Otomatik Analiz Sistemlerini Atlatma ve Alınabilecek Olası Önlemler

Bu tez Bilgi Güvenligi Mühendisliği'nde
Tezli Yüksek Lisans Programının bir koşulu olarak

Alican Akyol

tarafından

Fen Bilimleri Enstitüsü'ne
sunulmuştur.

İSTANBUL
ŞEHİR
ÜNİVERSİTESİ

Bu tezi okudu[illegible] kapsam ve nitelik açısından Bilgi Güvenliği Mühendisliği alanında Yüksek Lisans [illegible] için tümüyle uygun olduğu görüşüne vardık.

ONAYLAYANLAR:

Osman Pamuk ..... (Tez Danışmanı)

Prof. Dr. Erkan Türe .....

Yrd. Doç. Dr. Selçuk Baktır .....

Bu tez İstanbul Şehir Üniversitesi, Fen Bilimleri Enstitüsü tarafından belirlenen tüm koşullara uygundur.

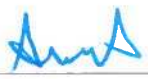
ONAYLANMA TARİHİ: 2 Haziran 2016

MÜHÜR

Yazarlık Beyanı

Ben, Alican AKYOL, başlığı 'Otomatik Analiz Sistemlerini Atlatma ve Alınabilecek Olası Önlemler' olan tezin ve içinde sunulan bilgilerin şahsıma ait olduğunu beyan ederim. Ayrıca:

- Bu çalışmanın bütünü veya esası bu üniversitede Yüksek Lisans derecesi elde etmek üzere çalıştığım süre içinde gerçekleştirilmiştir.
- Daha önce bu tezin herhangi bir kısmı başka bir derece veya yeterlik almak üzere bu üniversiteye veya başka bir kuruma sunulduysa bu açık biçimde ifade edilmiştir.
- Başkalarının yayınlanmış çalışmalarına başvurduğum durumlarda bu çalışmalara açık biçimde atıfta bulundum.
- Başkalarının çalışmalarından alıntıladığımda kaynağı her zaman belirttim. Tezin bu alıntılar dışında kalan kısmı tümüyle benim kendi çalışmamdır.
- Esaslı yardım aldığım bütün kaynaklara teşekkür ettim.
- Tezde başkalarıyla birlikte gerçekleştirilen çalışmalar varsa onların katkısını ve kendi yaptıklarımı tam olarak açıkladım.

İmza: 

Tarih: 02.06.2016

Otomatik Analiz Sistemlerini Atlatma ve Alınabilecek Olası Önlemler

Alican Akyol

Öz

Her geçen gün artan zararlı yazılım türü ve sayısı ile etkin bir mücadele için otomatik analiz sistemlerinin kullanımı büyük önem arz etmektedir. Otomatik analiz sistemleri ile şüpheli dosyalar güvenli ve kontrollü bir ortamdan çalıştırılarak zararlı yazılım olup olmadıkları ve eğer zararlı yazılım iseler ne tür özelliklere sahip oldukları gibi bilgilere hızlıca erişile bilinmektedir. Diğer taraftan zararlı yazılım yazarları da hazırladıkları uygulamaların, daha çok sisteme bulaşma ve bulaştığı sistemde daha uzun süre kalabilme adına otomatik analiz sistemleri tarafından tespit ve analiz edilebilme riskini azaltmaya çalışmaktadırlar. Bunun için zararlı yazılım yazarları, çeşitli otomatik analiz sistemi tespiti yöntemleri kullanmakta ve hazırladıkları zararlı yazılım eğer bu tür bir sistemde çalışıyor ise normalden farklı davranış sergileyerek analiz sistemini aldatmaya ve atlatmaya çalışmaktadırlar. Bu sebeple, zararlı yazılımlar ile mücadele de önemli bir rolü olan otomatik analiz sistemlerini etkisiz hale getiren ve devamlı olarak güncellenen otomatik analiz sistemi tespit yöntemlerine karşı yeni çözüm yöntemlerinin geliştirilmesi gerekmektedir. Bu ihtiyaç doğrultusunda da, bu çalışmamızda, otomatik analiz sistemi tespit yöntemleri ile mücadele yöntemleri araştırılmıştır. Bu bağlamda, otomatik analiz sistemi tespit etme yöntemleri araştırılmış ve bu yöntemlerin kullanımı herkese açık otomatik analiz sistemlerinde test edilerek, otomatik analiz sistemlerinin bu yöntemlere karşı yeterli düzeyde önlem almadığı tespit edilmiştir. Otomatik analiz sistemlerini tespit etme yöntemlerini engellemek için mevcut teknikler incelenmiş ve bunlara ilaveten yeni teknikler geliştirilmiştir. Bu tekniklerinin kullanılarak yapılan testlerde, otomatik analiz sistemi sonuçlarının önemli ölçüde iyileştirilebildiği ispatlanmıştır.

Anahtar Sözcükler: Zararlı Yazılım, Kum Havuzu, Otomatik Analiz Sistemi, Sanallaştırma Ortamı, Siber Güvenlik

Teşekkür

Tez çalışmamın başından sonuna kadarki tüm süreçte sabırla bana yardımcı olan ve ufkumu açan tez danışmanım Sayın Osman Pamuk' a teşekkür ederim.

Tezimin çeşitli yerlerinde teknik ve akademik konularda benden yardımlarını esirgemeyen başta Gökhan Alkan ve Abdurrahman Pektaş olmak üzere değerli iş arkadaşlarıma teşekkürü borç bilirim.

Son olarak, çalışma hayatım boyunca her zaman yanımda olan çok kıymetli eşim Esra' ya sevgilerimi sunarım.

İçindekiler

Şekil Listesi

Tablo Listesi

Kısaltmalar

API	**A**pplication **P**rogramming **I**nterface
APT	**A**dvanced **P**ersistent **T**hreat
CPU	**C**entrak **P**rocessing **U**nit
CPUID	**C**PU **I**dentification
DDOS	**D**istrubuted **D**enial **O**f **S**ervice
DLL	**D**ynamic **L**ink **L**ibrary
HTML	**H**ypertext **M**arkup **L**anguage
IBM	**I**nternational **B**usiness **M**achines
JSON	**J**avascript **O**bject **N**otation
LTS	**L**ong **T**erm **S**upport
MAC	**M**edia **A**ccess **C**ontrol
MAEC	**M**alware **A**ttribute **E**numeration and **C**haracterization
MHTML	**M**IME **H**TML
MIME	**M**ultipurpose **I**nternet **M**ail **E**xtensions
PEB	**P**rocess **E**nvironment **B**lock
SP2	**S**ervice **P**ack 2
XML	**E**xtensible **M**arkup **L**anguage

Bölüm 1

Giriş

Teknoloji ve internet dünyasının hızlı gelişimi hayatımıza getirdiği kolaylıkların yanında güvenlik açıklıklarını da beraberinde getirmektedir.Ev kullanıcılarından kurumsal ağ kullanıcılarına ve hatta devletlere kadar birçok kişi ve kurum bu durumdan olumsuz olarak etkilenmektedir. Zararlı yazılımlar da bu ortam içerisinde büyük bir tehlike haline gelmektedir.

Zararlı yazılım sayısının ve zararının artması bunlardan korunma ihtiyacını da artırmıştır. Zararlı yazılımlardan korunabilmek için zararlı yazılımların tespitinin gerçekleştirilmesi ve bunların engellenmesi gerekmektedir. Zararlı yazılımlar, zararlı yazılım analizi ile tespit edilebilmekte ve engelleme ve bulaştıkları sistemlerden temizlenme yöntemleri belirlenebilmektedir.

Zararlı yazılım analizi şüphelenilen dosyanın ağ, işletim sistemi üzerindeki davranışlarını, genel özelliklerini inceleyerek çıkarımlarda bulunmayı ve bunun sonucunda sistemlere bulaşmasını engellemeyi ve bulaştığı sistemlerden temizlenmesine yardımcı olmaktadır.

Zararlı yazılım sayısının çok fazla olması bütün dosyaların el ile analizini neredeyse imkansız hale getirmektedir. Bu sebepten dolayı zararlı yazılım analizini hızlı ve otomatik olarak gerçekleştirilebilmesi için otomatik zararlı yazılım analiz sistemleri geliştirilmiştir. Otomatik zararlı yazılım analiz sistemleri, şüphelenilen dosyaların analizini gerçekleştirerek dosya hakkında bilgi vermekte ve ilk temel incelemeleri, statik ve dinamik analizleri gerçekleştirebilmektedir.

Otomatik zararlı yazılım analiz sistemleri ile şüphelenilen dosyaların zararlı olup olmadığı hakkında çıkarımlarda bulunulabilmektedir. Bu sayede zararlı dosyalar ayırt edilebilmekte ve yapılan çıkarımların yardımı ile bunların el ile analizi gerçekleştirilebilmektedir. Ayrıca otomatik analiz sistemleri ile 0. gün, APT gibi daha öncesinden bilinmeyen tehditler da tespit edilebilmektedir.

Otomatik analiz sistemlerinin zararlı yazılım analizindeki yeri ve öneminin artmış olması zararlı yazılım yazarları tarafından bilinmektedir. Zararlı yazılım yazarları, uygulamalarının analiz edilmesini engellemek için analiz sistemlerine yönelik önlemler almışlardır. Aldıkları önlemler ile zararlının çalıştığı ortamın analiz sistemi olup olmadığını kontrol ederek uygulamanın analiz sisteminde çalışmasını engellemekte ya da aslından farklı davranışlarda bulunmasını sağlamaktadır. Böylelikle şüpheli dosya hakkında doğru bir sonuç elde edilememektedir.

Zararlı yazılımların otomatik analiz sistemlerinde analiz edilmelerini engellemeye yönelik aldıkları önlemleri atlatmak için analiz sistemleri tarafından bazı yetenekler geliştirilerek analiz sonuçlarının doğruluğu sağlanmaya çalışılmaktadır. Fakat zararlı yazılımların yeni yöntemler geliştirmeleri ile alınan bu önlemler etkisini kaybetmiştir. Bunun sonucu olarak otomatik analiz sistemlerinin doğru analiz sonuçları vermeleri için zararlı yazılımların uygguladıkları analiz sistemi engelleme yöntemlerine karşı yeni tekniklere ihtiyaç duyulmaktadır.

Bu çalışmada, zararlı yazılımların analiz sistemi tespiti için kullandıkları yöntemler atlatılarak otomatik analiz sistemlerinin doğru ve etkili sonuçlar verebilmeleri amaçlanmıştır. Bu sebeple öncelikle zararlı yazılımın tanımı, türleri ve zararlı yazılım analizi anlatılmaktadır. 3. bölümde, otomatik zararlı yazılım analiz sistemi olan kum havuzundan bahsedilmekte ve bu çalışma içerisinde kullanılan kum havuzları listelenmektedir. 4. bölümde, kum havuzlarının tespiti için kullanılabilecek yöntemler anlatılmaktadır. 5. bölümde, analiz ortamı tespit etme araçlarından bahsedilmektedir. Ayrıca bu çalışma içerisinde kullanılan kum havuzları bu araçlarla test edilerek elde edilen sonuçlar tablolar şeklinde gösterilmektedir. 6. bölümde, kum havuzlarını tespit etmek için geliştirilen yöntemleri atlatmak için var olan ve yeni geliştirilen teknikler anlatılmaktadır. Bu teknikler ile kum havuzlarını tespit etmek için kullanılan yöntemlerin başarılı bir şekilde atlatıldığı gösterilmektedir. 7. bölümde, sanallaştırma ortamlarının tespitinin atlatılması için kullanılabilecek yöntemlerle oluşturulan sanal makine ile normal bir sanal makinede analiz

edilen zararlı yazılımların analiz sonuçları karşılaştırılmıştır. 8. Ve son bölümde ise çalışmanın genel anlatımı yapılmakta ve neler yapılabileceği hakkında öneriler sunulmaktadır. Analiz sistemi tespit yöntemlerini atlatmak için kullanılabilecek teknikler ile gerçekleştirilen testlerin sonuçları gösterilmektedir.

Bölüm 2

Zararlı Yazılım

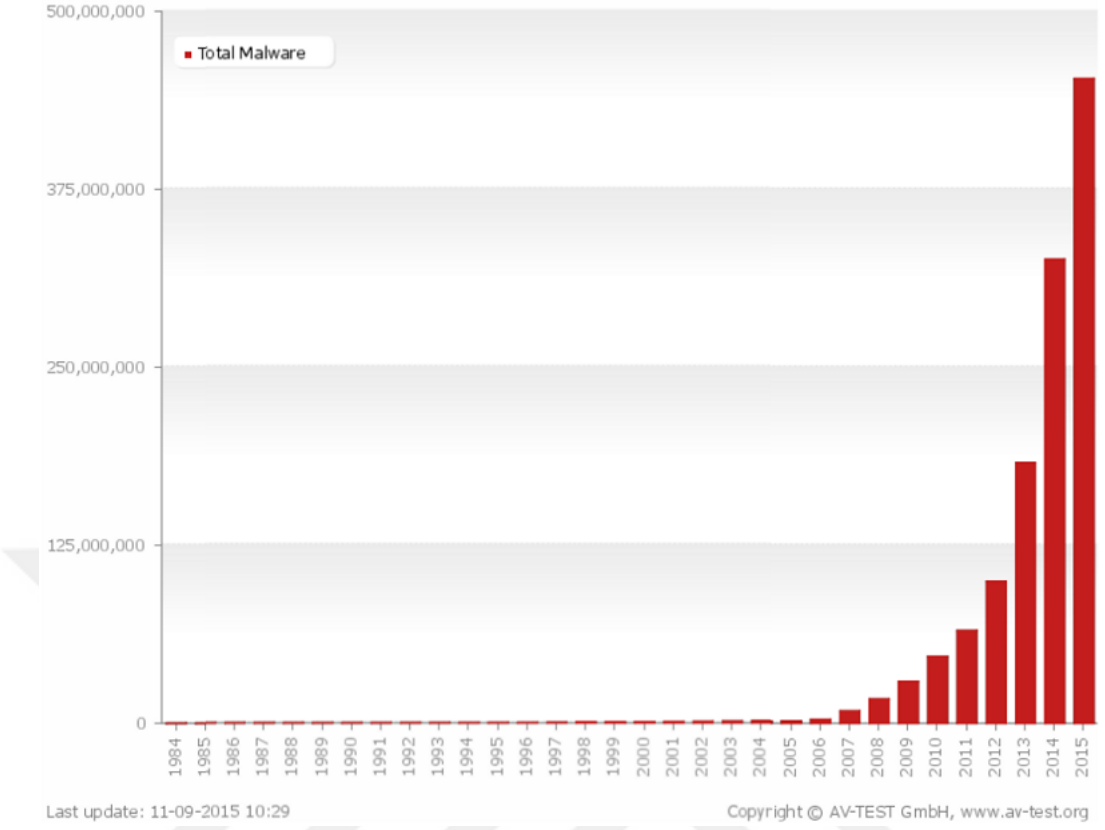
2.1 Zararlı Yazılım Nedir?

Teknolojik sistemlere zarar veren, çalışmalarını değiştiren veya etkileyen, bilgi çalan ya da bilgide değişiklik yapan, kullanıcılara zarar veya rahatsızlık verecek davranışlarda bulunan uygulamaların tümüne zararlı yazılım (kötücül yazılım) denmektedir. Zararlı yazılımlar birçok programlama dili ile yazılabildiği gibi betik dilleri ile de yazılabilmektedir.

Zararlı yazılımların kişilere verdiği ya da vereceği zararlar arasında banka hesaplarını ele geçirme, para çalma, bilgisayarlara sızma ve özel bilgileri çalma, web kamerasından görüntü elde etme, ses kaydı yapma gibi istenmeyen birçok davranış gösterilebilir. Devletlere verdiği ya da vereceği zararlar arasında ise gizli belgeleri ele geçirme, uçak, tren, savaş araçları gibi araçların kontrolünü ele geçirip bunları dilediği gibi yönlendirme, su, elektrik, kanalizasyon gibi sistemlerin kontrolüne hakim olup bu sistemlerde istediği aktiviteyi yapma gibi felaketler gösterilebilir.

Günümüzde teknolojinin hızla gelişmesi, hayatın her alanında büyük bir yer edinmesi ve birçok işlemin teknoloji ile yapılmasından dolayı zararlı yazılımların sayısı da çoğalmıştır. Şekil 2.1' de 1984 yılından 2015 yılının Eylül ayına kadar olan günlük yeni zararlı yazılım sayısı gösterilmektedir [1].

Statistics.png



Şekil 2.1: Yıllara Göre Günlük Yeni Zararlı Yazılım İstatistiği.

2.2 Zararlı Yazılım Çeşitleri

2.2.1 Virus

Çalıştırılabilir dosyalarda istenmeyen ya da beklenmeyen değişiklikler yapan yazılımlara virüs denmektedir. Virüslerin, bulaştıkları bilgisayarlarda aktivitelerini gerçekleştirebilmeleri için çalıştırılmaları gerekmektedir.

2.2.2 Solucan (Conficker)

Solucanlar, kullanıcılara ihtiyaç duymadan kendilerini kopyalayarak ya da yayarak başka bilgisayarlara bulaşırlar. Bulaştıkları bilgisayarlarda istedikleri gibi hareket ederek farklı alanlara erişebilirler ve buralarda istenmeyen ya da beklenmeyen etkiler yapabilirler [2].

- Kullanıcıların dosyaları, e-postalarındaki mailleri gibi özel verilerini alıp başkalarına gönderebilirler.

- Bilgisayarlara uzaktan bağlantı oluşturabilirler.
- Bilgisayarın bulunduğu ağdaki diğer bilgisayarlara da bulaşarak onları da etki altına alabilirler.
- Gereksiz internet bağlantıları oluşturarak internetin yavaşlamasına neden olurlar.
- Bilgisayarda çok fazla kopyalama yaparak belleği doldurabilirler.

2.2.3 Truva Ati (Trojan)

Normal bir bilgisayar programı olarak görünen ve işlem gerçekleştiren fakat arka planda zararlı içeriğini faaliyete geçiren bir zararlı yazılım türüdür. İsminden de anlaşılacağı üzere zararlı programı, kendi içinde gizli bir şekilde barındırmaktadır. Bilgisayarda faaliyet gösterebilmeleri için çalıştırılmaları gerekmektedir. Truva atları yamalı programlarda fazlaca olmaktadır [3].

2.2.4 Reklam Zararlı Yazılımı (Adware)

Otomatik reklam gösterimi gerçekleştirerek kullanıcıları rahatsız eden yazılımlardır. Reklam zararlı yazılımları, internet sitelerinde birden ortaya reklam çıkarmakta ya da sayfanın tamamını kaplayan reklam göstermektedir. Bu zararlı yazılım yazarlarının, büyük bir kısmı reklam şirketleri tarafından desteklenmektedir [3].

2.2.5 Bot

Adını robot kelimesinden alan bot zararlı yazılımları, otomatik olarak özel işlemler gerçekleştiren programlardır. Daha çok DDOS saldırıları gerçekleştirmek için kullanılan bot zararlıları, bulaştıkları sistemlerdeki dosyaları, belirlenmiş sunuculara aktarma işlemini de yapabilmektedirler [4].

2.2.6 Fidyeci (Ransomware)

Fidyeci zararlı yazılımlar, bulaştıkları sistemlerdeki dosyaları şifreleme, disk bölümlerini şifreleme gibi kötücül işlemler yapıp bunların çözümü için kullanıcılardan fidye talep eden programlardır [5].

2.2.7 Arka Kapı (Backdoor)

Truva atı ve solucanlara benzeyen arka kapı zararlıları, bulaştıkları sistemlerde bir arka kapı açarak buradan internet bağlantısı sağlamaktadır. Bu bağlantı ile zararlı yazılım yazarı dilediği faaliyetleri gerçekleştirebilmektedir [3].

2.2.8 Tarayıcı Gaspçısı (Browser Hijacker)

İstenmeyen bir şekilde tarayıcıların ana sayfalarını, arama adreslerini değiştiren zararlı yazılımlara tarayıcı gaspçısı zararlı yazılımı denmektedir. Bu zararlılar, tarayıcılara enjekte olmuş bir şekilde çalışmaktadır. Reklam gösterme gibi rahatsız edici aktiviteler gerçekleştirse de yapabilecekleri en kötü işlem, internet bankacılığı işlemlerinde kullanıcı bilgilerini çalabilmektir.

2.2.9 Rootkit (Kok Kullanici)

Bilgisayarda çalışan işlemleri, sistem ile ilgili bilgileri, dosyaları işletim sisteminden saklayan rootkitler, kendilerini gizleyerek çalışmalarını gerçekleştirirler. Kök kullanıcı adı da verilen rootkitler, yetkili olmayan kullanıcıların sistem bilgilerine, yönetim uygulamalarına erişimlerini engelleme niyetiyle oluşturulmasına rağmen zararlı yazılım yazarları tarafından kötü niyetli rootkitler de oluşturulmuştur [3].

2.2.10 Tus Kaydedici (Keylogger)

Bulaştıkları sistemlerde, kullanıcıların klavyeden girdikleri verileri toplayan ve bunu saldırgan şahsa ileten zararlı yazılımlardır. Genel olarak şifre ele geçirmek için oluşturulmaktadırlar [3].

2.2.11 Casus Yazilim (Spyware)

Kullanıcıların bilgisi dışında kullanıcılara ait bilgi ve belgeleri ele geçirmek için oluşturulmuş yazılımlardır. Casus yazılımlar, bulaştıkları sistemlerde sayısız pencere açma, tarayıcılarda kullanıcı isteği dışında farklı araç çubukları kurma ve ana sayfayı değiştirme gibi aktivitelerde bulunabilirler [3].

2.3 Zararlı Yazılım Analizi

Zararlı yazılım analizi yöntemleri, genel olarak, statik analiz ve dinamik analiz olarak iki ana kategori altında değerlendirilebilir.

2.3.1 Statik Analiz

Şüpheli dosyanın çalıştırılmadan incelenme süreci statik analiz olarak adlandırılmaktadır. Statik analiz sürecinde analiz edilecek dosyanın özet değeri, boyutu, içerisinde yer alan katar ifadeleri ve kullandığı fonksiyonlar gibi dosya bilgileri incelenmekte ve tersine mühendislik yöntemleri kullanılmaktadır. Statik analiz süreci kendi içerisinde temel statik analiz ve ileri derece statik analiz olmak üzere 2'ye ayrılabilir. Özet değeri ve boyut bilgisi alma, dosya içerisinde geçen ifadelerin ve kullanılan fonksiyonların çıkartılması gibi işlemler basit statik analiz süreci içerisinde yer alırken tersine mühendislik yöntemleri ile analiz etme gibi işlemler ise ileri seviye statik analiz süreci içerisinde yer almaktadır. İleri derece işletim sistemi ve programlama bilgisi gerektiren tersine mühendislik yöntemleri ile dosyanın analizine ilişkin ayrıntılı bilgiler elde edilebilmektedir.

Statik analizde dosyanın anormal çalışması durumunda gerçekleştireceği davranışlar incelenebilirken dinamik analizde ise dosya çalıştığında hangi davranışlarda bulunuyorsa sadece bunlar incelenebilmektedir. Statik analiz, dinamik analize göre daha yavaştır ve dosyanın gerçekleştirdiği davranışlar hakkında daha az bilgi vermektedir.

2.3.2 Dinamik Analiz

Dosyanın çalıştırılarak analiz edilme süreci dinamik analiz olarak adlandırılmaktadır. Dinamik analiz, dosyanın bağlantı kurmaya çalıştığı ip adresi ve komuta kontrol sunucuları gibi ağ aktivitelerinin, dosya, dizin ve kayıt defteri gibi işletim sistemi üzerinde gerçekleştirdiği işlemlerinin ve dosyanın istenilen bir andaki bellek, CPU ve disk sistemi üzerindeki durumunun incelenmesini kapsamaktadır. Dinamik analiz kendi içerisinde basit dinamik analiz ve ileri seviye dinamik analiz olmak üzere 2' ye ayrılmaktadır. Dosyanın gerçekleştirdiği ağ işlemlerinin, işletim sistemi üzerindeki değişikliklerin incelenmesi basit dinamik analiz süreci içerisinde yer alırken dosyanın bellek, CPU ve disk sistemi

üzerindeki durumlarını adım adım incelenmesi ileri seviye dinamik analiz sürecinde yer almaktadır.

Zararlı yazılım analizi dosyanın statik ve dinamik analiz süreçlerinden elde edilen bilgiler doğrultusunda gerçekleştirilmektedir.

Bölüm 3

Kum Havuzu

Kum havuzu, zararlı içerik bulundurma ihtimali olan uygulamaları kontrollü bir ortamda çalıştırarak ağ ve işletim sistemi üzerindeki davranışlarını ve genel özelliklerini raporlayan otomatik analiz sistemidir. Kum havuzu analizi çıktıları, şüheli bir dosyanın zararlı olup olmadığı yönünde çıkarımda bulunmak, zararlı yazılımların nasıl temizlenebileceği veya engellenebileceği gibi konularda son derece yardımcı olmaktadır. Bunun yanında kum havuzları, zararlı yazılımları sergilediği davranışa göre kategorize edilmesine ve el ile gerçekleştirilmek istenecek ayrıntılı analiz sürecine fayda sağlamaktadır.

3.1 Kum Havuzu Yöntemleri

Kum havuzlarının, analiz işlemlerini gerçekleştirecekleri izole edilmiş ortamları bulunmaktadır. Bu ortamlarda gerçekleştirilen analizler ana bilgisayarda bir değişiklik gerçekleştirmemektedir. Kum havuzları genel olarak 2 bölümden oluşmaktadır. Bunlar ana makine ve şüpheli dosyaların çalıştırıldığıizole edilmiş ortamda bulunan makinedir.

Kum havuzlarının efektifliği 3 özellik ile ölçeklendirilebilmektedir. İlk olarak kum havuzları analiz ettiği şüpheli dosyanın gerçekleştirdiği bütün işlemleri takip edebilmelidir. İkinci olarak kum havuzları, zararlı yazılımların analizi engelleme için gerçekleştirmeye çalıştıkları analiz sistemi tespitini atlatabilmelidir. Son olarak ise birçok şüpheli dosya eş zamanlı olarak kum havuzunda analiz edilebilmelidir [6].

Bu bağlamda kum havuzları kullandıkları izole ortam ve bu ortamlarda gerçekleştirdikleri analizlerin efektifliği bakımından kullanıcı modu, çekirdek modu ve tam sistem analizi yapanlar olarak 3' e ayrılmaktadır.

3.1.1 Kullanıcı Modu

Şüpheli dosyaların sadece kullanıcı modunda gerçekleştirdiği işlemlerin takibini yapan kum havuzları sistem çağrılarının ya da Windows API fonksiyonlarının analizini yapabilmektedir. Bu nedenle çekirdek seviyesinde ve donanımda gerçekleşen işlemlerin takibi yapılmamaktadır. Sadece kullanıcı modunda analiz yapan kum havuzları analiz ortamı olarak sanallaştırma ortamları kullanmaktadır. Sanallaştırma ortamlarında bulunan sanal makineler birbirinden bağımsız ve izole edilmiş bir şekildedir. Fakat bir donanım zararlı yazılımının bu ortamlarda çalıştırılması ile sanallaştırma ortamının fiziksel kaynaklarında hatalar oluşabilmektedir. Oluşan bu hatalar sonucunda ortamda bulunan diğer sanal makineler de bu durumdan etkilenmekte ve çalışmalarını sonlandırarak analiz işlemi gerçekleştirememektedir. Böylelikle eş zamanlı analiz gerçekleştirilememektedir. Bu durum sanallaştırma ortamlarının eksik özelliklerinden biridir. Sanallaştırma ortamlarının bir diğer eksik özelliği ise zararlı yazılımlar tarafından tespit edilmeleri kolaydır. Sanallaştırma ortamlarının geliştirilmesi kolay olması ve bu ortamlarda çalışan programların normal bir hızda çalışabilmeleri ise avantajları olarak gösterilmektedir. Sanallaştırma ortamlarına örnek olarak Virtualbox ve Vmware verilebilir.

3.1.2 Çekirdek Modu

Exploit, Apt saldırıları gibi işletim sistemlerinin çekirdeklerinde işlemler yapan zararlıları inceleyebilmek için çekirdek modunda analiz gerçekleştirilmektedir. Çekirdek modunda analiz gerçekleştirilen kum havuzları, kullanıcı modundaki işlemlerin yanı sıra kullanılan çekirdek fonksiyonlarını, çekirdek bellek bölgesinde ve çekirdekte gerçekleşen diğer işlemleri takip edebilmektedir. Bu kum havuzları, sadece kullanıcı modunda analiz yapanlara göre daha doğru ve efektif sonuçlar sunmaktadır.

3.1.3 Tam Sistem

Kullanıcı modu ve çekirdek modu analiz işleminin 2 ayrı parçasıdır. Analizin diğer kısmını ise analiz edilen dosyanın donanımda gerçekleştirdiği işlemler oluşturmaktadır. Bu sebeple analiz edilen dosyanın çalıştığı ortamda tam analizin yapılabilmesi için emülasyon ortamları kullanılmaktadır. Emülasyon ortamı CPU, fiziksel bellek gibi donanımı simüle edebilen bir yazılımdır. Bu ortamda çalışan şüpheli dosyanın hem kullanıcı modunda hem çekirdek modunda hem de donanımda gerçekleştirdiği işlemler takip edilerek tüm sistemin analizi gerçekleştirilebilmektedir. Emülasyon ortamı donanımı simüle edebildiği için bu ortamda bulunan bir makinede donanım zararlı yazılımının çalışması, ortamda ve diğer makinelerde bir etki yapmamaktadır. Böylelikle sanallaştırma ortamlarının aksine eş zamanlı olarak zararlı yazılım analizi gerçekleştirilebilmektedir. Emülasyon ortamlarının zararlı yazılımlar tarafından zor tespit edilebilmesi bu ortamların bir diğer avantajı olarak gösterilebilmektedir. Bu ortamların dezavantajı ise performansının yavaş ve geliştirilmesinin zor olmasıdır. Emülasyon ortamına örnek olarak Qemu verilebilir.

3.2 Kum Havuzu Nasıl Çalışır?

Çekirdek modu ve tam sistem analizi gerçekleştiren kum havuzlarına erişimin kısıtlı olmasından dolayı bu ortamlarda detaylı inceleme gerçekleştirilememiştir. Bu sebeple kum havuzlarından sanallaştırma ortamı olarak Vmware veya Virtualbox ortamlarını kullananlar detaylı olarak incelenebilmiştir. Bu kum havuzlarının çalışma şekli ve sırası şu şekilde olmaktadır;

Şüpheli dosya izole edilmiş ortama gönderilerek çalıştırılmaktadır. Dosya, kayıt defteri ve ağ aktiviteleri gibi kullanıcı modunda gerçekleştirilen bütün işlemler analiz araçları ile takip edilerek kayıt altına alınmaktadır. Kaydedilen bu işlemler, analiz işlemi ya da kum havuzlarının belirledik analiz süresi bittiği zaman kum havuzları tarafından raporlanmaktadır.

3.3 Kum Havuzu Çeşitleri

Zararlı yazılımların sayısının hızla artması zararlı yazılımları engelleme ve inceleme ihtiyacını da artırmıştır. Bu sebeple zararlı yazılım analizini hızlandıran ve kolaylaştıran kum havuzlarının da önemi artmış ve kum havuzları sayısında bir artış gerçekleşmiştir.

Günümüzde birçok kum havuzu bulunmaktadır. Bu çalışmada, ücretsiz bir şekilde herkesin erişebileceği kum havuzları kullanılmıştır. Kullanılan kum havuzları aşağıda anlatılmaktadır.

3.3.1 Cuckoo Sandbox

Cuckoo Sandbox' da gerçekleştirilen analizlerde şüpheli dosya izole edilmiş sanal ya da fiziksel bir makinede çalıştırılmaktadır. Temel altyapısı bir adet ana makine ve bir veya birden fazla izole edilmiş misafir makinelerden oluşmaktadır. Ana makine şüpheli dosyayı ve analiz araçlarını izole edilmiş ortama göndermektedir. Daha sonra ise dosyayı çalıştırmakta ve analizi başlatmaktadır. Analiz işlemi gerçekleşirken ana makine, dosyanın oluşturduğu ağ trafiğini de kaydetmektedir. İzole ortamda çalışan şüpheli dosyanın gerçekleştirdiği işlemler burada takip edilmekte ve analiz tamamlandığında sonuçlar ana makineye gönderilerek raporlanmaktadır. Analiz raporlarını HTML, JSON ve MAEC formatlarında sunabilmektedir [7].

Cuckoo Sandbox Windows, Linux ve Android işletim sistemi dosyalarının analizini gerçekleştirmekte ve Virtualbox, VmWare ve KVM sanallaştırma ortamlarını desteklemektedir. Cuckoo kum havuzu kullanıcı modundaki işlemleri takip etmekte ve buna göre analiz sonucu üretmektedir.

3.3.2 Anubis

Otomatikleştirilmiş zararlı yazılım analiz aracı Anubis kum havuzu [8], izole edilmiş ortam olarak Qemu kullanmakta ve tam sistem analiz gerçekleştirmektedir. İşletim sistemi olarak Windows XP SP2 kullanmaktadır. Qemu emülasyon ortamına dışarıdan tam erişimin olması ve temel sanal makine tespit izlerinin bu ortamda bulunmaması Qemu'nun tercih edilme sebeplerindendir. 4 dakika içerisinde tamamen otomatik bir şekilde şüpheli dosya analizini gerçekleştirmekte ve raporlamaktadır. Anubis, zararlı yazılım analizinde

sistem çağrılarını, Windows API fonksiyonlarını, dosya ve kayıt defteri işlemlerini ve ağ trafiğini takip ederek detaylı bir analiz raporu oluşturmaktadır. Analiz raporunu HTML, XML, PDF ve Text formatlarında sunabilmektedir. 2016 itibari ile çevrimiçi hizmeti yoktur.

Anubis kum havuzunun dinamik kod analizi gerçekleştirerek şüpheli dosyanın çalışma süresinde gerçekleşebilecek kod karmaşıklaştırılma, hata ayıklamayı engelleme ve paketleme yöntemlerinden etkilenmemektedir.

3.3.3 Comodo Otomatik Analiz Sistemi

Comodo kum havuzu [9], şüphelenilen dosyaların tamamen izole edilmiş ortamlarda otomatik olarak analizlerini gerçekleştiren bir analiz sistemidir. Windows çalıştırılabilir dosyalarını analizinin gerçekleştirildiği Comodo kum havuzu analiz çıktılarında şüpheli dosyanın gerçekleştirdiği işlemleri raporlamakta ve dosyanın zararlı olup olmadığı hakkında çıkarımlarda bulunmaktadır. Comodo kum havuzu, analiz işlemleri sırasında ana makineden sanallaştırılmış ortamda çalışan uygulamaları gösterebilme gibi bir özelliğe de sahiptir. Vmware sanallaştırma ortamı kullanan Comodo, kullanıcı modundaki işlemlerin takibini yapmaktadır.

3.3.4 Threat Expert

Threat Expert kum havuzu [10], analiz ettiği şüpheli dosyanın tüm işlemlerini takip ederek raporlamaktadır. Ayrıntılı teknik bir rapor sunan Threat Expert, zararlı dosyaların türlerini ve tehlike derecesini de belirtmektedir. Analiz ortamı olarak Vmware kullanan kum havuzu analiz raporunda dosya ve kayıt defteri, gerçekleşen ağ trafiği gibi kullanıcı modunda gerçekleşen işlemleri kullanıcıların anlayabileceği şekilde sunmaktadır. Ayrıca zararlı yazılımın kaynağının hangi ülke veya ülkeler olduğu konusunda çıkarımlarda bulunmakta ve zararlı yazılımın türü ile ilgili istatiksel veriler de paylaşmaktadır. Analiz raporunu Microsoft MHTML formatında sunmaktadır.

3.3.5 Payload Security

VxStream kum havuzu olarak da bilinen otomatik zararlı yazılım analiz sistemi hibrit analiz gerçekleştirmektedir. VxStream kum havuzu, sanallaştırma ortamı olarak VMWare ESX ve Virtualbox kullanmaktadır. Windows Xp' den Windows 10'a kadar olan tüm Windows işletim sistemlerini desteklemektedir. 380'den fazla dosya davranış türü filtrelemesi bulunduran kum havuzu şüpheli dosyanın gerçekleştirdiği işlemleri XML, JSON, HTML formatlarında raporlamaktadır.

VxStream kum havuzu hibrit analiz ile şüpheli dosya tarafından gerçekleştirilen bütün dosya işlemlerinin listelenmesini ve hareketsiz kod, kabuk kodu, bütün hafıza ve şüpheli dosya tarafından indirilen dosyaların analizini de sağlamaktadır. Fakat hibrit analizin kendisine ait bazı özelliklerinden dolayı VxStream kum havuzu, zararlı yazılımlar tarafından analiz sistemi tespitinde kullanılabilecek daha fazla yöntem bulundurmaktadır [11].

Bölüm 4

Kum Havuzu Tespit Yöntemleri

Zararlı yazılım analizi dosyaların zararlı olup olmadığının anlaşılmasını sağlamakta, zararlı dosyanın bulaştığı sistemlerin tespit edilmesinde ve bu sistemlerden temizlenebilmesinde yardımcı olmaktadır. Daha sonra ise bu ve buna benzer aktiviteleri gerçekleştiren dosyalara, antivirüs firmaları tarafından önlemler alınarak zararları en aza indirgenmektedir. Ayrıca zararlı yazılımlar tarafından yapılan saldırılar kategorilendirilerek gerekli önlemler hızlıca alınabilmektedir. Tüm bu durumlar zararlı yazılım yazarlarının analizleri engelleyecek yöntemler üretmelerini gerekli hale getirmiştir.

Zararlı yazılım yazarları, otomatik analiz sistemlerinde analiz işlemlerini engelleyebilmek için çalışma ortamları, analiz sırasında çalışan ya da eklenen modülleri, dosya izleri gibi analiz sistemlerini diğer sistemlerden ayıran özellikleri belirleyerek bunlara karşı tespit yöntemleri geliştirmişlerdir.

Analiz sistemleri için geliştirilen tespit yöntemleri 3 ana başlıkta kategorize edilebilmektedir.

4.1 Sanallaştırma ve Emülasyon Ortamlarının Tespiti

Sanallaştırma ortamları, gerçek bir bilgisayarda işletim sisteminin üzerine kurulan bir yazılımdır. Bu ortamlarda birden fazla işletim sistemi kurularak sanal bilgisayarlar oluşturulabilmektedir. Oluşturulan sanal bilgisayarlar için gerçek bilgisayarın hafıza alanından belirli bir bölüm tahsis edilir. Sanallaştırılmış ortamda bulunan işletim sistemi gerçek bir bilgisayardaki fonksiyonların ve özelliklerin neredeyse tamamını sağlamaktadır.

Sanallaştırılmış ortamdaki işletim sistemlerinde gerçekleştirilen işlemler gerçek bilgisayarda herhangi bir değişiklik oluşturmamaktadır. Aynı gerçek bilgisayarda birden fazla sanallaştırma ortamı bulunabilmekle birlikte aynı sanallaştırma ortamında birden fazla işletim sistemi kurulu ortam hazırlanabilmektedir. Sanallaştırma ortamlarında bulunan işletim sistemleri birbirlerinden bağımsızdır ve kendi aralarında herhangi bir etkileşimde bulunamazlar.

Sanallaştırma ortamlarının gerçek bilgisayar özelliklerinin neredeyse tamamına sahip olması ve içerisinde birçok sanal bilgisayar oluşturulabilmesi kum havuzları tarafından zararlı yazılım analizinde kullanılmasını sağlamıştır. Çünkü bu ortamda çalıştırılan zararlı yazılımlar gerçek bilgisayara herhangi bir zarar verememektedir. Ayrıca gerçek bilgisayarda birden fazla sanal bilgisayar oluşturulabilmesi ve bunların üzerine farklı işletim sistemleri kurulabilmesi, analizlerin daha hızlı ve güvenilir olmasını sağlamaktadır. Günümüzde kullanılan birçok sanallaştırma ortamı bulunmaktadır. Bu çalışmada kum havuzlarının kullandığı ortamlarından 3 tanesinde analiz ortamı tespiti gerçekleştirmek için testler gerçekleştirilmiştir. Testlerin gerçekleştirildiği ve en çok kullanılan 3 ortam aşağıda açıklanmaktadır.

4.1.1 Virtualbox

İlk olarak Innotek GmbH tarafından oluşturulmuştur. Daha sonra Sun Microsystems firmasına satılmıştır. Bu firmanın Oracle tarafından satın alınması ile Oracle tarafından geliştirilen VirtualBox [12] sanallaştırma ortamı birçok işletim sistemini desteklemektedir. Ayarlamaları kolay bir şekilde gerçekleştirilen Virtualbox sanallaştırma ortamına başka sanallaştırma ortamları da kurulabilmektedir.

Kullanılması kolay olan Virtualbox açık kaynaklı bir yazılımdır. Farklı işletim sistemleri üzerinde çalışıp içerisine farklı ve birçok işletim sistemi kurulabilen Virtualbox sanallaştırma ortamı kum havuzları tarafından çokça tercih edilmektedir.

Virtualbox sanallaştırma ortamı gerçek bir bilgisayarda kurulu işletim sistemi özellikleri göstermesine rağmen kendisine ait bazı özel izleri bulunmaktadır. Bu izler zararlı yazılımlar tarafından analiz sistemi tespiti olarak değerlendirilmekte ve sanallaştırma ortamı izi olarak belirtilmektedir. Bu izler aşağıda gösterilmektedir:

- Kayıt Defteri İzleri: Virtualbox sanallaştırma ortamının kayıt defterinde bulunan izleri Tablo 4.1 gösterilmektedir.

Tablo 4.1: Virtualbox Kayıt Defteri İzleri.

	Kayıt Defteri İzi
Virtualbox Guest Additions	HKEY_LOCAL_MACHINE\SOFTWARE\Oracle\ VirtualBox Guest Additions
Sistem Bilgileri	HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\ Enum\IDE, HARDWARE\ACPI\DSDT\VBOX_, HARDWARE\ACPI\FADT\VBOX_, HARDWARE\ACPI\RSDT\VBOX_
Bios	HARDWARE\DESCRIPTION\ System (SystemBiosVersion, vbox), HARDWARE\DESCRIPTION\ System (VideoBiosVersion, virtualbox)
Ürün Adı	HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\ SystemInformation (SystemProductName,virtualbox)
Pencere İsmi	VboxTrayToolWnd, VboxTrayWndClass
Araçlar	VboxTrayIPC, VboxMiniRdrDN, VboxMiniRdDN

- Dosya İzleri: Virtualbox sanallaştırma ortamının dosya izleri Tablo 4.2 gösterilmektedir.

Tablo 4.2: Virtualbox Dosya İzleri.

	Dosya İzi
Sürücüler	vBoxMouse.sys, vBoxGuest.sys, vBoxSF.sys, vBoxVideo.sys
DLL Dosyaları	vboxdisp.dll, vboxhook.dll, vboxoglerrorspu.dll, vboxoglpackspu.dll, vboxogl.dll, vboxmrxnp.dll, vboxoglcrutil.dll, vboxoglfeedbackspu.dll, vboxoglpassthroughspu.dll, vboxoglarrayspu.dll
Çalıştırılabilir Dosyalar	vboxservice.exe, vboxControl.exe, vboxtray.exe
Servisler	VBoxGuest, VBoxMouse, VBoxService, VBoxSF, VBoxVideo

- Mac Adres İzi: Virtualbox sanallaştırma ortamının statik bir MAC adresi bulunmakta ve “08:00:27” ile başlamaktadır.

- Paylaşılan Klasör İzi: Virtualbox sanallaştırma ortamlarında gerçek makine ile sanal makine arasında dosya paylaşımı yapabilmek için kullanıcılar tarafından paylaşım klasörü oluşturulabilmektedir. Bu klasör sanal makine izi olarak zararlı yazılımlar tarafından tespit edilmektedir.

4.1.2 Vmware Workstation

Vmware şirketi tarafından 1999 yılında piyasaya sürülmüştür. Vmware Workstation [12] sanallaştırma ortamının geliştirilme fikri IBM tarafından 1960 yılında üretilmiştir. İçerisine birden fazla işletim sistemi kurulabilen Vmware sanallaştırma ortamı aynı zamanda birçok işletim sistemini desteklemektedir. Ücretli olarak marketlerde satışa sunulan Vmware sanallaştırma ortamı ücretsiz olarak da Vmware Player olarak internet ortamından temin edilebilmektedir.

Günümüzde en çok satılan sanallaştırma ortamı olan Vmware Workstation, diğer sanallaştırma ortamlarından daha zengin özelliklere sahiptir. Kum havuzları tarafından en çok tercih edilen sanallaştırma ortamlarından bir diğeridir. Bu sanallaştırma ortamında kurulan işletim sistemlerinde donanım tanımları ve Vmware Workstation' a ait yardımcı programlar gibi sanallaştırma ortamına has bazı özellikler bulunmaktadır. Bu özellikler, zararlı yazılımlar tarafından analiz sistemi tespitinde kullanılabilmektedir. Vmware sanallaştırma ortamına ait izler aşağıda gösterilmektedir:

- Kayıt Defteri İzleri: Vmware sanallaştırma ortamına ait kayıt defteri izleri Tablo 4.3 gösterilmektedir.
- Dosya İzleri: Vmware sanallaştırma ortamında bulunan ve kullanılan sanallaştırma ortamına ait dosya izleri Tablo 4.4 gösterilmektedir.
- Mac Adres İzi: Vmware sanallaştırma ortamının statik bir mac adresi bulunmakta ve 00-0c-29 ile başlamaktadır.
- Giriş Çıkış Portu: Vmware sanallaştırma ortamı, barındırdığı sanal makinelerle iletişimini 0x5658 portu ile gerçekleştirmektedir. Bu porttan okuma yapılarak Vmware tespiti gerçekleştirilebilmektedir.

Tablo 4.3: Vmware Kayıt Defteri İzleri.

	Kayıt Defteri İzi
Araçlar	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\ CurrentVersion\Run (VMware User Process), HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\ CurrentVersion\Setup\SharedDlls
Sistem Bilgileri	HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Video\ 4BEF3D64-1F2B-4026-9EE4-B6D8CD9FEA1B\ 0000 (Device Description, vmware svga ii), HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Video\ 3A8088C5-4419-4572-801C-A10BA858952F\0000 (Device Description, vmware svga 3d), VirtualDeviceDrivers, root#vmwvmcihostdev, HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\IDE\ CdRomNECVMWar_VMware_SATA_CD01_1.00_, HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\IDE\ CdRomNECVMWar_VMware_IDE_CDR10_1.00_, HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\SCSI\ Disk&Ven_VMware_&Prod_VMware_Virtual_S&Rev_1.0, HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\SCSI\ Disk&Ven_VMware_&Prod_VMware_Virtual_S
Sürücüler	HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\ 4D36E968-E325-11CE-BFC1-08002BE10318\0000 (DriverDesc, vmware svga ii), HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\ 4D36E968-E325-11CE-BFC1-08002BE10318\0000 (DriverDesc, vmware svga 3d), HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\ 4D36E968-E325-11CE-BFC1-08002BE10318\0000 (DriverDesc, vmware vmscsi controller), vmmouse
Devicemap	HARDWARE\DEVICEMAP\Scsi\Scsi Port 0\Scsi Bus 0\ Target Id 0\Logical Unit Id 0 (Identifier, vmware)

Tablo 4.4: Vmware Dosya İzleri.

	Dosya İzi
Sürücüler	vBoxMouse.sys, vBoxGuest.sys, vBoxSF.sys, vBoxVideo.sys
DLL Dosyaları	vm3dgl64.dll, vm3dgl.dll, vm3dum64.dll, vm3dum.dll, VmbuxCoinstaller.dll, vmGuestLib.dll, vmGuestLibJava.dll,vmhgfs.dll, vmwogl32.dll, vmmreg32.dll, vmx_fb.dll, vmx_mode.dll, VMUpgradeAtShutdownWXP.dll
Çalıştırılabilir Dosyalar	vmicsvc.exe
Çalışan İşlemler ve Servisler	Vmtoolsd, Vmacthlp, Vmhgfs, VMMEMCTL, Vmmouse, Vmrawdsk, VMTools, Vmusbmouse, Vmvss, Vmscsi,Vmxnet, VMware Physical Disk Helper Service, vmx_svga

4.1.3 Qemu

Qemu [13] açık kaynaklı emülasyon ortamıdır. Qemu kum havuzları tarafından tercih edilen izole edilmiş ortamlardan biridir.

Qemu ortamının kendisine ait bazı özellikleri zararlı yazılımlar tarafından analiz sistemi tespiti için kullanılmaktadır. Qemu emülasyon ortamının sanallaştırma ortamları kadar

tespit yöntemleri bulunmamaktadır. Analiz sistemi tespiti için kullanılan özelliklerden sadece kayıt defterinde olanlar aşağıda gösterilmektedir:

- Kayıt Defteri İzleri: Qemu sanallaştırma ortamına ait kayıt defteri izleri Tablo 4.5 gösterilmekedir.

Tablo 4.5: Qemu Kayıt Defteri İzleri.

	Kayıt Defteri İzi
Devicemap	HARDWARE\DEVICEMAP\Scsi\Scsi Port 0\Scsi Bus 0\Target Id 0\ Logical Unit Id 0 (Identifier, QEMU)
Sistem Bilgileri	HARDWARE\Description\System (SystemBiosVersion, QEMU)
CPU	QEMU Virtual CPU

4.1.4 İşlemci Kontrolü

Kum havuzlarının kullandıkları sanallaştırma ortamlarında bulunan işletim sistemlerinin işlemcileri genel olarak gerçek makinelerden daha düşük işlemcilere sahip olacak şekilde ayarlanmaktadır. Gerçek makinelerde kullanılan işlemcilerin çekirdek sayısı en az 2 adettir. Zararlı yazılım yazarları bu durumu bildiklerinden dolayı işlemcilerin çekirdek sayılarını kontrol etmektedirler. Çekirdek sayısı 2'den küçük olan ortamlarda kum havuzu ya da sanallaştırma ortamı tespiti yapabilmektedirler [14, 15].

Bazı zararlı yazılımlar [16] işlemci kontrolü yapmaktadırlar. Eğer ki işlemci Intel(R) Core(TM) 2 CPU T7200 @ 2.00 GHz ya da Intel(R) Xeon(R) CPU L5640 @ 2.27 GHz ise kum havuzu tespiti yapmaktadırlar. Ayrıca zararlı yazılımlar işlemci isimleri karşılaştırması yaparak da kum havuzu tespiti yapabilmektedirler. Windows kayıt defterinde görünen işlemci ismi ile CPUID'de görünen işlemci ismi aynı olmalıdır. Aynı olmadığı durumda kum havuzu tespiti yapılabilmektedir.

İşlemci kontrolü ile ilgili bir diğer analiz sistemi tespit yöntemi ise instruction uzunluğunun kontrolüdür. Gerçek bilgisayarların işlemcilerindeki instruction uzunluğu 0x15 byte değerinden büyük olamazken sanallaştırma ortamlarında böyle bir sınır bulunmamaktadır. Gerçek bilgisayarların işlemcilerinde 0x15 byte değerinden büyük bir instruction girildiğinde hata ayıklayıcı tetiklenirken sanallaştırma ortamlarında herhangi bir hata ayıklayıcı tetiklenmemektedir. Zararlı yazılımlar 0x15 byte değerinden uzun bir instruction girip hata ayıklayıcının tetiklenip tetiklenmediğini kontrol ederek sanallaştırma

ortamı tespiti yapabilmektedir. Buna örnek zararlı yazılım olarak Qakbot zararlısı verilebilir [15].

4.1.5 Disk Kontrolü

Otomatik analiz sistemlerinde kullanılan bilgisayarların disk boyutları küçük olabilmektedir. Bazı zararlı yazılımlar otomatik analiz sistemini tespit için disk boyutunu kontrol etmekte ve 60 gb değerinden düşük ise analiz sistemi olarak tanımlamaktadırlar [17].

4.2 Kum Havuzu Modüllerinin Tespiti

Kum havuzlarının analiz işlemlerinde kullanmak için geliştirdikleri modüller bulunmaktadır. Bu modüller analiz edilecek dosyaya enjekte edilerek çalışmaktadır. Modüller sayesinde analiz edilecek dosyanın davranışları kayıt altına alınabilmekte, kullandığı fonksiyonlar kancalanarak kum havuzunun belirlediği fonksiyonlar çalışabilmektedir.

Kum havuzlarının her birinin kendisine ait modülü olmakla birlikte bazı modüller diğer kum havuzlarında da kullanılabilmektedir. Zararlı yazılımlar bu modülleri ya da modüllerin uyguladıkları bazı fonksiyonları kontrol ederek kum havuzu tespiti yapabilmektedir.

4.2.1 Kancalanan Windows API Fonksiyonları

Kum havuzları, analiz işlemlerinde zararlı yazılımın kullanmış olduğu bazı fonksiyonları kancalamaktadır. Bu fonksiyonlar analiz edilen dosya tarafından çağrıldığında kum havuzunun kancalamasına takılmaktadır. Kancalama işlemi gerçekleştikten sona kum havuzunun fonksiyonlar devreye girerek çalışmaktadır.

Kum havuzlarının kancalama işlemi yaptığını bilen zararlı yazılımlar, bu kancalama işlemine engel olmakta ya da kancalama işleminin gerçekleştiğini fark ettiklerinde çalışmalarını değiştirmekte ya da sonlandırmaktadır.

4.2.2 DLL

DLL, dinamik link kütüphanesi olarak da adlandırılan, bir programın çalışması esnasında çağırabileceği fonksiyonları barındıran Windows dosyalarıdır. DLL dosyaları sayesinde programlar kullanmak istedikleri fonksiyonları dinamik bir şekilde kullanabilmektedirler.

Kum havuzlarının analiz işlemi için kullandıkları DLL dosyaları bulunmaktadır. Bu dosyalar uygulama çalışırken uygulamanın sergilemiş olduğu davranışları kaydetmek için uygulamaya enjekte edilerek kullanılmaktadır. Kum havuzlarının kullanmış oldukları DLL dosyaları dbghlp.dll, sbiedll.dll, cuckoomon.dll olarak bilinmektedir. Zararlı yazılımlar kendilerine bu dosyaların enjekte edilip edilmediğini çalışma esnasında kontrol etmektedirler. Bu kontrol işlemi GetModuleHandleA fonksiyonunu kullanılarak gerçekleştirilebilmektedir [18].

GetModuleHandleA fonksiyonuna parametre olarak kum havuzlarının kullanmış oldukları DLL dosyalarının isimleri verilmektedir. Fonksiyondan dönen değer Non Null ise bu DLL dosyasının, zararlı yazılımın adres alanına yüklü olduğu anlaşılmaktadır. Bu sayede zararlı yazılım kum havuzu tespiti yapabilmektedir.

Kum havuzu tespitini gerçekleştiren zararlı yazılım bu durumda üç farklı durum sergileyebilmektedir. Zararlı yazılım, kum havuzu tespitini yaptığı zaman ya çalışmasını sonlandırmakta ya da aslından farklı bir davranış gerçekleştirebilmektedir. Bu iki durumdan hariç olarak da zararlı yazılım kendi adres alanına yüklenmiş DLL dosyasını kaldırma yöntemine başvurabilmektedir. Bunun için de FreeLibrary fonksiyonunu kullanarak yüklü DLL dosyasını adres alanından kaldırmakta ve gerçekleştireceği davranışların analiz sistemi tarafından kayıt edilmesini engelleyebilmektedir [15].

4.3 Kum Havuzu İzleri

Kum havuzlarının kullandıkları izole edilmiş ortamlarda analiz sistemlerine ait bazı izler bulunabilmektedir. Bu izler kum havuzlarının kullandıkları özel dosyalar, klasörler ve kullanıcı kontrolleri olarak ifade edilebilmektedir. Zararlı yazılımlar bu izlerin varlığını kontrol ederek çalışmalarını sonlandırmakta ya da aslından farklı davranışta bulunmaktadır. Böylelikle kum havuzu yanlış analiz sonucu vermektedir.

Zararlı yazılımlar tarafından kontrol edilebilecek kum havuzu izleri aşağıda kategorize edilmiş bir şekilde anlatılmaktadır.

4.3.1 Özel Dosyalar

Bazı kum havuzları, analiz işlemleri için sanallaştırma ortamlarına analiz dosyaları göndermektedir veya sanallaştırma ortamlarında bu dosyalar halihazırda bulunmaktadır. Bu dosyalar analiz işlemi esnasında kullanılmaktadır ve statik isimlere sahiptir. Zararlı yazılım bu statik isimlerin kontrolünü gerçekleştirerek kum havuzu tespiti yapabilmektedir.

Örnek olarak Cuckoo kum havuzu verilebilir. Cuckoo kum havuzu, analiz işlemi başlamadan önce sanallaştırma ortamına analiz işlemi için rastgele isimli bir klasör göndermektedir. Bu klasör ve dosyalar Tablo 4.6 gösterilmektedir.

Tablo 4.6: Dosya İzleri.

	Cuckoo Kum Havuzu Dosya İzleri
./	analyzer.py
bin	execsc.exe
dll	cuckoomon.dll, cuckoomon_bson.dll, cuckoomon_netlog.dll
lib/api	process.py, screenshot.py
lib/common	abstract.py, constants.py, defines.py, errors.py, exceptions.py, hashing.py, rand.py, results.py
lib/core	config.py, packages.py, privileges.py, startup.py
modules/auxiliary	disguise.py, human.py, screenshots.py
modules/packages	applet.py, bin.py, cpl.py, dll.py, doc.py, exe.py, generic.py, html.py, ie.py, jar.py, msi.py, pdf.py, ppt.py, ps1.py, python.py, vbs.py, xls.py, xip.py

Ayrıca Cuckoo kum havuzu, analiz sonuçlarını sanallaştırma ortamında rastgele oluşturduğu bir isimdeki klasörün altında depolamaktadır. Bu klasörün altında da drop, files, logs, memory, shots adlarında klasörler oluşturmaktadır. Zararlı yazılım yazarları bu klasör ve dosya adlarını içeren bir kontrol mekanizması oluşturarak kum havuzu tespiti yapabilmektedirler.

Cuckoo kum havuzu haricindeki diğer kum havuzlarının da kullandığı özel statik isimli dosyalar bulunmaktadır. Zararlı yazılımlar bu dosyaların ve bulundukları dizinlerin varlıklarını kontrol ederek de kum havuzu tespiti yapabilmektedir [16]. Bu dosyalar Tablo 4.7 gösterilmektedir.

4.3.2 ID

Bilgisayarların kullandıkları işletim sistemlerinin bir ürün numarası(product id) bulunmaktadır. Bazı kum havuzlarının bu numaraları statik bir şekilde olduğundan dolayı kum havuzu tespiti yapılabilmektedir [18]. Bu kum havuzlarına örnek olarak Anubis, JoeBox ve CWSandbox verilebilir. Ürün numaraları;

- Anubis : 76487-337-8429955-22614
- JoeBox : 55274-640-2673064-23950
- CWSandbox : 76487-644-3177037-23510

4.3.3 Çalışma Zamanı

Kum havuzları, analiz işlemini gerçekleştirirken maksimum analiz süresi belirlemektedir. Gerçekleşen analizin süresi kum havuzlarının belirledikleri maksimum süreye ulaştığında analiz işlemi otomatik olarak sonlandırılmakta ve bu süreye kadar gerçekleşen işlemler analiz sonucu olarak raporlanmaktadır. Zararlı yazılımlar kum havuzlarında analiz edilmelerini engellemek için uygulamanın başlangıcına sleep fonksiyonu koyarak kum havuzlarının belirledikleri analiz süresi boyunca uygulamayı uyutmakta yani beklemektedir. Belirledikleri süre bittikten sonra zararlı yazılım asli aktivitesini yapmaya başlamaktadır. Bu sayede kum havuzlarının analiz süresi bittiğinden dolayı analiz işlemi doğru olarak gerçekleşmemektedir.

Çalışma zamanı ile analiz edilmeyi engelleme yöntemlerine bir başka örnek olarak belirli zamanlarda çalışan zararlı yazılımlar gösterilebilir. Bazı zararlı yazılımlar [19] ayın belirli günleri çalışmakta ve bu sayede otomatik analiz sistemlerinde analiz edilmelerini zorlaştırmaktadır.

4.3.4 Port Numarası

Kum havuzları kullandıkları sanal ortamlarla iletişimi gerçekleştirmek için bir port numarası kullanırlar. Bu port numarası eğer sabit ise zararlı yazılımlar bunu kontrol ederek kum havuzu tespiti gerçekleştirebilmektedir.

Cuckoo kum havuzu sanal makine ile iletişimini 2042 portu üzerinden gerçekleştirir. Zararlı yazılım bu portun açık olup olmadığını kontrol ederek Cuckoo kum havuzunu tespit edebilmektedir.

4.3.5 Kullanıcı Kontrolü

Otomatik analiz sistemlerinde kullanıcı müdahalesi olmadığından dolayı zararlı yazılımlar kullanıcı aktivitelerini kontrol ederek analiz sistemlerini tespit edilebilmektedir. Zararlı yazılımların kontrol ettikleri kullanıcı aktivitelerinden birisi fare hareketleridir. Fare uzun süre hareket etmiyorsa zararlı yazılımlar bunu analiz sistemi olarak algılamakta ve çalışmamakta ya da asli fonksiyonlarını gerçekleştirmemektedir [19].

Bazı kum havuzları fare hareketinden dolayı tespit edilmeleri atlatabilmek için fareyi rastgele hızlı bir şekilde hareket ettirmektedir. Bazı zararlı yazılımlar da bu duruma karşılık hızlı fare hareketlerinin anormal bir durum olarak algılamakta ve bunu analiz sistemi olarak tanımlayarak asli çalışmalarını sergilememektedir [20].

Kullanıcı kontrolü ile analiz sistemi tespit yöntemlerinden bir diğeri de komut girme gerekliliğidir. Kum havuzlarındaki analiz işlemlerinde dışarıdan bir kullanıcının herhangi bir hamlesi yoktur. "Zararlı yazılımı çalıştır" komutu verilerek zararlı yazılım çalıştırılır ve analiz edilir. Zararlı yazılım yazarları otomatik analiz sistemlerinin tespitinde bu durumdan faydalanmaktadır. Otomatik analiz sistemini tespit için zararlı yazılım ilk çalıştığında güvenlik kodu istemekte ve bu sayede kullanıcı kontrolü yapmaktadır. İstenilen güvenlik kodu girilemediğinden dolayı zararlı yazılım asli fonksiyonlarını gerçekleştirmemektedir. Bu nedenle zararlı yazılım kum havuzunda doğru bir şekilde analiz edilememektedir.

Zararlı yazılımların kullanıcı kontrolü yaparak analiz sistemi tespit yöntemlerinden bir diğeri ise kullanıcı isimleridir. Otomatik analiz sisteminde çalışan bilgisayarın kullanıcı adı sabit ve zararlı yazılım yazarları tarafından bu durum biliniyorsa zararlı yazılım bu kullanıcı adını kontrol ederek kum havuzu tespiti yapabilmektedir. Anubis kum havuzunda kullanıcı adı andy ve user, ThreatExpert kum havuzunda COMPUTERNAME olarak kullanılırken CUCKOO, SANDBOX, NMSDBOX, XXXX-OX, CWSX,

WILBERT-SC, XPAMAST-SC isimleri de diğer kum havuzları tarafından kullanılabilmektedir. Bu sabit isimler kum havuzlarının zararlı yazılımlar tarafından tespit edilmesine neden olmaktadır [21].

Bazı zararlı yazılımlar, kullanıcı adı rastgele bir metin ise bunu analiz sistemi olarak değerlendirerek çalışmamaktadır [22].

Tablo 4.7: Dosya İzleri.

	Diğer Kum Havuzlarının Dosya İzleri
DLL	c:/agent/MemoryDump.dll,c:/bin/AHookMonitor.dll, c:/MWS/bin/agent/hipsengine32.dll, c:/MWS/bin/agent/hipspcap32.dll, c:/original/AutoRepGui/autorep.dll, C:/SandCastle/tools/FakeHTTPServer/python27.dll, C:/SandCastle/tools/syelog.dll, C:/SandCastle/tools/TmSigChk.dll, C:/SandCastle/tools/tracer.dll, D:/plugins/import/import_pe.dll,D:/plugins/import/sb_import.dll, D:/plugins/import/import_launcher.dll, D:/plugins/process/filesystem.dll, D:/plugins/process/sniffer.dll, D:/plugins/process/wincheck.dll, D:/plugins/process/procmon.dll, D:/plugins/process/screenshot.dll, D:/plugins/process/dumper.dll, D:/plugins/process/hosts.dll, D:/plugins/process/mutexenum.dll, D:/plugins/process/sb_status.dll
Çalıştırılabilir	C:/exec.exe, C:/sample.exe, c:/agent/ProcessMemDump.exe, c:/bin/hookanaapp.exe, c:/cwsandbox/cwsandbox.ini, c:/cwsandbox/cwsandboxstarter.exe, c/gfisandbox/starter.exe, c:/manual/grabme.exe, C:/manual/SilipTCPIP.exe, c:/MWS/bin/agent/hips32.exe, c:/original/AutoRepGui/AutoRepGui.exe, C:/sandbox/w64h/wow64hlp.exe, C:/SandCastle/tools/BehaviorDumper.exe, C:/SandCastle/tools/FakeHTTPServer/_hashlib.pyd, C:/SandCastle/tools/FakeHTTPServer/_socket.pyd, C:/SandCastle/tools/FakeHTTPServer/_ssl.pyd, C:/SandCastle/tools/FakeHTTPServer/FakeHTTPServer.exe, C:/SandCastle/tools/FakeHTTPServer/select.pyd, C:/SandCastle/tools/FakeServer.exe, C:/SandCastle/tools/LiteClient.exe, c:/tools/aswsnx/snxcmd.exe, c:/Tools/findt2005.exe, c:/totalcmd/gfiles/360tray.exe, c:/totalcmd/gfiles/avcenter.exe, c:/totalcmd/gfiles/avgnt.exe, c:/totalcmd/gfiles/avguard.exe, c:/totalcmd/gfiles/avp.exe, c:/totalcmd/gfiles/Avp32.exe, c:/totalcmd/gfiles/Avpcc.exe, c:/totalcmd/gfiles/Avpm.exe, c:/totalcmd/gfiles/Avpupd.exe, c:/totalcmd/gfiles/CCenter.exe, c:/totalcmd/gfiles/cpd.exe, c:/totalcmd/gfiles/filemon.exe, c:/totalcmd/gfiles/fsav32.exe, c:/totalcmd/gfiles/fsbwsys.exe, c:/totalcmd/gfiles/fsdfwd.exe, c:/totalcmd/gfiles/Fsgk32.exe, c:/totalcmd/gfiles/kavmm.exe, c:/totalcmd/gfiles/KavPFW.exe, c:/totalcmd/gfiles/kavsvc.exe, c:/totalcmd/gfiles/navapsvc.exe, c:/totalcmd/gfiles/Navrunr.exe, c:/totalcmd/gfiles/Navw32.exe, c:/totalcmd/gfiles/Navwnt.exe, c:/totalcmd/gfiles/NISSERV.EXE, c:/totalcmd/gfiles/nod32cc.exe, c:/totalcmd/gfiles/nod32krn.exe, c:/totalcmd/gfiles/nod32kui.exe, c:/totalcmd/gfiles/nod32m2.exe, c:/totalcmd/gfiles/outpost.exe, c:/totalcmd/gfiles/procexp.exe, c:/totalcmd/gfiles/procmon.exe, c:/totalcmd/gfiles/regmon.exe, c:/totalcmd/gfiles/SAVScan.exe, c:/totalcmd/gfiles/symlcsvc.exe, c:/totalcmd/gfiles/tcpview.exe, c:/totalcmd/gfiles/zonealarm.exe,c:/tracer/FortiTracer.exe, Documents and Settings/Administrator/Desktop/AVCTestSuite/AVCTestSuite.exe, d:/sandbox_svc.exe, C:/TEST/sample.exe, C:/TEST/

Bölüm 5

Kum Havuzu Tespit Araçları

Zararlı yazılımların kum havuzlarını tespit etmek için kullandıkları ya da kullanabilecekleri yöntemlerin karşı kum havuzlarında etkili olup olmadığını gözlemlemek için tespit araçları kullanılmıştır.

5.1 Pafish

Kum havuzu, sanal makine ve bazı analiz ortamlarının tespitini gerçekleştiren Pafish [23], açık kaynaklı bir şekilde internet ortamında bulunmaktadır.

Pafish tespit aracı, bu çalışmada anlatılan analiz sistemi tespit yöntemlerini atlatma tekniklerini içeren sanal makinede nasıl sonuçlar vereceğini görebilmek için kullanılmıştır. Pafish tespit aracının gerçekleştirdiği işlemler şunlardır:

- Virtualbox Tespiti
- VMWare Tespiti
- QEMU Tespiti
- Cuckoo Kum Havuzu Tespiti
- Anubis Kum Havuzu Tespiti
- ThreatExpert Kum Havuzu Tespiti

Pafish tespit aracı analiz sistemi tespitinde kullanılan araçlardan biridir. Pafish tespit aracında kullanılan tekniklerin eksik olduğu görüldüğü için Sems tespit aracı geliştirilmiştir.

5.2 Sems Tespit Aracı

Bu çalışma için oluşturulan tespit aracı Sems [24], C++ dili ile yazılmış olup Visual Studio ile derlenmiştir. İçerisinde birçok otomatik analiz sistemi ve sanallaştırma ortamı tespit yöntemi bulunan araç, akademik tezlerden ve bazı çalışmalardan faydalanılarak oluşturulmuştur.

Bu çalışmada belirtilen 3.1.4, 3.3.1, 3.3.2, 3.3.3, 3.3.5 başlıklarında anlatılanların bir kısmı ile diğer tüm başlıklarda yer alan yöntemler Sems tespit aracında kullanılmıştır. Tespit aracının gerçekleştirdiği aktiviteler şunlardır:

- Virtualbox Tespiti : 3.1.1, 3.1.4, 3.1.5 başlıklarında anlatılan tespit yöntemleri.
- VMWare Tespiti : 3.1.2, 3.1.4, 3.1.5 başlıklarında anlatılan tespit yöntemleri.
- QEMU Tespiti : 3.1.3 başlığında anlatılan tespit yöntemleri.
- Cuckoo Kum Havuzu Tespiti : 3.2, 3.3.4, 3.3.5 başlıklarında anlatılan tespit yöntemleri.
- Anubis Kum Havuzu Tespiti : 3.3.5 başlığında anlatılan tespit yöntemleri.
- ThreatExpert Kum Havuzu Tespiti : 3.3.5 başlığında anlatılan tespit yöntemleri.
- Sandboxie Tespiti : 3.2 başlığında anlatılan tespit yöntemleri.
- Analiz Araçları Tespiti : Immunity Debugger, Process Hacker, Process Explorer, Process Monitor, IDA, Wireshark, Regshot

5.3 Sems Tespit Aracı ile Gerçekleştirilen Kum Havuzu Tespit Testleri

Sems tespit aracı ile Cuckoo, Payload Security, Anubis, Threat Expert ve Comodo kum havuzlarında testler gerçekleştirilmiştir. Testler ile kum havuzlarının analiz sistemi tespit

yöntemlerine karşı ne kadar korunaklı olduğu gösterilmeye çalışılmıştır. Testlerin sonucunda bu kum havuzlarının tamamı tespit edilebilmiştir. Bu durum kum havuzlarının, otomatik analiz ortamı tespit yöntemlerine karşı tam olarak korunamadığını göstermektedir. Kum havuzları, analiz edeceği zararlı yazılımlar tarafından tespit edilirse doğru analiz sonuçları verememektedir.

Test edilen kum havuzlarından Comodo ve Threat Expert, Vmware sanallaştırma ortamını kullanmaktadır. Bu iki kum havuzu için gerçekleştirilen test sonucunda Comodo Kum havuzunda 3 farklı bölüm için analiz ortamı tespiti yapılmıştır. Bunlar kayıt defteri, dosya ve sihirli numara bölümleridir. Threat Expert kum havuzunda ise 6 farklı bölüm için analiz ortamı tespiti yapılmıştır. Bunlar servisler, kayıt defteri, dosya, sihirli numara, bilgisayar adı ve internet erişimi bölümleridir. Gerçekleştirilen test sonuçları Tablo 5.1 ve Tablo 5.2 gösterilmektedir.

Cuckoo ve Payload Security kum havuzları Virtualbox sanallaştırma ortamını kullanmaktadır. Bu kum havuzları için gerçekleştirilen testler sonucunda Cuckoo kum havuzunda 10 farklı bölümde analiz ortamı tespit edilmiştir. Tespit edilen bölümler mac adresi, araçlar, sistem bilgileri, dosya, servisler, paylaşılan klasör, kayıt defteri, çekirdek sayısı, disk boyutu ve kancalanan fonksiyonlardır. Payload Security kum havuzunda ise 5 farklı bölümde analiz ortamı tespit edilmiştir. Bunlar araçlar, sistem bilgileri, servisler, paylaşılan klasör bölümleridir. Gerçekleştirilen test sonuçları Tablo 5.3 ve Tablo 5.4 gösterilmektedir.

Anubis kum havuzu, analiz ortamı olarak QEMU kullanmaktadır. QEMU için gerçekleştirilen testin sonucunda 4 farklı bölüm için de analiz ortamı tespit edilmiştir. Tespit edilen bölümler sistem bilgileri, kayıt defteri, disk boyutu ve çekirdek sayısıdır. Test sonucu Tablo 5.5 gösterilmektedir.

Tablo 5.1: Comodo Kum Havuzunda Gerçekleştirilen Tespitler.

	Comodo [25]
Servis	-
İşlemler	-
Kayıt Defteri	HARDWARE\DEVICEMAP\Scsi\Scsi Port 1\Scsi Bus 1\Target Id 0\Logical Unit Id 0, HKLM\SOFTWARE\VMware, Inc.\VMware Tools, HKLM\SOFTWARE\Clients\StartMenuInternet\VMWAREHOSTOPEN.EXE
Dosya	C:\TEST\sample.exe
Giriş Çıkış Portu	+
Kullanıcı Adı	-
İnternet Erişimi	-
Disk Boyutu	-

Tablo 5.2: Threat Expert Kum Havuzunda Gerçekleştirilen Tespitler.

	Threat Expert [26]
Servis	vmmouse, VMTools
İşlemler	-
Kayıt Defteri	HARDWARE\DEVICEMAP\Scsi\Scsi Port 1\Scsi Bus 1\Target Id 0\Logical Unit Id 0, HKLM\SOFTWARE\VMware, Inc.\VMware Tools, HKLM\SOFTWARE\Clients\StartMenuInternet\VMWAREHOSTOPEN.EXE
Dosya	sample_1.exe
Giriş Çıkış Portu	+
Kullanıcı Adı	COMPUTERNAME
İnternet Erişimi	Erişim yok
Disk Boyutu	Küçük

Tablo 5.3: Cuckoo Kum Havuzunda Gerçekleştirilen Tespitler.

	Cuckoo [27]
Mac	+
Araçlar	Enum
Sistem Bilgileri	vboxbios
Dosya	agent.py
Servis	Vboxvideo
Guest	SOFTWARE\Oracle\VirtualBox Guest Additions
Kayıt Defteri	HARDWARE\ACPI\DSDT\VBOX__, HARDWARE\ACPI\FADT\VBOX__, HARDWARE\ACPI\RSDT\VBOX__
Çekirdek Sayısı	1 çekirdekli
Disk Boyutu	Küçük
Kancalanan Fonksiyonlar	+

Tablo 5.4: Payload Security Kum Havuzunda Gerçekleştirilen Tespitler.

	Payload Security [28]
Mac	-
Araçlar	Enum
Sistem Bilgileri	vboxbios
Dosya	sub.exe
Servis	-
Guest	SOFTWARE\Oracle\VirtualBox Guest Additions
Kayıt Defteri	SYSTEM_ControlSet001_Services_VBoxGuest, SYSTEM\ControlSet001\Services\VBoxMouse, SYSTEM\ControlSet001\Services\VBoxService, SYSTEM\ControlSet001\Services\VBoxSF, SYSTEM\ControlSet001\Services\VBoxVideo, SOFTWARE\Oracle\VirtualBox Guest Additions, HARDWARE\ACPI\DSDT\VBOX__, HARDWARE\ACPI\FADT\VBOX__, HARDWARE\ACPI\RSDT\VBOX__
Çekirdek Sayısı	-
Disk Boyutu	-
Kancalanan Fonksiyonlar	-

Tablo 5.5: Anubis Kum Havuzunda Gerçekleştirilen Tespitler.

	Sistem Bilgileri	Kayıt Defteri	Disk Boyutu	Çekirdek Sayısı
Anubis	+	+	+	+

Bölüm 6

Kum Havuzu Tespit Yöntemlerini Aşma Teknikleri

Belirlenen kum havuzu tespit yöntemlerinin birçoğunun atlatılabilmesi için daha önceden geliştirilmiş veya bu çalışmada geliştirilen teknikler aşağıda anlatılmaktadır. Bu yöntemler ile analiz ortamı tespitlerinden sanallaştırma ortamı, kum havuzu modülleri, çalışma zamanı, ürün numaraları, işlemci kontrolü, disk kontrolü, port numarası ve internet erişimi tespitleri atlatılabilmektedir.

6.1 Sanal Makine İzlerinin Kaldırılması

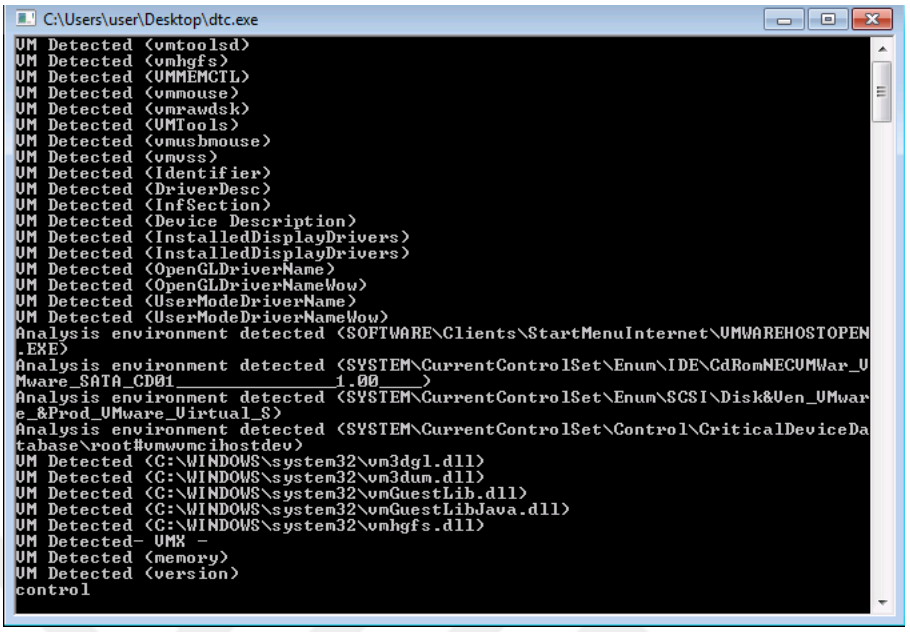
Sanal makine izleri otomatik analiz sistemlerinin tespit edilmesinde kullanılan en yaygın yöntemlerden biridir. Sanal makine izleri kaldırılarak ya da statik olmayacak bir şekilde değiştirilerek analiz sisteminin tespit edilmesi büyük ölçüde engellenebilmektedir.

Sems tespit aracı ile Vmware sanallaştırma ortamında bulunan 1 adet normal sanal makine ile 1 adet sanal izlerinin kaldırıldığı sanal makine karşılaştırılmıştır.

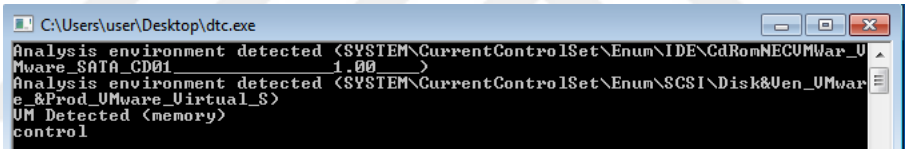
İlk olarak, bu çalışma için oluşturulan otomatik analiz sistemi tespit aracı, bu iki sanal makinede çalıştırılarak sonuçlar gözlemlenmiştir. Bunun sonucunda normal sanal makine-

de 30 tane tespit yöntemi doğru çalışırken sanal makine izlerinin kaldırıldığı sanal makinede

3 tane tespit yönteminin doğru çalıştığı gözlemlenmiştir. Bu sonuç sanal makine izleri kaldırılarak büyük ölçüde analiz ortamı tespitinin engellendiğini göstermektedir.



Şekil 6.1: Sems Aracının Normal Sanal Makinedeki Sonuçları.



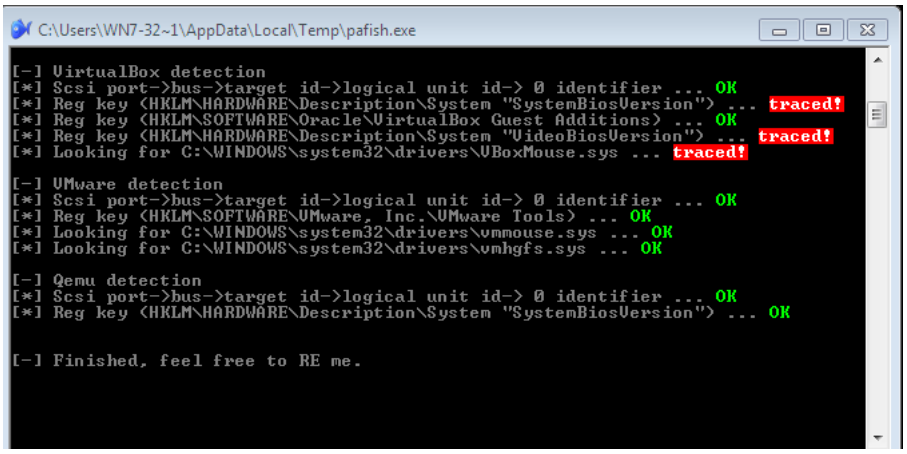
Şekil 6.2: Sems Aracının Değiştirilmiş Sanal Makinedeki Sonuçları.

Diğer bir test aracı olarak da Pafish sanal ortam tespit aracı kullanılmıştır. Sanallaştırma ortamı olarak VirtualBox kullanılmıştır.

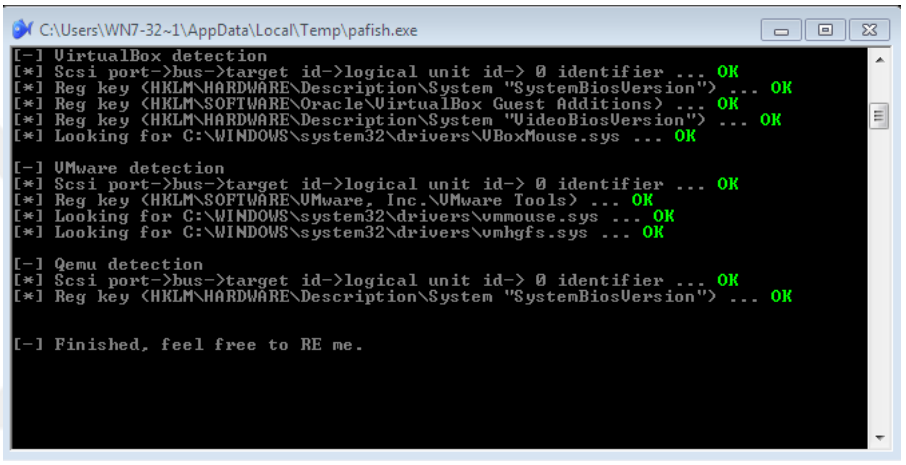
Pafish aracı ile yapılan test sonucunda normal sanal makinede yapılan 5 kontrolden 3 tanesinde tespit yöntemi doğru çalışırken sanal makine izlerinin kaldırıldığı sanal makinede 5 kontrolden hiçbiri doğru çalışmamaktadır. Bu test de sanal makine izlerinin kaldırılması ile tespit yönteminin büyük ölçüde atlatıldığını göstermektedir. İki sanal makine için sonuçlar aşağıda gösterilmektedir.

6.2 İnternet ve Port Kontrolü

Zararlı yazılımların, analiz ortamı tespitinde kullandıkları yöntemlerden biri de internet kontrolüdür. Zararlı yazılım, çalıştığı ortamda internetin varlığını kontrol ederek tespit



Şekil 6.3: Pafish Aracının Normal Sanal Makinedeki Sonuçları.



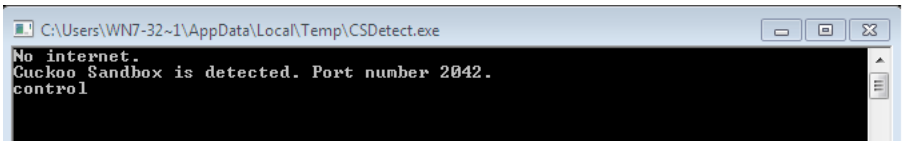
Şekil 6.4: Pafish Aracının Değiştirilmiş Sanal Makinedeki Sonuçları.

işlemi gerçekleştirebilmektedir. Bu tespit yöntemi, kum havuzlarının internet bağlantısı gerçekleştirilmesi sağlanarak aşılabilmektedir.

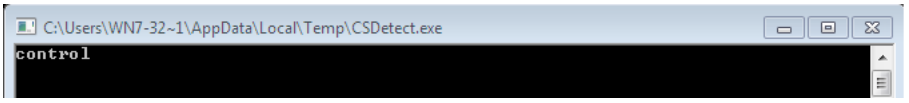
Bir diğer yöntem de açık port numaralarının kontrolüdür. Örnek olarak Cuckoo kum havuzu 2042 numaralı porttan sanallaştırma ortamı ile haberleşmektedir. Zararlı yazılımlar, bu port numarasından dolayı Cuckoo kum havuzunu tespit edebilmektedir. Bu tespit yönteminin aşılması, Cuckoo kum havuzunun sanallaştırma ortamı ile haberleşme portunu farklı bir numara yaparak gerçekleştirilebilmektedir.

Cuckoo kum havuzunda gerçekleştirilen internet ve port kontrol testinde, kullanılan tespit aracı otomatik analiz ortamını tespit edebilmektedir.

Sanallaştırılmış ortam için internet bağlantısı kullanılabilir duruma getirildiğinde ve Cuckoo kum havuzunun kullandığı port numarası değiştirildiğinde otomatik analiz ortamı tespit edilememektedir.



Şekil 6.5: Sems Aracının Normal Cuckoo Kum Havuzundaki Sonuçları.



Şekil 6.6: Sems Aracının Değiştirilmiş Cuckoo Kum Havuzundaki Sonuçları.

6.3 DLL Dosyalarının Gizlenmesi

Otomatik analiz sistemleri, analiz işlemlerinde özel DLL dosyaları kullanmaktadır. Bu DLL dosyalarını analizi yapılacak yazılıma enjekte etmektedirler. Zararlı yazılımlar, otomatik analiz sistemlerinde analiz edilmelerini engellemek için kum havuzlarının kullandıkları özel DLL dosyalarının kendi adres alanlarına yüklenip yüklenmediklerini kontrol etmektedir. Zararlı yazılımların bu DLL dosyalarını fark etmelerini engellemek için birkaç yöntem kullanılabilmektedir.

6.3.1 DLL Referans Sayısı

DLL referans sayısı, hedef işleme yüklenen DLL sayısını ifade eden bir değerdir. İşleme her DLL yüklenmesinde bu değer 1 artarken DLL serbest bırakılmasında 1 azalmaktadır. DLL yükleme işlemi LoadLibrary fonksiyonu ile gerçekleşirken serbest bırakma işlemi FreeLibrary fonksiyonu ile gerçekleşmektedir. Eğer referans sayısı 0 ise DLL dosyası işlemden tamamen kaldırılabilmektedir [29].

Kum havuzları tarafından zararlı yazılıma, analiz işlemi için enjekte edilen DLL dosyası, zararlı yazılımlar tarafından FreeLibrary fonksiyonu kullanılarak kaldırılabilmektedir. Kum havuzu, LoadLibrary fonksiyonu ile DLL dosyasını zararlı yazılıma enjekte ettiğinde referans sayı değeri 1 olmaktadır. Zararlı yazılım FreeLibrary fonksiyonunu kullandığında referans sayı değeri 0 olmakta ve DLL dosyası zararlı yazılımdan tamamen kaldırılmaktadır. Bu duruma engel olmak için kum havuzları tarafından DLL referans sayı değeri artırılabilmektedir. Kum havuzları LoadLibrary fonksiyonunu birden fazla kullandığı zaman referans sayı değeri 1 değerinden büyük olmaktadır. Zararlı yazılım DLL dosyasını

kaldırmak istediğinde FreeLibrary fonksiyonunu kullandığı zaman referans sayı değeri 0 olmayacağından DLL dosyası zararlı yazılımdan kaldırılamayacaktır [15].

6.3.2 Modülleri Saklamak

Bir işlemde yüklü olan DLL dosyaları PEB bloğundan elde edilebilmektedir. PEB bloğunda bulunan PPEB_LDR_DATA yapısında 3 tane birbirine bağlı liste(Linked list) vardır. Bunlar, işleme yüklenmiş modüllerin bilgilerini tutmaktadır. İşleme yüklü bir DLL dosyası için GetModuleHandleA fonksiyonu çağırıldığı zaman DLL hakkında bilgiler bu listelerden dönmektedir.

Zararlı yazılımlar, kum havuzu tarafından bir DLL yüklenip yüklenmediğini GetModuleHandleA fonksiyonunu çağırarak tespit edebilmektedir. Fonksiyondan dönen değer null değilse zararlı yazılım, kum havuzu tespiti yapabilmektedir. Bu duruma engel olabilmek için DLL dosyalarını saklamak çözüm olabilmektedir. DLL dosyalarını saklamak için PEB bloğundaki PPEB_LDR_DATA yapısında yer alan bağlı listelerin bağının çözülmesi(unlink) gerekmektedir. Bağ çözme işlemi için DllMain içerisinde DLL_PROCESS_ATTACH kısmında DLL saklama fonksiyonu yazılır. Daha sonra kum havuzu DLL dosyasını LoadLibrary fonksiyonu ile yüklemeyi gerçekleştireceği zaman, yazılan saklama fonksiyonu da çalışır ve saklama işlemi gerçekleşmiş olur. Bu listelerin birbiriyle olan bağı çözüldüğü zaman GetModuleHandleA fonksiyonundan null değeri dönecek ve zararlı yazılım kum havuzu DLL dosyası tespiti yapamayacaktır [15].

6.3.3 Farklı DLL Adları

Otomatik analiz sistemlerinin kullandıkları özel bazı DLL dosyaları sabit isimlere sahiptir. Zararlı yazılımlar, analiz sistemlerinin kullandıkları bu DLL dosyalarının varlıklarını kontrol ederek analiz sistemi tespiti yapabilmektedir. Bu tespit yöntemini atlatabilmek için analiz sistemlerine ait özel DLL dosyalarına sabit isimler yerine farklı isimler verilebilmektedir. Böylelikle analiz sistemlerinin kullandığı özel DLL dosyalarının isimlerine göre bir kontrol mekanizması, analiz sistemi tespitinde başarısız sonuç verebilmektedir.

6.4 Fiziksel Kısımların Gizlenmesi

Otomatik analiz sistemlerindeki sanallaştırma ortamlarının kullandıkları fiziksel kısımlar kontrol edilerek analiz sistemi tespit edilebilmektedir. Bu kontrol zararlı yazılımların analiz sistemi tespiti için kullandıkları yaygın yöntemlerdendir.

Sanallaştırma ortamlarının, tespit için kullanılan fiziksel kısımlarına MAC adresi, bios ayarları, sistem, anakart, işlemci bilgileri, çekirdek sayısı, disk boyutu örnek olarak gösterilebilir. Zararlı yazılımların bu bilgileri kontrol ederek sanallaştırma ortamını tespit etmelerini engelleyebilmek için 2 yöntem bulunmaktadır.

- Fiziksel kısımların bilgileri değiştirilebilir. Örnek olarak sanallaştırma ortamının MAC adresini değiştirmek için MAC adresi klonlanarak bir yere kaydedilir. Daha sonra da rastgele sayılardan oluşan bir MAC adresi oluşturularak sanallaştırma ortamına bu değer atanabilir. Böylelikle tespit yönteminde bulunan MAC adresinden analiz sistemi tespiti başarısız olacaktır.
- Fonksiyonlardan dönen değerler değiştirilebilir. Zararlı yazılımlar yukarıda anlatılan bilgileri kontrol etmek için belirli fonksiyonları kullanabilmektedirler. Bu fonksiyonlara kancalama işlemi yapılarak fonksiyonlar çağrıldığı zaman dönecek değer manipüle edilerek sanallaştırma ortamlarına ait olmayan bir değer döndürülür. Bu sayede zararlı yazılım sanallaştırma ortamı tespiti yapamamaktadır [30].
- VmWare Workstation sanallaştırma ortamındaki vmx dosyasında bazı eklentiler ile sanallaştırma ortamına ait fiziksel izler kaldırılabilmektedir. vmx dosyasına aşağıdaki değerler eklenerek fiziksel kısımların tespiti engellenebilmektedir [31].
  - isolation.tools.getPtrLocation.disable = "TRUE"
  - isolation.tools.setPtrLocation.disable = "TRUE"
  - isolation.tools.setVersion.disable = "TRUE"
  - isolation.tools.getVersion.disable = "TRUE"
  - monitor_control.disable_directexec = "TRUE"
  - monitor_control.disable_chksimd = "TRUE"
  - monitor_control.disable_ntreloc = "TRUE"
  - monitor_control.disable_selfmod = "TRUE"

- monitor_control.disable_reloc = "TRUE"
- monitor_control.disable_btinout = "TRUE"
- monitor_control.disable_btmemspace = "TRUE"
- monitor_control.disable_btpriv = "TRUE"
- monitor_control.disable_btseg = "TRUE"

6.5 Çalışma Zamanı Yönteminin Atlatılması

Kum havuzları gerçekleştirdikleri analizler için maksimum bir çalışma zamanı belirlemektedir. Analiz, belirlenen maksimum zamana kadar bitirilemezse zaman aşımı olarak sonlandırılmaktadır. Zararlı yazılımlar maksimum çalışma zamanı faktörünü analizi engellemek için kullanabilmektedir. Çalışma zamanı yöntemi ile analizi engellemek için zararlı yazılımlar çalışmalarına başlarken, ilk olarak sleep fonksiyonu kullanmaktadır. Sleep fonksiyonuna parametre olarak zaman verilmektedir. Bu parametreye kum havuzlarının analiz süresinin tamamını kapsayacak bir zaman verilip, analiz işlemleri zaman aşımına maruz bırakılarak zararlı yazılımın analizi gerçekleşmemektedir.

Zararlı yazılımların bu engelleme yöntemini atlatmak için kullandıkları sleep fonksiyonları değiştirilebilmektedir [32].

Zararlı yazılım çalıştırıldığında otomatik analiz sisteminin kancalamaya çalıştığı bazı fonksiyonlar bulunmaktadır. Bu fonksiyonlardan bir tanesi de sleep fonksiyonudur. Zararlı yazılım, sleep fonksiyonunu ana işlemlerden önce çağırdığı zaman analiz sistemi kancalama işlemini gerçekleştirmekte ve kancalama durumunda gerçekleştireceği işlemler aktif olmaktadır. Kancalama yapıldığında, kullanılan sleep fonksiyonu işleme tabi tutulmadan geçilmektedir. Böylelikle zamana göre analizi engelleme yöntemi atlatılabilmektedir.

Bölüm 7

Sanallaştırma Ortamı Tespitlerini Atlatma Tekniklerinin Testi

Kum havuzlarında çalışan sanallaştırma ortamlarını tespit ederek analiz edilmelerini engelleyen zararlı yazılımlar bulunmaktadır. Bu zararlı yazılımlar sanallaştırma ortamı tespit yöntemlerini kullanmaktadır. Zararlı yazılımların, doğru bir şekilde analiz edilmeleri için bu tespit yöntemlerinin atlatılması ve sanallaştırma ortamı tespiti gerçekleştiremiyor olmaları gerekmektedir. Bu sebeple sanallaştırma ortamları tespit yöntemlerini atlatabilecek veya engelleyebilecek teknikler geliştirilmiştir. Bu tekniklerin, tespit yöntemlerini ne kadar atlatabildiğini ya da engelleyebildiğini gözlemlemek için Cuckoo kum havuzu ile gerçek zararlı yazılımlar analiz edilerek testler gerçekleştirilmiştir.

Gerçekleştirilecek testler için VirusShare internet sitesinden 2015 yılının başlarında ortaya çıkan zararlı yazılımlar ele alınmıştır. Bu zararlı yazılımlarından da Virüstotal sonuçlarında antivirüs firmalarının en az %50' si tarafından zararlı olarak tanımlanan 1065 adet zararlı yazılım testlerde kullanılmıştır.

Testler ile normal sanal makinede ve tespit yöntemlerini aşma ya da engelleme için kullanılabilecek tekniklerin uygulandığı sanal makinede, belirlenen 1065 adet zararlının analizi gerçekleştirilmiştir. Analiz sonuçları arasındaki farklar karşılaştırılarak kullanılan tekniklerin etkileri gözlemlenmiştir.

7.1 Test Ortamı

Testin gerçekleştirileceği ana makine Ubuntu 12.04 LTS işletim sistemi kurulu bir bilgisayardır. Bu bilgisayara Cuckoo Sandbox 1.2 kurulumu gerçekleştirilmiştir. Cuckoo kum havuzunun analizlerini yapacağı sanallaştırma ortamı olarak Virtualbox seçilmiş ve test bilgisayarına Virtualbox sanallaştırma ortamı kurulmuştur. Kurulan sanallaştırma ortamında iki adet sanal makine oluşturulmuştur. Sanal makinelerde zararlı yazılımların ihtiyaç duyabileceği Adobe 9, Microsoft Office 2007, .net Framework 4, Python yazılımları yüklenmiştir. Sanal makinelerden birinde 4.1.1 bölümünde belirtilen Virtualbox sanal makine izleri kaldırılmış diğerinde ise herhangi bir değişiklik yapılmamıştır.

7.2 Test Edilen Örnekler

Test için gerçek zararlı yazılımlar kullanılmak istenmiştir. Bu sebeple zararlı yazılım paylaşım adreslerinden VirusShare' den 2015 yılının başlarında tespit edilen virüsler alınmıştır. Bu virüslerden, Virüstotal sonuçlarında antivirüs firmalarının %50' si tarafından virüs olarak belirlenen ve ".exe" uzantılı olanlar seçilmiştir. Yapılan bu filtreleme sonucunda 1065 tane virüs iki sanal makinede de analiz edilmiştir.

7.3 Test Değerlendirme Kriterleri

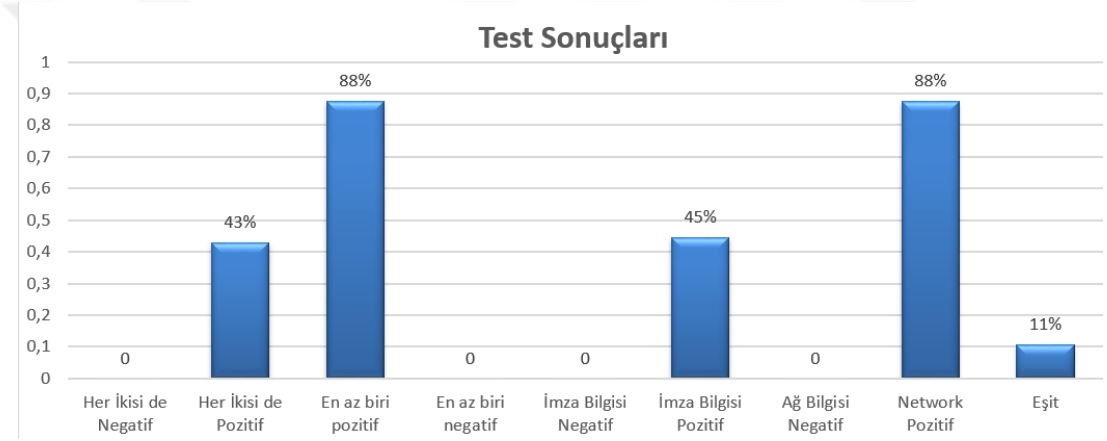
Test sonuçları iki nokta temel alınarak değerlendirilmiştir. Bunlar imzalar ve ağ bilgisi olarak belirlenmiştir. İmzalar, zararlı yazılımın gerçekleştirdiği davranışlardan önemli olarak belirlenen bilgileri özet şeklinde ifade etmektedir. Ağ bilgisi ise zararlı yazılımın gerçekleştirdiği ağ hareketlerini belirten bilgilerdir.

Zararlı yazılımlar, analiz edilmelerini engellemek için sanallaştırma ortamı tespiti yapmaktadırlar. Sanallaştırma ortamı tespiti yapan zararlı yazılımlar, aslından farklı davranış sergilemekte ya da çalışmalarını sonlandırmaktadırlar. Bu sebeple zararlı yazılımın sanallaştırma ortamı tespit ettiği ile edemediği durumdaki sonuçlar farklı olmaktadır. Bu durum göz önüne alınarak normal ile sanal makine izleri kaldırılan sanal makinede yapılan analizlerde imza ve ağ bilgisi sonuçları karşılaştırılarak aradaki farklar göz önüne alınmıştır.

Sanal makine izleri kaldırılan makinede imza veya ağ bilgisi sonuç sayısı normal sanal makinedekinden daha fazla ise bu durum pozitif, sayılar eşit ise eşit, daha az ise negatif olarak belirlenmiştir.

7.4 Test Sonuçları

Gerçekleştirilen testler sonucunda hiçbir analizde imza veya ağ bilgisi sayısında negatif bir durum oluşmamıştır. Analizlerin %43' ünde imza ve ağ bilgisi sonuçlarının her ikisi, %45' inde imza bilgisi, %88' inde ağ bilgisi, %88' inde en az biri pozitif sonuç verirken %11' inde ise eşit sonuçlar alınmıştır.



Şekil 7.1: Test Sonuçları.

Pozitif sonuç alınan belirli sayıdaki örnekler elle analiz edildiğinde sanal makine kontrolü gerçekleştirdikleri gözlemlenmiştir. Elle analiz edilen zararlı yazılımlardan MD5 özet değeri 1e7d76c470ad7761cdf172fac6f0152b olan zararlı yazılım Product Name değerini kontrol etmektedir. de1af0e97e94859d372be7fcf3a5daa5 MD5 özet değerine sahip bir başka zararlı yazılım da Bios değerlerini kontrol ederek sanallaştırma ortamı tespiti etmeye çalışmaktadır.

Elde edilen sonuçlara göre sanallaştırma ortamı tespitini engellemeye yönelik belirtilen yöntemler kullanıldığında analiz çıktılarında %43 daha fazla sonuç elde edildiği gözlemlenmiştir. Alınan bu sonuç, analiz edilen zararlı yazılımların en az %43' ünün sanallaştırma ortamı tespiti yaptığını göstermektedir. Bu durum zararlı yazılım analizinde sanallaştırma ortamı tespit yöntemlerinin atlatılması ya da engellenmesinin ne denli önemli olduğunu ispatlamaktadır. Sanallaştırma ortamı tespit yöntemleri atlatılamaz

ya da engellenemez ise bahsi edilen zararlı yazılımların analizlerinden doğru sonuçlar elde edilemeyecektir. Zararlı yazılım analizlerinde doğru sonuca varılamadığı taktirde bu zararlı yazılımlara karşı ne tür önlemler alınabileceği belirlenemez ve zararlı yazılımların verdiği ya da vereceği zararlar engellenemez.

Sonuç olarak sanallaştırma ortamı tespit yöntemlerinin atlatılması zararlı yazılım analizi ve zararlı yazılımlara alınabilecek önlemlerin belirlenmesinde büyük önem ifade etmektedir. Gerçekleştirilen testlere göre sanallaştırma ortamı tespit yöntemlerini atlatabilmek için bu çalışmada belirtilen teknikler kullanılabilir. Böylelikle zararlı yazılım analizlerinde daha doğru sonuçlar alınabilecektir. Test sonuçları bu durumu ispatlar niteliktedir.

Bölüm 8

Sonuç

8.1 Sonuç

Zararlı yazılımların sayısı ile birlikte zararlarının hızla artması bunlara karşı önlemlerin alınmasını gerekli kılmıştır. Alınabilecek önlemlerden biri de otomatik analiz sistemleridir. Otomatik analiz sistemlerinin amacı, zararlı yazılımları analiz etmek ve doğru sonuçlar vererek el ile gerçekleştirilecek zararlı yazılım analizine yardımcı olmak ve hız kazandırmak, zararlı yazılımları ve önceden tespiti yapılmamış olan saldırıları tespit ederek bunların engellenmesine ve temizlenmesine katkıda bulunmaktır.

Otomatik analiz sistemlerinin doğru sonuç vermelerini engelleyecek yöntemler bulunmaktadır. Bu yöntemler ile zararlı yazılımlar otomatik analiz sistemi tespiti yapmaktadır. Otomatik analiz sistemi tespiti gerçekleştiren zararlı yazılımlar, çalışmalarını sonlandırmakta ya da aslından farklı davranışlarda bulunmaktadır. Bu durum otomatik analiz sistemlerinin işlevselliğini bozmaktadır.

Bu çalışmada, otomatik analiz sistemi olan kum havuzlarının çalışmasını ya da doğru sonuçlar vermesini engelleyebilecek yöntemler ele alınıp detaylandırılmıştır. Bu yöntemler;

- Sanallaştırma ortamı tespiti.
- Bilgisayar özellikleri.
- Kum havuzu dosyaları izi.

- İnternet ve port bilgileri.
- Kum havuzu modülleri.
- Kullanıcı kontrolü.

Çalışmanın etkin olabilmesi için kum havuzu tespit yöntemlerinin kullanıldığı Sems adında bir tespit aracı geliştirilmiştir. Bu tespit aracı kullanılarak kum havuzu tespiti gerçekleştirilmektedir. Sems tespit aracı ile birlikte, kum havuzlarında var olan tespit açıklıkları belirlenerek gerekli önlemler belirlenebilmektedir.

Kum havuzu tespit yöntemleri ayrıntıları ile ele alınarak bu yöntemleri atlatma teknikleri geliştirilmiştir. Halihazırda bazı tespit yöntemlerini atlatma teknikleri bulunmaktadır. Fakat bunlardan bazıları yetersiz olmakla birlikte bazıları da başarılı sonuç verememektedir. Bu çalışmada bu teknikler geliştirilmiş ve bunlarla birlikte yeni teknikler keşfedilmiştir.

Kum havuzlarını tespit etmeye yönelik gerçekleştirilen Sems tespit aracı ile 5 adet farklı ve herkese açık kum havuzlarında testler gerçekleştirilmiştir. 5 farklı kum havuzunda gerçekleştirilen testlerin sonucunda, kum havuzlarının hepsinde analiz sistemlerine ait bulgular tespit edilmiştir.

Sems tespit aracı ile belirtilen 5 kum havuzunda otomatik analiz sistemi tespiti için açıklıklar bulunmuştur. Bu durum, test edilen kum havuzlarının, tespit aracında bulunan yöntemleri kullanan zararlı yazılımların analizinde doğru sonuç veremeyeceğini göstermektedir. Analiz sonuçlarının doğru olabilmesi için tespit yöntemlerinin atlatılması gerekmektedir. Böylelikle zararlı yazılımlar normal aktivitelerini gerçekleştirecektir.

Otomatik analiz sistemlerinin tespitinin engellenebilmesi için kullanılabilecek teknikler;

- DLL dosyalarının gizlenmesi.
- DLL dosyalarının kaldırılmasının engellenmesi.
- DLL dosyalarının isimlerinin değiştirilmesi.
- Kum havuzu dosyalarının isimlerinin değiştirilmesi.
- Sanallaştırma ortamlarının izlerinin kaldırılması.
- Kullanılan bilgisayarın gerçek bilgisayar konumuna getirilmesi.

- Gerçek kullanıcı aktivitelerinin gösterilmesi.

Tespitleri engelleyebilmek için belirtilen bu teknikler doğrultusunda tam olarak erişim sağlanabilen Cuckoo kum havuzunda gerekli değişiklikler gerçekleştirilmiştir. Daha sonra tespit aracı, Cuckoo kum havuzunda çalıştırılarak sonuçlar incelenmiştir. Bu test sonucunda %100 başarı elde edilmiş ve Cuckoo kum havuzu tespit edilememiştir.

Bu çalışma sonucunda belirtilen teknikler kum havuzlarına uygulandığı zaman kum havuzlarını tespit etmek için kullanılabilecek yöntemlerin atlatılabildiği görülmüştür. Bu sayede analiz sistemlerinin doğru çalışabilmesi sağlanabilecek ve gerçekleştirilecek zararlı yazılım analizlerinde daha yararlı sonuçlar alınabilecektir.

8.2 Neler Yapılabilir?

Kum havuzlarının, zararlı yazılımlar tarafından tespit edilmesini engelleyip doğru analiz çıktıları üretmeleri zararlı yazılım ile mücadele çok önemlidir. Bu bağlamda kullanabilecek tekniklerden çalışmada bahsedilmiştir. Bu tekniklere ek olarak yarar sağlayabileceği düşünülen bazı teknikler de bulunmaktadır. Yarar sağlayabileceği düşünülen bu teknikler aşağıda listelenmektedir:

- Kum havuzlarının kullandıkları ortamlar zararlı yazılımların analiz sistemi tespiti için en çok kontrol ettikleri alanlardır. Kum havuzları tam sistem emülasyon ortamları kullanarak ve bu ortamlara ait fiziksel izleri kaldırarak bu tespiti atlatabilirler.
- Zararlı yazılımlar tarafından kum havuzlarına ait özel dosyaların varlığı kontrol edilerek analiz ortamı tespiti yapılabilmektedir. Bu tespit yönteminin atlatılabilmesi için kum havuzlarının kendilerine ait dosyaların isimleri ve özet değerleri sabit olmamalıdır. Bu dosyalar dinamik bir yapıda olacak şekilde düzenlenebilirse tespit yönteminin atlatılabileceği düşünülmektedir.
- Kum havuzlarının en fazla eksik oldukları yer gerçek kullanıcı işlemleri gerçekleştirmemesidir. Çalışmak için güvenlik kodu isteyen zararlı yazılımlar için güvenlik kodunu doğru bir şekilde sağlayacak programlar geliştirilerek mevcut eksiklik giderilebilir.

Çalışmada belirtilen analiz sistemi tespit yöntemlerini atlatma teknikleri ile birlikte bu bölümde anlatılan teknikler, analiz sistemi tespitinin engellenmesi ya da atlatılmasında büyük yararlar sağlayacağı düşünülmektedir. Böylelikle kum havuzu ile gerçekleştirilen zararlı yazılım analizlerinde daha doğru sonuçlar alınabilecek ve zararlı yazılımlar ile mücadele daha etkili bir şekilde yapılabilecektir.

Kaynaklar

[1] Malware, 2015. URL https://www.av-test.org/en/statistics/malware/.

[2] T. P. Wong. Active cyber defense: Enhancing national cyber defense. Master's thesis, Naval Postgraduate School, California, U.S., 2011.

[3] R. R. Ravula. Classification of malware using reverse engineering and data mining techniques. Master's thesis, The University of Akron, Akron, U.S., 2011.

[4] L. Bilge. Generating content-based signatures for detecting bot-infecting machines. Master's thesis, Bilkent University, Ankara, Turkey, 2008.

[5] G. O'Gorman and G. McDonald. Ransomware: A growing menace. whitepaper, Symantec, 2012.

[6] C. Kruegel. Full system emulation: Achieving successful automated dynamic analysis of evasive malware. whitepaper.

[7] Automated malware analysis - cuckoo sandbox. URL http://cuckoosandbox.org/.

[8] Anubis - malware analysis for unknown binaries, 2015. URL https://anubis.iseclab.org/.

[9] Comodo automated analysis system, 2015. URL http://camas.comodo.com/.

[10] Threatexpert. URL http://www.threatexpert.com/.

[11] Vxstream sandbox - automated malware analysis system. URL http://www.payload-security.com/.

[12] A. A. Masjedi. A study on the performance of virtualization programs. Master's thesis, Auckland University, Auckland, New Zeland, 2012.

[13] A. Pektaş. *Behavior based malware classification using online machine learning.* PhD thesis, Universite De Grenoble, Grenoble, France, 2015.

[14] C. Kolbitsch. Does dyre malware play nice in your sandbox?, 2015. URL `http://labs.lastline.com/dyre-malware-does-it-play-nice-in-your-sandbox`.

[15] S. Singh. Breaking the sandbox. whitepaper.

[16] A. Singh. Defeating darkhotel just-in-time decryption, 2015. URL `http://labs.lastline.com/defeating-darkhotel-just-in-time-decryption`.

[17] S. Sharma N. Bilogorskiy. Mmw anti-sandbox techniques, 2015. URL `http://www.slideshare.net/Cyphort/mmw-antisandbox-techniques`.

[18] A. Ortega. Your malware shall not fool us with those anti analysis tricks, 2012. URL `https://www.alienvault.com/open-threat-exchange/blog/your-malware-shall-not-fool-us-with-those-anti-analysis-tricks`.

[19] M. Marschalek. Sazoora:dissecting the bundle of evasion and stealth, 2014. URL `http://www.cyphort.com/sazoora-dissecting-bundle-evasion-stealth/`.

[20] A. Singh S. O. Vashisht. Turint test in reverse: New sandbox-evasion techniques seek human interaction, 2014. URL `https://www.fireeye.com/blog/threat-research/2014/06/turing-test-in-reverse-new-sandbox-evasion-techniques-seek-human-interaction.html`.

[21] U. Bayer. A platform for the analysis of malicious code. whitepaper, Secure Systems Lab, Technical University Vienna.

[22] T. Ueltschi. My name is hunter, ponmocup hunter. Technical report, SANS, 2013.

[23] A. Ortega. Pafish, 2015. URL `https://github.com/a0rtega/pafish`.

[24] A. Akyol. sems, 2016. URL `https://github.com/AlicanAkyol/sems`.

[25] Malware analysis report, 2015. URL `http://camas.comodo.com/cgi-bin/submit?file=8a7962a180d09fe3274c09abe4eb9182b500360cb72ef2f1070226db4c01e699`.

[26] Threat expert, 2016. URL `http://www.threatexpert.com/report.aspx?md5=03d2fa198a2654a358a6ab7723583815`.

[27] malwr, 2015. URL https://malwr.com/analysis/OTJmMDlhOWViMjlhNGY1MDgzNmM5ZDMzZGZlZjI2ZDg/.

[28] sub.exe, 2015. URL https://www.hybrid-analysis.com/sample/3a5481d105673bf20256512c9a32b60e946a240c1793e2603c226c788f234055?environmentId=1.

[29] The covert way to find the reference count of dll. URL http://www.securityxploded.com/dllrefcount.php.

[30] M. Boman. Making virtualbox nearly undetectable, 2014. URL http://blog.michaelboman.org/2014/01/making-virtualbox-nearly-undetectable.html.

[31] Malware anti-vm technics, 2012. URL http://www.simonganiere.ch/2012/11/20/malware-anti-vm-technics/.

[32] Cuckoomon, 2015. URL https://github.com/cuckoosandbox.