Fiche pratique sur les attaques par déni de service

Introduction

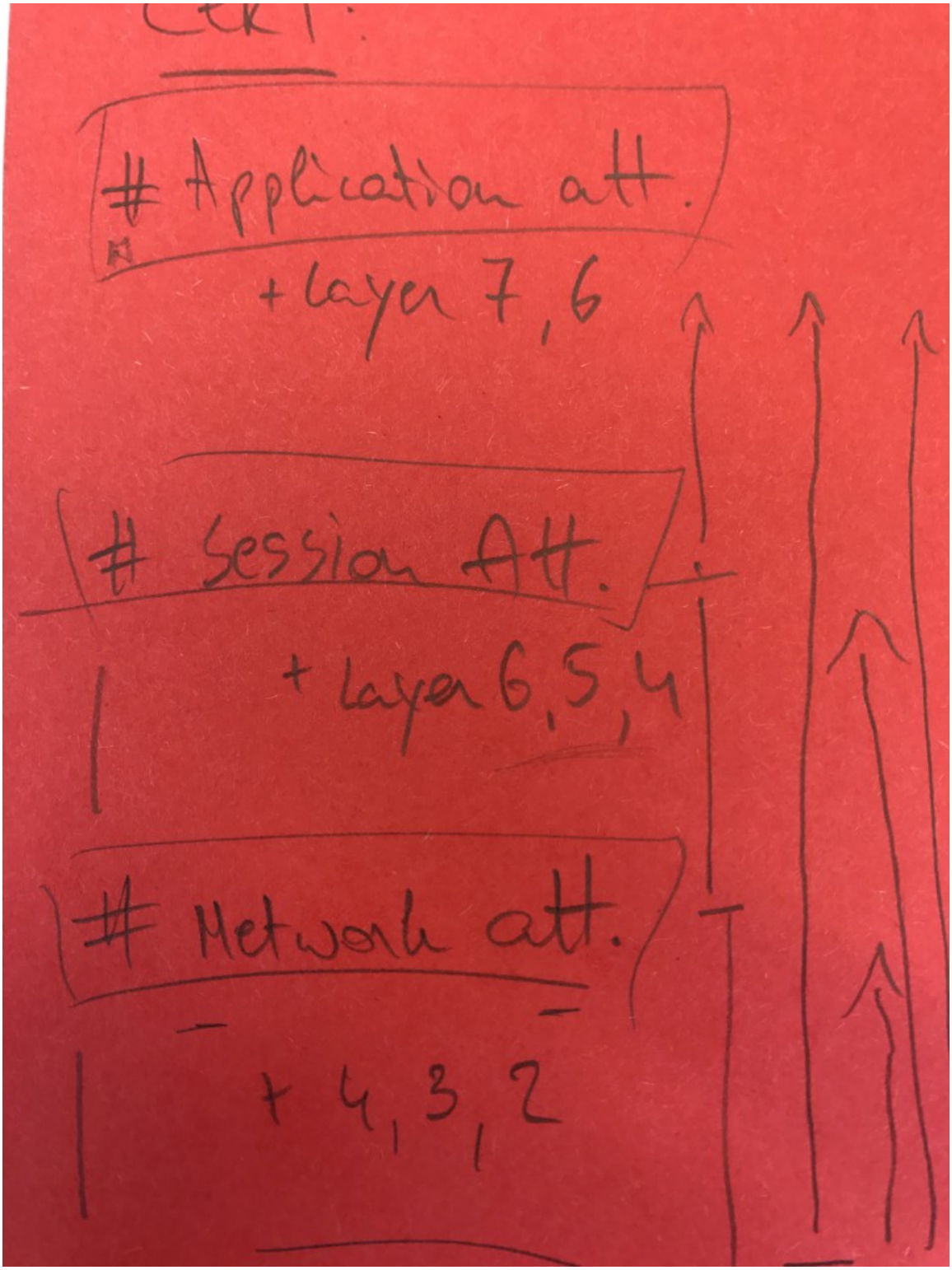
Cette fiche pratique décrit une méthodologie pour gérer les attaques par déni de service (DoS) à différents niveaux. Elle sert également de base à une discussion et une analyse plus approfondies, car il existe de nombreuses façons différentes de mener des attaques DoS.

Principes fondamentaux

Les méthodes anti-DoS ne pouvant être des solutions en une seule étape, vos développeurs et architectes d'applications/d'infrastructures doivent élaborer des solutions DoS avec soin. Ils doivent garder à l'esprit que la « disponibilité » est un élément fondamental de la triade CIA.

N'oubliez pas que si tous les éléments du système informatique impliqués dans le flux d'interopérabilité ne fonctionnent pas correctement, votre infrastructure en pâtit. Une attaque DoS réussie entrave la disponibilité des instances ou des objets d'un système et peut finir par rendre l'ensemble du système inaccessible.

Pour garantir la résilience des systèmes et leur résistance à une attaque DoS, nous vous recommandons vivement de procéder à une analyse approfondie des composants de votre inventaire en fonction de leur fonctionnalité, de leur architecture et de leurs performances (c'est-à-dire au niveau des applications, de l'infrastructure et du réseau).



Cet inventaire du système DoS doit rechercher les endroits potentiels où les attaques DoS peuvent causer des problèmes et mettre en évidence les points faibles du système, qui peuvent aller des erreurs liées à la programmation à l'épuisement des ressources. Il doit vous donner une image claire des problèmes en jeu (par exemple, les goulots d'étranglement, etc.). **Pour résoudre les problèmes, une bonne compréhension de votre environnement est essentielle pour développer des mécanismes de défense adaptés**. Ceux-ci pourraient être alignés sur :

1. Options de mise à l'échelle (**vers le haut** = composants matériels internes, **vers l'extérieur** = nombre de composants complets).
2. Des techniques conceptuelles/logiques existantes (telles que l'application de mesures de redondance, le bulk-heading, etc. - qui élargissent vos capacités internes).
3. Une analyse des coûts appliquée à votre situation.

Ce document adopte une structure d'orientation spécifique du CERT-EU pour analyser ce sujet, que vous devrez peut-être modifier en fonction de votre situation. Il ne s'agit pas d'une approche exhaustive, mais elle vous aidera à créer des blocs fondamentaux qui devraient vous aider à élaborer des concepts anti-DoS adaptés à vos besoins.

Analyse des surfaces d'attaque DoS

Dans cette fiche pratique, nous utiliserons la classification DDOS telle que documentée par le CERT-EU pour examiner les vulnérabilités du système DoS. Elle utilise le modèle OSI à sept couches et se concentre sur trois surfaces d'attaque principales, à savoir l'application, la session et le réseau.

1) Aperçu des faiblesses potentielles en matière de DoS

Il est important de comprendre que chacune de ces trois catégories d'attaques doit être prise en compte lors de la conception d'une solution résistante aux attaques DoS :

Les attaques applicatives visent à rendre les applications indisponibles en épuisant leurs ressources ou en les rendant inutilisables sur le plan fonctionnel.

Les attaques de session (ou de protocole) visent à consommer les ressources du serveur ou celles des équipements intermédiaires tels que les pare-feu et les équilibreurs de charge.

Les attaques réseau (ou volumétriques) visent à saturer la bande passante des ressources réseau.

Notez que les couches 1 et 2 du modèle OSI ne sont pas incluses dans cette catégorisation. Nous allons donc maintenant aborder ces couches et la manière dont les attaques DoS s'y appliquent.

La couche physique comprend les technologies de transmission matérielles d'un réseau. Il s'agit d'une couche fondamentale qui sous-tend les structures logiques des données des fonctions de niveau supérieur d'un réseau. Les scénarios DoS typiques impliquant la couche physique comprennent la destruction, l'obstruction et le dysfonctionnement du système. Par exemple, une femme âgée géorgienne a sectionné un câble souterrain, entraînant la perte de la connexion Internet dans toute l'Arménie.

La couche de données est la couche de protocole qui transfère les données entre les nœuds de réseau adjacents dans un réseau étendu (WAN) ou entre les nœuds d'un même segment de réseau local (LAN). Les scénarios DoS typiques sont l'inondation MAC (ciblant les tables MAC des commutateurs) et l'empoisonnement ARP.

Dans **les attaques par inondation MAC**, un commutateur est inondé de paquets qui ont tous des adresses MAC source différentes. L'objectif de cette attaque est de consommer la mémoire limitée utilisée par un commutateur pour stocker la table de traduction MAC et de ports physiques (table MAC), ce qui entraîne la purge des adresses MAC valides

et oblige le commutateur à passer en mode de basculement, où il devient un concentrateur réseau. Si cela se produit, toutes les données sont transmises à tous les ports, ce qui entraîne une fuite de données.

[Ajouts futurs à la fiche : impact en relation avec les attaques DoS et la correction compacte des documents]

Dans **les attaques par empoisonnement ARP**, un acteur malveillant envoie des messages ARP (Address Resolution Protocol) usurpés sur le réseau. Si l'adresse MAC de l'attaquant est associée à l'adresse IP d'un appareil légitime sur le réseau, l'attaquant peut intercepter, modifier ou bloquer les données destinées à l'adresse IP de la victime. Le protocole ARP est spécifique au réseau local et peut provoquer un déni de service sur la communication filaire.

La technologie de filtrage des paquets peut être utilisée pour inspecter les paquets en transit afin d'identifier et de bloquer les paquets ARP malveillants. Une autre approche consiste à utiliser des tables ARP statiques, mais celles-ci s'avèrent difficiles à maintenir.

Attaques applicatives

Les attaques de la couche application rendent généralement les applications indisponibles en épuisant les ressources du système ou en les rendant inutilisables sur le plan fonctionnel. Ces attaques n'ont pas besoin de consommer la bande passante du réseau pour être efficaces. Elles exercent plutôt une pression opérationnelle sur le serveur d'applications de telle sorte que celui-ci devient indisponible, inutilisable ou non fonctionnel. Toutes les attaques exploitant les faiblesses de la pile de protocoles de la couche 7 OSI sont généralement classées comme des attaques d'application. Elles sont les plus difficiles à identifier/atténuer.

[Ajouts futurs à la feuille : listez toutes les attaques par catégorie. Comme nous ne pouvons pas associer les mesures correctives à un vecteur d'attaque, nous devons d'abord les répertorier avant de discuter des mesures à prendre.]

Les attaques HTTP lentes envoient des requêtes HTTP très lentement et de manière fragmentée, une à la fois. Jusqu'à ce que la requête HTTP soit entièrement transmise, le serveur bloque ses ressources en attendant les données manquantes. À un moment donné, le serveur atteindra le nombre maximal de connexions simultanées, ce qui entraînera un déni de service (DoS). Du point de vue de l'attaquant, les attaques HTTP lentes sont peu coûteuses à mettre en œuvre, car elles nécessitent un minimum de ressources.

Concepts de conception logicielle

- **Utiliser d'abord une validation peu coûteuse en ressources** : nous voulons réduire l'impact sur ces ressources dès que possible. Une validation plus coûteuse (en termes de CPU, de mémoire et de bande passante) doit être effectuée par la suite.
- **Utilisation de la dégradation progressive** : il s'agit d'un concept fondamental à respecter lors de la phase de conception d'une application afin de limiter l'impact d'une attaque par déni de service. Vous devez maintenir un certain niveau de fonctionnalité lorsque des parties d'un système ou d'une application tombent en panne. L'un des principaux problèmes liés aux attaques par déni de service est qu'elles provoquent des arrêts soudains et abrupts des applications dans l'ensemble du système. Une conception tolérante aux pannes

permet à un système ou à une application de continuer à fonctionner comme prévu, éventuellement à un niveau réduit, plutôt que de tomber complètement en panne si certaines parties du système tombent en panne.

- **Prévenir les points de défaillance uniques** : détecter et prévenir les points de défaillance uniques (SPOF) est essentiel pour résister aux attaques DoS. La plupart des attaques DoS partent du principe qu'un système comporte des SPOF qui vont tomber en panne en raison d'une surcharge. Nous vous recommandons d'utiliser des composants sans état, des systèmes redondants, de créer des cloisons étanches pour empêcher les pannes de se propager à l'ensemble de l'infrastructure et de vous assurer que les systèmes peuvent continuer à fonctionner en cas de défaillance des services externes. Prévention
- **Évitez les opérations très gourmandes en ressources CPU** : lorsqu'une attaque DoS se produit, les opérations qui ont tendance à utiliser beaucoup de ressources CPU peuvent sérieusement ralentir les performances du système et devenir un point de défaillance. Nous vous recommandons vivement d'examiner les problèmes de performances de votre code, y compris les problèmes inhérents aux langages que vous utilisez. Voir Java JVM-IBM et Microsoft-IIS
- **Gérer les exceptions** : lorsqu'une attaque DoS se produit, il est probable que les applications génèrent des exceptions et il est essentiel que vos systèmes puissent les gérer correctement. Encore une fois, une attaque DoS part du principe qu'un système saturé ne sera pas en mesure de générer des exceptions de manière à pouvoir continuer à fonctionner. Nous vous suggérons de passer en revue votre code et de vous assurer que les exceptions sont gérées correctement. Voir Large-Scale-Systems Java et Java
- **Protect overflow and underflow** Étant donné que les débordements et sous-débordements de tampon entraînent souvent des vulnérabilités, il est essentiel d'apprendre à les prévenir. OWASP Overflow-Underflow-C Overflow
- **Threads** : évitez les opérations qui doivent attendre la fin de tâches volumineuses pour pouvoir se poursuivre. Les opérations asynchrones sont utiles dans ces situations.
- Identifiez les pages gourmandes en ressources et planifiez à l'avance.

Session

- **Limiter la durée des sessions côté serveur en fonction de l'inactivité et d'un délai d'expiration final** : (épuisement des ressources) Bien que le délai d'expiration des sessions soit le plus souvent abordé sous l'angle de la sécurité des sessions et de la prévention du détournement de sessions, il s'agit également d'une mesure importante pour éviter l'épuisement des ressources.
- **Limiter le stockage des informations liées à la session** : moins il y a de données liées à une session, moins la session utilisateur pèse sur les performances du serveur web.

Validation des entrées

- **Limiter la taille et les extensions des fichiers téléchargés** : cette tactique empêche les attaques par déni de service (DoS) sur l'espace de stockage des fichiers ou d'autres fonctions d'applications web qui utilisent le téléchargement comme entrée (par exemple, redimensionnement d'images, création de PDF, etc. (épuisement des ressources) - Liste de contrôle.
- **Limiter la taille totale des requêtes** : pour rendre plus difficile la réussite des attaques DoS gourmandes en ressources. (épuisement des ressources)

- **Empêcher l'allocation de ressources basée sur les entrées** : là encore, pour rendre plus difficile la réussite des attaques DoS consommant beaucoup de ressources. (épuisement des ressources)
- **Empêcher l'interaction entre les fonctions et les threads basée sur les entrées** : les entrées utilisateur peuvent influencer le nombre d'exécutions d'une fonction ou l'intensité de la consommation du processeur. Le fait de dépendre des entrées utilisateur (non filtrées) pour l'allocation des ressources pourrait permettre un scénario DoS par épuisement des ressources. (épuisement des ressources)
- **Les puzzles basés sur la saisie**, tels que les captchas ou les problèmes mathématiques simples, sont souvent utilisés pour « protéger » un formulaire Web. L'exemple classique est un formulaire Web qui envoie un e-mail après l'envoi de la demande. Un captcha peut alors empêcher la boîte mail d'être inondée par un attaquant malveillant ou un robot spammeur. **Les puzzles servent à lutter contre les abus de fonctionnalités, mais ce type de technologie ne permet pas de se défendre contre les attaques DoS.**

Contrôle d'accès

- **L'authentification comme moyen d'exposer les fonctionnalités** : le principe du moindre privilège peut jouer un rôle clé dans la prévention des attaques par déni de service en empêchant les pirates d'accéder à des fonctions potentiellement dommageables à l'aide de techniques de déni de service.
- **Le verrouillage des utilisateurs** est un scénario dans lequel un attaquant peut exploiter les mécanismes de sécurité de l'application pour provoquer un déni de service en abusant des échecs de connexion.

Attaques réseau

Pour plus d'informations sur les attaques réseau, consultez :

Juniper eSecurityPlanet

[Ajouts futurs à la fiche pratique : discuter des attaques qui saturent la bande passante du réseau. De nature volumétrique. Les techniques d'amplification rendent ces attaques efficaces. Liste des attaques : amplification NTP, amplification DNS, inondation UDP, inondation TCP]

Concepts de conception réseau

- **Prévention des points de défaillance uniques** : voir ci-dessus.
- **Mise en cache** : concept selon lequel les données sont stockées afin que les futures demandes concernant ces données puissent être traitées plus rapidement. Plus le volume de données traitées via la mise en cache est important, plus l'application devient résistante à l'épuisement de la bande passante.
- **L'hébergement de ressources statiques sur un domaine différent** réduit le nombre de requêtes http sur l'application web. Les images et les fichiers JavaScript sont des exemples typiques de fichiers chargés à partir d'un domaine différent.

Limitation du débit

La limitation du débit est le processus qui consiste à contrôler le débit du trafic entrant et sortant d'un serveur ou d'un composant. Elle peut être mise en œuvre au niveau de l'infrastructure ou au niveau de l'application. La limitation du débit peut être basée sur les adresses IP (incriminées), sur des listes de blocage d'adresses IP, sur la géolocalisation, etc.

- **Définissez une limite minimale pour le débit de données entrant** et supprimez toutes les connexions inférieures à ce débit. Notez que si la limite de débit est trop basse, cela pourrait avoir un impact sur les clients. Inspectez les journaux pour établir une base de référence du débit de trafic réel. (Protection contre les attaques HTTP lentes)
- **Définissez un délai d'expiration absolu pour les connexions**
- **Définissez une limite maximale de débit de données entrantes**, puis rejetez toutes les connexions supérieures à ce débit.
- **Définissez une limite totale de bande passante** pour éviter l'épuisement de la bande passante
- **Définissez une limite de charge**, qui spécifie le nombre d'utilisateurs autorisés à accéder à une ressource donnée à un moment donné.

Correctifs au niveau du FAI

- **Filtrer les adresses d'expéditeurs non valides à l'aide de routeurs périphériques**, conformément à la norme RFC 2267, afin de filtrer les attaques par usurpation d'adresse IP visant à contourner les listes de blocage.
- **Vérifiez au préalable les services de votre FAI en termes de DDOS** (prise en charge de plusieurs points d'accès Internet, bande passante suffisante (xx-xxx Gbit/s) et matériel spécial pour l'analyse du trafic et la défense au niveau des applications).

Correctifs au niveau mondial : services commerciaux de filtrage dans le cloud

- Envisagez d'utiliser un service de filtrage afin de résister à des attaques plus importantes (jusqu'à 500 Gbit/s).
- **Les services de filtrage** prennent en charge différents mécanismes pour filtrer le trafic malveillant ou non conforme.
- **Respectez les lois applicables en matière de protection des données/de confidentialité** : de nombreux fournisseurs acheminent le trafic via les États-Unis/le Royaume-Uni.

Articles connexes

- Publication du CERT-EU