T.C.
GEBZE TEKNİK ÜNİVERSİTESİ

Bilgisayar Mühendisliği Bölümü

Kötücül Yazılım Tespit ve Engelleme Aracı

Gulzada IISAEVA

Danışman
Prof. Dr. İbrahim SOĞUKPINAR

2019
Gebze, KOCAELİ

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ

Bilgisayar Mühendisliği Bölümü

Kötücül Yazılım Tespit ve Engelleme Aracı

Gulzada IISAEVA

Danışman
Prof. Dr. İbrahim SOĞUKPINAR

2019
Gebze, KOCAELİ

Bu çalışma ..../..../200.. tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Bölümü'nde Lisans Bitirme Projesi olarak kabul edilmiştir.

Bitirme Projesi Jürisi

Danışman Adı	İbrahim SOĞUKPINAR	
Üniversite	Gebze Teknik Üniversitesi	
Fakülte	Bilgisayar Mühendisliği	

Jüri Adı	Didem GÖZÜPEK	
Üniversite	Gebze Teknik Üniversitesi	
Fakülte	Bilgisayar Mühendisliği	

Jüri Adı		
Üniversite		
Fakülte		

ÖNSÖZ

Bu kılavuzun ilk taslaklarının hazırlanmasında emeği geçenlere, projenin son halini almasında yol gösterici olan Sayın Prof. Dr. İbrahim SOĞUKPINAR hocama ve bu çalışmayı yapmam için bana desteğini esirgemeyen Mert NAR ve Arzu Görgülü KAKIŞIM hocalarıma en içten teşekkürlerimi sunarım.

Ayrıca eğitimim süresince bana her konuda tam destek veren aileme ve bana hayatlarıyla örnek olan tüm hocalarıma saygı ve sevgilerimi sunarım.

Mayıs, 2019

Gulzada IISAEVA

İÇİNDEKİLER

ŞEKİL LİSTESİ 2
KISALTMA LİSTESİ 3
ÖZET 4
1. GİRİŞ 6
1.1. PROJE TANIMI 6
1.2. PROJENİN NEDEN VE AMAÇLARI 7
1.3. PROJE İLE İLGİLİ ÇALIŞMALAR 7
1.3.1. MAIL DİLİ KULLANILARAK KÖTÜCÜL KOD TESPİTİ 8
1.3.2 GİZLİ MARKOV MODEL (HİDDEN MARKOV MODEL (HMM)) 9
1.3.3. AĞIRLIKLI ÇİZGE VE KOMŞULUK MATRİSİ 9
1.3.4. İŞLEM KOD SIKLIĞINI HİSTOGRAM ÜZERİNDE TEMSİL ETME 10
2. PROJE TASARIMI 12
2.1. PROJE GEREKSİNİMLERİ 12
2.2.SİSTEM MİMARİSİ 12
2.2.1. İMZA TABANLI TESPİTİ: 12
2.2.2. YENİ METAMORFİK ZARARLI YAZILIM TESPİTİ 13
2.2.2.1.TEST 13
2.3. USE CASE DİYAGRAMI 14
4. GERÇEKLEME VE TEST 15
4.1.NOVEL METAMORFİK KÖTÜCÜL YAZILIM TESPİTİ 15
4.1.2.EĞİTİM 15
4.1.3.TEST 16
4.2.İMZA TABANLI TESPİT 17
5. SONUÇ 18

ŞEKİL LİSTESİ

ŞEKİL 1:PROJE TASARIMI 7
ŞEKİL 2:MARKOV MODEL 9
ŞEKİL 3:-İŞLEM KOD ÇIKARIMI 10
ŞEKİL 4:İŞLEM KOD HİSTOGRAM 10
ŞEKİL 5:ÖKLİD HİSTOGRAM MESAFESİ METRİĞİ 11
ŞEKİL 6: SİSTEM MİMARİSİ 14
ŞEKİL 7:USE CASE DİYAGRAMI 14
ŞEKİL 8: UNİQUE KOD ÖRNEĞİ 15
ŞEKİL 9: KOMŞULUK MATRİSİ ÖRNEĞİ 15
ŞEKİL 10: GRAFIN EN UZUN YOL ÖRNEĞİ [1] 16
ŞEKİL 11: UNİQUE CODE VEKTÖRÜNE GÖRE İŞARETLEME ÖRNEĞİ 16
ŞEKİL 12: VİRÜS YAKALANDI 17
ŞEKİL 13: VİRÜSÜ KARANTİNE ALMA 17
ŞEKİL 14: SİGNATURE DATABASE VE BLACKLİST 18

KISALTMA LİSTESİ

GTÜ : Gebze Teknik Üniversitesi
MongoDB : MongoDB Veri Tabanı
PE : Portable Executable
TF : Term Frequency (Terim Sıklığı)
IDF : Inverse Document Frequency (Devrik Belge Sıklığı)
MAIL : Malware Analysis Intermediate Language(Malware Analizi Ara Dil)
MD5 : message-digest algorithm
NGVCK : Next Generation Virus Creation Kit
G2 : Second Generation Virus Kit
MPCGEN : Mass Code Generation

ÖZET

Bu raporda GTÜ Bilgisayar Mühendisliği bölümü BİL 496 dersi kapsamında geliştirilen Kötücül Yazılım Tespit ve Engelleme Aracı adlı projemiz açıklanmaya çalışılacaktır.

Günümüzde milyonlarca insan aktif olarak internet kullanmaktadır. Bu kullanıma bağlı olarak sıkça karşılaşılan problem veri güvenliği ve gizliliği olmuştur. İnternetteki bir çok yazılım, sistemi etkileyen veya uzak sunuculara bilgi sızdıran kötü amaçlı kodlar barındırmaktadır. Bu kötücül kodların sayısında, karmaşıklığında ve çeşitliliğinde hızlı artış, mevcut anti-virus sistemleri tarafından sıklıkla kullanılan bilinen kötücül kodların tespitine yönelik geliştirilmiş imza tabanlı yöntemini gerçekleştirmemize ve büyük veri kaynak kod analizine dayalı makine öğrenmesi destekli yeni Novel Metamorfik Köütücül yazılım tespiti yöntemin geliştirilmesine ihtiyaç doğrumuştur. [1]

Bu çalışmadaki amacımız kötücül kodların sistemimize zarar vermesine izin vermeden kaynak kodu incelenerek tespit edilmesidir. Bu çalışma kapsamında geliştirdiğimiz yöntemlerde Gtü zararlı yazılım kütüphanesinden veri seti kullanılmıştır. Yeni tespit edilen kötücül kodlar veritabanımızda kaydedilerek, kötücül kod tespitinin daha da hızlandırılması sağlanmıştır.

SUMMARY

In this report, our project named "Malware Detection and Prevention Tool" under the scope of GTÜ Computer Engineering department BIL 496 will be explained.

Today, millions of people actively use the Internet. The common problem with this use is data security and confidentiality. Many software on the Internet contain malicious codes that affect the system or leak information to remote servers. The need for the development of the new Novel Metamorphic Malware Detection Method based on large data source code analysis and the development of the new signature-based method for detecting known malicious codes frequently used by the existing anti-virus systems in the number of malicious codes and the rapid increase in the diversity of these malware codes. [1]

Our aim in this study is to detect the malicious code by examining the source code without letting it damage our system. In this study, a data set was used from the Gtü malware library. Newly identified malicious codes were recorded in our database to further accelerate malware detection.

1. GİRİŞ

Bilgisayar teknolojilerinin gelişmesi ile son zamanlarda bilgi ve bilgisayar güvenliği konusunda en ciddi tehditlerin başında kötücül yazılımlar gelmektedir . Bilinçsiz internet kaynakları kullanımı ve kaynak doğrulamadan indirilen yazılımlar, kötücül kod üretip çeşitli amaçlarla bu kodları kurbanların sistemlerine sızdırmayı hedefleyen saldırganlara açık bir kapı olmaktadır. Özellikle devletler siber ordular kurmaya ve birbirilerine yönelik siber saldırılar organize etmeye başlamıştır. Bu saldırılarda karmaşıklığı gelişmiş, profesyonel gruplar tarafından geliştirilmiş gelişmiş kötücül kodlar kullanılmaktadır. Bu kötücül kodlar metamorfik kod gizleme yöntemleri sayesinde her bir nesilde büyüklüğü, yapısı ve çalışma şekli farklı fakat işlevselliği aynı olan yeni kötücül kodlar üretmektedirler. Bu kodların etkisinden dolayı finansal, kurumsal şirketler geçtiğimiz yıllarda kötücül kodlar yüzünden veri kaybı, veri erişim engellenmesi, veri sızdırılması sebepleriyle zor durumda bırakılmışlardır.

Projedeki amacımız sistemimize gelen zararlı kodların sistemimize zarar vermeden makine öğrenmesi tabanlı yöntemlerle analiz ve tespit edilmesini sağlayan aracın geliştirilmesidir. Aynı anda tespit edilen kötücül kodlarla mevcut GTU zararlı yazılım kütüphanemizi güncelleyerek kütüphanemizin zenginleştirilmesi hedeflenmiştir

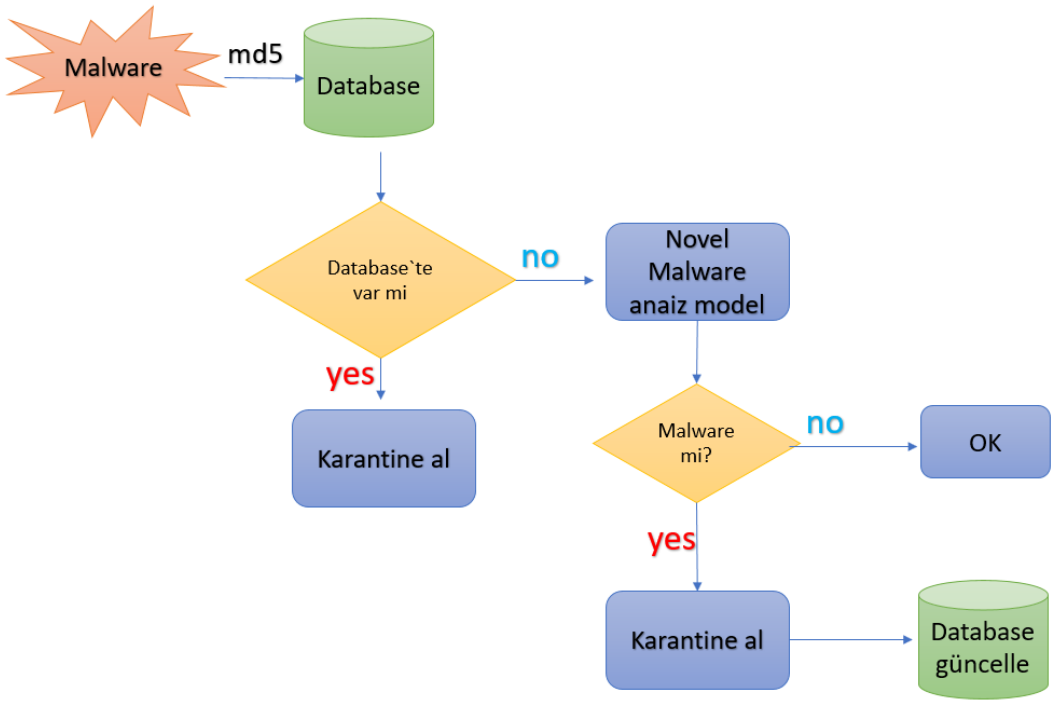
1.1. Proje Tanımı

Günümüzde kötücül kod saldırılarının artmasıyla bireysel internet kullanıcılarında ve kurumsal şirketlerde kayıplar ortaya çıkmıştır. Zararlı yazılımları kötü amaçlar için kullanan kitlenin artması ve zararlı yazılımların siber saldırı gibi özel amaçlar için profesyonel gruplar tarafından kullanılması kötücül kodların çeşitliliğini ve karmaşıklığını arttırdığından dolayı bu kodların tespitinde ideal bir yöntem henüz mevcut değildir. Geliştirilecek projede gerekli kötücül dosyalar GTÜ zararlı yazılım kütüphanesini oluşturan kötücül .exe dosyalarının veri kümesinden toplanacaktır. Elde edilen ya da kullanıcın yüklediği kötü amaçlı içeriği tanımlayan dosyaları imza tabanlı antivirüs aracımız ile analiz edilerek önleme alınacaktır.
İmza tabanlı yöntemimizde kodun imzası çıkartılacak ve mevcut veritabanıbızdan arama yapılacak. Eğer kodun imzası veritabanımızdaki herhangi imza ile eşlesiyorsa kullanıcı uyarılacak ve karantine alıp alınmaması istenecek. Eğer herhangi bir eşleşme olmazsa kötücül yazılımlar dissambler yapısı aracılığı ile çözümlenir ve öznitelik elde etmek

amacıyla kullanılan çeşitli yöntemlerden geçirilir ve makine öğrenmesi tabanlı yöntemlerle analiz edilerek gerekli işlemler yapılır. Tarama sonucu elde edilen bilgiler ile veritabanımız güncellenecektir.

1.2. Projenin Neden ve Amaçları

Proje, kötücül kodlar kullanılarak yapılan kötü amaçlı saldırılar sonucu ortaya çıkabilecek büyük zaafiyetleri ve veri kayıplarını engellemek amacıyla yapılmaktadır. Özellikle bilinçsiz son kullanıcılar ve bilinçsiz çalışanların maruz kalabileceği kötücül kod saldırıları, kurumsal şirketler, bankalar veya devlet kurumlarında geri dönüşü zor olabilecek veri kayıplarına ve hizmet kesintilerine sebep olabilmektedir.



Şekil 1:Proje tasarımı

1.3. Proje ile İlgili Çalışmalar

Literatürde bu konuda bir çok çalışma bulunmaktadır. İncelenen yöntemlerden bazıları;

- MAIL dilinin ürettiği özet yapılar kullanılarak metamorfik yapıdaki kötücül kodların tespitini yapan yöntem.

- Metamorfik kötücül üretim araçlarının benzerlik oranlarını inceleyen Gizli Markov Model (Hidden Markov Model (HMM)) tabanlı metamorfik kötücül kod tespit yöntemi.

- Ağırlıklı bir çizge oluşturarak komşuluk matrisi üzerinden benzerlik bularak kötücül kod tespit yöntemi.

- Kötücül kodları çözümleyerek (disassembler) işlem kod sıklığını histogram üzerinde temsil ederek histogram karşılaştırma temeline dayanan kötücül kod sınıflandırma yöntemi.

- Şifre kriptanalizi (substitution cipher cryptanalysis) yöntemden esinlenilerek metamorfik kötücül kodları tespit eden yöntem.

1.3.1. MAIL Dili Kullanılarak Kötücül Kod Tespiti

Önerilen çözümde, ilk olarak öznitelik sayısını azaltmak için ikili yapıdaki program kodları özet koda dönüştüren kötücül kodların tespiti için ara bir dil olarak tasarlanan MAIL diline dönüştürülmektedir.

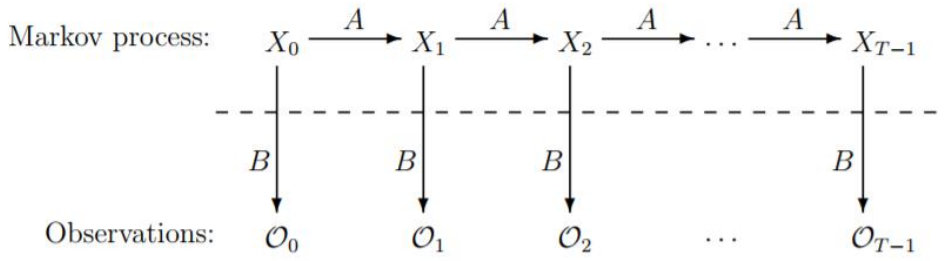
MAIL, ikili program kodlarını çözümleyip (disassembly), programın kontrol akış diyagramının daha iyi tanımlabilmesi ve daha kolay analiz edilebilmesi amacıyla programın işlem kodları (OpCode) üzerinden tanımlanan 26 adet MAIL cümlesinden oluşan özet bir yapıya dönüştüren ara bir dildir.

Öznitelik çıkarma yöntemi için MAIL diline ait cümlelerin TF, IDF, TF-IDF değerleri hesaplanarak öznitelik vektörleri oluşturulmaktadır. Oluşturulan öznitelik vektörleri için öznitelik seçim işlemi ile değerli öznitelikler seçilerek gereksiz, sınıflandırmaya katkısı olmayan öznitelikler filtrelenerek öznitelik sayısı daha da azaltılmaktadır. Seçilen sınıflandırma algoritmalarıyla sistem eğitilmektedir.

MAIL yapsına dönüştürülen kodlar içerisinden %20'lik kısmı seçilerek test verisi olarak ayrılmakta kalan %80'lik veri seti eğitim veri seti olarak kullanılmaktadır. Model için uygun öznitelik çıkarma, öznitelik seçme ve sınıflandırma algoritmalarının belirlenmesi için modelin başarım oranı 10 çapraz doğrulama yöntemi kullanılarak her defasında farklı test ve eğitim veri seti ile ölçülmektedir. Bu yönteme göre göre en başarılı sınıflandırma algoritması KNN,en başarılı öznitelik seçme algoritması ise SVM olarak belirlenmiştir.. [2]

1.3.2 Gizli Markov Model (Hidden Markov Model (HMM))

Kötücül kod tespit tekniklerini 3 ana başlığa ayırabiliriz; imza tabanlı teknik, davranış tabanlı teknik ve sezgisel tablı teknik. Gizli Markov modelleri (HMM) sezgisel tabanlı teknik başlığı altında konumlandırılır. Genellikle istatistiksel model analizi için kullanılır. Ayrıca ses tanıma, kötü amaçlı kod tespiti ve biyolojik dizi analizinde kullanılabilir. Gizli Markov modeli, durumları olan istatistiksel bir modeldir ve durum geçişlerinin bilinen olasılıklarına Markov modeli denir.



Şekil 2:Markov Model

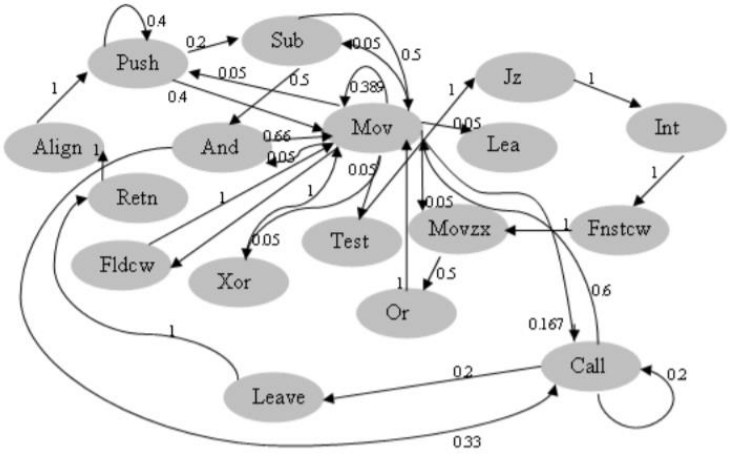
Böyle bir Markov modelinde durumlar gözlemciye görünür durumdadır. Fakat gizli bir Markov modeli (HMM) doğrudan gözlemlenemeyen durumlara sahiptir. HMM bir makine öğrenme tekniğidir. HMM bir durum makinesi(state machine) gibi davranır. Her durum gözlemlerin olasılık dağılımı ile ilişkilidir. Durumlar arasındaki geçiş, sabit ihtimallere sahiptir. Gözlemlenmiş bir veri kümesini kullanarak bir HMM eğitiriz. Böyle bir dizinin tekrar görülme ihtimalini belirlemek için bir gözlem sırasını eğitilmiş bir HMM ile eşleştirebiliriz. Olasılık yüksekse, gözlem sırası eğitim dizilimlerine benzer.
Bu çalışmada HMM tekniği, başlangıçta, istatistiksel özellikleri açısından farklı kötü niyetli kod durumları için eğitilmiştir.Sonuç olarak metamorfik ve polimorfik tabanlı, imzasını değiştirip yenilenebilen kötücül kodların hangi aileden geldiğini belirlemede büyük başarı sağlamıştır. Eğitim kümesi ne kadar büyükse başarı oranı o kadar artacaktır. [3]

1.3.3. Ağırlıklı Çizge ve Komşuluk Matrisi

Bir çalıştırılabilir dosya göz önüne alındığında, işlem kod dizisi ayıklanır ve bir ağırlıklı yönlendirilmiş çizge aşağıdaki gibi oluşturulur. Programda görünen her ayrı işlem kod yönlendirilmiş çizgedeki bir düğümdür. Yönlendirilmiş bir kenara bir düğüm eklenir.

```
1   PUSH    ebp                     24  MOV   ebp, esp
2   MOV     ebp, esp                25  PUSH  edi
3   SUB     esp, 8                  26  PUSH  esi
4   AND     esp, 0FFFFFFF0h         27  PUSH  ebx
5   MOV     eax, ds:dword_404000    28  SUB   esp, 7Ch
6   TEST    eax, eax                29  MOV   edi, [ebp+arg_0]
7   JZ      Short loc_401013        30  MOV   esi, [ebp+arg_4]
8   INT     3                       31  AND   esp, 0FFFFFFF0h
9   FNSTCW  [ebp+var_2]             32  CALL  sub_401930
10  MOVZX   eax, [ebp+var_2]        33  CALL  main
11  AND     eax, 0FFFFF0C0h         34  MOV   [ebp+var_4C], 0
12  MOV     [ebp+var_2], ax         35  MOV   [esp+88h+var_88], ...
13  MOVZX   eax, [ebp+var_2]        36  CALL  CORBA_exception_init
14  OR      eax, 33Fh               37  MOV   dword ptr ...
15  MOV     [ebp+var_2], ax         38  XOR   edx, edx
16  FLDCW   [ebp+var_2]             39  MOV   eax, offset ...
17  MOV     [esp+8+var_8], ...      40  MOV   [esp+88h+var_78], edx
18  CALL    sub_401960              41  MOV   [esp+88h+var_7C], eax
19  LEAVE                           42  MOV   dword ptr ...
20  RETN                            43  MOV   [esp+88h+var_88], ...
21  ALIGN   10h                     44  CALL  poptGetContext
22  PUSH    ebp                     45  MOV   ebx, eax
23  MOV     eax, 10h                46  LEA   esi, [esi+0]
```

Şekil 3:-İşlem Kod Çıkarımı



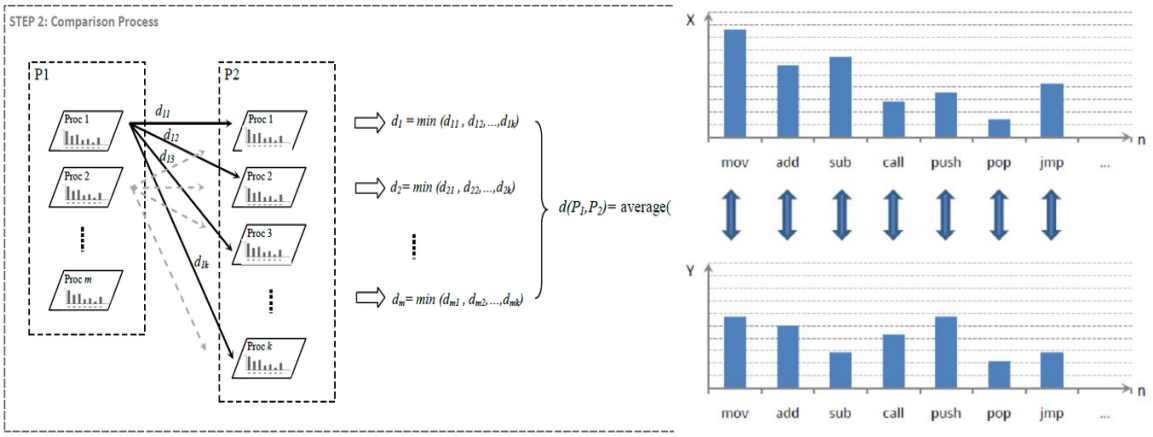
Şekil 4:İşlem Kod Histogram

Bu yöntemle elde edilen işlem kod grafiklerine dayalı bir benzerlik oranı yakalanmaya çalışılmıştır. Bu yöntem metamorfik zorlu soruna uygulanmış ve etkili olduğu kanıtlanmıştır. Benzerlik oranlama işlevine yapılacak küçük değişiklikler şaşırtıcı derecede büyük etkilere sahip olma eğilimindedir.

1.3.4. İşlem Kod Sıklığını Histogram Üzerinde Temsil Etme

Metamorfik virüsler, imza tabanlı tarama yöntemlerini kullanan antivirüs yazılımlarını atlatmak için çeşitli muhafaza yöntemleri kullanmaktadır. Kodlarını yeni örneklerde tamamen veya kısmen farklı olarak değiştirirler ,ancak davranışları ve işlevleri

değişmeden kalır. Bu yöntem ile imza tabanlı antivirüs yazılımları atlatılabilir. Bu araştırmada, verilen bir metamorfik virüsün iki yeni versiyonu tarafından enfekte edilmiş iki dosya arasındaki benzerliği karşılaştırmak için istatistiksel bir teknik kullanılır. Önerilen çözüm statik analiz temelli olup, karmaşık virüslerin çeşitli yavrularında makine komut frekansının histogramını kullanmaktadır. Bir çift yürütülebilir (PE) dosyayı karşılaştırmak için Öklid histogram mesafesi metriği kullanılır. Bu çalışmanın amacı, bazı özel bulanıklaştırma yöntemleri için sunulan bir çözümün bir dosyanın morphed çeşitlerini tespit etmek için kullanılabileceğini göstermektir. Dolayısıyla, bir dosyanın, metamorfik virüsün bir versiyonu olup olmadığını belirlemek için dizgelere dayalı olmayan imza taraması ile kullanılabilir.



Şekil 5:Öklid Histogram Mesafesi Metriği

2. PROJE TASARIMI

Bu bölümde proje gereksinimleri ve use case diyagram yer almaktadır.

2.1. Proje Gereksinimleri

Bunların sağlanması için gerkli ihtiyaçlar:

- Ubuntu işletim sistemli bir bilgisayar.
- 8-16GB RAM
- **Programlama dilleri:**
  Python, JSON, AngularJS, HTML/CSS
- **Programlar:**
  PyCharm, veri tabanındaki verilerin görüntülenmesi için Robo 3T,
- **Database**: MongoDB
- **API**: Flask
- **Kötücül yazılımlar:** NGVCK, G2, MPCGEN
- **İşlem kodları için:** Mail dili
- **Algoritmalar:** Largest Connected Component, Normalized Mutual Information metric

2.2.Sistem Mimarisi

2.2.1. İmza tabanlı tespiti:

- Öncelikle sistemimizi sürekli tarayan bir servisin yazılması
- Veritabanımızın üstünde kurulan sunucudan web apilerin yayınlaması
- Sistemizi tarayan servisin yakaladığı dosaynın imzasının çıkartılması
- Sistemimize bir şekilde gelen dosyandan elde edilen hash değerin sunucu üzerinden yayınladığımız apilerle kontrol edilmesi
- Dosyanın veritabanımızda bulunan kötücül kodla eşleşmesi durumunda engellenmesi
- Veritabanın her dosya sonucunda güncellenmesi
- İnternetin olmadığı durumda da (çevrimdışı) kod incelenmesi. Blacklist oluşturulması [4]

- Kullanıcı dostu arayüz tasarlanarak, bilgisayarımızda yaptığımız tarama sonucunun görselleştirilmesi.

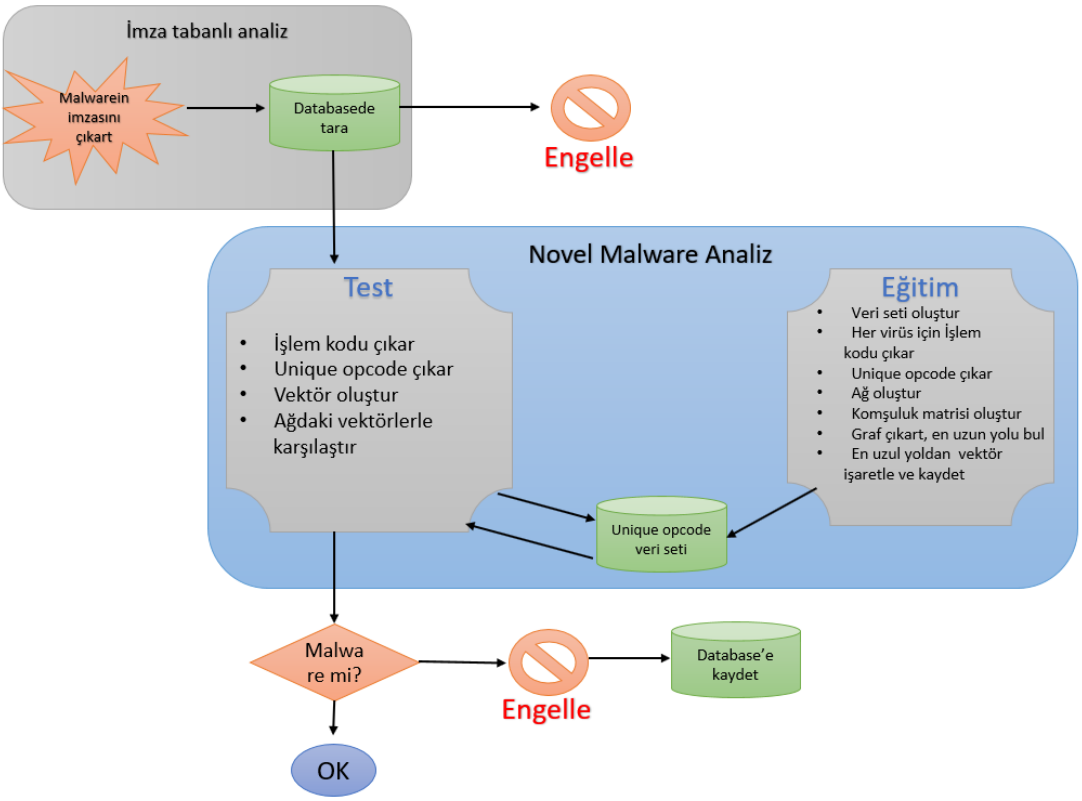
2.2.2. Yeni Metamorfik Zararlı Yazılım Tespiti

2.2.2.1. Eğitim

- Metamorfik kötücül yazılım seti
- Dissambler yardımıyla kötücül yazılımın işlem kodlarının (OpCode) çıkartılması
- Her virus kodundan çıkartılan işlem kodlarından özel (unique) kod vektörünün çıkartılması
- Kullanılacak veri setindeki virus ailesinden ağ oluşturulması (A, B, C ailesi)
- Her bir ağ için unique kodlardan oluşan komşuluk matrisi üretilmesi
- Oluşturulan komşuluk matrisinden graf çıkartma ve Largest Connected Component algoritmasını uygulayarak path oluşturma
- Ağdaki maximim unique code sayısı kadar vektör oluşturarak en son elde edilen grafın pathini işaretlemek [1]

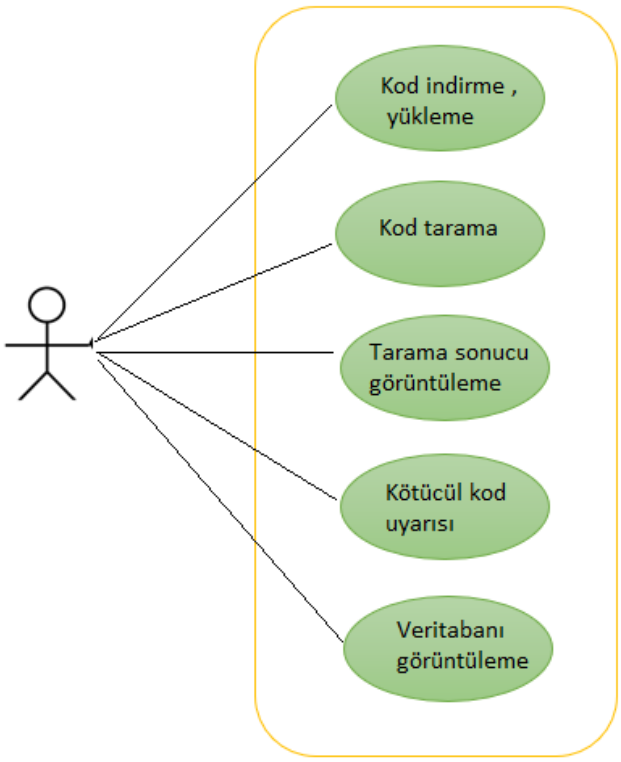
2.2.2.1.Test

- Test edilecek kodun işlem kodu ve unique kodu çıkartılması
- Unique kodundan vektör oluşturulması
- Normalized Mutual Information metric ile vektörün elimizdeki ağların vektörleri ile benzerlikleri karşılaştırılarak ve en yüksek oranda sonuç veren sınıfa ait olduğu etiketlenmesi
- Veritabanın her dosya sonucunda güncellenmesi
- Kullanıcı dostu arayüz tasarlanarak, bilgisayarımızda yaptığımız tarama sonucunun görselleştirilmesi.



Şekil 6: Sistem Mimarisi

2.3. Use Case Diyagramı



Şekil 7:Use Case Diyagramı

4. GERÇEKLEME VE TEST

Bu aşamada daha önce projenin teorik bakımdan öne atılan yargıların nasıl gerçekleştirildiği, hangi yöntemlerde pratiğe aktarıldığı anlatılmaktadır.

4.1.Novel Metamorfik kötücül yazılım tespiti

4.1.2.Eğitim

Algoritma:

1. Veri seti seç
2. Veri setindeki her bir virüs için işlem kodu çıkar
3. İşlem kodundan unique kodları çıkar
4. Ağ oluştur
5. Her bir ağın maximum unique kodlardan oluşan genel vektörü çıkart
6. Bir ağa ait olan virüslerin unique kod vektörlerin komşuluk matrisini çıkart
7. Komşuluk vektörden graf oluştur ve en uzun yolu bul
8. En uzun yolu genel vektöre göre işaretleyip yeni vektör oluştur
9. 6.7.8 adımlarını tüm ağlar için dene

Veri seti 860 tane NGVCK, 80 G2, 80 MPCGEN, 400 MWOR, 80 PSMPC metamorfik kötücül kod sınıflarından oluşturuldu. İşlem kodların çıkartılması için Dissambler kullanıldı.

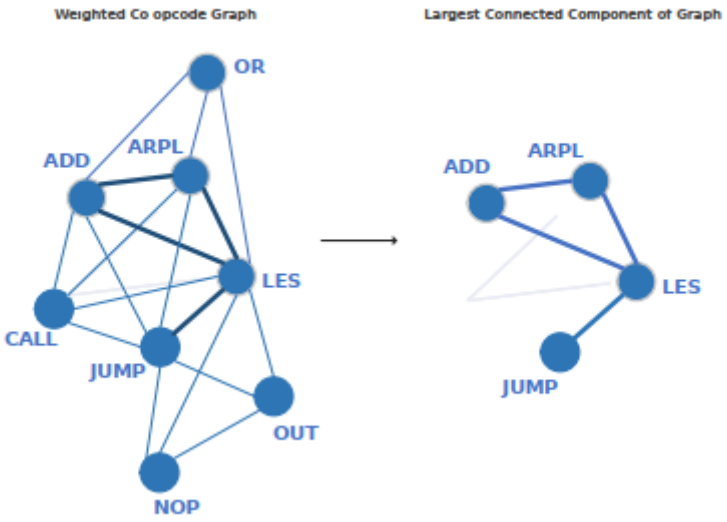
ADD	JUMP	OR	LES	OUT	NOP	ARPL	CALL

Şekil 8: Unique kod örneği

	ADD	JUMP	OR	LES	OUT	NOP	ARPL	CALL
ADD	0	1	1	4	0	0	3	1
JUMP	1	0	0	1	1	1	0	1
OR	1	0	0	2	0	0	1	0
LES	2	1	1	0	1	1	3	1
OUT	0	1	0	1	0	1	0	0
NOP	0	1	0	1	1	0	0	0
ARPL	3	1	1	4	0	0	0	1
CALL	1	1	0	1	0	0	1	0

Şekil 9: Komşuluk matrisi örneği

Grafın en uzun yolunu bulmak için Longest Connected Component kullanıldır.



Şekil 10: Grafın en uzun yol örneği [1]

En uzun yol ADD, ARPL, LES, JUMP olduğuna göre:

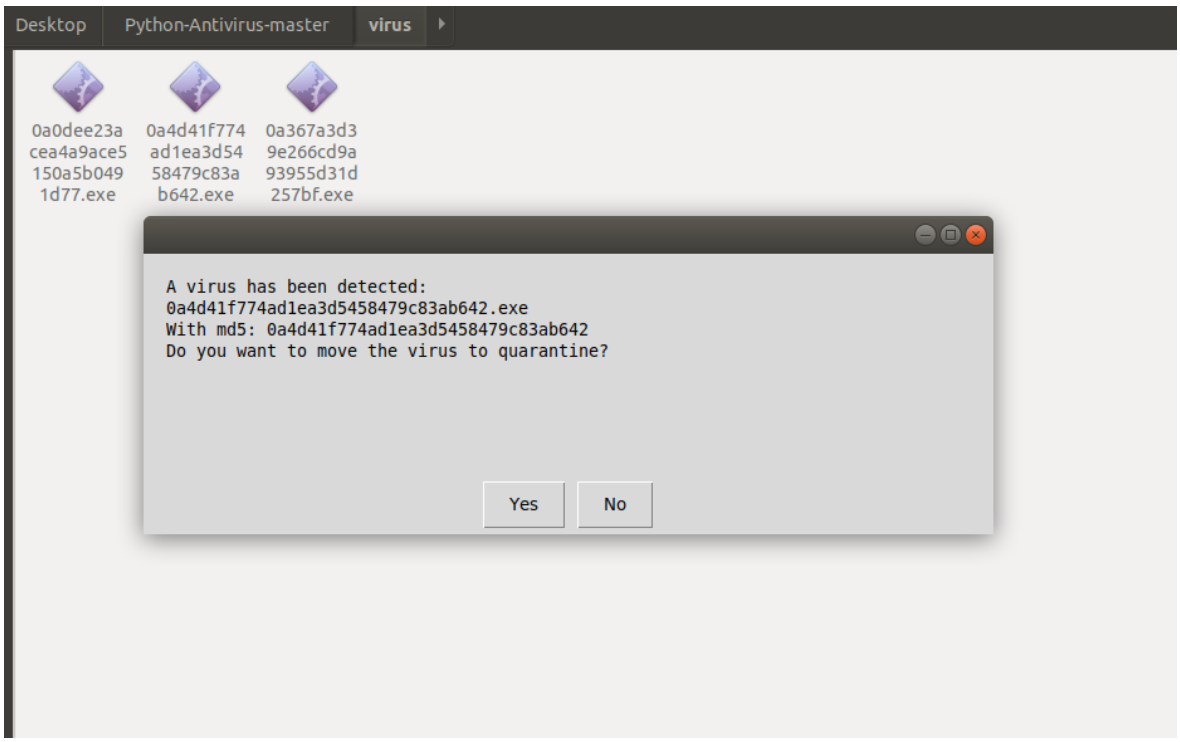
ADD	JUMP	OR	LES	OUT	NOP	ARPL	CALL
1	1	0	1	0	0	1	0

Şekil 11: Unique code vektörüne göre işaretleme örneği

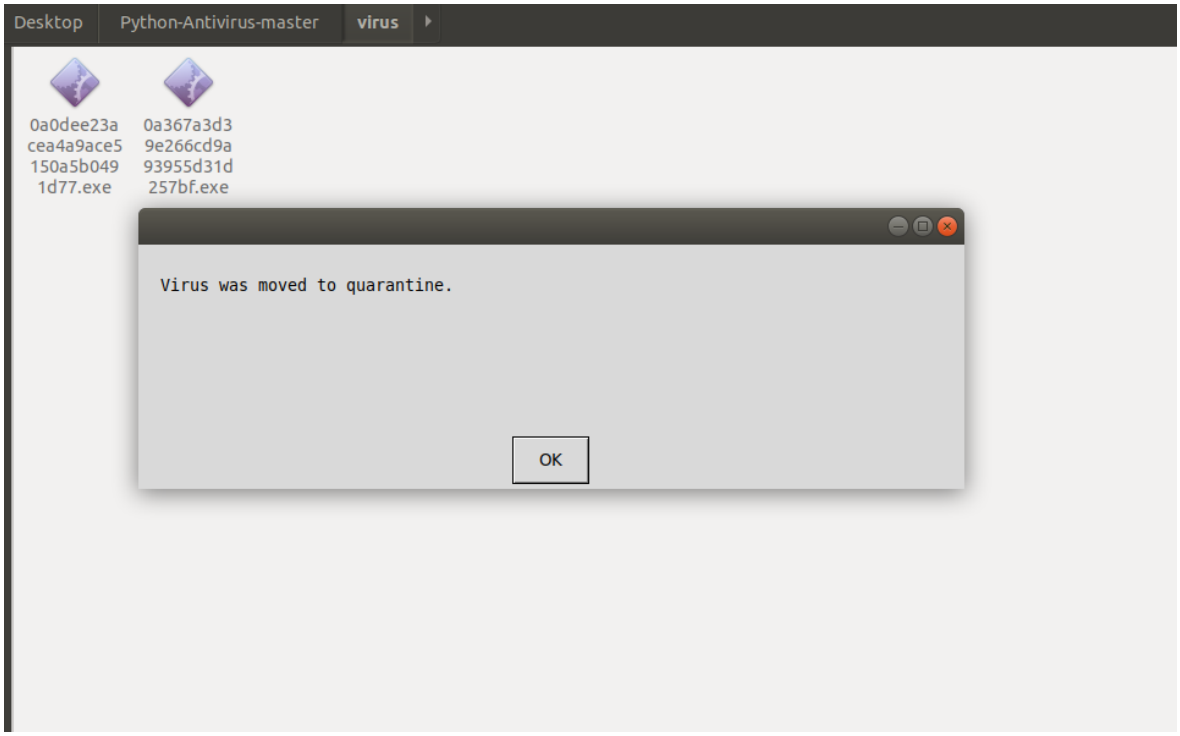
4.1.3.Test

Algoritma:

1. Test edilmesi gereken virüsün işlem kodunu çıkar
2. Eğitimden oluşan genel vektöre göre işaretleyip yeni vektör oluştur
3. Elde edilen vektör ile tüm ağların vektörünü karşılaştır
4. En çok hangi sınıfa ait olduğunu tespit et
5. Veritabanını güncelle
6. Kullanıcıyı uyar



Şekil 12: Virüs yakalandı



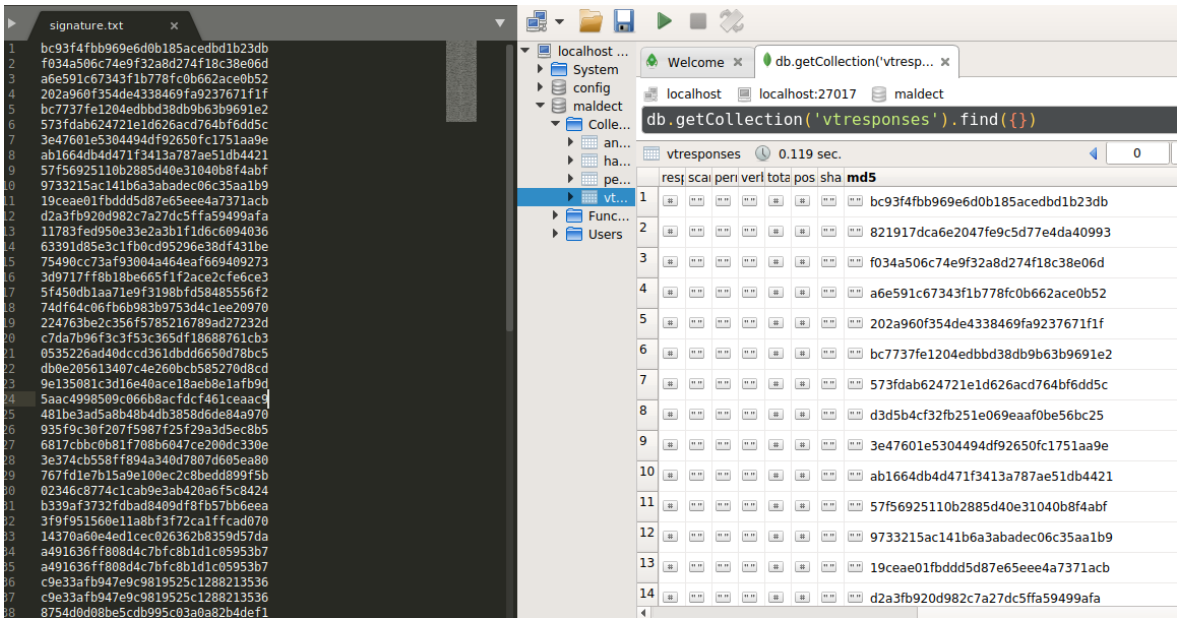
Şekil 13: Virüsü karantine alma

4.2.İmza tabanlı tespit

Algoritma:

1. Başlangıç dosya ya da klasör konumu belirle

2. Klasör konumundaki tüm dosyaları tara
3. Her bir virüs imzasını (MD5) çıkart
4. Çıkarılan MD5'i veritabanından kontrol et
5. Eğer herhangi bir eşleşme var ise kullanıcıya virüs olduğunu belirterek uyar
6. Eğer eşleşme yoksa Novel Metamorfik Kötücül yazılım tespiti modelinin test kısmına gönder



Şekil 14: Signature database ve blacklist

5. SONUÇ

Son dönemlerde kötücül kod tespitinde tersine mühendislik ve makina öğrenmesi yaklaşımları başarılı sonuçlar elde edilmeye başlamıştır. İmza tabanlı tespiti ise en çok yaygın olan tespit sistemidir. Çünkü yeni bir dosyanın kötü amaçlı olduğunu doğrulamak karmaşık ve zaman alıcı olabilir ve genellikle kötü amaçlı yazılımlar o zamana kadar evrim geçirmiştir. Cisco 2017 Yıllık Siber Güvenlik Raporu, analiz ettikleri kötü amaçlı yazılım dosyalarının % 95'inin 24 saat bile eski olmadığını ve hızlı bir "gelişme zamanı" olduğunu gösteriyor [5]. Yeni kötü amaçlı yazılım biçimlerinin tanımlanmasındaki gecikme, şirketleri ciddi zararlara karşı savunmasız kılar. Bundan dolayı kötücül kod tespitinde ilk olarak imza tabanlı kullanıldı. Fakat İmza tabanlı kötü amaçlı yazılım önleme sistemleri, bilinen kötü amaçlı yazılımlara dayanarak oluşturulduğundan, bilinmeyen kötü amaçlı yazılımları veya

bilinen kötü amaçlı yazılım türevlerini bile tespit edemezler. Onun için Novel Metamorfik Kötücül Yazılım Tespiti önerildi ve gerçekleştirildi. Bu çalışmada metamorfik kötü amaçlı yazılımların ortak kod grafiklerinden elde edilen imzasını kullanarak kötü amaçlı yazılımları tespit eden yöntem geliştirilmiştir. Önerilen yöntem diğer yöntemlerle karşılaştırıldığında daha yüksek performans sağlar.

KAYNAKLAR

[1] A. G. Kakışım, İ. Soğukpınar ve M. Nar, «Novel Metamorphic Malware Detection Method Using,» 2019.

[2] S. Alam, N. R. Horspool ve I. Traore, «MAIL: Malware Analysis Intermediate Language - A step Towards Automating and Optimizing Malware Detection,» %1 içinde *6th International Conference on Security of Information and Networks*, Aksaray, Turkey, 2013.

[3] W. Wong ve M. Stamp, «Hunting for Metamorphic engines,» *Journal of Computer Virology and Hacking Technique,* cilt 2, no. 3, pp. 211-229, 2006.

[4] A. Mujumdar, G. Masiwal ve B. B. Meshram, «Analysis of Signature-Based and Behavior-Based Anti-Malware Approaches,» %1 içinde *International Journal of Advanced Research in Computer Engineering and Technology (IJARCET)* , 2013.

[5] J. Cloonan, «Infosecurity-magazine,» 11 4 2017. [Çevrimiçi]. Available: https://www.infosecurity-magazine.com/opinions/malware-detection-signatures/. [Erişildi: 16 5 2019].

[6] N. Çarkacı ve İ. Soğukpınar, «Frequency based metamorphic malware detection.,» %1 içinde *24th Signal Processing and Communication Application Conference*, 2016.