AYI NEDJIMI Consultants

Bypass EDR : Techniques d'Évasion et

18 April 2026 • Mis à jour le 18 April 2026 • 50 min de lecture

Guide technique complet sur les techniques de bypass EDR : unhooking us
obfuscation, LOLBins. Défenses et détection incluses.

Le contournement des solutions EDR (Endpoint Detection and Response) constitue
complexes de la cybersécurité offensive contemporaine. Alors que les entreprises
détection sophistiquées capables d'analyser chaque appel système, chaque alloc
attaquants développent parallèlement des techniques d'évasion toujours plus ingé
numériques. Cet article propose une analyse exhaustive des techniques de bypas
classiques d'unhooking en espace utilisateur jusqu'aux approches les plus avance
Nous examinerons en détail le fonctionnement interne des EDR pour comprendre
taxonomie complète des vecteurs d'évasion, puis nous étudierons les contre-mes
pour détecter ces tentatives de contournement. Que vous soyez pentesteur cherc
antivirus, analyste SOC souhaitant comprendre les angles morts de vos outils, ou
stack de détection, ce guide technique de référence vous fournira les connaissanc
en perpétuelle évolution.

Qu'est-ce qu'un EDR et comment fonctionne-t-il en profondeur

Avant de pouvoir contourner un EDR, il est indispensable de comprendre précisém
n'est pas un simple antivirus basé sur des signatures : c'est un système multi-cou
différents niveaux pour collecter de la télémétrie comportementale, l'analyser loca
de remédiation automatisées. Cette compréhension architecturale est le fondemen

Les hooks userland : la première ligne de détection

La technique de détection la plus répandue parmi les EDR consiste à injecter une l
du système. Cette DLL, souvent appelée « sensor » ou « agent », intercepte les ap
bibliothèque qui sert d'interface entre le mode utilisateur et le noyau Windows. Co
fonctions comme `NtAllocateVirtualMemory`, `NtWriteVirtualMemory`, `NtCreat`
(saut inconditionnel) vers son propre code d'analyse.

Lorsqu'un programme appelle par exemple `NtAllocateVirtualMemory` pour allou
(RWX), le flux d'exécution est d'abord redirigé vers la DLL de l'EDR. Celle-ci inspe
permissions demandées, processus cible — et décide si l'opération est suspecte.
alerte, ou les deux. Si l'opération semble légitime, le hook transmet l'appel à la vé
exécution normale.

Voici un exemple simplifié de ce à quoi ressemble un hook inline typique en mémo

```
// Avant le hook (ntdll.dll originale)
NtAllocateVirtualMemory:
  mov r10, rcx          ; 4C 8B D1
  mov eax, 0x18         ; B8 18 00 00 00 (numéro de syscall)
  syscall               ; 0F 05
  ret                   ; C3
```