Attaquer et Défendre l'XDR : Une Perspective d'Attaquant et de Défenseur

Introduction

Pour un attaquant, l'objectif n'est pas toujours de "casser" l'XDR (Extended Detection and Response) de manière frontale, mais plutôt de le rendre aveugle, de lui mentir, ou de le noyer pour que les véritables actions malveillantes passent inaperçues. Ce document explore les stratégies qu'un attaquant pourrait employer pour "corrompre" un XDR, ainsi que les contre-mesures que les équipes de sécurité (Blue Team) peuvent mettre en œuvre pour se défendre.

"Corrompre" un XDR peut signifier :

- **L'aveugler (Évasion)** : Empêcher l'XDR de voir les activités malveillantes.
- **Le tromper (Manipulation)** : Lui faire croire que les activités malveillantes sont légitimes.
- **Le saturer (Déni de Service)** : Créer tellement de bruit et de fausses alertes que les analystes sont débordés.
- **Le compromettre (Prise de contrôle)** : Utiliser l'XDR comme un outil pour pivoter dans le réseau.

Stratégie 1 : L'Attaque par l'Ingestion de Données (Data Poisoning)

L'XDR se fie aux données qu'il collecte depuis de multiples sources (endpoints, réseau, cloud, email...). Si l'attaquant peut corrompre ces données à la source, l'XDR prendra de mauvaises décisions.

Phase 1 : Comprendre les Sources de Données

Une fois un premier accès obtenu sur une machine (via phishing, exploit, etc.), la première tâche de l'attaquant est de découvrir : Quel XDR est utilisé ? Quels logs remonte-t-il ? Depuis quels endpoints ? Quels serveurs ? Est-ce qu'il ingère les logs du pare-feu, de l'Active Directory, de Microsoft 365 ? Cette reconnaissance est cruciale pour empoisonner les données.

Phase 2 : Empoisonnement Lent et Progressif (Boiling the Frog)

L'un des points forts de l'XDR est sa capacité à établir une "baseline" du comportement normal grâce au Machine Learning (ML). L'attaquant va retourner cette force contre lui.

Action : Introduire des outils (par exemple, un script PowerShell personnalisé ou un binaire non signé) et les utiliser pour des tâches bénignes pendant des jours ou des semaines (ex: ping, requêtes DNS vers des adresses légitimes).

Effet : Le moteur de ML de l'XDR va progressivement apprendre que cet outil, ce script, et ce type de comportement sont "normaux" pour cet environnement ou cet utilisateur. Il va l'intégrer à sa baseline. Le jour de l'attaque réelle, l'XDR pourrait ne pas générer d'alerte ou lui assigner un score de risque très faible.

Phase 3 : Manipulation Directe des Logs (Log Tampering)

Si l'attaquant a des privilèges suffisants sur un endpoint ou un serveur, il peut altérer les logs avant qu'ils ne soient envoyés à l'XDR.

Action : * **Suppression d'événements** : Cibler les logs PowerShell (Event ID 4104) ou les créations de processus (Event ID 4688) et supprimer les lignes spécifiques qui tracent l'activité malveillante. Des outils comme Mimikatz peuvent effacer les traces. * **Modification d'événements** : Modifier un log pour remplacer une adresse IP malveillante par une adresse légitime, ou changer le hash d'un fichier malveillant par celui d'un fichier connu et inoffensif.

Effet : L'XDR reçoit des informations incomplètes ou fausses, brisant sa capacité de corrélation.

Stratégie 2 : Attaquer l'Agent XDR Lui-même

L'agent installé sur les endpoints est le principal capteur et une cible de choix.

- **Désactivation ou "blinding"** : Obtenir des droits d'administrateur ou SYSTEM pour désactiver, tuer le processus de l'agent XDR, ou le mettre dans un mode où il ne remonte plus rien. Les techniques d'"EDR/XDR Evasion" sont courantes.
- **Exploitation de vulnérabilités** : Découvrir une faille 0-day dans l'agent pour le neutraliser ou l'utiliser à son profit.

Stratégie 3 : Génération de Bruit pour Saturer les Analystes

C'est une attaque psychologique contre l'équipe de sécurité (le SOC).

Action : Lancer des milliers d'activités "légèrement suspectes" mais finalement bénignes depuis de nombreuses machines contrôlées (ex: tentatives de connexion réseau sans succès, exécution de commandes `whoami` en boucle).

Effet : L'XDR génère des centaines ou des milliers d'alertes de faible ou moyenne priorité, provoquant une "fatigue des alertes" (alert fatigue) chez l'équipe SOC. Au milieu de ce chaos, l'attaque réelle est lancée, noyée dans la masse et potentiellement ignorée ou traitée trop tard.

Comment se Défendre de ces Approches ? (La Perspective du Blue Teamer)

Contre le Data Poisoning :

- **Intégrité des logs** : Mettre en place des mécanismes garantissant l'intégrité des logs entre la source et le collecteur (hachage, canaux sécurisés).
- **Ne jamais faire confiance aveuglément** : Compléter les modèles de ML par des règles de détection plus "rigides" (ex: alerter si un processus non signé tente de contacter une IP extérieure, même s'il est connu).

- **Threat Hunting proactif** : Chercher activement des anomalies "faibles" qui, une fois corrélées par un humain, révèlent une attaque lente.

Contre les Attaques sur l'Agent :

- **Hardening des postes** : Appliquer le principe du moindre privilège pour empêcher l'attaquant d'obtenir les droits admin et neutraliser l'agent XDR.
- **Surveillance de la santé des agents** : L'XDR doit surveiller l'état de ses propres agents et générer une alerte si un agent cesse de communiquer ou si son processus est tué.

Contre la Saturation :

- **Affiner la détection** : Travailler continuellement sur les règles de détection pour réduire les faux positifs.
- **Orchestration (SOAR)** : Utiliser des outils d'orchestration pour automatiser le traitement des alertes de faible criticité, permettant aux analystes de se concentrer sur les signaux importants.

Conclusion

L'ingestion de données est à la fois la plus grande force et une faiblesse potentielle de l'XDR. Un attaquant sophistiqué ne cherchera pas à forcer la porte, mais à obtenir la clé en manipulant l'information, qui est le carburant de tous les systèmes de sécurité modernes. C'est un jeu constant du chat et de la souris.