T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SİBER SALDIRI TÜRLERİNİN SİMÜLE EDİLMESİ VE YARA İLE TESPİTİ

Mustafa Emre DEMİR

Yüksek Lisans Tezi

ADLİ BİLİŞİM MÜHENDİSLİĞİ ANABİLİM DALI

TEMMUZ 2022

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Adli Bilişim Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

SİBER SALDIRI TÜRLERİNİN SİMÜLE EDİLMESİ VE YARA İLE TESPİTİ

Tez Yazarı
Mustafa Emre DEMİR

Danışman
Doç. Dr. Türker TUNCER

TEMMUZ 2022
ELAZIĞ

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Adli Bilişim Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Başlığı:	Siber Saldırı Türlerinin Simüle Edilmesi ve YARA ile Tespiti
Yazarı:	Mustafa Emre DEMİR
İlk Teslim Tarihi:	04.06.2022
Savunma Tarihi:	04.07.2022

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. Türker TUNCER — Onayladım
Fırat Üniversitesi Teknoloji Fakültesi

Başkan: Doç. Dr. Şengül DOĞAN — Onayladım
Fırat Üniversitesi Teknoloji Fakültesi

Üye: Dr. Öğr. Üyesi Fahrettin Burak DEMİR — Onayladım
Bandırma Onyedi Eylül Üniversitesi Mühendislik ve Doğa Bilimleri Fakültesi

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

İmza

Prof. Dr. Kürşat Esat ALYAMAÇ
Enstitü Müdürü

BEYAN

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım "Siber Saldırı Türlerinin Simüle Edilmesi ve YARA ile Tespiti" 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.

04.07.2022

Mustafa Emre DEMİR

ÖNSÖZ

Bu tez çalışması Fırat Üniversitesi Fen Bilimleri Enstitüsü Adli Bilişim Mühendisliği Anabilim Dalı Yüksek Lisans Programı'nda hazırlanmıştır.

Bu çalışmanın ortaya çıkmasında katkısı olan ve tez çalışmalarım süresince benden yardımlarını esirgemeyen değerli hocam ve tez danışmanım Doç. Dr. Türker TUNCER'e teşekkürlerimi sunarım.

Ayrıca tez çalışmam ve makale yazım süresince okuma ve eleştirileriyle beni yönlendiren, desteklerini esirgemeyen saygıdeğer hocam Doç. Dr. Şengül DOĞAN'a teşekkürlerimi sunarım.

Yaşamımın tüm safhalarında desteklerini benden esirgemeyen anne ve babama minnet ve şükranlarımı sunarım.

Mustafa Emre DEMİR
ELAZIĞ, 2022

İÇİNDEKİLER

Sayfa

ÖZET

Siber Saldırı Türlerinin Simüle Edilmesi ve YARA ile Tespiti

Mustafa Emre DEMİR

Yüksek Lisans Tezi

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Adli Bilişim Mühendisliği Anabilim Dalı

Temmuz 2022, Sayfa: xi + 48

Bilişim sistemlerinin oldukça hızlı gelişimi, kurum ve kuruluşların tüm iş süreçlerini bilişim sistemleri üzerine taşımasını ve tüm süreçlerin bu sistemler üzerinde yürütülmesini ortaya çıkarmıştır. Bununla birlikte sistemler üzerinde bulunan şirket verilerinin ve kişisel verilerin güvenliğini sağlamak da bir sektör haline gelmiştir. Aynı zamanda kuruluşların bilişim sistemlerindeki verilerini hedef alan kötü niyetli kişi veya gruplar tarafından verilerin güvenliği tehlikeye atılmaktadır. Kuruluşların bu saldırgan kişi veya gruplar tarafından hedef alınması her geçen gün daha da artmaktadır. Dolayısıyla kuruluşlar kendi sistemlerini korumak amacıyla siber güvenlik için çeşitli çözümler kullanmaya başlamışlardır. Ancak bu çözümler sürdürülebilir bir siber güvenlik alt yapısı ve fiziksel ekibi olmayan kuruluşlar için yeterli değildir ve bu kuruluşlar saldırganların hedefi olmaktan kaçınamazlar.

Saldırganların hedefi olmuş bir kurum veya kuruluşta oluşacak hasarın en aza indirgenmesi, oluşan hasarın onarılması ve yetkisiz erişimlerin kaldırılması için Adli Bilişim ve Olay Müdahalesi süreçleri işletilir. Adli bilişim ve olay müdahalesi süreçlerinde inceleme yapılan kurum alt yapısındaki tüm uç noktaların hızlı bir şekilde analiz edilmesi ve tüm tehdit unsurlarının tespit edilip ortadan kaldırılması kurum faaliyetlerinin devam etmesi açısından önem arz etmektedir.

Bu tez çalışmasında siber dünyada aktif olarak sıkça kullanılan saldırı türlerinin analizi yapılmaktadır. Aynı zamanda bu saldırı türleri simüle edilmektedir. Adli Bilişim ve Olay Müdahale süreçlerinde Windows ve Linux sistemlerin bulunduğu bir sanal ortam oluşturularak bu ortamda sık kullanılan siber saldırı yöntemlerinin uygulanması ve bu yöntemlere karşı savunma mekanizmalarının geliştirilmesi, olay örgüsünün çözülerek kök sebebin ortaya çıkarılmasına katkı sağlamak hedeflenmiştir. Bu çalışmanın motivasyonu tüm kurum ve kuruluşların karşı karşıya olduğu siber tehditlerle baş edebilmektir.

Tez çalışmasında yapılan uygulamada Windows sistemlerde sıkça karşılaşılan zararlı yazılım türevlerinin manuel tekniklerle ve açık kaynak çözümlerden olan YARA ile tespitinin nasıl yapıldığı açıklanmıştır.

Anahtar Kelimeler: YARA, Zararlı Yazılım, Adli Bilişim, Siber Saldırı

ABSTRACT

Simulations Cyber Security Attack Types and Detection with YARA

Mustafa Emre DEMİR

Master's Thesis

FIRAT UNIVERSITY
Graduate School of Natural and Applied Sciences

Department of Digital Forensic Engineering

July 2022, Pages: xi + 48

Information systems are developing rapidly. Companies move their business processes to information systems and all processes are carried out on these systems. However, it has become important to ensure the security of company data and personal data on the systems. At the same time, the security of the data was compromised by malicious individuals or groups targeting the data of the organizations' information systems. The targeting of organizations by these aggressive individuals or groups is increasing day by day. Therefore, organizations started to use various solutions for cyber security in order to protect their systems. However, these solutions are not enough for organizations that do not have a sustainable cybersecurity infrastructure and physical team, and these organizations cannot avoid being the target of attackers. In the first few sentences, the importance of the subject and the aim of the thesis should be defined and brief information about the method and findings should be presented. Finally, the information produced should be expressed briefly.

A company that has been targeted by attackers operates DFIR processes to minimize damage. In DFIR processes, all endpoints in the infrastructure of the institution should be analyzed quickly and all threat elements should be detected and eliminated. This institution is important for the continuation of its activities.

In this thesis, the analysis of attack types that are actively used in the cyber world is made. At the same time, these attack types are simulated. A virtual environment with Windows and Linux systems was created in Digital Forensic and Incident Response processes. In this environment, frequently used cyber-attack methods were applied and developed defense mechanisms against these methods. The motivation of this study is to cope with the cyber threats faced by all institutions and organizations.

In the application made in the thesis study, it is explained how the malware variants, which are frequently encountered in Windows systems, are detected with manual techniques and YARA, which is one of the open-source solutions.

Keywords: YARA, Malware, Digital Forensic, Cyber Attack

ŞEKİLLER LİSTESİ

Sayfa

TABLOLAR LİSTESİ

Sayfa

KISALTMALAR LİSTESİ

Kısaltmalar

DFIR	: Digital Forensic Incident Response
DDOS	: Distributed Denial of Service
IoT	: Internet of Things
TTP	: Tactic, Technic, Procedure
MiTM	: Man in the Middle
MiTB	: Man in the Browser
RAM	: Random Access Memory
UPX	: Ultimate Packet for Executables
XSS	: Cross Site Scripting
ICMP	: Internet Control Message Protocol
AD	: Active Directory
NIST	: National Institute of Standards and Technology
IT	: Information Technology
NT	: New Technology
NTFS	: New Technology File System
CERT	: Cyber Emergency Response Team
IP	: Internet Protocol
CEO	: Chef Executive Officer
CFO	: Chef Finance Officer
TCP	: Transmission Control Protocol
C2	: Command and Control

1. GİRİŞ

Siber saldırı, bilişim sistemleri üzerinde verilerini bulunduran kuruluşları hedef alan kişi veya gruplar tarafından birden fazla bilgisayar sistemi kullanarak bilgisayar sistemlerine veya ağ sistemlerine karşı başlatılan bir saldırıdır. Bu tür ataklar sistemlerin çalışmasına engel olabilir, sistemler içerisinde var olan özel veya kişisel nitelikli veriler çalmayı hedefleyebilir veya ele geçirilen bilgisayarı başka siber saldırılar için kullanabilir [1]. Siber suçlular, siber atak yapmak için birçok farklı yönteme başvurabilirler. Bu yöntemler, hizmet durdurma, fidye yazılımı saldırısı, çeşitli zafiyetlerin kullanılması ile zararlı yazılım yüklenmesi gibi çoğaltılabilir [1,2]. Siber saldırılar, bilişim sistemleri üzerinde verilen hizmetlerin devre dışı olmasına sebep olabilir veya veriler çalınabilir. Siber saldırganlar, istismar edilebilecek bir veya daha fazla güvenlik açığı olan bir kaynağı (fiziksel veya mantıksal) hedefler. Saldırının bir sonucu olarak, kaynağın gizliliği, bütünlüğü veya kullanılabilirliği tehlikeye girebilir. Bazı siber saldırılarda, bir kuruluşun Wi-Fi ağı, sosyal medya hesapları, işletim sistemleri, kredi kartı veya banka hesap numaraları gibi verilerin ele geçirilmesi hedef olabilir [3]. Bu tür saldırı yapan profesyonel gruplar mevcuttur. Bu gruplar Advanced Persistent Threat olarak isimlendirilir. Advanced Persistent Threat (APT), Türkçeye Gelişmiş Kalıcı Tehdit veya Gelişmiş Sürekli Tehdit olarak çevrilen terim, dünya genelinde profesyonel siber saldırı yapan gruplara verilen genel isimdir [4]. Gruplara özel bireysel isimleri mevcuttur. Örneğin Lazarus veya APT38 olarak isimlendirilen grubun Kuzey Kore ile ilişkili olduğu bilinmektedir [5]. Bu grup genellikle IT ve Kimya sektöründe çalışan firmaları hedef almaktadır. Günümüzde bu tür saldırı grupları yaygınlaşmıştır dolayısı ile bu tür saldırıların analizi için kullanılan yeni tekniklerin geliştirilme ihtiyaçları ortaya çıkmaktadır [6,7].

1.1. Tezin Amacı

Bu tezin amacı siber saldırıya uğramış bir kurumda saldırı yapılan tekniğin ve saldırganların sistemler üzerindeki kalıcılık mekanizmalarının açık kaynak bir teknik olan YARA modeli ile tespit edilmesini sağlamaktır. YARA, kendine özgü bir kural yazım formatı olan, özel bir şirket tarafından açık kaynak olarak geliştirilmiş ve genel kullanıma sunulmuş bir araçtır. Bu araç özelinde, genele hitap edecek kurallar geliştirilerek literatüre katkı sağlanması hedeflenmiştir.

Aynı zamanda bu tez çalışmasını okuyan bir siber güvenlik uzmanı kendi kurumunda yaşanması muhtemel siber ataklara karşı önceden önlem alabilecek veya hali hazırda bu saldırılara maruz kalmış ise hızlıca tespitini sağlayacak ve sonrasında neler yapması gerektiğini bu çalışmada öğrenecektir.

Buradaki analiz ve inceleme için Windows ve Linux işletim sistemleri tercih edilmiştir. Bu sistemlerin hem sunucu hem de son kullanıcı için üretilmiş olan dağıtımları üzerinde testler

yapılmıştır. Bu sistemler üzerinde en çok kullanılan saldırı türleri uygulanmış ve bilgisayarlar üzerinde bu saldırı teknikleri yardımıyla çeşitli zararlı yazılımlar yüklenmiştir. Bu sayede sistemlere uzaktan yetkisiz sürekli erişim için kalıcılık sağlanmıştır.

Yetkisiz erişim sağlanan ve ele geçirilen sistemlerde YARA kullanılarak zararlı yazılımların tespiti ve temizlenme işlemleri manuel tekniklerle gösterilmiştir. Burada kullanılan teknik olan YARA bir çeşit tespit aracıdır. Bu araç içerisinde kendi yapısına bağlı kalarak kurallar geliştirilebilmektedir. Bu çalışmada elde edilen çıktılar doğrultusunda kurallar geliştirilerek zararlı türlerinin açık kaynak çözümlerle tespiti sağlanmıştır.

Bu tezin 6 temel çıktısı olması hedeflenmiştir;

- Saldırgan grupların sık kullandıkları saldırı ve tekniklerin ortaya çıkarılması,
- Adli bilişim için manuel bir Linux aracının tanıtılması,
- Bu tekniklerle oluşturulan zararlıların YARA kuralları ile tespit edilmesi,
- Yazılan YARA kurallarının açık bir şekilde paylaşılarak literatüre katkı sağlanması,
- YARA Yazımını açıklayarak kuruluşların kendi bünyesinde bu kuralları geliştirip kullanılmasının sağlanması,
- Gelecek çalışmalarda otomatize bir YARA kuralı üretim aracı geliştirilmek istenirse bu çalışmadan faydalanılmasıdır.

1.2. Tezin Yapısı

Bu tez çalışması genel itibariyle 6 bölümden oluşmaktadır.

Birinci bölümde siber saldırı kavramının tanımı yapılmıştır. Aynı zamanda saldırganlara literatürde ne tür isimler veriliyor gibi soruların cevabı burada açıklanmıştır.

İkinci bölümde literatür taraması yapılarak sık kullanılan siber saldırı türleri araştırılmış ve Siber Saldırı Türleri başlığı altında özetlenmiştir. Burada değinilen saldırı türleri açıkça anlatılmıştır.

Üçüncü bölümde Adli Bilişim ve Olay Müdahalesi kavramı açıklanmıştır.

Dördüncü bölümde siber saldırı türlerinde kullanılan tekniklerin nasıl uygulandığı açıklanmıştır.

Beşinci bölümde tezin ana çalışma konusu olan YARA'nın ne olduğu nasıl kullanıldığı, kuralların nasıl yazılması gerektiği ve yazılan kuralların çıktıları açıklanmıştır.

Tez çalışmasının altıncı ve son bölümünde ise çalışmanın çıktılarından, sonuçlarından ve önerilerden bahsedilmiştir.

2. SİBER SALDIRI TÜRLERİ

Siber saldırı, bilgisayarları veya bilgisayar ağlarını hedef alan, bu sistemler içerisindeki kritik verileri ele geçirmeyi, değiştirmeyi veya sistemlere zarar vermeyi hedefleyen saldırı türleridir. Saldırgan kavramı, hedeflenen sistemde yetkisiz erişim sağlayabilen kişi veya buradaki süreçlerin bütünü olarak açıklanır.

2.1. Hizmet Durdurma Saldırısı (DDoS)

DDoS veya dağıtılmış hizmet reddi saldırısı, hedeflenen bir bilişim sisteminin, sunucu üzerinde verilen bir servisin veya bilişim sistemlerinin bulunduğu bir ağın normal trafiğini manipüle etmek amacıyla mevcut altyapı kapasitesinin üzerinde bir internet trafiği göndererek sistemin çalışmaz veya hizmet veremez hale gelmesini sağlayan saldırı türüdür [8]. DDoS atakları, hedef aldığı sistemde trafiği artırmak ve kapasiteyi doldurmak için daha önceden ele geçirilmiş sistemleri kullanır. İstismar edilen sistemler, sunucu sistemi, bilgisayar, telefon ve IoT cihazları gibi ağa bağlı olan birçok farklı tür kaynağı içerebilir [8,9]. Yüksek trafiklere çıkabilen bir DDoS saldırısı, bir araba yolunda trafiğin çok sıkışık olması ve normal düzenin dışına çıkması gibi değerlendirilebilir [9].

DDoS saldırısını tespit etmek için en yaygın yöntem, bir servisin, hizmetin ya da sistemin aniden yavaşlaması veya erişilemez hale gelmesidir. Normal seyrinde devam eden bir trafiğin böyle bir artışla karşılaşması durumunda detaylı analiz ve araştırma yapma ihtiyacı ortaya çıkar. Bu tür araştırmaları yapabilmek için ağ trafiğini dinleyecek ve analiz edebilecek profesyonel araçlara ihtiyaç vardır. Bu tür analiz araçlarında DDOS saldırısın tespit edebilmek için aşağıdaki bilgilere ihtiyaç vardır [9]:

- Bir IP aralığı kullanan veya tek bir IP'den gelen şüpheli trafiğin miktarını kontrol etmek gereklidir.
- Saldırı türüne göre DDOS saldırıları tespit edebilmek için çeşitli kalıntılar olabilir.
- Servis verilen sistem üzerinde tek bir sayfanın veya tek bir servisin çalışmaması veya bu alanlara yapılan isteklerde ani bir artış olması.
- Servise gelen isteklerin ani bir şekilde artması veya belirli zaman aralıklarıyla artışın devam etmesi durumu oluşabilir.
- Konum, cihaza türü veya web tarayıcısına ait teknik bilgilerin tespit edildiği primlenebilir bir trafik akışı oluşabilir.

2.1.1. DDoS Saldırısı Nasıl Çalışır?

DDoS saldırılarının gerçekleştirilebilmesi için bilgisayarların ağa bağlı olması gereklidir. Bahsedilen bu ağların içerisinde, zararlı yazılım ile ele geçirilmiş ve saldırgan tarafından yetkisiz erişim sağlanmış, uzaktan kontrol edilebilen cihazlardan ve diğer IoT cihazları gibi cihazlar bulunur. Bu bireysel olarak ele geçirilmiş ve bahsedilen ağa dâhil olmuş cihazlara bot makineler denir. Bu cihazlardan oluşan gruplara ise botnet ağı veya botnet grubu denir [9].

Saldırgan tarafından oluşturulan botnet ağları yönetilebilir şekilde tasarlanır. Saldırgan botnet ağındaki her bir cihaza uzaktan talimat gönderebilir. Planlanan bir saldırı için cihazları yönlendirebilir. Saldırgan hedef olarak seçtiği bir ağa, cihaza ya da servise, oluşturduğu botnet ağından hedef alınan IP adresine istekler gönderir ve potansiyel olarak ağın veya sunucu servisin aşırı yüklenmesine neden olur. Bu durumda normal trafiğin karşılanamamasına ve verilen hizmetin saldırı sebebiyle reddedilmesine neden olur. Botnet içerisindeki her bir bot meşru ve internete bağlı bir cihaz olması sebebiyle, normal bir trafiği saldırgan trafiğinden ayırt etmek bazı durumlarda zorlaşabilir [9].

2.2. Oltalama Saldırısı

Kimlik avı veya oltalama olarak isimlendirilen bu teknik, hedef alınan kişi veya topluluğa kendisini legal bir servis, şirket veya kişi olarak tanıtır ve güven sağlar. Bu sayede kurbanın kişisel bilgilerini, kredi kartı bilgileri veya şifreleri gibi hassas verilerini elde etmeyi hedefler. Bu tekniği genellikle e-posta, kısa mesaj veya telefon ile arama yoluyla mağdura iletir. Bu saldırı sonucunda elde edilen bilgiler daha sonra önemli hesaplara erişmek için kullanılır ve kimlik hırsızlığına veya mali kayıplara neden olabilir [10].

Kimlik avı tekniği ile işlenen ilk suça ilişkin dava 2004 yılında America Online isimli web sitesinin taklidini oluşturan Kaliforniyalı bir kişiye karşı açıldı. Tasarlanan sahte web sitesi ile kullanıcılara ait hassas bilgileri elde etmeyi ve hesaplarından para çekmek için kredi kartı bilgilerine erişmeyi hedeflemişti. Bu tekniklerin yanı sıra, saldırganların sürekli olarak ortaya çıkardığı 'vishing' (sesli phishing), 'smishing' (SMS Phishing) ve diğer birkaç phishing tekniği de vardır [11].

2.3. Zararlı Yazılım Saldırısı

Zararlı yazılım veya kötü amaçlı yazılım saldırıları, son kullanıcı bilgisi olmadan bir bilgisayara, sunucuya, istemciye veya bilgisayar ağına ve/veya altyapısına zarar vermek veya izlemek için tasarlanmış her türlü kötü amaçlı yazılımdır [12].

Siber saldırganlar birçok farklı nedenle kötü amaçlı yazılım oluşturur, kullanır ve satar, ancak en sık olarak kişisel, finansal veya ticari bilgileri çalmak için kullanılır. Motivasyonları değişse de, siber saldırganlar taktiklerini, tekniklerini ve prosedürlerini (TTP) neredeyse her zaman görevlerini

yerine getirmek için daha yetkili kimlik bilgilerine ve hesaplara erişim elde etmeye odaklanırlar [12].

2.3.1. Zararlı Yazılım Türleri

Çoğu kötü amaçlı yazılım türü aşağıdaki kategorilerden birinde sınıflandırılabilir [13];

En Sık Kullanılan Aktif Siber Atak Türleri

Virüs: Bir bilgisayar virüsü çalıştırıldığında, diğer programları değiştirerek ve kötü amaçlı kodu normal bir program işleyişine ekleyerek kendini çoğaltabilir. Diğer dosyalara bulaşabilen tek kötü amaçlı yazılım türüdür ve kaldırılması en zor kötü amaçlı yazılım türlerinden biridir [13].

Solucan: Bir solucan, son kullanıcı müdahalesi olmadan kendi kendini çoğaltma gücüne sahiptir. Bir makineden diğerine geçerek hızla tüm ağlara bulaşabilir. Yani virüslü sistemlerde etkin kalırken diğer bilgisayarlara da bulaşan bir tür kötü amaçlı yazılımdır [13].

Truva Atı: Truva atı kötü amaçlı yazılımı, kendisini meşru bir program olarak gizler ve bu da onu tespit edilmesi en zor kötü amaçlı yazılım türlerinden biri haline getirir. Genellikle diğer kötü amaçlı yazılım türlerinin sisteme girmesine izin vermek için kullanılır. Kullanıcıları gerçek amacı konusunda yanlış yönlendiren herhangi bir kötü amaçlı yazılım türüdür [13].

Kötü amaçlı kod: Zararlı olan herhangi bir program veya dosyadır [13].

Hibrit kötü amaçlı yazılım: Modern kötü amaçlı yazılım, genellikle bir "hibrit" veya kötü amaçlı yazılım türlerinin birleşimidir. Örneğin, "botlar" önce Truva atı olarak görünür, ardından yürütüldükten sonra solucan gibi davranır. Ağ çapında daha büyük bir siber saldırının parçası olarak bireysel kullanıcıları hedeflemek için sıklıkla kullanılırlar [14].

Reklam Zararlısı: Reklam yazılımı, son kullanıcıya istenmeyen ve saldırgan reklamlar (örn. pop-up reklamlar) sunar [14].

Kötü amaçlı reklamcılık: Kötü amaçlı reklamcılık, son kullanıcı makinelerine kötü amaçlı yazılım dağıtmak için yasal reklamları kullanır [14].

Casus yazılım: Casus yazılım, şüphelenmeyen son kullanıcıyı gözetler, kimlik bilgilerini ve parolaları toplar, tarama geçmişi alır. Hassas bilgileri ifşa etmek, internet kullanım verilerini

çalmak, bilgi işlem cihazınıza erişmek veya cihazlara fiziksel zarar vermek için tasarlanmış zararlı yazılımdır [15].

Fidye Yazılımı: Fidye yazılımı makinelere bulaşır, dosyaları şifreler ve kurbandan talep edilen değeri ödeyene kadar fidye için gerekli şifre çözme anahtarını saklar. Fidye ödenene kadar bir bilgisayar sistemine veya verilere erişimi engellemek için tasarlanmış kötü amaçlı yazılımdır. Fidye yazılımı, kimlik avı e-postaları, kötü amaçlı reklamlar, virüslü web sitelerini ziyaret ederek veya güvenlik açıklarından yararlanarak yayılır [15].

Kaba kuvvet saldırıları: Bir sisteme veya hassas verilere yetkisiz erişim elde etmek için kullanıcı adlarını ve şifreleri tahmin etmeyi içeren popüler bir saldırı yöntemidir [15].

Siteler arası komut dosyası çalıştırma (XSS): Genellikle web uygulamalarında bulunan bir tür güvenlik açığı olan XSS, saldırganların diğer kullanıcılar tarafından görüntülenen web sayfalarına istemci tarafına komut dosyaları eklemesine olanak tanır ve erişim denetimini atlatmak için kullanılır [16].

Hizmet Reddi saldırıları (DoS): Kötü niyetli bir siber tehdit aktörünün eylemleri nedeniyle meşru kullanıcıların bilgi sistemlerine, cihazlara veya diğer ağ kaynaklarına erişimin kesilmesine sebep olan saldırı türüdür [16].

Exploit: İstenmeyen davranışlara neden olmak veya hassas verilere yetkisiz erişim sağlamak için güvenlik açığından yararlanan bir yazılım, veri veya komut dizisidir [16].

E-posta Sahtekârlığı: Sahte gönderici adresi oluşturularak mağdurlara e-posta gönderilmesi yöntemi ile planlanan saldırı türüdür [17].

Kimlik Avı: Giriş kimlik bilgileri, kredi kartı numaraları, banka hesap numaraları veya diğer finansal bilgiler gibi hassas bilgileri meşru bir siteymiş gibi göstererek toplamayı hedefleyen saldırı türüdür [17].

Ortadaki Adam (MiTM): Bir saldırgan, doğrudan iletişim kuran iki taraf arasındaki iletişimin arasına girerek trafiği kendi üzerinden geçirir ve isterse bu trafiği değiştirir. Bu, saldırganın iletişimi yönlendirmesine, dinlemesine ve hatta her bir tarafın söylediklerini değiştirmesine olanak tanır [18].

Tarayıcıdaki adam (MiTB): Web sayfalarını ve işlem içeriğini değiştirmek veya gizli bir şekilde yeni içerik eklemek için tarayıcıdaki güvenlik açıklarından yararlanarak bir web tarayıcısına bulaşan bir truva atı için proxy gibi davranan saldırı tekniğidir [19].

Ping Taşması (ping flood): Saldırganın, kurbana ICMP protokolünü kullanarak sürekli ping paketleri ilettiği basit bir hizmet reddi (DOS) saldırısıdır [20].

Ölüm Ping'i (ping death): Bir bilgisayara hatalı biçimlendirilmiş veya başka bir şekilde kötü amaçlı ping göndermeyi içeren saldırı türüdür [20].

Smurf saldırısı: Smurf saldırısı, bir saldırganın, kurbanın sunucusunu sahte İnternet Protokolü (IP) ve İnternet Kontrol Mesaj Protokolü (ICMP) paketleri ile doldurduğu dağıtılmış bir hizmet reddi (DDoS) saldırısıdır. Sonuç olarak, hedef sistem çalışmaz hale getirilir. Bu saldırı türü, adını 1990'larda yaygın olarak kullanılan "DDoS Smurf" isimli zararlı yazılım aracından alır. Kötü amaçlı yazılım aracı tarafından oluşturulan küçük ICMP paketi, kurbanın sistemine önemli ölçüde zarar verebilir [21].

Arabellek taşmaları (buffer owerflow): Saldırganlar, bir uygulamanın bellekte kullandığı alanın üzerine veri yazarak bellekleri taşırabilir. Taşırılan kısımda programın işleyişi değiştirilerek zararlı yazılımın çalışması hedeflenir [22].

Yığın taşmaları (heap owerflow): Yığın veya bir uygulama için bellekte bir alan ayrıldığında ve veriler üzerinde herhangi bir sınır denetimi yapılmadan veriler bu belleğe yazıldığında meydana gelen bir arabellek taşması biçimidir. Taşırılan kısımda zararlının çalışması hedeflenir [22].

Yığın taşmaları (stack owerflow): Bir programın yığında bulunan bir arabelleğe, arabellek için ayrılandan daha fazla veri yazmasına neden olan ve programın çökmesine veya yanlış çalışmasına neden olan aynı zamanda yığın içindeki verilerin bozulması ile sonuçlanan bir arabellek taşması türüdür. Taşırılan kısımda zararlının çalışması hedeflenir [22].

Doğrudan erişim saldırıları: Bir bilgisayar korsanının veya saldırganın bir bilgisayara erişebildiği ve ondan doğrudan veri indirebildiği bir saldırı türüdür [23].

Yetki yükseltme: Genellikle uygulama veya kullanıcı tarafından kısıtlanan kaynaklara yetkisiz erişim elde etmek için bir işletim sistemi veya uygulamada var olan programlama hatası, güvenlik açığı, tasarım hatası, yapılandırma hatası veya erişim denetimini kullanarak yetki verilmeyen alana ulaşılmayı hedefleyen saldırı türüdür [23].

Balina saldırısı: Bir şirketten hassas bilgileri çalmak için CEO veya CFO gibi üst düzey yöneticileri hedefleyen bir tür kimlik avı saldırısıdır. Bu, finansal bilgileri veya çalışanların kişisel bilgilerini içerebilir [23].

Sosyal mühendislik: Sosyal mühendislik, kurbanları gizli bilgileri ve hassas verileri ifşa etmeye veya olağan güvenlik standartlarını ihlal eden bir eylem gerçekleştirmeye yönlendirmek için insan psikolojisini ve duyarlılığını kullanan bir saldırı türüdür [24].

En Sık Kullanılan Pasif Siber Atak Türleri

Bilgisayar gözetimi: Bilgisayar etkinliğinin ve bir sabit sürücüde depolanan verilerin izlenmesi olarak tanımlanmaktadır [25].

Ağ gözetimi: Bilgisayar ağları üzerinden aktarılan faaliyet ve verilerin izlenmesidir [25].

Telefon dinleme: Üçüncü bir şahıs tarafından, genellikle gizli yollarla telefon ve internet tabanlı konuşmaların izlenmesidir [25].

Bağlantı noktası taraması (port scan): Bir ağ ana bilgisayarında bulunan açık bağlantı noktalarını ve hizmetleri tanımlamak için kullanılan bir tekniğidir [25].

Idle tarama: Hangi hizmetlerin kullanılabilir olduğunu bulmak için bir bilgisayara sahte paketler göndermekten oluşan bir TCP bağlantı noktası tarama yöntemidir [25].

Tuş vuruşu izleme (keylogging): Klavyede basılan tuşları kaydetme eylemidir [25].

Veri kazıma: Bilişim sistemi üzerindeki verilerin son kullanıcı tarafından normal şartlarda okunamaz hale geldiğinde bu verileri farklı teknikler kullanarak okunması yöntemiyle pasif veri elde edilen yöntemdir [25].

Typosquatting: Bilgisayar alan adlarının veya adreslerinin benzerine yakın şekilde taklit edilmesiyle internet kullanıcılarını hedef aldığı bir siber saldırı biçimidir [25].

Gizlice dinleme: Başkalarının özel konuşmalarını veya iletişimlerini rızaları olmadan gizlice dinleme eylemidir [25].

Endüstriyel Siber Atak Türleri

Kontrol sistemleri: Fiziksel altyapı üzerindeki sistemler kontrol etmek için endüstriyel veya mekanik kontrolleri etkinleştiren sistemleri yöneten yazılım veya bilişim sistemlerine yapılan saldırı türleridir. Aşağıdaki örnek verilen sektörleri hedef alır [26,27];

- **Enerji:** Siber suçlular, şehirlere, bölgelere veya hanelere güç sağlayan elektrik şebekelerini veya doğal gaz hatlarını hedef alabilir [26].

- **Finans:** Finansal altyapı, bilgisayar sistemleri ve finansal sistemlerin birbirine bağlılığı nedeniyle genellikle siber suçların hedefidir [26].

- **Telekomünikasyon:** Hizmet Reddi (DoS) saldırıları genellikle İnternet üzerinden çalışan telekomünikasyonları hedef alır ve iletişim yeteneğini azaltır [26].

- **Ulaşım:** Ulaşım altyapısına yönelik başarılı siber saldırılar, ulaşımın zamanlamasını ve erişilebilirliğini etkileyen telekomünikasyon saldırılarına benzer bir etkiye sahiptir [27].

- **Su:** Su altyapısı genellikle bilgisayarlar tarafından kontrol edilir ve bu da onu siber suçlular için büyük bir hedef haline getirir. Ele geçirilmesi durumunda en tehlikeli olanlardan biridir. Kanalizasyon sistemleri de tehlikeye girebilir [27].

2.4. Sıfırıncı Gün Saldırısı

Sıfır gün veya sıfırıncı gün saldırısı, bilişim sistemi üzerinde çalışan bir uygulamaya veya işletim sistemine ait yamanın yayınlanmadığı veya uygulama geliştiricilerinin farkında olmadığı/incelemek için yeterli zamana sahip olmadığı bir bilgisayar yazılımında veya uygulamasında bilinmeyen bir güvenlik açığı tehdidini tanımlamak için kullanılan terimdir [28]. Güvenlik açığı önceden bilinmediği için, açıklar genellikle kullanıcıların bilgisi dışında gerçekleşir. Bir uygulamanın verimli ve güvenli olması için tasarlanırken sıfır gün hatası önemli bir bileşen olarak kabul edilir [28].

Sıfırıncı gün saldırılarının göze çarpan özellikleri şunlardır [29]:

- Sıfır gün saldırıları, genellikle, güvenlik açığının ilk bulunduğu ve istismar edildiği zaman ile uygulama geliştiricilerin, istismara karşı koymak için gerekli çözümü yayınladığı zaman

arasında gerçekleşir. Bu zaman çizelgesi genellikle güvenlik açığı penceresi olarak adlandırılır.

- Sıfır gün saldırıları, ilgili uygulamaların güvenlik açıklarından yararlanarak bir ağı kullanılamaz hale getirebilir.

- Her zaman virüs değildirler ve Truva atları veya solucanlar gibi diğer kötü amaçlı yazılım biçimlerine bürünebilirler.

- Kişisel bilgisayar kullanıcıları için, saldırının doğası gereği güvenilir bir varlık aracılığıyla yapıldığından sıfır gün saldırısını teşhis etmek son derece zordur.

- Kötü amaçlı yazılımlardan korunmak için kullanılan çözümlerin son sürümlerinin kullanılması, bir sıfır gün saldırısına karşı yalnızca minimum güvenlik sağlayabilmesine rağmen en etkili yöntemdir.

2.5. Ortadaki Adam Saldırısı

Ortadaki adam (MiTM) saldırısı, saldırganın birbirleriyle doğrudan iletişim kurduklarına inanan iki taraf arasında gizlice araya girip mesajları ilettiği bir tür siber saldırıdır. Saldırı, saldırganın araya girip tüm konuşmayı kontrol ettiği bir gizli dinleme türüdür [30]. MiTM siber saldırıları, saldırgana oturum açma kimlik bilgileri, hesap ayrıntıları veya kredi kartı numaraları gibi hassas kişisel bilgileri gerçek zamanlı olarak yakalama ve değiştirme yeteneği verdiği için çevrimiçi güvenlik açısından ciddi bir tehdit oluşturur [30].

MiTM saldırıları bazen ortadaki canavar, ortadaki makine, ortadaki maymun ve tarayıcıdaki adam saldırıları olarak da adlandırılır. Tarayıcıdaki adam, saldırganların tarayıcıya odaklandığı ve kurbanın cihazına kötü niyetli proxy kötü amaçlı yazılımları enjekte ettiği en yaygın MiTM saldırısı türüdür. Kötü amaçlı yazılım genellikle kimlik avı e-postaları yoluyla tanıtılır. Bu saldırıların arkasındaki temel amaç, bir kullanıcının bir bankacılık veya finansal web sitesine giden trafiğini engelleyerek finansal bilgileri çalmaktır [30].

MiTM saldırıları sırasında siber suçlular kendilerini veri işlemlerinin veya çevrimiçi iletişimin ortasına sokar. Saldırgan, kötü amaçlı yazılımın dağıtımı yoluyla, kullanıcının web tarayıcısına ve işlemler sırasında gönderip aldığı verilere kolay erişim sağlar. Bir ortak anahtar ve bir özel anahtar ile güvenli kimlik doğrulaması gerektiren çevrimiçi bankacılık ve e-ticaret siteleri, saldırganların oturum açma kimlik bilgilerini ve diğer gizli bilgileri ele geçirmesini sağladıklarından MiTM saldırılarının ana hedefleridir [31].

Tipik olarak bu saldırılar, veri yakalama ve şifre çözme olarak bilinen iki aşamalı bir süreç aracılığıyla gerçekleştirilir. Veri ele geçirme, bir saldırganın bir istemci ile bir sunucu arasındaki veri aktarımını engellemesini gerektirir. Saldırgan, istemciyi ve sunucuyu birbirleriyle bilgi alışverişinde bulunduklarına inandırırken, verileri keser, gerçek siteyle bağlantı kurar ve iletişime yanlış bilgileri okumak ve eklemek için bir proxy görevi görür [31].

3. ADLİ BİLİŞİM VE OLAY MÜDAHALESİ

Adli bilişim, dijital ortamlardan elde edilen delillerin, bir konuyu ortaya çıkarmak amacıyla kabul edilebilir bir şekilde farklı ortamlara taşınması ve daha sonra elde edilen delillerin değerlendirme, inceleme ve raporlama süreçlerini kapsayan ve Adli Bilimlerin altında yer alan bir bilim dalıdır [32,33].

Olay müdahalesi ise bir kurum veya kuruluşun gelişmiş kalıcı tehditlere (APT) maruz kalması sonucunda, tüm sistemlerde var olan tehdit unsurlarının tespit edilmesi, olay ile ilgili kök nedenin ortaya çıkarılması, kalıcılık sağlayan tehdit unsurlarının tekrar oluşmamasını sağlayacak şekilde temizlenmesini ve sistemlerin yeniden güvenli hale getirilmesini sağlayan süreçler bütünüdür [34].

Olay müdahalesi süreçlerinde incelemeler yapılırken Adli Bilişim tekniklerine başvurulur. Olay müdahalesinde temel amaç tehdit unsurlarının en hızlı şekilde ortadan kaldırılıp sistemlerin yeniden güvenliğini sağlamaktır. Bu sebeple adli bilişim tekniklerinde kullanılan delil elde etme süreçleri olay müdahalesinde kullanılmamaktadır. Delil elde etme süreçlerinde başvurulan yöntemler işlemlerin yavaş yapılmasına ve tehdit unsurlarının kalıcılığının artmasına sebep olmaktadır. Dolayısıyla yapılması gereken en öncelikli işlem tüm sistem envanterinin çıkartılması ve sistemlerin hepsinin tekelden görünür hale getirilmesini sağlamaktır. Görünürlük sağlandıktan sonra sistemler üzerinde bulunan kritik logların kural setlerinden geçirilerek aktif yetkisiz işlemlerin tespiti amaçlanmaktadır [34].

Adli bilişim ve olay müdahalesi (DFIR), siber güvenlik olaylarını tanımlamaya, düzeltmeye ve araştırmaya odaklanan özel bir alandır. DFIR, olayların tam ve ayrıntılı bir resmini çizmek için adli kanıtları toplamayı, korumayı, analiz etmeyi ve raporlandırmayı kapsar. Aynı zamanda DFIR, genellikle bir saldırıyı kontrol altına almayı, durdurmayı, önlemeyi ve hasarları minimuma indirerek sistemleri eski haline çevirmeyi amaçlar [35].

Olay müdahalesi süreçlerinin işletilebilmesi için kurum ve kuruluşlarda bulunan tüm sistemlerin, işletim sistemi ve platform bağımsız olarak gerekli log kaynaklarında istenilen logların merkezi yönetim birimine aktarılması gereklidir [36].

Bütün bu bilgiler sonucunda adli bilişim ve olay müdahalesi, saldırgan kişi veya gruplar tarafından ele geçirilmiş sistemler üzerindeki zararlıların tespitini, bu zararlıların sisteme yerleşmesine imkân sağlayan güvenlik açıklarının belirlenmesini ve kapatılması için gerekli yönergeyi sağlar. Kurum operasyonlarını hedef alan suçlulara karşı savunma yapabilmek veya siber sigortalara kanıt sağlamak için ihtiyaç duyulan verileri sağlar [37].

Tek bir saldırının bile ne kadar zarar verici olabileceği düşünüldüğünde, bir siber güvenlik olayına nasıl müdahale edileceğini, tüm yönleriyle nasıl analiz ve raporlama yapılacağını bilmek çok önemlidir [37].

Adli bilişim ve olay müdahale süreçlerini açıklamak gerekirse,
DFIR, aktif bir siber güvenlik olayını durdurmaya çalışan çok disiplinli bir dizi görev ve süreçtir. Olay müdahale planlaması, BT mimarisi dokümantasyonu, playbook'ların geliştirilmesi gibi süreçleri içerir [37].

3.1. Adli Bilişim

Adli bilişim, dijital teknolojiyi kapsayan adli bilimler altında yer alan bir bilim dalıdır. Siber güvenlik analistleri, dijital ortamlarda bulunan verilerin elde edilmesine, araştırılmasına ve incelenmesine odaklanır. Adli bilişimin nihai amacı, bir saldırının arkasındaki suçluların cezai suçlamalarla karşı karşıya kalması durumunda siber suçların kovuşturulmasına yardımcı olacak kanıtları toplamak ve korumaktır [38].
Bir kuruluşun adli bilişimle ilgilenmesinin genellikle dört ana nedeni vardır:

- Bir siber saldırının gerçekleşip gerçekleşmediğini doğrulamak için,
- Bir siber olayın tam etkisinin tespiti için,
- Siber saldırının nedenini araştırmak için,
- Siber saldırının gerçekleştiğini kanıtlamak için,

Herhangi bir adli soruşturmada olduğu gibi, özellikle bir saldırı veya uzlaşma devam ediyorsa hız çok önemlidir. Hızlı hareket etmek, aktif bir siber olayı durdurmaya yardımcı olabilir. Aktif bir bilgisayar, ağ veya cihaz, boşta dururken bile sürekli olarak bir araştırma için çok önemli olabilecek verileri üretir. Adli vakalarda olaya hızlı müdahale edilmezse bu verilerin silinmesi, üzerine yazılması veya başka bir şekilde değiştirilmesi riski ortaya çıkar [38].

Adli bilişim, dijital delilleri toplamak ve korumak bir gözetim zinciri doğrultusunda kabul edilen yöntemdir. Üç temel adımdan oluşur [38]:

Elde Etme: Bu adımda araştırmacılar, genellikle sabit diskin bir adli kopyasının alınmasıyla veya özel adli bilişim inceleme yazılım araçları kullanarak söz konusu ortamın tam bir kopyasını oluşturur. Orijinal ortam üzerinde, herhangi bir veri kaybını önlemek için çalışma yapılmaz.
Analiz: Adli bilişim uzmanları elde etme aşaması sonrasında çoğaltılan verileri veya teknolojiyi analiz edip bir hipotezi destekleyen veya onunla çelişen tüm kanıtları kaydeder. Bir olaydaki verileri ve eylemleri yeniden yapılandırmak için devam eden analizler yapılır ve ne olduğu, bilgisayar korsanlarının sistemleri nasıl ele geçirdiği hakkında sonuçlara ulaşmalarına yardımcı olur.

Raporlama: Bir adli bilişim araştırması tamamlandığında, analistlerin ortaya çıkardığı bulgular ve sonuçlar, teknik olmayan personelin anlayabileceği bir raporda sunulur. Bu raporlar, soruşturmayı yürütenlere iletilir.

NIST'e göre, gözetim zinciri; "Delilleri inceleyen her analist, tarihi/saati belgeleyerek elde etme, koruma ve analiz yaşam döngüsü boyunca kanıtın hareketini izleyen bir süreçtir."

Bu doğrultuda tez çalışması kapsamında linux sistemler üzerinde manuel tekniklerle adli bilişim çalışması gerçekleştirilmiştir. Çalışmanın çıktıları "3.2. Linux Sistemlerde Adli Bilişim Örneği" Başlığında açıklanmıştır.

3.2. Linux Sistemlerde Adli Bilişim Örneği

Linux sistemlerde manuel yöntemlerle adli bilişim analizi yapılması amacıyla "bash programlama" dilinde aşağıda açıklaması yapılan araç geliştirilmiştir. Geliştirilen aracın nasıl çalıştığı ve hangi verileri topladığı aşağıda açıklanmıştır.

Bu araç "bash programlama" dilinde yazılmıştır. Yöntemin temel amacı Linux işletim sistemi üzerinde yerleşik olarak geliştirilmiş araçların kullanarak sistem verilerini toplamaktır. Toplanan verileri makine ismini de ekleyerek bir "zip" dosyasına kaydetmektedir. Uygulamayı çalıştırmak için yönetici haklarına sahip olmak gereklidir. Uygulamanın temelini standart linux komutları oluşturmaktadır.

Bu çalışmanın katkıları aşağıdaki gibidir.

- Adli bilişim yazılımları hem yüksek maliyete sahiptirler hem de bazı delilleri paket adli bilişim yazılımlarıyla elde etmek mümkün değildir çünkü her adli bilişim yazılımının kendine has özelliği vardır. Bu çalışmada önerilen yöntemle amaca yönelik adli bilişim analizi yapabilecek script yazılmıştır.
- Adli analiz yazılımları genellikle yavaş çalışırlar (karmaşıklıkları yüksek). Bu çalışmada sunulan metotla düşük karmaşıklıkta Linux sistemler için adli analiz yapılabilecektir.
- Bu çalışmada operasyonel olarak kullanılabilecek bir analiz aracı detaylı bir şekilde açıklanmıştır.

Uygulama temel olarak sistem komutlarını sırasıyla çalıştırıp çıktıları bir konuma yazmaktadır. Buradaki çıktılar adli bilişim açısından önem arz eden verileri içermektedir. Bu çıktıları okuyan analistler yaşanan siber güvenlik vakasına çözüm üretirler. Uygulamada çalışan fonksiyonlara bir örnek Tablo 3.1'de verilmiştir.

Tablo 3.1 Çalışan Servislerin Tespiti İçin Gerekli Fonksiyon

```
sozdekod servis tespiti()
{
if chkconfig -list &> //$Location/IRCase 'chkconfig.txt'
else
chkconfig -l &> //$Location/IRCase 'chkconfig.txt'
cp -RH //etc//cron.allow /$Location/IRCase 'cronallow.txt'
}
```

Geliştirilen araç bu alanlardaki verilerin okunmasını sağlamakta ve bir metin çıktısı oluşturmaktadır. Aynı zamanda bu dosyaların özet fonksiyon değerlerinin de hesaplayarak veri bütünlüğünü de sağlamaktadır. Bu işlemi tek seferde ve hızlıca yapması en temel kazanımıdır. Normal şartlarda bu verilerin elde edilebilmesi için ya sisteme uzak bağlantı sağlanarak tek tek ilgili alanlardan verilerin okunması gerekir ya da sistemin imajının alınması ve Bölüm 3 ve Bölüm3.1'de bahsedilen tekniklerin kullanılması gerekmektedir. Bizim yöntemimiz sayesinde buradaki lisans maliyeti ve zamandan kazanım sağlanmaktadır. Tablo 3.2'de araca ait örnek çıktılar gösterilmiştir.

Tablo 3.2 Özet Fonksiyonlarının Hesaplanması

```
1 d4fb107b25fe3f4d637df1b56445960332ccc56fe027ec6a3f849b35d663c3dd  /usr/bin/znew
2 04484a4171fb583cf0f6645de2efcfc66a53ffa31d6b802533a6e00355c75ee2  /usr/bin/dpkg
3 eac9ecba762c54dd50150d743414c4800a22aa8a4951794849b5f451d08f1937  /usr/bin/xsm
4 b52a1b162880dba8af6eef90f9a80f9a6d10b8af41750e28ce78bb0b76050948  /usr/bin/gdb
5 a934cdb1643a924d96a0edbe85b8bb058fad9e51a7b7b66bcc74b0520227f39a  /usr/bin/pkg-config
6 31ada0d1b8f1b182910d90d6f560f30166c1fefb053f97b0b28da96236f78167  /usr/bin/tzselect
7 c69f113aeafff76af527f11533ba10a7f0cdcaaab0ce47b696a06d51b60234d4  /usr/bin/ptx
8 cbdfc9f8b4b58d645917138197bdf8a5e36541b6778aa0e395741d9e97e5220d  /usr/bin/foo2zjs-pstops
9 e3ed27ec700fe277a48d137cdd39913c7141f4dde9074733f926eb2c8b4f2350  /usr/bin/who
```

Yapılan çalışmalar sonucunda önerilen çözümün adli bilişim ve olay müdahalesi süreçlerinde katkı sağladığı gözlemlenmiştir. Kurumların bu çözüm sayesinde olay müdahale süreçlerini daha efektif yönetebileceği değerlendirilmektedir. Ayrıca bu çözüm yöntemi sektörde bu alanda çalışan araştırmacılar tarafından da kullanıldığı ve başarılı sonuçlar elde edildiği gözlemlenmiştir.

3.3. Olay Müdahalesi Nedir?

Olay müdahalesi (IR), bir işletmenin siber güvenlik olayının ortasındayken gerçekleştirdiği bir dizi faaliyettir. IR'nin amaçları doğrultusunda, bir siber olay bilgi güvenliğinin temel ilkeleri olan bilgi gizliliğini, bütünlüğünü ve/veya kullanılabilirliğini tehlikeye atan herhangi bir olay olarak tanımlanabilir [39].

IR faaliyetleri genellikle, bir olayın genel hasarını azaltırken IT altyapısını mümkün olduğunca hızlı bir şekilde yeniden kurmak ve çalıştırmak için tasarlanmış bir IR planı tarafından takip

edilmesi gerekir. Bu çerçeveler olaydan meydana gelebilecek hasarı azaltma çabalarını desteklemek için tasarlanmıştır, ancak daha geniş anlamda kuruluşların siber olgunluk ve yeterlilik oluşturmasına da yardımcı olur. Bu, savunmaların geliştirilmesine, saldırıların ve olayların ilk etapta işletmeleri etkilemesine yardımcı olabilir [39,40].

Bir siber güvenlik saldırısı tarafından hedef alınan işletmeler için sistemlerin yeniden kullanılabilir hale gelmesi en önemli endişedir, ancak tekrar çalışmaya başlamanın ötesinde, bir olayın nasıl ve neden olduğunu anlamak da önemlidir [40].

DFIR uzmanları, sistemlere kimin saldırdığını, nasıl içeri girdiğini, saldırganların sistemleri ele geçirmek için attıkları adımları ve bu güvenlik açıklarını kapatmak için neler yapabileceklerini belirlemeye yönelik bilgileri toplar, denetler ve analiz eder. Bu bilgiler ayrıca, tanımlanan saldırganlara karşı yasal bir dava açılmasına yardımcı olmak için sıklıkla kullanılır. Bilgiler, araştırmacıların dijital kanıtları ortaya çıkarmasına ve korumasına yardımcı olan adli bilişim süreçleri kullanılarak toplanır [41].

3.4. Manuel Bir Olay Müdahalesi Örnek Çalışması

Bu tez çalışması kapsamında kurulan laboratuvar ortamında adli bilişim ve olay müdahalesi (DFIR) süreçlerini detaylı açıklayabilmek için bir batch script geliştirilmiştir. Bu script sayesinde sektörde var olan ve ücretsiz kullanıma sunulan 3 farklı araç kullanılarak otomatik bir analiz süreci işletilmesi hedeflenmiştir. Bu araçlardan ve analiz ettikleri sistem veri tabanlarından kısaca bahsetmek gerekirse;

Rawcopy: Windows sistemler üzerinde aktif olarak çalışan sistem dosyalarını sisteme zarar vermeden ve son kullanıcı tarafından müdahale edilemeyen dosyaların kopyalanmasını sağlar. Rawcopy aracı sayesinde aşağıda açıklanan sistem dosyaları geliştirilen script ile otomatize bir şekilde elde edilmiştir.

- **MFT:** Bir NT Dosya Sistemi (NTFS) birimindeki her dosya ve dizin hakkındaki bilgilerin tutulduğu bir veritabanıdır. Bu veritabanı adli bilişim ve olay müdahalesi (DFIR) süreçlerine önemli seviyede katkı sağlamaktadır.
- **Amcache:** Windows işletim sistemlerinde çalışan programların tüm detaylarının tutulduğu veritabanıdır. Buradan elde edilecek bilgiler DFIR süreçlerine çok önemli katkı sağlamaktadır.

MFTCmd: Açıklanan MFT veritabanını son kullanıcıların inceleyebileceği şekilde anlamlı hale çeviren açık kaynak kullanıma sunulan araçtır.

Amcacheparser: Açıklanan Amcache veritabanının son kullanıcıların inceleyebileceği şekilde anlamlı hale çeviren açık kaynak kullanıma sunulan araçtır.

Reg: Registry aracının windows tarafından geliştirilen komut satırı aracıdır. Bu araç sayesinde DFIR için önemli veriler elde edilir.

Burada açıklanan bilgiler doğrultusunda geliştirilen Manuel DFIR analiz script'i Şekil 3.1'de gösterilmiştir.

```
::batch script detaylari
@ECHO OFF
TIMEOUT /T 30
mkdir C:\YLcalisma\mft
START /WAIT C:\temp\get-artifact\RawCopy64.exe C:0 C:\YLcalisma\mft
mkdir C:\YLcalisma\Amcache
START /WAIT C:\temp\get-artifact\RawCopy64.exe
C:\Windows\appcompat\Programs\Amcache.hve C:\YLcalisma\Amcache
START /WAIT C:\temp\get-artifact\RawCopy64.exe
C:\Windows\appcompat\Programs\Amcache.hve.LOG1 C:\YLcalisma\Amcache
START /WAIT C:\temp\get-artifact\RawCopy64.exe
C:\Windows\appcompat\Programs\Amcache.hve.LOG2 C:\YLcalisma\Amcache
START /WAIT C:\temp\get-artifact\AmcacheParser.exe -f
"C:\YLcalisma\Amcache\Amcache.hve" --csv C:\YLcalisma\Amcache
TIMEOUT /T 30
START /WAIT C:\temp\get-artifact\MFTECmd.exe -f
"C:\YLcalisma\mft\$MFT" --csv "C:\YLcalisma\mft" --csvf mft.csv

systeminfo > C:\YLcalisma\sysinfo.thy
netstat -ano > C:\YLcalisma\connections.txt
reg query
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run >
C:\YLcalisma\runkey1.txt
reg query
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run >
C:\YLcalisma\runkey2.txt
reg query
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce
> C:\YLcalisma\runkey3.txt
sc queryex type=service state=all > C:\YLcalisma\allservice.txt
powershell.exe Get-Service | Where-Object {$_.Status -eq "Running"}
> C:\YLcalisma\allservicepw.txt
type
%userprofile%\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadlin
e\ConsoleHost_history.txt > C:\YLcalisma\ch-hist.txt
```

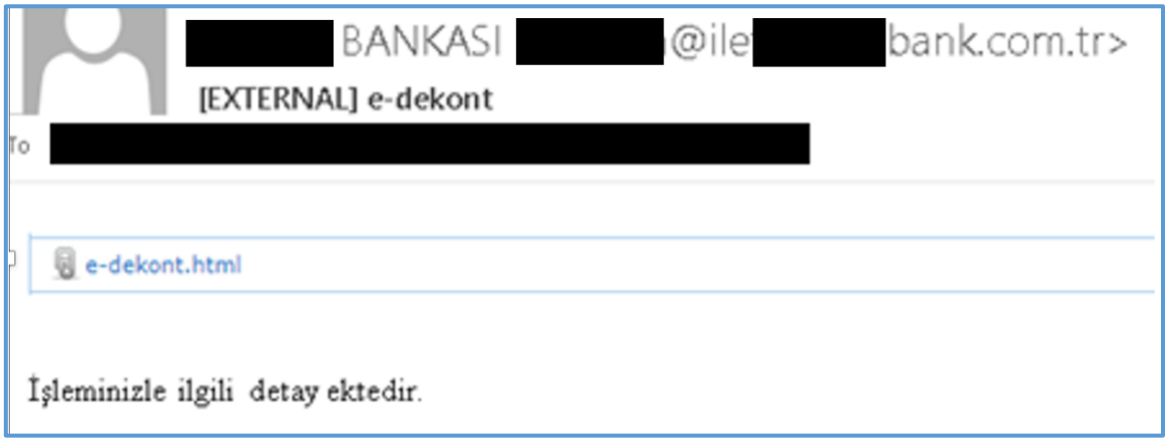
Şekil 3.1 DFIR Analiz Scripti

4. SALDIRI TÜRLERİNİN UYGULANMASI

Saldırı türlerini simüle etmek amacıyla Windows ve Linux sistemlerin bulunduğu bir sanal ortam oluşturulmuştur. Bu ortam üzerinde ikinci bölümde bahsedilen ataklara örnek olması amacıyla bu saldırı türleri uygulanmıştır. Çıktıları bu başlık altında sunulmuştur.

4.1. Oltalama Saldırısı Örneği

Kuruluşların En sık karşılaştığı siber saldırı tekniği olan oltalama saldırılarını simüle etmek amacıyla ve kullanıcı tarayıcısı üzerindeki kayıtlı parolaları çalmayı hedefleyerek bu çalışma yapılmıştır. Bu doğrultuda legal bir kuruluşun e-Posta adresi taklit edilerek sanal ortamda bulunan kullanıcı cihazına e-Posta gönderilmiştir. Senaryo gereği sanal ortamda oluşturulan kullanıcı gelen e-Postaya tıkladığında "discord" kanalından zararlı yazılım kullanıcı cihazına inmektedir. Burada zararlı yazılım açık bir ortam olan "discord" kanalına yüklenerek oradan indirilmesi hedeflenmiştir. Kullanıcı indirilen zararlıyı çalıştırdığında, kullanıcının tarayıcısında kayıtlı parola ve e-Posta kutusunda kayıtlı parolası web kanalı üzerinden saldırganların daha önceden temin ettiği sisteme gönderilmektedir. Burada bir ip adresi üzerinden dosya transfer protokolü kullanılarak buradaki verilerin aktarılması hedeflenmiştir. Gönderilen e-Posta maskelenerek Şekil 4.1'de sunulmuştur.

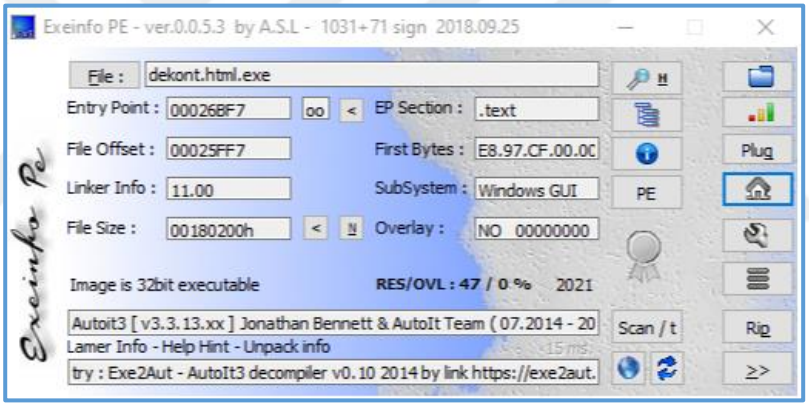


Şekil 4.1 Oltalama Amaçlı Gönderilen e-Posta

Şekil 4.1'de gösterilen e-Posta ekinde yer alan zararlı dosya kullanıcı tarafından açıldığında zararlı yazılım kullanıcının bilgisayarına kaydedilecektir ve otomatik olarak çalışacaktır. Burada kullanılan zararlı yazılım statik analizi "4.1. Zararlı Yazılım Saldırısı Örneği" başlığı altında açıklanmıştır. Bu senaryonun oluşturulma amacı oltalama saldırısı etkilerini göstermektir. Burada kalıcılık sağlayan zararlı yazılımın önce analizi açıklanacak sonrada tespit için YARA kuralı oluşturulacaktır.

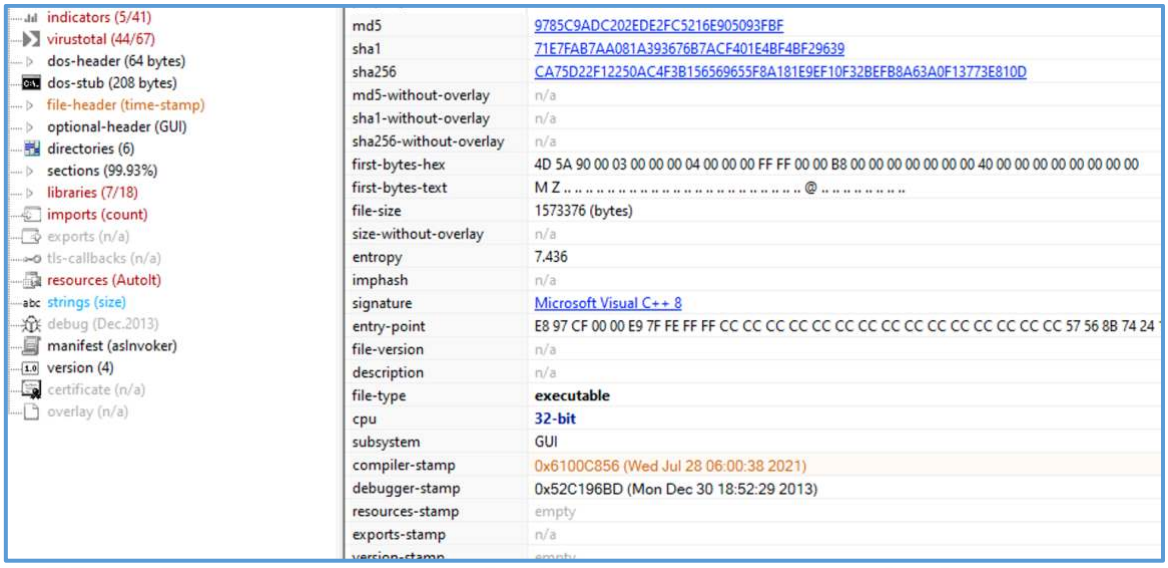
4.2. Zararlı Yazılım Saldırısı Örneği

Zararlı yazılım 32 bit olarak derlenmiş ve windows ara yüzüne sahiptir. Uygulama içerisinde "autoit" kodları vardır. AutoIt, Windows ara yüzü ve genel betikler için geliştirilmiş otomasyon amaçlı kullanılan bir "scripting" dilidir. Simülasyonda kullanılan uygulamanın paketlenmemiş olduğunu görüyoruz. Bu demektir ki zararlı yazılımın analiz edilmesini zorlaştırmak için ve decompile işleminin zorlaştırılması için kullanılan tekniklerin bu zararlıda kullanılmadığı anlamına gelmektedir. Böyle bir zararlı yazılımın incelenmesi ve analiz edilmesi daha kolaydır. Zararlı yazılımın sınıfı "dropper" olarak seçilmiştir. Dropper, bir kötü amaçlı yazılımı (virüs, arka kapı vb.) hedef sisteme "yüklemek" için tasarlanmış bir tür truva atıdır. Kötü amaçlı yazılım kodu, virüs tarayıcıları tarafından algılanmayacak şekilde tek aşamalı olarak tasarlanır, etkinleştirildikten sonra kötü amaçlı yazılımı hedef makineye indirebilir. Bu bilgiler doğrultusunda Bölüm 4.1'de uygulanan saldırıda kullanılan zararlı yazılıma ait detaylı bilgiler Şekil 4.2'de gösterilmiştir.



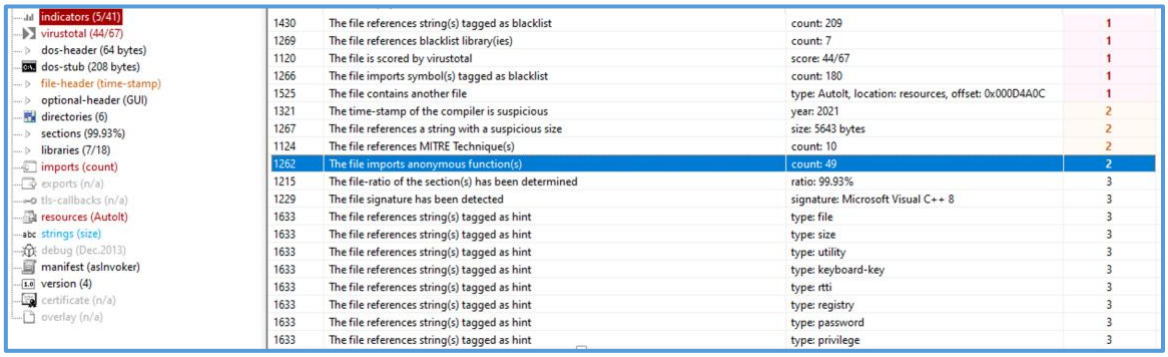
Şekil 4.2 Yürütülebilir Dosyaya Ait Detay Bilgilerin Öğrenilmesi

Uygulama C++ dilinde derlenmiş ve derlenme zamanı 28 Temmuz 2021 olarak görülmektedir. Derlenme zamanı kod geliştiricisi tarafından kolayca değiştirilebileceği için çok fazla önem arz etmemektedir. Pestudio aracı kullanılarak zararlı yazılıma ait çok fazla bilgiye ulaşabiliriz. Pestudio aracına ait çıktılar Şekil 4.3'te gösterilmiştir.



Şekil 4.3 Pestudio Aracına Ait Ekran Görüntüsü

Uygulamanın zararlı olduğunu işaret edebilecek 41 göstergeden 5 tanesi bu uygulamada eşleşmiştir. Bunlar arasında virustotal sonuçları, uygulamanın kullandığı tespit edilen şüpheli kütüphaneler, "mitre framework" olarak isimlendirilen zararlı yazılımların sınıflandırması için kullanılan bir platformda yer alan taktiklerle eşleşmelerin olduğu görülmektedir. Bu bilgilere ait detaylar Şekil 4.4'te gösterilmiştir.



Şekil 4.4 Zararlı Yazılım İnceleme Detayları

Uygulamanın kullandığı şüpheli kütüphaneler görülmektedir. "IsDebuggerPresent" gibi fonksiyonların çağrılmasıyla zararlı yazılım debug edildiğini anlayıp kendisini sonlandırabilir. Bu tür yöntemlere anti-debugging yöntemleri denir. Bu bilgilere ait detaylar Şekil 4.5'te gösterilmiştir.

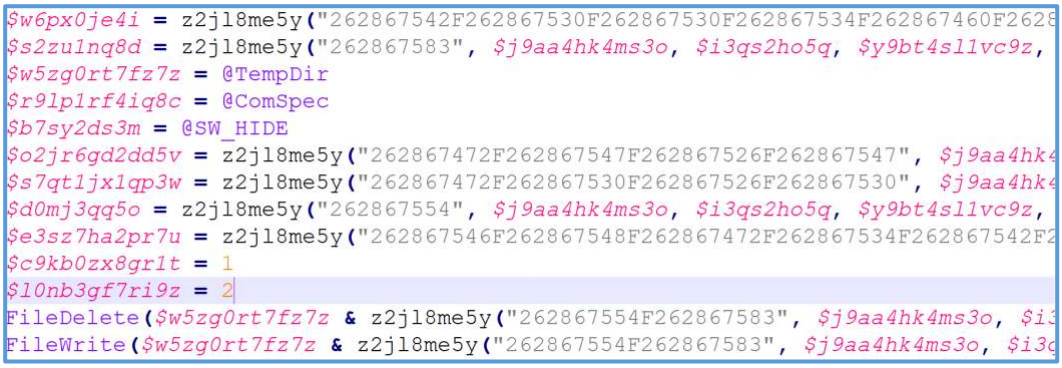
name (516)	group (17)	MITRE-Technique (8)	type (1)	anonymous (49)	blacklist (180)	anti-debug (0)	undocumented (0)	deprecated (21)	library (18)
Sleep	execution	T1497	implicit	-	-	-	-	-	kernel32.dll
GetSystemTimeAsFileTime	file	T1124	implicit	-	-	-	-	-	kernel32.dll
RegDeleteValueW	registry	T1112	implicit	-	×	-	-	-	advapi32.dll
RegDeleteKeyW	registry	T1112	implicit	-	×	-	-	-	advapi32.dll
RegSetValueExW	registry	T1112	implicit	-	×	-	-	-	advapi32.dll
CreateProcessW	execution	T1106	implicit	-	×	-	-	-	kernel32.dll
CreateProcessAsUserW	execution	T1106	implicit	-	×	-	-	-	advapi32.dll
CreateProcessWithLogonW	execution	T1106	implicit	-	×	-	-	-	advapi32.dll
ShellExecuteExW	execution	T1106	implicit	-	×	-	-	-	shell32.dll
ShellExecuteW	execution	T1106	implicit	-	×	-	-	-	shell32.dll
LoadLibraryA	dynamic-link-library	T1106	implicit	-	-	-	-	-	kernel32.dll
LoadLibraryExW	dynamic-link-library	T1106	implicit	-	-	-	-	-	kernel32.dll
LoadLibraryW	dynamic-link-library	T1106	implicit	-	-	-	-	-	kernel32.dll
GetComputerNameW	system-information	T1082	implicit	-	-	-	-	-	kernel32.dll
IsDebuggerPresent	system-information	T1082	implicit	-	-	-	-	-	kernel32.dll
CreateToolhelp32Snapshot	execution	T1057	implicit	-	×	-	-	-	kernel32.dll
Process32FirstW	execution	T1057	implicit	-	×	-	-	-	kernel32.dll
Process32NextW	execution	T1057	implicit	-	×	-	-	-	kernel32.dll
WriteProcessMemory	memory	T1055	implicit	-	×	-	-	-	kernel32.dll
RegEnumValueW	registry	T1012	implicit	-	-	-	-	-	advapi32.dll
RegEnumKeyExW	registry	T1012	implicit	-	-	-	-	-	advapi32.dll
SetWindowPos	windowing	-	implicit	-	-	-	-	-	user32.dll
SetWindowLongW	windowing	-	implicit	-	×	-	-	-	user32.dll
GetClassLongW	windowing	-	implicit	-	×	-	-	-	user32.dll
RedrawWindow	windowing	-	implicit	-	-	-	-	-	user32.dll
GetWindowTextLengthW	windowing	-	implicit	-	-	-	-	-	user32.dll
CallWindowProcW	windowing	-	implicit	-	-	-	-	-	user32.dll
FindWindowExW	windowing	-	implicit	-	×	-	-	-	user32.dll
EnumThreadWindows	windowing	-	implicit	-	×	-	-	-	user32.dll
SetForegroundWindow	windowing	-	implicit	-	×	-	-	-	user32.dll
DestroyWindow	windowing	-	implicit	-	-	-	-	-	user32.dll
EnumWindows	windowing	-	implicit	-	×	-	-	-	user32.dll
GetDesktopWindow	windowing	-	implicit	-	×	-	-	-	user32.dll

Şekil 4.5 Fonksiyonların Mitre Framework Eşleştirmesi

Uygulama içerisinde geçen string ifadeler de zararlı yazılım hakkında bilgi verir. Çalışmamınızın temelini oluşturan YARA kuralı oluşturulmasındaki en önemli faktör zararlı yazılım içinde bulunan ve o zararlıya ait benzersiz string değerlerinin elde edilmesidir.

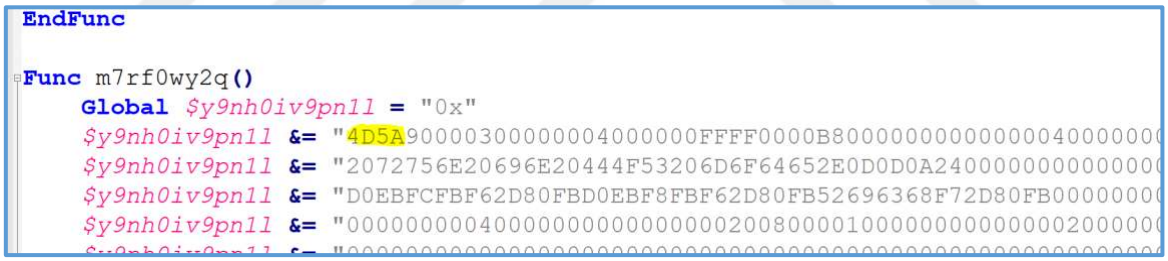
Burada kullandığımız zararlı yazılımın sertifikası bulunmamaktadır. Uygulama içerisinde geçen string ifadelerden uygulamanın bazı proseslerini çalıştırdığı, registry değerlerinde değişiklik yaptığı, sistem hakkında bilgi toplamaya çalıştığı görülmektedir.

AutoItExtractor uygulamasını kullanarak zararlı içerisindeki kod bloğu görüntülenebilmektedir. Burada ilgili uygulama kullanılarak zararlının kodları görüntülenmiştir. Zararlı yazılımın analizi zorlaştırılmak için kod karmaşıklaştırma tekniği kullanılmıştır. Bu teknik sayesinde değişken isimleri ve değerler rastgele ifadelerle değiştirilir ve inceleme yapan analistin zararlı yazılımın temel amacı öğrenmesi zorlaşır. Burada kodlar karmaşıklaştırıldığı durumlarda zararlı yazılım analizi zaman alabilmektedir. Bu sebeple yapılan inceleme sonucunda; dosya oluşturma, silme ve dosyaya yazma işlemlerinin yapıldığı ancak anlaşılmaktadır. Bu bilgilerin elde edildiği ekran görüntüsü Şekil 4.6'da sunulmuştur.



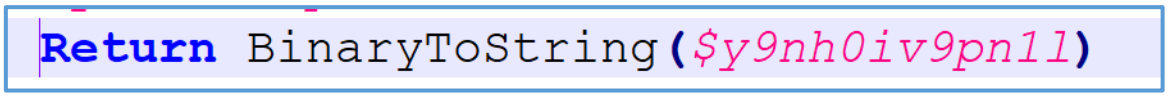
Şekil 4.6 Zararlı Kodların Analizi

Autoit ile elde edilen zararlı yazılım kodunun içerisinde bir fonksiyon dikkat çekmektedir. Fonksiyonun içerisindeki bir değişken 4D5A ile başlamaktadır. Windows işletim sisteminde yürütülebilir her uygulama MZ olarak bilinen değer ile başlar. Zararlı yazılımın içerisinde gömülü farklı bir çalıştırılabilir dosyayı çıkartması zararlının "dropper" kategorisine ait olduğunu işaret etmektedir. Dosya açıldığında en başta bulunan "MZ" ifadesi çalıştırılabilir dosyalar için kullanılmaktadır. Hexadecimal olarak karşılığı "4D5A" 'dır. Bu bulgunun ekran görüntüsü Şekil 4.7'de gösterilmiştir.



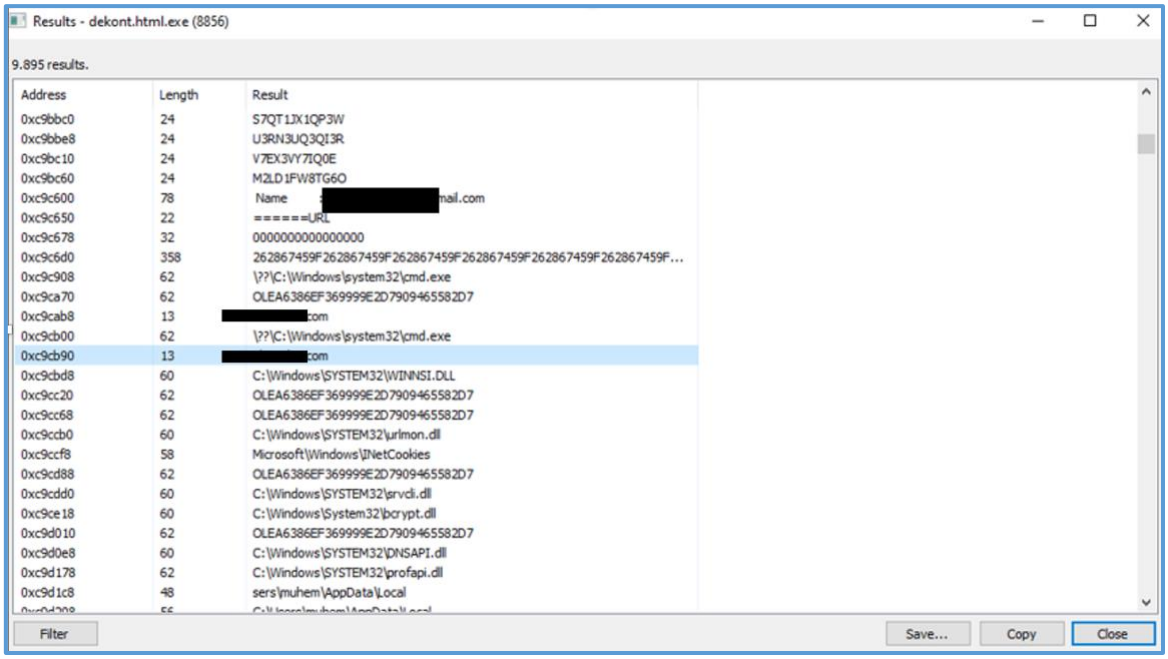
Şekil 4.7 Çalıştırılabilir Dosyanın Tespit Edilmesi

"BinaryToString" fonksiyonu ile gömülü çalıştırılabilir dosya binary formatından string formata çevrilmektedir. Daha sonrasında "%temp%" klasörüne dosyalar çıkarılmaktadır. Bu bulguya ait ekran görüntüsü Şekil 4.8'de verilmiştir.



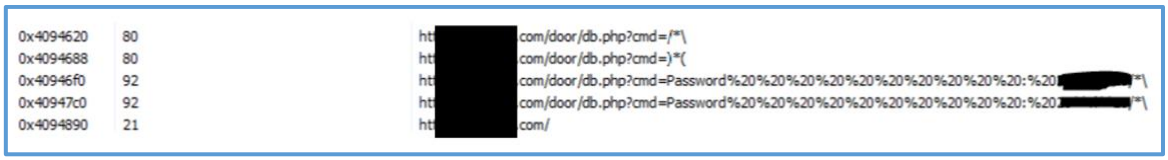
Şekil 4.8 BinaryToString Fonksiyonu Tespiti

Uygulamayı çalıştırıp RAM bellek üzerinde içerisinde geçen stringler analiz edildiğinde bir domain ismi dikkat çekmektedir. Burada tespit edilen domain'e sistem üzerinden tespit edilen şifrelerin yüklendiği daha sonra anlaşılacaktır. Bu domain legal olup hâlen kullanılmaktadır. Saldırganlar amaçları doğrultusunda dikkat çekmemek için daha önceden ele geçirdikleri sistemleri kullanmaktadır. Bu bilgi maskelenmiştir. Bu bilgilere ait detaylar Şekil 4.9'da gösterilmiştir.



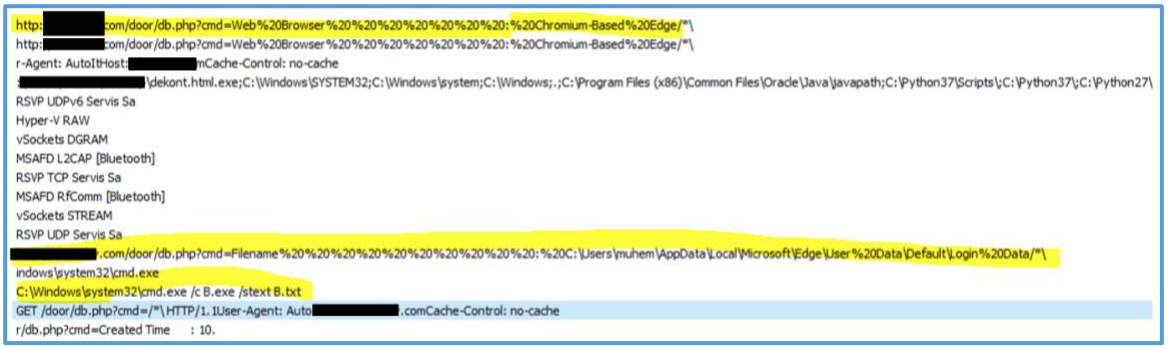
Şekil 4.9 Komuta Kontrol Merkezi Tespiti

Zararlının "password" parametresiyle ilgili domaine bağlantı kurduğu geçici hafızada (memoryde) görülmektedir. Bu bilgiye ait detay Şekil 4.10'da gösterilmiştir.



Şekil 4.10 RAM Analizi

Aşağıdaki verilen Şekil 4.11'deki resimde de görülebileceği gibi zararlı Microsof Edge tarayıcısını kayıtlı parolaları tuttuğu "\\AppData\\Local\\Microsoft\\Edge\\User Data\\Default\\Login Data\\" dosyasına erişmektedir.



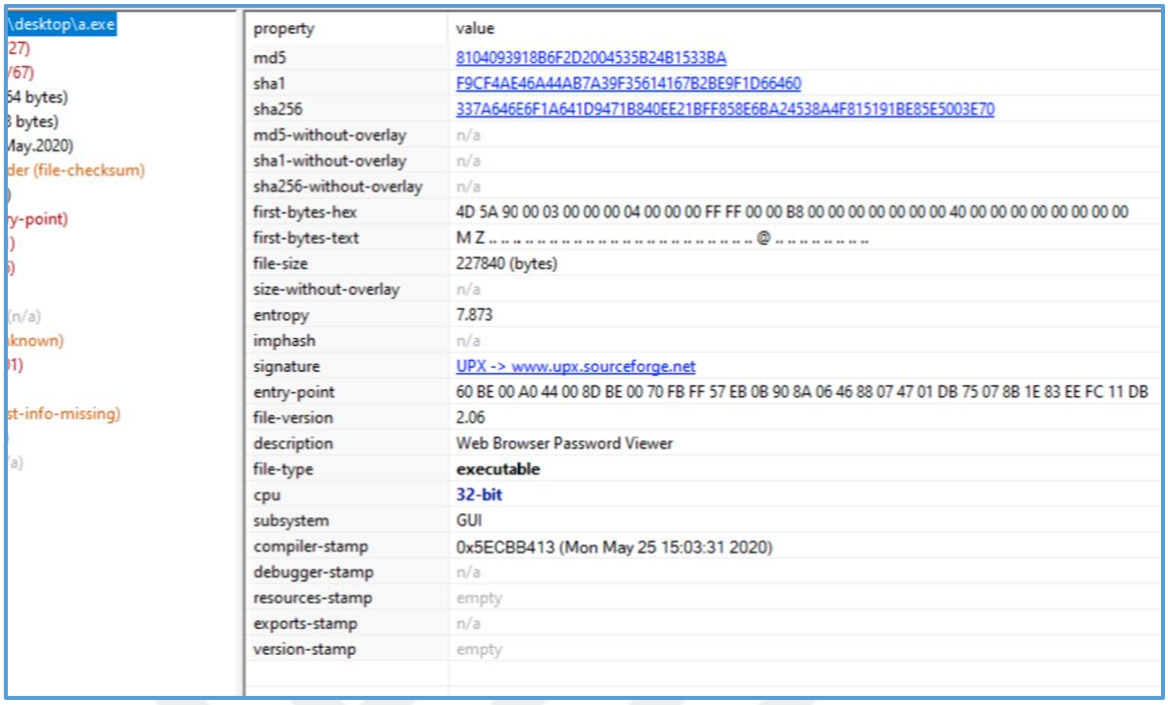
Şekil 4.11 Zararlı Yazılımın Hedefinin Tespiti

Zararlı çalıştığında "c:\Windows\system32\cmd.exe /c B.exe /stext B.txt" komutunun çalıştığı tespit edilmiştir. Zararlı yazılım çalıştığında "temp" dizini altında "A.exe" ve "B.exe" isimli iki uygulama çalışmaktadır. Aynı zamanda temp dizini altında "A.txt" ve "B.txt" isimli iki adet de metin dosyası oluşmaktadır. Bu uygulamaların ekran görüntüleri Şekil 4.12'de verilmiştir.

Şekil 4.12 A.exe ve B.exe Uygulamaları

Zararlı yazılım bu 2 uygulamayı çalıştırıp çıktılarını metin dosyasına yazarak web kanalı üzerinden komuta kontrol(C2) adresine göndermektir. Komuta kontrol olarak kullanılan adres haber içerikli bir web sitesi olduğu anlaşılmıştır. Saldırganlar şüphe uyandırmamak için bu web sitesindeki açıklıktan yararlanarak burayı komuta kontrol sunucusu olarak kullanmaktadır.

"A.exe" (Web Browser Password Viewer) isimli uygulama zaten kullanılan tarayıcı üzerindeki kayıtlı parolaları görmeye yarayan bir uygulamadır. Bu Uygulama UPX ile paketlenmiştir. UPX(Ultimate Packer for Executables) farklı işletim sistemleri tarafından çok fazla dosya formatını destekleyen açık kaynak kodlu bir paketleme uygulamasıdır. Şekil 4.13 ve Şekil 4.14'te zararlılara ait detaylar gösterilmiştir.



property	value
md5	8104093918B6F2D2004535B24B1533BA
sha1	F9CF4AE46A44AB7A39F35614167B2BE9F1D66460
sha256	337A646E6F1A641D9471B840EE21BFF858E6BA24538A4F815191BE85E5003E70
md5-without-overlay	n/a
sha1-without-overlay	n/a
sha256-without-overlay	n/a
first-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00
first-bytes-text	M Z . . . . . @ . . . . .
file-size	227840 (bytes)
size-without-overlay	n/a
entropy	7.873
imphash	n/a
signature	UPX -> www.upx.sourceforge.net
entry-point	60 BE 00 A0 44 00 8D BE 00 70 FB FF 57 EB 0B 90 8A 06 46 88 07 47 01 DB 75 07 8B 1E 83 EE FC 11 DB
file-version	2.06
description	Web Browser Password Viewer
file-type	**executable**
cpu	**32-bit**
subsystem	GUI
compiler-stamp	0x5ECBB413 (Mon May 25 15:03:31 2020)
debugger-stamp	n/a
resources-stamp	empty
exports-stamp	n/a
version-stamp	empty

Şekil 4.13 A.exe Detayları

Yine aynı şekilde "B.exe" (E-Mail Password Recovery) isimli e-Posta kutusu parola görüntüleme uygulaması da UPX ile paketlenmiş olduğu inceleme sonucunda anlaşılmaktadır.

property	value
md5	62B2864C32CB33F57A65F47269D91BE4
sha1	D072FF4E71B3F53E3D198067A61BCDD835CA0D92
sha256	40257944035022D81474E714C256585977F8A89D8F960FA040A64567DE67194A
md5-without-overlay	n/a
sha1-without-overlay	n/a
sha256-without-overlay	n/a
first-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00
first-bytes-text	M Z . . . . . @ . . . . .
file-size	195584 (bytes)
size-without-overlay	n/a
entropy	7.874
imphash	n/a
signature	UPX -> www.upx.sourceforge.net
entry-point	60 BE 00 90 42 00 8D BE 00 80 FD FF 57 EB 0B 90 8A 06 46 88 07 47 01 DB 75 07 8B 1E 83 EE FC 11 DB
file-version	1.90
description	Email Password-Recovery
file-type	**executable**
cpu	**32-bit**
subsystem	GUI
compiler-stamp	0x5DE03B9B (Fri Nov 29 00:26:51 2019)
debugger-stamp	n/a
resources-stamp	empty
exports-stamp	n/a
version-stamp	empty

Şekil 4.14 B.exe Detayları

Zararlı yazılım çalıştırılıp network paketleri incelendiğinde yine aynı şekilde kullanıcı adı ve parola bilgilerini komuta kontrol sunucusuna göndermeye çalışıldığı anlaşılmaktadır. Bu bilgilere ait detaylar Şekil 4.15'te gösterilmiştir.

Şekil 4.15 Zararlı Çalışırken Alınan Ağ Trafiğinin Analizi

Burada anlatılan tüm analizlerden elde edilen ve bu uygulamaya özel olan benzersiz karakterler ayrıştırılır ve not edilir. Daha sonra bu benzersiz string değerleri gerek hexadecimal'e çevrilerek gerekse açık metin olarak kullanılacak şekilde dönüşümleri yapılır. Bu bilgilere ek olarak zararlıya ait özet fonksiyon değerleri de not edilir. Bütün bu bilgiler kullanılarak ve 5. Bölümde detayları açıklanan şekilde bu zararlının tespiti için yara kuralı yazılır. Burada tespiti ve analizi yapılan zararlıya ait YARA kuralı bölüm 4.2.1'de sunulmuştur.

4.2.1. Yapılan Zararlı Yazılım Analizine Ait Yara Kuralı

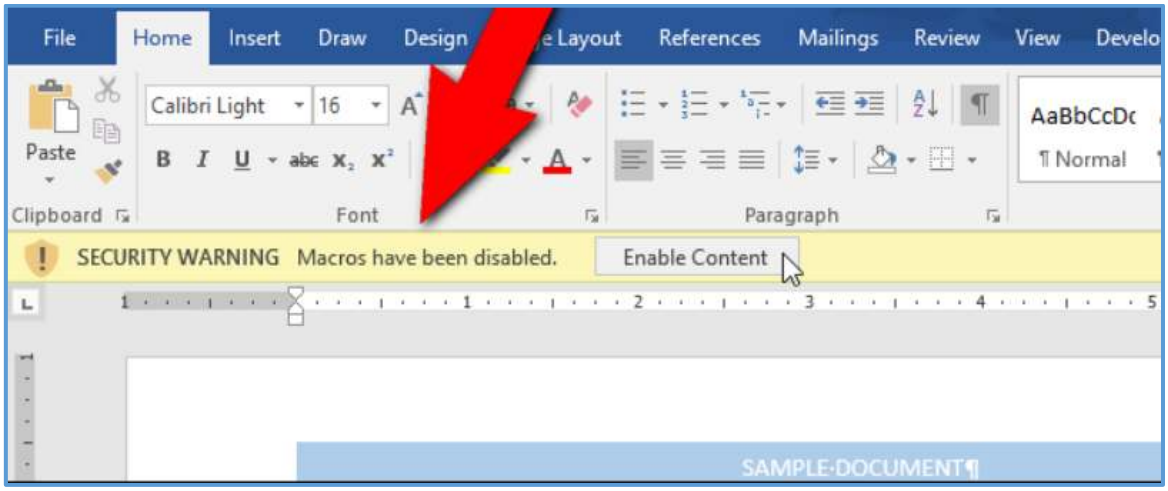
Şekil 4.16'da Bölüm 4.2'de Analiz edilen zararlıya ait yara tespit kuralı verilmiştir.

```
rule Dropper_dekont {
   meta:
      description = "dekont - file dekont.exe"
      date = "2021-12-08"
      hash1 = "ca75d22f12250ac4f3b156569655f8a181e9ef10f32befb8a63a0f13773e810d"
   strings:
      $s2 = "/AutoIt3ExecuteScript" fullword wide
      $s4 = "/AutoIt3ExecuteLine" fullword wide
      $s5 = "PROCESSGETSTATS" fullword wide
      $s6 = "WINGETPROCESS" fullword wide
      $s7 = "SCRIPTNAME" fullword wide /* base64 encoded string 'H$H=3@0' */
      $s8 = "SHELLEXECUTE" fullword wide
      $s9 = "SHELLEXECUTEWAIT" fullword wide
      $s10 = "*Unable to get a list of running processes." fullword wide
      $s11 = "HTTPSETUSERAGENT" fullword wide
      $s12 = "PROCESSCLOSE" fullword wide
      $s13 = "PROCESSEXISTS" fullword wide
      $s14 = "PROCESSLIST" fullword wide
      $s15 = "PROCESSSETPRIORITY" fullword wide
      $s16 = "PROCESSWAIT" fullword wide
      $s17 = "PROCESSWAITCLOSE" fullword wide
      $s18 = "PROCESSORARCH" fullword wide
      $s19 = "STRINGREVERSE" fullword wide /* base64 encoded string 'I4H4dDTDR' */
      $s20 = "Error parsing function call.0Incorrect number of parameters in function call.'\"ReDim\" used without an array variable.>Illegal " wide
      $o1 = "IsDebuggerPresent"
      $o2 = "VirtualAlloc"
      $o3 = "InternetOpenUrlW"
      $o4 = "InternetReadFile"
      $o5 = "HttpOpenRequestW"
      $o6 = "HttpSendRequestW"
      $o7 = "FtpOpenFileW"
      $o8 = "LoadLibraryA"
      $o9 = "GetProcAddress"
      $o10 = "ReadFile"
      $o11 = "WriteFile"
      $o12 = "GetTempPathW"
   condition:
      uint16(0) == 0x5a4d and filesize < 5000KB and
      8 of ($s*) and all of ($o*)
}
```

Şekil 4.16 Bölüm 4.2'de Analiz Edilen Zararlıya Ait Yara Tespit Kuralı

4.3. İkinci Oltalama Saldırısı Örneği ve Zararlı Yazılım İle Kalıcılık Sağlama

Birinci örnekte olduğu gibi bilinen bir bankacılık şirketi olarak kendisini gösteren "xxx@Xbank[.]com" isimli e-Posta adresinden "E-DEKONT.doc" isimli ek dosyasını içeren (zararlı) e-Posta gönderimi yapılmıştır. Senaryo gereği "E-DEKONT.doc" isimli dosya windows outlook client uygulaması üzerinde Microsoft korumalı modda çalıştırılmıştır. Kullanıcı bu uygulamayı korumalı moddan çıkarmak için gereken butona tıklamıştır ve zararlı makronun çalışmasına sebep olmuştur. Bu senaryoya ait örnek Şekil 4.17'de gösterilmiştir.



Şekil 4.17 Zararlı Makronun Aktif Edilmesi

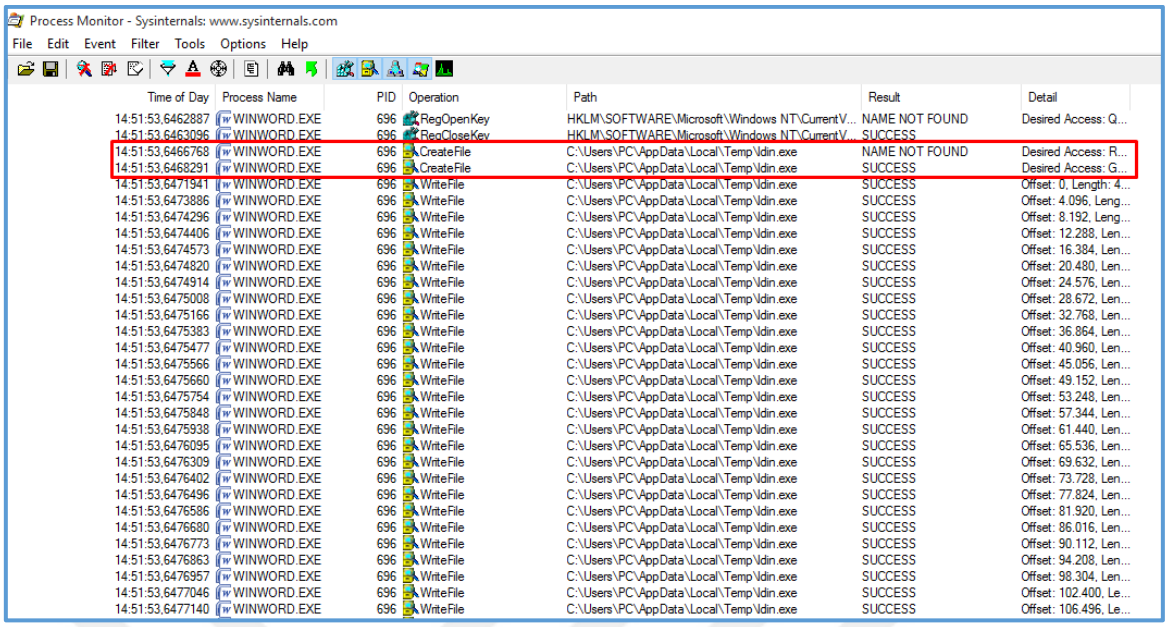
Bahsedilen e-posta adresinden gönderilen “E-DEKONT.doc” isimli zararlı yazılım içeren dosya incelenmiştir. “E-DEKONT.doc” zararlı Word dokümanı test ortamında çalıştırılarak sistem üzerindeki aktiviteleri incelenmiştir. Zararlı doküman ilk çalıştığı anda “C:\Users\<Kullanıcı Adı>\AppData\Local\Temp” dizininde “1.a” ve “ldin.exe” isimli zararlı dosyaları oluşturduğu tespit edilmiştir. Ayrıca oluşan “ldin.exe” isimli zararlı yazılımın “C:\Users\<Kullanıcı Adı>\AppData\Roaming” dizinine “SYSTEMGT” isimli ve “ldin.exe” dosyası ile birebir aynı olan dosyayı oluşturduğu belirlenmiştir. “SYSTEMGT” isimli zararlı yazılım oluştuktan sonra “C:\Users\<Kullanıcı Adı>\AppData\Local\Temp” dizininde oluşan “1.a” ve “ldin.exe” isimli zararlı yazılımların silindiği gözlemlenmiştir. Oluşturulan “SYSTEMGT” dosyası ilk olarak 30 dakika sonrasına çalışması için zamanlanmış görev olarak kendisini kayıt ettiği tespit edilmiştir. Ayrıca zararlı yazılım ilk çalıştırmadan itibaren her 1 dakikada kendisini yeniden çalıştırmakta olduğu belirlenmiştir.

Bu bilgiler doğrultusunda, incelenen uç nokta bilgisayarı üzerinden;

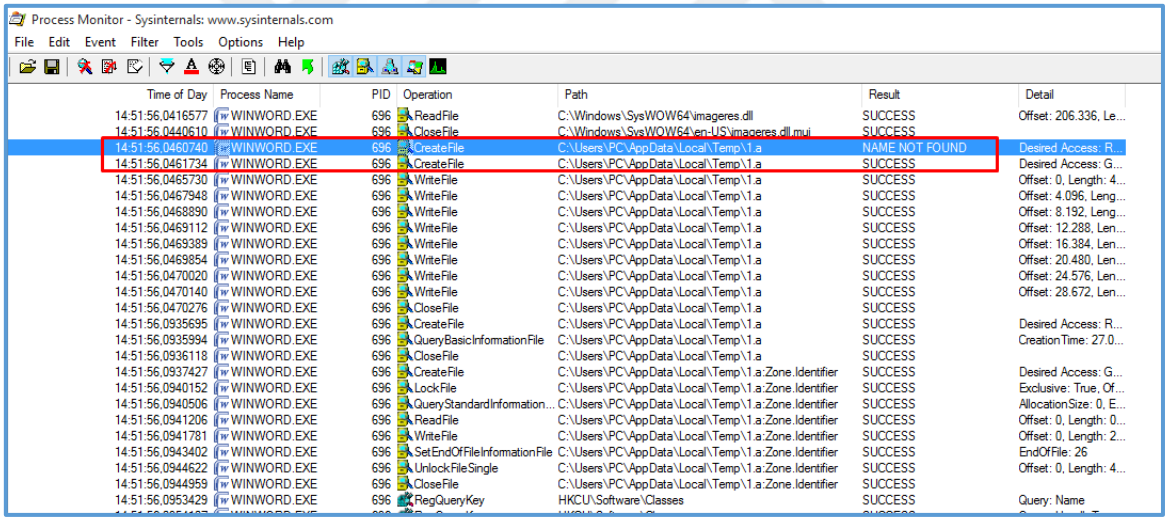
- “C:\Windows\Prefect”, “C:\Windows\System32\winevt\Logs\Security.evtx”,
- “\$MFT”,
- “\$Logfile”,
- “\$USNJournal”,
- “C:\Windows\System32\wbem\Repostory\OBJECTS.DATA”,
  “C:\Windows\appcompat\Programs\Amcache.hve”,
  “C:\Windows\System32\config\SYSTEM”,
- “C:\Users\User\NTUSER.DAT”,

konumlarında bulunan, çalışan uygulamalar ile ilgili kayıtların tutulduğu dosyalar elde edilmiştir. Elde edilen dosyalar adli bilişim inceleme teknikleri kullanılarak analiz edilmiştir.

“E-DEKONT.doc” isimli zararlı doküman analiz edildiğinde “CVE-2017-11882” kodlu Microsoft Office zafiyetini kullanarak sistem üzerinde “ldin.exe” isimli zararlı yazılım oluşturduğu ve çalıştırdığı tespit edilmiştir. Ayrıca zararlı yazılımı çalıştırmak için kullandığı tespit edilen “1.a” isimli bir dosyanın da sistem üzerinde oluşturulduğu belirlenmiştir. Bulgulara ait ekran görüntüleri Şekil 4.18 ve Şekil 4.19’da sunulmuştur.

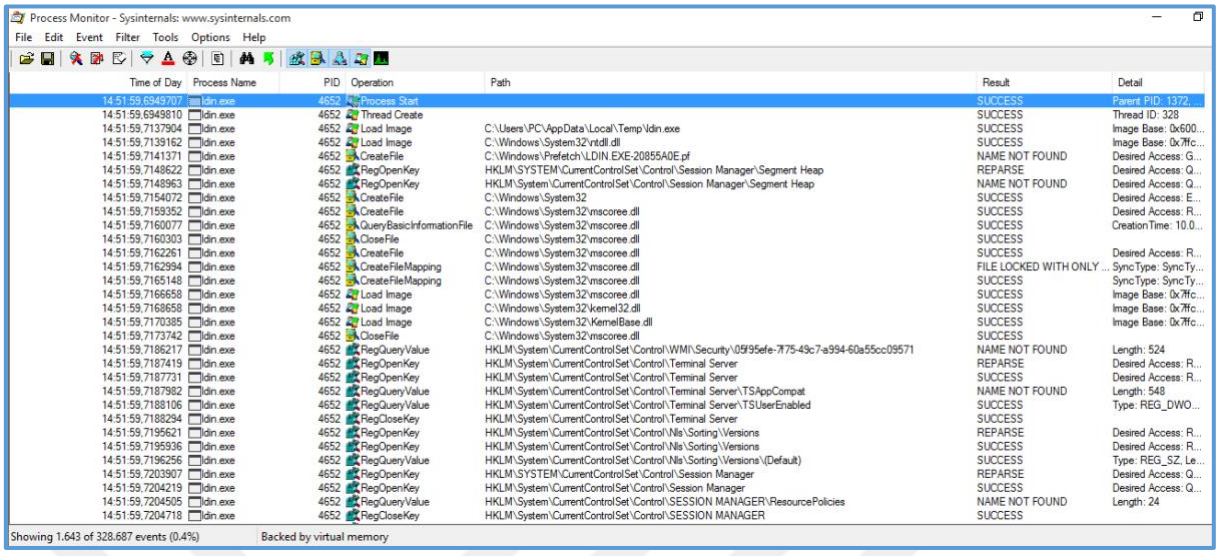


Şekil 4.18 " WINWORD.EXE"'nin "ldin.exe"'yi Oluşturması

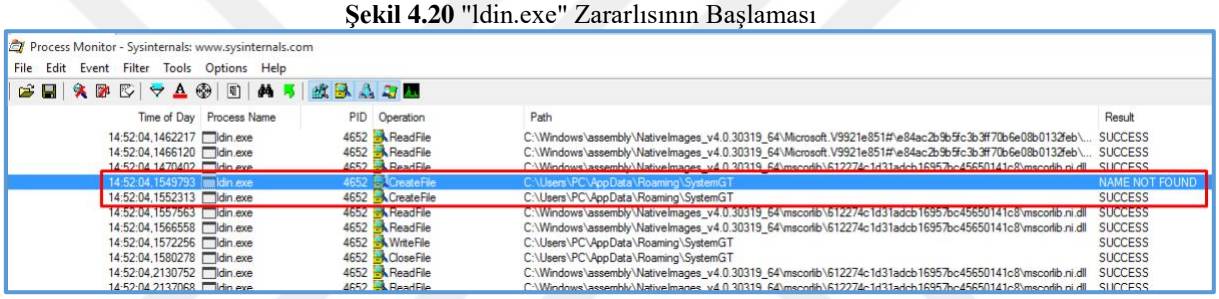


Şekil 4.19 "WINWORD.EXE"'nin "1.a"'yi Oluşturması

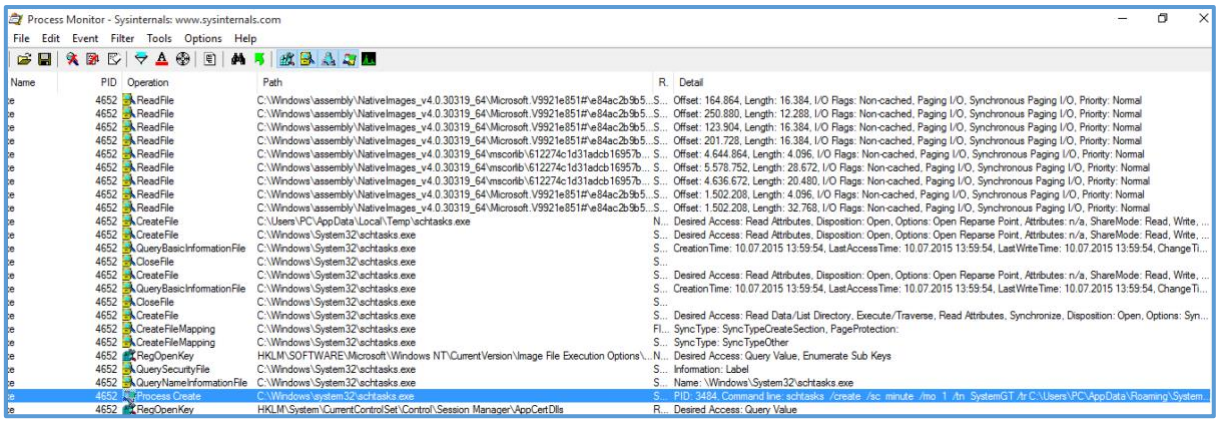
Zararlı doküman tarafından sistem üzerinde oluşturulan “ldin.exe” isimli dosya analiz edildiğinde kendisini “C:\Users\<Kullanıcı Adı>\AppData\Roaming” altına kopyaladığı ve “SYSTEMGT” ismiyle kaydettiği ve kendisini sildiği tespit edilmiştir. Daha sonra ise “PID: 3484, Command line: schtasks /create /sc minute /mo 1 /tn SystemGT /tr C:\Users\PC\AppData\Roaming\SystemGT” komutu çalıştırılarak “SYSTEMGT” zararlısı 1’er dakika aralıklarla çalışmak üzere bir zamanlanmış görev oluşturduğu ve kendisinin bir kopyası olan “SYSTEMGT” yazılımını çalıştırdığı belirlenmiştir. Bulgulara ait ekran görüntüleri Şekil 4.20, Şekil 4.21 ve Şekil 4.22’de sunulmuştur.



Şekil 4.20 "ldin.exe" Zararlısının Başlaması



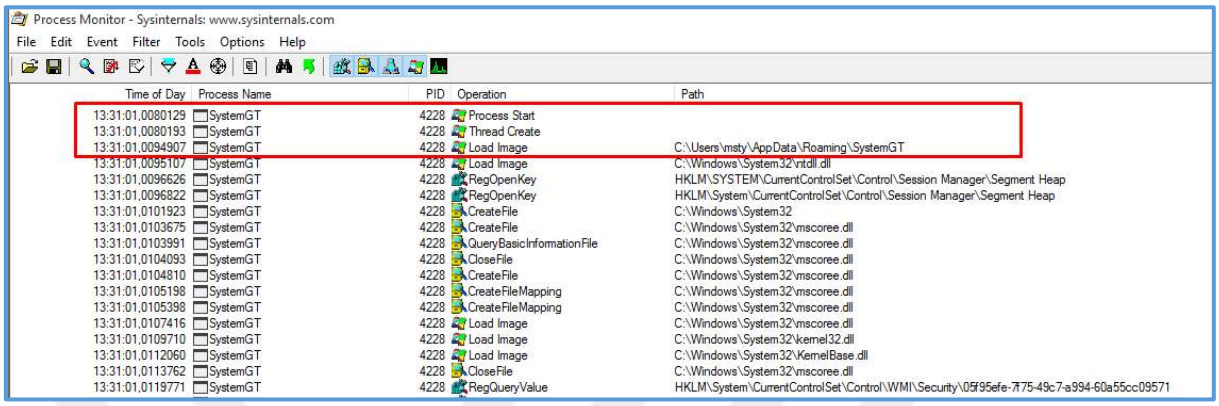
Şekil 4.21 " ldin.exe"'nin "SYSTEMGT"'yi Oluşturması



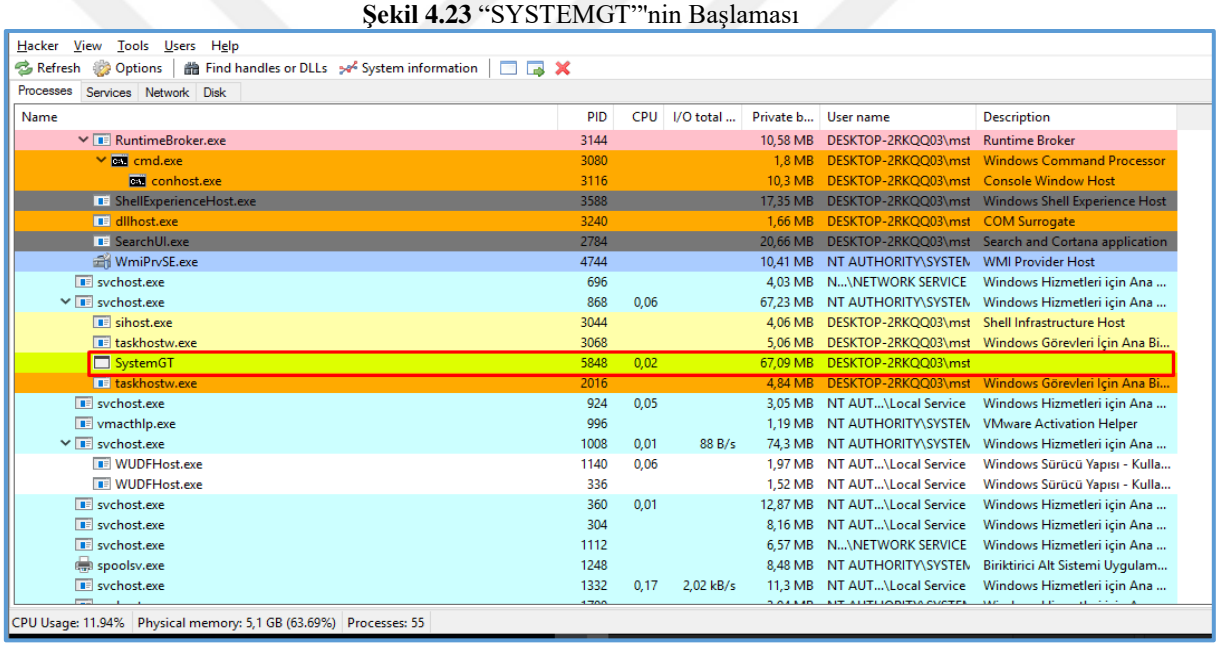
Şekil 4.22 "ldin.exe" 'nin Zamanlanmış Görev Oluşturması

"SYSTEMGT" isimli dosya çalıştığı süre boyunca ağ hareketleri kayıt edilmiştir. Kayıt edilen ağ paket dosyası incelendiğinde "SYSTEMGT" isimli dosya her çalıştığında, saldırgan kontrolünde olan uzak **"93.125.99[=]72 (ftp.gr[=]by)"** IP adresine **FTP protokolü** ile bağlanarak tarayıcı

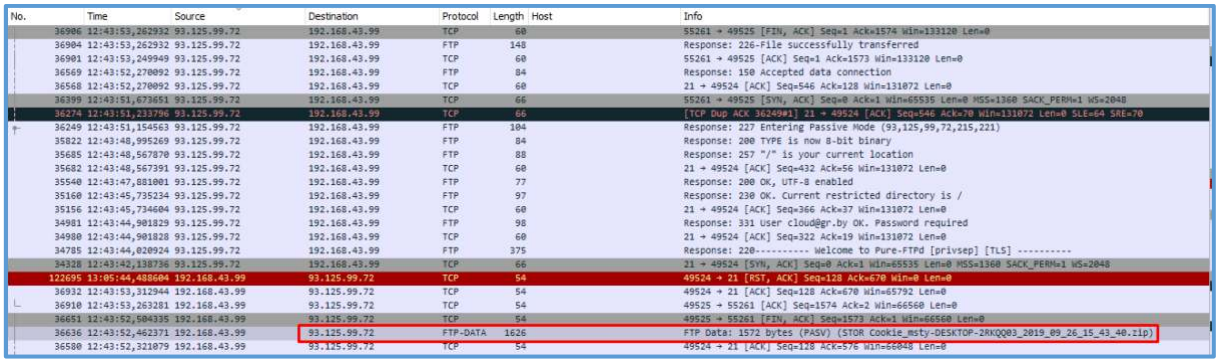
üzerinde tutulan kimlik ve oturum bilgilerini, zip ve HTML dosyası olarak gönderdiği tespit edilmiştir. Bulgulara ait ekran görüntüleri Şekil 4.23, Şekil 4.24 ve Şekil 4.25'te sunulmuştur.



Şekil 4.23 "SYSTEMGT"'nin Başlaması



Şekil 4.24 "SYSTEMGT" nin Sistemde Aktif Çalıştığının Görülmesi



Şekil 4.25 Belirtilen IP Adresine FTP Protokolü ile Kimlik/Oturum Bilgilerinin Gönderilmesi

"E-DEKONT.doc" ve "ldin.exe" isimli zararlı yazılımlara ait SHA256 hash değerleri sırasıyla;

- "ce35722bd6bd289ecba98612897b4f4b927410948ec2619544f2e81843834eec",
- "e64fff0aa41afd4ec73fc7bd502b8e71189f6faa6fee3898106dd867d5f51ae9",

Olarak tespit edilmiştir. Bu bilgiler de yara kuralı yazılırken kullanılmıştır.

4.3.1. Yapılan Zararlı Yazılım Analizine Ait Yara Kuralı

Şekil 4.26'da Bölüm 4.3'te Analiz edilen zararlıya ait yara tespit kuralı verilmiştir.

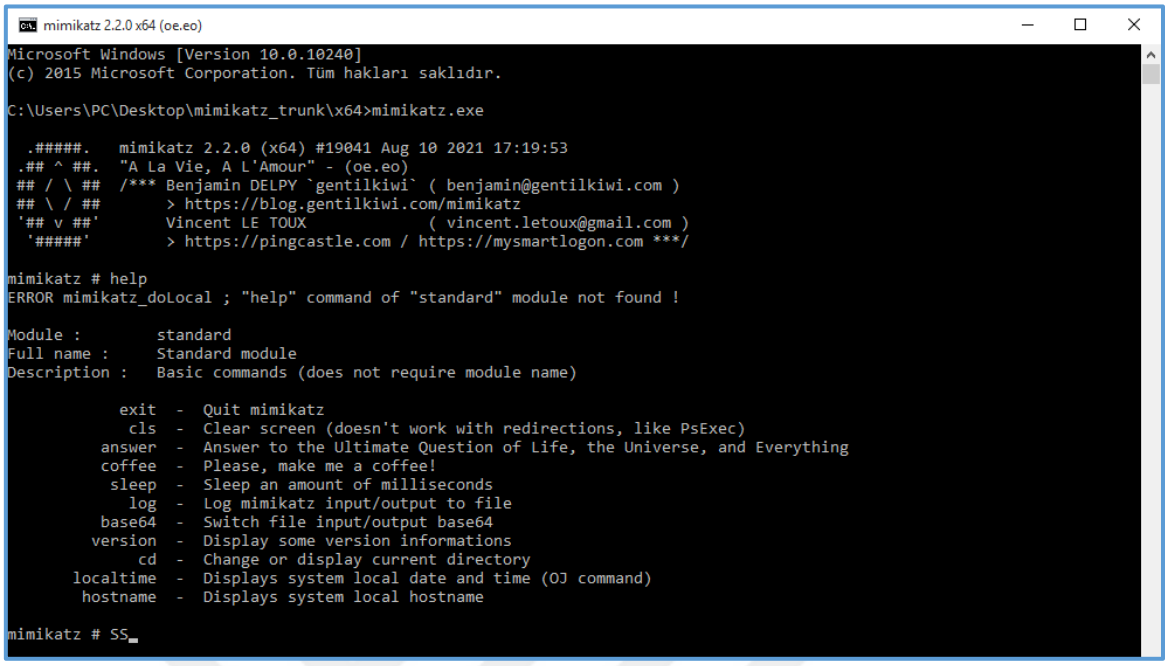
```
rule e-dekont-password-stealer-detect {
    strings:
        $s1 = { 3D D4 40 36 19 DB A1 64 A1 32 23 D7 }
        $s2 = {13 BB FB 43 48 E5 D7 F5 35 9F A6 59 9D 9B F5}

    condition:
        all of them
}
```

Şekil 4.26 Bölüm 4.3'te Analiz Edilen Zararlıya Ait Yara Tespit Kuralı

4.4. Kimlik Bilgisi Elde Etme Saldırısı Örneği

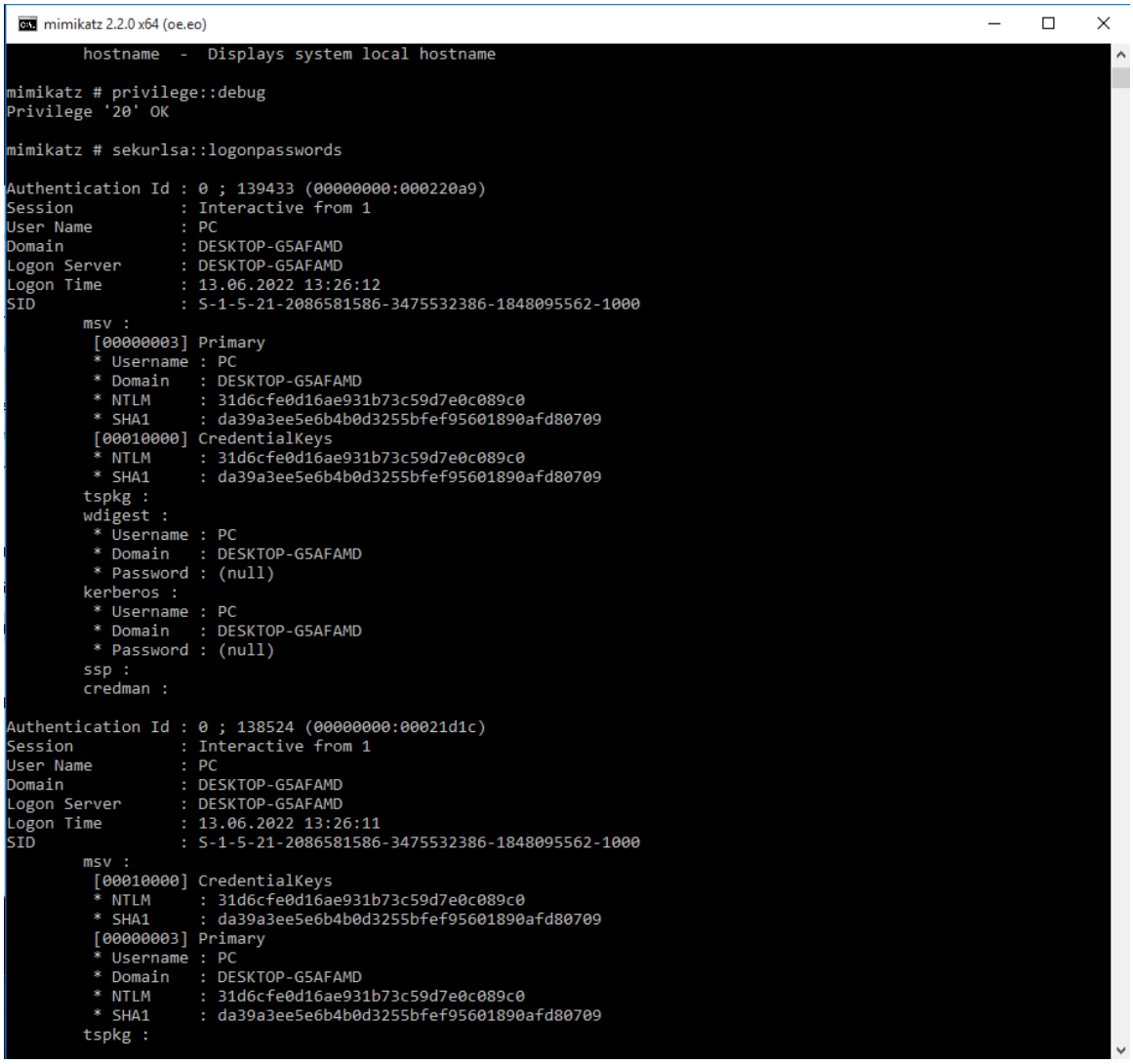
Bu atak tekniğinde saldırganının hedef makineye Bölüm 2'de açıklanan tekniklerden herhangi birini kullanarak eriştiği ve kimlik bilgilerini elde etmek için kullanılan zararlı yazılımın simüle edildiği senaryo örneği açıklanmaktadır. Bu bölümde hedef makineye saldırganın senaryo gereği erişiminin olduğu ve mimikatz zararlısını kullanarak hedefine ulaştığı senaryo açıklanmıştır. Buradaki analiz çıktısı sonucunda bu tür saldırıyla karşılaşan kuruluşların oluşturulan yara kuralı ile tespiti yapması hedeflenmektedir.

Mimikatz, "Benjamin Delpy" isimli açık kaynak kod geliştirici bir kişi tarafından geliştirilen "post-explotation" aracıdır. "Post–explotation", bir sisteme yetkisiz erişim sağlandıktan sonra gerçekleştirilebilen eylemleri tanımlayan genel bir siber güvenlik kavramıdır. Aracın amacı, düz metin şifrelerini, Microsoft windows işletim sistemleri içerisinde kimlik bilgilerinin tutulduğu proses içinden kimlik bilgilerine ait hash değerlerini ve yine Windows işletim sistemlerine ait kerberos biletlerini bellekten elde etmeyi amaçlayan ve kimlik bilgisi hırsızlığına hizmet eden zararlı yazılımdır. Bu araçla elde edilecek olan hash ve biletlerle AD ortamında yatayda hareket etmeyi sağlar. Bu araca ait giriş ekranı görüntüsü Şekil 4.27'de verilmiştir.

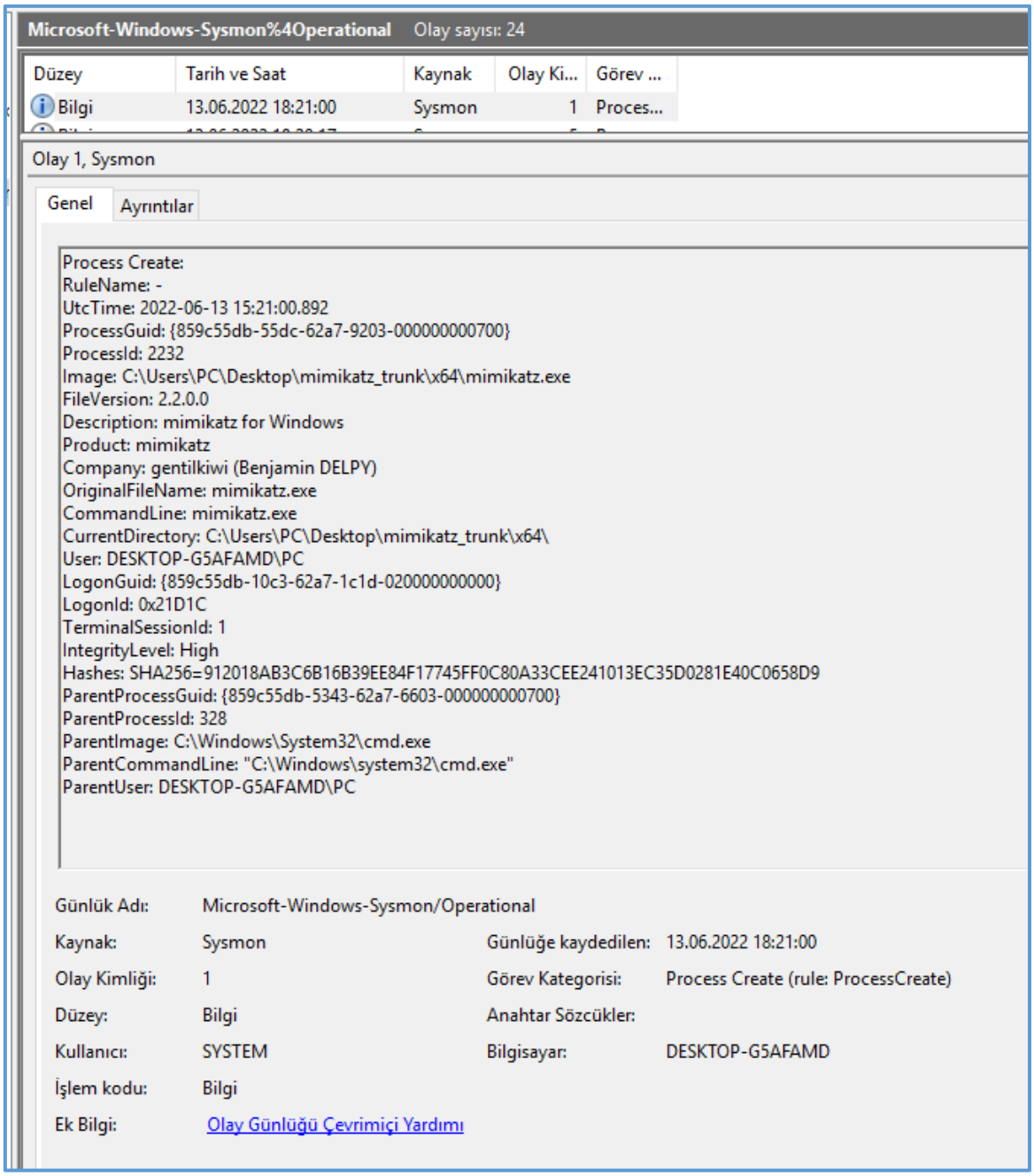


Şekil 4.27 Mimikatz Zararlısına Ait Ekran Görüntüsü

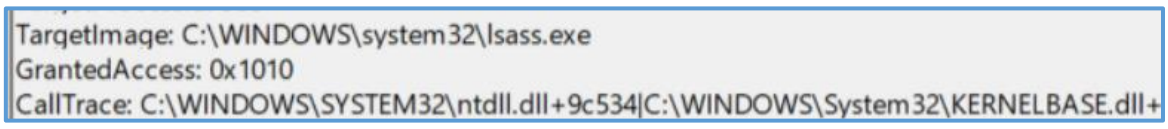
Bu senaryoda kurban sisteminin hedefli kimlik avı kullanarak veya sıfır gün istismarı nedeniyle güvenliğinin ihlal edildiğini varsayalım. Saldırgan hedef sisteme ulaşmış ve "mimikatz" isimli zararlıyı çalıştırmıştır. Zararlının yeteneklerini kullanarak sistemde oturum açmış kullanıcılara ait hesap bilgilerini ele geçirmiştir. Bu bilgilere ait detaylar Şekil 4.28, Şekil 4.29 ve Şekil 4.30'da verilmiştir.



Şekil 4.28 Mimikatz İle Kimlik Bilgilerinin Ele Geçirilmesi



Şekil 4.29 Mimikatz Programının Çalıştığını Sysmon İle Tespit Etme



Şekil 4.30 Mimikatz ile Kimlik Verilerinin Okunduğunun Tespiti

4.4.1. Yapılan Zararlı Yazılım Analizine Ait Yara Kuralı

Şekil 4.31'de Bölüm 4.4'de Analiz edilen zararlıya ait yara tespit kuralı verilmiştir. Buradaki zararlı türü çok genel olması sebebiyle tespit için birden fazla kural önerilmiştir.

```
rule Invoke_Mimikatz {
    meta:
        description = "Detects Invoke-Mimikatz String"
        license = "Detection Rule License 1.1 https://github.com/Neo23x0/signature-base/blob/master/LICENSE"
        reference = "https://github.com/clymb3r/PowerShell/tree/master/Invoke-Mimikatz"

    strings:
        $x2 = "TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGAEAAA4fug4AtAnNIbgBTM0hVGhpcyBwcm" ascii
        $x3 = "Write-BytesToMemory -Bytes $Shellcode1 -MemoryAddress $GetCommandLineWAddrTemp" fullword ascii
    condition:
      1 of them
}

rule Powerkatz_DLL_Generic {
    meta:
        description = "Detects Powerkatz - a Mimikatz version prepared to run in memory via Powershell (overlap with other Mimikatz versions is possible)"
        reference = "PowerKatz Analysis"
        date = "2016-02-05"
        super_rule = 1
        score = 80
        hash1 = "c20f30326fcebad25446cf2e267c341ac34664efad5c50ff07f0738ae2390eae"
        hash2 = "1e67476281c1ec1cf40e17d7fc28a3ab3250b474ef41cb10a72130990f0be6a0"
        hash3 = "49e7bac7e0db87bf3f0185e9cf51f2539dbc11384fefced465230c4e5bce0872"
    strings:
        $s1 = "%3u - Directory '%s' (*.kirbi)" fullword wide
        $s2 = "%*s  pPublicKey         : " fullword wide
        $s3 = "ad_hoc_network_formed" fullword wide
        $s4 = "<3 eo.oe ~ ANSSI E>" fullword wide
        $s5 = "\\*.kirbi" fullword wide

        $c1 = "kuhl_m_lsadump_getUsersAndSamKey ; kull_m_registry_RegOpenKeyEx SAM Accounts (0x%08x)" fullword wide
        $c2 = "kuhl_m_lsadump_getComputerAndSyskey ; kuhl_m_lsadump_getSyskey KO" fullword wide
    condition:
        ( uint16(0) == 0x5a4d and filesize < 1000KB and 1 of them ) or 2 of them
}

rule mimikatz_sekurlsa_detection {
    strings:
        $s1 = { 33 DB 8B C3 48 83 C4 20 5B C3 }
        $s2 = {83 64 24 30 00 44 8B 4C 24 48 48 8B 0D}
        $s3 = {83 64 24 30 00 44 8B 4D D8 48 8B 0D}
        $s4 = {84 C0 74 44 6A 08 68}
        $s5 = {8B F0 3B F3 7C 2C 6A 02 6A 10 68}
        $s6 = {8B F0 85 F6 78 2A 6A 02 6A 10 68}

    condition:
        all of them
}

rule mimikatz_decryptkeysign {
    strings:
        $s1 = { F6 C2 07 0F 85 0D 1A 02 00 }
        $s2 = { F6 C2 07 0F 85 72 EA 01 00 }
        $s3 = { 4C 8B CB 48 89 44 24 30}
        $s4 = { 4c 89 1b 48 89 43 08 49 89 5b 08 48 8d }

    condition:
        3 of them
}
```

Şekil 4.31 Bölüm 4.4’de Analiz Edilen Zararlıya Ait Yara Tespit Kuralı

5. YARA İLE ZARARLI YAZILIM TESPİTİ

Adli bilişimde soruşturmanın önemli yönlerinden biri, bir sistem veya ağdaki olası kötü amaçlı yürütülebilir dosyaların tanımlanması ve sınıflandırılmasıdır [42]. Zararlı yazılım tespiti, dosyalara ait bilinen özelliklerin tespit edilmesi yöntemi kullanılarak yapılmaktadır. Bu tür kötü amaçlı zararlı yazılımları tespit etmek için, şüpheli olarak değerlendirilen dosyanın özet fonksiyon değerini (hash) hesaplamak ve bunu bilinen kötü amaçlı yazılımların özet fonksiyon değerleriyle karşılaştırmaktır. Ancak bu teknikle zararlıların tespitinin yapılması her zaman mümkün olmamaktadır. Bu yöntemle zararlılar tam manasıyla tespit edilemez [42].

Bunun sebebi özet fonksiyon değerlerinin kolaylıkla değiştirilebilmesine dayanmaktadır. Özet fonksiyon algoritmaları girdi ne olursa olsun sabit bir çıktı üretir. Bu durumda girdide oluşabilecek çok küçük bir değişiklik çıktının tamamen değişmesine sebep olur. Bu sayede zararlı yazılımlara ait özet fonksiyon değerlerini değiştirmek kolay bir hale gelmektedir. Örneğin zararlı kod içerisindeki bir parametre isminin değiştirilmesi veya bir yorum satırı eklenmesi zararlının özet fonksiyon değerini tamamen değiştirecek ve gizli kalmasını sağlayacaktır [42].

Virüsten koruma yazılımları veya kurumsal zararlı yazılım tespiti çözümleri bir dosyada belirli türleri veya zararlı yazılımın dâhil olduğu aile gruplarını tanımlayan belirli bir metin kümesini tarar. Virüsten koruma yazılımları, belirli bir zararlıya özgü bir dizi bayt da arayabilir. Bazı çözümler davranış analizi yaparak zararlı yazılımları tespit etmeyi hedefler. Kötü amaçlı yazılım veya zararlı yazılım, karmaşık ve kötü niyetli bir yazılımın parçasıdır. Davranışı, bilgisayar sistemlerindeki basit değişiklikler gibi temel eylemlerden gelişmiş davranış kalıplarına kadar uzanır [43].

Tanım olarak kötü amaçlı yazılım, veri ve kimlik çalma, casusluk, meşru kullanıcı bulaşması gibi bilgisayar sistemlerine zarar vermek ve geliştiricisine tam veya sınırlı kontrol sağlamak amacıyla kötü niyetli bir yazılım parçasıdır [43].

YARA, metin veya ikili desenler gibi özel anti virüs benzeri imza algılamasına dayalı olarak kötü amaçlı yazılımları tanımlamak için kural tabanlı bir yaklaşım kullanan açık kaynaklı bir araçtır. Kurallar veya açıklamalar, dizilerden ve mantıktan oluşturulur ve örneği belirli zararlı yazılım ailelerine veya varyantlarına göre sınıflandırmak için kalıplar veya özelliklerle eşleşir [44].

5.1. Yara Tanımı

Yara kuralları, siber güvenlik analistleri için her alanda kullanabileceği pratik bir araçtır. Kötü amaçlı yazılım ailesini tespit etmek için yüksek kaliteli Yara kuralları geliştirmek, bu alanda kendisini geliştirmiş uzmanlar için bile yoğun emek ve zaman alabilir. Yara kuralları, metinsel veya ikili kalıplara dayalı olarak kötü amaçlı yazılım ailelerinin açıklamalarını oluşturarak kötü amaçlı yazılım örneklerini sınıflandırır ve tanımlar. Zararlı dosyaları hızlı bir şekilde bulmak için bir dosya veya bir dosyanın bileşeniyle eşleşecek metin veya ikili kalıpları tanımlamak için Yara kurallarını kullanabiliriz. YARA, kurallar tarafından tanımlanan kıstasları karşılayan herhangi bir dosyayı tanımlamak için YARA'nın bir dosya sistemini taradığı "yar" veya "yara" formatında bir kural dosyası kullanarak çalışır. VirusTotal firması tarafından geliştirilmiş ve ücretsiz kullanıma sunulmuştur [45].
En basit tabiri ile bir yara kuralı Şekil 5.1'de gösterilmiştir.

```
rule sozdekod_yara : ornek
{
      meta:
            description = "Bu örnek bir YARA kuralıdır."
            threat_level = 3
      strings:
            $a = {6A 40 68 00 30}
            $b = "malware"
      condition:
            $a OR $b
}
```

Şekil 5.1 Örnek Yara Kuralı

Şekil 5.1'de verilen ve "rule" ile başlayan sözdekod_yara isimli örnek kural, bir Yara kuralının başlangıcını gösteren başlık olarak tanımlanabilir. Sozdekod_yara dizesi, yara kuralı için seçilen isimdir. Yara kuralının başarılı olabilmesi için içeriğine uygun başlık seçmek önemlidir. "meta:" bölümü, kural ile ilgili detayların ve üst verilerin bulunduğu alandır. Buradaki bilgiler kuralın tanımlanması için önemlidir. "strings" bölümü, kural ile ilgili tespit yapılmak istenen ve desteklenen her formatta verinin yazılabileceği alandır. "condition:" veya koşul bölümü, Yara kurallarınızın bir eşleşmeyi tetikleyebilmesi için kontrol etmek istediğiniz koşulları tanımlar. Tüm koşul şartları burada uygulanabilir [45].

Yara, her türlü bilgisayar ortamındaki kötü amaçlı dosyaları tespit etmek için bir kural kümesidir. Kötü niyetli faaliyetlerle ilgili tespiti kolaylaştırır. Hızlı ve plaform bağımsız çalışır. Kurulum gerektirmez. Kod karmaşıklaştırma gibi karmaşık tekniklerin Tespitinde önemli rol oynar. Genel kullanım standartları aşağıda açıklanmıştır [45].

- Strings:
    - Hex => {} arasına yazılır.
    - Text => "" arasında yazılır.
    - Regex => // arasına yazılır.
- Condition:
    - And
    - Or
    - Any of them
    - All of them
    - Counting strings (#a == 6 and #b > 10)

Yara'ya ait tüm koşullar Şekil 5.2'de sunulmuştur.

Keyword	String Types	Summary	Restrictions
nocase	Text, Regex	Ignore case	Cannot use with xor, base64, or base64wide
wide	Text, Regex	Emulate UTF16 by interleaving null (0x00) characters	None
ascii	Text, Regex	Also match ASCII characters, only required if wide is used	None
xor	Text	XOR text string with single byte keys	Cannot use with nocase, base64, or base64wide
base64	Text	Convert to 3 base64 encoded strings	Cannot use with nocase, xor, or fullword
base64wide	Text	Convert to 3 base64 encoded strings, then interleaving null characters like wide	Cannot use with nocase, xor, or fullword
fullword	Text, Regex	Match is not preceded or followed by an alphanumeric character	Cannot use with base64 or base64wide
private	Hex, Text, Regex	Match never included in output	None

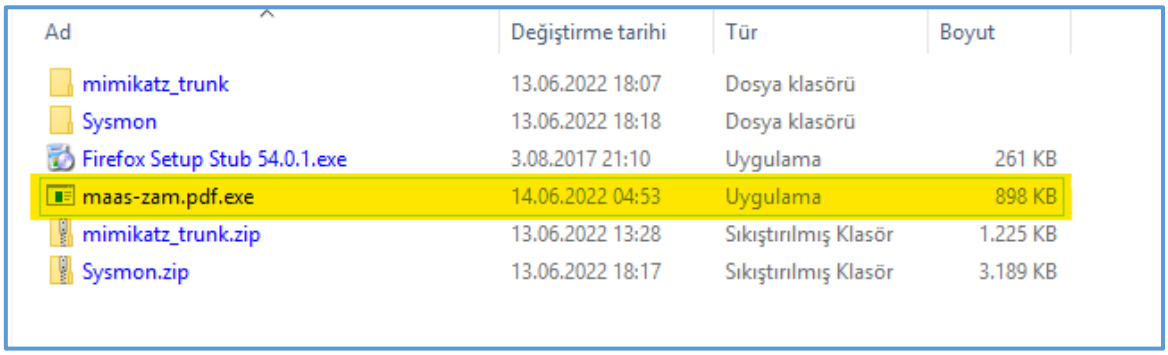
Şekil 5.2 Yara Koşulları

Yara Kullanımı için "http://virustotal.github.io/yara/" adresinden araç indirilir ve "Yara.exe –mrs <kural.yar> <taranacak _dizin>" komut dizesi çalıştırılarak tarama yapılır.

5.2. Örnek Yara Kullanımı

Siber güvenlik analisti olarak CERT ekibinde çalışıyorsunuz ve sistemlerinizden birinde kötü amaçlı bir dosya buldunuz. Yöneticiniz size dosyayı analiz etmenizi ve benzersiz verileri toplamanızı, bir Yara kuralı yazmanız ve ardından bunu kurum genelinde aramanız için görevlendirdi. Kötü amaçlı dosya, yürütülebilir ".exe" dosyası olsun ancak kullanıcıları kandırmak için kendisini bir ".pdf" dosyası olarak gizlediğini ve bu nedenle kötü amaçlı yazılımı çalıştıran kişiler bunu normal bir pdf dosyası olarak düşünmesini sağladığını varsayalım. Bu dosyanın adı

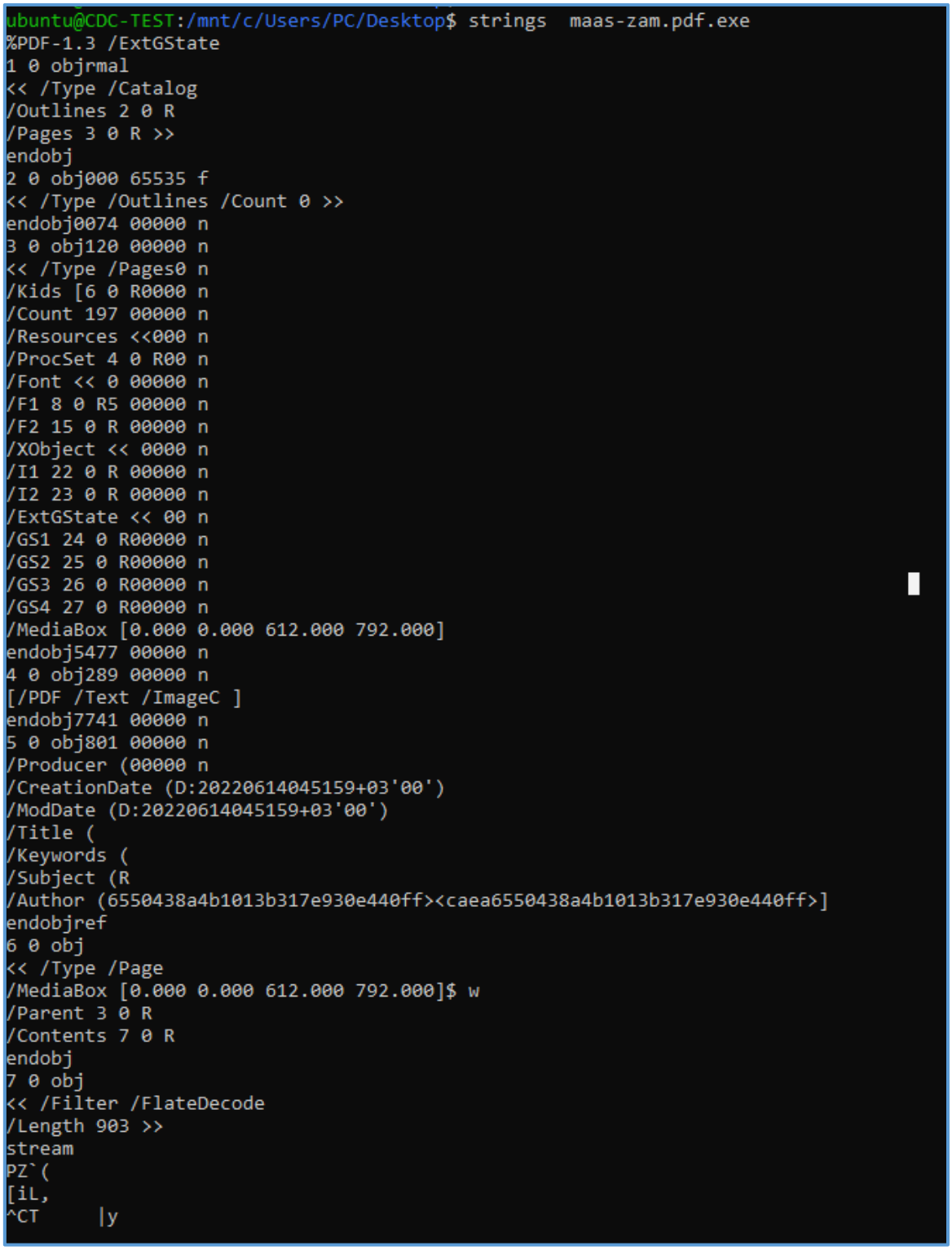
"maas-zam" olduğunu tespit ediyorsunuz. Bunun incelemesini yapmanız ve tüm kurumda bu dosya varlığını kontrol etmeniz gerekiyor. Bu örnekte bunun nasıl yapılacağını açıklayacağız.



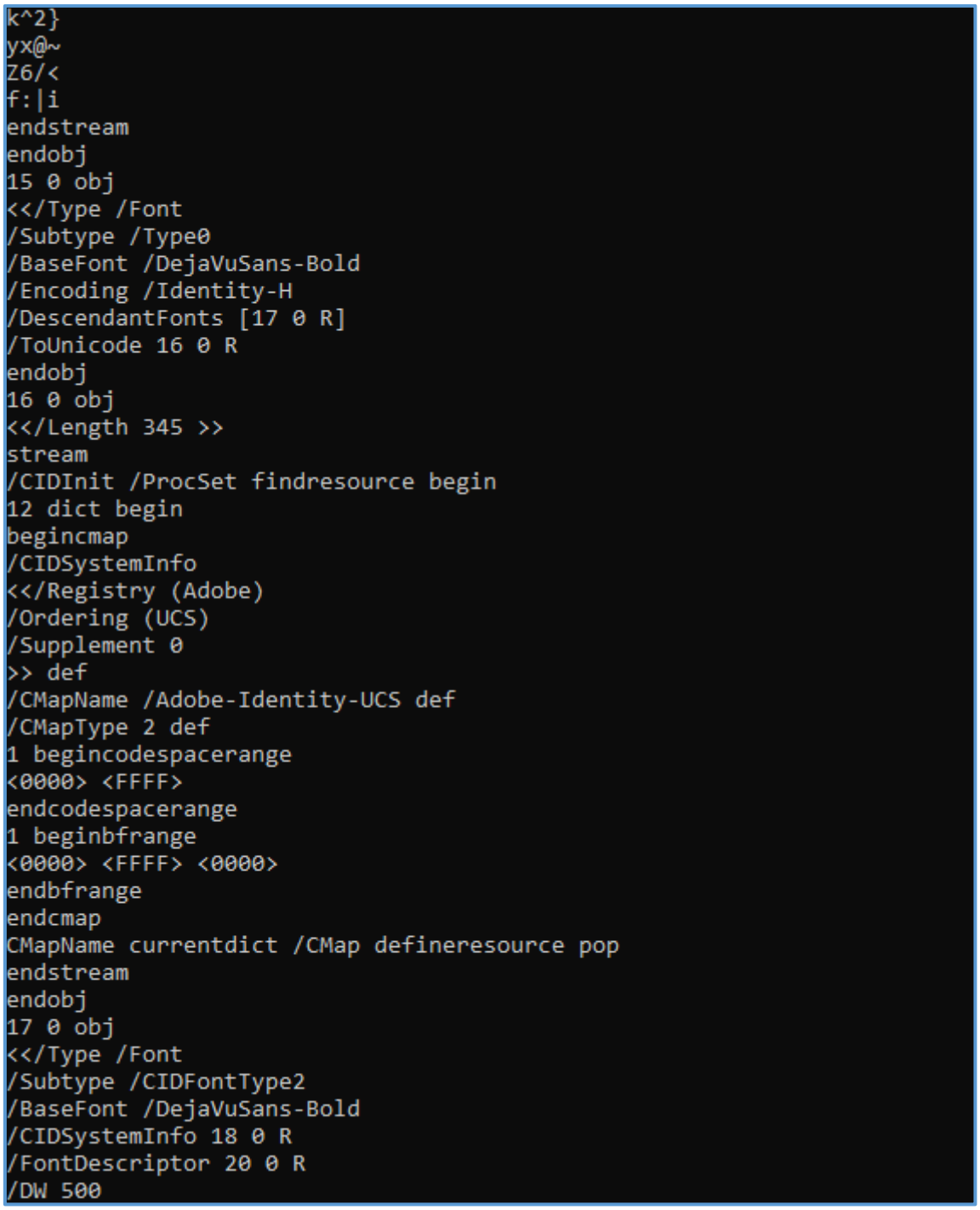
Şekil 5.3 "maas-zam" Zararlısı

Şekil 5.3'te gösterilen ekran görüntüsü bize dosyanın bir ".exe" dosyası olduğunu ancak bir ".pdf" belgesi olarak göründüğünü açıkça gösteriyor (Yara kuralı yazarken bu bilgiyi kullanacağız). Bu dosyanın Hash değerini çeşitli araçları kullanarak hesaplayabilir ve VirusTotal gibi platformlarda hit olup olmadığını kontrol edebiliriz.

Dosyanın hash değerinin hesaplamasını yaptıktan sonra yara kuralı oluşturmak için zararlı dosya içerisinde benzersiz stringleri ayrıştırmamız gerekmektedir. Bu işlemi yapabilmek için bölüm 4'te açıklanan statik analiz teknikleri yanı sıra Microsoft tarafından geliştirilen "string.exe" aracı kullanılabilir. Bu araç dosya içerisindeki tüm string değerlerini çıkarır. Devamında dosyaya özel olan alanların gözle inceleyerek ayrıştırılması kalır. Bu örneğimizde yer alan dosyanın "string.exe" aracı tarafından verilen çıktısı Şekil 5.4 ve Şekil 5.5'te gösterilmiştir.



Şekil 5.4 Zararlı Yazılım String Çıktısı



Şekil 5.5 Zararlı Yazılım String Çıktısı

Şekil 5.4 ve Şekil 5.5'te elde edilen string değerler içerisinden benzersiz olabilecek değerler ayrıştırılır ve kural yazmak için kullanılır. En basit şekilde bir yara kuralı bu şekilde yazılabilir.

5.3. Örnek Yara Kuralları

Burada iki farklı örnek yara kuralı gösterilmektedir;

```
rule {kural_ismi}
{
        meta:
                author:
                description:
        string:
                $a:     “string1” nocase #case sensitive NO
                $b:     “string2” wide #unicode string
                $c:     “str” wide ascii  #2 farklı karakter dizesini aramak için wide kullanılır
                $d:     “thycdc” fullword
        condition:
                $a or $b
}


rule detect_calculator
{
        meta:
                description = “Calculator” nocase
                author = “mustafa”
        strings:
                $a = {43 61 6c 63 75 6c 61 74 6f 72 53 74 61 72 74 65 64}
                $b = “Calculator”
        condition:
                $a or $b
}
```

6. SONUÇ VE ÖNERİLER

“4.2.1. Yapılan Zararlı Yazılım Analizine Ait Yara Kuralı” başlığında çalışma içerisinde örnek olması amacıyla geliştirilen yara kuralı paylaşılmıştır. Yapılan çalışma tamamında buna benzer 300’den fazla yara kuralı sektör genelinde araştırılarak derlenmiştir. Bu kurallar çalışmaya ek olarak EK1’de sunulmuştur. Bu çalışma sayesinde kurum ve kuruluşlarda çalışan siber güvenlik analistleri kendi çalıştıkları kurumlarda zararlı yazılım tespiti yapabilmesi için hızlı ve pratik bir yöntem önerilmiştir.

Aynı zamanda “3.4. Manuel Bir Olay Müdahalesi Örnek Çalışması” başlığında geliştirilen script sayesinde kurum genelinde bir olay müdahale çalışmasına destek olunması hedeflenmiştir. Buna ek olarak Linux sistemlerde manuel adli bilişim incelemeleri yapılabilmesi için bir script geliştirilmiştir. Bu sayede lisans maliyeti olmadan Linux sistemlerde hızlı analiz yapılabilmektedir.

Geliştirilen Yara kuralları ile kuruluşlarda zararlı tespit yöntemlerine katkı sağlanmıştır. Nasıl Yara geliştirileceği anlatılarak bu çalışmanın sürekliliği sağlanmıştır. Literatürde yapılan araştırma sonucunda elde edilen diğer kurallar gruplandırılmış ve bu çalışmada ek olarak sunulmuştur. Bu yöntem sayesinde kurum genelinde, platform bağımsız her türlü işletim sistemi üzerinde zararlı yazılım varlığının tespiti kolaylıkla yapılabilmektedir.

6.1. Tez Çıktılarının Literatüre Katkıları

Bu çalışma sonucunda elde edilen YARA kuralları kullanılarak her şirket veya kuruluş kendi bünyesinde sıkça kullanılan zararlı yazılımların varlığını pratik bir şekilde tespit edebilecektir. Aynı zamanda yeni karşılaştığı bir zararlı yazılımı YARA ile nasıl tespit edebileceğini anlayacaktır. Burada elde edilen YARA kuralları kullanılarak sistemler geriye dönük taranabilecek ve olası var olan bir yetkisiz erişim tespiti kolaylaşacaktır. Kurum ve kuruluşların kullandığı ve yüksek maliyet gerektiren zararlı yazılım analizi yapabilen çözümler yerine bu bütçeyi ayırmadan da bu çalışmalar yapılabilecektir. Veya bu çözüm diğer kurumsal çözümlere destek mahiyetinde kullanılabilecektir. Bu çalışma yapılırken araştırılan saldırı tekniklerinin tespiti için web genelinde yara kuralları araştırılmıştır. Toparlanan kurallar gruplanarak Ek-1’de sunulmuştur.

6.2. Öneriler

Buradan yola çıkılarak zararlı yazılımların tespiti için otomatik bir yara kural oluşturabilen bir araç geliştirilebilir. Bu araç sayesinde süreçler daha da hızlandırılabilir. Ancak bu aracın çıktılarında oluşabilecek hataların en aza indirgenmesi için detaylı çalışma gerekecektir.

KAYNAKLAR

[1] Derbyshire, R., Green, B., Prince, D., Mauthe, A., Hutchison, D., (2018). An Analysis of Cyber Security Attack Taxonomies, 2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), 2018, 153-161, doi:10.1109/EuroSPW44102.2018

[2] Çelik, S., (2018). Siber Uzay ve Siber Güvenliğe Multidisipliner Bir Yaklaşım, Dergipark, 1(2), 110-119

[3] Doğan, D., (2021). Siber Güvenlik Açısından Siber Saldırı Senaryolarının İncelenmesi, Yüksek Lisans Tezi, Maltepe Üniversitesi, Lisansüstü Eğitim Enstitüsü.

[4] Chen. J., Su C, Yeh, K-H., Yung, M., (2018). Special issue on advanced persistent threat. vol 79. Elsevier Future Generation Computer Systems, Volume 79, Part 1, February 2018, 243-246, doi:10.1016/j.future.2017.11.005

[5] Niakanlahiji, A., Wei, J., Chu, B-T A., (2018). natural language processing based trend analysis of advanced persistent threat techniques. In: 2018 IEEE International Conference on Big Data (Big Data), 2018, 2995-3000

[6] Ahmad, A., Webb, J., Desouza. KC., Boorman, J., (2019). Strategically-motivated advanced persistent threat: Definition, process, tactics and a disinformation model of counterattack. Computers & Security 86:402-418

[7] Auty, M., (2015). Anatomy of an advanced persistent threat. Elsevier Network Security, Volume 2015, Issue 4, April 2015, 13-16, doi:0.1016/S1353-4858(15)30028-3

[8] Riza, A., Yusof, R., Udzir, N., Selamat, A., (2019). Systematic literature review and taxonomy for DDoS attack detection and prediction, January 2019, International Journal of Digital Enterprise Technology, 1(3):292, doi:10.1504/IJDET.2019.097849

[9] Yin, D., Zhang, L., Yang, K., (2018). DDoS Attack Detection and Mitigation With Software-Defined Internet of Things Framework, IEEE Access, 6, 24694 – 24705, doi: 10.1109/ACCESS.2018.2831284

[10] Chiew, K., Yong, K., Tan C., (2018). A survey of phishing attacks: Their types, vectors and technical approaches, Elsevier, 2018, 106, 1-20, doi:10.1016/j.eswa.2018.03.050

[11] Kiran D. T., Sunil, N. P., (2021). Different Types of Phishing Attacks and Detection Techniques: A Review

[12] Page, S., Jourdan, G-V., Bochman, G., Flood, J., Onut I-V., (2018). Using URL shorteners to compare phishing and malware attacks, PhD Thesis, University of Ottawa

[13] Pirscoveanu, R., Hansen, S., Larsen, T., Stevanovic, M., Pedersen, J., Czech, A., (2015). Analysis of Malware behavior: Type classification using machine learning, 2015 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), 31, 1-7, doi: 10.1109/CyberSA.2015.7166115

[14] Tahir, R., (2018). A Study on Malware and Malware Detection Techniques, I.J. Education and Management Engineering, 2018, 2, 20-30 Published Online March 2018 in MECS (http://www.mecs-press.net), doi: 10.5815/ijeme.2018.02.03

[15] Zolkipli, M., Jantan, A., (2010). Malware Behavior Analysis: Learning and Understanding Current Malware Threats, 2010 Second International Conference on Network Applications, Protocols and Services, 2010, 2, 222 - 227, doi: 10.1109/NETAPPS.2010.46

[16] Kara, İ., (2019). A Basic Malware Analysis Method, Elsevier, Volume 2019, Issue 6, June 2019, Pages 11-19, doi:10.1016/S1361-3723(19)30064-8

[17] Hong, J., (2012). The State of Phishing Attack, January 2012, Communications of the ACM 55(1):74-81, doi:10.1145/2063176.2063197

[18] Mallik, A., Ahsan, A., Shahadat, M., Tsou, J., (2018). Man-In-The-Mıddle-Attack: Understandıng In Sımple Words, January 2019, International Journal of Data and Network Science 3(2):77-92, doi:10.5267/j.ijdns.2019.1.001

[19] Dougan, T., Curran, K., (2012). Man-In-The-Browser Attack, International Journal of Ambient Computing and Intelligence (IJACI), IGI Global, vol. 4(1), pages 29-39

[20] Stiawan, D., Suryani, M., Susanto, Idris, M., Aldalaien, M., Alsharif, N., Budiarto, R., (2021). Ping Flood Attack Pattern Recognition Using a K-Means Algorithm in an Internet of Things (IoT) Network, IEEE Access, Volume 9, 116475 – 116484, doi:10.1109/ACCESS.2021.3105517

[21] Kumar, S., (2007). Smurf-based Distributed Denial of Service (DDoS) Attack Amplification in Internet, Second International Conference on Internet Monitoring and Protection (ICIMP 2007), 2007, 36, 25, doi: 10.1109/ICIMP.2007.42

[22] Alzahrani, Sabah, M., (2021). Buffer Overflow Attack and Defense Techniques, International Journal of Computer Science & Network Security, 2021, Volume 21, 207-212, doi:10.22937/IJCSNS.2021.21.12.30

[23] Maithili K., Vinothkumar V., Latha P., (2018). Analyzing the Security Mechanisms to Prevent Unauthorized Access in Cloud and Network Security, June 2018, Journal of Computational and Theoretical Nanoscience 15(6), 2059-2063, doi:10.1166/jctn.2018.7407

[24] Salahdine, F., Kaabouch, N., (2019). Social Engineering Attacks: A Survey, Future Internet 2019, 11(4), 89, doi:10.3390/fi11040089

[25] Amiel, F., Villegas, K., Feix, B., Marcel, L., (2007). Passive and Active Combined Attacks: Combining Fault Attacks and Side Channel Analysis, Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC 2007), 2007, 41, 92 - 102, doi: 10.1109/FDTC.2007.12

[26] Morris, T., Gao, W., (2013). Industrial Control System Cyber Attacks, September 2013, Conference: 1st International Symposium for ICS & SCADA Cyber Security Research 2013 (ICS-CSR 2013), doi:10.14236/ewic/ICSCSR2013.3

[27] Duo, W., Zhou, M., Abusorrah, A., (2022). A Survey of Cyber Attacks on Cyber Physical Systems: Recent Advances and Challeng, IEEE/CAA Journal of Automatica Sinica, May 2022, 9(5), 784 - 800, doi: 10.1109/JAS.2022.105548

[28] Blaise, A., Bouet, M., Conan, V., Secci, S., (2020). Detection of zero-day attacks: An unsupervised port-based approach, Elsevier Computer Networks, Volume 180, 24 October 2020, 107391, doi:10.1016/j.comnet.2020.107391

[29] Kim, J., Bu, S., Cho, S., (2018). Zero-day malware detection using transferred generative adversarial networks based on deep autoencoders, Elsevier Information Sciences, September 2018, Volumes 460–461, Pages 83-102, doi:10.1016/j.ins.2018.04.092

[30] Conti, M., Dragoni, N., Lesyk, V., (2016). A Survey of Man In The Middle Attacks, IEEE Communications Surveys & Tutorials, March 2016, 18(3),1 , doi:10.1109/COMST.2016.2548426

[31] Ornaghi, A., Valleri, M., (2003). Man In The Middle Attacks, Blackhat Conference, Europe, 2016

[32] Reith, M., Carr, C., Gunsch, G., (2002). An examination of digital forensic models. International Journal of Digital Evidence 1 (3), 1-12

[33] Henkoğlu, T., (2020). Adli bilişim: Dijital delillerin elde edilmesi ve analizi. Pusula Yayıncılık, Türkiye

[34] Schneier, B., (2014). The future of incident response, IEEE Security & Privacy, 12 (5), 96

[35] Dunsin, Dipo, Ghanem, Mohamed and Ouazzane, Karim (2022). The Use of Artificial Intelligence in Digital Forensics and Incident Response (DFIR) in a Constrained Environment, International Conference on Digital Forensics and Security of Cloud Computing ICDFSCC, May 17-18, 2022, Australia

[36] Granadillo G., González-Zarzosa, S., Diaz, R., (2021). Security Information and Event Management (SIEM): Analysis, Trends, and Usage in Critical Infrastructures, July 2021, Sensors, 21(14), 4759, doi:10.3390/s21144759

[37] Werlinger R., Muldner K., Hawkey K., Beznosov K., (2010), Preparation, detection, and analysis: the diagnostic work of IT security incident response, Information Management & Computer Security, 18(1), 26-42, doi:10.1108/09685221011035241

[38] Akbari, Y., Al-maadeed, S., Elharrous, O., Khelifi, F., Lawgaly, A., Bouridane, A., (2022). Digital forensic analysis for source video identification: A survey, Elsevier Forensic Science International: Digital Investigation, Volume 41, June 2022, doi:10.1016/j.fsidi.2022.301390

[39] Ahmad, A., Hadgkiss, J., Ruighaver, A.B., (2012). Incident response teams – Challenges in supporting the organisational security function, Elsevier Computers & Security, Volume 31, Issue 5, July 2012, 643-652, doi:10.1016/j.cose.2012.04.001

[40] Naseer, A., Naseer, H., Ahmad A., Maynard, S., Siddiqui A., (2021). Real-time analytics, incident response process agility and enterprise cybersecurity performance: A contingent resource-based analysis, Elsevier International Journal of Information Management, Volume 59, August 2021, doi:10.1016/j.ijinfomgt.2021.102334

[41] Kamal, M., Davis, A., Nabukenya, J., Schoonover, T., Pietron, L., Vreede, G., (2007). Collaboration Engineering For Incident Response Planning: Process Development and Validation, 2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07), 2007, 8, 15, doi:10.1109/HICSS.2007.128

[42] Lee J., Lee S., (2015). A Study on Unknown Malware Detection using Digital Forensic Techniques -21, Journal of the Korea Institute of Information Security & Cryptology, 2014, Volume 24 Issue 1, 107-122, doi:10.13089/JKIISC.2014.24.1.107

[43] Duby, A., Taylor, T., Zhuang, Y., (2022). Malware Family Classification via Residual Prefetch Artifacts -22, 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), Jan 2022, 256-259, DOI: 10.1109/CCNC49033.2022

[44] Naik, N., Jenkins, P., Savage, N., Yang, L., Naik, K., Song, J., Boongoen, T., Iam-On, N., (2020) Fuzzy Hashing Aided Enhanced YARA Rules for Malware Triaging -23, 2020 IEEE Symposium Series on Computational Intelligence (SSCI), 2020, 3, 1138 - 1145, doi:10.1109/SSCI47803.2020.9308189

[45] Christopher S. Culling, (2018), Which YARA Rules Rule: Basic or Advanced, Sans Whitepaper, August 2018

ÖZGEÇMİŞ

Mustafa Emre DEMİR

[redacted]