Analyse de logs

Description

Dans le domaine de l'analyse forensique, l'analyse de logs consiste à exploiter les journaux d'événements et les données de trace (logs) pour reconstituer les activités qui ont eu lieu sur un système informatique.

Cette activité s'avère indispensable lorsqu'il s'agit d'enquêter sur un incident de sécurité (attaque, défaillance, comportement déviant, ...).

Prérequis d'exploitation

Pour mener une analyse de logs, il est impératif d'avoir accès aux fichiers de journaux d'événements du système concerné.

Connaissances nécessaires

- Compréhension du fonctionnement des systèmes informatiques ;
- Connaissance des techniques d'attaques utilisées et des vulnérabilités pouvant être exploitées pour effectuer une analyse de log efficace ;
- Maîtrise des formats de logs courants et des protocoles réseau.

Outils nécessaires

- Des outils d'analyse de logs tels que Splunk, ELK Stack (Elasticsearch, Logstash, Kibana) ou encore Graylog peuvent grandement faciliter l'analyse, bien qu'il soit possible d'effectuer une analyse de base sans outils spécifiques ;
- Un éditeur de texte avancé ou un outil capable de traiter de grands volumes de données textuelles peut également être utile.

Flux d'exécution

Explorer

Collecter des données de log à partir de différentes sources comme les serveurs, les routeurs, les firewalls et les postes informatiques. Il est important de s'assurer que les données de log soient complètes, intègrent et couvrent la période d'intérêt, car elles constitueront la base de l'analyse.

Expérimenter

Analyser les données de log collectées pour trouver des indices pertinents en utilisant, soit une approche manuelle, soit des outils d'analyse automatisés. L'analyse peut inclure la recherche de motifs d'attaques connus, de comportements anormaux, la reconstruction de séquences d'événements et l'identification des acteurs malveillants.

Par exemple, lors de l'analyse des logs d'une application web sur un serveur Apache, pour comprendre une attaque récente, cherchez des modèles d'attaques connus, des **adresses IP** suspectes ou des **User-Agent** ayant effectué un nombre anormalement élevé de requêtes sur une courte période.

Exploiter

Consultez les solutions de chaque challenge.

Bénéfices potentiels

L'analyse de logs peut permettre :

- La découverte de failles de sécurité ;
- L'identification d'acteurs malveillants et de leurs méthodes d'attaque ;
- La compréhension des impacts d'une attaque ou d'une défaillance sur un système.

Prérequis

Les prérequis suivants peuvent permettre de faciliter l'analyse de logs :

- Mettre en place et appliquer une politique de journalisation adaptée au exigences légales et réglementaires ;
- Utiliser des outils de surveillance et d'analyse de logs en temps réel.

Exemples

Exemple 1

Voici un exemple de logs Apache qui pourraient indiquer une tentative d'injection_sql :

```
127.0.0.1 - - [21/Jan/2022:10:00:00 +0000] "GET /index.php?id=1' OR '1'='1 HTTP/1.1" 200 612
127.0.0.1 - - [21/Jan/2022:10:00:01 +0000] "GET /index.php?id=1' OR '1'='2 HTTP/1.1" 500 612
127.0.0.1 - - [21/Jan/2022:10:00:02 +0000] "GET /index.php?id=1'; DROP TABLE users; -- HTTP/1.1" 500 612
```

Dans cet exemple, trois requêtes HTTP distinctes sont envoyées au serveur Apache. -* La première requête contient une chaîne de caractères

```
1' OR '1'='1
```

dans le paramètre id. Cette chaîne est souvent utilisée par les attaquants pour tester la vulnérabilité SQL Injection. Si la requête renvoie un code de retour HTTP 200 (succès), cela peut indiquer que l'injection SQL est possible. -* La deuxième requête, avec la chaîne

```
1' OR '1'='2
```

, est une autre tentative d'injection, visant à vérifier la réaction du serveur. Un code de retour http_500 (erreur interne du serveur) peut suggérer que l'injection SQL n'a pas réussi ou que le serveur a rencontré une erreur. -* La troisième requête est une tentative d'exploitation plus malveillante, visant à supprimer une table de la base de données. Encore une fois, un code de retour http_500 peut indiquer une erreur ou un blocage par des mesures de sécurité.

exemple_2}

Voici un exemple de logs Nginx qui pourraient indiquer une tentative d'attaque de type lfi_local_file_inclusion:

```
127.0.0.1 - - [21/Jan/2022:10:00:00 +0000] "GET /index.php?file=../../../etc/passwd HTTP/1.1" 403 612
127.0.0.1 - - [21/Jan/2022:10:00:01 +0000] "GET /index.php?file=../../../var/log/nginx/access.log
HTTP/1.1" 403 612
```

Dans cet exemple, deux requêtes HTTP sont envoyées au serveur Nginx. -* La première requête tente d'accéder au fichier

```
/etc/passwd
```

du système, ce qui est une tentative commune d'inclusion de fichier local. Si la requête renvoie un code de retour http_403 (accès interdit), cela indique que des mesures de sécurité peuvent être en place pour bloquer l'accès aux fichiers sensibles. -* La deuxième requête tente d'accéder au fichier de logs d'accès de Nginx. Un code de retour http_403 indique également que l'attaque LFI n'a pas réussi en raison des restrictions d'accès.

Retour fiches vulnérabilités

- Cyber fiches vulnérabilités