T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

WINDOWS İŞLETIM SISTEMI KALINTILARININ ANALIZINI GERÇEKLEŞTIREN SIBER OLAY MÜDAHALE ARACININ GELIŞTIRILMESI VE OLAY ZAMAN ÇIZELGESININ ÇIKARILMASI

ALİ HÜSAMİDDİN UÇAR

Yüksek Lisans Tezi

ADLI BILIŞIM MÜHENDISLIĞI ANABİLİM DALI

AĞUSTOS 2021

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Adli Bilişim Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

WINDOWS İŞLETIM SISTEMI KALINTILARININ ANALIZINI GERÇEKLEŞTIREN SIBER OLAY MÜDAHALE ARACININ GELIŞTIRILMESI VE OLAY ZAMAN ÇIZELGESININ ÇIKARILMASI

Tez Yazarı
ALİ HÜSAMİDDİN UÇAR

Danışman
Doç. Dr. Fatih ERTAM

AĞUSTOS 2021
ELAZIĞ

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Adli Bilişim Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Başlığı:	Windows İşletim Sistemi Kalıntılarının Analizini Gerçekleştiren Siber Olay Müdahale Aracının Geliştirilmesi ve Olay Zaman Çizelgesinin Çıkarılması
Yazarı:	ALİ HÜSAMİDDİN UÇAR
İlk Teslim Tarihi:	05.07.2021
Savunma Tarihi:	09.08.2021

TEZ ONAYI

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına göre hazırlanan bu tez aşağıda imzaları bulunan jüri üyeleri tarafından değerlendirilmiş ve akademik dinleyicilere açık yapılan savunma sonucunda OYBİRLİĞİ ile kabul edilmiştir.

		İmza
Danışman:	Doç. Dr. Fatih ERTAM Fırat Üniversitesi, Teknoloji Fakültesi	Onayladım
Başkan:	Dr. Öğr. Üyesi Mustafa KAYA Fırat Üniversitesi, Teknoloji Fakültesi	Onayladım
Üye:	Dr. Öğr. Üyesi Aykut DİKER Bandırma Onyedi Eylül Üniversitesi, Mühendislik Fakültesi	Onayladım

Bu tez, Enstitü Yönetim Kurulunun ...../...../20..... tarihli toplantısında tescillenmiştir.

İmza

Doç. Dr. Kürşat Esat ALYAMAÇ
Enstitü Müdürü

BEYAN

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım "Windows İşletim Sistemi Kalıntılarının Analizini Gerçekleştiren Siber Olay Müdahale Aracının Geliştirilmesi ve Olay Zaman Çizelgesinin Çıkarılması" Başlıklı Yüksek Lisans Tezimin içindeki bütün bilgilerin doğru olduğunu, bilgilerin üretilmesi ve sunulmasında bilimsel etik kurallarına uygun davrandığımı, kullandığım bütün kaynakları atıf yaparak belirttiğimi, maddi ve manevi desteği olan tüm kurum/kuruluş ve kişileri belirttiğimi, burada sunduğum veri ve bilgileri unvan almak amacıyla daha önce hiçbir şekilde kullanmadığımı beyan ederim.

09.08.2021

ALİ HÜSAMİDDİN UÇAR

ÖNSÖZ

Windows işletim sistemlerinin analizlerinde birden çok yöntem bulunmaktadır. Bu yöntemler ile birbirinden farklı veriler elde edilmekte ve incelemeci olarak zaman kavramı uzun sürmektedir. Bu tez çalışmasında bir windows işletim sisteminin analizinde, analizci için inceleme zamanının önemi düşünülerek bir vakanın en hızlı şekilde incelenmesi yöntemlerinden bahsedilmektedir.

Bu tez çalışmasında tamamlanmasında yardımlarını esirgemeyen danışmanım Sayın Doç. Dr. Fatih ERTAM'a, Fırat Üniversitesi Adli Bilişim Mühendisliği'nde görevli hocalarıma ve desteğini esirgemeyen aileme teşekkür ederim.

ALİ HÜSAMİDDİN UÇAR
ELAZIĞ, 2021

İÇINDEKILER

ÖZET

Windows İşletim Sistemi Kalıntılarının Analizini Gerçekleştiren Siber Olay Müdahale Aracının Geliştirilmesi ve Olay Zaman Çizelgesinin Çıkarılması

ALİ HÜSAMİDDİN UÇAR

Yüksek Lisans Tezi

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Adli Bilişim Mühendisliği Anabilim Dalı

Ağustos 2021, Sayfa: xiii + 43

Windows işletim sisteminin dünya genelinde günden güne kullanımı yaygınlaşmaktadır. Bununla birlikle, windows işletim sistemine ait sürümler gelişmekte ve güncellenmektedir. Windows'un yaygın olarak kullanılan sürümlerinde suç teknikleri ve türevleri sürekli değişmekte, saldırganların kurbanlarını etkisi altına alması hızla artmaktadır. Bu zaman zarfında kurbanların makinelerindeki adli analiz ve adli kopya yöntemlerinin gerçekleştirilmesi uzun sürdüğü için saldırganların tespiti ve saldırının önlenmesi zaman kaybettirmektedir. Windows işletim sistemleri olarak makine üzerinde gerçekleşen süreçleri, kullanıcı aktivitelerini, dosya sistem ve kayıt defteri kayıtlarını, uygulama ve kullanıcı kalıntılarını tespit etmek için güncel yöntemler, araç-gereçler kullanılarak zaman kaybı azaltılmaya çalışılmaktadır. Windows işletim sistemlerinde adli kopya alma ve alınan kopyaların analizinde kullanılan araçlarının genellikle ticari yükünün olması, alanın geliştirilebilirliği yönünde engel arz etmekte ve incelemeleri geciktirmektedir. Bu sebeple açık kaynak kodlu yazılımlar ile kurbanın dahil olduğu olayları doğrudan analiz edebilmeyi sağlayan log kayıtlarının incelenebilmesi zaman ve ticari boyut açısından kayıpları azaltılabilmektedir. Aynı zamanda elde edilen log kayıtları ile saldırı zaman çizelgesi oluşturularak saldırının verdiği zarar gözlemlenebilmektedir. Bu tez çalışmasında Windows işletim sistemlerinde saldırganların genellikle iz bıraktıkları log kayıtları ve kalıntıların toplanmasının ardından ayrıştırılması ve ardından olay zaman çizelgesinin çıkarılması sunulmuştur. Kalıntıların ayrıştırılması için güncel yöntemlerin yanı sıra açık kaynak kodlu siber olay müdahale aracının geliştirilmesi gerçekleştirilmiştir. Ayrıştırılan kalıntıların, olay-zaman çizelgesi oluşturularak saldırgan-kurban ilişkisinin kayıt çizelgesinden bahsedilmektedir. Bu tez çalışmasında geliştirilen ve IR-Parser adı verilen yazılımın adli bilişim ve siber güvenlik alanında çalışan insanlara faydalı olacağı düşünülmektedir.

Anahtar Kelimeler: Adli Bilişim, Windows Adli Bilişim, Windows Adli Kopya, Windows Dosya Sistemi analizi, Dosya Sistem Kalıntıları

ABSTRACT

Development Of Cyber Incident Response Tool That Performs Analysis Of Windows Operating System Remnants And Extraction Of Event Timeline

ALİ HÜSAMİDDİN UÇAR

Master's Thesis

FIRAT UNIVERSITY
Graduate School of Natural and Applied Sciences

Department of Digital Forensic Engineering

August 2021, Pages: xiii + 43

The use of the Windows operating system is increasing day by day around the world. However, versions of the windows operating system are developing and being updated. Crime techniques and variants are constantly changing in widely used versions of Windows, and attackers' impact on their victims is increasing rapidly. During this time, forensic analysis and forensic copy methods on the victims' machines take a long time to perform, so detecting the attackers and preventing the attack is time consuming. Windows operating systems are trying to reduce time loss by using up-to-date methods and tools to detect processes, user activities, file system and registry records, application and user residues on the machine. In Windows operating systems, the commercial load of the tools used in forensic copying and analysis of the obtained copies poses an obstacle to the development of the field and delays the investigations. For this reason, the loss of time and commercial dimension can be reduced by examining the log records, which enable to analyze the events involving the victim directly, with open source software. At the same time, the damage caused by the attack can be observed by creating an attack timeline with the log records obtained. In this thesis, it is presented that the log records and residues that attackers usually leave traces in Windows operating systems are collected and then parsed and then the event timeline is extracted. In addition to current methods for separating residues, the development of an open source cyber incident response tool was carried out. The record schedule of the aggressor-victim relationship is mentioned by creating an event-timeline of the separated remains. It is thought that the software developed in this thesis study, called IR-Parser, will be beneficial to people working in the field of forensic information and cyber security.

Keywords: : Digital Forensics Windows Forensics, Windows Forensic Copy, Windows File System Analysis, File System Artifact

ŞEKİLLER LİSTESİ

Sayfa

TABLOLAR LİSTESİ

Sayfa

SİMGELER VE KISALTMALAR

Simgeler Kısaltmalar

BAGMRU : ShellBags Registry (ShellBags Kayıt Defteri)
BAM : Background Activity Moderator (Arka Plan Etkinliği Moderatörü)
CMD : Command Line / Komut Satırı Derleyicisi
CSOC : Cyber Security Operation System (Siber Güvenlik Operasyon Merkezi)
CSV : Comma-Separated VariabIes (Virgülle Ayrılmış Veriler)
IR : Incident Response (Olay Müdahalesi)
IOC : Indicator Of Compromise (Uzlaşma Göstergesi)
JSON : JavaScript Object Notation (JavaScript Nesne Gösterimi)
LNK : Windows Shortcut (Windows Kısayolları)
MFT : Master File Tablo / Ana Dosya Tablosu
MRU : Most Recently Used Process (En Son Yapılan İşlemler)
MSİECF : Microsoft Internet Explorer cache (Microsoft Internet Explorer Önbelleği)
NTFS : New Technology File System (Yeni Teknoloji Dosya Sistemi)
SAM : Security Account Management / Güvenlik Hesapları Yöneticisi
SCCM : System Center Configuration Manager (Sistem Merkezi Yapılandırma Yöneticisi)
SI : Standard Information (Genel / Standart Bilgi)
SRUM : System Resource Usage Monitor / Sistem Kaynağı Kullanım İzleyicisi
PE : Process Execution / Çalıştırılabilir İşlem
PDF : Portable Document Format (Taşınabilir Belge Formatı)
RAM : Random Acces Memory / Rastgele Erişim Belleği
RB : Recycle Bin / Silinen Dosya Dizini
RDP : Remote Desktop Protocol / Uzak Masaüstü Protokolü
Vb. : Ve Benzeri
WEBHİST : Internet Web History (Internet Geçmişi)
WİNEVT : Windows EventLog (EVT) files. (Windows Olay Günlüğü Dosyası)
WİNİİS : Microsoft IIS log files (Microsoft IIS Günlük Dosyaları),
WİNJOB : Windows Scheduled Task Job Files (Windows Zamanlanmış Görev İş Dosyaları)
WİNLOGON : Windows log-on Registry (Windows Oturum Açma Kayıt Defteri)
WİNREG : Windows NT Registry (Windows Kayıt Defteri)
XLSX : Excel Open XML Spreadsheet (Excel Açık XML Elektronik Tablosu)

1. GİRİŞ

Microsoft windows işletim sistemleri dünyadaki diğer işletim sistemlerine karşı en çok kullanılan sistem olarak varlığını korumaya devam etmektedir. Günümüzde kullanılan bilgisayarların %90'dan fazlasının windows işletim sisteminin herhangi bir sürümünü kullandığı gözlemlenmiştir. Böylelikle windowsun kullanım kolaylığı, bilgisayar suçlarının çoğalmasına sebep olmaktadır. Windows işletim sistemi ilk olarak 1985 de "Windows 1.0" olarak kullanılmaya başlanıldı. Ardından windowsun sürümleri olarak 98, XP, Vista, 7, 8 ve günümüzde aktif olarak kullandığımız 10 sürümü ile kullanıcılara hizmet vermeye devam etti.

Windows işletim sistemi kurulduğundan bu yana gelişip güncellenmekte ve kullanıcıların yaptığı aktiviteler sürümden sürüme farklılık göstermektedir. Windows da sürüm ilerledikçe log kayıtları, kalıntılar, kullanıcıların aktivite kayıtları vb. gibi kayıt alanları artmakta ve gelişmektedir. Çeşitli kuruluş ve şirket sistemlerinin bilişim bilgilerini siber saldırılardan korumak amacıyla gelişmiş bilgi teknolojileri kullanılmaktadır. Aynı zamanda bilgisayar sistemlerini korumak için şirkete ait siber güvenlik politikalarının düzenlemesine ihtiyaç vardır. Virüsler, kimlik avı, Truva atları, solucanlar, yasadışı erişim (örneğin, fikri mülkiyetin veya gizli bilgilerin çalınması) ve kontrol sistemi saldırıları gibi çeşitli siber saldırılar mümkündür [1]. Windows geliştikçe, adli analiz yapmak zorlaşmakta ve analiz sürecinin daha da detaylandırılması gerekmektedir. Bu durum gelişmiş adli analiz yapmayı zorunlu kılmaktadır. Gelişmiş adli analiz yapmanın önemli bir yönü, önyükleme işlemi, dosya oluşturma, silme ve çıkarılabilir depolama ortamının kullanımı gibi windows işletim sistemlerindeki temel işlemlerin altında yatan mekanizmaları anlamaktır. Ek olarak, windows işletim sistemlerinde çeşitli yöntemlerle verilerin elde edilmesi gerçekleştirilmekte ve bu verilerin birbirleriyle ilişkilendirilmesiyle "büyük resim" (örneğin, genel bir kullanıcı eylemi teorisi ve bir zaman çizelgesi) ortaya çıkarılmaktadır.

Analistler çalışırken çoğunlukla gizli olan ve saldırganların izlerini silmek için tasarlanmış kötü programlarla (örneğin, kötü amaçlı yazılım ve fidye yazılımı) enfekte olmuş bilgisayar sistemleriyle karşı karşıya kalırlar. Windows adli analiz yaparken toplanması gereken büyük miktarda veriyi görmek oldukça zor olabilir. Adli analiz sırasında, analistin planlı çalışmaması durumunda tüm süreç iki kat daha zor hale gelebilmektedir. Bu nedenle analistler, vakalarda doğrudan adli kopya almak yerine zamandan kar etmek amacıyla windows kalıntı ve izlerini kullanmayı hedef almışlardır. Böylelikle kullanıcının makinesinde uçucu verileri yakalayabilecek, medya görüntülerini oluşturabilecek, dosya sistemlerini, ağ trafiğini ve bellek analizini gerçekleştirebilecektir. Kullanıma hazır ücretsiz ve açık kaynaklı araçlarla bir kullanıcıların veya saldırganların bıraktığı izler incelenebilmektedir.

2. WİNDOWS İŞLETİM SİSTEMİ VE ADLİ BİLİŞİM

Teknolojinin özellikle son yirmi yılda artan hızla gelişmesine paralel olarak bilgisayarlar ve diğer elektronik cihazlar modern hayatın vazgeçilmezi haline gelmiştir. Bilgi işlem ve iletişim teknolojilerinin kullanım alanlarının artmasıyla kullanıcı sayısındaki artış suç işlemeyi amaç edinmiş kişilerin de dikkatini çekerek istismar edilebilecek teknik ve yöntemleri geliştirmeye yöneltmektedir [2]. Geliştirilen teknikler farklı saldırganlar tarafından kullanılsa da windows işletim sisteminde iz bıraktıkları yerler değişmemektedir.

Windows işletim sistemi kullanan bir sistemin adli analizi, adli kopyalar yoluyla veya kalıntıların analiziyle gerçekleştirilebilmektedir. Windows adli bilişimi de adli kopyalar alınarak toplanmış veriyi gerekli sınıflandırmalar kullanarak ayrıştırmaya ve kullanıcı-sistem aktivitelerini ortaya çıkarmaya yaramaktadır. Aynı zamanda adli analiz yapılan windows işletim sistemi hakkında veriler de elde edilmektedir. Bu aktiviteler kalıntılardan da çıkabilmektedir. Kalıntıların adli analizi, adli kopya alınarak yapılan adli analizden daha hızlı bir süreçte gerçekleştiği bilinmektedir. Log kalıntılarının adli analizi günlükler, kayıt defteri vb. gibi incelenecek nesnelerden oluşmaktadır. Uygulama düzeyi bilgileri, yaygın olarak kullanılan windows uygulamalarından çıkarılan kullanıcı giriş ve aktivite bilgileridir [3].

2.1. Windows Adli Bilişimi

İnternet artık günlük yaşamın bir parçası olmuş olup, çevrimiçi alışveriş, çevrimiçi bankacılık vb. alanlarda önemli bir rol oynuyor. Bununla birlikte, internet kullanımıyla ilgili güncel sorunlardan biri güvenlik eksikliğidir. Çünkü bir dinleyicinin aktarılan verileri yakalayabilmesi hala mümkündür [4]. Suçlular; dolandırıcılık, gasp, kara para aklama ve finansman faaliyetleri için finansal teknolojileri kötüye kullanmaktadır [5]. Kötü amaçlı yazılım türleri arasında casus yazılımlar, tuş kaydediciler, rootkit'ler ve reklam yazılımları bulunmaktadır [6]. Kuruluşların siber güvenlik olay müdahalesinde (Cyber Security Incident Responce-CSIR) analitik bilgileri geliştirmek, işlemek ve kullanmak için iş analitiğinden nasıl yararlandığı hakkında çok az şey bilinmektedir [7]. Günümüzde, banka işlemlerinde, alışverişlerde, okuldaki derslerde, bilgisayar oyunlarında, randevu ve devlet işlemlerinde, kurumlarda vb. hemen hemen her yaşantımızda bir dijitalleşme mevcuttur. Bu dijitalleşmenin evrenin büyümesiyle ve son sistem cihazların gelişmesiyle doğru orantılı olduğu görülmektedir. İnsanların dijital dünyada attıkları her adım, yaptıkları her işlev bir dijital kayıt altına alınmaktadır. Bu kayıtların tümüne dijital kanıtta denilebilir.

Dijital kanıt terimi hem dijital cihazları hem de cihazlarda bulunan ilgili bilgi parçalarını (verileri) kapsar. Dizüstü bilgisayar veya akıllı telefon gibi dijital cihazdan elde edilen veriler

sıklıkla tek başına kanıt olarak kabul edilir[8]. Bir dijital verinin herhangi bir şekilde dijital kanıt olarak görülebilmesi için hukuki eylemler ışığında istenilmesi durumunda erişilebilir olması ve işlenebilir olması gerekmektedir. Bu verilerin genel olarak incelenmesi ve işlenebilmesi eylemine "adli bilişim" de denilmektedir. Doğası gereği "adli" olduğu iddia edilse de, bilgisayar adli analizinde kullanılan temel süreç ve mekanizmaların canlı olay müdahale alanında nadiren uygulandığı bilinmektedir[9].

Windows işletim sistemine ait bir bilgisayarda kullanıcı aktivitelerinin ve sistem aktivitelerinin tutulduğu birden fazla dijital kayıt bulunmaktadır. Bu kayıtlara örnek olarak kayıt defteri, olay günlükleri, silinen dosya kayıtları, bilgisayarın açılış saati, takılan harici bellek bilgisi, ağ kayıtları ve bellek kayıtları gibi örnekler verilebilir. Bu kayıtlar suç durumunda, siber saldırı durumunda veya işlenebilecek farklı durumlar söz konusu olması durumunda incelenmesi gerekebilmektedir.

Temel olarak, dijital adli soruşturma süreçleri, nihai bulguları incelemek, analiz etmek ve ilgili paydaşlara sunmak için kanıtların belirlenmesi, toplanması ve elde edilmesini içerir [10]. Soruşturmalar ve yasal işlemlerin desteklenmesi amacıyla bilgisayarlarda, sunucularda, kişisel ve kurumsal depolama ortamlarında bulunan kanıtlarla ilgili tanımlama, koruma, toplama, analiz ve raporlama süreçlerini içeren adli bilişim, kanıtlar açısından en eski disiplindir. Adli bilişimin alt dallarının birçoğu zamanla bu disiplinden ayrılarak farklı bir alt disiplin oluşturmuşlardır [11]. Bu alt disiplinler, ağ adli bilişimi, bellek adli bilişimi, mobil cihaz adli bilişimi ve bilgisayar adli bilişimi olarak tanımlanmaktadır.

Windows işletim sisteminde adli bilişimin çeşitlerinden olan bellek kayıtlarının analizi de önemlidir. Bellek Kayıtları, uçucu veriler olarak bilinmektedir. Çalışan bir bilgisayarın Rasgele Erişim Belleğinden (RAM) kanıtların toplanması, aynı zamanda canlı kazanım olarak da bilinir. RAM içeriğinden, şifreleme anahtarları gibi adli olarak çok önemli bilgiler elde edilebilmektedir. Ayrıca bilgisayarlar yürüttükleri işlemler ile ilgili geçici bilgileri RAM içerisinde saklamaktadır. Bu nedenle RAM içerisinde kanıt olabilecek birçok kritik veriye ulaşılabilir [12]. Aynı zamanda depolama aygıtlarını izlemek, dijital adli tıpta önemli alanlardan biridir. Kayıt defteri, olay günlüğü veya IconCache (simge önbelleği) analizi ile ilgili mevcut yöntemler ve araçlar, gizli sızıntı, yasadışı kopyalama ve güvenlik olayı ile ilgili vakaları çözmeye yardımcı olur [13].

Windows uygulamalarında kullanıcı girdi faaliyetlerinin tanımlanması, adli dijital soruşturmada hayati hale geldi. Kullanıcı girdi bilgilerinin fiziksel bellekten çıkarılması, suç davalarında kanıt olarak kullanılabilecek yararlı bilgileri ortaya çıkarabilmektedir [14]. Bir işletim sistemine ait makinenin çevrimiçi olarak başka kaynaklar ile iletişim kurması da dijital kayıtlar altına alınmaktadır. Bu kayıtlar kanıt olarak kullanabilmesi için hukuki süreçler içerisinde dinlenmesi gerekmektedir. Bazı dijital kanıtlar fiziksel olmaktan ziyade çevrimiçi olabilmektedir. Bu tür delillerin korunmasına ilişkin web sitelerini içeren suçlarda bulunabilir. Ağ kayıtları

güvenlik saldırıları, izinsiz girişler, kötü amaçlı yazılım saldırılarının, anormal ağ trafiğinin ve güvenlik ihlallerinin kaynağını bulmak için ağ etkinliklerinin veya olaylarının izlenmesi, yakalanması, saklanması ve analizi süreçlerini içermektedir.

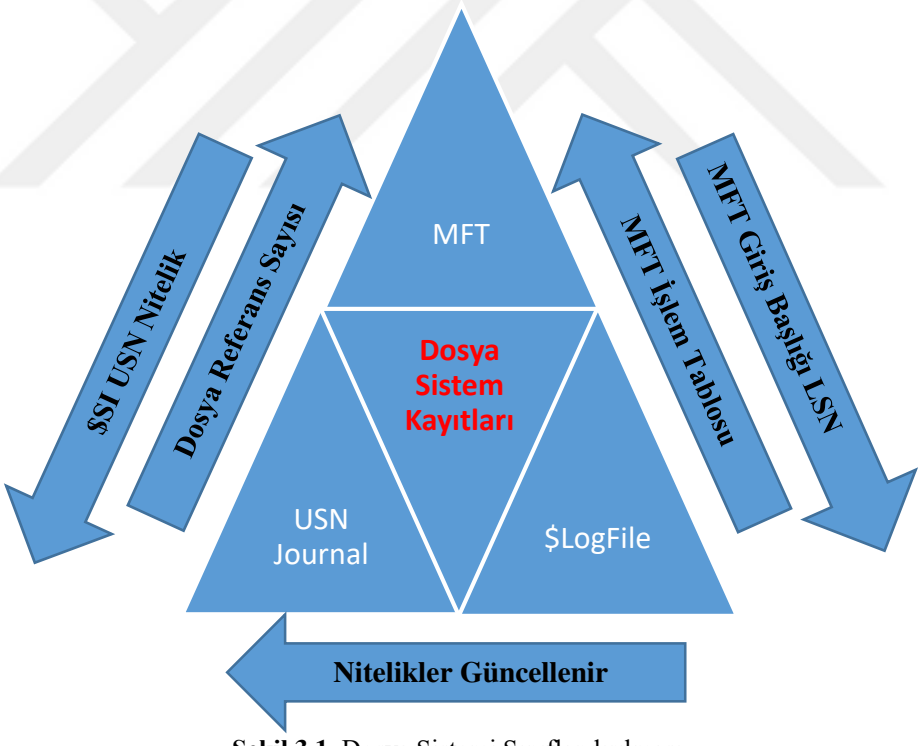
Adli bilişim analizlerini adli araç setleri olmadan manuel yapmak mümkün olmadığı gibi sadece adli araç setlerinin getirdiği sonuçlarla neticeye varmak da doğru değildir. Analiz işlemleri sırasında dijital kanıtları yorumlama işlemi de büyük önem arz etmektedir. Bu nedenle adli bilişim analizlerinde analiz yapacak uzman ile adli araç setlerinin bir bütün olarak birbirini tamamlamaları gerekmektedir. Hatta analizcinin soruşturmanın her safhasında bulunması ile ilgili öneriler bulunmaktadır [15]

3. WİNDOWS İŞLETİM SİSTEMİ LOG KAYITLARI VE SİSTEM KALINTILARI

Bu başlık altında windows işletim sisteminde bulunan log kayıtlarının ve sistem kalıntılarının neler olduğu, bu kalıntıların kullanım alanları, kalıntıların içerisinde hangi verilerin olduğu, logların ve kalıntıların birbirleriyle olan ilişkileri vb. konulardan bahsedilecektir.

3.1. Dosya Sistem Analizi

Dosya sistem analizi, adli analiz gerçekleştirilirken, kullanıcıdan veya sistemden silinen dosyaların, dosya dizin yapısının, sahipsiz dosyaların, gizli veya gizlenmiş dosyaların disk üzerindeki analizlerinin bütünüdür. Windows işletim sisteminde dosya sistem analizinde MFT, USNJOURNAL, LOGFİLE olmak üzere üç farklı sistem dosyaları analiz edilmektedir.
Şekil 3.1'de dosya sistemleri arasındaki bağlantı anlatılmaktadır.



Şekil 3.1. Dosya Sistemi Sınıflandırılması

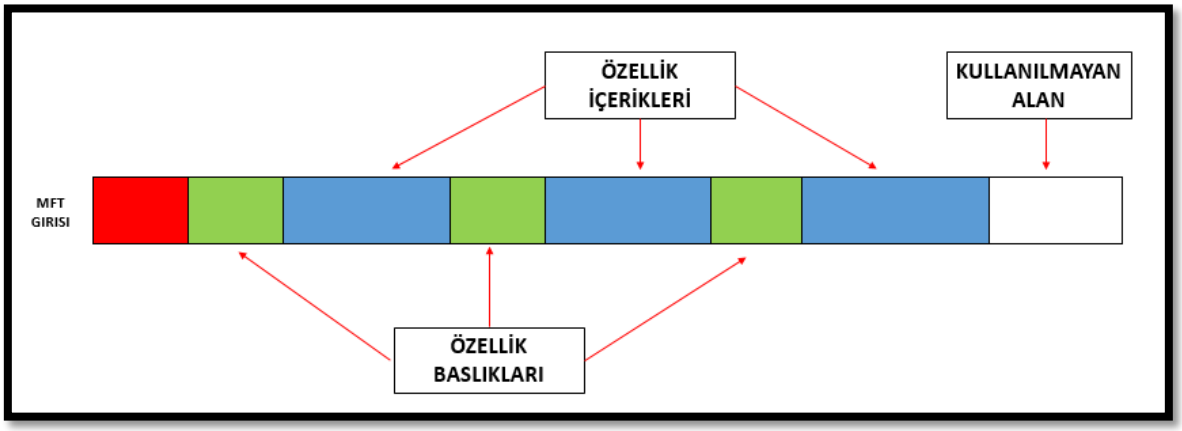
3.1.1. MFT (Master File Table) Kayıtları

Ana Dosya Tablosu ($ MFT), bir NTFS (New Technology File System) dosya sistemindeki en önemli dosyadır. Birimdeki tüm dosyaları, klasörlerdeki mantıksal konumlarını, donanımdaki fiziksel konumlarını ve dosyalarla ilgili meta verileri kapsamaktadır. Ana dosya tablosu (MFT) NTFS' in çekirdeğidir, çünkü birimdeki her dosya ve klasörün ayrıntılarını içerir [16].

- Dosyaların oluşturma tarihi,
- Değiştirme tarihi, erişim tarihi,
- Dosyaların son yazma tarihi,
- Dosyanın fiziksel ve mantıksal boyutu,
- Dosya izinleri güvenlik erişimi,

Örneğin; MFT' de 100 giriş varsa ve birimde bulunan dosya X silinirse, daha sonra yeni 1000 tane dosya oluşturulursa, oluşturulan dosya, silinen X dosyasına ait MFT kayıt alanının üzerine yazılmaktadır. Dosyanın içeriği sabit sürücüde bulunsa da, adı, meta verileri vb. içeren veriler MFT tablosunun üzerine yazılmaktadır. NTFS dosya sisteminin analiz edilebilmesi için, dosya sistemi hakkında verilerin fiziksel ve mantıksal konumları gibi önemli bilgiler araştırılmalı, birimin veri yapısının ve gizli verilerin incelenerek daha iyi deneyim elde edilmesi sağlanabilir. [17].
MFT 'de 10.000 giriş varsa, 1.000 silinir ve 2 yeni dosya hemen sürücüye eklendiği varsayılırsa, disk içerisindeki 998 giriş kurtarılabilir olduğu düşünülebilir. Silinen dosyaların kurtarılıp kurtarılamaması, yeni eklenen dosyaların üzerine yazılıp yazılmadıklarına bağlı olabilmektedir.

Şekil 3.2'de MFT tablosunda bulunan veriler ve bu verilerin sıralaması hakkında bilgi verilmektedir.



Şekil 3.2. MFT Tablosundaki Veriler

3.1.2. LogFile Kayıtları

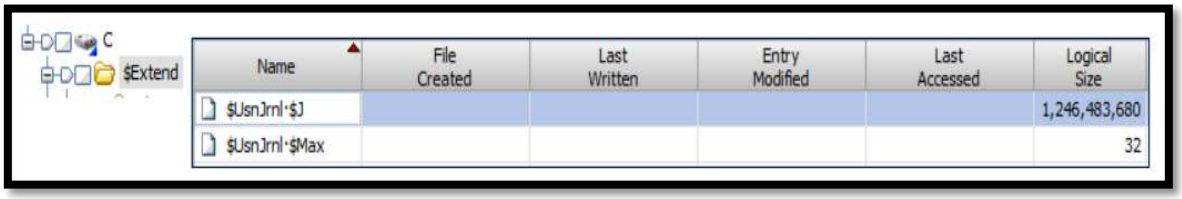
Günlük dosyaları, kullanıcıları, uygulamaları ve protokolleri, bulut ekosistemindeki etkinlikleri kaydetmenin birincil kaynağıdır. Bulut adli araştırmacıları, bir siber saldırganın veya kurumda çalışan personellerden birinin, suç mahallini belirleyerek ve olayın nasıl meydana geldiğini yeniden yapılandırarak bir sistemi ne zaman, neden ve nasıl tehlikeye attığını belirlemek için günlük kanıtlarını kullanabilir [18]. $Logfile kalıntıları günlük kayıtları olarak bilinmekte olup, windows işletim sistemi kurulumundan itibaren, kullanıcıların oluşturduğu klasörlerin ve dosyaların diskte barındırdığı konumunu hex olarak tutmaktadır. Logfile verileri sistem çökmesi, makinenin kapanması durumunda tekrardan yapılandırılmasında veya bilişim sisteminin açılmasına ait dosya sistemi verilerini tutan kayıtlardır [19]. Günlük kayıtları dosya veya klasörlerin verilerini tutar ve $logfile dosyasında büyük miktarda bilgi bırakır. Bir klasörün oluşturulmasıyla ilgili zaman damgaları, klasör adlarının veya konumlarının adli kayıtları incelemeciler için bir vakayı aydınlatmak konusunda büyük önem arz etmektedir.

3.1.3. USNJournal Kayıtları

Windows işletim sistemi dosya sistemi etkinliklerinin aktivitelerinin günlük olarak $J adlı özel bir veri akışında $ Extend \ $ UsnJrnl adlı bir dosyada tutmaktadır. USN günlüğü, NTFS günlük kaydı etkinleştirildiğinde, birimdeki dosyalarla ilgili işlemlerin bir listesini sağlar. Bu liste, tüm dosya oluşturma, yeniden adlandırma ve silme işlemlerinin bilgilerini içerir [20]. $J dosyası, birincil olarak yedekleme uygulamalarının en son yedeklemenin çalıştırılmasından bu yana değiştirilen dosyalara yönelik görünürlüğünü sağlamak için dosya sistemi işlemlerinin kayıtlarını içermektedir.

$Extend\$UsnJrnl: Windows işletim sistemi kurulduğunda $J oluşmakta ve boş birim olarak bulunmaktadır. Dosyalar sistemde değiştirildikçe, $J dosyası ek USN kayıtlarıyla genişletilebilmektedir.

Şekil 3.3'de USNJournal verilerinin boyutu ve bu verilerin bulunduğu dizin hakkında bilgi verilmektedir.



Şekil 3.3. USNJournal tablosu

3.2. Amcache Kayıtları

Amcache dosyası, yürütülen uygulamaların bilgilerini depolayan bir kayıt defteri dosyasıdır. Amcache'nin adli analizi hve dosyası, program adı ve sürümü, yürütme dosyası yolu, dosya boyutu, kurulum zaman damgası, ilk çalıştırma zaman damgası, yürütülebilir dosyanın karması vb. gibi önemli verileri ortaya çıkarabilir. Sistemdeki kalıntılar (prefetch) silinse bile amcache bu verileri tutmaya devam etmektedir. Amcache.hve dosyası ayrıca anti-adli inceleme (anti-forensic) programların izlerini kaydetmek için önemli bir yapıdır.

Amcache kalıntıları çoğunlukla tek başına bir vaka için kullanılsa da, windows işletim sistemi prefetch ve iconcache.db dosyaları ile beraber incelendiğinde uygulamaların genel bir zaman çizelgesini oluşturulabilmektedir. IconCache.db dosyaları, silinen dosyaların ve uygulamalarla ilgili simge önbellek bilgilerini içerir[21].

Tablo 3.1'de amcache verilerinin, farklı işletim sistemlerindeki fiziksel adresleri gösterilmektedir.

İşletim Sistemi	Dosya Konumu
Windows 7-8-8.1	%SystemDrive%\Windows\AppCompat\Programs\Amcache.hve
Windows 10	%SystemDrive%\Windows\appcompat\Programs\Amcache.hve

Tablo 3.1. Amcahce Dosya Konum Bilgisi

3.3. Shimcache Kayıtları

Windows Shimcache, microsoft tarafından Windows XP'den başlayarak yürütülen programlarla uyumluluk sorunlarını izlemek için oluşturulmuştur. Shimcache verileri en eski verilerin yeni girişlerle değiştirerek kaydetmektedir. Tutulan veri miktarı işletim sistemine göre değişir. Shimcache'de gerçekte yürütülmemiş girişler olabileceğini anlamak önemlidir. Shimcache'de kaydedilen ve adli analiz esnasında kullanılan birden çok kayıt bulunmaktadır. Bunlar şu şekildedir:

- Dosyanın tam yolu
- Dosya boyutu
- $Standard_ınformation (sı) son değiştirilme zamanı
- Shimcache son güncelleme zamanı
- İşlem çalıştırılma zamanı

Microsoft, Shimcache'i Windows Vista, 7, Server 2008 ve Server 2012'de her girdi için bir "işlem yürütme bayrağı" kategorisi içerecek şekilde tasarlanmıştır. Basitçe ifade etmek gerekirse,

işlem yürütme bayrağı, mevcut olduğu yerde araştırmacının bir girdinin yürütülüp yürütülmediğini veya bu girdinin bir dizine etkileşimli olarak göz atmak gibi dosya yürütme dışındaki bir etkinliğin bir sonucu olup olmadığını belirlemesini sağlamaktadır. Bu girişler, işlem yürütme bayrağının "FALSE veya TRUE" olarak işaretlenip işaretlenmediğini gözlemleyerek kolayca tespit edilebilmektedir. Kullanıcı-sistem işlemlerinin gerçekte kullanıcının tarafından yapılma durumlarına ve işlem çalıştırılma durumlarına dair örnek Tablo 3.2'de verilmiştir.

Son Değiştirilme Tarihi	**Dosya Yolu**	**Dosya Boyutu**	**Bayrak (Flag)**
02/18/15 05:25:00	C:\Program Files\Microsoft Office\Office14 \EXCEL.EXE	N/A	TRUE
07/14/09 01:15:00	C:\Windows\SysWOW64\EhStorShell.dll	N/A	TRUE
06/13/11 15:20:00	C:\Program Files (x86)\Common Files\ TortoiseOverlays\TortoiseOverlays.dll	N/A	TRUE
04/12/14 16:38:00	C:\SETUP64.EXE	N/A	FALSE
04/12/14 16:38:00	C:\Windows\System32\ncpa.cpl	N/A	FALSE
11/21/10 03:24:00	C:\Windows\system32\net1.exe	N/A	TRUE
07/14/09 01:39:00	C:\windows\system32\net.exe	N/A	TRUE

Tablo 3.2. Shimcache İşlem Yürütme Tablosu

Windows işletim sistemlerinde Shimcache kayıtlarının dosya yolu Tablo 3.3'de verilmiştir.

HKLM\\SYSTEM\\CurrentControlSet\\Control\\Session Manager\\AppCompatCache\\AppCompatCache

Tablo 3.3. Shimcache Konum Bilgisi

3.4. Samdatabase Kayıtları

Güvenlik hesabı Yöneticisi (SAM), windows işletim sisteminde parola, hesap grupları, erişim hakları ve özel ayrıcalıklar dahil olmak üzere kullanıcı hesabı bilgilerini depolamak için kullanılan bir veri tabanıdır. Samdatabase, SAM anahtar bilgilerini ve detaylarını vermektedir. Makinede açılan oturum bilgilerinin email hesapları, RDP (Uzak Masaüstü Protokolü) ile bağlantı sağlamış kullanıcı bilgileri de bu veritabanı içerisine dahildir. Kayıt defterinde, tüm kullanıcılar ve yöneticiler gibi genel kullanıcıları içermektedir. Samdatabase dosyası yönetici kullanıcılar tarafından sistem ayrıcalıklarına sahip kullanıcılar dışında okunamazlar. Genel kullanıcı ve yönetici kullanıcıları, sam dosyasını okumak için yönetici ayrıcalıklarının izniyle (yönetici olarak çalıştırma ve registry kayıt kontrolü) bu kayıt defteri anahtarlarının erişim denetim listesini değiştirerek okuma-yazma erişimi kazanmaları gerekmektedir [22].

SAM veritabanının güvenliğini artırmak için Microsoft, Windows NT 4.0'da SYSKEY etkinleştirildiğinde, SAM dosyasının disk üzerindeki kopyası kısmen şifrelenmekte ve böylece SAM'de depolanan tüm yerel hesapların parola karma değerleri bir anahtarla şifrelenmektedir[23]. SAM dosyasının şifre ve içerisinde bulunan herhangi bir kullanıcının karma bilgisi (kullanıcı oturum bilgisi) elde edilmesi ile saldırgan-kurban ilişkisi tam anlamıyla başlamış olmaktadır.
Tablo 3.4'de samdatabase verisine ait fiziksel konumları hakkında bilgi verilmektedir.

Dosya Merkezi	**Samdatabase Dosya Konumu**
İşletim Sistemi	%SystemRoot/system32/config/SAM%
Registry	HKLM/SAM ve HKLM/SECURİTY/SAM

Tablo 3.4. Sam Konum Bilgisi

3.5. Srumdatabase Kayıtları

Windows 8'den itibaren kullanılmaya başlanılan Srumdatabase, işletim sisteminin ağ bileşenlerini izlemeye, kablosuz veya kablolu ağların paketlerini, kullanımlarını kaydeden log dosyasıdır. Ağ ile çalışan programların veya uygulamaların ağ bağlantılarını, ağ kullanımlarını, kullanılan paketleri, tüketilen enerjiyi, bir uygulamanın ne kadar süre ile kullanıldığını genel olarak detaylandırılabildiği kayıt defteridir. Kullanıcı, kullanılan CPU döngüleri, okunan / yazılan veri baytları ve ağ verileri (gönderilen / alınan) gibi işlemle ilgili bilgiler, Sistem Kaynağı Kullanım İzleyicisi (SRUM) adı verilen bir mekanizma tarafından sürekli olarak kaydedilmektedir [24].

Örneğin bir word dosyası saat 3-5 arasında çalışmakta olduğunu düşünürsek 2 saat boyunca Word dosyasının kullandığı ağ paketleri istatistikleri processleri gösteren kayıtlardır. Aynı zamanda srumdatabase içerisinde farklı kaynakların kayıtlarının da olduğu dll dosyaları mevcuttur.

Tablo 3.5'de srum verilerinin bilgileri ve bu verilerin elde edilebilecek dll dosyalarına ait bilgiler verilmektedir.

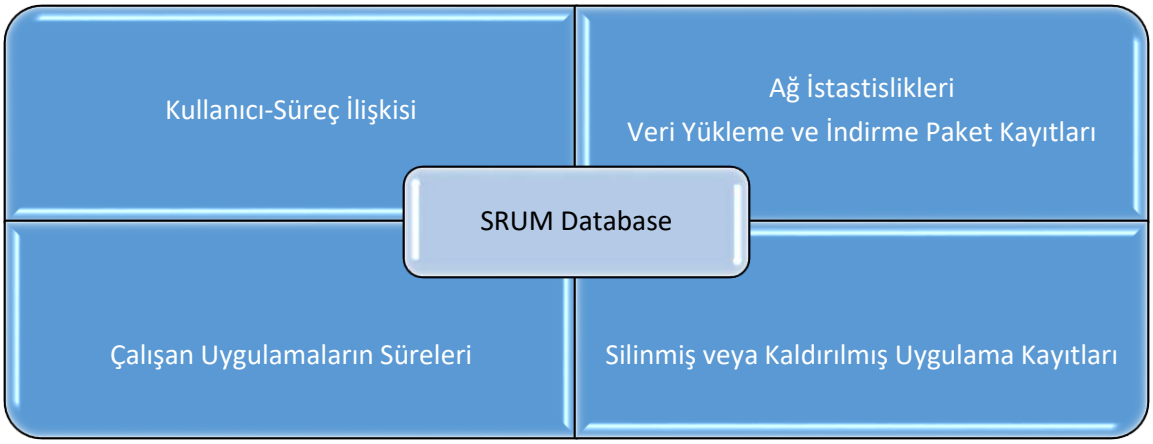
SRUM Uzantıları	Dll Dosyaları
Ağ Verilerinin Kullanım Kayıtları	nduprov.dll
Push Bildirimleri Srum Kayıtları	wpnsruprov.dll
Uygulama Kaynağı Kullanım Kayıtları	appsruprov.dll
Ağ Bileşenleri Kullanım Kayıtları	ncuprov.dll
Enerji Kullanım Kayıtları	energyprov.dll

Tablo 3.5. Srum Kayıt Bileşenleri

Tablo 3.6'da srum verilerinin fiziksel konumları hakkında bilgiler verilmektedir. Aynı zamanda Şekil 3.4'de bu konumlardaki dosyalar içerisinden hangi verilerin elde edilebileceğinden bahsedilmiştir.

SRUM Merkezi	SRUM Konum Yolu
Disk	C: \ Windows \ System32 \ sru \ SRUDB.dat
Registry	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ SRUM\Extensions

Tablo 3.6. SRUM Konum Yolu



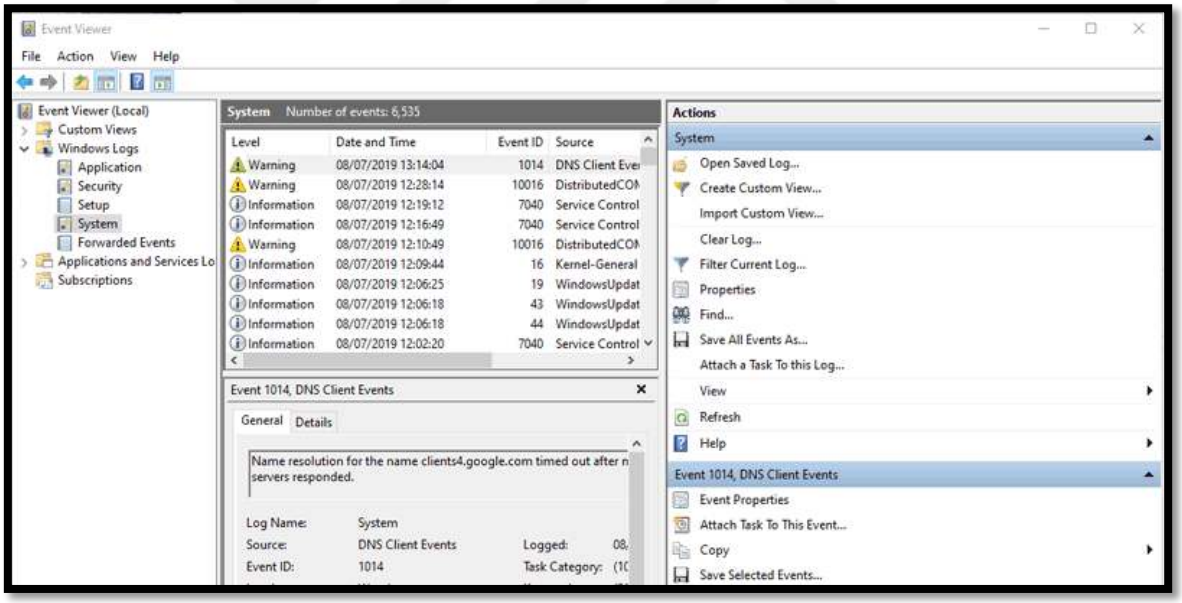
Şekil 3.4. SRUM Aktivite Bilgisi

3.6. Windows İşletim Sistemi Event Log Kayıtları

Windows işletim sistemine sahip makineler üzerinde yapılan işlemler ve aktiviteler, olay günlüğü kayıtlarında tutulmaktadır. Windows 7 ve sonraki sürümlerde olay günlükleri kayıtları. evtx uzantılı dosyalarda ve Windows Xp, Windows Server 2003, Vista gibi işletim sistemlerinde ise. evt uzantılı olarak kayıt edilmektedir. Olay günlüğü kayıtları makine üzerinde kullanılan servis veya uygulamalara göre farkındalık gösterebilmektedir. Örneğin, antivirüs uygulaması için program tarafından karantinaya alınan dosya bilgileri tarih ve zaman gibi kayıtlar ayrı olay günlüğü dosyalarında tutulmaktadır. Windows işletim sistemi varsayılan olarak "System.evtx, Security.evtx, Application.evtx" gibi olay günlüğü kayıtları oluşmaktadır. Sistemlerdeki uygulamaların veya aktivitelerin kayıt altına alınması için windows log policylerinin yapılandırılması gerekmektedir.

Windows işletim sistemi kullanıcı-sistem tarafından gerçekleşen aksiyonlara göre farklı event id'ler (örneğin: 4688, 4624, 4625 gibi.) bulunmaktadır. Bu ID'ler gerçekleşen aksiyonun sınıflandırılmasında kullanılmaktadır.

Şekil 3.5'de örnek event log analiz ekranı gösterilmiştir.



Şekil 3.5. Event Log Analizi

3.6.1. Security Evtx Kayıtları

Security.evtx dosyası çalışan process, kullanıcı aktiviteleri, servis durumları gibi kullanıcı-sistem aktivitelerinin kayıtlarının tutulduğu dosyadır. Kurumsal bir ağ üzerinde yer alan makinelere ait security.evtx dosya olay günlüğü kaydında çalıştırılan processlere ilişkin event id aktif hale getirilmeli ve process komut satırı gibi özellikleri açılması sağlanmalıdır. Bu sayede analizci olay günlüğü dosyalarında sistem üzerinde çalışan processlere ilişkin kayıtları analiz edebilir.

3.7. Registry (Kayıt Defteri) Logları

Windows işletim sistemi için alt düzey sistem ayarlarını depolayan bir veritabanıdır. Kayıt defteri verilerinin yapısı, anahtar düğümler ve değer verileri olmak üzere iki farklı veri türünden oluşmaktadır. Kayıt defterinin yapısı, anahtar düğümlerin dizin rolünü oynadığı ve değerlerin dosyalar gibi davrandığı bir dosya sistemine benzer olarak düşünülebilir[25]. Bu alanda, güvenlik, hizmetler ve kullanıcı hesap ayarlarına kadar bir çok bilgi bulunmaktadır. Kayıt defterinde, işletim sistemi zaman bilgisi, kapanma ve açılma saatleri, bilgisayar ismi, mail kullanıcıları, bilgisayar kullanıcılar, son belgeler, kaldırılan ve son yüklenen programlar, yazıcılar, işletim sisteminin kurulumundan bu yana kullanılan kablosuz ağ bilgileri, bilgisayara bağlanan harici disk kayıtları, silinen son öğeler, yedek kayıtları vb. gibi bir vakayı aydınlatabilecek önemli verilen mevcuttur. Aynı zamanda bir programın kalıcılığı da kayıt defterinde tespit edilmektedir[26].

Tablo 3.7'de işletim sisteminde bulunan kayıt defterleri gösterilmiş olup, kayıt defterlerinde elde edilebilecek verilerden bahsedilmektedir. Aynı zamanda Tablo 3.8'de windowsun ilk sürümünden günümüze kadar olan sürümlerindeki farklı sistem verilerinin neler olduğu ve bu verilerin hangi işletim sistemlerinde elde edilebileceği hakkında bilgi verilmektedir.

Registry Hives	**Hives Açıklaması**
HKEY_CLASSES_ROOT	Yazılım ve kullanıcı arayüzleri hakkında bilgiler
HKEY_CURRENT_USER	Mevcut oturum açmış kullanıcı için yapılandırma bilgileri
HKEY_LOCAL_MACHINE	Makine donanımına özgü bilgiler
HKEY_USERS	Tüm kullanıcılar için yapılandırma bilgileri
HKEY_CURRENT_CONFIG	Mevcut sistem konfigürasyonu

Tablo 3.7. Registry Verilerinin Açıklamaları

Sistem Verisi / Sistem Türü	95	98	2000	XP	VISTA	7	8	10
BCD	–	–	–	–	✓	✓	✓	✓
DRIVERS	–	–	–	–	–	–	–	✓
SAM	–	–	✓	✓	✓	✓	✓	✓
SECURITY	–	–	✓	✓	✓	✓	✓	✓
SOFTWARE	–	–	✓	✓	✓	✓	✓	✓
SYSTEM	–	–	✓	✓	✓	✓	✓	✓
DEFAULT	–	–	✓	✓	✓	✓	✓	✓
COMPONENTS	–	–	–	–	✓	✓	✓	✓
NTUSER.DAT	–	–	✓	✓	✓	✓	✓	✓
USRCLASS.DAT	–	–	–	✓	✓	✓	✓	✓
SYSTEM.DAT	✓	✓	–	–	–	–	–	–
USER.DAT	✓	✓	–	–	–	–	–	–
POLİCY.POL	–	✓	–	–	–	–	–	–

Tablo 3.8. Windows İşletim Sistemlerine Ait Sistem Verileri Türleri

3.8. Prefetch Log Kayıtları

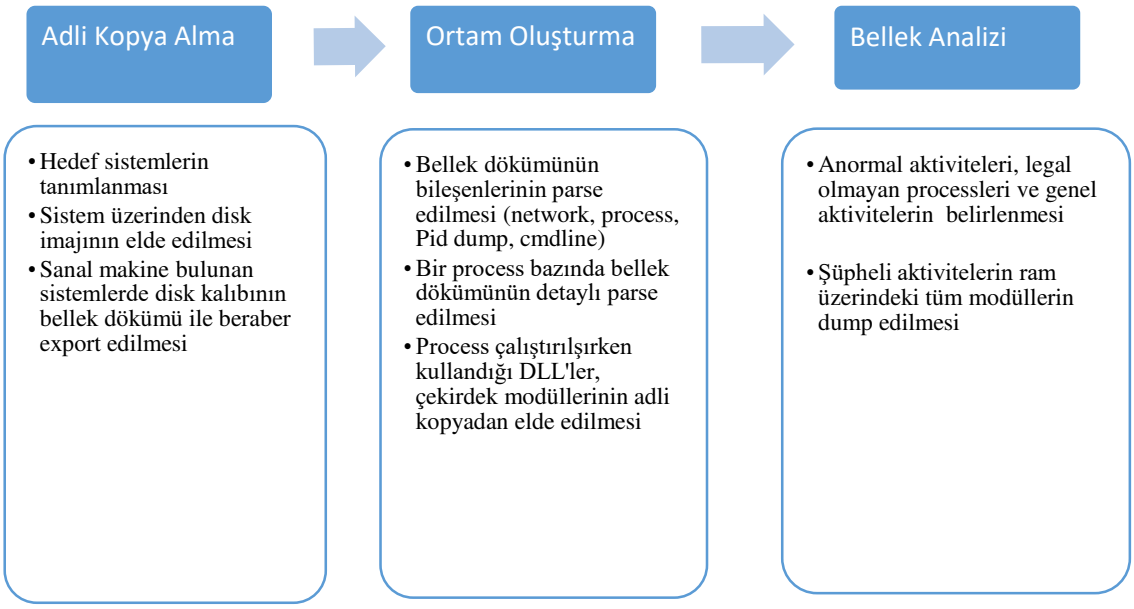
Windows işletim sisteminde kullanıcılar bir uygulamayı her çalıştırdığında, uygulama tarafından sisteme yüklenen dosyalar hakkındaki kayıtlar tutulmaktadır. Prefetch dosyasındaki bilgiler, çalıştırılan bir uygulamanın tekrar çalıştırılması durumunda uygulamayı en kısa sürede optimize etmek için tutulmaktadır. Bu bilgiler bazen word dosyası bazen bir internet tarayıcısı çerezleri olabilmektedir.

Aynı zamanda kullanıcıların Windows işletim sistemi tabanlı makineler üzerinde çalıştırıldığı programların çalıştırılma tarihi, çalıştırıldığı konum gibi tutan kayıtlardır.

3.9. Windows İşletim Sistemi Bellek Adli Bilişimi

Windows'ta disk görüntüsünden elde edilemeyen geçici bilgiler RAM (Random Access Memory) alanında tutulmaktadır. Bellekte, saklanan ve bilgisayar güç kaybettiğinde veya kapatıldığında kaybolacak veriler bulunmaktadır. Bu tür veriler kayıtlarda, önbellekte ve rasgele erişim belleğinde (RAM) bulunur [27]. Kullanıcıların bilgisayarı açtıktan kapatıncaya kadar yaptığı ve zamanlanmış tüm görevleri bellek üzerinde erişim sağlanmaktadır. Örnek olarak, son kopyalanan ögeler, değiştirilen dosyalar, internet çerezleri, vb. gibi kullanıcının bilgisayarda yaptığı aktivitelerin diske kaydedilmeden tutulduğu uçucu verilerdir. Saldırganlar veya kullanıcılar tarafından yapılan aktiviteler disk harici bellekte de tutulmakta olup, belleğin büyük bir kısmını, sabit sürücüden önbelleğe alınan dosya verileri barındırmaktadır [28].

Bellek görüntülerinden toplanan bilgiler, çalışan işlem bilgisi, çalıştıran kullanıcı bilgisi ve çalıştırılma tarihi bilgisi gibi veriler elde edilebilmektedir. Aynı zamanda çalıştırılan işlemin yaptığı belirli etkinlikler ve anlık olarak etkin ağ bağlantılarının durumu gibi bilgileri de içermektedir[29]. Bellek analizi yapılırken, "volatility, mimikatz, magnet forensic, dumpit, rekall forensic, belkasoft forensic ve dd" araçları kullanılabilmektedir. Şekil 3.6'da bellek analizinde işletilebilecek süreçten bahsedilmiştir.



Şekil 3.6. Bellek Analiz Süreci

3.10. Remote Desktop Protocol Cache (Uzak Masaüstü Protokülü Önbelleği)

Windows tarafından sağlanan "mstc" istemcisini kullanırken, RDP ağda yanal olarak hareket etmek için kullanılabilir. Bilinen bazı zafiyetler kullanılarak, analiz edilen makine ile ilgili gizli veriler (ip ve port bilgisi gibi) ile izinsiz giriş kayıtları tespit edilebilmektedir.[30]. Önbellek dosyaları, iletişim kurulan makinelerin bilgilerini ve bağlı olunan makinenin ekran resimlerini içermektedir. Tablo 3.9'da rdp isteği yapılan bir makinede bulunabilecek uzak masaüstü protokolü önbellek kayıtlarının dosya erişim yolu verilmiştir.

Cache Konumu: C:\Users\aliucar01\AppData\Local\Microsoft\Terminal Server Client\Cache

Tablo 3.9. Rdp Cache Konumu

3.11. LNK Dosya Kayıtları

Lnk dosyaları windows işletim sisteminde kullanıcıların genellikle kullandığı programların veya dosyaların kısayollarıdır. Lnk dosyaları kullanıcı tarafından en son erişilen dosyalar, en son kullanılan uygulamalar ve belgeler gibi son yapılan işlemlerin tespit edilmesi için önem arz etmektedir. Windows işletim sisteminde son olarak kullanılan dosyanın kısayolu gerçek bir dosya silinse bile duruyor ise, silinen dosyalar veya belgeler hakkında bilgi verebilmektedir. LNK dosyaları, erişim kolaylığı için diğer dosyalara veya çalıştırılabilir programlara erişim sağlattırır ve onların kullanıldığını işaret eder. Bu kayıt alanında şu bilgiler elde edilebilmektedir:

- Hedef dosyanın orijinal yolu,
- Hem hedef dosyaların hem de .lnk dosyalarının zaman damgası,
- Sistem, gizli vb. dosya öznitelikleri,
- Diskle ilgili ayrıntılar,
- Uzaktan veya yerel yürütme,
- Makinelerin MAC adresi.

Lnk dosyalarının konumları windows işletim sistemlerinde farklı farklı dosya yolunda bulunmaktadır. Tablo 3.10'da işletim sistemlerine ait dosya yolları verilmiştir.

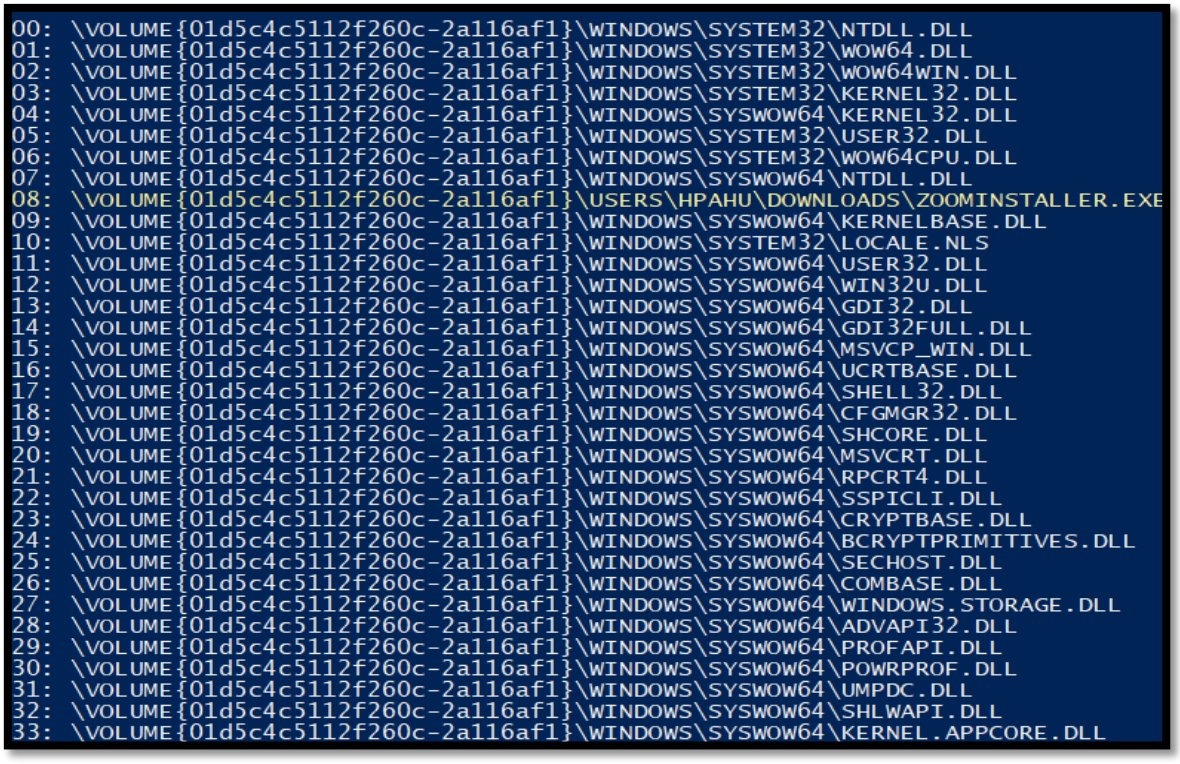
İşletim Sistemi / Dosya	Lnk Dosya Konumu
Windows Xp	\Documents and Settings\UserName\Recent
Windows Xp / Office	\Documents and Settings\UserName\Application Data\Microsoft \Office\Recent
WindowsVista,7,8,10	\Users\UserName\AppData\Roaming\Microsoft\Windows\Recent
WindowsVista,7,8,10 Office	\Users\UserName\AppData\Roaming\Microsoft\Office\Recent

Tablo 3.10. Lnk Dosya Konumları

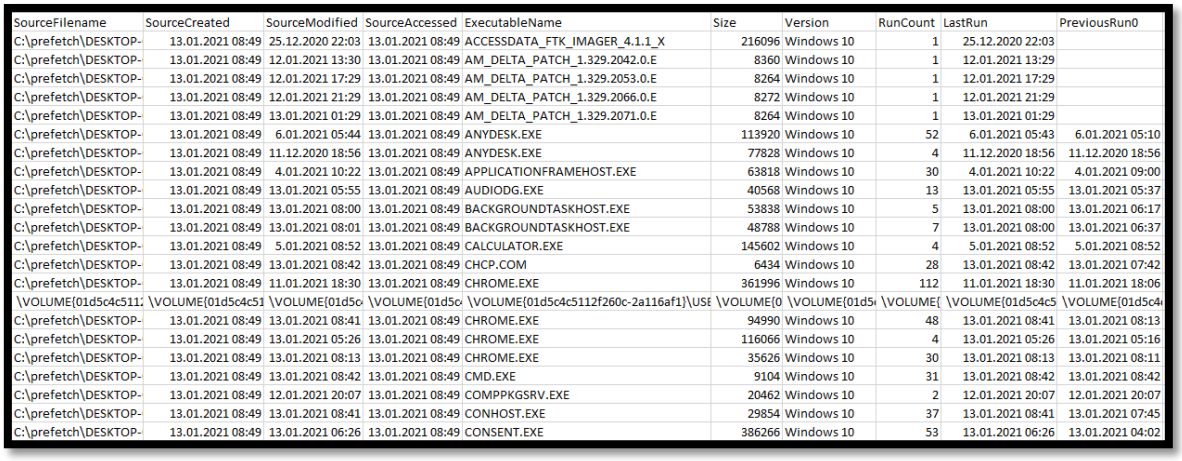
3.12. PECmd Prefetch Analizi

Windows işletim sisteminde bulunan Prefetch dosyalarını okuyan ve bunlarda depolanan bilgileri görüntüleyen, aynı zamanda sınıflandıran açık kaynak kodlu yazılımdır. Prefetch kayıtlarındaki bilgileri parse ederek, anlamlı hale getirip bir programın ya da uygulamanın hangi zaman aralıkları ile çalıştırıldığı ilk çalıştırılma zamanları ve çalıştırılma sayıları, hangi programları tetikleyerek çalıştırıldığını, her uygulamanın hangi dosyaları kullandığını ve windows önyüklemesinde hangi dosyaların yüklendiğini dair kayıtları tutmaktadır.

Şekil 3.7'de bir işletim sistemine yüklenen zoom programının ardından işletim sisteminde yüklenen programla beraber yüklenen dll dosyaları tespit edilmektedir. Aynı zamanda bu dosyalara ait bilgiler ise Şekil 3.8'de detaylı olarak gösterilmektedir.



Şekil 3.7. Zoom Prefetch Kayıtları



SourceFilename	SourceCreated	SourceModified	SourceAccessed	ExecutableName	Size	Version	RunCount	LastRun	PreviousRun0
C:\prefetch\DESKTOP-	13.01.2021 08:49	25.12.2020 22:03	13.01.2021 08:49	ACCESSDATA_FTK_IMAGER_4.1.1_X	216096	Windows 10	1	25.12.2020 22:03	
C:\prefetch\DESKTOP-	13.01.2021 08:49	12.01.2021 13:30	13.01.2021 08:49	AM_DELTA_PATCH_1.329.2042.0.E	8360	Windows 10	1	12.01.2021 13:29	
C:\prefetch\DESKTOP-	13.01.2021 08:49	12.01.2021 17:29	13.01.2021 08:49	AM_DELTA_PATCH_1.329.2053.0.E	8264	Windows 10	1	12.01.2021 17:29	
C:\prefetch\DESKTOP-	13.01.2021 08:49	12.01.2021 21:29	13.01.2021 08:49	AM_DELTA_PATCH_1.329.2066.0.E	8272	Windows 10	1	12.01.2021 21:29	
C:\prefetch\DESKTOP-	13.01.2021 08:49	13.01.2021 01:29	13.01.2021 08:49	AM_DELTA_PATCH_1.329.2071.0.E	8264	Windows 10	1	13.01.2021 01:29	
C:\prefetch\DESKTOP-	13.01.2021 08:49	6.01.2021 05:44	13.01.2021 08:49	ANYDESK.EXE	113920	Windows 10	52	6.01.2021 05:43	6.01.2021 05:10
C:\prefetch\DESKTOP-	13.01.2021 08:49	11.12.2020 18:56	13.01.2021 08:49	ANYDESK.EXE	77828	Windows 10	4	11.12.2020 18:56	11.12.2020 18:56
C:\prefetch\DESKTOP-	13.01.2021 08:49	4.01.2021 10:22	13.01.2021 08:49	APPLICATIONFRAMEHOST.EXE	63818	Windows 10	30	4.01.2021 10:22	4.01.2021 09:00
C:\prefetch\DESKTOP-	13.01.2021 08:49	13.01.2021 05:55	13.01.2021 08:49	AUDIODG.EXE	40568	Windows 10	13	13.01.2021 05:55	13.01.2021 05:37
C:\prefetch\DESKTOP-	13.01.2021 08:49	13.01.2021 08:00	13.01.2021 08:49	BACKGROUNDTASKHOST.EXE	53838	Windows 10	5	13.01.2021 08:00	13.01.2021 06:17
C:\prefetch\DESKTOP-	13.01.2021 08:49	13.01.2021 08:01	13.01.2021 08:49	BACKGROUNDTASKHOST.EXE	48788	Windows 10	7	13.01.2021 08:00	13.01.2021 06:37
C:\prefetch\DESKTOP-	13.01.2021 08:49	5.01.2021 08:52	13.01.2021 08:49	CALCULATOR.EXE	145602	Windows 10	4	5.01.2021 08:52	5.01.2021 08:52
C:\prefetch\DESKTOP-	13.01.2021 08:49	13.01.2021 08:42	13.01.2021 08:49	CHCP.COM	6434	Windows 10	28	13.01.2021 08:42	13.01.2021 07:42
C:\prefetch\DESKTOP-	13.01.2021 08:49	11.01.2021 18:30	13.01.2021 08:49	CHROME.EXE	361996	Windows 10	112	11.01.2021 18:30	11.01.2021 18:06
\VOLUME{01d5c4c511:	\VOLUME{01d5c4c51	\VOLUME{01d5c	\VOLUME{01d5c	\VOLUME{01d5c4c5112f260c-2a116af1}\USE	\VOLUME{0	\VOLUME{01d5	\VOLUME{	\VOLUME{01d5c4c5	\VOLUME{01d5c4
C:\prefetch\DESKTOP-	13.01.2021 08:49	13.01.2021 08:41	13.01.2021 08:49	CHROME.EXE	94990	Windows 10	48	13.01.2021 08:41	13.01.2021 08:13
C:\prefetch\DESKTOP-	13.01.2021 08:49	13.01.2021 05:26	13.01.2021 08:49	CHROME.EXE	116066	Windows 10	4	13.01.2021 05:26	13.01.2021 05:16
C:\prefetch\DESKTOP-	13.01.2021 08:49	13.01.2021 08:13	13.01.2021 08:49	CHROME.EXE	35626	Windows 10	30	13.01.2021 08:13	13.01.2021 08:11
C:\prefetch\DESKTOP-	13.01.2021 08:49	13.01.2021 08:42	13.01.2021 08:49	CMD.EXE	9104	Windows 10	31	13.01.2021 08:42	13.01.2021 08:42
C:\prefetch\DESKTOP-	13.01.2021 08:49	12.01.2021 20:07	13.01.2021 08:49	COMPPKGSRV.EXE	20462	Windows 10	2	12.01.2021 20:07	12.01.2021 20:07
C:\prefetch\DESKTOP-	13.01.2021 08:49	13.01.2021 08:41	13.01.2021 08:49	CONHOST.EXE	29854	Windows 10	37	13.01.2021 08:41	13.01.2021 07:45
C:\prefetch\DESKTOP-	13.01.2021 08:49	13.01.2021 06:26	13.01.2021 08:49	CONSENT.EXE	386266	Windows 10	53	13.01.2021 06:26	13.01.2021 04:02

Şekil 3.8. Prefetch Kayıtları

3.13. JumpList Kayıtları

Jumplist, bir kullanıcının son erişim sağladığı programları dosyaları resimleri belgeleri erişimlerini daha hızlı gerçekleştirmelerine veya ilgili uygulama üzerinden en sık gerçekleştirilen aksiyonu gerçekleştirmelerini sağlamaktadır. En son erişilen uygulamalar ve dosyalar hakkında bilgi içerirler. Jumplist özelliği Windows 7 ile birlikte sunulmuştur[31]. Dosya türüne göre jumplistin içeriği değişmektedir. Örneğin bir müzik programının jumplistleri son çalınan müzikler olabilirken, bir word dosyasının jumplisti son açılan word dosyaları olabilmektedir. Bu atlama listeleri (jumplist) özel olarak yapılır ve kullanıcı bir dosyayı veya uygulamayı pinglediğinde oluşturulur. Windows'ta iki tür atlama listesi oluşturulabilir:

AUTOMATICDESTINATIONS-MS: AutomaticDestinations türündeki listeler sistem tarafından otomatik olarak oluşturulan listelerdir.

CUSTOMDESTINATIONS-MS: CustomDestinations türündeki listeler ise uygulamaların kendileri tarafından oluşturulan listelerdir.

Windows işletim sisteminde Jumplist'ler aşağıdaki iki dizin altında saklanırlar ve bu dizinler gizlidirler.

Tablo 3.11'de Jumplist dosyalarının fiziksel adreslerinden bahsedilmektedir.

Jumplist Türü	Jumplist Dosya Konumu
Auto-Ms	C:\Users\aliucarabm\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations
Custom-Ms	C:\Users\aliucarabm\AppData\Roaming\Microsoft\Windows\Recent\CustomDestinations

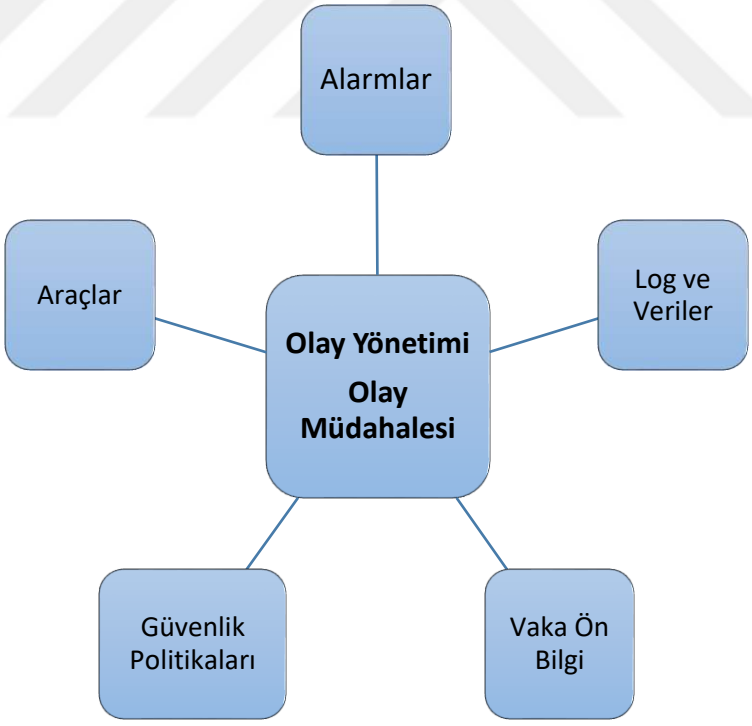
Tablo 3.11. Jumplist Dosya Konumları

4. MATERYAL VE METOT

Dijital adli analizde, dijital kalıntılar kavramı vardır. Bu kavram, bir yazılım, uygulama veya aygıtın geçmiş, şimdiki veya gelecekteki kullanımını veya işlevini tanımlayabilen verileri içeren dijital bir nesne olarak tanımlanmıştır [32]. Bilgi güvenliği olaylarına etkin müdahale, modern organizasyonların kritik bir işlevidir. Bununla birlikte, son araştırmalar, kuruluşların, acil durum tespiti ve sonraki düzeltici eylemlere odaklanan, dar ve teknik bir olay müdahalesi (Incident Response-IR) görüşünü benimsediğini göstermiştir [33].

Olay müdahalesi süreçlerinde yer alan tespit etme ve analiz aşamasında, bir kurumsal ağ üzerinde yer alan onlarca sayıda windows işletim sistemine sahip makinelerden elde edilecek triyaj kayıtlarının toplu olarak analizinin sağlanması olay müdahalesi süreçlerinde alınacak hızlı aksiyonların temelini oluşturmaktadır. Olay müdahalesi bir olay yönetiminin parçasıdır.

Olay yönetimi, iş sürekliliğini ve bir organizasyonun genel verimliliğini sağlamak için önemlidir. Olay yönetimine çeşitli yaklaşımlar vardır [34]. Bu yaklaşımlara Şekil 4.1'de bahsedilmiştir.



Şekil 4.1. Olay Yönetimi Yaklaşımları

Siber saldırı soruşturması, zararların azaltılmasını ve gelecekteki önleme yaklaşımlarının olgunlaşmasını destekleyebileceği için önemlidir[35]. Soruşturma esnasında makine sayısı arttıkça triyaj kaydı toplanmayan makinelerden elde edilecek fiziksel ve mantıksal imajların sayısı ve boyutlarının fazla olacağından dolayı analizcinin hızlı bir şekilde olayı çözümlemesine engel olabilmektedir. Bu sebeple enfekte olmuş sistemlerden elde edilecek triyaj kayıtlarının incelenmesi, bu başlığın altında sırasıyla global kullanımı kanıtlanmış araç tarafından ham logların toplanması ve geliştirmiş olduğumuz araç ile toplu ayrıştırma (parser) işlemi, olay zaman çizelgesinin çıkarılması anlatılacaktır.

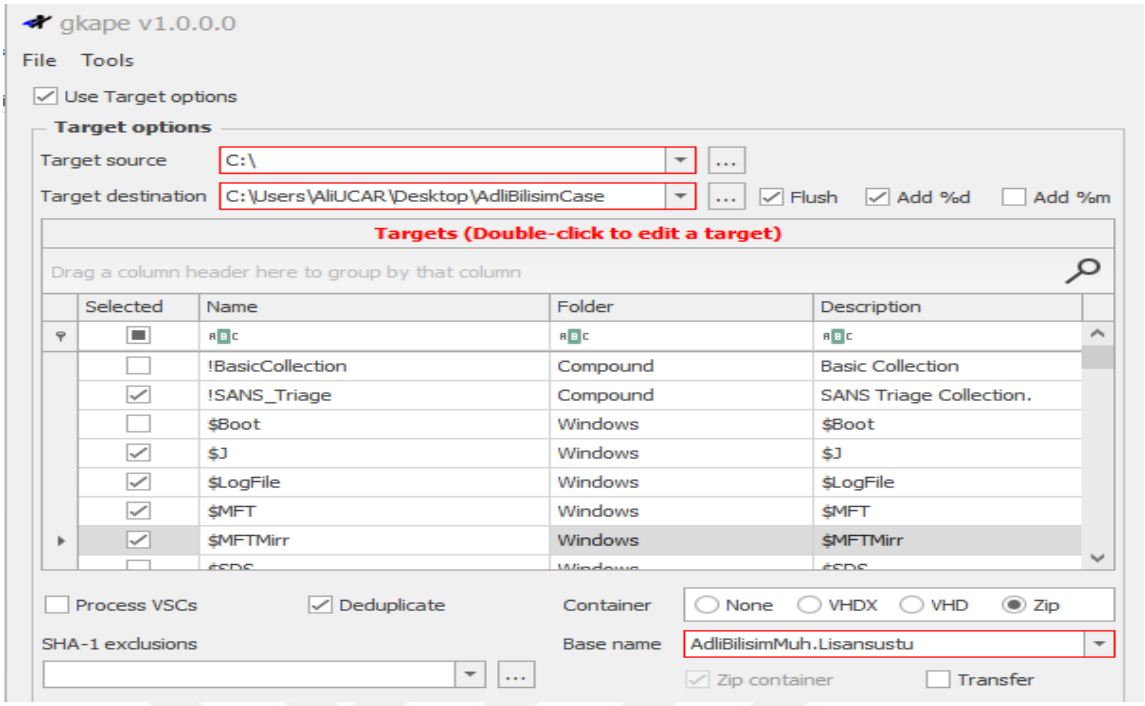
4.1. Windows İşletim Sistemi Kalıntılarının Toplanması

Bu başlık altında Windows işletim sistemlerinde bulunan sistem kalıntılarının toplanılması ve aynı zamanda sistem kalıntılarının toplanması için gerekli olan materyallerin kurulumu ve çalıştırılması anlatılacaktır.

4.1.1. Kape ile Kalıntıların Elde Edilmesi

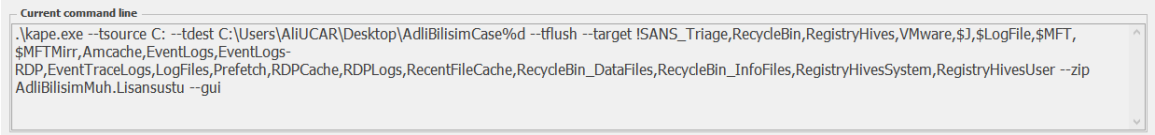
Windows işletim sistemlerinde kullanıcının ve işletim sisteminin kalıntılarının analiz edilmesi için, önce toplanması sonra da ayrıştırlması gerekmektedir. Bu başlık altında windows işletim sistemlerinde bulunan kalıntıların toplanması anlatılmıştır.

Kape aracında kullanım kolaylığı olması açısından aracın hem grafiksel tabanlı hemde komutsal tabanlı kullanımı bulunmaktadır. Kalıntıların toplanması kape grafiksel tabanlı olarak gerçekleştirilmiştir. Şekil 4.2'de gkape ile windows işletim sisteminde varolan kalıntılar seçilerek toplama parametreleri ve komut cümleciği oluşturulur. Ardından vaka adı belirtilerek makinede bulunan kalıntılar toplanmaya başlatılır. Makineden toplanan belirli kalıntılar şu şekildedir; amcache, prefetch, mft, msiecf, pe, recycle_bin, recycle_bin_info, sccm, usnjrnl, winevt, winevtx, winreg, windows_sam_users, windows_task_cache, mru, winlogon, vb.

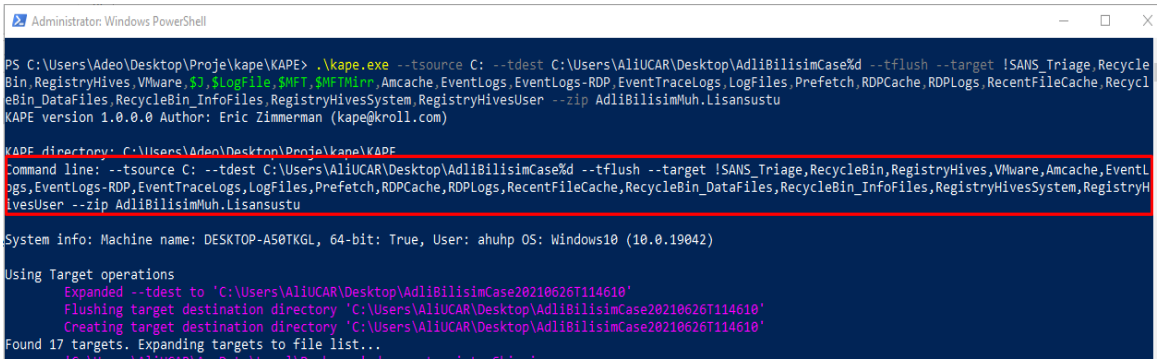


Şekil 4.2. Kape Kullanım Arayüzü

Ckape kullanılarak komutsal cümleciği ile de kalıntılar toplanabilmektedir. Şekil 4.3'de kalıntıların toplanması için hazırlanmış komutsal cümleciği verilmiştir. Şekil 4.4'de ise powershell ile komutsal cümlecik kullanarak makinedeki kalıntıların toplanması sağlanabilmektedir.



Şekil 4.3. Kape Kalıntı Toplama Cümleciği



Şekil 4.4. Komutsal Cümlceik ile Kalıntıları Toplama İşlemi

Kape aracının grafiksel ve komutsal işlem dosyalarının indirilme linki şu şekildedir[36]: https://www.kroll.com/en/services/cyber-risk/incident-response-litigation-support/kroll-artifact-parser-extractor-kape

4.1.2. Log2Timeline Aracının Kurulması

Log2Timeline bir sistemdeki çeşitli günlük dosyası ve olayların bir kronolojik sıralaması olan zaman çizelgeleri oluşturan ve olası bir durumu anlamaları için önemli bir araçtır[37]. MFT ile $Logfile dosyasının beraber kullanımı ile oluşturulmakta olup, işletim sistemi kurulumundan bu yana yapılan tüm aktiveleri, belirli bir zaman sıralaması göre sınıflandırarak analizciye kolaylık sağlamaktadır.

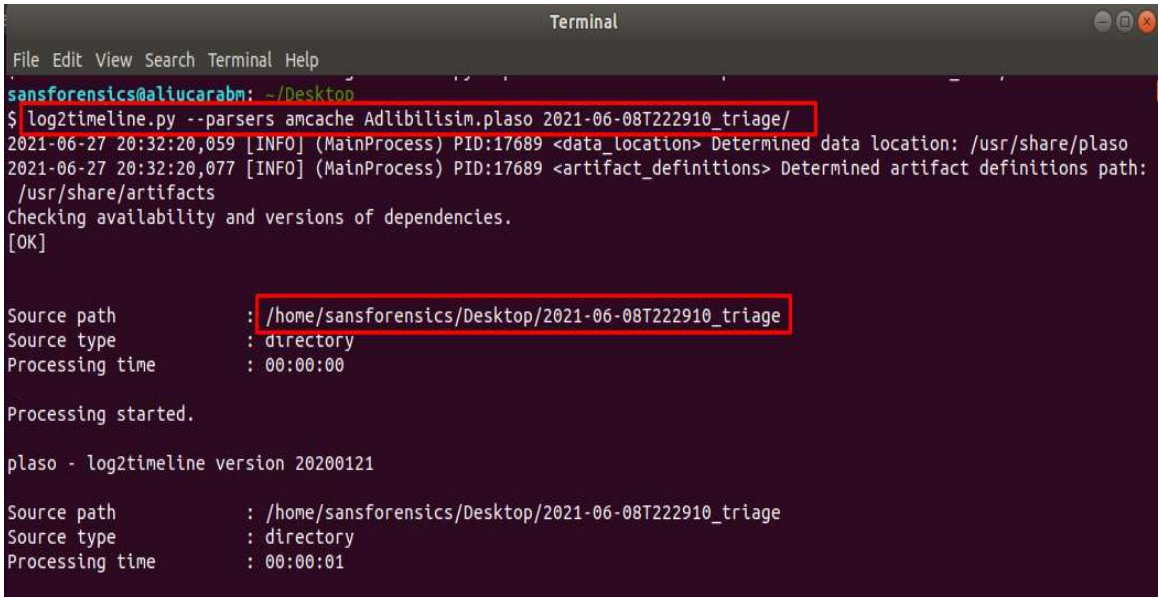
Log2timeline zaman çizelgelerinin otomatik olarak oluşturulması için çeşitli araçlar tarafından kullanılan Python tabanlı bir araçtır. Bu başlık da bir windows işletim sisteminin loglarının zaman hiyerarşisine göre sıralanmasını sağlayan log2timeline aracının temiz bir ubuntu makineye kurulumundan bahsedilmiştir.

Log2timeline github indirilme linki şu şekildedir[38]:

https://github.com/log2timeline/plaso

Öncelikle requirements.txt dosyası ilgili linkten indirilir ve ardından temiz ubuntu makinede aşağıdaki komutlar çalıştırılarak yüklenebilmektedir. Şekil 4.5'de log2timeline ait örnek kullanım gösterilmiştir.

```
pip install -r requirements.txt
sudo apt-get update
sudo apt-get install python-plaso plaso-tools
```



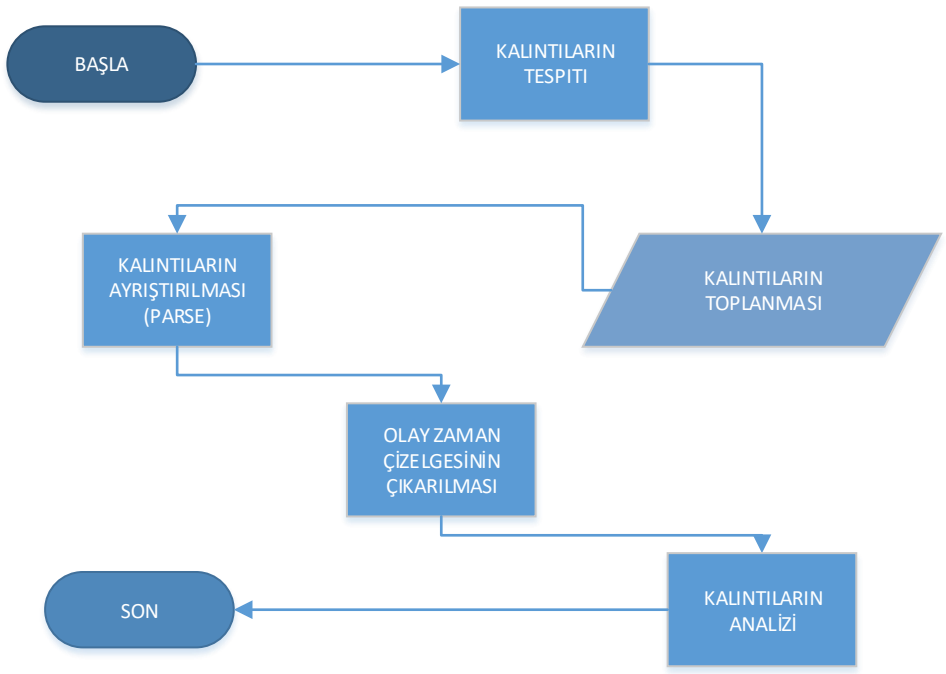
Şekil 4.5. Log2timeline Kullanım Örneği

4.2. IR-Parser Siber Olay Müdahale Aracının Geliştirilmesi

Gelişen saldırı paradigmaları, gelişmiş özel olarak oluşturulmuş araçlar, öngörülemeyen istismar kalıpları ve artan siber savunmalara uyum sağlama yeteneği ile kurumsal siber güvenliğe meydan okumaktadır[39]. Windows işletim sistemlerinde olay müdahalesi için işletim sisteminde bulunan kalıntıların analiz edilmesi, kalıntıların makinelerden toplanması ve ayrıştırılması gerekmektedir. Birçok güvenlik olayı müdahale yaklaşımındaki son aşama, geri bildirim/takip aşamasıdır. Bu aşamada, bir kuruluşun bir olaydan ders çıkarması, güvenlik olayı yanıt sürecini iyileştirmek ve daha geniş güvenlik ortamını olumlu yönde etkilemek için bir soruşturma sırasında makinelerden toplanan bilgileri kullanması beklenmektedir[40]. Bu nedenle toplanan verilerin elde tutulabilirliği, kanıtlanabilirliği önemlidir. Olay makinelerinden toplanan kalıntıların analiz edilmesi ve siber olaylara hızlı aksiyon alınması için zaman çok önem arz etmektedir. Bu başlık altında geliştirmiş olduğumuz siber olay müdahale aracı olan "IR-Parser" aracının amacı ve kaynak kodlarından bahsedilecektir.

4.2.1. Programın Aşamaları

Bu proje için geliştirilmiş olan program sadece python dili ve log2timeline açık kaynak kodlu siber olay müdahale aracı kullanılarak windows işletim sistemlerinde bulunan kalıntıların analizinin gerçekleştirilmesi için kalıntıların ayrıştırılması ve olay zaman çizelgesinin çıkarılması üzerine çalışılmıştır. Siber olay müdahale anında yapılan işlemlerin sıralamasına ait çalışma diyagramı Şekil 4.6'da verilmiştir.



Şekil 4.6. Olay Müdahalesi Akış Diyagramı

4.2.2. Modüllerin Araca Tanıtılması

Siber olaylarda windows işletim sistemi makinelerinden toplamış olduğumuz belirli kalıntılar bulunmaktadır. Kalıntıların her biri ayrı ayrı modül olmak üzere bu modüllere Başlık 3'de detaylı olarak değinilmiştir. IR-Parser aracımıza modüller tek tek entegre edilmiş olup, modüllerin çalışmaları kontrol edilmiştir. Geliştirilen araca entegre edilen modüller aşağıda belirtilmiştir.

- amcache,
- lnk,
- prefetch,
- mft,
- msiecf,
- pe,
- recycle_bin,
- recycle_bin_info,
- sccm, usnjrnl,
- winevt,
- winevtx,
- winiis,
- winjob,
- winreg,
- bagmru,
- bam,
- windows_run,
- windows_sam_users,
- windows_services,
- windows_task_cache,
- mru,
- windows_task_cache,
- windows_usbstor_devices,
- winlogon,
- webhist

4.2.3. Aracın Kullanımının Tanıtılması

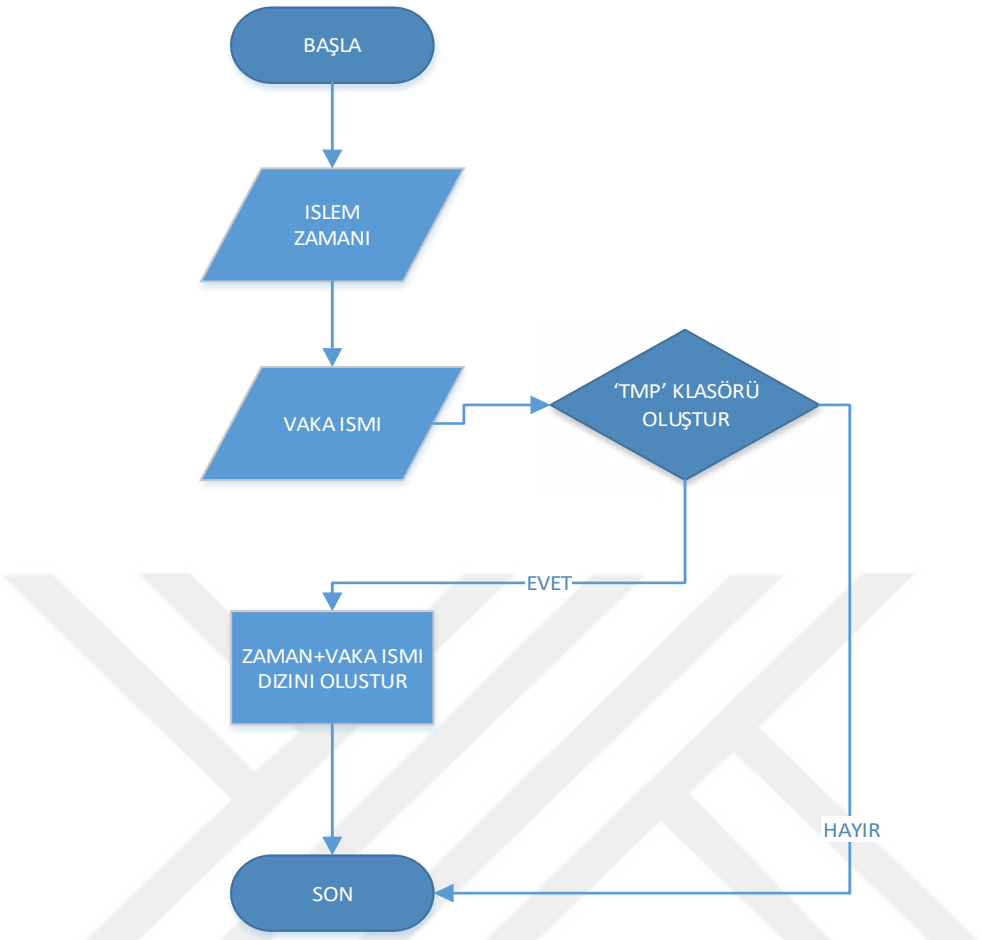
Yazılım dünyasında bir aracın veya bir programın akış diyagramı ve kullanma metodu programın help (-h) parametresi ile belirlenmekte olup Şekil 4.7'de olay müdahale aracının kullanan analizcinin sırasıyla dışarıdan vermiş olduğu analiz edilecek modül, kalıntıların ham klasör yolu, çıktıyı üreteceğimiz klasör yolu, makinenin kullanacağı işlem hız boyutu, vakanın yada makinenin isim bilgisi ve çıktı uzantı parametresi olmak üzere bilgiler alınarak programa entegre edilerek döngü oluşturulması sağlanmıştır.

```
try:
    myopts, args = getopt.getopt(sys.argv[1:],"i:o:c:p:f:")
except getopt.GetoptError as e:
    print (termcolor. colored("            !-IR-PARSER USAGE-!              \n          !-IR-PARSER
        KULLANIMI-!               ",'yellow'))
    print ("\n\t-i modules.name  "+str(modules)+ "\n\t-o output.path \n\t-c case.name \n\t-p worker.number [default:20,
        worker=25,30,35,40] \n\t-f write.format[csv or json or xlsx] ")
    screenusage = ("Example Use (Ornek Kullanım) =>>>> " + "%s -i modules.name -o output.path -c case.name -w worker.number
        -wf write.format " % sys.argv[0])
    print (termcolor.colored(screenusage,'blue'))
    sys.exit(2)

for o, a in myopts:
    if o == '-i':
        ifile=a
    elif o == '-o':
        ofile=a
    elif o == '-c':
        cfile=a
    elif o == '-p':
        processlimit=a
    elif o == '-f':
        outputformat=a
```

Şekil 4.7. Help Döngüsü ve Dışarıdan Girilen Elemanların Alınması

4.2.4. Vaka İsmine Göre Dizin Oluşturulması ve Zaman Damgasını Eklenmesi

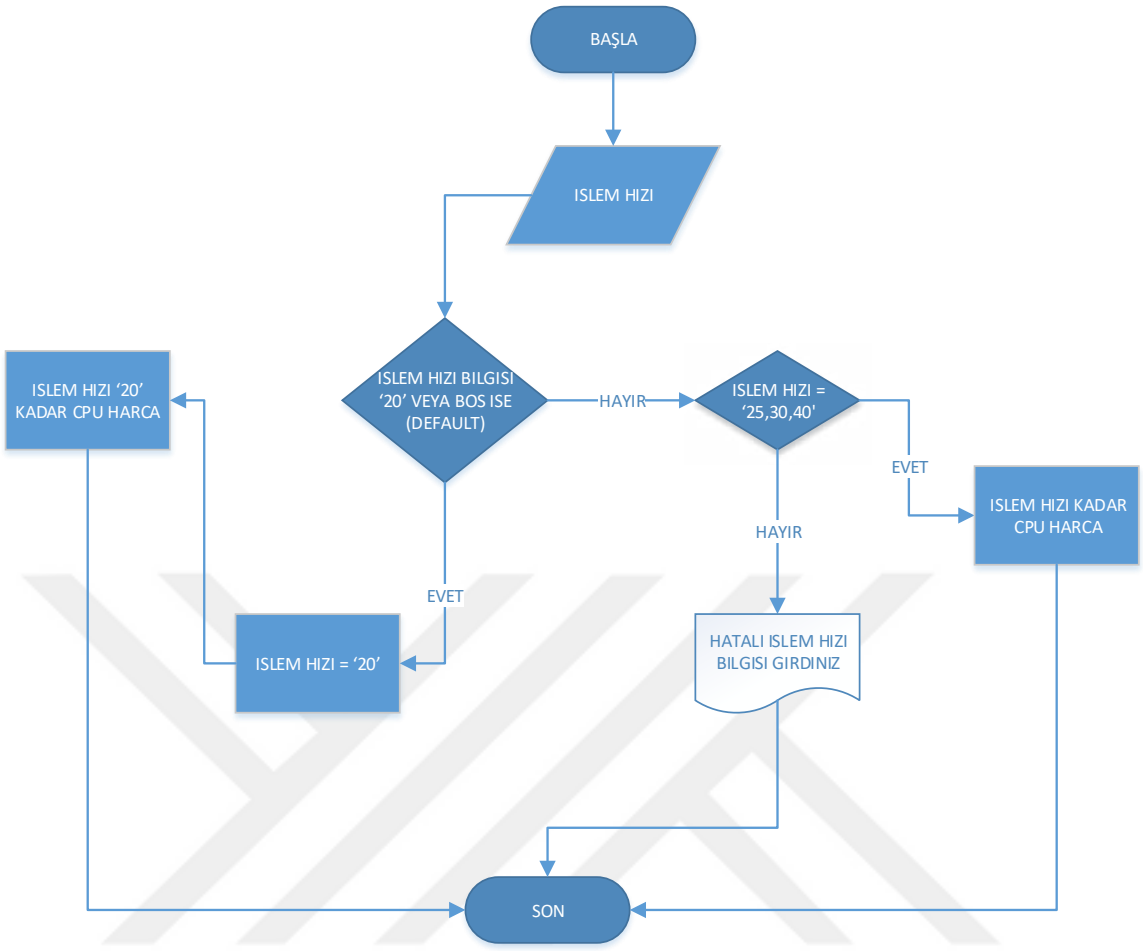
Geliştirmiş olduğumuz aracın diğer araçlarının özelliklerinden ayıran özelliği olan kalıntıların ayrıştırıldığı zaman damgası dikkate alınarak, Şekil 4.8'de dışarıdan analizcinin vermiş olduğu vaka adı veya makine adı birleştirilerek. tmp klasörünün altında parse dosyalarının oluşturulması ait akış diyagramı verilmiştir.



Şekil 4.8. Dosya Dizin Oluşturulmasına Ait Akış Diyagramı

4.2.5. Aracın İşlem Hızının Belirtilmesi

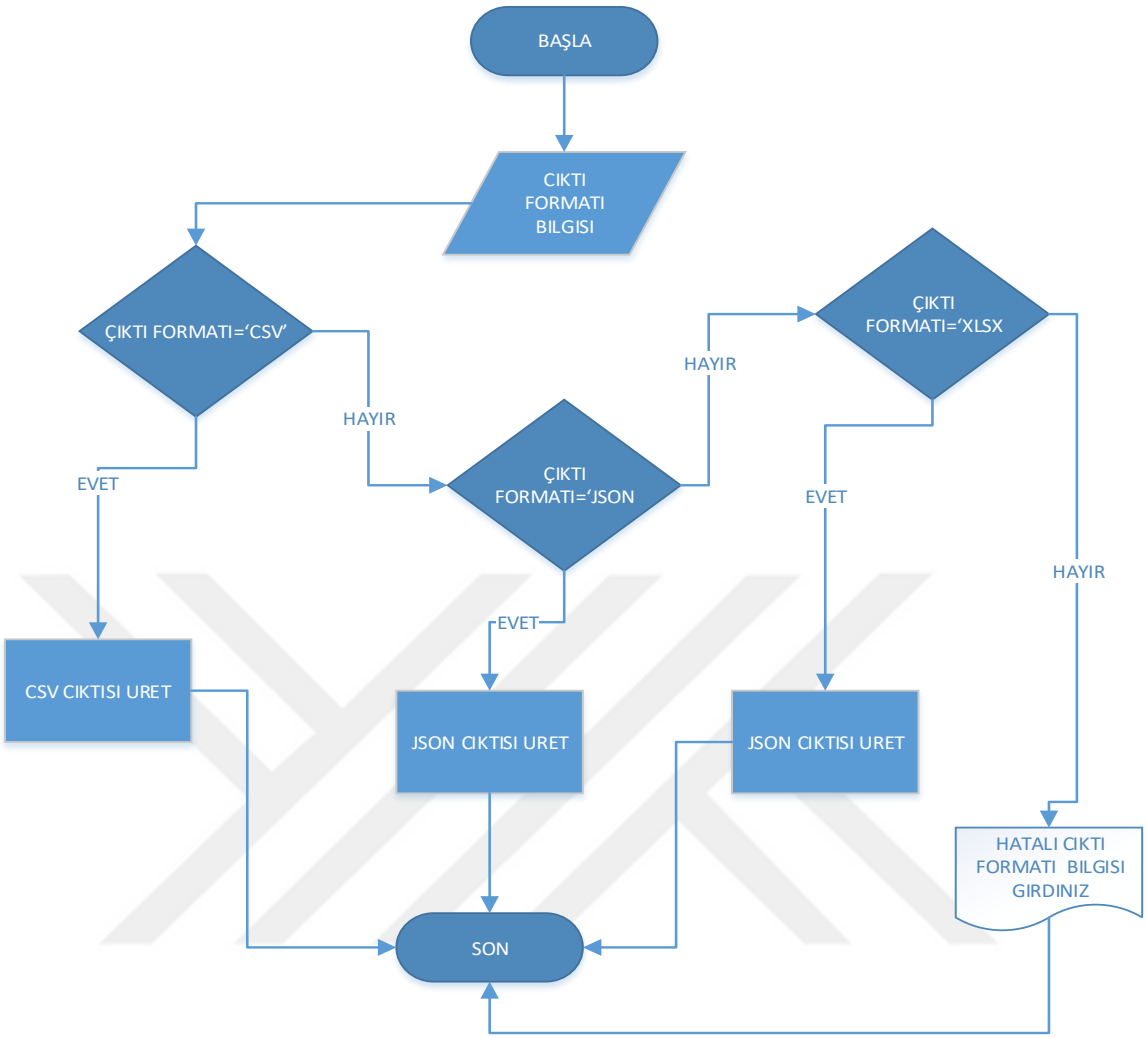
Bir analiz esnasında, aracı kullanmış olduğumuz analiz makinesinin işlemci hızı, analizin olayın çözümlenmesi açısından doğru orantılıdır. Bu nedenle makinelerin işlemci hızları önem arz etmektedir. Bir analiz makinesinde bulunabilecek işlemci hızlarının frekans değerleri tespit edilerek 20-25-30 ve 40 olmak üzere 4 farklı şekilde tanımlanmış olup Şekil 4.9'da işlem hızını modülünün çalışmasına ait akış diyagramı verilmiştir. Aracın en düşük işlem frekansı 20 işlemci hızı olarak belirlenmiş olup, son teknoloji daha hızlı işlemcili makinelerde 40'a kadar çıkmaktadır. Analizciye bağlı olarak işlem hızı parametresi –p olarak belirlenmiş olup aracın akış cümlesinde herhangi bir –p parametresi belirtilmediği durumda 20 işlem hızı ile çalışması planlanmıştır. Processlimit yani işlem hızı yanlış ya da hatalı girilmesi sonucunda, "İşlem hızı 25 30 40 olmalıdır. Default 20" ekranı dönülmektedir.



Şekil 4.9. İşlem Hızının Belirlenmesine Ait Akış Diyagramı

4.2.6. Çıktı Formatlarının Belirlenmesi

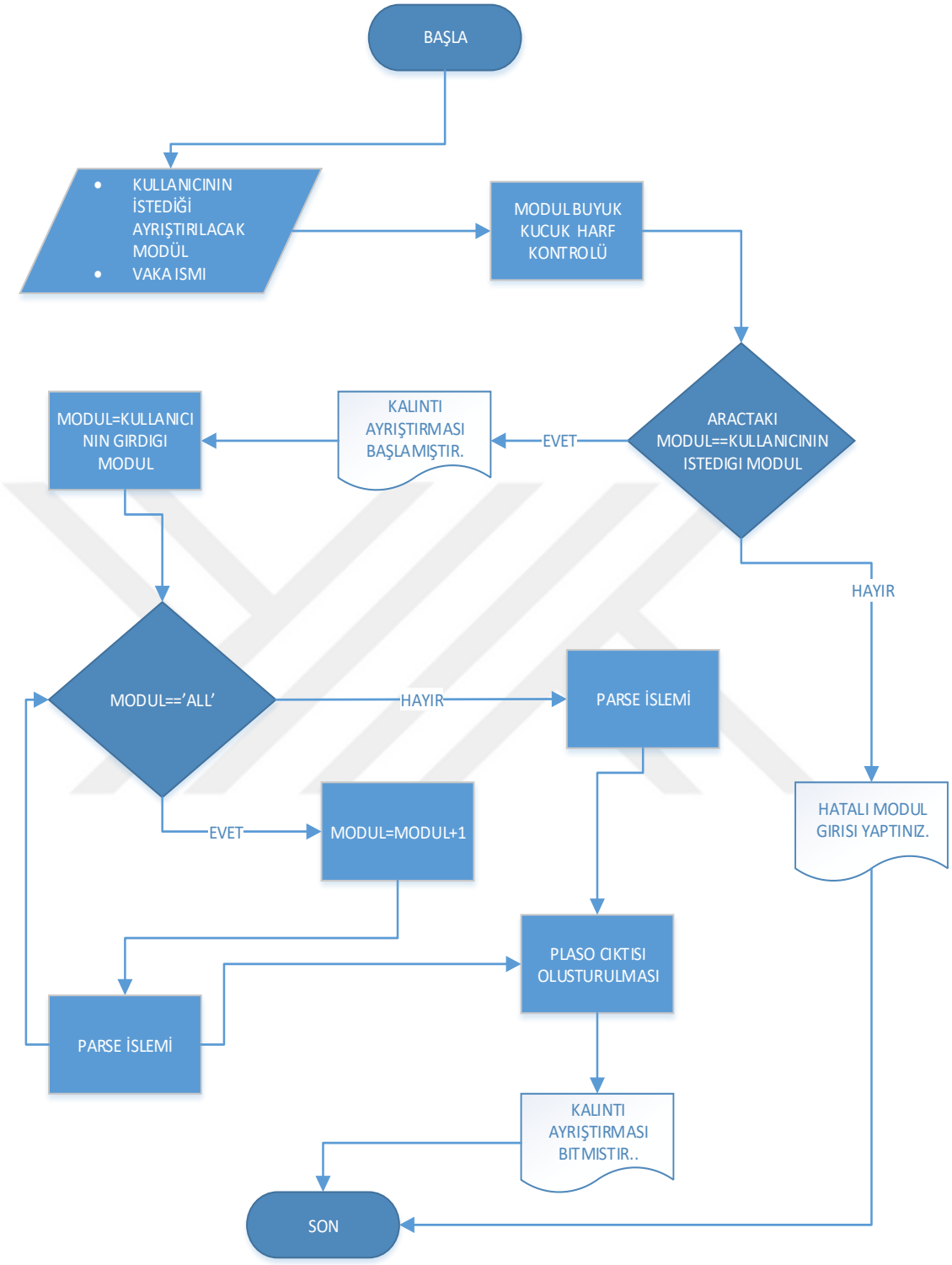
Aracın diğer olay müdahale araçlarından bir diğer farkı olan, analizcinin isteğine bağlı olarak çıktı üretebilecek formatın belirlenebilmesidir. Geliştirilen araca ayrı ayrı csv, json ve xlsx formatları entegre edilmiş olup –f parametresi ile tanımlanmıştır. Aracın akış cümlesinde –f ile parametre belirtilmediği durumlarda hata dönülerek aracın daha kararlı olarak çalışması düşünülmüştür. Hata ekranı olarak "Hatalı çıktı formatı girdiniz, lütfen csv, json, xlsx formatlarını kullanınız" dönüşü yapılmaktadır. Şekil 4.10'da geliştirilen aracın çıktı formatı modülünün çalışmasına ait akış diyagramı gösterilmiştir.



Şekil 4.10. Çıktı Oluşturulmasına Ait Akış Diyagramı

4.2.7. Modüllerin Belirlenmesi

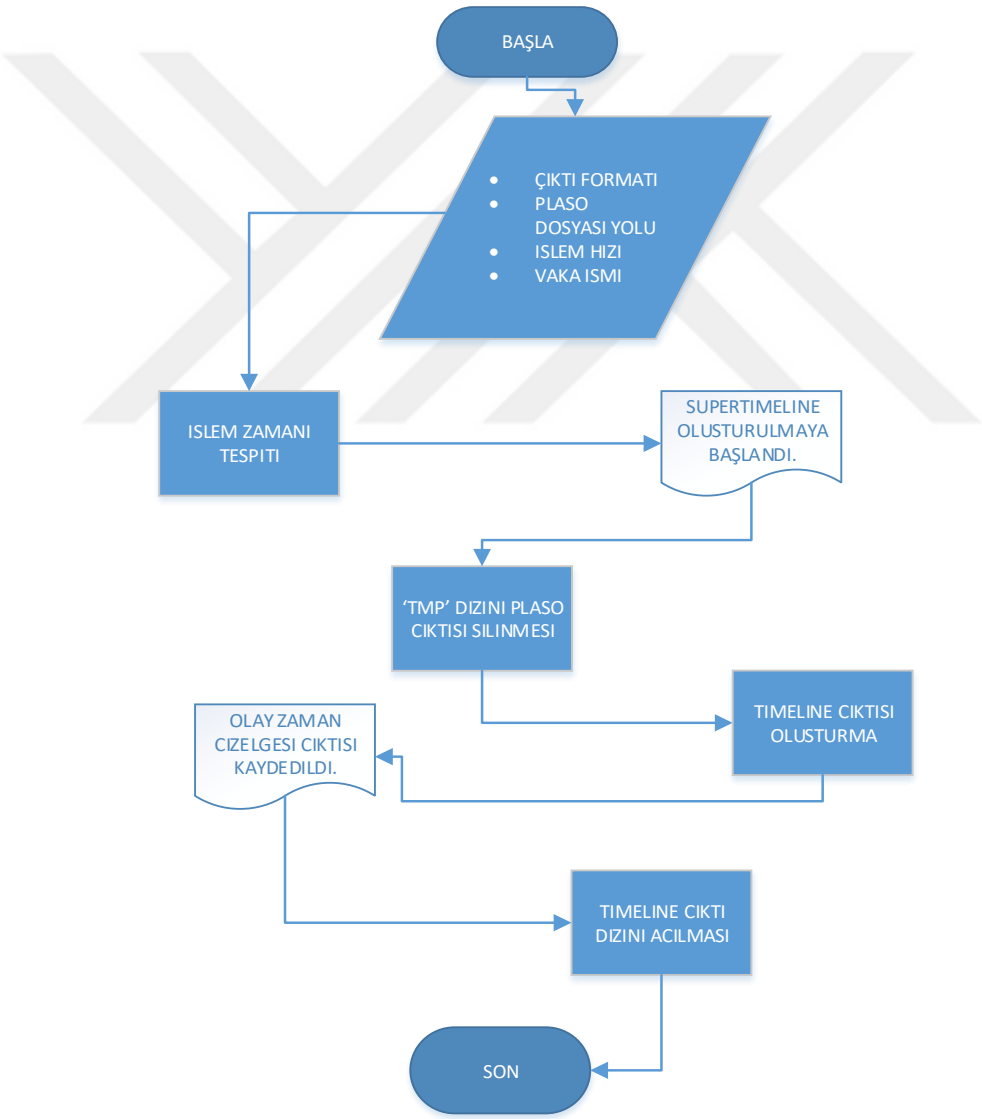
Olay müdahalesinde çoğu zaman ayrı ayrı modül incelemesinden çok tüm modüllerin incelemesi ihtiyaç duyulabilmektedir. Analizcinin aracı kullanırken ayrı ayrı analiz edilecek modülleri akış cümlesine giriş yapması kullanım kolaylığı sağlamadığından 'all' parametresi ile tüm modüller entegre edilmiş olup analizci için kullanım kolaylığı sağlanmıştır. Aynı zamanda kullanıcının bir modülü akış cümlesine –i parametresi ile giriş yapması sağlanmış olup büyük-küçük harf duyarlılığı düşünülmüş ve analizcinin modülleri büyük küçük harf karışık yazması durumunda modülün kabul edilebilirliği duyarlanmıştır. Analizci yanlış yada eksik bir modül belirtmesi durumunda ise hata ekran verilerek aracın sistematik çalışması sağlanmıştır. Modülün yanlış girilmesi durumunda ekrana " Hatalı modül girişi yapıldı modül bulunamamıştır." dönüşü yapılmaktadır. Şekil 4.11'de dışarıdan girilen ayrıştırılacak modülün, geliştirilen araçtaki kullanımına ait akış diyagramı gösterilmiştir.



Şekil 4.11. Modüllerin Kullanımı Akış Diyagramı

4.2.8. Parse İşlemi Parametreleri ve Plaso Çıktısının Silinmesi

Olay müdahalesinde kalıntıların ayrıştırılabilmesi kadar çıktının olay zaman çizelgesine göre üretilmesi de önem arz etmektedir. Ayrıştırma işleminin doğruluğu kadar çıktının içerisindeki bilgilerin bütünlüğü ve çıktının olay zaman sıralamasına göre listelenmesi önemli olup bu durum analizci açısından kolaylık, olay çözümlenmesi açısından zaman kazandırmaktadır. Şekil 4.12'de ayrıştırılan kalıntıların olay zaman çizelgesine göre çıktısının oluşturulmasına ait akış diyagramından bahsedilmiştir. Analizcinin akış cümlesinde belirtmiş olduğu çıktı formatı dikkate alınarak, vakanın adı veya makinenin ismi çıktı formatı ile birleştirilip. tmp klasörünün altında oluşması hedeflenmiştir.



Şekil 4.12. Olay Zaman Çizelgesinin Çıktısının Oluşturulması

Tablo 4.1'de geliştirilen yazılımın, başarılı işlem ve başarısız işlemlerine ait ekran çıktılarından bahsedilmiştir.

İşlem	Ekran Uyarısı
Kalıntıların Parse İşleminin Başlaması	Artifact Parser Started for "ADLIBILISIMBANK" "ADLIBILISIMBANK" için Kalıntı Ayrıştırma Başlamıştır."
Kalıntıların Parse İşleminin Bitmesi	Artifact Parser Ended for "ADLIBILISIMBANK" "ADLIBILISIMBANK" için Kalıntı Ayrıştırma Bitmiştir.
Olay - Zaman Çıktı İşleminin Başlaması	SuperTimeline Creating for "ADLIBILISIMBANK" "ADLIBILISIMBANK" için SuperTimeline Oluşturuluyor.
Olay - Zaman Çıktı İşleminin Bitmesi	SuperTimeline Saved to "ADLIBILISIMBANK" "ADLIBILISIMBANK" SuperTimeline Çıktısı Kaydedildi.
Hatalı Modül Girişi	"Be Careful modules(-i) not found examination artifacts! Dikkat! İncelenecek kalıntı,modüller (-i) arasında bulunamamıştır.
Hatalı Çıktı Formatı Girişi	Be Careful outputformat(-f) should be csv or json or xlsx! Dikkat! çıktı formatı (-f) csv, json veya xlsx olmalıdır.
Hatalı İşlem Hızı Girişi	processlimit(-p) should be 25,30,40 -- Default(20) İşlem Limiti(-p) 25-30-40 olmalıdır.

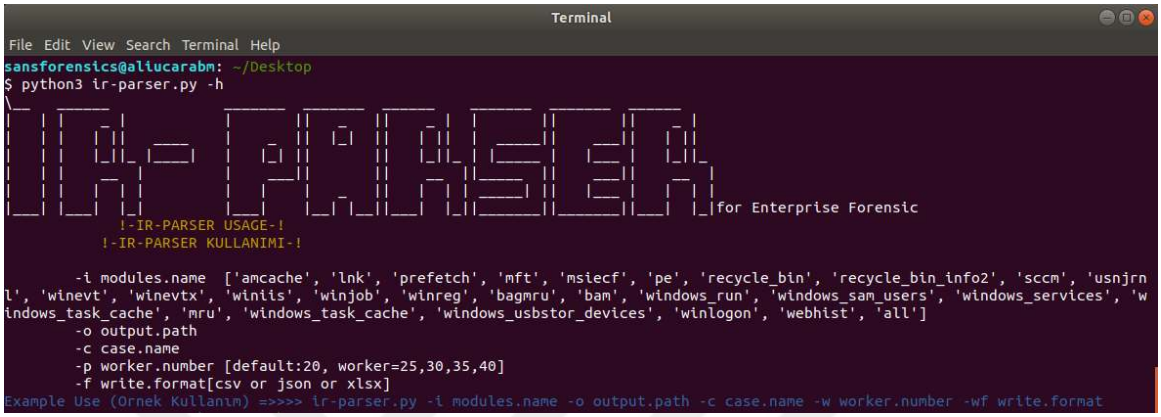
Tablo 4.1. IR-Parse Aracının Ekran Uyarıları

4.3. Windows İşletim Sistemi Kalıntılarının Ayrıştırılması ve Olay Zaman Çizelgesinin Çıkarılması

Siber adli analiz söz konusu olduğunda, süreci analistler için daha da zorlaştıran şey, büyük verinin genellikle birden fazla kaynaktan gelmesi ve farklı dosya biçimlerine sahip olmasıdır. Adli amaçlar için bu büyük miktarlardaki karmaşık verilerin analizi söz konusu olduğunda, adli analistlerin artan talepleri karşılamak için genellikle daha az zamanı ve bütçesi vardır[41].Windows işletim sistemlerinde, kullanıcının aktivitelerinin ve işletim sisteminin sistematik olarak kontrol edilmesi ve analiz edilmesi için windows veri tabanında tutulan belirli loglar bulunmaktadır. Bu loglara aynı zamanda kalıntılar (artifacts) denilmektedir. Olay müdahalesi esnasında analizci ve incelemeci olay-zaman ilişkisi içerisinde yarıştığından dolayı bu loglara kolaylıkla erişebilmeli ve incelemesini gerçekleştirmelidir. Bu başlık altında, windows işletim sistemlerinde bulunan kalıntıların olayın gerçekleştiği makinelerden toplanmasının ardından kalıntıların ayrıştırılması ve olaya ait olay-zaman çizelgesinin çıkartılıp zararlı veya şüpheli bağlantıların tespit edilmesinden bahsedilecektir.

4.3.1. Parse İşleminin Kullanımının Gösterilmesi

Bu başlık altında windows işletim sisteminde gerçekleştirilen bir siber saldırının ardından toplanmış kalıntıların geliştirmiş olduğumuz IR-Parser aracı ile ayrıştırılması anlatılmıştır. Şekil 4.13'de –h help parametresi ile analizcinin aracı kullanmadan önce hangi parametreleri hangi amaca göre kullanması gerektiği üzerine bilgilendirilmiş olup, akış cümlesi örneği gösterilmiştir.



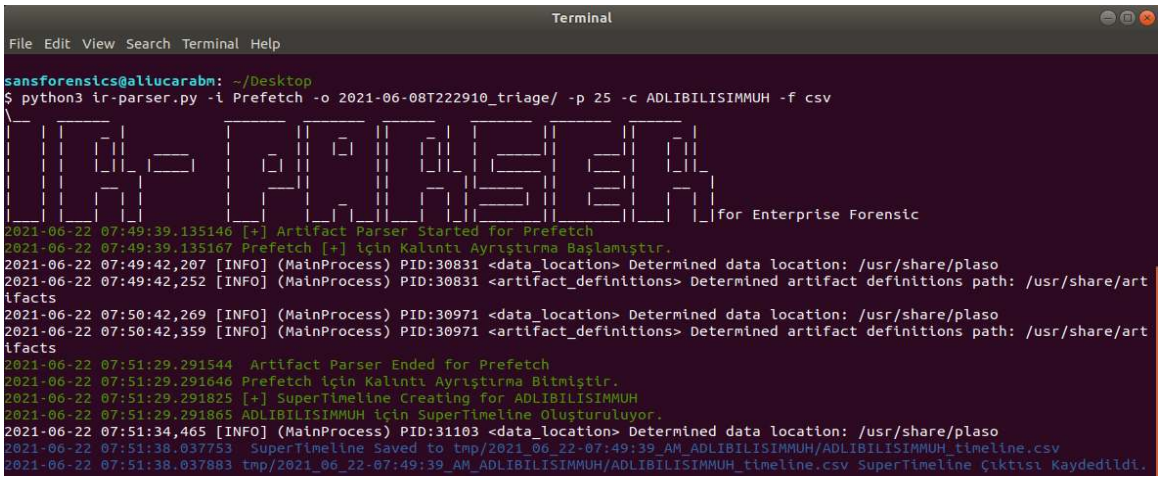
Şekil 4.13. Help Parametresi Görünümü

4.3.2. Parse İşleminin Gerçekleşmesi

Şekil 4.14'de görüldüğü üzere parse işlemi gerçekleştirilirken, analizciye aracın çalışması hakkında kısa cümleler ile bilgiler verilmiş olup aracın ayrıştırılma işlemi bittiğinde ise analizciye çıktının oluşturulduğu klasör-dizin bilgisi sunulmuştur.

Aracın örnek vaka için kullanımı şu şekildedir:

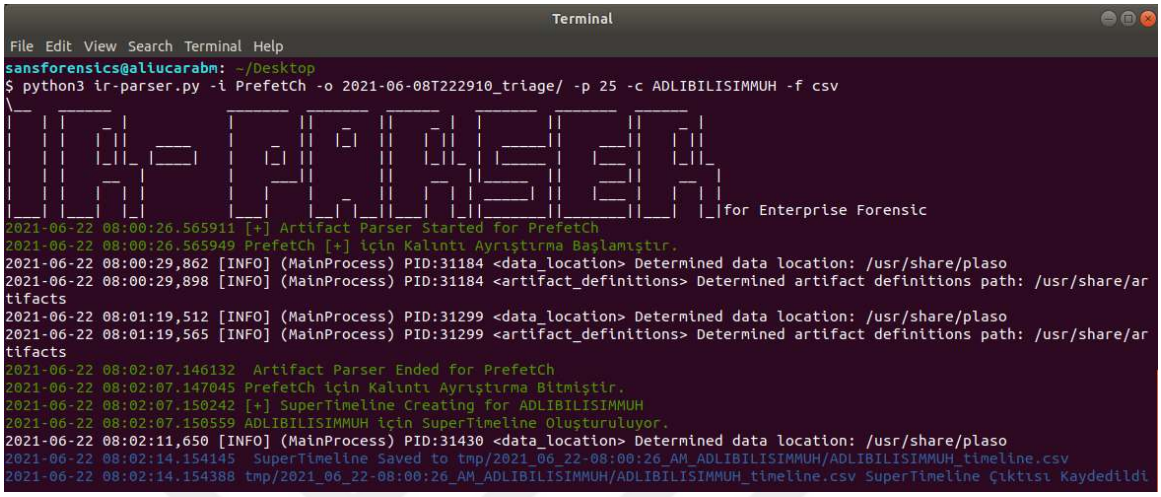
```
Python3 Ir-parser.py -i modules.name -o output.path -c case.name -w
worker.number -f write.format

Python3 Ir-parser.py –i Prefetch –o 2021-06-08T222910_triage/ -p 25
–c ADLIBILISIMMUH –f csv
```



Şekil 4.14. Parse İşlemi Gerçekleştirilmesi

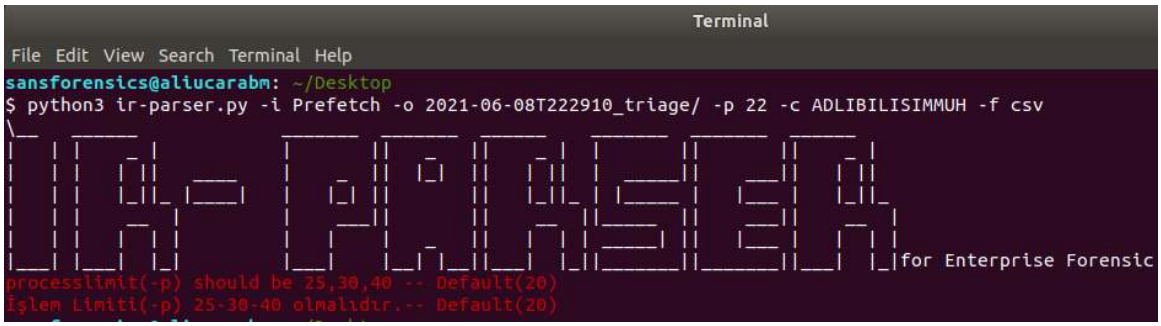
4.3.3. Girilen Parametrelerin Kontrolü

Şekil 4.15'de analizcinin modül bilgisinin büyük-küçük harf karışımı girmesi sonucu da modüller tanımlanmış olup programın kullanım kolaylığı göz önüne alınarak aracın ayrıştırma işlemini gerçekleştirdiği de görülmektedir.

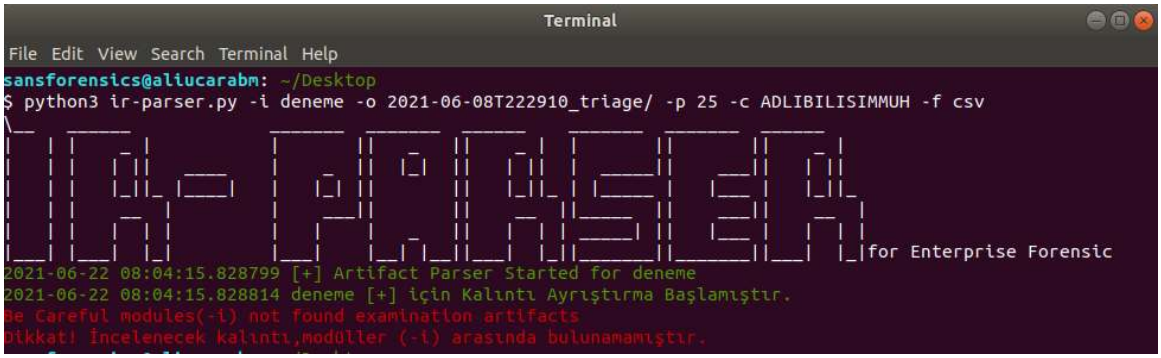


Şekil 4.15. Büyük Küçük Lower Kontrolü

Geliştirilen aracın sistematik olarak çalışması için dışarıdan girilen parametrelerin sorgulanması gerekmektedir. Analizci aracın kullanımı için dışarıdan girmiş olduğu parametrelerin sorgulanıp, doğru parametresi girilmiş ise aracın çalışması, yanlış parametre girilmiş ise uyarı dönülmesi sağlanmıştır. Örneğin Şekil 4.16'da analizcinin işlem hızını –p parametresi ile 22 girdiği görülmekte olup, aracın işlem hızı 20-25-30-40 olarak belirtilmiştir. Aracın işlem hızı koşullarına uyum sağlamadığı için hata dönülmüştür. Aynı durum Şekil 4.17'de modül bilgisinin 'deneme' girilerek incelenecek modüller içerisinde olmaması ve Şekil 4.18'de ise çıktı formatının 'html' olarak verilmesi ve çıktı format türleri arasında olmamasından dolayı hata alındığı, analizci bilgilendirilerek aracın sistematik olarak çalıştığı gösterilmiştir.



Şekil 4.16. Processlimit Kontrolü

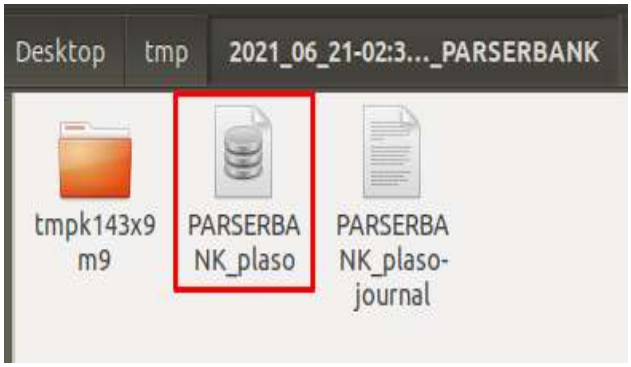


Şekil 4.17. Modül Kontrolü

Şekil 4.18. Çıktı Formatı Kontrolü

4.3.4. Kalıntı Parse Çıktısını Üretilmesi

Analizcinin aynı anda birden çok makinenin kalıntılarını ayrıştırması sonucunda elde edecek veri fazlalığının önlenmesi açısından, aracın kullanım esnasında akış cümlesinde –c parametresi ile verilen vaka ismi veya makine ismi ayrıştırma işleminin zaman damgası ile birleştirilerek .tmp dizini altında oluşturulmuş olup Şekil 4.19'da gösterilmiştir.



a-)

b)

Şekil 4.19. Çıktı Dosyalarını Oluşturulması.
a) Logların Parser Sonucu. b) Parser Çıktı Formatı Oluşturulması.

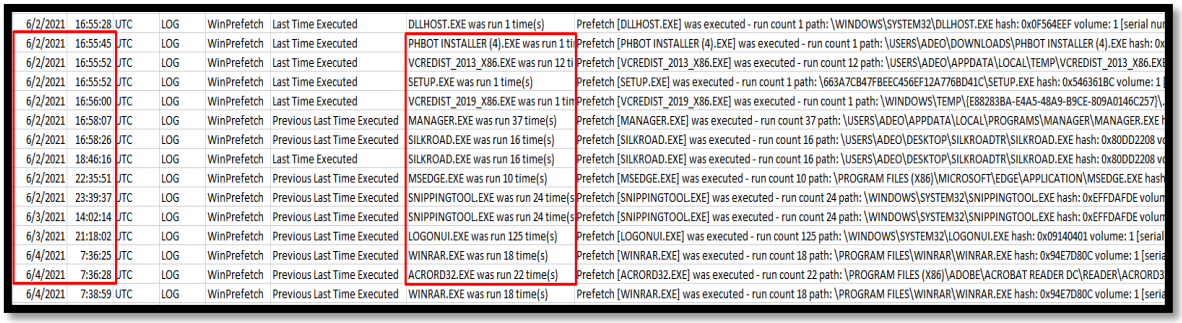
4.3.5. Olay Zaman Çizelgesi Çıktısının Üretilmesi

Tezin "Kalıntıların Parse Çıktısının Üretilmesi" bölümünde kalıntılar plaso yöntemiyle ayrıştırılmış ve vaka adı ile zaman damgası birleştirilerek tmp dizini altında kalıntı dosyası oluşturulmuştu. Bu bölümde ise oluşturulan kalıntının analizcinin istemiş olduğu çıktı formatına dönüştürülmesi gerçekleştirilmiştir. Kalıntılar parse edildikten sonra, plaso çıktısı olay zaman çizelgesine istenilen formatta dönüştürüldükten sonra Şekil 4.19'da parse dosyası silinmiş ve vaka ismi ile çıktı oluşturulmuştur. Bu durumda birden çok windows işletim sistemine ait makinenin kalıntılarının parse edilmesi sonucunda oluşabilecek karmaşıklık önlenmiştir.

4.3.6. IR-Parser ile Olay Zaman Çizelgesinin Çıkarılması

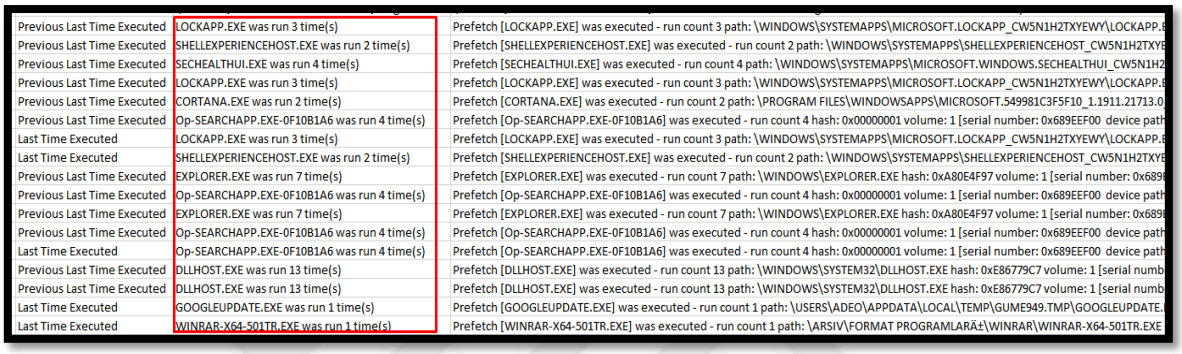
Windows işletim sistemlerinin kalıntılarının analizi için, kalıntılar toplanıp ayrıştırıldıktan sonra veri bütünlüğü bozulmadan, olaya müdahale zamanını en kısa sürede tutmak amacıyla olay zaman çizelgesinin çıkarılması gerekmektedir. Olay müdahalesini kısa tutmak ve daha sistematik bir sonucu analiz etmek için IR-PARSER aracının içinde olay zaman çizelgesini çıkarılan bir modül entegre edilmiştir. Bu modül sayesinde, kullanıcının belirtmiş olduğu kalıntının, belirtmiş olduğu çıktı formatına göre sonuç elde etmesi gerçekleştirilmiştir. İlgili çıktı tmp klasörünün altında çıktı formatı uzantısına göre belirtilmiştir.

Bir olay zaman çizelgesi incelenecek olursa, Şekil 4.20 ve Şekil 4.21'de görüldüğü üzere prefetch kalıntısının olay zaman çizelgesi çıkarılmış olup tarih saat sıralamasına göre, çalıştırılan programın ismi ve belirtilen tarihte kaç defa çalıştırılmış olduğu gösterilmiştir. Aynı zamanda Şekil 4.22'de prefetch kalıntısının parse edildiği dosya yolu ve kalıntı içerisinde bulunan çalıştırılan dosyaların sha256 hash değerleri de gösterilmektedir. Aracın bir sonraki adımında yapılması planlanan ve analizcinin de şuanda yapabileceği bir işlem olan, siber istihbarat araçları kullanılarak sha256 hash değerlerinin istihbaratları kontrol edilip, şüpheli aktivite tespitleri gerçekleştirilebilir.



6/2/2021	16:55:28	UTC	LOG	WinPrefetch	Last Time Executed	DLLHOST.EXE was run 1 time(s)	Prefetch [DLLHOST.EXE] was executed - run count 1 path: \WINDOWS\SYSTEM32\DLLHOST.EXE hash: 0x0F564EEF volume: 1 [serial nu
6/2/2021	16:55:45	UTC	LOG	WinPrefetch	Last Time Executed	PHBOT INSTALLER (4).EXE was run 1 ti	Prefetch [PHBOT INSTALLER (4).EXE] was executed - run count 1 path: \USERS\ADEO\DOWNLOADS\PHBOT INSTALLER (4).EXE hash: 0x
6/2/2021	16:55:52	UTC	LOG	WinPrefetch	Last Time Executed	VCREDIST_2013_X86.EXE was run 12 ti	Prefetch [VCREDIST_2013_X86.EXE] was executed - run count 12 path: \USERS\ADEO\APPDATA\LOCAL\TEMP\VCREDIST_2013_X86.EXE
6/2/2021	16:55:52	UTC	LOG	WinPrefetch	Last Time Executed	SETUP.EXE was run 1 time(s)	Prefetch [SETUP.EXE] was executed - run count 1 path: \663A7CB47FBEEC456EF12A776BD41C\SETUP.EXE hash: 0x546361BC volume: 1
6/2/2021	16:56:00	UTC	LOG	WinPrefetch	Last Time Executed	VCREDIST_2019_X86.EXE was run 1 tin	Prefetch [VCREDIST_2019_X86.EXE] was executed - run count 1 path: \WINDOWS\TEMP\{E88283BA-E4A5-48A9-B9CE-809A0146C257}\
6/2/2021	16:58:07	UTC	LOG	WinPrefetch	Previous Last Time Executed	MANAGER.EXE was run 37 time(s)	Prefetch [MANAGER.EXE] was executed - run count 37 path: \USERS\ADEO\APPDATA\LOCAL\PROGRAMS\MANAGER\MANAGER.EXE
6/2/2021	16:58:26	UTC	LOG	WinPrefetch	Previous Last Time Executed	SILKROAD.EXE was run 16 time(s)	Prefetch [SILKROAD.EXE] was executed - run count 16 path: \USERS\ADEO\DESKTOP\SILKROADTR\SILKROAD.EXE hash: 0x80DD2208 vo
6/2/2021	18:46:16	UTC	LOG	WinPrefetch	Last Time Executed	SILKROAD.EXE was run 16 time(s)	Prefetch [SILKROAD.EXE] was executed - run count 16 path: \USERS\ADEO\DESKTOP\SILKROADTR\SILKROAD.EXE hash: 0x80DD2208 vo
6/2/2021	22:35:51	UTC	LOG	WinPrefetch	Previous Last Time Executed	MSEDGE.EXE was run 10 time(s)	Prefetch [MSEDGE.EXE] was executed - run count 10 path: \PROGRAM FILES (X86)\MICROSOFT\EDGE\APPLICATION\MSEDGE.EXE hash
6/2/2021	23:39:37	UTC	LOG	WinPrefetch	Previous Last Time Executed	SNIPPINGTOOL.EXE was run 24 time(s	Prefetch [SNIPPINGTOOL.EXE] was executed - run count 24 path: \WINDOWS\SYSTEM32\SNIPPINGTOOL.EXE hash: 0xEFFDAFDE volum
6/3/2021	14:02:14	UTC	LOG	WinPrefetch	Previous Last Time Executed	SNIPPINGTOOL.EXE was run 24 time(s	Prefetch [SNIPPINGTOOL.EXE] was executed - run count 24 path: \WINDOWS\SYSTEM32\SNIPPINGTOOL.EXE hash: 0xEFFDAFDE volum
6/3/2021	21:18:02	UTC	LOG	WinPrefetch	Previous Last Time Executed	LOGONUI.EXE was run 125 time(s)	Prefetch [LOGONUI.EXE] was executed - run count 125 path: \WINDOWS\SYSTEM32\LOGONUI.EXE hash: 0x09140401 volume: 1 [serial
6/4/2021	7:36:25	UTC	LOG	WinPrefetch	Previous Last Time Executed	WINRAR.EXE was run 18 time(s)	Prefetch [WINRAR.EXE] was executed - run count 18 path: \PROGRAM FILES\WINRAR\WINRAR.EXE hash: 0x94E7D80C volume: 1 [seria
6/4/2021	7:36:28	UTC	LOG	WinPrefetch	Previous Last Time Executed	ACRORD32.EXE was run 22 time(s)	Prefetch [ACRORD32.EXE] was executed - run count 22 path: \PROGRAM FILES (X86)\ADOBE\ACROBAT READER DC\READER\ACRORD3
6/4/2021	7:38:59	UTC	LOG	WinPrefetch	Previous Last Time Executed	WINRAR.EXE was run 18 time(s)	Prefetch [WINRAR.EXE] was executed - run count 18 path: \PROGRAM FILES\WINRAR\WINRAR.EXE hash: 0x94E7D80C volume: 1 [seria

Şekil 4.20. Olay Zaman Çizelgesi



Previous Last Time Executed	LOCKAPP.EXE was run 3 time(s)	Prefetch [LOCKAPP.EXE] was executed - run count 3 path: \WINDOWS\SYSTEMAPPS\MICROSOFT.LOCKAPP_CW5N1H2TXYEWY\LOCKAPP.E
Previous Last Time Executed	SHELLEXPERIENCEHOST.EXE was run 2 time(s)	Prefetch [SHELLEXPERIENCEHOST.EXE] was executed - run count 2 path: \WINDOWS\SYSTEMAPPS\SHELLEXPERIENCEHOST_CW5N1H2TXYE
Previous Last Time Executed	SECHEALTHUI.EXE was run 4 time(s)	Prefetch [SECHEALTHUI.EXE] was executed - run count 4 path: \WINDOWS\SYSTEMAPPS\MICROSOFT.WINDOWS.SECHEALTHUI_CW5N1H2
Previous Last Time Executed	LOCKAPP.EXE was run 3 time(s)	Prefetch [LOCKAPP.EXE] was executed - run count 3 path: \WINDOWS\SYSTEMAPPS\MICROSOFT.LOCKAPP_CW5N1H2TXYEWY\LOCKAPP.E
Previous Last Time Executed	CORTANA.EXE was run 2 time(s)	Prefetch [CORTANA.EXE] was executed - run count 2 path: \PROGRAM FILES\WINDOWSAPPS\MICROSOFT.549981C3F5F10_1.1911.21713.0
Previous Last Time Executed	Op-SEARCHAPP.EXE-0F10B1A6 was run 4 time(s)	Prefetch [Op-SEARCHAPP.EXE-0F10B1A6] was executed - run count 4 hash: 0x00000001 volume: 1 [serial number: 0x689EEF00 device path
Last Time Executed	LOCKAPP.EXE was run 3 time(s)	Prefetch [LOCKAPP.EXE] was executed - run count 3 path: \WINDOWS\SYSTEMAPPS\MICROSOFT.LOCKAPP_CW5N1H2TXYEWY\LOCKAPP.E
Last Time Executed	SHELLEXPERIENCEHOST.EXE was run 2 time(s)	Prefetch [SHELLEXPERIENCEHOST.EXE] was executed - run count 2 path: \WINDOWS\SYSTEMAPPS\SHELLEXPERIENCEHOST_CW5N1H2TXYE
Previous Last Time Executed	EXPLORER.EXE was run 7 time(s)	Prefetch [EXPLORER.EXE] was executed - run count 7 path: \WINDOWS\EXPLORER.EXE hash: 0xA80E4F97 volume: 1 [serial number: 0x689
Previous Last Time Executed	Op-SEARCHAPP.EXE-0F10B1A6 was run 4 time(s)	Prefetch [Op-SEARCHAPP.EXE-0F10B1A6] was executed - run count 4 hash: 0x00000001 volume: 1 [serial number: 0x689EEF00 device path
Previous Last Time Executed	EXPLORER.EXE was run 7 time(s)	Prefetch [EXPLORER.EXE] was executed - run count 7 path: \WINDOWS\EXPLORER.EXE hash: 0xA80E4F97 volume: 1 [serial number: 0x689
Previous Last Time Executed	Op-SEARCHAPP.EXE-0F10B1A6 was run 4 time(s)	Prefetch [Op-SEARCHAPP.EXE-0F10B1A6] was executed - run count 4 hash: 0x00000001 volume: 1 [serial number: 0x689EEF00 device path
Last Time Executed	Op-SEARCHAPP.EXE-0F10B1A6 was run 4 time(s)	Prefetch [Op-SEARCHAPP.EXE-0F10B1A6] was executed - run count 4 hash: 0x00000001 volume: 1 [serial number: 0x689EEF00 device path
Previous Last Time Executed	DLLHOST.EXE was run 13 time(s)	Prefetch [DLLHOST.EXE] was executed - run count 13 path: \WINDOWS\SYSTEM32\DLLHOST.EXE hash: 0xE86779C7 volume: 1 [serial numb
Previous Last Time Executed	DLLHOST.EXE was run 13 time(s)	Prefetch [DLLHOST.EXE] was executed - run count 13 path: \WINDOWS\SYSTEM32\DLLHOST.EXE hash: 0xE86779C7 volume: 1 [serial numb
Last Time Executed	GOOGLEUPDATE.EXE was run 1 time(s)	Prefetch [GOOGLEUPDATE.EXE] was executed - run count 1 path: \USERS\ADEO\APPDATA\LOCAL\TEMP\GUME949.TMP\GOOGLEUPDATE.
Last Time Executed	WINRAR-X64-501TR.EXE was run 1 time(s)	Prefetch [WINRAR-X64-501TR.EXE] was executed - run count 1 path: \ARSIV\FORMAT PROGRAMLARÄ±\WINRAR\WINRAR-X64-501TR.EXE

Şekil 4.21. Runtime Tespiti

OS:/home/sansforensics/Desktop/2021-06-08T222910_triage	/C/Windows/prefetch/ACRORD32.EXE-ACF2947E.pf	prefetch	sha256_hash	ec8b7fe92c235204b43eda866cd2bbdadd891b854b8359fe6aaee5ae194087ab
OS:/home/sansforensics/Desktop/2021-06-08T222910_triage	/C/Windows/prefetch/AMDCLEANUPUTILITY.EXE-0127B38F.pf	prefetch	sha256_hash	35458586bfc83a3a0ead97b56bca733ab4222565ce2baf6de6f232848fe10b81
OS:/home/sansforensics/Desktop/2021-06-08T222910_triage	/C/Windows/prefetch/ANYDESK.EXE-7FDF62AC.pf	prefetch	sha256_hash	276e889acbc63aaecc928757469fbaf0deaf9b66fd038610ef678300d657c4ef
OS:/home/sansforensics/Desktop/2021-06-08T222910_triage	/C/Windows/prefetch/APPLICATIONFRAMEHOST.EXE-CCEEF759.pf	prefetch	sha256_hash	a8b87dc5462d021436068355499 4fd28e8d82d0a3539c7d82383046f52e72664
OS:/home/sansforensics/Desktop/2021-06-08T222910_triage	/C/Windows/prefetch/BACKGROUNDTASKHOST.EXE-508EA8A1.pf	prefetch	sha256_hash	3447812301becd9569745e990360cac3f0cc71c39a7c0db6be7a790f9d8459b7
OS:/home/sansforensics/Desktop/2021-06-08T222910_triage	/C/Windows/prefetch/CALCULATOR.EXE-8C56F4E4.pf	prefetch	sha256_hash	828afee6192a9b1c98c5e0aceccc91effe793b2324108926841583faaf022b70
OS:/home/sansforensics/Desktop/2021-06-08T222910_triage	/C/Windows/prefetch/CCLEANER64.EXE-779BD542.pf	prefetch	sha256_hash	14c4ce6c57928043b54d0d89f595e5fcf666d4defe237c1ad3189b3c7c3734eb
OS:/home/sansforensics/Desktop/2021-06-08T222910_triage	/C/Windows/prefetch/CHROME.EXE-5A1054AF.pf	prefetch	sha256_hash	e2865891ff775ebd2af1dee8b3d87181555a44c12b1a340c0b99d6202173a78d
OS:/home/sansforensics/Desktop/2021-06-08T222910_triage	/C/Windows/prefetch/CHROME.EXE-5A1054B0.pf	prefetch	sha256_hash	72fe1dc2ff615bc7fe791cddda1e92652d21b64ab6de0d75f8c1c061ab48b580
OS:/home/sansforensics/Desktop/2021-06-08T222910_triage	/C/Windows/prefetch/CHROME.EXE-5A1054B1.pf	prefetch	sha256_hash	5a5b5161292fa85407166ee2bef9b8073ba0297fce94a87b19b2f5c8cf87f447
OS:/home/sansforensics/Desktop/2021-06-08T222910_triage	/C/Windows/prefetch/CHROME.EXE-5A1054B6.pf	prefetch	sha256_hash	46721197c386f4dfd878abb93eddccb1fbaea2a2b7efbc4cdb93283d8d818914
OS:/home/sansforensics/Desktop/2021-06-08T222910_triage	/C/Windows/prefetch/CHROME.EXE-5A1054B7.pf	prefetch	sha256_hash	91b897d906dc7215c85cf64c9c1502f393c273dcb0ff2c3f38fa807a3b9cecc0
OS:/home/sansforensics/Desktop/2021-06-08T222910_triage	/C/Windows/prefetch/CMD.EXE-4A81B364.pf	prefetch	sha256_hash	760421c160f5c7ca33ec6aca3a318a06f34dcbc1227177aec231ecfe832b9866
OS:/home/sansforensics/Desktop/2021-06-08T222910_triage	/C/Windows/prefetch/CONHOST.EXE-1F3E9D7E.pf	prefetch	sha256_hash	73252a49dfca1d522cb0e90354fe2a06f3122d2710890a1377823d326fc1a9a7
OS:/home/sansforensics/Desktop/2021-06-08T222910_triage	/C/Windows/prefetch/CONSENT.EXE-531BD9EA.pf	prefetch	sha256_hash	3999ec90e2ea9f1f49c92d069991f6ffde7f02734a5a62aa68fd64e010b1519e
OS:/home/sansforensics/Desktop/2021-06-08T222910_triage	/C/Windows/prefetch/COREDPUSSVR.EXE-96FAFBD3.pf	prefetch	sha256_hash	8897decf5ffe62884a98c7c8ca80de5c7a3deac5f2ef778bda6c3e962f560e9a
OS:/home/sansforensics/Desktop/2021-06-08T222910_triage	/C/Windows/prefetch/CORTANA.EXE-3F34CB00.pf	prefetch	sha256_hash	adf02919e688cceb9fc302f335f1361297ab2f90a27445c63195c0308d528838
OS:/home/sansforensics/Desktop/2021-06-08T222910_triage	/C/Windows/prefetch/CORTANA.EXE-76E34F0C.pf	prefetch	sha256_hash	38404064ac0ca7ccc4ae0b0ad6abad8613793f596e97eec747425a4f3292bd68
OS:/home/sansforensics/Desktop/2021-06-08T222910_triage	/C/Windows/prefetch/CPU-Z_1.96-EN.TMP-6F7A85B9.pf	prefetch	sha256_hash	10f6e3b69eb97c880c33822bb20ddf358358e3faf4110b431ceb6fc40b9249fe
OS:/home/sansforensics/Desktop/2021-06-08T222910_triage	/C/Windows/prefetch/CPUZ.EXE-7A322D20.pf	prefetch	sha256_hash	dd0ff1c78d99c68b34b525487f307b99a0e03a28098b30f1b80ba9a0af6371c8

Şekil 4.22. Hash Tespiti

5. BULGULAR VE TARTIŞMA

Windows işletim sisteminde kalıntıların analizi ve ayrıştırılması (parse edilmesi) için çok yaygın olmasa da birden çok yazılım bulunmaktadır. Bu yazılımlara örnek olarak "Kape, Log2timeline, ANJP" gibi yazılımlar verilebilir. Yazılımların birbirinden farklılıkları çok fazla olmasa da benzerlikleri farklılık oranına göre daha fazladır. Kape yazılımı diğer yazılımlardan ayıran en belirgin özelliği grafiksel tabanlı olup windows işletimde çalışabilmektedir. Yazılımların yaptıkları işlevler ve çalıştıkları alanlar birbirinden aynı olup Kape yazılımı diğer yazılımlara göre daha fazla ün kazanmıştır.

Log2timeline yazılım, Linux tabanlı olup açık kaynak kodlu bir yazılımdır. Siber olay müdahale yazılımları olarak geliştirmiş olduğumuz araç ile log2timeline arasındaki benzerlikleri ve farklılıkları listelendiği bir tablo oluşturulmuştur. Bu tablo Tablo 5.1' de gösterilmiştir.

	Log2Timeline	**IR-Parser**
Modül Girişi	Var	Var
Vaka İsmi	Yok	Var
Zaman Damgası	Yok	Var
Dizin Oluşturma	Var	Var
Dizini Vaka İsmi ve Zaman'a Göre Oluşturma	Yok	Var
İşlem Hızı	Var	Var
Modül Kontrolü	Var	Var
Çıktı Formatı Kontrolü	Yok	Var
İşlem Hızı Kontrolü	Yok	Var
Olay Zaman Çizelgesini Çıkarılması	Var	Var
Dışarıdan Kalıntıların Alınması	Var	Var
Büyük Küçük Harf Kontrolü	Yok	Var
Programın Help Bilgilendirmesi	Var	Var
Linux'un Tüm Sürümlerinde Çalışmakta	Var	Var
Windows Kalıntılarının Hepsini Analiz Etmekte	Var	Var
Kullanım Durumu	Açık Kaynak	Açık Kaynak

Tablo 5.1. Geliştirilen Uygulama ile Log2timeline Kıyaslama

Tabloda görüleceği üzere bu iki yazılımın birbirinden avantajları ve dezavantajları bulunmaktadır. IR-Parser aracı diğer yazılımlara göre belirgin farklılıkları daha fazla kullanıcı dostu olmasıdır. Modül kontrolü, çıktı kontrolü, işlem hızı kontrolü gibi aracı kullanan analistin giriş parametrelerinde yanlış yazım hataları yapabileceği durumlar düşünülmüş olup, yazılımın daha kullanışlı ve daha düşük hata oranı sağlanmaya çalışılmıştır.

Kape ve ANJP gibi yazılımlar windows tabanlı çalışmakta olup geliştirilen yazılım ve log2timeline yazılımına göre daha yavaş çalıştıkları tespit edilmiştir. Bunun ana nedeni windows tabanlı olmalarından kaynaklanmaktadır. Siber olay müdahalesinde kullanılan ün kazanmış yazılımların birbirinden farklılıkları ve benzerliklerine ait tablo oluşturulmuştur. Benzerlikler ve farklılıkları anlatan bilgiler Tablo 5.2'de gösterilmiştir.

	KAPE	**ANJP**	**Log2timeline**	**IR-Parse**
Windows Tabanlı	Var	Var	Yok	Yok
Linux Tabanlı	Yok	Yok	Var	Var
Kütüphane Kurulu	Yok	Yok	Var	Var
Gui Tabanlı	Var	Var	Yok	Yok
Cli Tabanlı	Var	Yok	Var	Var
Log Dosyalarının Analizi	Var	Var	Var	Var
Tüm Kalıntıların Analizi	Var	Yok	Var	Var
Olay Zaman Çizelgesi Çıkarılması	Yok	Yok	Var	Var
Çıktı Formatı İstemesi	Yok	Yok	Var	Var
Makineden Log Toplama	Var	Yok	Yok	Yok
Hash Doğrulama	Var	Yok	Yok	Yok
IOC Çıktısı	Var	Yok	Var	Var
Kullanım Durumu	Ücretsiz	Ücretsiz	Açık Kaynak	Açık Kaynak

Tablo 5.2. Siber Olay Müdahale Araclarının Karşılaştırılması

Tablo 5.2'de görüldüğü üzere windows işletim sistemindeki logların ve kalıntıların analizinde kullanılan yazılımların farklılıkları ve benzerlikleri belirtilen parametrelere göre çok fazladır. Bu yazılımlar arasında belirlenen parametrenin yazılımda bulunma durumu en çok olan Kape yazılımı olduğu görülmektedir. Bu nedenle Kape yazılımı olay müdahalelerinde analistin isteklerinin genellikle daha fazla karşıladığı için kullanıldığı düşünülmektedir. Ek olarak yukarıdaki yazılımların karşılaştırmaları yazılımların son sürümleri ile kıyaslanmıştır.

6. SONUÇLAR

Bu tez çalışmasında, bir windows işletim sisteminin kurulumundan son kullanıcıya kadar geçen evrede, işletim sisteminin kayıt defteri, kullanıcı bilgileri, ağ kayıtları, çalıştırılan uygulamalar ve son çalıştırılan uygulamaların yapmış olduğu değişiklikler, silinen dosyalar, kopyalanan dosyalar vb. gibi kullanıcı-sistem aktivite kayıtlarının bir vakanın incelenmesi açısından önemine değinilmiştir. Siber olaylara müdahale de hızlı aksiyon alınabilmesinin önemi üzerinde durularak milli yazılım olan "Siber Olaylara Müdahale Aracı" geliştirilmiştir. Bu araç geliştirilirken siber olay müdahalesi ile ilgili şu sonuçlara ulaşılmıştır.

- Aktiviteler kronolojik sıralanmış hali ile bir vakanın gerçekleşmesinin ardından kalıntıların toplanması, ayrıştırılması (parse edilmesi) ve olay zaman çizelgesinin çıkarılması üzerine bir siber olay müdahale aracı geliştirilmiştir.
- Windows işletim sistemlerinde gerçekleşen aktivelerde olay günlüklerinin ve kalıntıların önemi belirtilmiş olup herhangi bir gerçekleşen aktivite de adli kopya almak yerine triyaj kayıtları ile daha hızlı aksiyon alınabildiği tespit edilmiştir.
- Geliştirilen IR-Parse aracına, işlem hızı, vaka ismi ve işlem sonucunda fazla dosyaların silinmesi, çıktı formatlarının analistin isteğine göre verilebilmesi gibi yeni özellikler eklenerek aracın kullanıcı dostu olduğu tespit edilmiştir.
- Windows işletim sisteminde bulunan kalıntıların çeşitliliği günden güne Windows geliştikçe arttığı görülmüş olup yeni oluşabilecek Windows türlerinde daha fazla kalıntı olabileceği sonucuna varılmıştır.
- Geliştirilen uygulama ücretsiz ve açık kaynak kodlu bir şekilde geliştirilmiştir. Mevcutta bulunan belirli kütüphaneleri kullanarak python dili ile makineden olay günlüğü kayıtlarının toplanabileceği ve kayıtların analiz edilebileceği görülmüştür.
- Olay zaman çizelgesi çıkarılarak inceleme yapılan makinelerde çalıştırılan dosyaların hash değerlerinin ayrıştırılarak siber istihbarat araçlarında istihbaratlarının kontrol edilebildiği tespit edilmiştir.

ÖNERİLER

Siber güvenlik dünyasında her gün yeni bir ürün ortaya çıkmakta veya kurumlarda hâlihazırda kullanılan ürünlerin her gün farkı bir sürümü yaygınlaşmaktadır. Siber güvenliğin gelişmesine bağlı olarak insanlar tarafından işlenen suçlar günden güne artmaktadır. Gelişen ve şuanda kurumlarda kullanılan siber güvenlik ürünlerinden kaynaklı olarak siber saldırıları, sistemlerin zafiyetleri, kullanıcı aktiviteleri, sızıntıları vb. birçok olay logları günden güne çoğalmakta ve farklı olaylara şahitlik etmektedir. Siber olay müdahalesi esnasında farklı log kaynakları görülebilmekte, bu log kaynaklarının incelenmesi, analiz edilmesi zaman açısından uzun sürmekte ve bu nedenle zorluklar yaşanmaktadır.

Sürekli gelişen işletim sistemlerinde bulunan uygulamaların ve işletim sistemi loglarının yenileri ortaya çıktıkça bu logların analiz edilebilmesi için log kaynaklarının geliştirilen araca entegre edilmesi ve özelliklerinin arttırılması, aracın kullanım kararlılığı açısından önerilmektedir.

Siber güvenlik alanında bir sistemin etkilenmesi için kullanılan en büyük zafiyet, insan zafiyetidir. Güvenlik çözümleri kullanan bir kurumdaki kullanıcıların insan zafiyetine dair bilgilendirilmesi önerilmektedir.

Siber olay müdahalesi esnasında hızlı aksiyon alınabilecek milli yazılımlar yazılarak, logların bütünlüğü bozulmadan hızlı bir şekilde olaya müdahale edilmesi beklenmektedir.

Siber olay müdahale aracını sadece kullanmakla kalmayıp, elde edilen IOC (hash, ip, domain) bilgilerini de içeren siber istihbarat ürünlerinin geliştirilmesi ve akabinde bu IOC'lerin siber istihbaratlarının kontrol edilmesi önerilmektedir.

Geliştirilen IR-Parser aracına ek bir özellik olarak hash, ip, domain bilgilerini analiste hali hazırda bir csv dosyası halinde sunabilecek bir özellik eklenmesi, tespit edilen şüpheli dosyaların sandbox (kum havuzu) ortamında analiz edilebilecek bir özelliğin eklenmesi, şüpheli veya zararlı gördüğü dosya bilgilerini ve aktivite bilgilerini blacklist dediğimiz bir ortamda tutulmasını sağlayan özelliğin eklenmesi, Olay- Zaman çizelgesinde elde edilen hash, ip, domain bilgilerinin False Positive (FP) durumlarından ayıklanması, şüpheli verilerden elde edilen istihbarat bilgilerinin pdf dosyasında birleştirilip bir çıktı oluşturulması, aracın geliştirilebilirliği açısından önerilmektedir.

KAYNAKLAR

[1] J. Srinivas, A.K. Das, N. Kumar, Government regulations in cyber security: Framework, standards and recommendations, Futur. Gener. Comput. Syst. 92 (2019). https://doi.org/10.1016/j.future.2018.09.063.

[2] R.D. Pittman, D. Shaver, Windows forensic analysis, in: Handb. Digit. Forensics Investig., 2010. https://doi.org/10.1016/B978-0-12-374267-4.00005-7.

[3] F. Olajide, N. Savage, Forensic extraction of user information in continuous block of evidence, in: Int. Conf. Inf. Soc. i-Society 2011, 2011. https://doi.org/10.1109/i-society18435.2011.5978501.

[4] M. Alfosail, P. Norris, Tor forensics: Proposed workflow for client memory artefacts, Comput. Secur. 106 (2021). https://doi.org/10.1016/j.cose.2021.102311.

[5] B. Nikkel, Fintech forensics: Criminal investigation and digital evidence in financial technologies, Forensic Sci. Int. Digit. Investig. 33 (2020). https://doi.org/10.1016/j.fsidi.2020.200908.

[6] A. Bhardwaj, S. Goundar, Keyloggers: silent cyber security weapons, Netw. Secur. 2020 (2020). https://doi.org/10.1016/S1353-4858(20)30021-0.

[7] H. Naseer, S.B. Maynard, K.C. Desouza, Demystifying analytical information processing capability: The case of cybersecurity incident response, Decis. Support Syst. 143 (2021). https://doi.org/10.1016/j.dss.2020.113476.

[8] T. Vidas, C. Zhang, N. Christin, Toward a general collection methodology for Android devices, Digit. Investig. 8 (2011). https://doi.org/10.1016/j.diin.2011.05.003.

[9] P. Turner, Applying a forensic approach to incident response, network investigation and system administration using Digital Evidence Bags, Digit. Investig. 4 (2007). https://doi.org/10.1016/j.diin.2007.01.002.

[10] C.H. Ngejane, J.H.P. Eloff, T.J. Sefara, V.N. Marivate, Digital forensics supported by machine learning for the detection of online sexual predatory chats, Forensic Sci. Int. Digit. Investig. 36 (2021). https://doi.org/10.1016/j.fsidi.2021.301109.

[11] E. Casey, Chapter 20 OF Digital Evidence and Computer Crime: Forensic Science, Computers and the Internet, 2011.

[12] R. Schramp, Live transportation and RAM acquisition proficiency test, Digit. Investig. 20 (2017) 44–53. https://doi.org/10.1016/j.diin.2017.02.006.

[13] D. Jeong, S. Lee, Forensic signature for tracking storage devices: Analysis of UEFI firmware image, disk signature and windows artifacts, Digit. Investig. 29 (2019). https://doi.org/10.1016/j.diin.2019.02.004.

[14] F. Olajide, N. Savage, G. Akmayeva, C. Shoniregun, Digital forensic research - The analysis

of user input on volatile memory of Windows application, in: World Congr. Internet Secur. WorldCIS-2012, 2012.

[15] G. Horsman, Tool testing and reliability issues in the field of digital forensics, Digit. Investig. 28 (2019) 163–175. https://doi.org/10.1016/j.diin.2019.01.009.

[16] L. Naiqi, W. Zhongshan, H. Yujie, Q. Ke, Computer forensics research and implementation based on NTFS file system, in: Proc. - ISECS Int. Colloq. Comput. Commun. Control. Manag. CCCM 2008, 2008. https://doi.org/10.1109/CCCM.2008.236.

[17] J.M. Aquilina, E. Casey, C.H. Malin, Malware Forensics: Investigating and Analyzing Malicious Code, 2008. https://doi.org/10.1016/B978-1-59749-268-3.X0001-1.

[18] K. Awuson-David, T. Al-Hadhrami, M. Alazab, N. Shah, A. Shalaginov, BCFL logging: An approach to acquire and preserve admissible digital forensics evidence in cloud ecosystem, Futur. Gener. Comput. Syst. 122 (2021). https://doi.org/10.1016/j.future.2021.03.001.

[19] G.S. Cho, M.K. Rogers, Finding forensic information on creating a folder in $LogFile of NTFS, in: Lect. Notes Inst. Comput. Sci. Soc. Telecommun. Eng., 2012. https://doi.org/10.1007/978-3-642-35515-8_18.

[20] C. Lees, Determining removal of forensic artefacts using the USN change journal, Digit. Investig. 10 (2013). https://doi.org/10.1016/j.diin.2013.10.002.

[21] B. Singh, U. Singh, Leveraging the Windows Amcache.hve File in Forensic Investigations, J. Digit. Forensics, Secur. Law. (2016). https://doi.org/10.15394/jdfsl.2016.1429.

[22] J. Du, J. Li, Analysis the Structure of SAM and Cracking Password Base on Windows Operating System, Int. J. Futur. Comput. Commun. (2016). https://doi.org/10.18178/ijfcc.2016.5.2.455.

[23] R.M. Saidi, S.A. Ahmad, N.M. Noor, R. Yunos, Windows registry analysis for forensic investigation, in: 2013 Int. Conf. Technol. Adv. Electr. Electron. Comput. Eng. TAEECE 2013, 2013. https://doi.org/10.1109/TAEECE.2013.6557209.

[24] Y. Khatri, Forensic implications of System Resource Usage Monitor (SRUM) data in Windows 8, Digit. Investig. (2015). https://doi.org/10.1016/j.diin.2015.01.002.

[25] B. Dolan-Gavitt, Forensic analysis of the Windows registry in memory, Digit. Investig. 5 (2008). https://doi.org/10.1016/j.diin.2008.05.003.

[26] A. Singh, H.S. Venter, A.R. Ikuesan, Windows registry harnesser for incident response and digital forensic analysis, Aust. J. Forensic Sci. (2020). https://doi.org/10.1080/00450618.2018.1551421.

[27] H. Carvey, Windows Forensic Analysis DVD Toolkit, 2007. https://doi.org/10.1016/B978-1-59749-156-3.X5000-X.

[28] R.B. van Baar, W. Alink, A.R. van Ballegooij, Forensic memory analysis: Files mapped in

memory, Digit. Investig. (2008). https://doi.org/10.1016/j.diin.2008.05.014.

[29] J.D. Kornblum, Using every part of the buffalo in Windows memory analysis, Digit. Investig. (2007). https://doi.org/10.1016/j.diin.2006.12.002.

[30] K.P. Chow, F.Y.W. Law, M.Y.K. Kwan, P.K.Y. Lai, The rules of time on NTFS file system, in: Proc. - SADFE 2007 Second Int. Work. Syst. Approaches to Digit. Forensic Eng., 2007. https://doi.org/10.1109/SADFE.2007.22.

[31] B. Singh, U. Singh, A forensic insight into Windows 10 Jump Lists, Digit. Investig. 17 (2016). https://doi.org/10.1016/j.diin.2016.02.001.

[32] G. Horsman, Raiders of the lost artefacts: Championing the need for digital forensics research, Forensic Sci. Int. Reports. 1 (2019). https://doi.org/10.1016/j.fsir.2019.100003.

[33] P. Shedden, A. Ahmad, A.B. Ruighaver, Organisational learning and incident response: Promoting effective learning through the incident response process, in: Proc. 8th Aust. Inf. Secur. Manag. Conf., 2010.

[34] N. Shinde, P. Kulkarni, Cyber incident response and planning: a flexible approach, Comput. Fraud Secur. 2021 (2021). https://doi.org/10.1016/S1361-3723(21)00009-9.

[35] A. Dimitriadis, N. Ivezic, B. Kulvatunyou, I. Mavridis, D4I - Digital forensics framework for reviewing and investigating cyber attacks, Array. 5 (2020). https://doi.org/10.1016/j.array.2019.100015.

[36] URL-1, https://www.kroll.com/en/services/cyber-risk/incident-response-litigation - support/kroll-artifact-parser-extractor-kapeNo Title, (n.d.).

[37] M. Debinski, F. Breitinger, P. Mohan, Timeline2GUI: A Log2Timeline CSV parser and training scenarios, Digit. Investig. 28 (2019). https://doi.org/10.1016/j.diin.2018.12.004.

[38] URL-2, https://github.com/log2timeline/plaso, Erişim: 15.06.2021. (n.d.).

[39] A. Naseer, H. Naseer, A. Ahmad, S.B. Maynard, A. Masood Siddiqui, Real-time analytics, incident response process agility and enterprise cybersecurity performance: A contingent resource-based analysis, Int. J. Inf. Manage. 59 (2021). https://doi.org/10.1016/j.ijinfomgt.2021.102334.

[40] G. Grispos, W.B. Glisson, T. Storer, Enhancing security incident response follow-up efforts with lightweight agile retrospectives, Digit. Investig. 22 (2017). https://doi.org/10.1016/j.diin.2017.07.006.

[41] N.M. Karie, V.R. Kebande, H.S. Venter, Diverging deep learning cognitive computing techniques into cyber forensics, Forensic Sci. Int. Synerg. 1 (2019). https://doi.org/10.1016/j.fsisyn.2019.03.006.

ÖZGEÇMİŞ

Ali Hüsamiddin UÇAR

[redacted]

[redacted] [redacted]
[redacted] [redacted]
[redacted] [redacted]
[redacted] [redacted]
[redacted] [redacted]
[redacted] [redacted]

[redacted]

[redacted] [redacted]
[redacted] [redacted]

[redacted]

[redacted] [redacted]
[redacted] [redacted]

[redacted]

- [redacted]
- [redacted]

[redacted]

[redacted] [redacted].)