des contraintes :

• Recrudescence des faux positifs :
Les dispositifs de protection WAAP génèrent souvent des faux positifs qui doivent être analysés. Les WAAP n'offrent pas de solutions simples pour gérer ces faux positifs, ce qui génère des coûts élevés, mais également des risques, puisque les administrateurs ont tendance à abaisser le niveau de sécurité pour déclencher moins de faux positifs. Enfin, ces faux positifs peuvent restreindre l'accès des utilisateurs légitimes à leurs applications web, un vrai frein à la croissance et aux opérations de l'entreprise.

• Mises à jour du WAAP :
Il est important de maintenir le WAAP à jour pour pérenniser la sécurité. Sauf que cette opération s'avère souvent fastidieuse. Puisque le WAAP est essentiel, les mises à jour ne doivent pas paralyser les applications déjà protégées, ce qui explique que les clients redoutent souvent cette procédure. Mais plus ils tardent à le faire, plus ils risquent de subir l'obsolescence de leurs composants (OpenSSL, TLS, OS…) et donc perdre de précieuses fonctionnalités.

• Assurer l'évolutivité du WAAP dans le cloud n'est pas toujours simple :
Parfois, le WAAP subit un niveau de charges tel qu'il devient difficile de gérer l'augmentation du trafic web. Un pic de trafic inattendu peut obérer la capacité à répondre à toutes les requêtes des utilisateurs. Dans ce cas, si le WAAP ne s'adapte pas simplement, les clients devront déployer des solutions complémentaires, ce qui creuse les coûts.

• Automatisation et intégration dans les cycles de développement :
Le virage vers DevSecOps permet d'automatiser l'ensemble du cycle de développement logiciel tout en relevant le niveau de sécurité des applications conçues via ce processus. L'un des objectifs consiste à intégrer en amont le WAAP dans le pipeline CI / CD de conception d'applications Web, afin d'accompagner les développeurs. Cependant, tous les WAAP ne sont pas conçus pour accompagner cette démarche.

Avant d'investir dans un WAAP, vous devez procéder à une analyse systématique de ces divers facteurs. Cette ligne de défense étant cruciale à la sécurité de votre entreprise, le WAAP mérite qu'on y consacre du temps et des efforts. **Ce livre blanc examine les principaux critères décisionnels pour retenir un WAAP robuste et adapté à vos problématiques, au travers de 12 questions que les décisionnaires doivent se poser.**

1. Votre environnement a-t-il besoin d'une protection spécifique pour vos ressources Web ?

Les pare-feu de nouvelle génération (NGFW), les systèmes de prévention d'intrusion (IPS) et les outils RASP (protection autonome des applications en production) ne fournissent pas une protection suffisante pour les applications et les services web.

1.1 OS / logiciels / sécurité réseau : IPS

Un IPS fonctionne surtout au niveau du réseau et des sessions. Ces systèmes sont juste capables d'identifier des vulnérabilités connues, sur la base de règles standards et de signatures d'attaque. Par conséquent, les hackers peuvent les contourner en modifiant légèrement les signatures. Les IPS n'identifient pas la plupart des attaques qui ciblent la couche 7 applicative et génèrent trop de faux positifs. Même si certains IPS sont capables de déchiffrer le trafic SSL, la plupart d'entre eux ne disposent pas d'un processus de confidentialité suffisant pour gérer les certificats en backend et n'identifient qu'un nombre très limité de requêtes SQL ou d'attaques XSS (cross-site scripting).

1.2 Sécurité réseau : NGFW

Un NGFW correspond à la 3e génération de pare-feu avec des fonctionnalités IPS, une visibilité sur les applications et la possibilité de prendre en compte les droits d'accès des utilisateurs en communiquant avec l'annuaire LDAP. Il surveille le trafic sortant et entrant non chiffré (20 % du trafic mondial) et protège les utilisateurs. De nombreux fournisseurs de NGFW affirment que leurs produits prennent en compte les applications. Ces NGFW ne permettent de contrôler que le trafic applicatif clair, par exemple en autorisant l'accès à Facebook, mais en bloquant le chat ou en n'offrant qu'un accès sélectif aux vidéos. Un NGFW permet aux administrateurs réseau de déterminer la bande passante allouée aux applications web, comme eBay ou Gmail, et de limiter les besoins en bande passante. Cependant, étant donné qu'il s'agit principalement d'un proxy direct, le NGFW ne dispose pas de dispositifs spécifiques de protection des API.

1.3 Auto-protection de l'application runtime : RASP

Un RASP analyse le contexte et le comportement d'une application en production pour détecter les attaques. Il peut différencier les requêtes légitimes et malveillantes, pour neutraliser les menaces réelles avec peu de faux positifs. Une solution RASP n'est pas indépendante des technologies, elle doit être compatible avec les langages et technologies utilisés dans votre entreprise. Cet outil présent au niveau du backend de l'application, peut porter préjudice à sa fiabilité, ses performances et sa stabilité. En outre, les frameworks de conformité tels que PCI DSS ne reconnaissent pas, le RASP en tant que mesure suffisante de sécurité applicative lorsque cet outil est déployé seul.

1.4 Sécurité des applications web et APIs : WAAP

Pour se prémunir des nouvelles attaques ciblant la couche 7, une solution de sécurité doit connaître de manière détaillée l'architecture d'une application web, son contexte, les utilisateurs et les sessions clients. C'est là que le WAAP entre en jeu. C'est un dispositif dédié dont l'objectif principal est de protéger les applications Web et les API. Déployé en tant que reverse proxy en amont de l'application web, il analyse le contenu de chaque requête entrante HTTP / HTTPS par rapport à leur comportement et leur logique, avant de les transmettre aux applications. Ainsi, ils peuvent détecter et prévenir les requêtes malveillantes inconnues ciblant les vulnérabilités du Top 10 OWASP, les attaques DoS ou DDoS, la falsification des cookies, etc.
Ils peuvent aussi assurer l'authentification des utilisateurs, le déchargement SSL, la répartition des charges des serveurs web et la gestion des faux positifs.
Ces quatre technologies se complètent et doivent être déployées conjointement pour renforcer le niveau de sécurité, selon vos exigences.

2. Le WAAP protège-t-il les applications web contre les principales attaques ? Et comment ?

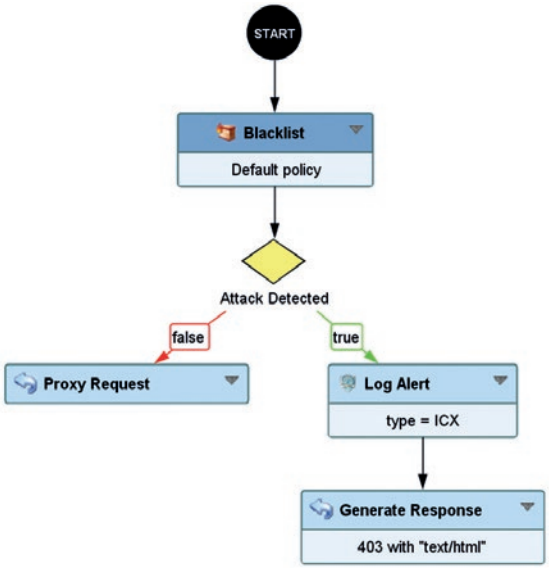
Chaque jour, les entreprises utilisent le web pour développer leur activité commerciale et assurer le fonctionnement de leur processus opérationnel. Elles doivent ainsi protéger leurs applications web des nouvelles menaces. Open Web Application Security Project (OWASP) est une communauté ouverte, regroupant des experts en sécurité applicative du monde entier, chacun partageant son expertise et collaborant pour identifier les faillesde sécurité les plus significatives des applications et services web. Son top 10 des vulnérabilités et risques les plus significatifs constitue un standard utilisé par la majorité des acteurs de la cybersécurité et sert de référence à nombre de réglementations (PCI DSS, NIS, GDPR, et HIPAA). Ci-dessous une source pour plus d'informations.

TOP 10 OWASP – 2022 :

A01 Contrôles d'accès défaillants
A02 Défaillances cryptographiques
A03 Injection
A04 Conception non sécurisée
A05 Mauvaise configuration de sécurité
A06 Composants vulnérables et obsolètes
A07 Identification et authentification de mauvaise qualité
A08 Manque d'intégrité des données et du logiciel
A09 Carence des systèmes de contrôle et de journalisation
A10 Falsification de requête côté serveur

SCHÉMA 1 :

Workflow



Voici un exemple qui décrit la technique d'attaque et le mécanisme de prévention des injections de commandes.
Lorsqu'une application Web envoie et exécute des commandes système malveillantes, le pirate peut en abuser pour exécuter d'autres commandes système et prendre le contrôle du serveur.

Considérons le cas suivant :

Requête du hacker : System (« whois google.fr | ls ; »);
Réponse : Applications Documents
Pictures Confidential
Solution : Les moteurs de détection UBIKA vérifient les données saisies par l'utilisateur pour un échappement de méta-caractères, afin de valider les données saisies et utiliser des commandes classiques qui fonctionnent sur les plateformes Windows, UNIX et Linux.

UBIKA WAAP Gateway est un outil idéal pour se protéger des attaques les plus courantes.
Il active par défaut différents moteurs de sécurité. Le moteur natif ICX utilise des politiques de sécurité composées d'une ou plusieurs règles. Ces moteurs de sécurité utilisent un workflow pour exécuter la séquence d'opérations (schéma 1).
Cette approche est matérialisée par un module dans le logiciel d'administration du WAAP. Qui permet aux administrateurs peu expérimentés en matière de sécurité web de configurer un WAAP.

Parallèlement aux moteurs standards, il est également possible d'activer, si nécessaire, des moteurs sophistiqués utilisant des méthodes de détection complémentaires (analyse grammaticale, analyse heuristique). UBIKA WAAP Gateway permet d'analyser des données d'entrée XML et JSON, afin d'appliquer les différents moteurs de sécurité et ainsi détecter les tentatives d'infection au niveau des API (SOAP / REST).
Les différents moteurs de sécurité (moteurs standards, liste de scoring ou moteur d'analyse des XSS) préviennent les attaques de cross-site scripting. Ils permettent de détecter les attaques concernant des vulnérabilités connues ou inconnues, à l'instar des attaques zero day. Les vulnérabilités identifiées donnent lieu à des signatures de protection.

3. Le WAAP peut-il arbitrer entre faux positifs et faux négatifs ?

Pour simplifier les phases de déploiement et de configuration, vous devez saisir les spécificités des applications et les types d'attaque. Vous ne pouvez dépendre uniquement de larges bases de données indexant des schémas comportementaux basés sur des expressions régulières. Cette approche a tendance à générer trop de faux positifs. De plus, des règles plus strictes génèrent davantage de faux positifs tandis que des règles plus indulgentes aboutiront à de faux-négatifs. Dans ce contexte, un éditeur de solution WAAP devra trouver l'arbitrage idéal entre les deux.

D'autre part, il est important de comprendre les spécificités des technologies web les plus récentes telles que les services JSON6 ou REST7. Ce sera beaucoup plus facile si une telle syntaxe est validée par RFC, normalisée aux paires http « paramètre=valeur » par défaut et ensuite envoyée via les moteurs de sécurité standards.

Au-delà de ces aides technologiques, un mode de configuration logique est important pour comprendre comment spécifier une politique de sécurité précise. Le concept de workflow et la configuration qui en résulte sont basés sur le flux réel d'une requête vers l'application et de retour vers le client.

SCHÉMA 2 :
translation JSON

JSON	{ "login": " 1\' OR 1=1# ", "password": " 1 \' OR 1=1# " }
HTTP	login=1'%20OR%201=1# &password=1 '%20OR%201=1#

3.1 Requêtes bloquées : une gestion pertinente des faux positifs

Outre la configuration de sécurité en elle-même, il est important de fournir un moyen facile d'ajuster la configuration vis-à-vis des requêtes bloquées. Cependant, plus un modèle de sécurité est précis dans l'identification d'une attaque, plus le taux de faux positifs est faible. Il est important de savoir à quelle vitesse il est possible de créer une exception, mais aussi de connaître la précision de cette exception.

La fonction de résolution automatique (schéma 3) de UBIKA WAAP Gateway crée automatiquement des exceptions et permet de traiter un à plusieurs centaines de logs en une seule fois.

Le processus de génération d'une exception devrait se résumer à deux étapes :

a) sélectionner une demande bloquée dans les journaux d'alertes,

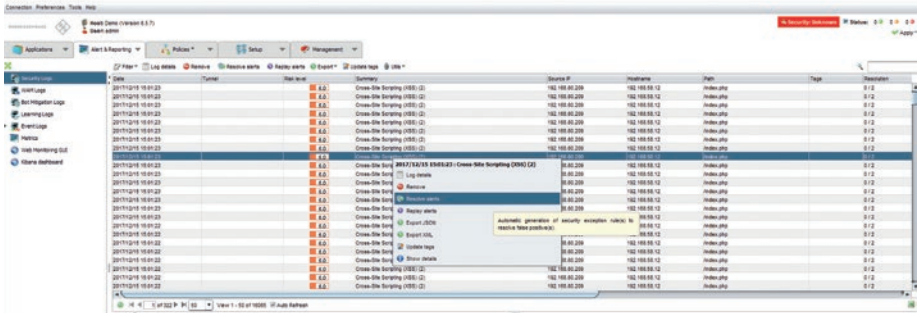
b) cliquer sur un bouton de résolution.

Par conséquent, la configuration crée toujours une exception avec ces prérequis :

- Une règle simple
- Une URL dédiée
- La partie de la requête à l'origine de l'exception

Si une requête est bloquée par erreur, elle sera ajoutée à la liste blanche et sera donc autorisée.

SCHÉMA 3 :
résolution automatique



4. Le WAAP se déploie-t-il facilement et rapidement sur toutes les plate-formes ?

Un WAAP doit être interopérable vis-à-vis des plateformes et donc ne pas dépendre d'une seule d'entre elles. De cette façon, il devient inutile de devoir tout recommencer lorsqu'une application est migrée. Le WAAP doit vous proposer une plateforme indépendante et avec la même technologie pour sécuriser vos API, quelle que soit la méthode de déploiement ou le nombre de fournisseurs d'infrastructure avec lesquels vous collaborez.

4.1 On-premises / sur site

Le WAAP se déploie sur les sites d'entreprise sous forme d'appliance matérielle ou de machine virtuelle pour protéger les applications critiques présentes au sein d'un data center d'entreprise. Dans ce cas, l'opérateur du data center doit s'impliquer les opérations de configuration et les mises à jour logicielles. L'entreprise peut également disposer d'un cloud privé sur site et sera, dans ce cas, en mesure de contrôler les aspects d'orchestration, d'évolutivité, etc. Cependant, un WAAP sur site subit les limites du cloud privé.

4.2 Dans le cloud

Le WAAP est proposé via des marketplaces cloud qui permettent de déployer une instance. Ce modèle allège votre investissement initial et bénéficie des mises à jour de sécurité les plus récentes. Un modèle pertinent consiste à associer un mode BYOL (Bring Your Own License), pour les instances opérationnelles en 24 / 7, avec un mode Pay As You Go (PAYG) qui encourage l'évolutivité. Les orchestrateurs cloud contrôlent la montée en charge ou la réduction des capacités, à partir de scripts Terraform. Ceci permet de déployer une approche IaC (Infrastructure as Code) qui automatise la totalité du processus.

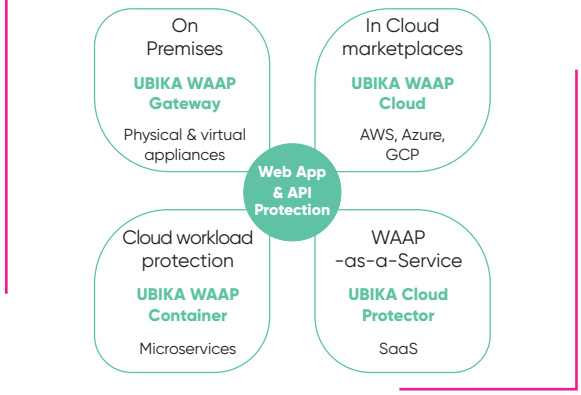
4.3 SaaS (WAAP-as-a-Service)

Le WAAP est parfaitement adapté à un modèle SaaS, avec un déploiement et une maintenance assurés par l'éditeur, sans aucune opération d'installation par le client. Simple à déployer et à configurer, le client ne s'occupe que de rediriger le trafic applicatif vers l'infrastructure hébergée. L'évolutivité, les mises à jour logicielles et le monitoring de la plateforme sont gérés par le fournisseur. Cette offre est souvent associée à une protection contre les DDoS pour prévenir les attaques volumétriques.

4.4 Cloud workload protection platform (CWPP)

Le CWPP se déploie au plus proche de l'application, les moteurs de sécurité étant hébergé dans un conteneur. Ceci permet de renchérir ou de diminuer les capacités selon les besoins de l'application. La sécurité, mise à jour automatiquement, est toujours adaptée à la version de l'application protégée. Ce modèle est conçu pour les équipes DevOps et DevSecOps qui doivent intégrer la sécurité des applications dans le pipeline CI / CD, et automatiser la phase de conception de leur cycle de développement applicatif. Il s'agit d'une approche Security as Code qui vient en complément du concept d'IaC. UBIKA propose l'ensemble de ces produits et technologies, sous différents formats, selon les besoins des utilisateurs. Ils sont compatibles avec des technologies comme Terraform, Cloud Formation et Cloud Watch. Le WAAP est disponible sur les marketplaces publiques d'AWS, Azure & GCP. La solution est également adaptée pour une exploitation au sein d'un cloud privé. Miser sur une approche interopérable est essentiel car il est important de déployer une solution cohérente sur l'ensemble des périmètres protégés, et pilotée à partir d'une seule console d'administration.

SCHÉMA 4 : Solutions WAAP d' UBIKA



5. Le WAAP offre-t-il des fonctions de listes blanches et noires pour le contrôle applicatif ?

De manière traditionnelle, les équipes de sécurité ont davantage fait appel aux listes noires pour sécuriser leurs environnements.
En effet, gérer les applications web se révélait bien trop complexe avec des listes blanches. Il y a une décennie, les fournisseurs de solutions n'avaient pas la maturité pour tout automatiser.
Ce qui n'est plus le cas aujourd'hui. De plus, les entreprises souhaitaient que leurs systèmes soient accessibles au plus grand nombre de personnes, pour justement étendre leur base de clients. Seuls les utilisateurs représentant une menace étaient bloqués. Avec l'avènement des API et de la méthodologie DevSecOps, la méthode des listes blanches a gagné en capacité pour protéger les applications web, compte tenu des nombreuses innovations dédiées à la sécurité des applications web.

5.1 Listes blanches et noires : les fondamentaux

Deux principaux modèles de sécurité sont utilisés par les WAAP modernes : le modèle de sécurité négatif qui intègre une liste noire répertoriant les schémas d'attaque et le modèle positif, basé sur une liste blanche. Une liste noire autorise toutes les requêtes, sauf celles qui correspondent à des schémas prédéfinis ; une liste blanche neutralise toutes les requêtes prohibées par ses règles. Ces deux méthodes sont complémentaires.

5.1.1 Modèle de sécurité négatif : les listes noires

Il est impossible de créer une liste de tous les comportements / schémas possibles d'une attaque. La façon la plus efficace d'utiliser une liste noire est de travailler avec des comportements génériques, plutôt que d'en définir un pour chaque vulnérabilité.
Cette technique permet de neutraliser les attaques zero-day et de renforcer les performances. Les vulnérabilités qui ne pourront être détectées à l'aune des comportements génériques inciteront à spécifier un comportement qui leur est propre. La méthode des listes noires peut aboutir à un taux élevé de faux positifs. Elle doit être associée à un système robuste de gestion de ces faux positifs qui permettra un véritable gain de temps auprès des administrateurs. UBIKA WAAP utilise le service de réputation IP Webroot en complément de son approche de blacklist.
Grâce à une veille en temps réel sur les menaces, les clients sont protégés efficacement contre les menaces associées à des adresses IP spécifiques.

5.1.2 Modèle positif de sécurité : listes blanches

L'utilisation de listes blanches est une façon simple et sécurisée pour protéger les petites applications statiques.
Vous devez définir, pour chaque URL, un paramètre et une valeur possible. La tâche s'annonce donc colossale pour les applications d'envergure qui comptent beaucoup d'URLs et de paramètres. Il s'agit, en effet, d'adapter la liste blanche à chaque changement de l'application. Peut-être avez-vous déjà entendu parler des processus d'auto-apprentissage, qui sont sensés intégrer automatiquement de nouvelles demandes à une liste blanche.
Sans intelligence artificielle, le logiciel ne peut décider de la légitimité d'une requête. De plus, les hackers expérimentés sont capables de comprendre et d'exploiter ces processus d'auto-apprentissage.

Les API doivent être gérées par un modèle positif de sécurité. En créant une API, les utilisateurs savent le type de données attendu par chaque endpoint. Les développeurs créent des fichiers Swagger ou OpenAPI qui décrivent le comportement de l'API.
Une technologie pertinente de liste blanche doit être capable d'utiliser ces formats standards et de les appliquer.

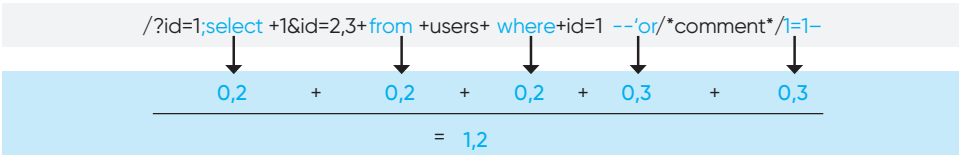
5.2 Techniques avancées : modèle basé sur le scoring

UBIKA WAAP gère tant les listes noires que blanches. De plus, la solution utilise un modèle de scoring (modèle de notation) qui attribue une valeur aux différents ensembles de données soumises au serveur. Le système additionne les valeurs, puis effectue un calcul et compare ce résultat à un seuil prédéfini. Si ce seuil est atteint, le système définit ce trafic comme étant malveillant et la requête est abandonnée (schéma 5).

Les coefficients de pondération dans le modèle de scoring sont déterminés par machine learning, à partir du trafic réel, pour affiner le modèle à l'aide de mots-clés et pondérations les plus pertinents. Ce mécanisme peut être comparé à celui utilisé par la plupart des logiciels anti-spam. De toute évidence, l'aspect le plus critique de sa mise en œuvre est de définir des valeurs appropriées. En effet, des valeurs erronées entraîneront une détection plus élevée de faux positifs ou de faux négatifs. En revanche, une fois ces valeurs testées et validées, il ne devrait pas être nécessaire de les mettre à jour, sauf si de nouvelles techniques d'attaque sont identifiées. Le modèle de scoring a prouvé son efficacité en neutralisant plus de 85 % des nouvelles attaques, sans besoin de mises à jour, ni phase d'apprentissage spécifique. Ce mécanisme obligatoire pallie les carences des modèles actuels.

Le scoring de réputation de l'utilisateur est une autre fonction intelligente de protection du pare-feu applicatif web qui identifie les visiteurs, même en amont de leur authentification, et leur affecte des attributs de données (score, compteurs ou propriétés). Cette information, lorsque suivie, est disponible pour chaque requête de votre workflow. Si la requête de l'utilisateur est légitime, le score progresse. Si elle est malveillante, le score baisse. Des seuils sont définis avec une action spécifique à chaque franchissement de seuil. À titre d'exemple, si un score baisse de 100 à 50, la requête est temporairement neutralisée. Si le score baisse encore à 30, une page d'authentification est activée pour valider l'utilisateur.

SCHÉMA 5 :
modèle de scoring



5.3 Techniques avancées : moteurs de détection avancée des menaces

Les attaques deviennent de plus en plus sophistiquées avec le temps. Les filtres linéaires basés sur des schémas, comme les listes noires ou les mécanismes de notation, ne seront pas en mesure de neutraliser toutes les attaques modernes. **UBIKA a investi dans le développement de nouveaux moteurs de sécurité capables d'identifier ces attaques avancées**. Il s'agit des technologies décrites ci-dessous. Générer une expression régulière lourde et complexe pour filtrer des technologies modernes comme JSON est nécessaire, mais insuffisant. Il serait plus logique de « décrypter » la requête JSON en paires http standards de type « paramètre=valeur » puis de la relayer via des moteurs de sécurité par défaut. Il en va de même pour les techniques d'injection SQL, qui peuvent en fait n'utiliser qu'un seul mot. Cependant, bloquer l'utilisation d'un mot spécifique entraînera de nombreux faux positifs. Il faut définir des mots-clés prédéterminés pour les instructions SQL réelles afin de vérifier ou falsifier une injection SQL. Les moteurs avancés de détection des menaces ne se limitent pas à ceux mentionnés précédemment. Ils intègrent également des technologies telles que la sécurité HTML, la séparation de réponse, la protection contre l'injection de scripts ou le calcul arithmétique.

6. Le WAAP est-il capable de gérer les pics de trafic ?

Il y a encore quelques années, à l'ère du pré-cloud, l'évolutivité imposait d'acheter de nouvelles licences ou de déployer de nouvelles appliances matérielles ou virtuelles. Ce processus pouvait prendre jusqu'à un mois ou davantage, obérant ainsi tout scaling rapide lors des pics de trafic saisonniers. Lorsque les limites du scaling vertical étaient atteintes, les entreprises déployaient de nouvelles machines pour assurer un scaling horizontal.

Le scaling est un facteur essentiel à prendre en compte pour choisir la bonne solution WAAP. En phase de croissance, les entreprises ont besoin d'une méthode simple pour gérer la croissance des visiteurs de leurs sites web. La solution WAAP doit donc pouvoir évoluer pour gérer davantage de trafic web, et répondre à des charges qui progressent. Le modèle tarifaire doit également être flexible pour permettre une facturation à l'utilisation et une consommation que de ce qui est strictement nécessaire.

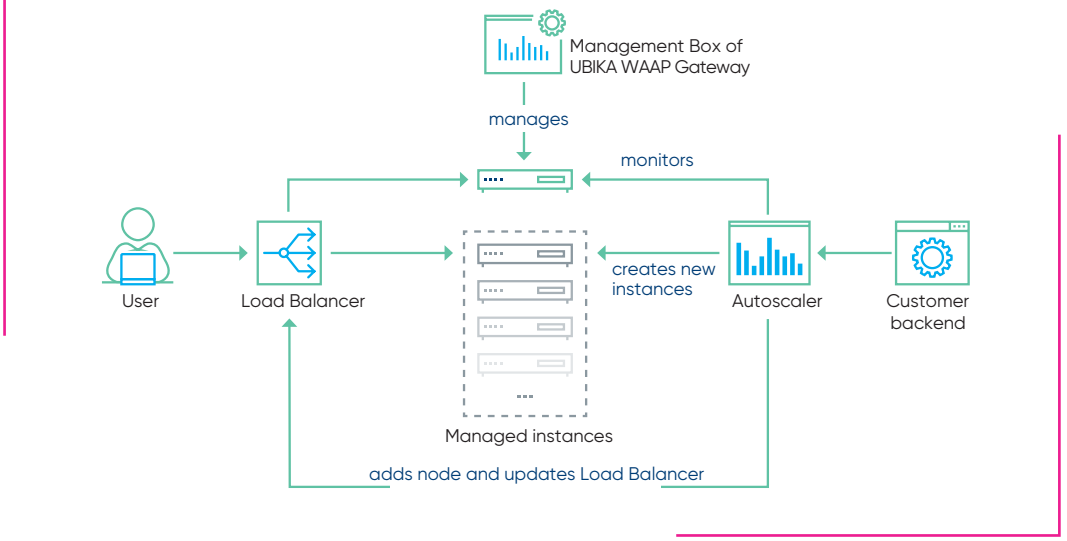
Les innovations en matière de WAAP tirent parti de micro-services hébergés dans des conteneurs. Ceci rend le scaling plus précis, en ne faisant évoluer que les services requérant des ressources supplémentaires.

UBIKA WAAP Gateway (on-premises) et UBIKA WAAP Cloud (version cloud public) sont particulièrement évolutives. Les solutions activent automatiquement de nouvelles instances grâce à une orchestration basée sur des API et des scripts Terraform, lors des périodes de trafic important. En cas de baisse de trafic, les instances sous-utilisées sont désactivées. L'avantage principal pour le client est lié à l'automatisation. Ils peuvent déployer une vraie approche IaC.

UBIKA Cloud Protector (WAAP fourni en mode SaaS) propose également ces avantages.
En cas de recrudescence du trafic web, la solution évolue pour répondre aux besoins supplémentaires.

UBIKA WAAP Container (solution DevSecOps) assure une évolutivité accélérée sur toutes les plateformes. La couche de sécurité est déployée en tant que micro-WAAP au sein de l'application, assurant ainsi aux utilisateurs un dimensionnement à la hausse ou à la baisse selon les besoins de l'application, grâce à un orchestrateur Kubernetes ou Docker.
Ceci diminue les coûts des ressources et améliore le retour sur investissement.

SCHÉMA 6 :
Évolutivité sans limite de UBIKA WAAP Gateway

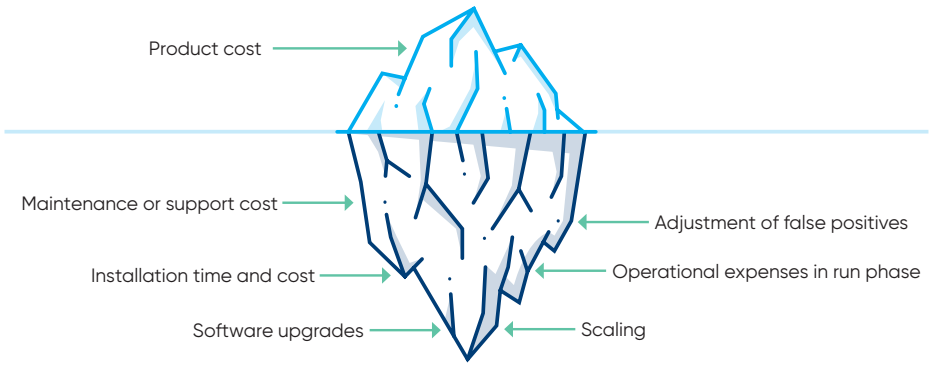


7. Le WAAP permet-t-il de maîtriser le coût total de possession ?

Pour assurer un ROI performant pour votre WAAP, vous devez prendre en compte deux facteurs. Le premier est que le WAAP doit bénéficier d'un déploiement rapide et d'une maintenance simple des règles de sécurité. D'autre part, le WAAP doit pouvoir réduire le nombre de faux positifs de manière significative.

L'automatisation des workflows peut être employée pour une maintenance efficace des règles de sécurité. Ceci est essentiel pour alléger le coût total de possession (TCO). Mais aussi pour accompagner la transformation digitale et le développement d'applications cloud-native, en permettant aux clients d'adopter un mode No-Ops.
Ceci fait partie de la transformation DevOps, donnant lieu à une absence de tâches manuelles pour déployer une application et gérer sa montée en charge. Une automatisation pertinente avec des approches performantes comme IaC (Terraform, Cloud Formation d'AWS) et Configuration as Code offrira la capacité à concevoir et déployer rapidement les logiciels, tout en réduisant les coûts d'exploitation lors de la mise en production.

SCHÉMA 7 :
coût total de possession



Un workflow graphique simplifie tous les cas d'utilisation et permet de réagir aux éléments de contexte. Le workflow générique est un vrai levier de simplification. Il devient possible d'intégrer des éléments de contexte grâce à des paramètres de workflow. La description du contexte permet d'intégrer les spécificités de sécurité d'une application dans le workflow.
Avec UBIKA WAAP Container, notre objectif est d'apporter des informations contextuelles au conteneur hébergeant le micro-WAAP.

Grâce à cette description de contexte, le nombre de faux positifs est moindre, des règles spécifiques de sécurité sont appliquées et les performances sont au rendez-vous.

UBIKA Cloud Protector présente un faible investissement initial et des charges d'exploitation modérées (installation et remplacement de matériel, maintenance, mises à jour logicielles).
La solution s'adapte en temps réel aux pics de trafic subis par votre application et traite les vulnérabilités du Top 10 OWASP.
De plus, la tarification sous forme d'abonnement est un vrai levier de flexibilité.

8. Le WAAP s'intègre-t-il dans votre approche DevSecOps ?

Un élément important pour accélérer la transformation digitale est de concevoir une stratégie DevSecOps offrant de la visibilité à différentes parties prenantes : développeurs, opérationnels et professionnels de la sécurité. Les développeurs doivent donc disposer des bons outils pour configurer leur stratégie de sécurité applicative. L'intégration de la sécurité dans le code logiciel est devenue inévitable compte tenu d'un développement logiciel toujours plus rapide au sein des entreprises.

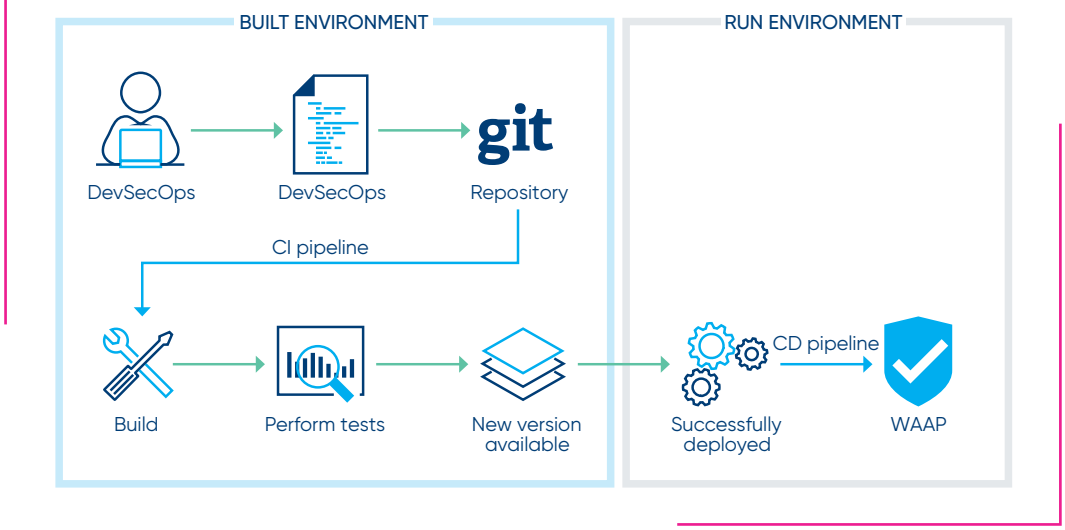
Pour sécuriser les applications cloud-native, il est essentiel de leur appliquer une approche DevSecOps qui présente tous les avantages d'une approche DevOps tout en pérennisant les bonnes pratiques de sécurité en place.
La culture DevSecOps est également l'opportunité d'améliorer la gestion des faux positifs. Elle tire parti du processus de déploiement continu pour réduire le nombre de faux positifs et les rendre plus gérables par l'équipe en charge de la qualité, tout au long du processus de développement.

Aujourd'hui, les équipes DevSecOps sont à la recherche de solutions intelligentes qui intègreraient un volet sécurité dès la phase amont d'un projet applicatif, tout en proposant une technologie simple. Pour garantir que les fonctions avancées de sécurité du WAAP sont utilisées à leur plein potentiel, il est important de mettre ce WAAP à disposition des développeurs, de manière intégrée au pipeline DevOps.

Ainsi, la solution DevSecOps doit être totalement intégrée avec les outils, langages et concepts du pipeline CI / CD, mais aussi automatisée de bout-en-bout. Elle doit pouvoir prendre en charge les configurations et règles des différentes parties prenantes. Un WAAP pertinent facilitera cette intégration.

Dans un contexte Security as Code, UBIKA WAAP Container est un excellent moyen de mettre en oeuvre une approche DevSecOps qui apporte la sécurité aux équipes de développement. La solution de sécurité est déployée au sein du pipeline CI / CD avec les outils existants qui simplifient la collaboration. Elle offre un accès aux bons outils, ceux qui vous feront penser à la sécurité dès la phase de conception et non en aval, en environnement de production.

SCHÉMA 8 :
Pipeline CI / CD pour DevSecOps



9. Le WAAP assure-t-il la haute disponibilité et les performances des applications ?

Les entreprises actuelles, quelle que soit leur envergure, sont dépendantes d'Internet pour le bon fonctionnement de leur activité. Les indisponibilités, qu'elles soient programmées ou pas, induisent des pertes substantielles, en clients et en revenu. Pour les entreprises, il est essentiel de préserver la haute disponibilité, en minimisant les indisponibilités systèmes et les interruptions de services au niveau des applications, surtout si celles-ci sont critiques à leurs opérations.

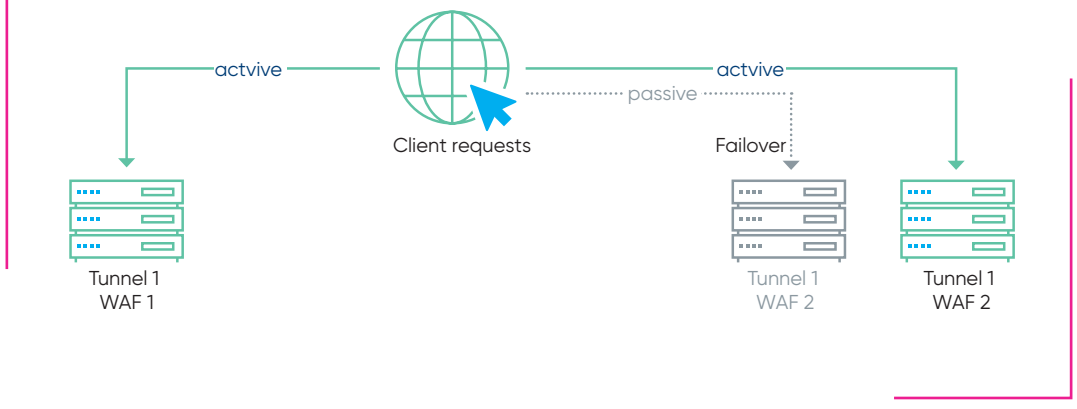
La haute disponibilité permet d'optimiser les performances opérationnelles de l'entreprise, sa productivité et sa réputation de manière générale. **Un WAAP pertinent propose une infrastructure évolutive qui permet aux clients de juguler les attaques ou de s'adapter à un contexte de trafic web important.**
Ceci valide que l'éditeur du WAAP accorde une réelle importance à votre activité métier.

La haute disponibilité se décline en deux modes : actif-actif et actif-passif. Dans un mode actif-actif, comme celui d'un répartiteur de charges, le trafic du tunnel en mode actif-actif se répartit sur plusieurs WAAP pour éviter toute charge excessive sur l'un d'entre eux. Lorsqu'un WAAP est indisponible, le trafic est réorienté vers les tunnels des autres WAAP, ce qui améliore les performances et les temps de réponse.

Avec le mode actif-passif, une seule instance est active tandis que l'autre est en standby. Lors d'un dysfonctionnement du tunnel primaire, le trafic est basculé vers le second tunnel.
Lorsque le tunnel primaire est de nouveau opérationnel, il récupère ce trafic. Ce mode opératoire prévient toute interruption liée à un crash système.

UBIKA WAAP Gateway vous offre la possibilité de déployer ces deux modes. Cependant, au sein d'un écosystème cloud, ces modes de haute disponibilité ne s'appliquent pas. Dans ce scénario, UBIKA WAAP CloudI utilise des répartiteurs de charges cloud-native pour piloter le trafic entre les instances.
D'autre part, les contenus statiques comme les images sont mis en cache pour être immédiatement mis à disposition des clients, ce qui accélère les temps de chargement et améliore les performances.

SCHÉMA 9 :
modes de haute disponibilité



10. Le WAAP s'intègre-t-il au sein de l'écosystème applicatif existant ?

Un WAAP doit gérer les fonctionnalités de sécurité de votre application, à l'image de l'authentification et SSO, ainsi que vos processus de sécurité en place comme les tests d'intrusion ou les programmes de pen testing ou bug bounty.

10.1 Gestion des accès web : authentification & Single Sign-On

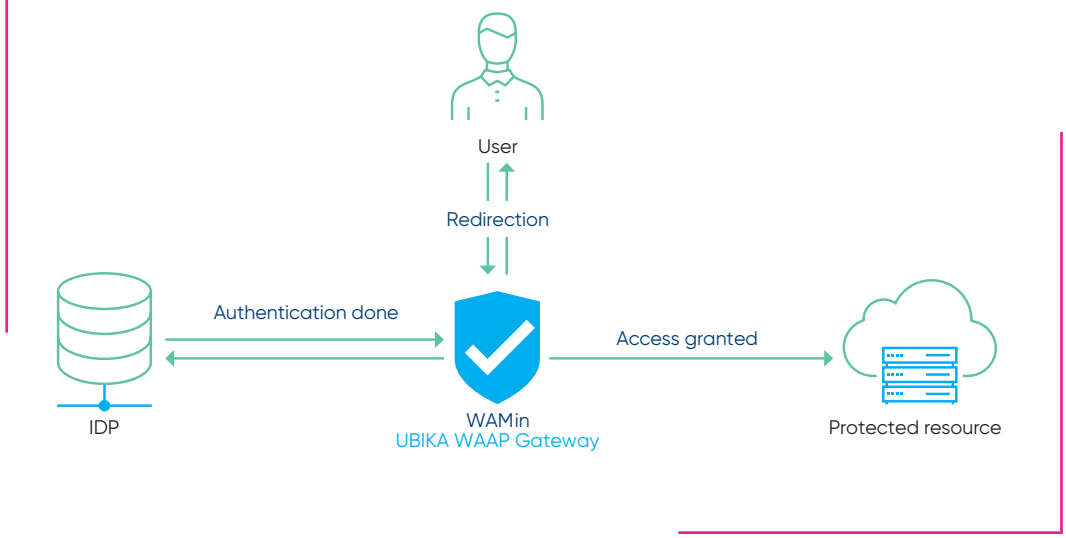
Un autre élément important d'un pare-feu applicatif Web (WAAP) de nouvelle génération est d'offrir une gestion complète de l'accès Web. Il doit gérer l'authentification des utilisateurs et le Single Sign-On pour simplifier l'accès aux applications web protégées. **La pré-authentification des utilisateurs est un prérequis courant et important pour élever le niveau de sécurité.** L'utilisation de services d'authentification au niveau WAAP empêche d'accéder à l'application avant d'être authentifié.

Les WAAP proposent différentes méthodes d'authentification, de l'authentification de base jusqu'aux méthodes d'authentification NTML ou SAML (schéma 10). **Web Access Manager (WAM) est un module optionnel disponible pour UBIKA WAAP Gateway et UBIKA WAAP Cloud qui applique la politique d'authentification dans la console de gestion aux applications protégées par le WAAP.** Il permet de regrouper les méthodes d'authentification des applications derrière une authentification forte unique présentée aux clients. À sa première requête, l'utilisateur reçoit un formulaire d'authentification externe (en périmètre du SI). Une fois l'authentification réussie, Web Access Manager traite automatiquement l'authentification des applications en fonction des informations d'identification stockées dans son annuaire interne ou dans l'annuaire d'entreprise.

10.2 Patchs virtuels

Aujourd'hui, les entreprises font appel à des plateformes de bug bounty (motivation à la recherche de bugs) pour identifier des vulnérabilités critiques au sein de leurs applications. Comme le souligne Gartner, « d'ici 2022, les plateformes et services de tests de sécurité basés sur le crowdsourcing seront utilisées par plus de 50% des entreprises, contre moins de 5% en 2018.2 » Il s'agit d'une attente forte pour les entreprises ayant adopté une culture DevSecOps qui se popularise avec le temps. **Cependant, l'identification des vulnérabilités, seule, n'est pas suffisante en termes de sécurité.**

SCHÉMA 10 :
Authentification SAML



2 Gartner Research, Market Guide for Application Crowdtesting Services, published: 19 September 2018

Les tâches (nouvelles fonctionnalités, étapes de la roadmap) des équipes produit étant déjà importantes en dehors de la correction des bugs, il leur est difficile de consacrer le temps nécessaire à l'identification et à la restauration des vulnérabilités. Les applications vulnérables développées en interne devront patienter de quelques jours à plusieurs semaines avant d'être patchées. Dans le cas des applications externes développées par des éditeurs, ce délai est susceptible d'être encore plus long. Bien évidemment, l'identification d'une vulnérabilité doit entraîner sa remédiation au plus vite.

UBIKA collabore avec des plateformes reconnues de bug bounty pour renforcer la cybersécurité de ses clients à l'aide de solutions de patching virtuel proposées par UBIKA WAAP Gateway et UBIKA WAAP Cloud. Ceci favorise la maîtrise des coûts et des efforts pour sécuriser les applications et restaurer les vulnérabilités identifiées, à partir d'une seule plateforme.

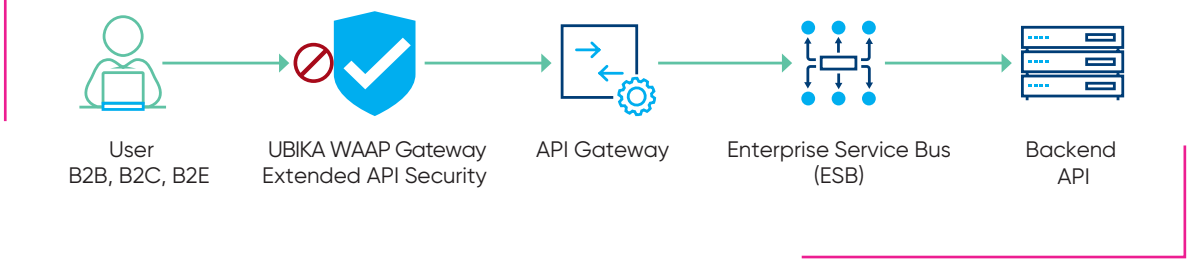
11. Le WAAP protège-t-il vos API ?

Toutes les entreprises actuelles utilisent des applications qui dépendent d'API. Les API (Application Programming Interface) jouent un rôle essentiel dans l'ère moderne actuelle. Elles sont utilisées pour interconnecter les services et assurer le transfert de différents types de données business. **Chaque application étant unique, il est devenu impératif pour les entreprises de disposer d'un mécanisme d'authentification commun à toutes.**

Les attaques par déni service ciblant les API sont en pleine expansion. Les entreprises doivent donc saisir l'importance de la sécurité des API et déployer un plan proactif pour juguler ces attaques. L'authentification, qui consiste à valider l'identité des utilisateurs, constitue une autre problématique majeure lorsqu'on fait appel aux API. Tous les utilisateurs ne doivent pas pouvoir accéder aux informations requérant un haut niveau de privilège. OAuth ou JSON Web Token peuvent aider en matière d'autorisations tandis qu'Open ID Connect peut déployer une authentification pour sécuriser davantage les API.

Les bots malveillants progressent ce qui implique d'instaurer une limite sur le nombre d'appels à une API par un client, sur une période donnée. Ce rate limiting peut aider à bloquer les adresses IP et à prévenir les attaques DoS. D'autre part, une défaillance lors de la validation des données entrantes ou de l'encodage des données sortantes peut aboutir à des attaques par injection qui vulnérabilisent les données sensibles. Dans ce scénario, la validation des schémas XML et JSON permet de vérifier les paramètres. De plus, le chiffrement peut être utilisé pour protéger les données, en amont de tout échange.

SCHÉMA 11 :
mécanisme de protection des API



Aujourd'hui, selon l'API et le type de données sensibles devant être transférées, le WAAP doit disposer de fonctions évoluées de protection des API pour éviter tout piratage de données.

UBIKA propose le module Extended API Security qui est directement intégré au sein de UBIKA WAAP Gateway et UBIKA WAAP Cloud, avec un même workflow de configuration. Ce module assure l'intégrité des API, ceux appartenant aux clients et ceux qu'ils utilisent.

Différentes techniques intelligentes sont appliquées, parmi lesquelles :

- Un filtrage avancé de sécurité, qui active les moteurs de sécurité sur la base de signatures et d'analyses heuristiques pour détecter les attaques, notamment celles utilisant les injections.
- Validation des API : une validation des schémas assure que les chemins, paramètres et méthodes sont conformes aux spécifications Swagger et OpenAPI.
- Authentification et autorisation d'accès aux API, grâce à des protocoles de sécurité comme SAML, OAuth et OpenID Connect, ou l'utilisation de JSON Web Token à des fins d'authentification.
- Manipulation des API : les données sensibles sont masquées et les appels aux API sont chiffrés avant tout échange de données.

12. L'éditeur du WAAP montre-il une approche orientée client ?

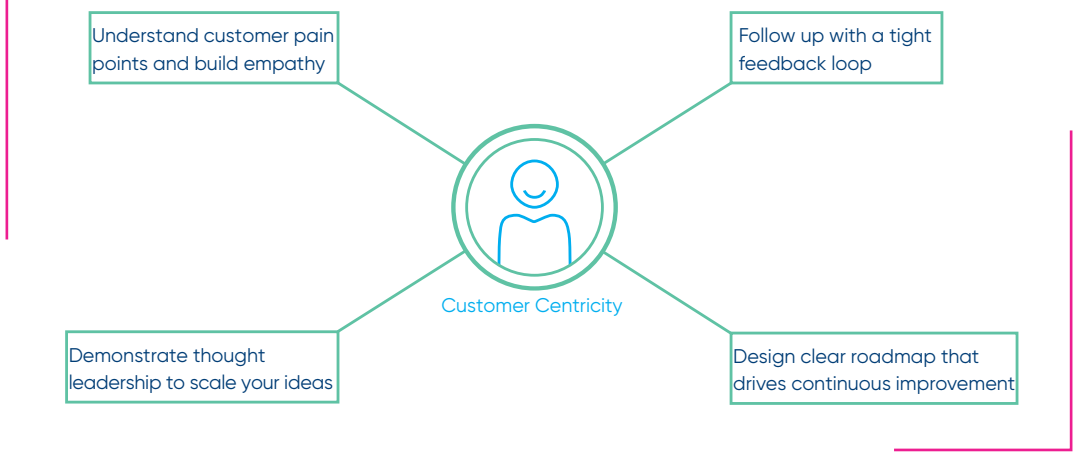
Le feedback des clients est essentiel, notamment lors des phases de développement logiciel. Il est important d'évaluer si les nouvelles fonctionnalités correspondent aux attentes des utilisateurs. **Un éditeur ayant mis en oeuvre une boucle de feedback est armé pour renforcer la productivité de son WAAP**. Il pourra mesurer les différentes facettes des performances globales, vous indiquer les axes d'amélioration, et comment les mettre en œuvre.

Un éditeur de solutions WAAP doit disposer d'une roadmap claire sur les évolutions à venir, d'une assise solide et d'un vrai leadership qui assurera la création de valeur via ses produits.

La roadmap doit refléter les demandes et retours des clients. Le développement produit doit être mené dans le respect du calendrier établi, et en informant tout le monde des nouveautés et changements du produit.

En tant qu'entreprise orientée client, UBIKA s'assure d'offrir une expérience client positive de bout en bout.
Les experts passionnés au sein de nos équipes comprennent et savent prendre en charge les problématiques des clients. L'équipe collabore étroitement avec leurs clients pour offrir une protection de haut niveau pour leurs API et applications.

SCHÉMA 12 :
une approche orientée client



SYNTHÈSE

La protection des applications et des API est un impératif pour les entreprises.
Les pare-feu applicatifs web (WAAP) constituent une mesure de sécurité indispensable pour toutes les applications Web et API.

Le WAAP le plus pertinent est celui qui :

- Offre une protection spécifique aux ressources web
- Neutralise les attaques d'envergure
- Sait arbitrer entre les volumes de faux positifs et de faux négatifs
- Est simple et rapide à déployer, sur différentes plateformes
- Utilise des listes blanches et des listes noires selon vos besoins
- Est évolutif pour gérer les pics de trafic
- Propose un TCO maîtrisé
- S'intègre avec votre approche DevSecOps
- Garantit une haute disponibilité et les performances applicatives
- Offre des fonctionnalités supplémentaires comme le SSO web et le patching virtuel
- Offre une sécurité sophistiquée des API
- S'adosse à des experts en sécurité orientés clients

Vous souhaitez investir dans un WAAP ? Ou vous rencontrez des limites avec celui que vous utilisez actuellement ? Vous devez connaître les critères de choix spécifiques à votre contexte, puis décider de la technologie la plus adaptée. Par exemple, si vous envisagez de migrer vers le cloud ou d'adapter une approche, la scalabilité (l'évolutivité) devient un critère important de choix.
Évaluez les différentes solutions WAAP disponibles sur le marché, sur la base des critères et interrogations présentés dans ce livre blanc.

Consultez notre site web pour découvrir comment améliorer votre sécurité applicative.

12 critères pour choisir un pare-feu applicatif Web

1. Protection spécifique aux ressources web

Pour se prémunir des attaques Web sur la couche 7, la solution doit appréhender les applications Web de façon granulaire et connaître leur contexte les utilisateurs et les sessions clients. C'est là que les WAAP entrent en jeu.

2. Répondre aux attaques majeures

Un WAAP doit protéger des attaques et vulnérabilités du Top 10 OWASP, en activant différents moteurs de sécurité. Il doit aussi détecter les attaques ciblant les vulnérabilités inconnues des développeurs, comme les attaques « zero day ».

3. Arbitrer entre faux positifs et faux négatifs

Des règles plus strictes génèrent plus de faux positifs, tandis que des règles plus indulgentes génèrent plus de faux négatifs. Un compromis est donc nécessaire. Outre la configuration de la sécurité elle-même, il est important de fournir un moyen facile d'ajuster la configuration par rapport aux requêtes bloquées.

4. Déploiement facile et rapide sur toute plateforme

Un WAAP doit être indépendant des plateformes. Il doit vous fournir une plateforme autonome, avec la même technologie qui sécurise vos API, quelle que soit la méthode de déploiement ou le nombre de fournisseurs d'infrastructures que vous utilisez.

5. Liste noire ou liste blanche ?

Les deux méthodes sont complémentaires. La méthode la plus efficace pour une liste noire est de travailler avec des modèles génériques, au lieu de créer un modèle pour chaque vulnérabilité. Les API sont destinées à être sécurisées via une liste blanche.

6. Flexibilité dans la gestion des pics de trafic

Un WAAP doit offrir une tarification flexible qui permet aux utilisateurs de payer à l'usage et à n'utiliser que ce dont ils ont besoin. L'utilisation de micro-services, hébergés dans des containers, vous permet de ne faire évoluer que les services qui ont besoin de plus de ressources.

7. Réduire le coût total de possession (TCO)

Un bon WAAP vous aide à sécuriser vos ressources critiques. Il réduit considérablement vos coûts en éliminant les coûts de mise en œuvre et de maintenance. L'automatisation des workflows est essentielle pour réduire le TCO.

8. Intégration à DevSecOps

Pour tirer le meilleur parti des fonctions avancées de sécurité d'un WAAP, il est important de les mettre à la disposition des développeurs. La solution DevSecOps doit être entièrement intégrée aux outils, langages et concepts de votre pipeline CI / CD et être automatisée.

9. Disponibilité et performance accrues des applications

Un WAAP doit vous permettre de mettre en place des clusters à très haute disponibilité, en mode actif-actif ou actif-passif. Il doit mettre en cache les contenus statiques, tels que les images, afin de les rendre disponibles pour les utilisateurs, ce qui réduit le temps de chargement des pages et améliore les performances.

10. Fonctionnalités supplémentaires comme le Web SSO et le patching virtuel

Un WAAP doit gérer les fonctionnalités sécurisées comme l'authentification et le SSO, pour simplifier l'accès aux applications web protégées. Il doit également superviser vos processus existants de sécurité, tels que les tests d'intrusion ou les programmes de bug bounty.

11. Protection des API

Les attaques par déni de service (DoS) sur les API augmentent chaque jour. Aujourd'hui, en fonction de l'API et du type de données sensibles transférées, le WAAP doit disposer de capacités avancées de protection des API pour éviter les piratages de données.

12. Une approche orientée client

Un éditeur qui dispose de retours d'expériences clients rapides et précis améliore la productivité de son WAAP. Le bon éditeur dispose d'une stratégie claire d'amélioration en continu, d'une assise solide et d'un leadership fort qui favorise la création de valeur via son produit.

A propos

Fondée en 2001 avec son siège social à Meudon en France et un centre de recherche à Montpellier, UBIKA est un fournisseur européen en matière de cybersécurité.

Sa mission est d'aider les organisations à sécuriser leur transformation digitale en protégeant les applications contre les cyberattaques.

Notre technologie Web Application & API Protection (WAAP) peut être déployée sur site, dans le Cloud, **en mode SaaS ou comme conteneur**, pour sécuriser à la fois les applications existantes et les applications cloud-native.

Plus de 600 entreprises et institutions publiques dans 35 pays nous confient la sécurité de leurs applications.

UBIKA
9-11 rue Jeanne Braconnier
92366 Meudon | France
Tél. : +33 (0)1 46 20 96 00
www.ubikasec.com