5 conseils pour prévenir une attaque cyber – La sécurité vue de l'intérieur

Ces dernières années, de nombreuses tendances ont vu le jour dans le domaine de la sécurité, notamment le concept de Shift Left, qui consiste à intégrer la sécurité plus tôt dans le cycle de développement, et DevSecOps, qui consiste à relier le développement, la sécurité et les opérations.

Malgré ces changements, les failles de sécurité continuent de jouer un rôle important dans notre vie quotidienne, ce qui nous amène à nous poser la question suivante : Est-il physiquement possible de mettre fin aux failles de sécurité ?

Il y a bien sûr un débat à ce sujet. Katie Arrington, RSSI au Pentagone, s'adressait à des entrepreneurs lorsqu'elle leur a fait répéter après elle : « Nous allons tous être victimes d'une attaque cyber ». Si les déclarations alarmistes et hyperboliques sur la cybersécurité sont pléthores, les professionnels de la sécurité ne savent que trop bien que le risque de perte de données peut être réduit, mais que nous ne pourrons jamais le ramener à 0.

Cela signifie-t-il qu'il ne faut pas se préoccuper des petits incidents ? Non, bien sûr, nous devons poursuivre nos efforts pour nous assurer que les attaquants n'accèdent pas à nos systèmes, mais le fait est qu'en plus de cela, nous devons également être en mesure de contrôler où un attaquant peut aller si un incident se produit et nous devons en être avertis.

Dans cet article, nous allons passer en revue les conseils et les outils que les développeurs et professionnels de la sécurité peuvent adopter pour aider non seulement à prévenir les incidents, mais aussi à éviter que les incidents ne se transforment en brèches.

Pour cela, nous avons besoin d'une approche inside-out. Voici ce que j'entends par approche « inside-out » :

Supposer que la menace peut venir de l'intérieur

Cela ne signifie pas qu'il faut se méfier de tous les employés et devenir paranoïaques quant à savoir qui, dans l'équipe, pourrait être un acteur malveillant. Il s'agit plutôt de partir du principe que les comptes et réseaux internes peuvent être violés et qu'un "interne" est simplement une personne qui a accès à nos systèmes, même si elle y a accédé par des moyens malveillants.

Souvent, nous envisageons la sécurité comme la construction d'un mur géant autour de notre infrastructure et de nos actifs, ce que l'on appelle parfois l'approche forteresse. Nous pouvons consacrer toutes nos ressources et tout notre temps à essayer de sécuriser ce mur, ce qui signifie que notre défense de sécurité est construite avec une seule conviction :

Personne de l'extérieur ne peut y pénétrer et les internes ne constituent pas une menace.

Bien sûr, il semble logique de penser ainsi, mais le résultat est que lorsque notre mur de sécurité est franchi, cela viole l'hypothèse de base autour de laquelle tout notre plan de sécurité a été

construit, nous laissant sans défense contre les menaces internes. Cela permet à un attaquant de se déplacer latéralement entre nos systèmes, nos services et notre infrastructure. En fin de compte, un incident se transforme en une attaque.

Nous devons mettre en place une sécurité qui part du principe que l'interne constitue une menace, ce qui exige un changement de mentalité et, en fin de compte, une couche de sécurité supplémentaire. En d'autres termes, nous devons mettre en place un environnement de confiance zéro. La sécurité zéro-confiance est une approche de la sécurité réseau basée sur le principe « coupable jusqu'à preuve du contraire » que John Kindervag, anciennement analyste principal chez Forrester Research et aujourd'hui directeur technique chez Palo Alto Networks, a formulé pour la première fois en 2010. Je défends l'idée que ses principes peuvent s'étendre au-delà de la sécurité du réseau et à la sécurité des applications.

Examinons les mesures pratiques qui peuvent être prises pour que cela devienne réalité.

1. Ne pas laisser de secrets dans les systèmes internes.

Il existe de nombreux exemples de failles ayant pour origine des secrets tels que des identifiants découverts dans des espaces publics, comme GitHub. Mais qu'en est-il des systèmes internes ?

Il s'agit d'un excellent exemple de l'échec d'une sécurité fondée sur le concept d'une forteresse autour des systèmes internes. Le code est un actif qui se propage et des secrets peuvent être trouvés partout où le code est copié, dans les dépôts Git, les wikis internes, les systèmes de messagerie, etc. Tous ces systèmes sont des cibles de grande valeur pour les attaquants. Il suffit d'un seul compte compromis dans votre dépôt Git interne pour qu'un attaquant effectue un audit de son historique et découvre un trésor d'informations sensibles. Ces secrets peuvent être utilisés pour passer des dépôts Git aux infrastructures et aux services.

Les systèmes internes doivent être nettoyés des informations sensibles comme les secrets.

Comment ? Eh bien, les secrets peuvent être enfouis dans l'historique, oubliés depuis longtemps, ou dans les logs, ce qui les rend très difficiles à trouver. De plus, de nombreuses informations transitent en permanence par ces systèmes, de sorte que des vérifications doivent être ajoutées au cycle de développement de manière programmatique.

La détection des secrets peut être ajoutée à deux endroits, du côté client, par exemple en tant que hook pré-commit sur la machine d'un développeur, et du côté serveur, après qu'un commit ait été effectué. La détection côté serveur est essentielle car on ne peut pas compter sur le fait que la détection côté client n'est pas contournée. Mais bien sûr, le scénario idéal est que les secrets n'arrivent pas jusqu'au serveur. La détection côté client peut être réalisée en utilisant la détection des secrets en ligne de commande (CLI), comme gg-shield, qui peut être installé sur les machines des développeurs pour détecter et empêcher les secrets d'être transmis au système de contrôle de version.

2. Par défaut, permissions minimales pour les clés et les services d'API

Imaginons qu'un attaquant ait réussi à compromettre vos défenses et à s'authentifier correctement sur vos systèmes internes. Cette activité peut être très difficile à détecter car ils semblent être des utilisateurs valides. En restreignant l'accès aux services et en réduisant les autorisations aux services et aux clés d'API au strict minimum, non seulement les dommages et les mouvements latéraux sont limités, mais une meilleure visibilité est fournie sur le moment où une clé d'API est utilisée en dehors de sa portée (en ayant les systèmes de journalisation appropriés en place).

Lorsqu'une entreprise utilise des services externes, elle doit s'assurer que les autorisations de cette API correspondent à la tâche qu'elle remplit. Elle doit notamment s'assurer qu'elle dispose d'API distinctes pour les autorisations de lecture seule et de lecture/écriture, si nécessaire.

De nombreuses API permettent également d'avoir un contrôle accru sur les données auxquelles il est possible d'accéder. L'utilisation de ces capacités pour répondre aux exigences minimales de la tâche est importante pour empêcher un attaquant d'accéder à des données sensibles ou de se déplacer latéralement dans les systèmes. Il est courant pour les développeurs inexpérimentés d'utiliser des API maîtres leur permettant d'utiliser une clé dans tous leurs projets. Mais cela augmente les dommages potentiels d'une intrusion.

Le cas échéant mettre en place une "liste blanche" des adresses IP

La "liste blanche" d'adresses IP fournit une couche supplémentaire de sécurité contre les acteurs malveillants qui tentent d'utiliser les API. En fournissant une "liste blanche" d'adresses IP d'un réseau privé, vos services externes n'accepteront que les demandes provenant de ces sources de confiance. Il est courant d'inclure une gamme d'adresses IP acceptables ou une adresse IP de réseau.

3. Segmentation des réseaux et des services

La segmentation du réseau et des services est une stratégie très efficace pour limiter l'impact des intrusions dans le réseau. Alors comment limiter quels services sont autorisés à parler à quels services ?

Segmentation du réseau

Chaque hôte et chaque réseau doivent être segmentés et séparés au niveau le plus bas qu'il est pratique de gérer. Pour un réseau physique, les routeurs ou les commutateurs de couche 3, divisent un réseau en réseaux distincts plus petits en utilisant des mesures telles que le réseau local virtuel (VLAN) ou les listes de contrôle d'accès (ACL). Les pare-feux de réseau sont mis en œuvre pour filtrer le trafic réseau entre les segments, et les pare-feux basés sur l'hôte filtrent le trafic du réseau local, ajoutant ainsi une sécurité supplémentaire.

Dans un environnement basé sur le cloud, la segmentation du réseau est réalisée par l'utilisation

de clouds privés virtuels (VPC) et de groupes de sécurité. Bien que les commutateurs soient virtualisés, l'approche consistant à configurer les règles d'entrée et les ACL pour segmenter les réseaux est essentiellement la même que pour l'infrastructure physique.

Segmentation des services

Si l'on considère que la segmentation du réseau vise à sécuriser le trafic entre les zones, la segmentation des services sécurise le trafic entre les services d'une même zone. La segmentation des services est une approche plus granulaire.

La mise en œuvre de la segmentation des services dépend de votre environnement d'exploitation et de votre infrastructure applicative. Les segments de services sont souvent appliqués par la configuration de pare-feu logiciels, de réseaux définis par logiciel tels que les réseaux superposés utilisés par les ordonnanceurs d'applications et, plus récemment, en exploitant un maillage de services.

Comme pour la segmentation du réseau, le principe du moindre privilège est appliqué et la communication de service à service n'est autorisée que lorsqu'il existe une intention explicite d'autoriser ce trafic.

4. Toujours crypter les données en transit et au repos

Les données en transit sont des données qui se déplacent d'un endroit à un autre, par exemple sur l'internet ou dans un réseau privé.

Les données au repos sont des données qui ne se déplacent pas d'un dispositif à l'autre ou d'un réseau à l'autre, comme les données stockées sur un disque dur, un ordinateur portable, une clé USB ou archivées/stockées d'une autre manière. La protection des données au repos vise à sécuriser les données inactives stockées sur n'importe quel dispositif ou réseau.

Comprendre que les données en transit doivent être cryptées et protégées est intuitif car les données quittent la sécurité interne. Mais les données au repos sont un domaine dans lequel nous pouvons à nouveau être victimes de l'idée que, parce qu'elles sont stockées en toute sécurité derrière notre forteresse, elles ne posent pas de problème.

Les données au repos sont stockées physiquement. Nous pouvons être tellement concentrés sur la prévention des cybermenaces qui se produisent dans le cloud que nous pouvons négliger l'endroit où se trouvent les données, physiquement, lorsqu'elles atteignent leur destination. C'est pourquoi il est crucial de crypter les données lorsqu'elles sont au repos, afin qu'un acteur malveillant ne puisse pas y avoir accès, même en cas de vol physique. Les données stockées dans le cloud n'échappent pas non plus à cette règle.

La façon dont les données sont stockées et la méthode de cryptage utilisée sont des questions importantes à poser lors du choix d'une solution de stockage dans le cloud. Selon McAfee, seuls 9,4 % des fournisseurs de cloud cryptent les données de leurs clients au repos. La sécurité des

données sur le cloud est une responsabilité partagée entre l'entreprise et son fournisseur de cloud et elle doit être en phase avec la sécurité que votre fournisseur de cloud accorde ou non.

Une autre erreur consiste à utiliser des protocoles de cryptage et des algorithmes de hachage médiocres pour protéger les données sensibles. Le plus grand exemple est le protocole de hachage MD5, encore largement utilisé pour protéger les mots de passe. Le hachage est différent du cryptage car il s'agit d'une fonction à sens unique, ce qui signifie qu'il faut faut vérifier si un mot de passe correspond à un hachage mais pas l'inverse. Mais grâce à l'augmentation de la puissance de calcul, il ne faut plus que quelques minutes pour déchiffrer les hachages MD5. La morale de tout cela est que le chiffrement des données au repos ne consiste pas seulement à chiffrer des données et à cocher la case. L'algorithme de cryptage doit être à la hauteur de son objectif et il s'agit d'une considération qui doit être revue à mesure que de nouvelles technologies apparaissent.

5. Maintenir ses dépendances à jour

Dans le développement de logiciels modernes, nous nous appuyons sur de nombreux éléments externes différents, ce qui crée une relation de confiance entre notre application et les services ou dépendances qu'elle utilise. Le problème est que ces dépendances peuvent être vulnérables aux attaques, ce qui signifie que notre application peut également être vulnérable à ces attaques.

Ce qui les rend particulièrement dangereux, c'est que si votre application utilise une dépendance dont la vulnérabilité a été signalée, les détails de cette vulnérabilité sont rendus publics, généralement après la publication d'un correctif. Si une entreprise utilise des dépendances obsolètes, elle donne à un attaquant des instructions sur la manière d'exploiter votre application. C'est pourquoi les dépendances doivent toujours être mises à jour régulièrement. Une tâche difficile si l'on considère qu'il est possible d'avoir des milliers de dépendances différentes qui peuvent elles-mêmes avoir des dépendances. Comme la détection des secrets, c'est un autre travail pour l'automatisation. Des outils permettent de vérifier automatiquement si les dépendances ne sont pas à jour et peuvent soumettre automatiquement une demande de mise à jour.

Il est important de construire des bases solides pour la sécurité, ce qui signifie qu'il faut réfléchir à des scénarios même évidents. Il est toujours intéressant d'examiner la sécurité de l'intérieur et de se demander si quelqu'un est capable de s'authentifier correctement dans des systèmes internes, et s'il est possible de le détecter et de l'arrêter.