تحلیل تکنیک‌های رایج دور زدن EDR توسط بدافزارها

راهنمایی کاربردی برای مدیران امنیت سایبری

مقدمه

امروزه ابزارهای ***Endpoint Detection and Response (EDR)*** به یکی از حیاتی‌ترین ستون‌های دفاعی سازمان‌ها تبدیل شده‌اند. EDR ها با **تحلیل رفتار سیستم، شناسایی تهدیدات و پاسخ خودکار به حملات**، نقشی کلیدی در امنیت نقاط پایانی ایفا می‌کنند.

اما با پیشرفت فناوری‌های امنیتی، مهاجمان نیز به همان اندازه پیچیده‌تر شده‌اند. در حال حاضر بسیاری از گروه‌های تهدید پیشرفته (APT) و بدافزارهای نسل جدید، به طور خاص طراحی می‌شوند تاEDR ها را دور بزنند و بدون شناسایی، به فعالیت‌های مخرب خود ادامه دهند.

این تکنیک‌ها به‌سرعت در حال تکامل هستند و تنها به مخفی‌کاری ساده یا رمزگذاری محدود نمی‌شوند؛ بلکه از روش‌های پیشرفته‌ای نظیر:

- ✓ **تزریق کد در حافظه(In-Memory Attacks)**
- ✓ **استفاده از آسیب‌پذیری‌های درایورها(BYOVD)**
- ✓ **استفاده از فراخوانی‌های سیستمی پایین‌سطح(Direct Syscalls)**
- ✓ **ابزارهای قانونی برای اهداف مخرب(Living off the Land Binaries - LOLBins)**

و حتی **غیرفعال‌سازی مستقیم خودِ سرویس‌های EDR** بهره می‌برند.

در این مقاله از سلام دیجی، به‌صورت ساختارمند و بر پایه مستندات معتبر، مهم‌ترین تکنیک‌های دور زدن EDR ها را بررسی می‌کنیم، و راهکارهای عملیاتی برای **مدیران امنیت** به‌منظور کاهش ریسک این حملات ارائه می‌دهیم.

بخش اول – معرفی اجمالی EDR و محدودیت‌های آن

۱,۱ EDR چیست و چگونه کار می‌کند؟

Endpoint Detection and Response (EDR) یا «تشخیص و پاسخ نقطه پایانی»، یکی از کلیدی‌ترین ابزارهای امنیت سایبری در سازمان‌هاست که وظیفه آن:

- ✓ **شناسایی تهدیدات سایبری** در سطح نقاط پایانی(EndPoints)
- ✓ **تحلیل و بررسی رفتارهای مشکوک**
- ✓ **ایجاد پاسخ خودکار یا هشدار به تیم امنیتی**
- ✓ **جمع‌آوری داده‌های Telemetry** برای تحلیل عمیق‌تر می باشد.

این ابزارها معمولاً از ترکیبی از روش‌های زیر استفاده می‌کنند:

- ✓ **Behavioral Detection :**تحلیل رفتار فرایندها و فایل‌ها
- ✓ **Signature-Based Detection :**شناسایی مبتنی بر الگوهای شناخته‌شده
- ✓ **Heuristic & Machine Learning Analysis :**تحلیل‌های هوشمند مبتنی بر یادگیری ماشین
- ✓ **Threat Intelligence Integration :**استفاده از تهدیدات شناخته‌شده در سطح جهانی

هدف اصلیEDR ها، جلوگیری از حملات پیشرفته‌ای مثل باج‌افزارها، APTها و حملات بدون فایل (fileless) است.

۱,۲ محدودیت‌های ذاتیEDR

با وجود پیشرفت‌های چشمگیر، EDR ها به دلایل مختلف دارای محدودیت‌های فنی هستند که مهاجمان به‌شدت روی آن‌ها حساب می‌کنند:

◆۱,۲,۱ دید محدود به سطح کاربر(User Mode)

بیشترEDR ها برای جلوگیری از تأثیر بر عملکرد سیستم، به‌صورت Process سطح کاربر عمل می‌کنند. این یعنی اگر حمله‌ای در سطح کرنل یا از طریق درایور آسیب‌پذیر انجام شود، بسیاری از این ابزارها توانایی تشخیص آن را نخواهند داشت.

◆۱,۲,۲ ضعف در تشخیص حملات Fileless وMemory-based

حملاتی که مستقیماً در حافظه اجرا می‌شوند و هیچ فایلی روی دیسک ایجاد نمی‌کنند مانند-PowerShell based Attacksیا Reflective DLL Injection ، همچنان یک نقطه‌ضعف جدی برای بسیاری ازEDR ها به شمار می‌رود.

◆۱,۲,۳ قابل دور زدن بودنHook ها وAPI Monitoring

EDR ها معمولاً با استفاده از تکنیک **API Hooking** فرایندهای مشکوک را تحت نظر می‌گیرند، اما مهاجمان حرفه‌ای می‌توانند اینhook ها را شناسایی کرده و دور بزنند (Unhookingیا استفاده ازSyscall های مستقیم).

◆۱,۲,٤ حجم بالای داده و تحلیل اشتباه(False Positive / Negative)

به دلیل جمع‌آوری حجم زیادی از داده‌ها، EDR ها در مواردی دچار خطاهای تحلیل می‌شوند که ممکن است منجر به **هشدارهای اشتباه** یا **ندیدن تهدیدات واقعی** شود.

◆۱,۲,۵ تمرکز روی تکنیک‌های شناخته‌شده

بسیاری ازEDR ها بیشتر بر تکنیک‌هایی تمرکز دارند که در **MITRE ATT&CK** شناسایی و ثبت شده‌اند. به همین دلیل، تکنیک‌های جدید یا سفارشی‌شده، گاهی از دید آن‌ها پنهان می‌ماند.

۱,۳ چرا مهاجمان روی دور زدن EDR تمرکز کرده‌اند؟

امروزه بسیاری از حملات سایبری با هدف‌های کلان، به‌گونه‌ای طراحی می‌شوند که ابتدا **با اطمینان کامل از دور زدنEDR** ، دسترسی پایدار را فراهم کنند. دلیل این تمرکز:

- ✓ **پوشش گسترده EDR در سازمان‌ها:** مهاجم بدون عبور ازEDR ، نمی‌تواند حمله را به سرانجام برساند.
- ✓ **افزایش هوشEDR ها:** مهاجمان مجبور به استفاده از تکنیک‌های پیچیده‌تر شده‌اند.
- ✓ **اتکای شدید تیم‌های SOC بهEDR** :مهاجمان از این وابستگی سوءاستفاده می‌کنند.

به بیان ساده، **دور زدن EDR به‌معنای باز شدن دروازه حملات پیشرفته به زیرساخت شماست.**

از این بخش چه نتیجه ای میگیریم؟

مدیران امنیت باید بپذیرند که **هیچEDR ی به‌تنهایی غیرقابل نفوذ نیست** و دور زدن آن، یکی از محورهای کلیدی حملات مدرن است.
شناخت دقیق تکنیک‌هایEDR Evasion ، اولین گام برای طراحی دفاع چندلایه و کارآمد در برابر تهدیدات پیچیده است.

۲ .تحلیل تکنیک‌های رایج دور زدن EDR توسط بدافزارها

۲,۱ تکنیک اول(Bring Your Own Vulnerable Driver)BYOVD :

آوردن درایور آسیب‌پذیر برای دور زدنEDR

مفهوم تکنیکBYOVD

در این تکنیک، مهاجم یک درایور قانونی امضاشده (Signed Driver) که دارای آسیب‌پذیری‌های شناخته شده است را به همراه حمله خود به سیستم قربانی منتقل می‌کند.
با بارگذاری این درایور آسیب‌پذیر، مهاجم به راحتی می‌تواند به **سطح دسترسی Kernel Mode (Ring 0)** دست یابد؛ یعنی دقیقاً همان جایی که بیشترEDR ها دسترسی کافی ندارند یا نمی‌توانند مداخله کنند.

هدف نهایی مهاجم:

- ✓ **غیرفعال‌سازی EDR یاAV**
- ✓ **اجرای کدهای مخرب با سطح دسترسی بالا**
- ✓ **دور زدن سیاست‌های امنیتی سیستم عامل**

چرایی موفقیت تکنیکBYOVD

- ✓ سیستم‌عامل Windows اجازه نصب درایورهای امضاشده را می‌دهد، حتی اگر درایور دارای آسیب‌پذیری باشد.
- ✓ بسیاری از این درایورها قبلاً در حملات واقعی استفاده شده‌اند، اما همچنان در اینترنت قابل‌دسترس‌اند.
- ✓ با تغییر تاریخ سیستم یا استفاده از گواهی‌های قدیمی، مهاجم می‌تواند فرآیند اعتبارسنجی امضا را دور بزند.

نمونه‌های واقعی از حملاتBYOVD

سال	بدافزار / گروه	درایور مورد سوءاستفاده	شرح حمله
۲۰۲۳	Medusa Locker	RTCore64.sys (MSI)	Kill کردن سرویس EDR و رمزگذاری سیستم
۲۰۲٤	Abyss Locker	mhyprot2.sys (MiHoYo)	غیر فعال‌سازی EDR و استقرار باج‌افزار
۲۰۲٥	LockBit	Multiple Drivers	بارگذاری چندین درایور آسیب‌پذیر برای حملات چند مرحله‌ای

نحوه کار دقیقBYOVD

۱ .بدافزار، درایور آسیب‌پذیر را در سیستم بارگذاری می‌کند.

۲. از آسیب‌پذیری‌های درایور برای اجرای دستورات با سطح کرنل سوءاستفاده می‌شود.

۳. مهاجم می‌تواند:

- ✓ حافظه EDR را Patch کند.

- ✓ سرویس‌های EDR و Security Suite را خاموش کند.
- ✓ فایل‌ها و Registry های محافظت‌شده را تغییر دهد.

٤. پس از اجرای موفق، معمولاً درایور حذف می‌شود تا شناسایی دشوارتر شود.

ابزارها و چارچوب‌های معروف BYOVD

- ✓ **KDMapper**
- ✓ **CheekyBlinder**
- ✓ **HookSignTool**
- ✓ ابزارهای شخصی‌سازی‌شده توسط APT ها و Ransomware-as-a-Service (RaaS)

چرا BYOVD برای EDR ها تهدید بزرگی است؟

- ✓ مهاجم از **درایور قانونی و امضاشده** استفاده می‌کند؛ EDR به‌راحتی نمی‌تواند آن را بلاک کند.
- ✓ تکنیک‌های متداول EDR مثل **Hooking و Behavioral Detection** در برابر آن ناکارآمدند.
- ✓ قابلیت **غیرفعال‌سازی یا نابودی EDR** توسط مهاجم، مسیر حمله را کاملاً باز می‌کند.

اقدامات پیشگیرانه برای مدیران امنیت

اقدام	شرح
محدودیت نصب درایور	تنها نصب درایورهای مجاز از طریق Policy ها
مانیتورینگ لاگ نصب درایور	بررسی دقیق Windows Event Logs و ابزار SIEM
کنترل تغییرات تاریخ سیستم	جلوگیری از تنظیم تاریخ به گذشته با ابزارهای سختگیرانه
استفاده از EDR با محافظت Kernel-level	برخی EDR ها، ماژول‌هایی با محافظت سطح کرنل دارند
Patch Management قوی	به‌روزرسانی و حذف درایورهای آسیب‌پذیر موجود در سیستم‌ها

نکته کلیدی برای مدیران امنیت

صرف داشتن EDR کافی نیست؛ لازم است **سیاست‌های سخت‌گیرانه نصب درایور** و **نظارت مداوم بر لاگ‌ها** را به بخشی از استراتژی امنیت سازمانی تبدیل کنید.

✅جمع‌بندی تکنیک BYOVD :

تکنیک BYOVD یکی از خطرناک‌ترین روش‌های دور زدن EDR است که به مهاجم دسترسی کرنل می‌دهد و می‌تواند هر ابزار امنیتی را از کار بیندازد.
تنها راه مقابله با آن، ترکیبی از:

- ✓ **سیاست‌های سختگیرانه نصب درایور**
- ✓ **استفاده ازEDR های نسل جدید باKernel Protection**
- ✓ **مانیتورینگ مداوم تغییرات سیستم**

است.

۲,۲ تکنیک دوم: تزریق کد در حافظه(In-Memory Code Injection & Fileless Attacks)

اجرای Fileless؛ جایی که بسیاری ازEDR ها شکست می‌خورند

مفهوم حملات تزریق حافظه وFileless

در این تکنیک، بدافزار بدون ایجاد هیچ فایل قابل‌مشاهده‌ای روی دیسک، کد مخرب را مستقیماً در **حافظه (RAM)** سیستم تزریق و اجرا می‌کند.
این روش، یکی از محبوب‌ترین شیوه‌ها برای حملات **fileless** به شمار می‌رود؛ چراکه بسیاری ازEDR ها همچنان به مانیتورینگ فایل و دیسک متکی‌اند.

رایج‌ترین روش‌های حملات حافظه‌ای

تکنیک	توضیح مختصر
Reflective DLL Injection	بارگذاری DLL مستقیماً در حافظه، بدون نیاز به فایل فیزیکی
Process Hollowing	تزریق کد به فرایند قانونی) مثلاً(explorer.exe ، جایگزین کردن فضای حافظه آن با Payload مخرب
Shellcode Injection	بارگذاری Shellcode رمزنگاری‌شده یا obfuscate شده در حافظه
Thread Hijacking	ربودن رشته‌های اجرایی (Threads) در فرایندهای مجاز
AtomBombing	استفاده از ساختارهای Windows Atom Tables برای تزریق غیرمستقیم کد به حافظه

چرا این تکنیک مؤثر است؟

- ✓ **عدم نیاز به فایل:** هیچ اثر ماندگاری روی دیسک باقی نمی‌ماند.
- ✓ **فرار از اکثر روش‌های signature-based:** چون فایل وجود ندارد که اسکن شود.
- ✓ **استفاده از رمزنگاری و Obfuscation:** تشخیص کد تزریق‌شده بسیار سخت‌تر می‌شود.
- ✓ **اجرای کد در فضای فرایندهای قانونی:** باعث پیچیده‌تر شدن تحلیل رفتاری توسط EDR ها می‌شود.

نمونه‌های واقعی حملات In-Memory

بدافزار / گروه	تکنیک استفاده‌شده	جزئیات حمله
Cobalt Strike	Reflective DLL Injection	بارگذاری کامل Agent در حافظه با ارتباط مستقیم به C2
Emotet	Process Hollowing & Thread Injection	بهره‌برداری از فرایندهای قانونی برای تزریق و مخفی‌سازی حملات فیشینگ

QakBot	In-memory Shellcode Execution	تزریق Shellcode رمزنگاری‌شده برای دور زدن EDR و ایجادPersistence

ابزارها و فریم‌ورک‌های مرتبط با این حملات

- ✓ **Cobalt Strike**
- ✓ **Metasploit**
- ✓ **Donut Tool**
- ✓ **SharpShooter**
- ✓ **PowerShell Empire**

چراEDR ها در برابر حملات حافظه‌ای آسیب‌پذیرند؟

- ✓ بیشترEDR ها روی تحلیل فایل و رفتار دیسک تمرکز دارند.
- ✓ Memory Scanning در بسیاری ازEDR ها یا غیرفعال است یا به‌صورت غیربرخط (On-Demand) انجام می‌شود.
- ✓ رمزنگاری و پیچیده‌سازی Shellcode می‌تواند بسیاری از تحلیل‌های حافظه را ناکارآمد کند.
- ✓ فرایندهای قانونی که قربانی تزریق می‌شوند (مانند svchost.exe یاexplorer.exe) معمولاً توسط EDR ها اعتماد زیادی دارند.

اقدامات پیشگیرانه و راهکارهای مدیریتی

اقدام	شرح
فعالسازی Memory Scanning مداوم	بسیاری ازEDR ها امکان Memory Scanning Real-Time دارند، اما باید به‌صورت دستی فعال شود.
مانیتورینگ رفتارهای مشکوک حافظه	تمرکز بر فعالیت‌هایی مانند VirtualAllocEx و NtWriteVirtualMemory.
تقویت SIEM با لاگ‌های Memory Events	لاگ‌برداری از رویدادهای حافظه به SIEM ارسال شود تا تحلیل لحظه‌ای ممکن شود.

محدودسازی ابزارهای اسکریپت‌نویسی مانند PowerShell	با تنظیم GPO و ابزارهایی مانند AppLocker یا WDAC می‌توان این ابزارها را کنترل کرد.
تحلیل Threat Hunting دوره‌ای روی حافظه سیستم‌ها	تحلیل عمیق دوره‌ای حافظه برای کشف Shellcode یا Payloadهای مخفی.

نکته مهم : حملات مبتنی بر حافظه وFileless ، تهدیدی بسیار جدی برایEDR های سنتی محسوب می‌شوند. بدون **Memory Scanning فعال و دقیق**، سازمان‌ها در برابر این حملات تقریباً بی‌دفاع خواهند بود.

✓ **جمع‌بندی تکنیک تزریق حافظه:**

حملات Fileless و تزریق کد به حافظه، به دلیل عدم ایجاد فایل و قابلیت پنهان‌کاری بسیار بالا، یکی از موفق‌ترین روش‌های دور زدن EDR محسوب می‌شوند.
مدیران امنیت باید به‌طور خاص روی **Memory Visibility** و تحلیل حافظه تمرکز داشته باشند.

۲,۳ تکنیک سوم: استفاده از Direct Syscalls وUnhooking

عبور از لایه‌های دفاعی EDR با دسترسی مستقیم به هسته سیستم عامل

مفهوم تکنیک Direct Syscalls وUnhooking

بسیاری ازEDR ها با استفاده از **API Hooking**، توابع حساس سیستم‌عامل مثلCreateProcess ، VirtualAllocو ReadFile را زیر نظر دارند تا رفتارهای مشکوک را شناسایی کنند.

در این تکنیک، مهاجم به جای استفاده ازAPI های معمولی، مستقیماً به **System Calls (Syscalls)** در سطح کرنل متصل می‌شود. به این ترتیب، تمامیHook هایی که توسط EDR در سطح کاربر اعمال شده‌اند، بی‌اثر می‌شوند.

همچنین در تکنیکUnhooking ، مهاجم عملاًHook های EDR را از بین می‌برد؛ یعنی بخش‌هایی از حافظه که EDR تغییر داده را به حالت اولیه بازمی‌گرداند تا EDR دیگر نتواند فعالیت او را کنترل کند.

چرا این تکنیک مؤثر است؟

- ✓ **عبور کامل از نظارتEDR :**چون مستقیماً با هسته سیستم‌عامل تعامل دارد، EDR قادر به مشاهده آن نیست.
- ✓ **پیچیدگی بسیار بالا:** پیاده‌سازی این تکنیک نیازمند دانش عمیق از معماری سیستم‌عامل است.
- ✓ **سکوت کامل در برابر ابزارهای امنیتی معمولی:** بسیاری از ابزارهای امنیتی، چنین فعالیت‌هایی را اساساً نمی‌بینند.

نحوه اجرایDirect Syscalls

۱. شناسایی Syscall Number توابع هدف (مثلاً NtAllocateVirtualMemory یا NtWriteVirtualMemory).
۲. ایجاد یک Stub در حافظه که Syscall مورد نظر را با دستور `syscall` مستقیماً فراخوانی کند.
۳. اجرای تابع به‌صورت مستقیم بدون عبور ازAPI های سطح کاربر.

روش‌های متداولUnhooking

- ✓ **Restoring Original Bytes :** بازگردانی بایت‌های اصلی توابع حساس در حافظه فرایند به حالت اولیه.
- ✓ **Mapping Clean DLLs :** بارگذاری یک نسخه تمیز ازDLL های حیاتی (مثلاًntdll.dll) از دیسک و جایگزین کردن آن‌ها با نسخه در حافظه.
- ✓ **Inline Hook Removal :** حذف مستقیمHook های Inline شده توسط EDR از حافظه.

نمونه‌های واقعی از حملات Direct Syscalls و Unhooking

ابزار / بدافزار	تکنیک استفاده شده	جزئیات حمله
SysWhispers	Direct Syscalls	ایجاد کد مخرب برای دسترسی مستقیم به Syscalls بدون نیاز به API استاندارد
Cobalt Strike	Unhooking & Direct Syscalls	استفاده از تکنیک unhooking برای دور زدن EDR هنگام اجرای shellcode
Sliver C2	Syscall-based Execution	استفاده از Direct Syscall برای فرار از ابزارهای تحلیل امنیتی

ابزارهای معروف در این حوزه

- ✓ **SysWhispers / SysWhispers2**
- ✓ **InlineWhispers**
- ✓ **Heavens Gate Techniques**
- ✓ **DIY Custom Syscall Stubs**

چرا این تکنیک برای EDR ها کابوس است؟

- ✓ مستقیماً با **Kernel Transition Mechanisms** در ویندوز تعامل دارد.
- ✓ شناسایی و تحلیل آن نیازمند ابزارهای بسیار تخصصی در سطح Kernel است.
- ✓ مهاجمان می‌توانند تقریباً هر فعالیتی را بدون شناسایی انجام دهند، از تزریق حافظه گرفته تا خواندن و نوشتن فایل‌های محافظت شده.

راهکارها و اقدامات پیشگیرانه برای مدیران امنیت

اقدام	شرح
استفاده از EDR های پیشرفته با Kernel Sensors	برخی EDR های نسل جدید، نظارت در سطح Kernel دارند و می‌توانند Syscall های غیرعادی را شناسایی کنند.

فعالسازی HVCI (Hypervisor-Protected Code Integrity)	این قابلیت امنیتی ویندوز، اجرای کدهای مشکوک در حافظه را به شدت محدود می‌کند.
تحلیل Memory Dumps به‌صورت دوره‌ای	شناسایی unhooking و Syscall Stubs در Memory Dumpها توسط تیم Threat Hunting.
آموزش تیم SOC برای شناسایی Unhooking	آموزش تکنیک‌های شناسایی رفتارهای غیرعادی مثل بارگذاری ntdll.dll تمیز یا تغییر مستقیم حافظه.
کنترل سختگیرانه اجرای ابزارهای مخرب بالقوه	با محدودسازی ابزارهای شناخته‌شده مانندSysWhispers ، بخشی از حملات پیشگیری می‌شود.

نکته : تکنیک Direct Syscall و Unhooking یکی از خطرناک‌ترین روش‌های EDR Evasion است که نیاز به دفاع چندلایه دارد. بدون نظارت در سطحKernel ، شناسایی چنین حملاتی تقریباً غیرممکن است.

✓ جمع‌بندی تکنیک Direct Syscalls وUnhooking :

این تکنیک یک سطح بالای حمله را نمایندگی می‌کند که عملاً بسیاری از ابزارهای امنیتی رایج را بی‌اثر می‌سازد.
مدیران امنیت باید به سراغ ابزارهای **Kernel-based Monitoring**و **سخت‌افزار محور** بروند تا بتوانند این تهدید را مهار کنند.

۲. تکنیک چهارم: استفاده از ابزارهای قانونی سیستم(LOLBins)

وقتی ابزارهای قانونی به سلاح مخفی مهاجمان تبدیل می‌شوند

مفهوم تکنیکLOLBins

Living off the Land Binaries and Scripts یا به‌اختصار **LOLBins**، به تکنیکی گفته می‌شود که در آن مهاجم به جای استفاده از ابزارهای مخرب خارجی، از **ابزارهای قانونی موجود در سیستم عامل یا نرم‌افزارهای مجاز سازمانی** برای انجام حملات خود استفاده می‌کند.

در این روش، مهاجم به جای اجرای فایل‌های مخرب، از ابزارهای درونی مثل **PowerShell**، **MSHTA**، **CertUtil** و سایر ابزارهای بومی ویندوز برای:

- ✓ دانلودPayload
- ✓ اجرای کد مخرب
- ✓ تزریق به حافظه
- ✓ برقراری ارتباط باC2

استفاده می‌کند.

چرا این تکنیک بسیار محبوب است؟

- ✓ ابزارهای مورد استفاده، *امضاشده و مورد اعتماد سیستم عامل* هستند.
- ✓ EDRها به سختی می‌توانند این ابزارها را به عنوان تهدید طبقه‌بندی کنند.
- ✓ نیازی به ایجاد فایل‌های جدید روی دیسک نیست.
- ✓ این ابزارها معمولاً در بسیاری از سازمان‌ها برای کارهای قانونی استفاده می‌شوند، و مسدود کردن آن‌ها می‌تواند باعث اختلال شود.

ابزارهای رایج در حملاتLOLBins

ابزار	عملکرد در حمله
PowerShell.exe	اجرای اسکریپت‌ها، تزریقShellcode ، Download & Execute
MSHTA.exe	اجرای کدهای HTML Application (HTA) برای اجرای Remote Scripts
CertUtil.exe	دانلود فایل‌ها از اینترنت با عبور از فایروال‌ها وProxy ها
Rundll32.exe	اجرای توابع DLL به‌صورت غیرمستقیم
WMIC.exe	اجرای دستورات و اسکریپت‌ها از راه دور
Regsvr32.exe	بارگذاریDLL های مخرب با استفاده از اسکریپت‌هایCOM

نمونه‌های واقعی از حملاتLOLBins

گروه / بدافزار	ابزار استفاده‌شده	جزئیات حمله
APT32 (OceanLotus)	MSHTA, PowerShell	بارگذاری Shellcode از طریق HTA Remote Script
FIN7	CertUtil, Rundll32	دانلود و اجرای Backdoor با دور زدن فایروال‌ها
TA505	PowerShell, WMIC	تزریق کد مخرب و دستورات Remote با ابزارهای قانونی

چراEDR ها در برابر LOLBins چالش دارند؟

- ✓ ابزارهای LOLBins دارای **امضای دیجیتال مایکروسافت** هستند.
- ✓ به‌طور طبیعی در بسیاری از فرایندهای قانونی سیستم عامل استفاده می‌شوند.
- ✓ شناسایی دقیق، نیازمند **Contextual Analysis** و تحلیل دقیق رفتار است نه فقط تشخیص ابزار.

راهکارهای دفاعی و اقدامات پیشنهادی برای مدیران امنیت

اقدام	شرح
کنترل اجرای ابزارهای پرخطر با AppLockerیاWDAC	محدودسازی یا مشروط‌سازی اجرای ابزارهای حساس به کاربران خاص یا امضاهای خاص.
Monitoringدقیق باEDR	مانیتورینگ رفتار ابزارهای LOLBins از جمله پارامترهای اجرایی مشکوک.
ایجاد Rules سفارشی درEDR/SIEM	شناسایی رفتارهای غیرمعمول ابزارهای قانونی از طریق Correlation Ruleها.
آموزش تیم‌های داخلی	آموزش کاربران و تیم‌های IT درباره خطرات ابزارهای بومی و نحوه سوءاستفاده از آن‌ها.
Threat Huntingدوره‌ای بر روی استفاده ازLOLBins	شناسایی الگوهای غیرعادی استفاده از این ابزارها در سازمان.

نکته کلیدی: مسدودسازی کورکورانه ابزارهای قانونی ممکن است باعث اختلال در سیستم‌های سازمان شود؛ راهکار صحیح، **کنترل دقیق و مشروط‌سازی استفاده از این ابزارها** است، نه حذف یا غیرفعال‌سازی کامل آن‌ها.

✓ جمع‌بندی تکنیکLOLBins :

تکنیک‌های مبتنی بر ابزارهای قانونی، یکی از رایج‌ترین و خطرناک‌ترین روش‌های حملات روزمره هستند که بسیاری ازEDR ها نیز هنوز در مقابله با آن‌ها ضعیف عمل می‌کنند.
هوشمندسازی تحلیل رفتار و کنترل دقیق استفاده از این ابزارها، تنها راه مقابله مؤثر است.

۲,۵ تکنیک پنجم: Hijacking و Side-Loading فایل‌هایDLL

سو استفاده از مسیرهای بارگذاری DLL برای عبور ازEDR

مفهوم تکنیک DLL Hijacking وDLL Side-Loading

در سیستم‌عامل ویندوز، زمانی که یک برنامه DLL خاصی را فراخوانی می‌کند، سیستم از یک الگوریتم مشخص برای پیدا کردن و بارگذاری آن DLL استفاده می‌کند. اگر مهاجم بتواند **یک نسخه مخرب از همان DLL** را در مسیر جستجوی زودتر قرار دهد، سیستم به جای نسخه اصلی، DLL مخرب را بارگذاری می‌کند.

در حملاتDLL Side-Loading ، مهاجم یک برنامه قانونی و امضاشده را همراه با DLL مخرب در همان مسیر اجرا می‌کند. چون فایل اصلی امضاشده است و توسط EDR به عنوان فایل امن شناخته می‌شود، فرآیند بارگذاری DLL مخرب بدون هشدار انجام می‌شود.

تفاوت بین DLL Hijacking وSide-Loading

تکنیک	تفاوت اصلی
DLL Hijacking	سوءاستفاده از مسیرهای جستجوی ناقص DLL برای بارگذاری نسخه مخرب از آن
DLL Side-Loading	سوءاستفاده از برنامه‌های امضاشده برای بارگذاری DLL مخرب در همان مسیر

چرایی محبوبیت این تکنیک

- ✓ استفاده از **برنامه‌های قانونی و امضاشده** برای اجرای DLL مخرب.
- ✓ سخت بودن تشخیص توسط EDR چون Process اصلی قانونی است.
- ✓ کارکرد بدون نیاز به آسیب‌پذیری در سیستم‌عامل؛ صرفاً نیاز به اشتباه در بارگذاریDLL ها.
- ✓ عبور از سیاست‌های اجرا، چون فایل‌های امضاشده اغلب از محدودیت‌های AppLocker عبور می‌کنند.

نمونه‌های واقعی از حملات DLL Hijacking وSide-Loading

گروه / بدافزار	تکنیک استفاده‌شده	شرح حمله
APT41	DLL Side-Loading	استفاده از برنامه‌های دیجیتالی امضاشده برای بارگذاری DLLهای مخرب
Lazarus Group	DLL Hijacking	حملات هدفمند با سوءاستفاده از مسیرهای بارگذاری ناقص DLL
PlugX	DLL Side-Loading	استفاده از برنامه‌های محبوب چینی برای بارگذاری ماژول‌های مخرب در حافظه

ابزارها و فریم‌ورک‌های مرتبط

- ✓ **Hijack Hunter**
- ✓ **Process Monitor (ProcMon)**

- ✓ **PE-Sieve**
- ✓ **Red Team Tools for DLL Hijacking Discovery**

چراEDR ها در برابر این تکنیک چالش دارند؟

- ✓ فرایند اجرایی، امضاشده و کاملاً قانونی است.
- ✓ شناسایی رفتار مخرب، نیازمند تحلیل دقیق فایل‌های بارگذاری شده و مسیرهای آن‌ها است.
- ✓ برخی ابزارهای تحلیل EDR فقط به شناسایی Process نگاه می‌کنند و بارگذاریDLL ها را بررسی نمی‌کنند.

اقدامات دفاعی و راهکارهای پیشنهادی برای مدیران امنیت

اقدام	شرح
بررسی دقیق مسیرهای بارگذاری DLL	اسکن مسیرهای بارگذاری و حذف فایل‌های مشکوک از پوشه‌های حساس.
مانیتورینگ بارگذاریDLL ها در EDR/SIEM	فعالسازی دقیق Event Logging مربوط به بارگذاریDLL ها، مخصوصاً در پوشه‌های غیرمعمول.
تحلیل‌های دوره‌ای با ابزارهای تخصصی	استفاده از ابزارهایی مانند PE-Sieve برای شناساییDLL های مشکوک در حافظه.
پیاده‌سازی AppLocker یا WDAC	محدودسازی اجرای DLL از مسیرهای خاص یا توسط برنامه‌های خاص.
آگاهی‌بخشی به تیم‌ها درباره این تکنیک	آموزش تیم‌های داخلی برای شناسایی نشانه‌های حملات DLL Hijacking و Side-Loading.

✓ جمع‌بندی تکنیک DLL Hijacking وSide-Loading :

این تکنیک همچنان یکی از موفق‌ترین روش‌های EDR Evasion است، مخصوصاً در حملات هدفمند. مهاجمان از امضای دیجیتال و مسیرهای بارگذاری ناقص به عنوان سلاح مخفی استفاده می‌کنند؛

۲,٦ تکنیک ششم: Tampering و غیرفعال‌سازی مستقیمEDR

وقتی مهاجم، خود EDR را از میدان خارج می‌کند

مفهوم تکنیکTampering

در تکنیکTampering ، مهاجم به‌صورت مستقیم به غیرفعال‌سازی یا نابودسازی اجزای EDR یا آنتی‌ویروس می‌پردازد تا بتواند بدون نظارت، Payloadنهایی خود را اجرا کند. در این حمله معمولاً به صورت زیر عمل می‌شود:

- ✓ توقف سرویس‌ها و پردازش‌هایEDR
- ✓ حذف فایل‌های حیاتی وLibrary هایEDR
- ✓ تغییر Registry یاPolicy های سیستم برای غیرفعال‌سازی ماژول‌های امنیتی
- ✓ دستکاری تنظیمات EDR برای خاموش کردن لاگ‌برداری و اسکن‌ها

چرا این تکنیک بسیار خطرناک است؟

- ✓ مهاجم مستقیماً به لایه‌های امنیتی حمله می‌کند و راه را برای دیگر تکنیک‌ها باز می‌سازد.
- ✓ به محض موفقیت، سیستم عملاً کور می‌شود و قادر به ثبت یا گزارش فعالیت‌های مهاجم نیست.
- ✓ تشخیص و بازگشت به حالت امن بسیار دشوار می‌شود، مخصوصاً اگر مهاجم سایر اجزا مثل Windows Defenderرا هم از کار بیندازد.

روش‌های رایجTampering

روش	توضیح
Service Stop / Kill Process	متوقف کردن سرویس‌های حیاتی EDR از طریق ابزارهای داخلی ویندوز مثل `sc`یا `taskkill`.

Registry Manipulation	تغییر کلیدهای Registry برای غیرفعال کردنAgent ها یا جلوگیری از.Auto-Start
EDR Uninstall or Corruption	سوءاستفاده از ابزارهای رسمی یا آسیب‌پذیری‌ها برای حذف کامل EDRیا خراب کردن فایل‌های آن.
EDR Configuration Tampering	تغییر تنظیمات EDR برای غیرفعال کردن بخش‌هایی مثل Memory Protectionیا.Network Protection

ابزارها و تکنیک‌های متداول در حملات Tampering

- **EDR Killer Tools :**ابزارهای آماده برای غیرفعال‌سازی سریع انواعEDR ها (غالباً در دارک‌وب عرضه می‌شوند)
- **PowerShell Scripts :**اسکریپت‌هایی برای توقف سرویس‌های امنیتی و حذف تنظیمات امنیتی
- **Manual Techniques :**استفاده از ابزارهای استاندارد ویندوز مثل:

  - ✓ `sc stop`
  - ✓ `taskkill /F`
  - ✓ `net stop`
  - ✓ تغییر مستقیمRegistry

نمونه‌های واقعی از حملات Tampering

گروه / بدافزار	روش استفاده‌شده	شرح حمله
LockBit	EDR Killing Tools	ابزار ویژه برای کشتن سرویس‌های EDR پیش از رمزگذاری
Black Basta	Registry Tampering & PowerShell	تغییر کلیدهای رجیستری و خاموش‌سازی Agentهای امنیتی
Conti	Uninstall Scripts & Process Killing	حذف کامل EDR و خاموش کردن تمام فرآیندهای امنیتی قبل از حمله

چرا Tampering برایEDR ها تهدیدی بزرگ است؟

- ✓ برخی سرویس‌های EDR فاقد **Self-Defense** کافی هستند.
- ✓ ابزارهای ویندوزی مثلPowerShell ، `sc`و `reg`خود در سیستم وجود دارند و مسدودسازی کامل آن‌ها دشوار است.
- ✓ اگر مهاجم دسترسی Local Admin بگیرد، بسیاری ازEDR ها دیگر مقاومتی ندارند.
- ✓ خاموش کردن EDR باعث ناپدید شدن سایر تکنیک‌های حمله درLog ها می‌شود.

اقدامات دفاعی و راهکارهای پیشنهادی برای مدیران امنیت

اقدام	شرح
استفاده ازEDR های دارای-Self Defenseقوی	برخیEDR ها دارای ماژول‌هایی هستند که از توقف یا حذف سرویس‌هایشان جلوگیری می‌کنند.
محدودسازی دسترسی به PowerShell و ابزارهای مدیریتی	ایجاد سیاست‌های سخت‌گیرانه برای کاربران در خصوص اجرای ابزارهای حساس.
مانیتورینگ دقیق توقف سرویس‌های امنیتی	بررسی لاگ‌ها و آلارم‌دهی فوری در صورت توقف سرویس‌های حیاتی امنیتی.
فعالسازیTamper Protection	فعال‌سازی قابلیت‌های پیش‌فرض Tamper Protection در EDR و.Windows Defender
تقویت سیاست‌های Privileged Access Management	محدود کردن شدید دسترسی به حساب‌های Admin و Local System.

نکته کلیدی مدیریتی

حمله به خود EDR یعنی حمله به آخرین لایه دفاعی شما.

هر سازمان باید در کمترین زمان ممکن متوجه غیرفعال‌سازی EDR شود؛ وگرنه مهاجم عملاً آزادانه در شبکه حرکت خواهد کرد.

✓ **جمع‌بندی تکنیکTampering :**

Tamperingبه معنای حذف کامل ابزارهای امنیتی از میدان است و معمولاً در مراحل نهایی حمله رخ می‌دهد.

تنها راه مقابله با آن، استفاده ازEDR های پیشرفته با **Self-Defense** قوی، نظارت لحظه‌ای بر سرویس‌ها و محدود کردن دسترسی به ابزارهای مدیریتی است.

بخش پایانی: جدول مقایسه + چک‌لیست اجرایی

جدول مقایسه جامع تکنیک‌ها

تکنیک	پیچیدگی	موفقیت در عبور ازEDR	نیاز به فایل روی دیسک	میزان رایج بودن	ابزارهای رایج
Obfuscation & Packing	پایین تا متوسط	متوسط	دارد	بسیار رایج	UPX, Themida, ConfuserEx
In-Memory Injection & Fileless	بسیار بالا	بسیار بالا	ندارد	بسیار رایج	Cobalt Strike, PowerShell Empire
Direct Syscalls & Unhooking	بسیار بالا	بسیار بالا	ندارد	متوسط به بالا (در APTها)	SysWhispers, InlineWhispers
LOLBins	متوسط	بالا	ندارد یا کم	بسیار رایج	PowerShell, MSHTA, CertUtil
DLL Hijacking & Side-Loading	متوسط	بالا	دارد	رایج در حملات هدفمند	PE-Sieve, Hijack Hunter
Tampering (EDR Disabling)	بسیار بالا	بسیار بالا	ندارد یا کم	رایج در حملات Ransomware	EDR Killer Tools, PowerShell Scripts

چک‌لیست اجرایی برای مدیران امنیت(EDR Evasion Defense Checklist)

🎯 هدف این چک‌لیست:

کمک به مدیران امنیت شبکه برای تقویت دفاع در برابر تکنیک‌های پیچیده دور زدنEDR .

اقدام کلیدی	جزئیات و توصیه‌ها
فعالسازی-Memory Scanning Real Time	بررسی فعال بودن کامل ماژول‌های Memory Scanningدر.EDR
تحلیل عمیق بارگذاریDLL ها	استفاده از ابزارهای تخصصی مثل PE-Sieve برای شناسایی DLL های مشکوک.
فعال‌سازی Tamper Protection در EDR وWindows Defender	جلوگیری از توقف یا تغییر تنظیمات ابزارهای امنیتی.
سخت‌گیری بر ابزارهای LOLBins با AppLocker/WDAC	کنترل دقیقPowerShell ، MSHTA، WMIC و... با سیاست‌های اجرای سخت‌گیرانه.
استفاده از EDR با قابلیت Kernel-level Monitoring	بهره‌گیری از ابزارهایی که رفتار در سطح Kernel را هم شناسایی می‌کنند.
پیاده‌سازی Threat Hunting منظم	شکار تهدیدات دوره‌ای با تمرکز بر رفتارهای حافظه، LOLBinsو بارگذاریDLL های مشکوک.
مانیتورینگ توقف سرویس‌های امنیتی و EDR درSIEM	ایجاد آلارم‌های فوری برای هرگونه توقف غیرمنتظره سرویس‌های حیاتی امنیتی.
تقویت کنترل حساب‌هایAdmin	محدودسازی شدید دسترسی به حساب‌های دارای دسترسی بالا.
آموزش تیم‌های SOC وIT Security	آموزش تخصصی درباره تکنیک‌های EDR Evasionو روش‌های شناسایی آن‌ها.

جمع‌بندی:

تکنیک‌های دور زدن EDR روزبه‌روز پیچیده‌تر می‌شوند و مهاجمان با ترکیب چندین روش، حملاتی غیرقابل‌شناسایی طراحی می‌کنند.

مهم‌ترین ضعف سازمان‌ها، **تمرکز بیش از حد برEDR های سنتی و غفلت از لایه‌های تکمیلی دفاعی** است. EDRبه‌تنهایی کافی نیست؛ سازمان‌های پیشرو از ترکیب ابزارهای زیر استفاده می‌کنند:

- ✓ **EDR پیشرفته با Self-Defense قوی و Memory Scanning فعال**
- ✓ **SIEM با مانیتورینگ رفتار محور(Behavioral Monitoring)**
- ✓ **Threat Hunting منظم**
- ✓ **سیاست‌های سخت‌گیرانه اجرای فایل و اسکریپت‌ها**

✓ **نکته کلیدی:**

اگر سازمان شما هنوز به سطحی نرسیده که بتواند رفتارهای پیچیده در حافظه، مسیرهای بارگذاری DLL و ابزارهای LOLBins را شناسایی کند، در برابر این حملات عملاً بدون دفاع است.

توصیه ویژه ما به شما این است که به‌جای تلاش برای حذف کامل این تهدیدات (که تقریباً غیرممکن است)، روی **کاهش زمان شناسایی (Mean Time To Detect – MTTD)** و **حداکثر کردن پاسخ سریع (Incident Response Readiness)**تمرکز کنید.