
GAZİ ÜNİVERSİTESİ
Gazi
1926

YEREL AĞLARDAN DNS TÜNELLEME YÖNTEMİYLE VERİ KAÇIRILMASININ TESPİTİNE YÖNELİK BİR YAKLAŞIM

UĞUR TANIK GÜDEKLİ

YÜKSEK LİSANS TEZİ
BİLİŞİM SİSTEMLERİ ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ

ARALIK 2019

Uğur Tanık GÜDEKLİ tarafından hazırlanan "YEREL AĞLARDAN DNS TÜNELLEME YÖNTEMİYLE VERİ KAÇIRILMASININ TESPİTİNE YÖNELİK BİR YAKLAŞIM" adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Bilişim Sistemleri Ana Bilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Doç.Dr. Bünyamin CİYLAN

Bilgisayar Mühendisliği Anabilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum ……..……..

Başkan: Prof. Dr. M. Ali AKCAYOL

Bilgisayar Mühendisliği Anabilim Dalı, Gazi Üniversitesi ...……

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum

Üye: Doç. Dr. Harun ARTUNER ...……

Bilgisayar Mühendisliği Anabilim Dalı, Hacettepe Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum

Tez Savunma Tarihi: 20 /12 /2019

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

……..…….

Doç. Dr. Aslıhan TÜFEKCİ

Bilişim Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Bilişim Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Uğur Tanık GÜDEKLİ

20/12/2019

YEREL AĞLARDAN DNS TÜNELLEME YÖNTEMİYLE VERİ KAÇIRILMASININ TESPİTİNE YÖNELİK BİR YAKLAŞIM

(Yüksek Lisans Tezi)

Uğur Tanık GÜDEKLİ

GAZİ ÜNİVERSİTESİ

BİLİŞİM ENSTİTÜSÜ

Aralık 2019

ÖZET

Alan adı sistemi kısaca DNS, bilgisayarları, hizmetleri veya diğer ağ kaynaklarını adlandırmak için kullanılan bir protokoldür. Protokolün en önemli amacı, insanlar için anlamlı olan isimleri, bilgisayarlar için anlamlı olan IP adreslerine çevirerek bilgisayarların ve hizmetlerin kolayca bulunabilmesini sağlamaktır. DNS protokolü tasarlanırken, internete ve protokollere yönelik tehditler olgunlaşmadığı için, protokol güvenliği bir etken olarak ele alınmamış ve bunun sonucu olarak protokol üzerinde birçok veri korsanlığı yöntemi geliştirilmiştir. DNS protokolü kullanılarak geliştirilen önemli bir tehdit türü de DNS tünellemedir. DNS tünelleme bilgisayar ağlarından yetkisiz olarak veri çıkarılmasına imkân tanımaktadır ve bunu normal DNS sorguları ile yaptığı için tespit edilmesi güçtür. DNS tünellemeyi engellemeye yönelik daha önceki çalışmalarda DNS sorgu ve cevaplarına ait istatistiksel özellikler kullanılarak paket ve trafik analizi yapılmıştır. Güncel çalışmalar ise sınıflandırma algoritmaları kullanılarak DNS trafiğindeki anomalililerin tespitine yoğunlaşmaktadır. Bu çalışmada DNS paketi içerisinde bulunan ve anomali tespitinde etkisi ispatlanmış özelliklere, etki alanı itibar analizi eklenmiş ve çok etiketli destek vektör makineleri sınıflandırma algoritması kullanılarak DNS tünelleme yapılan paketler tespit edilmeye çalışılmıştır. Deneysel sonuçlar sunulan çalışmanın 0.85 duyarlılık – recall ölçümü ile DNS tünelleme tespitindeki efektifliğini göstermiştir.

Bilim Kodu : 92407

Anahtar Kelimeler : DNS saldırıları, DNS tünelleme, DDOS, network güvenliği

Sayfa Adedi : 69

Danışman : Doç. Dr. Bünyamin CİYLAN

AN APPROACH TO DETERMINING DATA HIJACKING WITH DNS TUNNELING METHOD FROM LOCAL NETWORKS

(M. Sc. Thesis)

Uğur Tanık GÜDEKLİ

GAZİ UNIVERSITY

INFORMATICS INSTITUTE

December 2019

ABSTRACT

The domain name system is a protocol used for briefly a naming system for DNS, computers, services or other network resources. The most important purpose of the protocol is to translate names that are meaningful to people into IP addresses that are meaningful to computers, making it easy to find every computer or service in the world. Since attacks to internet and protocols had not been developed when DNS protocol has been designed, security is not considered as a matter and lots of data piracy method have been developed as a result of this. DNS tunneling is an important threat developed using DNS protocol. DNS tunneling allows data hijacking on computer networks and it is hard to detect due to using standart DNS queries. Previous studies analyzed packet and traffic using statistics features belong to DNS queries and responses. Current studies concentrate detect anomalies in DNS traffic using classification algorithm. In this thesis, domain reputation analysis has been added to the DNS packet features that have been proven for effectiveness in the anomaly detection and tried to detect packets that have been including DNS tunneling using Support Vector Machine classification algorithm. Experimental results demonstrate the efficacy of the proposed SVM classification method by obtaining an recall of 0.85.

Science Code : 92407

Key Words : DNS attack, DNS tunneling, DDOS, network security

Page Number : 69

Supervisor : Assoc. Prof. Dr. Bünyamin CİYLAN

TEŞEKKÜR

Yüksek lisans eğitimim boyunca bana danışmanlık yaparak gösterdiği büyük emek ve destekten dolayı tez danışmanım saygıdeğer Doç. Dr. Bünyamin CİYLAN'a, yoğun çalışmalarım sırasında ümit verdiği, sabır gösterdiği için ve motivasyon desteği için sevgili eşime, varlıklarıyla bana güç sağlayan aileme ve çalışmam sırasında yardımını esirgemeyen bütün çalışma arkadaşlarıma teşekkür ederim.

İÇİNDEKİLER

Sayfa

ÖZET iv

ABSTRACT v

TEŞEKKÜR vi

İÇİNDEKİLER vii

ÇİZELGELERİN LİSTESİ ix

ŞEKİLLERİN LİSTESİ ix

RESİMLERİN LİSTESİ xi

SİMGELER VE KISALTMALAR xii

1. GİRİŞ 1

2. DNS PROTOKOLÜ 5

2.1. DNS Çalışma Yapısı 5

2.2. DNS Güvenliği Uzantıları 10

3. DNS'E YÖNELİK SALDIRILARA GENEL BAKIŞ 13

3.1. DNS Sunucularını Hedefleyen Saldırılar 13

3.1.1. Temel DoS saldırısı 13

3.1.2. DNS önbellek zehirlenmesi 15

3.1.3. Arabellek aşımı 17

3.2. DNS Sunucularını Kullanan Saldırılar 18

3.2.1. Yansıtma saldırıları 18

3.2.2. DNS tünelleme 20

4. DNS'E YÖNELİK SALDIRILARA KARŞI TESPİT YÖNTEMLERİ. 21

4.1. Saldırı Tespit Yaklaşımları 22

4.2. IDS'de Veri Madenciliği Teknikleri 23

4.2.1. İstatistiksel teknikler 24

Sayfa

4.2.2. Makine öğrenmesi 25
4.2.3. Hibrit sınıflandırıcılar ve optimizasyon teknikleri 30
5. DNS TÜNELLEME 31
5.1. Tünelleme Temelleri 32
5.2. Tünelleme Bileşenleri 32
5.3. Kodlama ve Teknikler 34
5.4. DNS Tünelleme Yardımcı Programları 36
5.4.1. Dns2tcp 36
5.4.2. DNSCat 37
5.4.3. Iodine 38
5.4.4. OzymanDNS 38
5.5. DNS Tünelleme Tespit Etme Yöntemleri 38
5.5.1. İstek ve yanıtın boyutu 39
5.5.2. Ana bilgisayar adlarının entropisi 39
5.5.3. İstatistiksel analiz 39
5.5.4. IP adresi başına DNS trafiği hacmi 40
5.5.5. Etki alanı başına DNS trafiği hacmi 40
5.5.6. Etki alanı başına ana bilgisayar adı sayısı 40
5.6. DNS Tünelleme Tespitinde Makine Öğrenmesi Yöntemleri 41
6. ÖNERİLEN ÇÖZÜM YÖNTEMİ 43
6.1. Verinin Hazırlanması 46
6.2. Geliştirilen Uygulama 51
7. SONUÇ VE ÖNERİLER 63
KAYNAKLAR 65
ÖZGEÇMİŞ 69

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge 6.1. Hata matrisi..... 57

Çizelge 6.2. Parametre ölçüm sonuçları 60

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil 2.1. Özyinelemeli DNS sorgulama senaryosu 6

Şekil 2.2. Yinelemeli DNS sorgulama senaryosu 7

Şekil 2.3. DNSSEC çalışma yapısı 12

Şekil 3.1. Temel bir DoS saldırısı 14

Şekil 3.2. Önbellek zehirlenmesi şeması 16

Şekil 3.3. Örnek bir yönlendirme saldırısı 19

Şekil 4.1. IDS çalışma yapısı 22

Şekil 5.1. DNS tünelleme çalışma yapısı 34

Şekil 5.2. DNS2TCP çalışma mantığı 37

Şekil 6.1. Lineer ayrıştırılabilen ve lineer ayrıştırılamayan iki veri seti 46

Şekil 6.2. Uygulama akış şeması 52

RESİMLERİN LİSTESİ

Resim **Sayfa**

Resim 2.1. Örnek bir DNS sorgusu 8

Resim 6.1. DNS2TCP sunucu kurulumu 47

Resim 6.2. DNS2TCP istemci kurulumu 48

Resim 6.3. Iodine aracı ile tünelli trafik oluşturma 48

Resim 6.4. Temiz DNS trafiği 50

Resim 6.5. Tünelli DNS trafiği 50

Resim 6.6. Uygulamanın öğrenme amacıyla çalıştırılması 52

Resim 6.7. Uygulamanın test amacıyla çalıştırılması 52

Resim 6.8. Entropi hesaplama kod parçacığı 53

Resim 6.9. Alt etki alanı sayısı hesaplama kod parçacığı 54

Resim 6.10. Etki alanı adı uzunluğu hesaplama kod parçacığı 54

Resim 6.11. Alan adı oluşturulma tarihi hesaplama kod parçacığı 55

Resim 6.12. Alan adı IP adresinin zararlı sitelerde kontrol edilmesi kod parçacığı 55

Resim 6.13. Entropi, alt alan adı sayısı ve sorgu uzunluğu sonuçları 58

Resim 6.14. Entropi, alt alan adı sayısı, sorgu uzunluğu ve zararlı IP kriteri sonuçları 58

Resim 6.15. Entropi, alt alan adı sayısı, sorgu uzunluğu ve paket boyutu sonuçları 59

Resim 6.16. Entropi, alt alan adı sayısı, sorgu uzunluğu ve alan adının oluşturulma tarihi sonuçları 59

Resim 6.17. Tüm parametrelerin kullanılması ile elde edilen sonuçlar 60

Resim 6.18. Alan adı uzunluğu, alt alan adı sayısı ve entropi değerleri vektör dağılımı 61

Resim 6.19. Paket boyutu, alt alan adı sayısı ve entropi değerleri vektör dağılımı 61

Resim 6.20. Sorgu uzunluğu, entropi ve oluşturulma tarihi değerleri vektör dağılımı .. 62

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	**Açıklamalar**
kb	Kilobayt
ms	Milisaniye
sn	Saniye

Kısaltmalar	**Açıklamalar**
ACO	Karınca kolonisi optimizasyonu (Ant colony optimization)
ARPANET	The advanced research projects agency network
CNAME	Canonical name
DDoS	Dağıtık hizmet aksatma (Distributed denial of service)
DNS	Alan adı sistemi (Domain name system)
DNSSEC	Domain name system security extension
DoS	Hizmet aksatma (Denial of service)
DT	Karar ağacı (Decision tree)
EDNS	DNS eklentisi **(**Extension mechanisms for DNS)
GA	Genetik algoritma
GP	Genetik programlama
HIDS	Host ıntrusion detection system
HMM	Hidden markov model
ICANN	Internet corporation for assigned names and numbers
IDS	Saldırı tespit sistemi (Intrusion detection system)
IP	İnternet protokolü (Internet protocol)
IPV4	İnternet protokolü versiyon 4 (Internet protocol version 4)
ISS	İnternet servis sağlayıcı
k-NN	k-En yakın komşu (k-nearest neighbour)
KSK	Anahtar imzalama anahtarı (Key signing Key)
MITM	Man in the middle

Kısaltmalar	Açıklamalar
MLP	Multiple layer probe
MLT	Makine öğrenmesi teknikleri (Machine learning technices)
MX	Mail exchanger DNS record
NB	Naive bayes
NIDS	Network ıntrusion detection system
NS	Ad sunucusu DNS kaydı (Name server DNS record)
PCAP	Packet capture
PRNG	Pseudo-random number generator
PSO	Particle swarm optimization
PTR	İşaretçi kaydı (Pointer record)
QID	Sorgu kimliği (Query ıdentification)
QNAME	Sorgu adı (Query Name)
RFC	Yorumlar için rica (Request for comments)
RRset	DNS kayıt takımı (DNS record set)
RSA	Rivest–shamir–adleman
SOM	Self organized map
SVM	Destek vektör makinesi (Support vector machine)
TCP	İletim kontrol protokolü (Transmission control protocol)
TTL	Yaşama süresi (Time to live)
TXT	Text kaydı (Text record)
UDP	Kullanıcı veri paket protokolü (User datagram protocol)
YSA	Yapay sinir ağları
ZSK	Zone imzalama anahtarı (Zone signing key)

1. GİRİŞ

İnternet, günümüzde artan sayıda kullanıcı sayısı ve yüksek hızlı bağlantıların kullanılabilirliği ile geniş kullanım oranlarına ulaşmıştır. Bu genişleme, uygulama, servis veya bir hizmet sunan ana bilgisayarlara yapılan saldırı sayısında da önemli bir artışa sebep olmuştur. Alan Adı Sistemi (DNS), internetin önemli kısımlarından biridir. DNS olmadan web sitelerine kolaylıkla bağlanılamaz ve hizmet sağlayan sunucuların IP adreslerinin bilinmesi gerekliliği durumunda kalınırdı. Bu anlamda DNS sunucularının da saldırganlarının hedefleri kapsamında olduğu söylenebilmektedir. Bu saldırılar, DNS protokolündeki istismarlar veya DNS yazılımındaki hatalar veya zafiyetler nedeniyle başarıyla sonuçlanabilmektedir [1]. Ayrıca DNS sunucusunu kullanıcılar için kullanılamaz duruma getirme amacıyla büyük miktarda bant genişliği ve işlem gücü kullanarak sunuculara aşırı yük bindiren bir saldırı grubu da bulunmaktadır. Bu saldırılara Hizmet Reddi (Denial of Service - DoS) saldırıları adı verilmektedir. Bu saldırıların nedenleri farklıdır: Örneğin bir DNS önbellek zehirlenmesi saldırısı, bir etki alanı üzerinde kontrol sağlamak için kullanılırken, DoS saldırıları sadece kullanıcılara sunulan normal hizmeti kesmek amacıyla gerçekleştirilmektedir [2]. Her ne kadar sistem yöneticileri altyapılarını korumak için sürekli yeni savunma hatları ekleseler de, yeni nesil saldırganlar sürekli olarak bu savunmaları atlatabilmek için yeni yaklaşımlar denemektedirler. Son zamanlarda birkaç "Atanan İsimler ve Numaralar için İnternet Şirketi (Internet Corporation for Assigned Names and Numbers)" adreslerinin etki alanı ele geçirildiğine dair raporlar bulunmaktadır. Bu, ICANN gibi saldırılar için önlemler alan bir kuruluş etki alanının dahi ele geçirilme olayına maruz kalabildiğini göstermektedir [3].

İnternete ve protokollere yönelik geliştirilen tehditlerde sıkça kullanılan bir teknik de tünellemedir. Tünelleme basit anlamda bir protokole ait veriyi başka bir protokol içerisinde taşıma işlemidir. DNS tünellemeyi de DNS protokolü içerisinde farklı protokollere ait verileri taşımak olarak tanımlayabiliriz. DNS tünelleme tekniği genellikle Güvenlik Duvarı ve saldırı engelleme sistemleri ile güvenliğe alınmış, internet erişimi de proxy ile belirli kullanıcı veya bilgisayarlara sınırlanmış ağlarda güvenlik cihazlarına ve proxy cihazlarına takılmadan ve iz bırakmadan internet erişimi sağlamak için kullanılmaktadır [4]. Günümüzde çoğu işletmelerde internet erişimi yetkiye bağlanmış veya sınırlandırılmıştır. Şirketler, kurum ve kuruluşlar kullanıcıların ve işletmelerinin güvenliği için kullanıcıların internet erişimlerini sınırlandırmakta ve yönetmektedirler [5]. Kumar ve bahis sitelerine,

zararlı alışkanlık içeren sitelere ve şiddet içerikli sitelere erişimin engellenmesi bu sınırlandırmalara örnek gösterilebilir. Kimi işletmeler bu erişim kısıtlamalarını genişleterek dosya indirme/yükleme, e-posta ve bulut saklama sitelerine erişimi de engellemektedir. DNS tünelleme ile şirketin/işletmenin uygulamış olduğu bu engeller atlatılabilmekte ve internete sınırsız erişim sağlanabilmektedir. Bu durum da şirket/işletme için ciddi bir güvenlik tehdidi oluşturmaktadır [5].

İşletme, kurum ve kuruluşlar, çalışanlarının ve işletmenin güvenliği için internet erişiminin kısıtlanmasına ek veya bundan farklı birçok yönetimsel faaliyet yürütmektedirler. Kurum/kuruluşa ait verilerin sınıflandırılarak belirli tasnife ait verilerin şirket dışına çıkarılmasının engellenmesi bu güvenlik faaliyetlerinden biridir. Korumak istenen verinin şirket ağı dışına çıkmaması için farklı veri çıkış noktalarında veri kaybını engelleme (DLP) çözümleri kullanılmaktadır. DNS tünelleme ile kurum/kuruluş içerisinde devreye alınmış olan bu veri kaybını engelleme çözümleri de çok kolay bir şekilde atlatılabilmektedir. Böylece şirket için çok önemli olan veya çok iyi bir şekilde korunmaya çalışılan bir veri yalnızca DNS trafiği oluşturularak şirket dışına çıkarılabilmektedir [6].

Tamamen kapalı ağlardan DNS tünelleme yöntemiyle dahi veri kaçırılamayacaktır. Ancak günümüzde internet erişimi veya diğer protokoller üzerinden şirket ağının dışına çıkılması ne kadar engellenirse engellensin, DNS, kurum/şirket uygulamalarının sağlığı için hiçbir zaman engellenmeyen veya en son engellenen protokoldür. DNS tünellemeyi başarılı şekilde gerçekleştirebilmek için tüm internet trafiği kapalı olsa da DNS trafiğine izin verilmiş olması yeterlidir [6, 7].

DNS tünelleme işlemi ağ uç birimlerince izlendiğinde normal bir DNS sorgusu olarak görülmektedir. Bu nedenle mevcut teknolojiler ile DNS tünelleme yöntemiyle veri kaçırılmasının tespit edilmesi oldukça zordur. DNS tünellemeyi tespit etmek için birçok akademik çalışma yürütülmüştür ve yeni yöntemler sunulmaya devam edilmektedir [8, 9].

DNS tünellemeyi engellemek için bugüne dek yapılan çalışmalar paket analizi ve trafik analizi olmak üzere iki kategoride toplanmaktadır [10]. Paket analizi üzerine yürütülen çalışmalar DNS paketi içerisindeki özelliklere yoğunlaşırken, trafik analizi çalışmaları bir istemciden çıkan trafiğin boyutu veya bir alan adına yönelik yapılan sorguların yoğunluğu

gibi özelliklere odaklanmaktadır [10]. Çalışmaların gittiği bir diğer yönün de istatistiksel analizler olduğu söylenmektedir [11].

Güncel yaklaşımlar ise DNS tünellemeyi tespit etmek için DNS trafiğini sınıflandırma algoritmaları yardımıyla sınıflandırarak tespit etmeye çalışmaktadır [12].

Sunulan çalışmada DNS tünelleme trafiği oluşturmak için çeşitli araçlar kullanılmış, geleneksel yöntemlerden öğrenilen DNS paket özellikleri dikkate alınarak bunlara yeni parametreler eklenmiş ve genişletilmiş bu özellikler dikkate alınarak sınıflandırma algoritması yardımıyla elde edilen tünelli ve temiz trafiklere ait ağ trafiği paketleri sınıflandırmaya tabi tutulmuştur. Çalışmaya girdi olması amacıyla daha önce tek başına ele alınmamış bir özellik olarak DNS paket boyutunun DNS tünelleme yapıldığında DNS trafiğine etkisi özel olarak incelenmiştir [13].

Sunulan çalışma ile yerel ağlardan DNS tünelleme yöntemiyle veri kaçırılmasına yönelik geliştirilen sınıflandırma tabanlı çözümlere; DNS paket boyutu, sorgulanan alan adının ne zaman oluşturulduğu (whois bilgisi), sorgulanan alan adının daha önce bir zararlı ip veya alan adı listesinde yayınlanıp yayınlanmadığı ve sorgulana kayıt türü kontrolleri eklenerek katkıda bulunulmaya çalışılmış, sunulan yöntem ile başarım oranı artırılmaya çalışılmıştır. Bu sayede internet erişiminin kısıtlandığı ancak DNS trafiğinin engellenmediği ağlarda DNS tünelleme yapan istemcilerin tespitine imkân tanımak amaçlanmıştır.

Bu amacı gerçekleştirmek için daha önceki çalışmalarda elde edilen en iyi sonuçlar incelenmiştir. Sunulan yeni özelliklere bakılmadığında yapılan sınıflandırmalarda en yüksek oran olarak 0.80 f1 ölçümüne ulaşıldığı öğrenilmiştir [14].

Bu tez kapsamında sunulan çalışmada eklenen yeni özelliklerin etkinliğini test etmek için daha önce DNS tünelleme tespitinde yalnızca paket boyutunun etkisi ayrıca incelenmiştir. Sorgulanan etki alanının whois bilgilerine erişebilmek ve daha önce bir zararlı listesinde bulunup bulunmadığının kontrol edilebilmesi için bu iki bilginin sorgulanabileceği bir API'nin bulunması gerekmektedir. Sunulan çalışmada bu alt amaç test edilerek alan adının ne zaman oluşturulduğu bilgisinin iana.org adresinden API kullanılarak sorgulanabildiği görülmüştür [1]. Bir diğer alt amaca yönelik gerçekleştirilen test ise sorgulanan adının daha önce bir zararlı IP veya etki alanı listesinde yayınlanıp yayınlanmadığının

kontrolünün test edilmesidir. Bu alt amacı test etmek için daha önce geliştirilmiş, bir web sitesi içerisindeki belli bir metin ifadesini arayan açık kaynak kodlu “isthisipbad” kod parçacığı kullanılmıştır. Bu uygulama ile sorgulanan alan adının birçok zararlı IP listesinde aranabildiği görülmüştür.

Alan adı repütasyon sorgusu yapabilmek için kullanılan bu API'ler çalışmayı sınırlayan faktörler olarak değerlendirilebilmektedir. İlgili API'ler daha fazla aktif olmadığında veya hizmet vermediğinde uygulanabilecek yeni çözümlere sonuçlar bölümünde yer verilmiştir.

2. DNS PROTOKOLÜ

Alan Adı Sistemi (DNS), günümüz internetinin işletilmesinde büyük önem taşımaktadır. Bir DNS sunucusu, kullanım kolaylığı sağlayan alan adlarının ilgili IP adresi ile eşleştirilmesi için kullanılmaktadır. Anlaşılabilir alan adlarını IP adreslerine eşleme kavramı, ARPANET'in (The Advanced Research Projects Agency Network) [1] kullanılmaya başladığı zamanına dayanmaktadır. O günlerde, mevcut tüm eşlemeleri içeren küresel bir tablo kullanılmaktaydı. Bir ana bilgisayar ağa bağlanmak zorunda kaldığında, en son sürüme sahip olduğundan emin olmak için bu küresel tabloyu indirmesi gerekmekteydi. İnternetin yaygın olarak kullanılmasıyla, bu yöntem bir çözüm olarak yetersiz kalmıştır. Buna bağlı olarak, 1983 yılında Mockapetris isimli bilgisayar bilimcisi, merkezi olmayan bir DNS önerisini içeren (RFC) 882 ve 883'ü sunmuştur [15]. Bunların yerini daha sonra (1987) RFC 1034 ve 1035 almıştır [15]. En eski DNS sunucuları ise 1984 ve 1985'te yazılmıştır [16]. En son sunucuya ise BIND (Berkeley İnternet Adı Alan Adı) adı verilmiştir ve bugüne kadar adını çeşitli sürümlerle korumuştur [16].

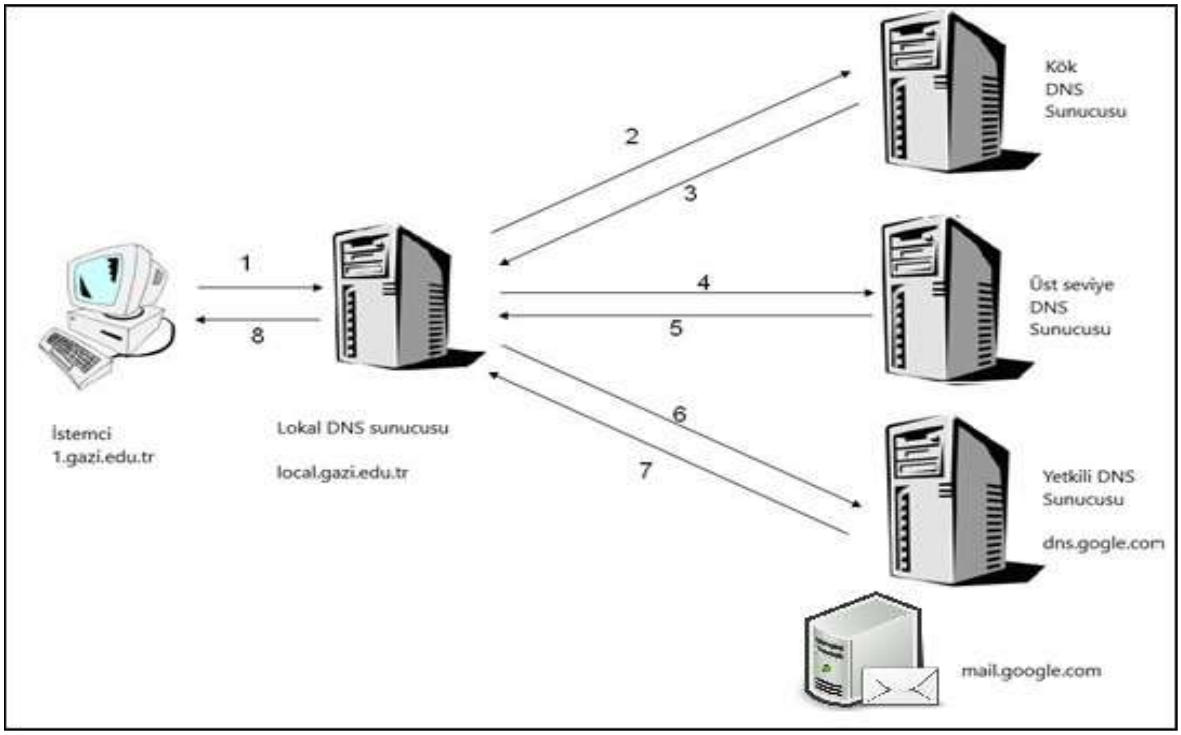
Günümüzde DNS, posta aktarma uygulamaları veya kara liste uygulamaları gibi çok çeşitli servisler tarafından kullanılmaktadır. Sonuç olarak, DNS'in değeri IP adreslerinin alan adlarıyla eşleştirilmesinin ötesine geçmektedir ve bugün sıklıkla internet, büyük oranda DNS'e bağlı durumdadır. Etki alanı adları genellikle iyi niyetli olarak kullanılsa da kötü amaçlara da olanak sağlamaktadır. Günümüze kadar alan adları ve DNS, siber suçlular tarafından botnetler oluşturmak, hizmet kesintisi yapmak, kötü niyetli içerik barındırmak veya kimlik avı yapan web sitelerinin yönetimini çalmak için kullanılmıştır. DNS'in dağıtılmış ve küresel bir sistem olması, küresel ölçekte saldırılara devam etmek isteyen siber suçlular için etkin bir olanaktır [8].

2.1. DNS Çalışma Yapısı

Temel bir DNS çözümleme senaryosu aşağıda Şekil 2.1'de görülmektedir. Adres çözümlemesini tamamlamak için bir dizi adım gerçekleşmektedir.

- İstemci kendi yerel DNS sunucusundan bir adresi çözümlemesi talebinde bulunur.
- Yerel DNS sunucusu, kök DNS sunucusundan adres çözümlemesi talebinde bulunur.
- Kök DNS sunucusu, üst düzey DNS sunucusuna yapılan bir başvuru ile yanıt verir.

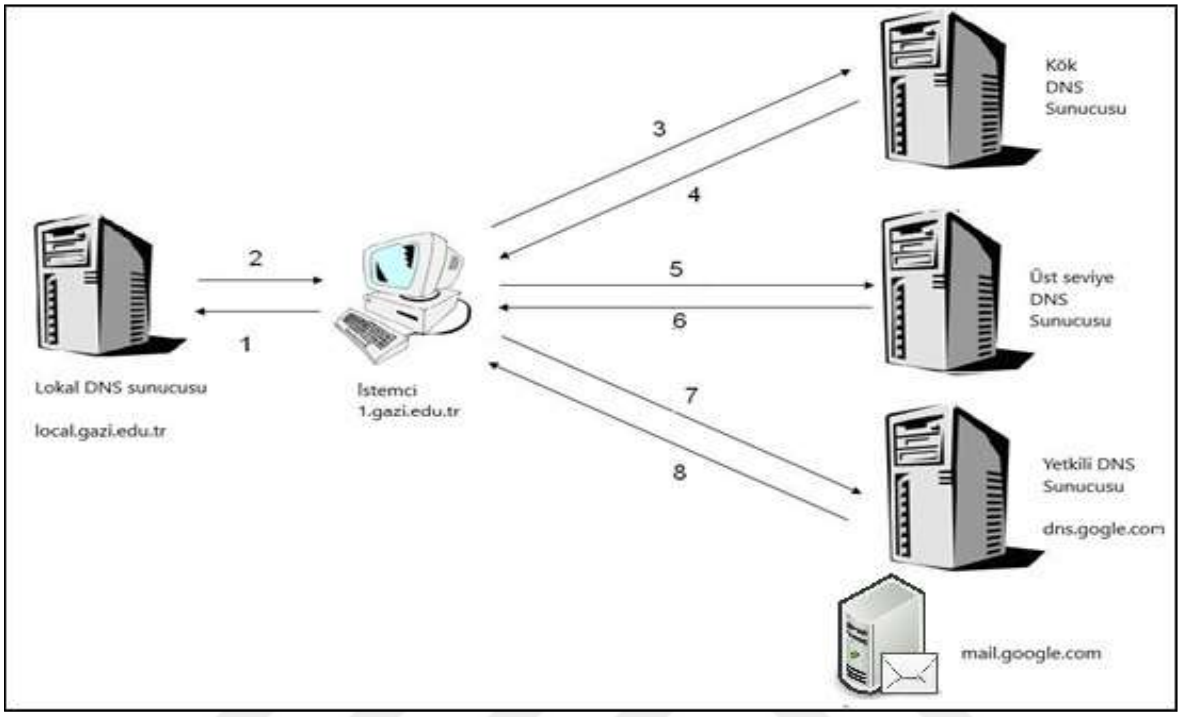
- Yerel DNS sunucusu, adres çözümlemesi için en üst düzey DNS sunucusuna sorgulama yapar.
- Üst seviye DNS sunucusu, ikinci seviye DNS sunucusuna yapılan bir başvuru ile yanıt verir.
- Yerel DNS sunucusu, adres seviyesi için ikinci seviye DNS sunucusuna sorgulama yapar.
- Bu ikinci seviye DNS sunucusu, sorgulanan adres için yetkili DNS sunucusuysa, ana bilgisayarın IP adresiyle veya bir hatayla yanıt verir. Aksi halde, adres ile üçüncü seviye DNS sunucusuna cevap verir.
- Yerel DNS sunucusu, istemciye sorusunun cevabını kullanarak yanıt verir [6].



Şekil 2.1. Özyinelemeli DNS sorgulama senaryosu

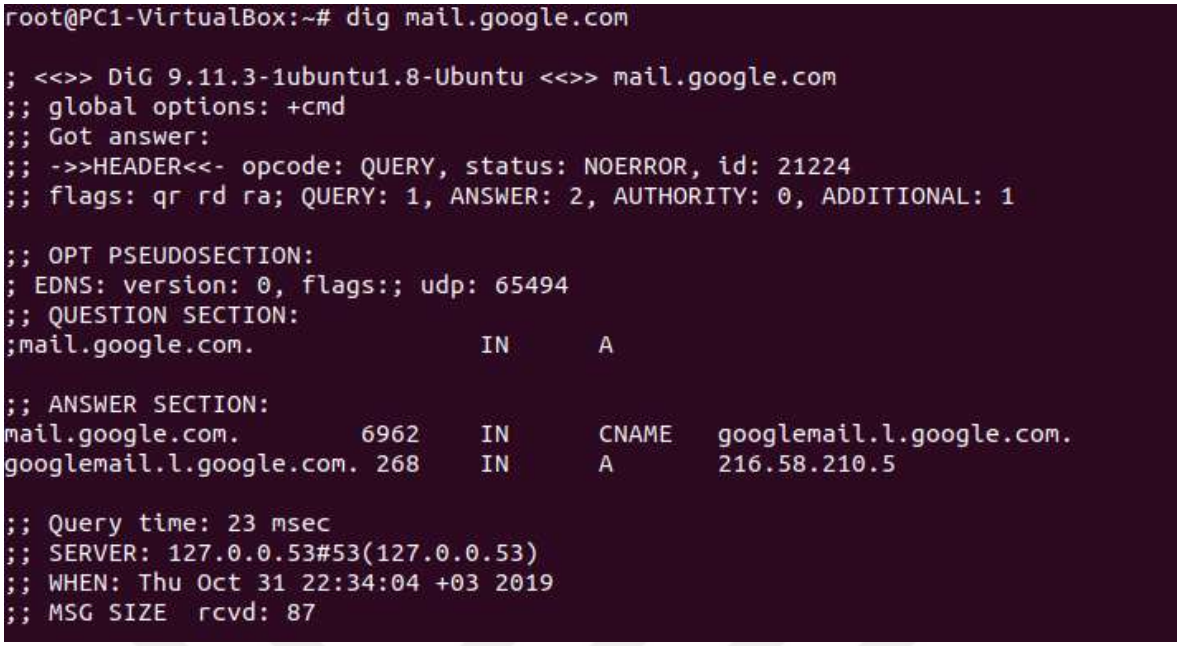
Toplamda 127 düzey DNS sunucusu olabilmektedir, bu nedenle bir arama birkaç istek için devam edebilmektedir. Yerel DNS sunucusu kendisine yapılan sorgulamaların sonucunu önbelleğinde saklamaktadır, böylece aynı adres tekrar sorgulandığında daha hızlı yanıt vermektedir. Yukarıda açıklanan DNS sunucusu, özyinelemeli bir DNS sunucusudur, çünkü arama sunucusunun içinde bulunması gereken yetkili DNS sunucusunu bulana kadar DNS sunucularını tekrarlı olarak sorgulamaktadır, ardından yanıtı iletmektedir. Özyinelemeli sorgulamaya bir alternatif ise yinelemeli sorgulamalardır. Yinelemeli sorgulama ile istemci, yerel DNS sunucusu yerine DNS adres çözümlemesini yapmaktadır [6]. Şekil 2.2'de, bir

yinelemeli sorgulama istemcisinin olay dizisi gösterilmektedir. Bu yinelemeli aramada sorgulanan ilk adımın yerel DNS sunucusuna yapılan bir sorgu olduğu görülmektedir.



Şekil 2.2. Yinelemeli DNS sorgulama senaryosu

Adres önbelleğinde ilgili adrese ait bir çözümleme varsa, bunu döndürmektedir, aksi takdirde en iyi adrese sahip DNS sunucusuna işaret etmektedir. DNS protokolü kayıt türlerine sahiptir. Bu kayıt türleri istek içerisinde bulunabilmektedir ve DNS protokolündeki mesajları yanıtlamaktadırlar [17]. Resim 2.1'de "mail.google.com" sorgusuna ait bir DNS cevap mesajı görülmektedir.



Resim 2.1. Örnek bir DNS sorgusu

- Soru bölümü: Bu bölüm, sorgulanan adresi içermektedir. Buradaki örnekte "mail.google.com" değerine karşılık gelmektedir.
- Cevap bölümü: Bu bölüm sorgulanan adrese ait IP adresini içermektedir. CNAME kaydı, aranan adresin diğer adı olan adres adlarını vermektedir. Bu kayıtlara kurallı ad kayıtları adı verilmektedir. Buradaki A kaydı mail.google.com ana bilgisayarının görseldeki IP adresine sahip olduğunu göstermektedir.
- Yetkili bölüm: Bu bölüm, adresin bulunduğu etki alanı için yetkili olan DNS sunucularını listelemektedir. NS kaydı, bu satırdaki NS etiketini izleyen ana bilgisayarın, satırın başında yazan etki alanı için bir DNS sunucusu olduğunu göstermektedir.
- Ek bölüm: Bu bölüm, etki alanı hakkında ek bilgileri listelemektedir. Sorgulama süresi, mesaj süresi gibi [2].

Her DNS kaydında bir Yaşam Süresi (TTL) alanı bulunmaktadır. TTL süresi dolmuşsa, bir DNS önbelleğe alma işlemini gerçekleştirir. TTL alanı, etki alanı yöneticilerinin etki alanlarında değişiklik yapabilmeleri için tasarlanmıştır. Daha yakın yıllarda etki alanı yöneticileri yük dengeleme için TTL alanını kullanmışlardır. Bu, TTL değerini, DNS sunucusunun yetkili DNS sunucusunu daha sık sorgulamaya zorlayan ve ardından

sunucunun yüküne eşit olan bir sunucu IP adresini döndürecek olan çok düşük bir değere ayarlayarak yapılmaktadır [6].

DNS protokolünün bir diğer özelliği hem UDP hem de TCP'nin bir taşıma mekanizması olarak kullanılması özelliğine sahip olmasıdır. DNS protokolünde UDP'nin ne zaman kullanılacağı ve TCP'ye ne zaman geçileceği ile ilgili bazı kurallar bulunmaktadır. Ana kural, bir istemci bir sorgu için bir DNS sunucusuyla bağlantı kurarsa ve bu sorgunun cevabı UDP paketini 512 bayttan daha büyük yaparsa, DNS sunucusunun yanıt almak için TCP kullanımının gerekli olduğunu belirten bir iletiye yanıt vermesidir. Bir istemcinin, UDP yerine DNS sunucusuyla bağlantı kurmasa bile UDP yerine TCP kullanmasına izin verilmektedir. Tipik olarak UDP kullanılmaktadır ancak TCP, bölge aktarımları için veya 512 bayt üzerindeki yükler için kullanılmaktadır [5]. Ayrıca DNS için Genişleme Mekanizmaları (EDNS) da bulunmaktadır. EDNS, bir DNS iletişiminde her iki ana bilgisayar tarafından destekleniyorsa, 512 bayttan büyük UDP yükleri kullanılabilmektedir. EDNS, DNS tünelleme için bant genişliğini artırmak üzere geliştirilebilecek bir özelliktir [6].

DNS, 30'un üzerinde kayıt türüne sahiptir ve bu kayıt türlerinin çoğu ortak internet servislerinde kritik öneme sahiptir. A kayıt tipi bir alan adını IPV4 adresine eşlemektedir. AAAA kaydı bir etki alanını bir IPV6 adresine eşlemek için kullanılmaktadır. CNAME kayıt türü, bir etki alanı adını kanonik adayla eşleştirmek için kullanılmaktadır. MX kayıt türü, bir etki alanı için posta sunucularını tanımlamak için kullanılmaktadır. NS kayıt türü, bir etki alanı için yetkili ad sunucularını tanımlamak için kullanılmaktadır. PTR veya işaretçi kaydı, bir IP adresini etki alanı adına eşlemek için yaygın olarak kullanılmaktadır. Bu genel olarak geriye doğru arama olarak adlandırılmaktadır. TXT kayıt türü, metin verilerini döndürmek için kullanılmaktadır [2].

DNS hiyerarşik bir sistemdir; Hiyerarşideki her seviye farklı sahipliğe sahip başka bir sunucu tarafından sağlanabilmektedir. İnternet için, A ile M etiketli 13 kök DNS sunucusu bulunmaktadır. Bunlar 13'ten fazla fiziksel sunucu tarafından temsil edilmektedir. DNS'in hiyerarşik yapısı bir örnek kullanılarak şu şekilde açıklanabilmektedir:

- yeni.test.gudekli.com adlı bir alanın IP adresi için örnek bir istek alınır.

- Yeni bir istek ilk önce hangi sunucunun .com üst düzey etki alanını kontrol ettiğini bulmak için kök sunuculara gider.
- .Com sunucusu gudekli.com etki alanını kontrol eden sunucuyu sağlayacaktır.
- Daha sonra gudekli.com DNS sunucusu test.gudekli.com etki alanını kontrol eden sunucuyu sağlayacaktır.
- Son olarak test.gudekli.com DNS sunucusu, yeni.test.gudekli.com için IP adresini sağlayacaktır [2].

Bu hiyerarşik sistemde, belirli bir etki alanı sahibi, etki alanları için yetkili sunucuları tanımlayabilmektedir. Bu, etki alanlarına yapılacak DNS sorgularının nihai hedef sunucusun kontrolünde oldukları anlamını taşımaktadır. Tipik bir işletmede, uç noktalar DNS isteklerini doğrudan internete yapmamaktadırlar. Bir uç noktaya DNS hizmetleri sağlayan dâhili DNS sunucularına sahiptirler. Bununla birlikte, yetkili ad sunucusuyla bağlantı kuruluncaya kadar DNS'nin isteklerini iletmeye devam edeceği göz önüne alındığında, dâhili bir uç noktaya erişimi olan bir saldırgan, denetlediği bir alana DNS tünellemesi için kuruluşun DNS altyapısından yararlanabilmektedirler [2].

2.2. DNS Güvenliği Uzantıları (DNSSEC)

DNSSEC [14], Alan Adı Sistemi Güvenlik Uzantısı anlamına gelmektedir. DNS sorguları için kimlik doğrulamaya imkân tanımakta ve bütünlük sağlamak için bir şifreleme mekanizması olarak kullanılabilmektedir [6].

DNS Güvenlik Uzantıları (DNSSEC), Etki Alanı Adı Sistemi'ndeki (DNS) güvenlik açıklarını gidermek ve çevrimiçi tehditlerden korumak için IETF tarafından oluşturulan bir standarttır. DNSSEC'in amacı, DNS güvenlik zayıflıklarını göz önünde bulundurarak bir bütün olarak İnternet güvenliğini artırmaktır. Temel olarak, DNSSEC, sistemi daha güvenli hale getirmek için DNS'e kimlik doğrulama yöntemleri eklemektedir. Yetkili DNS sunucusundaki DNS kayıtlarını ortak anahtar şifrelemesi ile dijital olarak imzalayarak çalışmaktadır [6].

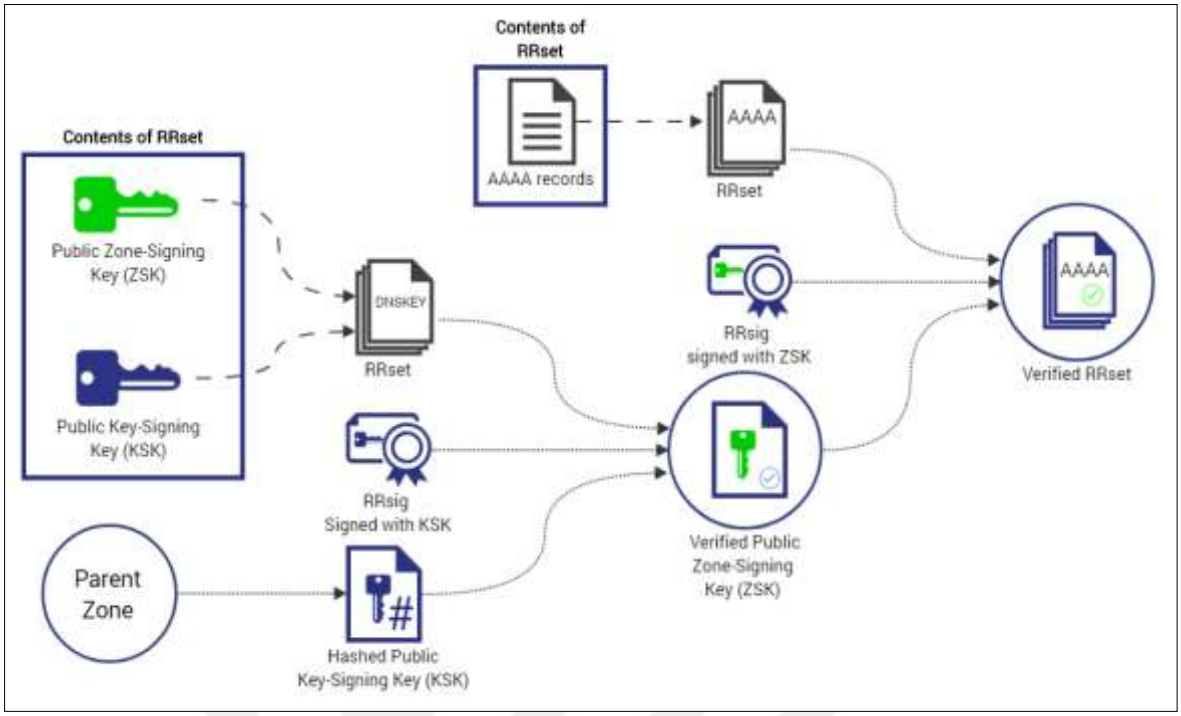
Dijital imzalama, kullanıcıların verilerin belirtilen kaynaktan kaynaklandığından ve aktarım sırasında değiştirilmediğinden emin olmalarını sağlamaktadır. DNSSEC, bir etki

alanı adının bulunmadığını da belirtebilmektedir. Bu özellikler internette güvenli bir ortam oluşturulması için önemli etkenlerdendir [6].

DNSSEC'de her alanın (zone) ortak / özel bir anahtar çifti vardır. Alanın genel anahtarı DNS kullanılarak yayınlanırken, alanın özel anahtarı güvende tutulmaktadır ve çevrimdışı olarak saklandığı düşünülmektedir. Bir alanın özel anahtarı, o alandaki tüm DNS verilerini imzalayarak DNS ile birlikte yayınlanan dijital imzalar oluşturmaktadır [6].

DNSSEC katı bir güven üzerine dayalı model kullanmaktadır ve bu güven zinciri ana alandan alt alana aktarılmaktadır. Yüksek seviye (ebeveyn) alanlara, alt seviye (çocuk) alanların genel anahtarlarını işaretlemektedirler veya bu anahtarları sağlamaktadırlar. Bu çeşitli alanlar için yetkili ad sunucuları, kayıt şirketleri, İnternet servis sağlayıcıları (ISS'ler), web hosting şirketleri veya web sitesi operatörleri tarafından yönetilebilmektedirler [6].

Bir son kullanıcı bir web sitesine erişmek istediğinde, kullanıcının bilgisayarındaki çözücü, web sitesinin IP adresini özyinelemeli bir ad sunucusundan istemektedir. Sunucu bu kaydı istedikten sonra, bölge ile ilişkilendirilmiş DNSSEC anahtarını da talep etmektedir. Bu anahtar, sunucunun aldığı IP adres kaydının yetkili ad sunucusundaki kayıt ile aynı olduğunu doğrulamasını sağlamaktadır [6].



Şekil 2.3. DNSSEC çalışma yapısı

Özyinelemeli ad sunucusu adres kaydının yetkili ad sunucusu tarafından gönderildiğini ve aktarım sırasında değiştirilmediğini belirlerse, etki alanı adını düzeltmekte ve kullanıcı siteye erişebilir durumda olmaktadır. Bu işleme doğrulama adı verilmektedir. Adres kaydı değiştirilmişse veya belirtilen kaynaktan değilse, özyinelemeli ad sunucusu kullanıcının sahte adrese ulaşmasına izin vermemektedir. DNSSEC, bir etki alanı adının bulunmadığını da kanıtlayabilmektedir. Bu sürecin bir sonucu olarak, DNS sorguları ve yanıtları MITM saldırılarından ve internet kullanıcılarını kimlik avı ve zararlı sitelere yönlendirebilecek sahteciliklerden korunmaktadır [6].

Alan adı sisteminin güvence altına alınması, İnternet altyapısının bütünlüğünün bir parçasıdır. Uygun şekilde bakımı yapıldığında, DNSSEC imzalı alanlar, MITM saldırılarını önleyerek ekstra güvenlik sağlamaktadır. DNSSEC uyumlu çözümlemeye sahip hiçbir istemcinin, DNS sahteciliği nedeniyle risk altında olmayacağı beklenmektedir. Kullanıcılar DNSSEC'in etkin olduğunun farkına varmayacakları gibi herhangi bir olumsuz etki de görmeyeceklerdir. DNSSEC'i ne kadar fazla etki alanı adı kullanıyorsa, daha fazla web sitesi ve e-posta adresi internette korunacaktır [6].

3. DNS'E YÖNELİK SALDIRILARA GENEL BAKIŞ

DNS hem özel ağlarda hem de internet üzerindeki en önemli servislerden birisidir. Bu hizmet olmadan internetten günümüzde olduğu gibi kolayca faydalanmak oldukça zor olacaktır. Bununla birlikte, DNS'in bu özelliği ve buna olan bağımlılık, onun dezavantajlarından biridir. Bir siteye veya uygulamaya yapılan her yeni bağlantı, genellikle hedefin IP adresi için DNS sorgusu ile başlamaktadır. Herhangi bir sebepten dolayı iade edilen IP yanlış ise, kullanıcı bilgisayar korsanlarının hedefi olabilmektedir. Böyle bir sorunun temel nedeni, DNS protokolünün herhangi bir referans veya güvenlik tasarımından yoksun olmasıdır. Bu nedenle, yıllar içinde, DNS Sunucuları saldırganların sürekli hedefi olmuştur [4].

DNS protokolü herhangi bir güvenlik içermediğinden, Etki Alanı Adı Sistemi Güvenlik Uzantıları (DNSSEC) RFC 3833 geliştirilmiştir. DNSSEC yeni bir protokoldür ve son zamanlarda kritik parçalarının sadece bazıları resmen tanımlanmıştır [6].

Yıllar boyunca saldırganlar, önceki saldırı türlerine karşı korunmak için ortaya konan savunma önlemlerine de saldırmak için yeni yollar bulmuşlardır [18]. Bu bölümde, DNS sunucularına yönelik farklı saldırı türleri açıklanacaktır. DNS sunucularına yönelik saldırılar temelde iki gruba ayrılmaktadır:

- DNS sunucularını hedef alan saldırılar ve
- Başka bir sistemi hedef alarak DNS sunucularını kullanan saldırılar

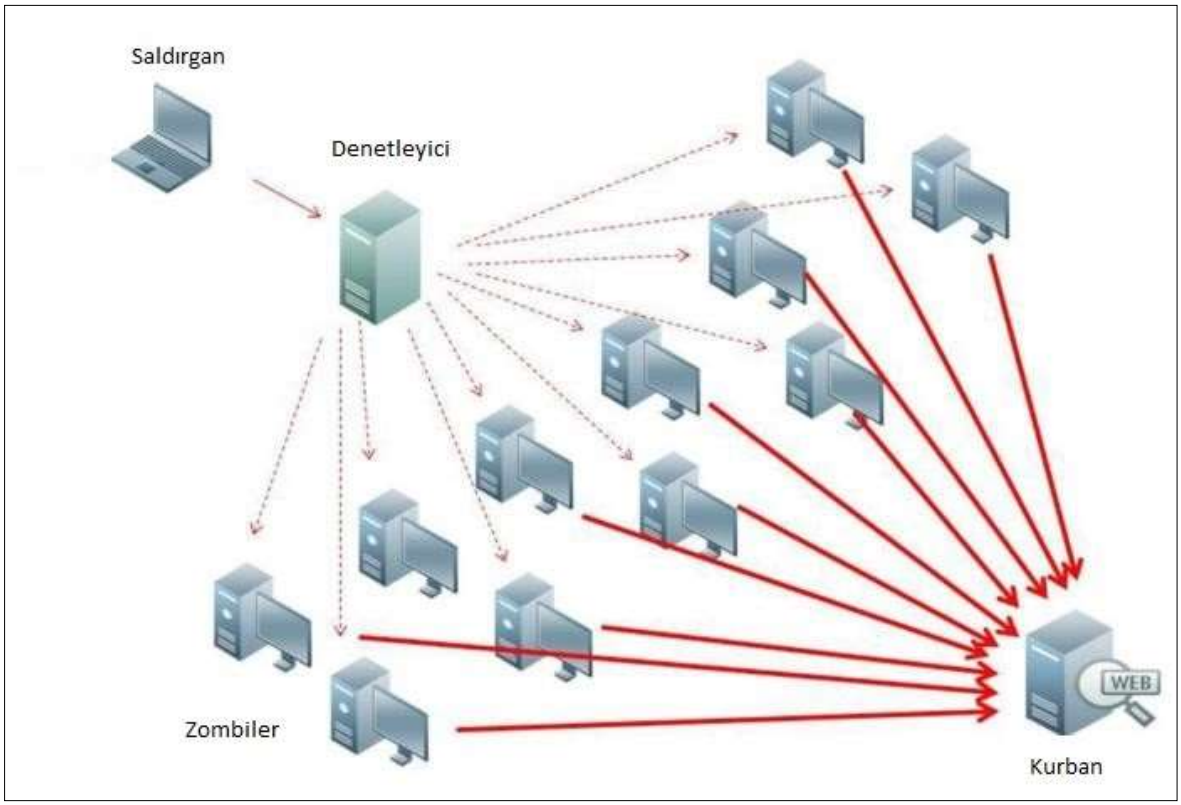
3.1. DNS Sunucularını Hedefleyen Saldırılar

Bu bölümde, literatürde yer alan, doğrudan DNS sunucularını hedefleyerek gerçekleştirilen farklı saldırı türleri açıklanmaktadır. Temel DoS saldırısı, DNS önbellek zehirlenmesi, arabellek aşımı bu grup içerisinde yer almaktadır.

3.1.1. Temel DoS saldırısı

Bir DNS sunucusu ağdaki bir ana bilgisayar olduğundan, Internet'teki diğer ana bilgisayarlarla aynı şekilde saldırıya uğrayabilmektedir. Bununla birlikte, bir DNS sunucusunda mevcut olan yüksek bant genişliği nedeniyle, bir DoS saldırısı, ilgili DNS

sunucusu çok küçük bir ölçeğe sahip olmadıkça DNS sunucularının performansı üzerinde neredeyse hiçbir etkiye sahip olmayacaktır. Bir saldırganın bir DNS sunucusuna saldırmak için Botnet DDoS saldırısı veya Yönlendirme DDoS saldırısı kullanması çok daha yüksek olasılığa sahiptir [9].



Şekil 3.1. Temel bir DoS saldırısı

DoS saldırısında, tek bir ana bilgisayar internetin DNS sunucusunu zorlamaya çalışmaktadır. Saldırıya uğraması gereken tek hedef olduğu belirtilmesine rağmen, çoğu durumda küçük bir saldırgan grubunun bu saldırıyı gerçekleştirme ihtimalinin daha yüksek olduğunu ve saldırının doğrudan saldırganın bilgisayarından yapıldığının düşünülmesi gerekmektedir. DoS saldırısında aşağıdaki özellikler görülmektedir [9]:

- Çok az sayıda ana bilgisayarın çok fazla sayıda istek oluşturması: Ancak, DNS kara listeleri kullanan spam filtreleri gibi programlar da bulunmaktadır. Bu özellik yalnızca, saldırgan IPv4 kaynak adresini taklit etmediğinde görülmektedir [12].
- Temas eden ana bilgisayarların sayısındaki artış: Eğer saldırgan IPv4 kaynak adresini taklit etmişse, saldırı sırasında DNS sunucusuyla iletişim kuran ana bilgisayarların sayısı artacaktır [12].

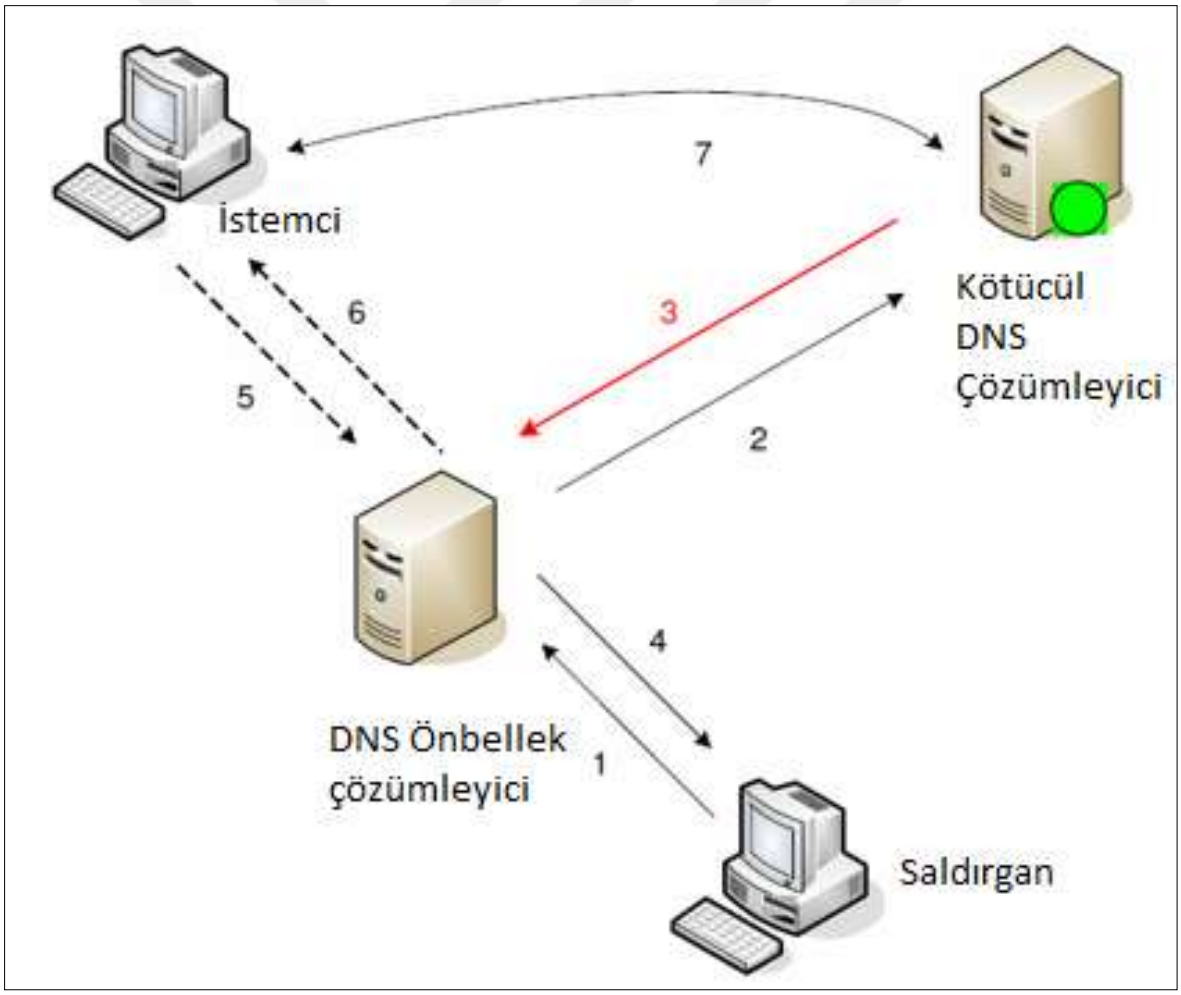
- Yanıt sayısının azaltılması: Bir DoS saldırısındaki esas fikir, saldırıya uğramış sunucuyu, her isteği işleyemeyecek kadar büyük miktarda veriyle yoğunlaştırmak, böylece normal sorguların işlenmemesini sağlamaktır. Bu, DNS sunucusundan dönen yanıtların sayısının DNS sunucusunda daha fazla sorgu olduğu algısına yol açmamalıdır [17].
- Ortalama paket büyüklüğü: Boş bir DNS paketinin 40 bayt olacağı hesaplanabilmektedir (UDP başlığı için 8 Bayt + 20 Bayt IPv4 başlığı + 12 Bayt DNS başlığı). Bir hata yanıtını tetiklemeyen en küçük sorgu, 5 bayt sorgu verisi olan 45 baytlık bir pakettir. Bir DoS saldırısı sırasında saldırgan, bozuk paketler (44 bayt veya daha az) kullanabilmekte veya DNS sunucusunu kaynak yoğun işlemler gerçekleştirmeye zorlayan veya mevcut bant genişliğinin büyük bir bölümünü kaplayan başka büyük paketler kullanmaktadır. Bu özellikteki varsayım, bir DoS saldırısında, DNS sunucusuna büyük paketlerin gönderildiği varsayımıdır [9].

3.1.2. DNS önbellek zehirlenmesi

DNS girişlerinde imza doğrulaması olmaması, bir saldırganın bir etki alanı veya ana bilgisayar için yanlış çözümlemeler enjekte etmesine neden olmaktadır. Önbellek zehirlenmesi özyinelemeli sorgulama veya bir son ana bilgisayar ile bir DNS sunucusunda yapılmaktadır. Saldırı, bir ana bilgisayarın adres çözümleme istediği anda başlamaktadır. Önbellek zehirleyicisi, sorguya, sorgulanan DNS sunucusundan daha hızlı tepki vermeye çalışacaktır, çünkü sorgunun cevabı kabul edilecektir. Bunun çalışması için saldırganın, bu adres çözümlemesi için hangi DNS sunucusunun yetkili DNS sunucusu olduğunu bilmesi ve sonra kaynak adresini bu DNS sunucu adreslerine yönlendirmesi gerekmektedir. Yetkili DNS sunucusunun zamanında yanıt vermemesini sağlamak için bir yaklaşım, üzerinde bir DoS saldırısı yaparak yavaşlatmaktır. Bu tür saldırı mümkündür, çünkü buna karşı tek koruma, bir DNS paketindeki Kimlik (QID) alanıdır. Günümüzde kullanılan tüm DNS sunucuları, QID alanı için Sözde Rastgele Sayı Üretimi (PRNG) uygulamaktadır. Ancak bu alan sadece 16 bit uzunluğundadır ve PRNG'nin kalitesi olması gerektiği kadar iyi değildir. Son zamanlarda Dan Kaminsky tarafından yeni bir DNS önbellek zehirlenmesi saldırısı bulunmuştur ve bu da saldırganın saldırıyı 10 saniye kadar kısa bir sürede gerçekleştirdiğini göstermiştir [19]. Bu saldırı için saldırganın internet üzerinde yayınlanan bir web sitesi kurması gerekmektedir. Bu web sitesi, saldırmak istediği alandaki bir ana bilgisayara çok sayıda bağlantı içermelidir. Bu bağlantılara, aaa.gudekli.com, bbb.gudekli.com gibi bir ana bilgisayar adı verilecektir. Şekil

3.2'de, bu saldırının adımları gösterilmektedir. Bir istemci bu web sitelerini açtığında, bu bağlantılardaki adresleri çözmeye çalışacaktır. gudekli.com için Yetkili DNS sunucusu bu misafiri tanımayacak ve yalnızca A kaydını değil, NXDOMAIN değerini döndürecektir. Bu işlem sırasında saldırgan, talep edilen ana bilgisayar adının IP adresini bildiğini söyleyerek yanıt veren DNS sunucusuna yanıt vermeye çalışacaktır. Bununla birlikte, bir DNS sunucusunu da gudekli.com alanı için yetkili DNS olarak belirtecektir [19].

Web sitesi çok sayıda ana bilgisayar adı içereceğinden, saldırganın yetkili DNS sunucusundan çok daha yüksek yanıt verme olasılığı nedeniyle saldırganın yanıtında daha hızlı olması gerekmektedir. Bu işlemlerin sonunda saldırganın girişimi gudekli.com için DNS önbelleğini zehirleyebilmektedir [19, 20].



Şekil 3.2. Önbellek zehirlenmesi şeması

DNS önbellek zehirlenmesi saldırısının temel tanımından itibaren, saldırının yetkili ad sunucusundaki enjeksiyon ve olası bir DoS olmak üzere iki bölümden oluştuğu bilinmektedir. Bu, DNS önbellek zehirlenmesi saldırısı için aşağıdaki özelliklere yol açmaktadır:

- DNS başlığında sorgu kimliği olan tek bir sorguya ait yüksek yanıt sayısı: Bu büyük yanıt hacmi, sorgu kimliği tahmininden kaynaklanmaktadır. DNS yazılımında kullanılan çok sayıda Sözde Rastgele Sayı Üreticisinden (PRNG), belirli sayıda paket içinde PRNG tarafından üretilecek bir sonraki sorgu kimliğinin ne olacağını tahmin etmek mümkündür. DNS sunucularına bağlı olarak PRNG cevaplarının sayısı değişmektedir. Yanıt paketlerinin içeriği net olarak görülemez, ancak çok fazla cevap görülebilir.
- Belirli bir DNS sunucusuna yönelik artan özyinelemeli DNS istekleri sayısı: Yakın zamanda bulunan bir saldırı ihtimalinde, saldırgan, bir etki alanındaki çok sayıda ana bilgisayar için ana bilgisayar çözümleme talebinde bulunacaktır.
- Yetkili ad sunucusunda olası DoS saldırısı: Yetkili ad sunucusunda yapılan DoS saldırısı, Temel DoS saldırısı ile aynı özelliklere sahip olacaktır, ancak yetkili ad sunucusunun saldırı altında olduğundan başka bir çıkarım yapılamamaktadır [19].

3.1.3. Arabellek aşımı

Birçok yazılımda olduğu gibi, DNS yazılımındaki hataların varlığı, bazı DNS sunucularına daha fazla saldırı gerçekleştirmesini mümkün kılmaktadır. Arabellek aşımı, yazılım, alıcının olası bir içeriğin daha büyük halini yanlış kullanması durumunda, alıcının bitişik belleğe yüklenmesine neden olan bir işlem söz konusu olduğunda gerçekleşebilmektedir. Yanındaki belleğe kayan veri baytları, çalışacak kodu oluşturmaktadır. Bir saldırganın, bir DNS sunucusunu sorgulayarak veya yanıt veren bir yetkili ad sunucusu olarak, arabellek aşımını gerçekleştirebileceği iki yöntem bulunmaktadır [21, 22].

Birincisi, saldırganın bir DNS sunucusunda bir sorgu gerçekleştirmesidir. Bu durumda, saldırgan tarafından yapılan sorgu, DNS sunucusu yazılımının bir akıma neden olacağı şekilde değiştirilecektir. Bu manipülasyon, verilerinin kendisinden tespit edilememektedir. Yükü aşan bir paketin varlığını gösteren tek etmen gönderilen paketin büyüklüğüdür. Yük, bir paketten sonra başarılı bir şekilde taşınmış olabilir, bu nedenle paket boyutunda net bir değişiklik görülmeyebilmektedir [23].

Diğer durum ise, saldırganın yetkili bir ad sunucusunu kullanmasıdır. Bu durumda, saldırgan bir talebe cevaben gönderdiği kaynak kayıtlarını manipüle etmektedir, daha sonra manipüle edilen kaynak kayıtları, saldırıya uğramış DNS sunucusu tarafından okunduğunda alıcıya teslim edilmesine neden olmaktadır.

Arabellek aşımları aşağıdaki özellikleri bulundurmaktadır:

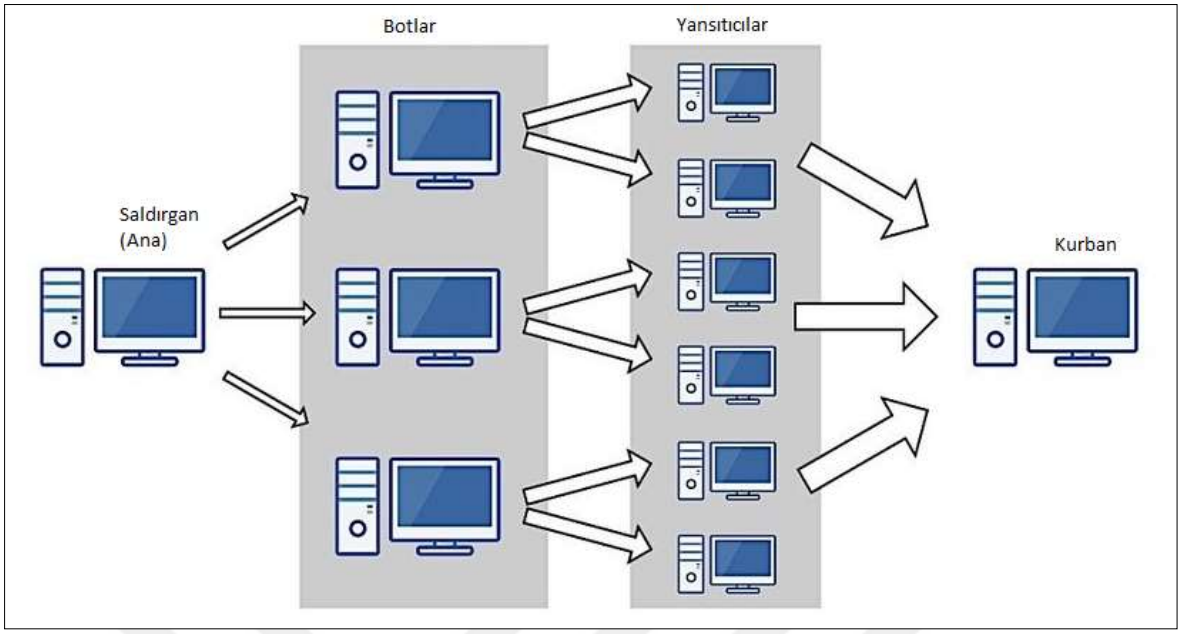
- Kötü biçimlendirilmiş paketler: Bir DNS paketinin içeriği, DNS sunucusu yazılımının yükleneceği şekilde manipüle edilmektedir.
- Artırılmış paket büyüklüğü: Bir paketin manipüle edilen verilerin yükünü azaltabilmek ve kullanılabilir bir kod çalıştırmak için yeterince büyük olması gerekmektedir [23].

3.2. DNS Sunucularını Kullanan Saldırılar

Bu bölümde, literatürde yer alan DNS sunucuları kullanılarak gerçekleştirilen farklı saldırı türleri açıklanmaktadır. Yansıtma saldırıları ve dns tünelleme bu grupta yer almaktadır.

3.2.1. Yansıtma saldırıları

Yansıtma saldırısında hedeflenen konağa yönelik paketler olduğu için, saldırganlar geçerli bir hizmeti kullanmaktadırlar. DNS sunucularındaki yansıtma saldırılarında bir DNS sunucusundan gelen bir yanıt paketi çoğu durumda orijinal gönderme paketinden üç kat kadar büyüktür. Bu saldırı yönlendirme saldırıları ile bir DoS saldırısının etkinliğini artırmaktadır. Eğer bir saldırgan www.gudekli.com adresinin çözümlenmesini sorgularsa, DNS sunucusu neredeyse beş kat daha büyük bir paketle cevap verecektir. Bunun gibi çok sayıda sorgunun cevabı mağdurlara gönderilirse, saldırgan daha az kaynakla hedefine ulaşmış olacaktır. TCP protokolü kullanarak bir saldırı yapmak da mümkündür. Saldırganların kullandığı TCP'nin özelliği, gönderilen paketler diğer ana bilgisayar tarafından kabul edilmezse, bir ana makinenin belirli bir süre yeniden paketleri göndermesini sağlayan yeniden iletim mekanizmasına sahip olmasıdır [24-26]. Şekil 3.3’te örnek bir yönlendirme saldırısı görüntülenmektedir.



Şekil 3.3. Örnek bir yönlendirme saldırısı

Yönlendirme saldırılarında aşağıdaki özellikler bulunmaktadır:

- Bir yönlendirme saldırısında, tek bir saldırgan bilgisayarın çok fazla DNS çözümlemesi istediği görülmektedir. İlk bakışta, kısmen doğru olan bir DoS saldırısı olarak görünmektedir. Ancak, son ana bilgisayar DNS sunucusu yerine hedef olduğundan, paket simetrisinde herhangi bir anormallik görüntülenmeyecektir. Bu ifadenin bir karşı örneği durumda ise, saldırganlar saldırılarını başlattıklarında, çoğu durumda saldırgan ana bilgisayarla aynı ağda bulunmamaktadır. Veri toplama noktası, DNS sunucusuna yapılan sorgunun bulunmamasına yol açmaktadır [24-26].
- Bazı durumlarda, bu saldırıdaki verilere bakıldığında, bağlantı kurulan DNS sunucularının sayısındaki artışla birlikte birden fazla alan adının kullanıldığı görülebilecektir.
- Amplifikasyon: Yansıtma saldırılarının kullanılmasının en önemli sebeplerinden biri, istek mesajına kıyasla cevap mesajındaki oktetlerin dağılımıdır. Bir sorguya verilen büyük bir yanıt paketi, DNS takibi için çok normaldir, bu nedenle bir saldırı için amplifikasyon kullanımını tespit etmek daha zor olabilmektedir [27].
- Yeniden Aktarımlar: TCP tabanlı aktarımda, onaylanmamış bir paketin yeniden aktarımı gerçekleşebilmektedir. Yansıtma saldırılarında da, bir paketin kabul edilmediği durumlarda sorgu yanıtının yeniden iletimi gerçekleşmektedir [27].

3.2.2. DNS tünelleme

DNS tünelleme tam anlamıyla bir saldırı olarak kabul edilmemektedir, ancak DNS sisteminin kötüye kullanılması olarak bilinmektedir. DNS tünelinde, DNS paketinin içeriği, verilerin (ortam, virüsler) DNS yoluyla tünel ağındaki DNS sunucusu gibi davranan bir uygulama sunucusuna gönderilebilmesi için değiştirilmektedir. DNS paketinin değiştirilmiş bölümleri ise kayıtlardır. Sorgu türü herhangi bir tür olabilmektedir, BASE64 karakter kodlamasının uygulanmasına izin veren TXT kaydı gibi belirli değerler daha çok dikkat çekmektedir, ancak A kaydı yalnızca BASE32 karakter kodlamasına izin vermektedir ve 255 bayttan fazla verilere izin verilmemektedir [28].

4. DNS'E YÖNELİK SALDIRILARA KARŞI TESPİT YÖNTEMLERİ

Ağ tabanlı hizmetlerin kullanımının artması ve ağlarda hassas bilgilerin kullanılması, internetin medya olarak kullanılması ile bilgi güvenliğinin dikkatlice ele alınması gereksinimini ön plana çıkmaktadır. Güvenli bir ağ için aşağıdaki unsurlar sağlanmalıdır:

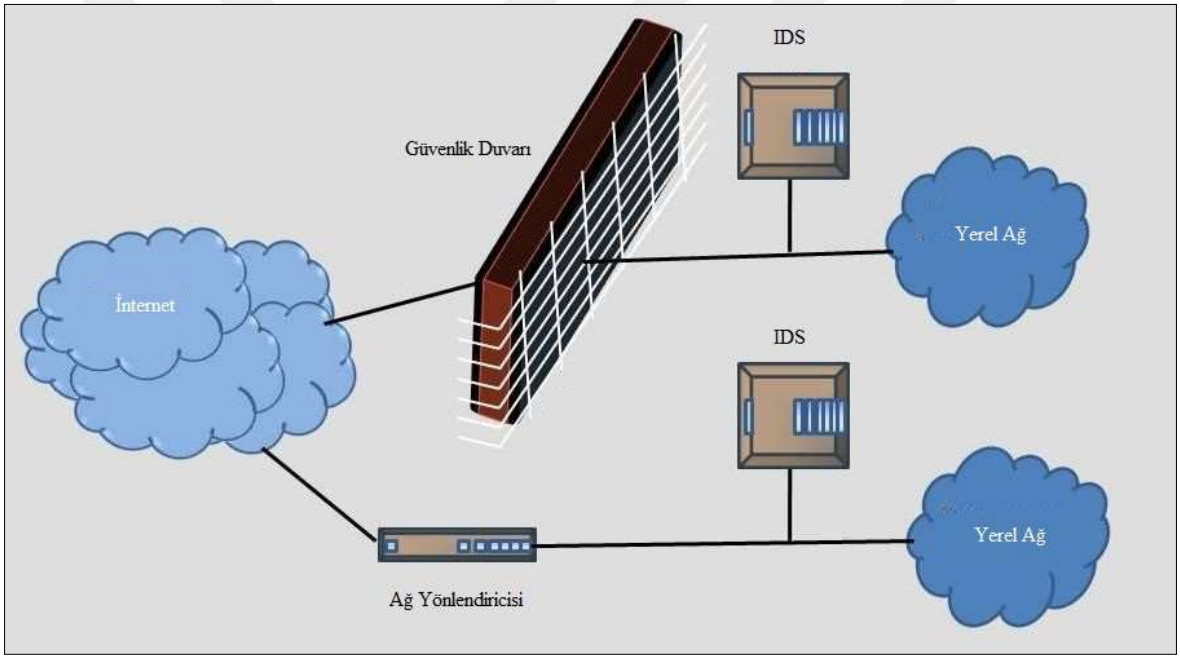
- Veri gizliliği: Gizlilik özelliği, belirli bilgilere erişme yetkisi olan kişilerin girmesine izin verildiğini belirtmektedir.
- Veri bütünlüğü: Yalnızca yetkili kuruluşlar sistem içindeki bilgileri değiştirebilir. Bu, gerekmedikçe bilgilerin değiştirilmesini önleyen özelliktir.
- Veri kullanılabilirliği: Uygunluk, bir sistemdeki bilgilerin gerektiğinde elde edilebileceği özelliktir.

Güvenlik risklerini önlemek için savunma katmanları kullanılmaktadır. Başlıca önlemler filtreleri ve güvenlik duvarlarını içermektedir. Hostingler ayrıca, virüsten koruma ve casus yazılım önleme teknolojileri kullanarak, gereksiz hizmet taleplerini ortadan kaldırarak, kullanıcı kimlik doğrulama ve erişim denetimleri uygulayarak proaktif bir şekilde müşterilerini korumaktadırlar. Özellikle, izinsiz giriş tespit sistemleri (IDS'ler) ağa yönelik gerçekleştirilen harici saldırılara karşı direnmek için yardımcı olmaktadır. Yani, IDS bir bilgisayar sisteminin veya ağın yetkisiz faaliyetlerini tespit etmek için kullanılan bir güvenlik aracıdır.

İnternet ve yerel ağların ticarileşmesi nedeniyle bilgisayar sistemlerinin saldırı aktiviteleri artmaktadır. Bir IDS, bilgileri toplamakta ve nadir veya beklenmedik olaylar için analiz işlemleri gerçekleştirmektedir. Kimlik ve güvenlik sorunlarının belirtilerini tespit etmek için bir bilgisayar sisteminde meydana gelen olayları izleme ve analiz etme sürecidir. Şekil 4.1 IDS'nin yapısını göstermektedir. IDS, ilk olarak 1980 yılında James P. Anderson isimli bilimci tarafından tanıtılmıştır [11]. Kullanıcı davranışındaki anormallikleri tespit etmeye dayalı bir güvenlik izleme sistemi geliştirmek için izinsiz girişleri sınıflandıran bir tehdit modeli oluşturmuştur. 1986'da Dr. Dorothy Denning, istatistiklere dayanarak IDS için çeşitli Markov zincirler, zaman serileri gibi farklı modeller önermiştir [11]. Biermann ve arkadaşları 2001'de farklı IDS yaklaşımlarını karşılaştırmak ve değerlendirmek için bir çalışma sunmuşlardır [29].

İzinsiz Giriş Tespit Sistemleri (IDS), ağ yöneticilerinin politika ihlallerini tespit etmelerine izin verdiği için güvenlik altyapılarında standart bir bileşen haline gelmiştir. Bu politika ihlalleri, erişimi kötüye kullan dış saldırganlardan ve içeriğe yetkisiz erişim sağlamaya çalışanlardan kaynaklanmaktadır. Bunun yanında mevcut IDS'lerin birçok önemli dezavantajı bulunmaktadır:

- Yanlış pozitifler: Normal saldırı yanlışlıkla kötü niyetli olarak sınıflandırıldığında ve ona göre davranıldığında yanlış pozitif durumları oluşmaktadır.
- Yanlış negatifler: Bir saldırının gerçekleştiği sırada IDS'in bir uyarı oluşturmadığı durumlarda meydana gelmektedir.



Şekil 4.1. IDS çalışma yapısı

4.1. Saldırı Tespit Yaklaşımları

Geleneksel olarak, izinsiz giriş tespit teknikleri iki geniş kategoride sınıflandırılmaktadır:

- Yanlış kullanım veya imza temelli IDS

Yanlış algılama, imza olarak depolanan iyi bilinen saldırıların izlerini veya kalıplarını aramaktadır. Bu imzalar, geniş kapsamlı saldırı tekniklerine dayanan bir uzman tarafından

sağlanmaktadır. Bu işlemde, eşleşen bir örüntü bulunursa alarm oluşturularak ilgili olay işaret edilmektedir.

- Anomali tabanlı IDS

Anomali saptama, normal kullanıcı veya sistem davranışı modelini kullanmaktadır ve bu modelden önemli ölçüde sapma potansiyeli taşımaktadır. Bu normal kullanıcı veya sistem davranışı modeli genellikle kullanıcı veya sistem profili olarak bilinmektedir. Denetimli, yarı denetimli ve denetimsiz anomali tespit teknikleri olmak üzere üç tip anomali tespit tekniği kategorisi bulunmaktadır.

- Hibrit IDS veya bileşik tespit sistemi

Hem yanlış hem de anormallik tespit tekniklerinin birleşimi olarak bilinmektedir. Hem sistemin normal davranışına hem de davetsiz misafirlerin müdahaleci davranışına dayanan bir "karma model" kullanarak kararlar almaktadır.

IDS'ler analiz ettikleri girdi bilgisine göre de ana bilgisayar tabanlı ve ağ tabanlı izinsiz giriş tespit sistemi olarak kategorilere ayrılmaktadırlar.

- Ana Bilgisayar Tabanlı IDS (HIDS): Ana Bilgisayar tabanlı izinsiz giriş tespit sistemleri, çoğunlukla tek bilgisayarla ilgilenmektedirler ve sistem çağrısı, çekirdek, güvenlik duvarı ve sistem günlüklerine dayalı izinsiz giriş tespitini gerçekleştirmektedirler.
- Ağa dayalı IDS (NIDS): Ağa izinsiz giriş algılama sistemi büyük oranda sağlıklı çalışmaktadır. Ağ trafiğini izlemekte, trafiği incelemekte ve trafiği normal veya şüpheli olarak kategorize etmektedir [29, 30, 31].

4.2. IDS'de Veri Madenciliği Teknikleri

İzinsiz giriş tespiti için birkaç teknik bulunmaktadır. Veri madenciliği izinsiz giriş tespitinde kullanılan etkili tekniklerden biridir. Veri madenciliği, veri tabanlarında bilgi keşfi olarak bilinmektedir. Büyük miktarda veriden faydalı desenler bulan bir süreçtir. Veri madenciliği teknikleri, temel olarak tanımlayıcı madencilik ve tahmine dayalı madencilik olmak üzere iki kategoride sınıflandırılmaktadır [21].

IDS'lerde veri madenciliği yaklaşımları aşağıdaki nedenlerle kullanılmaktadırlar:

- Bilginin kullanımı ve resmileştirilmesini gerektiren sıradan programlama dillerini kullanarak IDS'yi programlamak çok zor bir süreçtir.
- Makine öğreniminin uyarlanabilir ve dinamik yapısı, bu durum için uygun bir çözüm olmasını sağlamaktadır [21, 32, 33].

Bir IDS'nin ortamı ve sınıflandırma görevi tercihlere bağlı bir durumdur. Bir ortamda olay gibi görünen durum, diğer ortamlarda normal sayılabilmektedir.

Veri madenciliğinin izinsiz giriş tespitine katkıda bulunabileceği birkaç özel durum bulunmaktadır:

- Analistlerin gerçek saldırılara odaklanmalarını sağlamak için normal etkinliği alarm verilerinden kaldırması gerekmektedir.
- Yanlış alarm üreticilerinin ve kötücül sensör imzalarının tanımlanması gerekmektedir.
- Gerçek bir saldırıyı ortaya çıkaran anormal aktivitelerin bulunması gerekmektedir.
- Farklı IP adreslerinden gerçekleştirilen aynı faaliyetler gibi uzun ve devam eden kalıpların tanımlanması gerekmektedir [20, 21, 32, 33].

4.2.1. İstatistiksel teknikler

"Yukarıdan aşağıya" öğrenme olarak da bilinen istatistiksel teknikler, ilişkinin araştırıldığı ve araştırmamıza yardımcı olmak için matematik kullanılabileceği konusunda bir fikrimiz olduğunda kullanılır. Doğrusal, regresyon eğrisi gibi doğrusal olmayan ve karar ağaçları olmak üzere üç temel istatistiksel teknik sınıf bulunmaktadır. İstatistikler ayrıca Markov modelleri ve Bayes tahmin edicileri gibi daha karmaşık teknikleri de içermektedir. Gizli bir Markov modeli (HMM), modellenmekte olan sistemin bilinmeyen parametrelere sahip bir Markov işlemi olduğu varsayıldığı ve zorlukların gözlenebilir parametrelerden gizli parametreleri belirlemesini mümkün kılan istatistiksel bir modeldir. Çıkarılan model parametreleri daha sonra örüntü tanıma uygulamaları gibi daha fazla analiz yapmak için kullanılabilmektedir [29, 31].

4.2.2. Makine öğrenmesi

Makine öğrenme teknikleri, çıkarılacak kalıpların ne olabileceğine ilişkin önceden hiçbir bilgisi olmayan öğrenme kalıplarına çok uygundur, bu nedenle bazen "aşağıdan yukarıya" öğrenme teknikleri olarak adlandırılmaktadır.

Sınıflandırma tabanlı bir IDS, tüm trafiği normal veya kötü amaçlı olarak sınıflandırmaya çalışmaktadır. Bu yaklaşımdaki birincil zorluk, sistemin bu kalıpların ne olduğunu tam olarak öğrenebilmesidir. Elde edilecek sonuçta hem düşmanca olmayan bir faaliyetin işaretlenip işaretlenmediği (hatalı pozitif) hem de kötü niyetli faaliyetin kaçırılmayacağı (yanlış negatif) açısından sistemin doğruluğu etkilenmektedir [29].

Endüktif kural oluşturma

RIPPER tarafından yapılanlar gibi endüktif kural oluşturmanın, trafiği normal veya çeşitli izinsiz giriş şekillerine göre sınıflandıran bir sistem üretmede oldukça etkili ve kolay bir yöntem olduğu gösterilmiştir. Sistem, düzgün şekilde sınıflandırmak için ağ trafiğine uygulanabilecek bir ilişkilendirme kuralı içermektedir [29].

Avantajları: Oluşturulan kural kümesinin anlaşılması kolaydır; dolayısıyla bir güvenlik analisti bunu doğrulayabilmektedir. Bir başka avantajlı kılan özelliği ise, çoklu kural kümelerinin bir meta-sınıflandırıcı ile üretilip kullanılabilmesidir.

Genetik algoritmalar (GA)

Genetik algoritmalar çoğunlukla ortamları ve bir popülasyon içerisindeki bireylerin davranışlarını simule etmek için kullanılmıştır. Bilgisayar biliminde genetik algoritmaları, doğal seleksiyon işleminden ilham alan bir sezgisel ötesi algoritma olarak tanımlanabilir. Genetik algoritmalar, mutasyon ve çaprazlama gibi operatörler aracılığıyla optimizasyon ve arama problemlerinde yüksek kaliteli çözümler üretmek için yaygın olarak kullanılmaktadır [29].

Avantajları: Çok fazla veriye ihtiyaç duymamaktadır.

Dezavantajları: Karmaşık bir yöntemdir ve genelden çok özel bir şekilde kullanılmaktadır.

Bulanık mantık

Klasik yüklem mantığından kesin olarak çıkarılmak yerine, yaklaşık olan muhakeme ile ilgili bulanık küme teorisinden türetilmiştir. “Karmaşık bir problem için gerçek dünya değerleriyle iyi düşünülmüş bulanık küme teorisinin uygulama tarafı” olarak düşünülmektedir. Luo çalışmasında bulanık birleştirme kuralları madenciliği algoritmasını ek bir normalizasyon adımıyla kullanmıştır [29]. Bulanık olmayan yöntemlere göre yanlış pozitif oranında anlamlı bir azalma olduğunu göstermiştir.

Avantajları: Nicel özellikler için kullanılmaktadır ve belirsiz bazı problemlerde daha iyi esneklik sağlamaktadır.

Dezavantajları: Algılama doğruluğu yapay sinir ağlarına oranla daha düşüktür [31, 34, 35].

Yapay sinir ağları (YSA)

İzinsiz giriş tespitinde uygulanan ana modeller olan Çok Katmanlı Algılayıcılar (MLP'ler) ve Kendi Kendini Düzenleyen Haritaları (SOM'lar) içeren çeşitli modelleri kapsamaktadır. YSA'ların yanlış tespit uygulamalarının çoğu, ileri beslemeli MLP'ler olarak uygulanmaktadır. Konak tabanlı uygulamalar tipik olarak anomali tespit sistemleri olarak uygulanırken yanlış algılama uygulamalarının çoğu ağ tabanlıdır. Avantajları: YSA'daki çoklu gizli katmanlar yapılandırılmamış ağ paketini verimli bir şekilde sınıflandırmaktadırlar, sınıflandırma verimliliğini arttırır, bağımlı ve bağımsız değişkenler arasındaki karmaşık doğrusal olmayan ilişkileri, gürültülü verilere duyulan yüksek toleransı dolaylı olarak saptayabilmektedirler.

Dezavantajları: Daha fazla zaman ve daha fazla örnek eğitim aşaması gerektirmektedir, Daha az esnekliğe, "Kara kutu" niteliğine ve büyük hesaplama yüküne sahiptir [29, 31, 34, 35].

İmmünolojik yöntem

Hofmeyr ve Forrest isimli araştırmacılar 1999 yılında immünolojik kavramlara dayanan ilginç bir teknik sunmuşlardır. Normal trafikten gelen bağlantıları "öz" olarak

tanımlamışlardır ve daha sonra çok sayıda makineye normal trafiğin bir parçası olmayan bağlantı örnekleri oluşturmuşlardı. Bu örnekler bayt yönelimli bir karma yöntem ve permütasyon kullanılarak üretilmiştir [31, 34-36].

Kümeleme

İzinsiz giriş tespitinde faydalı bir yöntem olarak bilinmektedir. Kötü niyetli etkinlikler, kendisini kötü amaçlı olmayan etkinliklerden ayırarak kümelenmektedirler. Kümeleme algoritmalarına örnek olarak K-Means, Küresel-Gauss, Kendi Kendini Düzenleyen Haritalar verilmektedir [31, 34, 35].

Avantajları: Kümelenme bir denetimsiz öğrenme yöntemidir. Verilerin etiketlenmesine ihtiyaç duyulmamaktadır ve verilerdeki doğal desenler çıkarılmıştır. Eğitim için etiketli bir veri setinin kullanılması gerekmemektedir [31, 34, 35].

K-en yakın komşu (k-NN)

Örnekleri sınıflandırmak için uygulanan parametrik olmayan bir tekniktir. Giriş vektörlerindeki farklı noktalar arasındaki yaklaşık mesafeler hesaplanmaktadır ve daha sonra etiketlenmemiş nokta K en yakın komşusunun sınıfına atanmaktadır. Sınıflandırma aşamasında, k-NN yerel olarak en uygun bir hipotez fonksiyonunu belirlemek için benzerlik temelli bir arama stratejisi kullanmaktadır [31, 34-36].

Avantajları: Uygulamada basittir. Paralel uygulamalara çok kolay katılmaktadır, model eğitim aşamasını içermemektedir.

Dezavantajları: Büyük örnek kümeleri için performans maliyetli olabilmektedir. Örnek tabanlı öğrenme, test kayıtlarının sınıflandırılmasında ilişkili olmayan niteliklere duyarlı olabilmektedir ve yavaştır [31, 34-36].

Kendi kendini düzenleyen harita (SOM)

Kendi kendine yapılanma süreci olan, denetlenmeyen bir rekabetçi öğrenme algoritması ile eğitilmektedir. Genellikle bir giriş katmanı ve n boyutlu girişi iki boyutlu olarak eşleyen

nöronların iki boyutlu düzenlemesi olarak tasarlanmış Kohonen katmanı olarak adlandırılan bir çıkış katmanından oluşmaktadır [29, 31, 34].

Avantajları: SOM'lar denetlenmemiş öğrenme kullanılarak eğitilmektedir, yani önceden bir bilgi mevcut değildir ve verinin sınıf üyeliğine ilişkin herhangi bir varsayım yapılmamaktadır, SOM algoritması büyük veri setlerinin işlenmesinde oldukça etkili bir yöntemdir.

Dezavantajları: Küme sayısının belirtilmesi gerekmektedir, kullanıcının küme sınırlarını bulmak için hiyerarşik bir yöntemle manüel olarak inceleme yapması veya geleneksel algoritmalar uygulaması gerekmektedir [29, 31, 34].

Destek vektör makinesi

Destek vektör makineleri (SVM'ler) sınıflandırma ve regresyon için kullanılan bir dizi denetimli öğrenme yöntemidir. Bunlar genelleştirilmiş doğrusal sınıflandırıcılar ailesine aittir. SVM'ler, bir hiper-düzlem kullanılmasına rağmen verileri (temel durumda iki) birden fazla sınıfa ayırmaya çalışmaktadır. Karar sınırına yakın eğitim örneklerindendir. SVM ayrıca ceza faktörü olarak adlandırılan kullanıcı tarafından belirlenen bir parametre de sağlamaktadır. Kullanıcıların yanlış sınıflandırılmış örnek sayısı ile karar sınırının genişliği arasında bir denge kurmasına olanak tanımaktadır [29, 31, 34].

Avantajları: Karmaşık doğrusal olmayan karar sınırlarını modelleyebilen, diğer yöntemlerden daha az uydurma eğiliminde olan, yüksek boyutlu verilerden çok etkili bir şekilde öğrenmektedirler, eğer sınırlı sayıda örnek veri verilirse ve çok sayıda özelliği işleyebilirse izinsiz girişleri doğru şekilde sınıflandırabilmektedir.

Dezavantajları: Hem eğitim hem de test de yavaş hesaplama süresine ve yüksek algoritmik karmaşıklığa sahiptir [29, 31, 34, 36].

Karar ağacı (DT)

Bir karar ağacı, sınıf etiketlerine karşılık gelen niteliklere ve yaprak düğümlerine karşılık gelen dâhili düğümlere sahip köklü bir ağaçtır. İç düğümler, ilişkili niteliğinin aldığı her

değer için bir alt düğüme sahiptir. Öğrenme elemanı, örnekleri uygun sınıflarına en iyi şekilde ayıran özelliği seçerek, seçilen özelliğin her değeri için alt düğümler oluşturarak ve örnekleri seçilen özelliğin değerlerine dayanarak bu alt düğümlere dağıtarak özyinelemeli bir ağaç oluşturmaktadır. Algoritma daha sonra seçili özniteliği kaldırmakta ve her alt düğüm için aynı sınıftan örnekler içeren düğümler üretene kadar tekrar etmektedir. Bu yöntemler, örnekleri kendi sınıflarına en iyi şekilde ayıran bir eşik bularak sürekli özellikleri ele almaktadır. F. Öztürk ve arkadaşları, izinsiz giriş tespit sisteminde karar ağacı yöntemlerinin etkinliğini karşılaştırmışlardır [30].

Avantajları: Ağaç oluşturma işlemi herhangi bir alan bilgisi gerektirmemektedir, Yüksek boyutlu veriler yönetilebilmektedir. Hem sayısal hem de kategorik veriler işlenebilmektedir.

Dezavantajları: Veri yalnızca eksen hizalı olarak ayrılabilmektedir. Muhteris özellikle bir algoritmadır. Katlanarak artan sayıda ağaç çıkarımı söz konusu olabilmektedir [30].

Naive Bayes (NB)

Naive Bayes konsept açıklaması olarak her bir sınıfın önceki olasılığını ve sınıfa verilen her özellik değerinin koşullu olasılığını saklamaktadır. Öğrenme elemanı, bu olasılıkları basitçe oluşma sıklıklarını sayarak örneklerden tahmin etmektedir. Öncelikli olasılık, her sınıftan örneklerin bir kısmıdır. Koşullu olasılık, sınıfa verilen değerleri gösteren sıklıktır. Bir gözlem göz önüne alındığında, performans elemanı, niteliklerin koşullu olarak bağımsız olduğu varsayımıyla çalışmaktadır ve her bir sınıfın arka olasılığını hesaplamak için Bayes kuralını kullanmaktadır ve karar olarak en yüksek olasılıkla sınıf etiketine geri dönmektedir [31].

Avantajları: Sayısal olarak hızlıdır, uygulaması kolaydır, çok katmanlı veri setlerinde iyi performans gösterir.

Dezavantajları: Bağımsızlık varsayımına dayanır ve bu varsayım karşılanmazsa kötü performans gösterir [31].

Rastgele orman

Rastgele ormanlar, daha önce sınıflandırılmamış sınıflandırma veya regresyon ağaçlarının bir birleşimidir. Rastgele orman birçok sınıflandırma ağacı oluşturmaktadır. Her ağaç, bir ağaç sınıflandırma algoritması kullanılarak orijinal verilerden farklı bir önyükleme örneğinden oluşturulmaktadır.

Avantajları: İlişkilendirilebilecek birçok özellik bulunduran veri setlerinde ilintisiz ağaçlar oluşturabilir. Değişimi düşüktür.

Dezavantajları: Görsel olarak yorumlanması güçtür [32, 18, 37].

4.2.3. Hibrit sınıflandırıcılar ve optimizasyon teknikleri

Bir hibrit sınıflandırıcının arkasındaki fikir, sistemin performansını önemli ölçüde iyileştirmek için birkaç tekniği bir araya getirmektir. Öte yandan, hibrit sınıflandırıcılar kademeli olarak farklı sınıflandırıcılara dayanabilir. Optimizasyon teknikleri genellikle Genetik Algoritmalar (GA), Genetik Programlama (GP), Parçacık Sürüsü Optimizasyonu (PSO) ve Karınca Kolonisi Optimizasyonu (ACO) gibi arama ve optimizasyon problemlerini çözmek için kullanılmaktadır.

Avantajları: Kuralları doğru bir şekilde sınıflandırmak için etkili bir yaklaşımdır ve sistem performansı önemli ölçüde iyileştirilebilir.

Dezavantajları: Hesaplama maliyeti yüksektir [29-31].

5. DNS TÜNELLEME

Tünel oluşturma, bir ağ protokolünün bir başka protokolün içerisine aktarılmasıyla olmaktadır. Bu işleme kapsülleme adı da verilmektedir. Örneğin, uzaktaki bir sunucuya bir SSH tüneli oluşturmak ve tüm ağ iletişimini güvenli bir şekilde SSH tüneli üzerinden aktarmak mümkündür. Dışarıdan bakıldığında bağlantı sıradan bir SSH bağlantısı gibi görünecektir, ancak iç kısımda HTTP, FTP, SMTP veya diğer iletişim protokolleri olabilmektedir. SSH gibi bazı protokoller, ağ tünellemesini desteklemek için özel olarak tasarlanmıştır. DNS gibi diğer protokoller ise bunun için tasarlanmamıştır, ancak ağ tünelleri oluşturmak ve içindeki diğer iletişimi gizlemek için kötüye kullanılabilmektedirler.

DNS, internet üzerinde genel amaçlı olarak her sistemde kullanılan bir hizmettir. Bu nedenle kötüye kullanım için uygun bir hedeftir. Burada ele alınan kötüye kullanım ise tünel açma işlemidir. DNS tünellemesinde, DNS üzerinden başka bir protokol tünellenebilir. Bir DNS tüneli komut ve kontrol kanalı açma veya veri sızdırma amaçlı kullanılabilmektedir. RSA konferansındaki 2012 sunumunda Ed Skoudis, DNS tabanlı kötü amaçlı yazılımların komuta ve denetiminin en tehlikeli altı yeni saldırıdan biri olduğunu ifade etmiştir. Ed, “Avcıların bu tekniği yakın zamanda milyonlarca hesabın çalınmasını içeren davalarda kullandıklarını” iletmiştir [42].

DNS tüneli saldırı olarak kabul edilmemektedir, ancak DNS sisteminin kötüye kullanılması olarak bilinmektedir. DNS tünelinde, DNS paketinin içeriği, verilerin DNS yoluyla tünel ağındaki DNS sunucusu olarak davranan bir uygulama sunucusuna gönderilebilmesi için değiştirilmektedir. DNS paketinin değiştirilmiş bölümleri kayıtlardır. Sorgu türü herhangi bir tür olabilmektedir, ancak belirli değerler BASE64 karakter kodlamasının uygulanmasına izin veren TXT kaydı gibi daha dikkat çekicidir, ancak A kaydı yalnızca BASE32 karakter kodlamasına izin vermektedir ve 255 bayttan fazla verilmemektedir.

Bir DNS tüneli aşağıdaki özellikleri barındırmaktadır:

- Büyük miktarda trafik: Bir DNS tünel sunucusuna bağlanan bir istemci, belirli bir etki alanı için sorgular içeren sürekli bir paket akışı oluşturmaktadır. Büyük dosyaları veya

uzak terminal oturumlarını aktarmak için kullanan DNS tünel yazılımı için bu akış devam edecektir.

- Yüksek oranda büyük paketler DNS tüneli üzerinden gönderilmektedir. Dosya aktarımı için DNS tünellerinin kullanılması durumunda, yazılım, paketleri izin verilen maksimum boyuta göre ayarlamaktadır. Dosya yeterince büyükse, verileri göndermek için birden fazla paket gerekmektedir.
- DNS tüneli, çok sık kullanılmayan bazı kayıt türlerini kullanmaktadır. Bunun bir örneği, betimsel ifadeler içerebilen TXT kayıt türüdür.
- BASE64 ve BASE32 karakter kodlamasının kullanılması, yapılan sorguların normal sorgularla karşılaştırıldığında farklı görünmesini sağlamaktadır.

5.1. Tünelleme Temelleri

DNS tüneli oluşturma araçlarının birçoğu ücretli Wi-Fi hizmetleri için kimlik doğrulama adımlarını atlatmak amacıyla oluşturulmuştur. Wi-Fi hizmetinden faydalanılacak sistem kimlik doğrulama adımından bağımsız olarak tüm DNS trafiğine izin veriyorsa, hizmete kimlik doğrulamadan veya ödeme yapmadan IP trafiğini tünellemek için bir DNS tüneli ayarlanabilmektedir. DNS tünelleme yardımcı programlarından bazıları, uç nokta sisteminde yerel olarak bir arabirim oluşturmaktadır. DNS tünelleme aracını barındıran DNS sunucusunda bir tünel uygulaması da olacaktır. Bu, kullanıcının IP trafiğini internete tünellemesini sağlamaktadır. Bu teknik VPN yazılımının OpenVPN gibi çalışmasına benzemektedir. Sunucu tarafı tünelini bir servis olarak sağlayan ticari servis sağlayıcılar da bulunmaktadır. Bu hizmetler DNS üzerinden VPN olarak sunulabilmektedir. Kullanıcı istemci yazılımını yükler ve sunucu tarafında tünel açma yazılımını çalıştıran servis sağlayıcısının DNS sunucusuna bağlanır. Bu hizmetler Android dâhil olmak üzere çeşitli İşletim Sistemleri için istemci yazılımına sahiptir. Bazı durumlarda, Iodine gibi mevcut DNS tünelleme yazılımları kullanılmaktadır.

5.2. Tünelleme Bileşenleri

DNS tünellemesinin bilinen ilk çalışması, 1998 yılının Nisan ayında Bugtraq posta listesinde Oskar Pearson tarafından sunulmuştur [38]. O zamandan itibaren de bir dizi DNS tüneli hizmet programı geliştirilmiştir. Yardımcı programların tümü benzer temel

teknikleri kullanmaktadır, ancak kodlama ve diğer uygulama ayrıntılarında değişkenlik gösterilmektedir. Tüm DNS tünelleme yardımcı programları tarafından kullanılan temel teknikler, kontrollü bir etki alanı veya alt etki alanı, bir sunucu tarafı bileşeni, bir istemci tarafı bileşeni ve DNS verilerinde yüklenen kodları içermektedir. Denetlenen etki alanı, bu etki alanı veya alt etki alanı için yetkili ad sunucusunu tanımlamak için kullanılmaktadır. Örneklendirme amacıyla, test.gudekli.com adresi kullanılacaktır [38].

Sunucu tarafı bileşen ise, bir DNS tüneli sunucusu olarak adlandırılmaktadır.

DNS tüneli sunucusu, kontrol edilen etki alanı için yetkili ad sunucusu olacaktır. DNS tüneli sunucusu tipik olarak tünel kullanıcısı tarafından kontrol edilen internetten erişilebilir bir sunucu olacaktır.

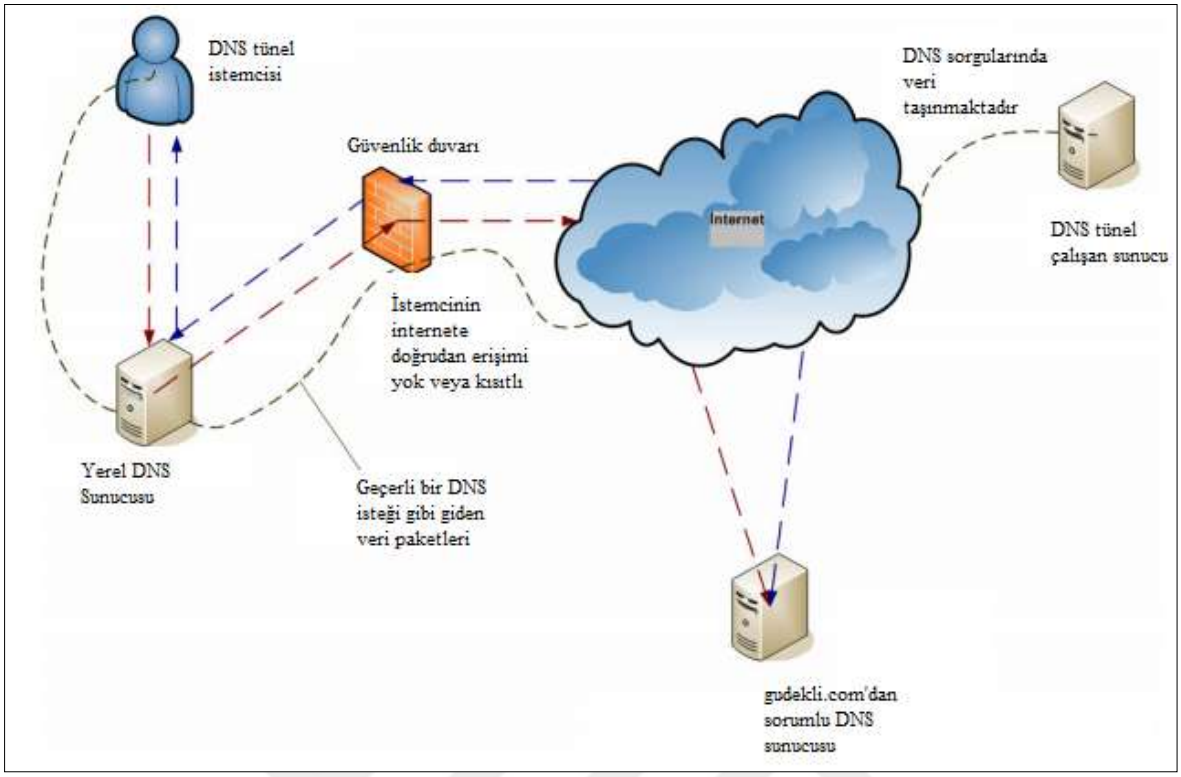
İstemci tarafı bileşeni, tünelin diğer ucunu barındırmaktadır. Bu, güvenlik kontrollü bir işletme ortamında bir son nokta olmaktadır.

Tünel, güvenlik kontrollerini geçerek iletişim kurmak ve kontrol edilen uç nokta ile internetteki herhangi bir ana bilgisayar arasındaki iletişime izin vermek için kullanılabilmektedir [38].

İstemci tarafı bileşeni, DNS'in DNS isteğini başlatmaktadır.

Tünel sunucusu yetkili ad sunucusudur.

DNS tüneli, iki düğüm arasında verileri DNS protokolü paketlerine yerleştirerek paylaşmanın bir yolunu sağlamaktadır. Birçok ağda bu durum güvenlik duvarı politikası ile doğrudan erişim yasaklanarak engellenebilmektedir. DNS tüneli, ağ iletişimini DNS isteklerine yerleştirerek oluşturulur. DNS istekleri, Ad Sunucusu etki alanının alt etki alanlarına yöneliktir ve veriler, genellikle Base32 kodlaması kullanılarak bu alt etki alanlarına kodlanmaktadırlar. Daha önce de belirtildiği gibi, kodlanmış iletişim herhangi bir tür ağ iletişimi olabilmektedir. Örneğin, yalnızca dışarıdan DNS istekleriyle iletişim kuran SSH bağlantıları oluşturmaya izin vermektedir. DNS tünelleme araçları sunucu tarafının istemcilerden gelen bağlantıları dinlerken her zaman bir istemci-sunucu mimarisine sahiptir. Gelen bilgi DNS istekleri ve DNS cevapları içerecek şekilde gelmektedir [38].



Şekil 5.1. DNS tünelleme çalışma yapısı

5.3. Kodlama ve Teknikler

Bilinen temel tekniklerden birisi, DNS yüklerindeki verileri kodlamaktır. Bu, her bir yardımcı programın özelliklerinin geniş ölçüde değiştiği bir alandır. Temelde, istemci sunucuya veri göndermek istemektedir ve bu veriyi DNS yükünde kodlamaktadır. Örneğin, istemci verinin ana bilgisayara kodlandığı bir an 'A' kayıt isteği gönderebilir.

Adı: MRZGS3TLEBWW64TFEBXXMYLMORUW4ZI.test.gudekli.com.

Sunucu, CNAME yanıtı olarak bir yanıt verebilir:

NVWW2IDPOZQWY5DJNZSQ.test.gudekli.com. Bu şekilde herhangi bir veri kodlanabilir ve sunucuya gönderilebilmektedir. Sunucu ayrıca herhangi bir veriyle yanıt verebilir. Sunucunun bir iletişimi başlatması için bir ihtiyaç varsa, doğrudan yapılamaz. Bununla birlikte, sunucu tarafından başlatılan iletişim, istemcinin sunucuyu düzenli olarak sorgulamasını sağlayarak gerçekleştirilebilmektedir. Ardından, sunucu istemci için verilere sahipse, yoklama taleplerine cevap olarak gönderebilmektedir.

DNS tünelleme yardımcı programları uygulama detaylarında farklılık göstermektedir.

C, Java, Perl ve Python DNS tünelleme yardımcı programlarının kodlama dillerinden bazılarıdır. Yardımcı programlardan bazıları ana bilgisayar ve istemci üzerinde yeni bir arayüz ve IP adresi oluşturarak çalışmaktadır. Diğerleri ise Netcat'e benzer şekilde işletim sistemi komutları ile ikili veriyi tünel içerisinde transfer etmektedir.

Yardımcı araçların birbirlerinden farklılık gösterdiği bir diğer nokta DNS kayıt türüne özel şekilde kullanılan kodlama metodudur. Kimileri yaygın olarak kullanılan 'A' kayıt türünü kullanmaktadır. Diğerleri, performansı iyileştirmek EDNS'i ve "Null" kayıtlar gibi deneysel kayıt türlerini kullanmaktadırlar [39].

Base32 veya 5 bit kodlama, istemciden gelen istekler için yaygın olarak kullanılmaktadır. DNS adları büyük ve küçük harf olabilirken, protokol büyük ve küçük harf farkını göz ardı etmektedir. DNS protokolü sayılara ve '-' karakterlerinin kullanımına da izin vermektedir. Bu durumda, 26 harf ile toplam 37 adet benzersiz karakter kullanma imkânı sağlanmaktadır. 5 bitlik bir veride 32 değer alınabilmektedir. Bu 32 değer 37 uygun karaktere sığabilmektedir. Daha sonra kodlanmış verilerden bir dizi iç içe alt alan adı oluşturulmaktadır. DNS, her bir alt alan adı 63 veya daha az karakterli olmak üzere toplamda 255 karaktere izin vermektedir.

Base64 veya 6 bit kodlama, TXT kayıt yanıtları için kullanılabilmektedir. Bir TXT kaydında 52 karakter sağlayan büyük ve küçük harf olmaktadır. Rakamlar 10 karakter olmak üzere ve ek olarak '-' ve ' +' gibi iki ek karakter eklenirse, standart 64 kodlaması için kullanılabilecek 64 benzersiz değer elde edilmektedir. Base32 kodlanmış isteğine benzer şekilde, yanıt, bir TXT yanıtı kullanılarak bir seferde 6 bit olarak kodlanabilmekte ve istemciye geri gönderilmektedir.

Hex kodlama, başka bir kodlama yöntemidir. Onaltılık kodlama için, iki karakterli onaltılık değerler her baytı temsil etmek için kullanılmaktadır. Bu yöntem yalnızca DNSCat-B tarafından kullanılan bir yöntemdir.

DNS tünelleme yardımcı programları farklı DNS kayıt türlerinden ve kodlama yöntemlerinden yararlanmaktadır. Iodine gibi bazı araçlarda, yardımcı program mümkün olan en iyi kodlamayı otomatik olarak algılamaktadır. DNS tünelleme programları, 512 bayttan daha büyük yükleri kullanmalarını ve böylece performansı artırmalarını sağlayan EDNS'den de bu amaçla

yararlanabilmektedirler. Bir DNS tünel oluşturma programı olan Heyoka, istemcinin görünürlüğünü azaltmak için sunucuya yapılan taleplerde kaynak IP adreslerini taklit etmektedir.

5.4. DNS Tünelleme Yardımcı Programları

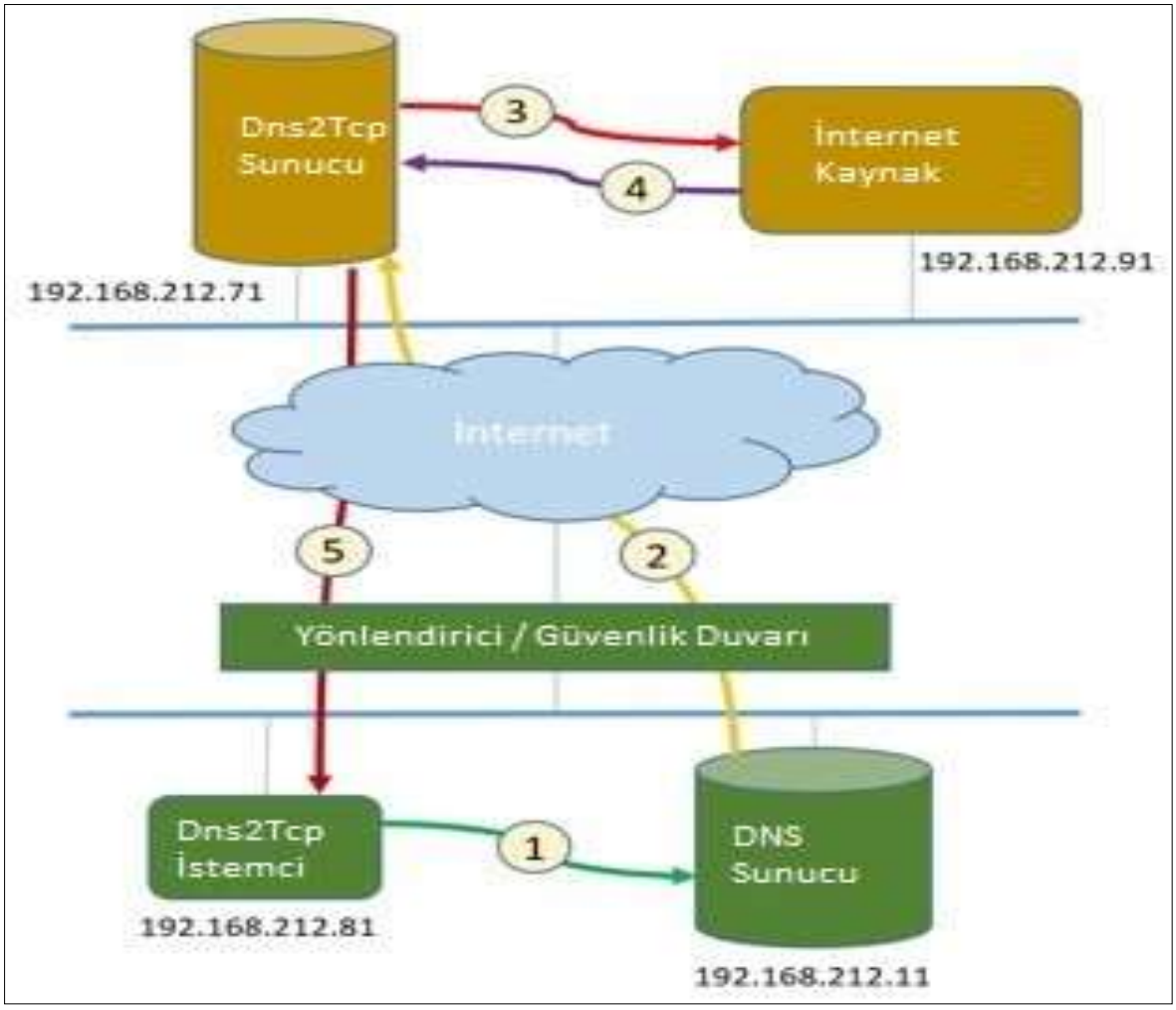
Günümüzde birçok DNS Tünelleme aracı bulunmaktadır. Bu bölümde sık kullanılan birkaç tanesi hakkında özet bilgi sunulmaktadır.

5.4.1. Dns2tcp

Dns2tcp aracı, TCP trafiğini DNS trafiği kullanarak tünellemeyi sağlayan bir uygulamadır. Temelde Dns2tcp aracı iki ana bileşene sahiptir. Birincisi, genellikle uzak bir sunucuda çalışan Dns2tcpd, ikincisi ise istemci olarak çalışan Dns2tcpc'dir.

Sunucudaki bir yapılandırma dosyası, sahip olduğu kaynakların bir listesini içermektedir. Her kaynak, TCP bağlantılarını dinleyen yerel veya uzak bir servis özelliği taşımaktadır. İstemci uygulaması önceden tanımlanmış bir TCP bağlantısını dinlemektedir ve gelen bağlantı isteklerini uç nokta servisine iletmektedir.

Aşağıda Dns2tcp çalışma mantığını gösteren bir şekil bulunmaktadır.



Şekil 5.2. DNS2TCP çalışma mantığı

5.4.2. DNSCat

Dnscat, iki sunucunun internette birbirleriyle iletişim kurmasına izin verecek şekilde tasarlanmıştır. Tüm trafiği yerel bir DNS sunucusu üzerinden yönlendirmektedir. Bu yapının önemli avantajları bulunmaktadır:

- Tüm ağ güvenlik duvarları geçilebilmektedir.
- Birçok yerel güvenlik duvarı geçilebilmektedir.
- Gizlidir, çünkü bilinen bir ağ geçidi / proxy'den geçmemektedir.

Bu avantajlara ek olarak, bazı dezavantajlara da sahiptir. Örneğin; DNS büyük / küçük harf duyarlı olmadığından, veriler alfa nümerik olarak kodlanmalıdır. Bu, herhangi bir sıkıştırma işlemi olmadığı için boyutun iki katına çıkarıldığı anlamına gelmektedir.

5.4.3. Iodine

Iodine aracı, IPv4 verilerinin bir DNS sunucusu aracılığıyla tünellenmesini sağlamaktadır. Bu, İnternet erişiminin güvenlik duvarına sahip olduğu ancak DNS sorgularına da izin verildiği durumlarda kullanışlı olmaktadır. Diğer DNS tünelleme araçlarına kıyasla, Iodine bazı avantajlar sunmaktadır:

- İndirme verilerinin kodlama olmadan iletilmesine izin veren NULL veri türünü kullanır.
- Unix sistemlerinde ve Windows32 platformlarında çalışabilmektedir.
- MD5 şifreleme yöntemi ile şifrelenmiş bir zorluk-tepki giriş yöntemini kullanmaktadır. Ek olarak, giriş IP'si dışındaki tüm paketleri diğer IP'lerden de filtrelemektedir.

5.4.4. OzymanDNS

OzymanDNS, dört temel Perl betiği içeren bir araçtır. Bunlardan ikisi, son kullanıcının DNS kullanarak dosya yüklemesine ve indirmesine izin vermektedir. Diğer iki komut dosyası bir sunucu-istemci yapısı rolündedir. Sunucu bir DNS sunucusu işletim yapısını taklit eder ve gelen talepler için port 53'ü dinlemektedir. Sonrasında, İstemci bilgisayar gelen verileri belirli bir alana gönderilen DNS isteklerine dönüştürmektedir. Bu iki komut SSH ile bir tünel oluşturmak için kullanılabilmektedir. Son kullanıcının tünelden trafiği geçirmek için portları elle eşleştirmesi gerekmektedir.

5.5. DNS Tünelleme Tespit Etme Yöntemleri

DNS tünelleme yapan uygulamaların çoğu gizli kalmaya çalışmamaktadır. DNS, sıklıkla izlenmemesine güvenilen bir protokol olarak bilinmektedirler. Bu kapsamda çeşitli DNS tüneli algılama teknikleri önerilmiştir. Tespit teknikleri yük analizi ve trafik analizi olmak üzere iki ayrı kategoride ele alınmıştır. Yük analizi için bir veya daha fazla istek ve yanıtın DNS yükü, tünel göstergeleri için analiz edilecektir. Trafik analizi için DNS trafiği zaman içinde analiz edilmektedir. Sayı, sıklık ve diğer istek özellikleri dikkate alınacaktır.

Bu bölümde tespit etme yöntemlerinde kullanılan bazı kriterler hakkında özet bilgi sunulacaktır.

5.5.1. İstek ve yanıtın boyutu

Kullanılan yöntemlerinden biri, istek ve yanıtın boyutunu analiz etmeyi içermektedir. Bir blog yazısında [40], yazar, kaynak ve hedef bayt oranına bağlı olarak şüpheli DNS tüneli trafiğini tanımlamak için yöntemler tanımlamıştır [40]. Oran daha sonra bir limit değerle karşılaştırılmaktadır. Pietraszek, (2004) ve Skoudis, (2012), DNS tünellemeyi tespit etmek için DNS sorgusunun uzunluğuna ve verilen yanıtlara bakılmasını önermişlerdir [40-42]. DNS tüneli yardımcı programları genellikle istek ve yanıtlara mümkün olduğunca fazla veri girmeye çalışmaktadırlar. Bu nedenle, tünel isteklerinin uzun etiketlere, 63 karaktere ve uzun genel isimlere, 255 karaktere sahip olması muhtemeldir. Guy isimli araştırmacı tarafından sunulan diğer bir öneri ise 52 karakterden uzun olan tüm ana bilgisayar adı isteklerine bakılmasıdır [43].

5.5.2. Ana bilgisayar adlarının entropisi

DNS tünelleri, sorgulanan ana bilgisayar adlarının entropisine dayanarak tespit edilebilmektedir [44, 45]. Meşru DNS isimleri genellikle sözlük kelimelerine veya anlamlı görünen ifadelere sahiptir [44]. Kodlanmış isimler daha yüksek bir entropiye ve ayarlanan karakterlerin daha da eşit kullanılmasına sahiptir. Bununla birlikte, DNS adlarının bir tür bilgiyi temsil etmek için kullanıldığı yerlerde istisnalar bulunmaktadır. Bu bazen içerik dağıtım ağlarında da geçerlidir. Entropisi yüksek olan DNS adlarını sorgulamak, tünel açmanın bir göstergesi olabilmektedir.

5.5.3. İstatistiksel analiz

DNS adlarının özel karakter yapısına bakmak, tüneli tespit etmek için kullanılabilecek başka bir yöntemdir. Meşru DNS isimleri birkaç numaraya sahipken, kodlanmış isimler çok sayıda numaraya sahip olabilmektedir. Alan adlarındaki sayısal karakterlerin yüzdesine bakılması önerilmiştir [46]. En uzun anlamlı alt kelime (LMS) uzunluğunun yüzdesine bakmak da başka bir yöntemdir [46]. Benzersiz karakterlerin sayısına bakmak ise başka bir olasılıktır. Ayrıca 27'den fazla benzersiz karakter içeren herhangi bir şey hakkında uyarıda bulunma önerisi verilmiştir [43].

Meşru alan adlarının ortak dilleri bir dereceye kadar yansıttığı göz önüne alındığında, karakter tüneli analizinin kullanımı, DNS tünellemesinden üretilen adları tespit etmek için de kullanılabilmektedir [47-49].

DNS tünellemesini tespit etmek için tekrarlanan ünsüzleri ve sayıların adedine de bakılabilmektedir [50]. Tünel oluşturma yardımcı programları, ortak dilleri yansıtan alan adlarında görülmeyecek ardışık ünsüz harfleri ve sayıları olan potansiyel adlar oluşturmaktadır.

5.5.4. IP adresi başına DNS trafiği hacmi

Temel ve yalın yöntemlerden biri, belirli bir istemci IP adresinin ürettiği DNS trafiğinin miktarına bakmaktır [41, 44].

Tünellenmiş veriler tipik olarak istek başına 512 bayt ile sınırlı olduğundan, iletişim için çok sayıda istek gerekmektedir. Ayrıca, istemci sunucu ile bağlantısını kontrol ediyorsa, sürekli olarak istek göndermektedir [48].

5.5.5. Etki alanı başına DNS trafiği hacmi

Diğer bir temel yöntem ise, belirli bir alan adına doğru olan trafik yoğunluğuna bakmaktır [45]. DNS tüneli hizmet programlarının tümü, verileri belirli bir etki alanı adı kullanarak tünellemek için ayarlanmıştır, bu nedenle tüm tünel trafiği bu etki alanı adına olacaktır. DNS tünellemesinin birden fazla etki alanı adıyla yapılandırılmış olabileceği düşünülerek, etki alanı başına düşen trafik miktarı incelenmelidir [48, 51].

5.5.6. Etki alanı başına ana bilgisayar adı sayısı

Belirli bir etki alanı için ana bilgisayar adı bir gösterge olabilmektedir [43]. DNS tüneli yardımcı programları, her istek için benzersiz bir ana bilgisayar adı istemektedir. Bu, tipik bir meşru alan adından çok daha büyük bir sayıya yol açmaktadır. Bu yöntem trafik analizine dayalı örnek bir uygulama olarak kullanılacaktır.

5.6. DNS Tünelleme Tespitinde Makine Öğrenmesi Yöntemleri

Son zamanlarda, DNS tünelleme sorununa bir cevap olarak, bazı araştırmacılar tünelleri tespit etmek için makine öğrenim tekniklerini kullanma eğiliminde olmuştur. Denetimli makine öğrenme teknikleri, makineye belirli bir deneyimden öğrenme yeteneği kazandırmayı amaçlamaktadır. Makine öğrenimi araştırmalarında yapılan ilk çalışmalar, temel olarak, belirli bir koşulun ortaya çıkmasının makineye bir şeyler yapmayı öğretebileceği önceden belirlenmiş kurallara bağlı olarak belirtilmiştir. Bununla birlikte, kural temelli yaklaşımın ardındaki problem, kuralları oluşturmanın zaman alan bir işlem olmasıdır [34, 35].

Bu şekilde, makine öğrenme teknikleri (MLT), kuralların istatistiksel bir modele dayalı olarak otomatik olarak üretilebileceği bir problemin çözümü olarak ortaya çıkmıştır. MLT'nin ardındaki temel özellik, çıkarım için sağlanması gereken tarihsel verilerde yatmaktadır. Makine eğitilmekte ve istatistiksel model oluşturulmaktadır. Bazı araştırmacılar, makine öğrenmenin kullanımını DNS tünellerini tespit etme açısından incelemiştir. Bununla birlikte, bu çalışmalar, DNS tünelleme problemini, sınıf etiketinin meşru veya tünel olduğu ikili bir sınıflandırma olarak ele almıştır. Akış verilerinin genel davranışının tespit edilmesi ve davranış analizi kullanılarak trafiğin doğru bir şekilde sınıflandırılması durumunda, daha az miktarda bilinmeyen trafik akışının görülebildiği bilinmektedir. Bu nedenle, yalnızca DNS tünelini algılamak değil, aynı zamanda bu tüneli sınıflandırmak için de önemli bir gereksinim bulunmaktadır [29, 30].

Makine öğrenme teknikleri yeteneklerinin avantajlarından yararlanarak, bu araştırmada, DNS tünelleme tiplerini sınıflandırma problemini ele almak için çoklu etiketli veya kernel destek vektör makinesi sınıflandırıcısının kullanımı düşünülmüştür.

6. ÖNERİLEN ÇÖZÜM YÖNTEMİ

Makine öğrenmesi tekniği, bir ağ üzerinde gerçekleşen işlemlerde normal kabul edilecek işlemler ile daha önce gerçekleşmemiş, anormal kabul edilecek işlemlerin ayrımında kullanılabilecek etkili bir tekniktir. DNS tünelleme tespitinde makine öğrenmesi teknikleri kullanılarak, tünelleme bulunmayan normal ağ trafiği üzerinde öğrenme gerçekleştirilerek, tünelleme gerçekleştiğinde ağ üzerindeki, öğrenilen normal duruma aykırı durumlar tespit edilebilir.

Önerdiğimiz çalışmada DNS tünelleme işleminin DNS paketine ve DNS trafiğine getirdiği etkiler ele alınarak, temiz DNS trafiği ile tünel içeren DNS trafiğinin birbirinden ayrılabileceği tezi sunulmuştur.

Tezin dayanağı olan bu kabulü desteklemek için daha önce yapılan DNS tünelleme çalışmalarında ve araştırmalarında DNS tünellemenin DNS trafiğine ve paketine etkileri detaylı olarak incelenmiştir.

I. Homem çalışmasında Entropi sınıflandırmasına dayalı bir DNS tünelleme tespit sistemi önermiştir [52].

Buczak DNS tünelleme tespiti için rastgele-orman sınıflandırma metodunu önermiştir. Bu çalışmada sınıflandırma için DNS yanıt paketindeki cevap sayısı, bir etki alanı için yapılan iki sorgu arasındaki ve iki cevap arasındaki süre gibi değişik özellikler incelenmiştir [18, 37].

Benzer şekilde Aiello DNS tünellemeyi tespit etmek için iki paket arasında geçen süre ve paket boyutu gibi özellikleri adresleyen bir makine öğrenmesi önermiştir [28, 34].

Yakov Bubnov da çalışmasında önerdiği paket analiz yönteminde DNS paketlerinin ortalama boyutunu ve sorgulanan etki alanı adının entropisini (karmaşıklığını) incelemiştir [14].

Bu örneklerden de anlaşılabileceği üzere DNS tünelleme tespitine yönelik çalışmalar paket analizi ve trafik analizi olmak üzere iki ana kategoride toplanmaktadır. Farnhman ve Atlasis de DNS tünellemeyi tespit etmeye yönelik yaklaşımları inceledikleri

çalışmalarında, DNS tünelleme tespiti çalışmalarının DNS paket analizi ve DNS trafik analizi olmak üzere iki kategoriye ayrıldığını belirtmişlerdir [14]. Paket analizi üzerine yapılan araştırmalar sorgu ve cevap paketlerinin incelenmesine dayanırken, trafik analizi üzerine yapılan araştırmalar bir istemciden çıkan veya belli bir ana bilgisayara doğru oluşan DNS trafiğinin yoğunluğuna eğilmektedir.

Paket analizi teknikleri arasında;

- Sorgu ve cevap paketine ait boyut analizi (Krmicek)
- Domain ismi entropi analizi (Batler, Xu, Yao)
- Yapılan sorgu türü
- Etki alanı isminde belli karakterlerin tekrar etme sayısı gibi maddeler incelenmiştir [46].

Trafik analizi teknikleri arasında;

- Tek bir istemciden oluşturulan DNS trafiğinin boyutu
- Belirli etki alanı isimlerine doğru oluşturulmuş DNS trafiğinin boyutu
- DNS sunucuların coğrafik konumları gibi özellikler dikkate alınmaktadır.

Brownlee N., Villamarin-Salomon, Brustoloni Wang ve Tseng'in çalışmalarında belirttiği gibi DNS tünelleme üzerine olan çalışmaların yoğunlaştığı bir diğer yöntem ise istatistiksel analizdir. Ancak istatistiksel algoritmalar önceden belirlenmiş sınır değerlerine oldukça bağlıdır [53].

Maurizio Aiello, Maurizio Mongelli ve Gianluca Papaleo çalışmalarında gözetimli öğrenme tekniklerine dayanan sınıflandırma algoritmalarının DNS tünelleme tespitinde kullanılabileceğini göstermişlerdir. Bu yaklaşım, geleneksel istatistiksel algoritmalardaki önceden tanımlı sınır değerine dayanan problemi ortadan kaldırmakta ve daha iyi sonuç vermektedir [28, 34].

DNS tünelleme tespitine yönelik güncel çalışmaların bir diğerinde çalışmada Ahmed Almusawi ve Haleh Amintoosi gözetimli öğrenme tekniklerinden biri olan Destek Vektör Makineleri sınıflandırma algoritması ile çeşitli veri setleri üzerinde DNS tünelleme tespiti

yapılabildiğini ve elde edilen sonuçların Bayes sınıflandırma algoritmasına göre daha iyi sonuçlar ürettiğini göstermişlerdir [54].

DNS paket boyutları üzerinde yaptığımız çalışmada DNS tünellemenin yalnızca DNS paket boyutu incelenerek tespit edilip edilemeyeceği incelenmiştir [6].

Bu örneklerden de elde edilen bilgiler ile DNS trafiğinde DNS paketlerinin özellikleri incelenerek ve normal bir DNS trafiği ile karşılaştırılarak DNS tünellemenin tespit edilebileceği öngörülmüştür.

Bu bağlamda önerdiğimiz çalışmada, daha önce yapılan çalışmalarda kontrol edilen parametrelerden en etkili olanlar çalışma sonuçları incelenerek tespit edilmiş ve bu parametrelerin tümü ile bunlara ek olarak incelemiş olduğumuz DNS paket boyutu parametresi ile domain ismine yönelik itibar (repütasyon) analizi sonucu bir arada ve aynı anda dikkate alınarak bir çözüm sunulmuştur.

Önerilen çalışmada DNS paketi üzerinde;

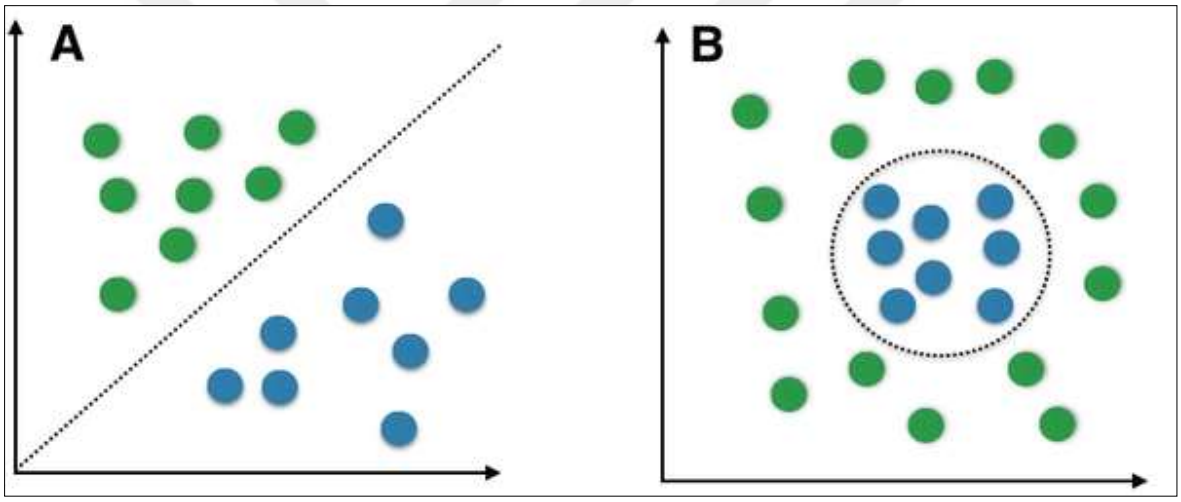
- Etki alanı adı entropisi (karmaşıklığı)
- Sorgu yapılan alt domain sayısı
- Etki alanı ad uzunluğu
- Etki alanı adının ne zaman oluşturulduğu
- Etki alanı IP'sinin daha önce zararlı olarak işaretlenip işaretlemediği
- DNS paket boyutu

özellikleri dikkate alınarak Kernel- SVM sınıflandırma algoritması kullanılarak temiz DNS paketleri, tünelleme yapılmış DNS paketlerinden ayrıştırılmaya çalışılmıştır.

SVM algoritması veri içerisinde farklı sınıflara ait grupları tespit etmede kullanılan bir sınıflandırma algoritmasıdır. SVM algoritması içerisinde sunulan her veri bir destek vektörü olarak ele alınır. Tek boyutlu, iki boyutlu, 3 boyutlu ve hatta 3'den fazla boyutlu düzlemlerde bu destek vektörlerinden birbirine yakın noktalarda bulunanları diğer birbirine yakın olarak bulunan gruplardan ayırmak için karar sınır çizgisi kullanır. 2 boyutlu düzlem için kullanılan bu karar sınır çizgisi 3 boyutlu düzlemlerde karar sınır düzlemi olarak ifade edilmektedir. SVM 3'den fazla boyutlarda da başarılı şekilde çalışsa da insan beyni ancak 3

boyutlu düzlemi hayal edebilmektedir. 4 ve daha fazla boyutlu veri setleri için de ayrı bir tanımlama olmasa da karar sınır çizgisi ifadesi kullanılabilir. Mümkün olduğunca iki grup verinin düz bir çizgi ile birbirinden ayrılabilmesi hedeflenmektedir. Ancak bu her zaman mümkün olmamaktadır. SVM, verinin ikiden fazla özelliğinin bulunması durumunda daha efektif kullanılabilen bir sınıflandırma algoritmasıdır. Uygulaması kolaydır, genellikle iyi genelleştirme performansı göstermektedir, küçük bir bakımla aynı algoritma birçok problemi çözebilmektedir, çok sayıda bağımsız değişkenle çalışabilmektedir ve karmaşık karar sınırlarını modelleyebilmektedir.

Aşağıda düz bir çizgi ile ayrılabilen iki grup ve düz bir çizgi yardımıyla ayrılamayacak 2 grup veri setinin vektörsel gösterimi verilmiştir:



Şekil 6.1. Lineer ayrıştırılabilen ve lineer ayrıştırılamayan iki veri seti

Bu çalışmada DNS tünellemesinin tespiti için “Kernel Support Vector Machine” makine öğrenme yöntemi denenmiştir.

6.1. Verinin Hazırlanması

Makine öğrenimini kullanarak önerilen DNS tünel tespitini test etmek ve doğrulamak için hem iyi hem de kötü amaçlı DNS trafik verilerine ihtiyacımız bulunmaktadır.

DNS tünellemede kullanılan araçlar birbirlerinden farklı davrandığından, DNS tünelleme veri seti oluşturmak için çeşitli araçlar kullanılmıştır. Dns2tcp aracı DNS tünelleme için TXT kayıtları kullanırken Iodine DNS tünelleme aracı NULL kayıtlarını işlemektedir.

Bu çalışmada tünelli DNS trafiği oluşturmak için de Iodine, Dns2tcp yardımcı araçları kullanılarak satın alınan ve çalışma için daha önce şeması açıklandığı üzere DNS tünellemeye uygun şekilde düzenlenen etki alanına DNS tünelleme işlemi gerçekleştirilmiş ve DNS trafiği Wireshark uygulaması aracılığıyla pcap dosyasına kaydedilmiştir.

Verinin hazırlanması için DNS tünelleme araçlarının çalıştırılacağı istemci olarak 4 çekirdekli işlemcili 8 GB hafızalı ve 120 GB saklama kapasiteli bir Windows 10 işletim sistemine sahip işlemci kullanılmıştır.

Tünelli trafik elde edebilmek amacıyla, Dns2Tcp aracılığıyla gerçekleştirilen tünelleme işlemi için aşağıdaki adımlar izlenmiştir:

- Öncelikle, Ubuntu işletim sistemine dns2tcp paketi aşağıdaki kod parçacığı ile kurulmuştur.

– apt-get install dns2tcp

- Sonrasında DNS2TCP sunucusu seçenekleri yapılandırılmıştır. Listelenen IP adresi içi yerel ağ ara yüzü 85.98.12.57 ve port numarası DNS portu tcp / 53’ tür.
- DNS tüneli için kullanılacak etki alanı: dns2tcp.gudekli.com’ dur.
- DNS2TCP sunucusunun IP adresi 85.98.12.57 olduğundan, bu arabirime listeleme IP adresinin port numarasının 53 olması sağlanmıştır.

Bu çalışmada sadece DNS tünellemesi üzerinden SSH oturumuna odaklanılmıştır.

```
root@kali-server:~#cat >>.dns2tcpdrc <<END
listen = 0.0.0.0
port = 53
user=nobody
chroot = /root/dns2tcp
pid_file = /var/run/dns2tcp.pid
doman = dns2tcp.gudekli.com
key = secretkey
resource = ssh:127.0.0.1:22
END
root@kali-server:~# dns2tcp -f .dns2tcpdrc
root@kali-server:~#
```

Resim 6.1. DNS2TCP sunucu kurulumu

- Sonrasında, aşağıdaki komutu çalıştırarak DNS tüneli arka plan uygulamasını başlattık:

– dns2tcpd –F –d 3 –f dns2tcpdrc

- Ardından DNS2TCPClient işlevi için DNS2TCP sunucusu gibi, önce Ubuntu işletim sistemine dns2tcp paketinin kurulması gerekmektedir. İkinci olarak, DNS2TCP istemcisi Resim 6.2'de gösterildiği şekilde yapılandırılmıştır.

```
root@ubuntu:~# more settings.tunnelclient.txt
domain = tunnel.gudekli.com
resource = ssh
local_port = 22
debug_level = 3
key = secretkey
server = dns2tcpdrc
root@ubuntu:~#
```

Resim 6.2. DNS2TCP istemci kurulumu

- Son olarak, istemcide DNS2TCP uygulaması proxy olarak çalıştırılmıştır:

– dns2tcpc –d 3 –f settings.tunnelclient.txt

- Bu işlemlerin ardından DNS tüneli kullanılarak 85.98.12.57 adresli SSH sunucusuna SSH oturumu başlatılabilmiştir.

– ssh jxia@127.0.0.1 – s 2222 – D 8081

DNS tünel kanalı üzerinden bu SSH sunucusuna bağlanılabilmelidir.

Iodine aracı ile gerçekleştirdiğimiz tünelli trafik oluşturma işleminde ise Resim 6.3'te yer alan adımlar gerçekleştirilmiştir.

```
root@PC1-VirtualBox:~# sudo ./iodine -I 50 -f -P Ankara06 t1.ugurgudekli.com
sudo: ./iodine: command not found
root@PC1-VirtualBox:~# iodine
iodine               iodine-client-start  iodined
root@PC1-VirtualBox:~# iodined -c -f 10.0.0.1 -P Ankara06 t1.ugurgudekli.com
Opened dns0
Setting IP of dns0 to 10.0.0.1
Setting MTU of dns0 to 1130
Opened IPv4 UDP socket
Listening to dns for domain t1.ugurgudekli.com
```

Resim 6.3. Iodine aracı ile tünelli trafik oluşturma

Bu yöntemlerle elde ettiğimiz tünelli trafik verisinin yanı sıra açık kaynak olarak ulaşılabilen Mendeley veri seti de testlerimizde kullanılmıştır. Bu veri seti Nisan-Mayıs 2016 tarihleri arasındaki rastgele 10 gün boyunca en yoğun yük saatlerinde 4000'den fazla etkin kullanıcıdan oluşan kampüs DNS ağ trafiği, saatlik PCAP dosyalarından oluşmaktadır.

Çalışma kapsamında ayrıca tünelli DNS trafiği içermeyen veri elde etme amacıyla, Kaliforniya San Diego Üniversitesi'nde (UCSD) CAIDA (Uygulamalı İnternet Veri Analizi Merkezi) tarafından araştırma amacıyla toplanan ve sağlanan bir veri kümesi de kullanılmıştır. Bu veriler CAIDA Kabul Edilebilir Kullanım Sözleşmesine uygun olarak kullanılmaktadır [55].

Wireshark ile temiz DNS trafiği oluşturmak için kullanılan filtre aşağıdadır:

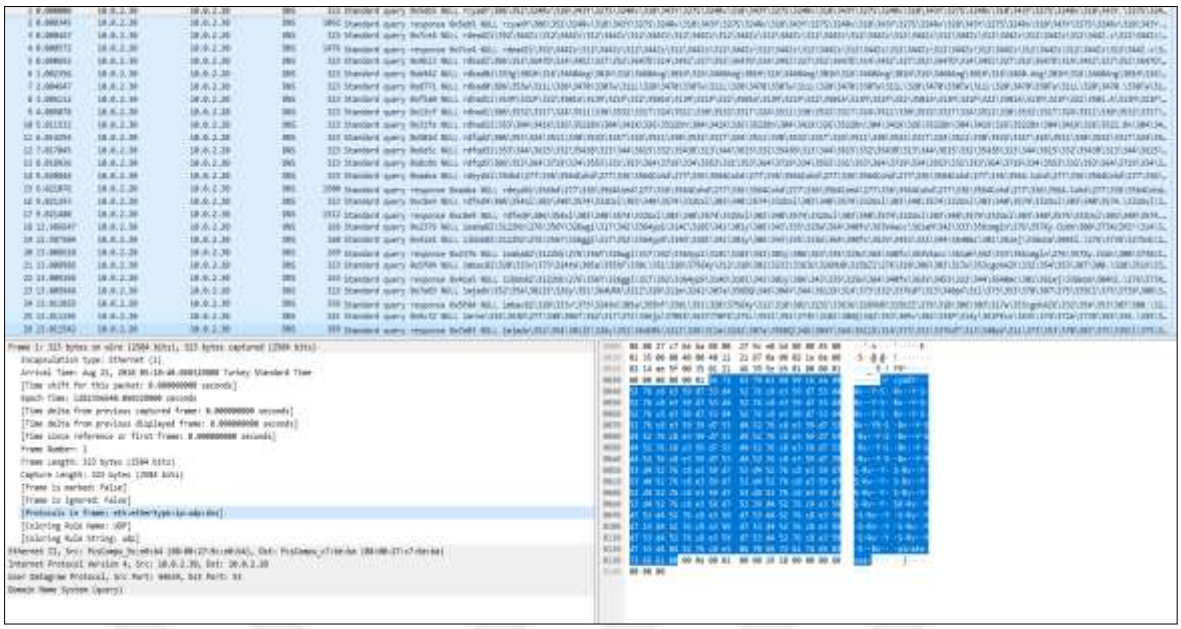
```
port 53
```

DNS tünelleme araçları kullanılmadan kaydedilen pcap dosyası temiz DNS trafiği olarak kabul edilmiştir.

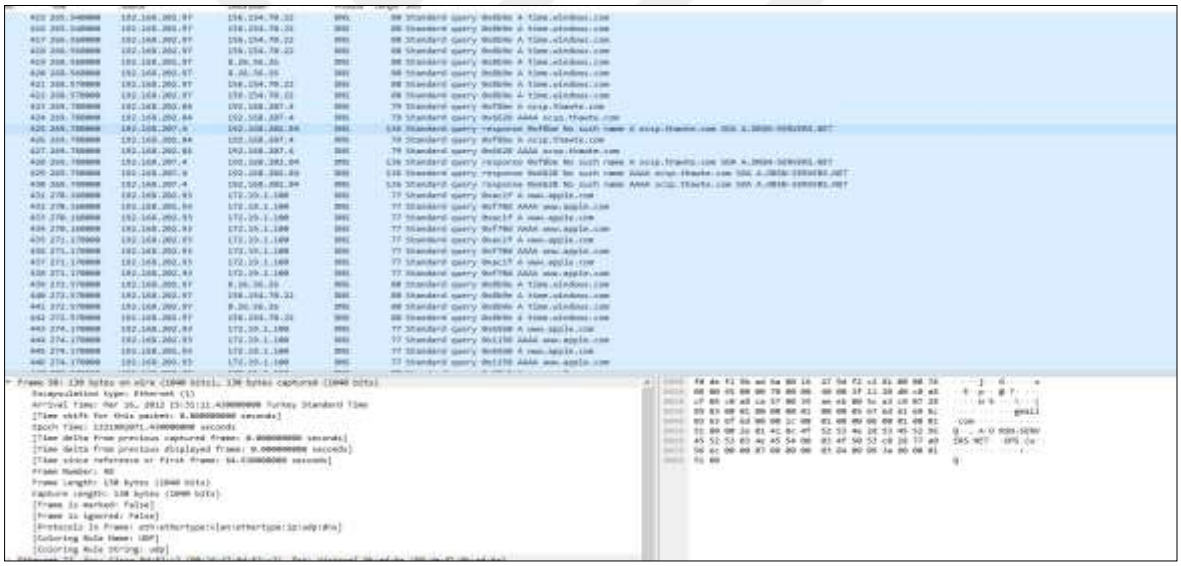
DNS tünelleme araçları kullanılarak kaydedilen pcap dosyası ise tünelleme içeren pcap olarak kabul edilmiştir.

Tüm DNS trafiği daha sonra Wireshark aracı tarafından toplanmıştır ve saklanmıştır.

Resim 6.4’te iyi trafiğe sahip PCAP dosyası, Resim 6.5’te kötü trafiğe sahip PCAP dosyasına ait görüntüler bulunmaktadır.



Resim 6.4. Temiz DNS trafiği



Resim 6.5. Tünelli DNS trafiği

Bu iki pcap dosyası kullanılarak kernel SVM sınıflandırma algoritması ile belirlenen niteliklerin temiz pcap dosyası ile tünelleme yapılmış pcap dosyası arasındaki ayrım elde edilmeye çalışılmıştır.

SVM lineer ayrım hesabı ve her bir parametre için gerekli ölçümlerin tamamı Python programlama dili kullanılarak geliştirilmiştir.

DNS tünelleme tespiti için önerilen makine öğrenme yöntemini test etmek ve doğrulamak için, SciTit-Learn, Python için makine öğrenmesi sınıflandırıcıları oluşturma fonksiyonlarını içeren ve eğitim ve test için destek fonksiyonlar içeren bir kütüphanedir. SciKit-Learn'deki metrik sınıfı, sınıflandırıcıları değerlendirmek için birçok işlev içermektedir. Bir uç-değer ve sınıflandırma sınıflandırması için başarının belirlenmesinin normal yolu, kesinliği, hatırlamayı ve F-skorunu ölçmektir. Bir sınıflandırıcının doğruluğu, seçilen öğelerin gerçek pozitif olan yüzdesini belirlemektedir. Hatırlama ise, yüzdesini belirlemektedir.

Scikit-Learn modülü verileri standartlaştırmanın ve tüm dönüşümleri daha sonra kullanmak üzere farklı veri setlerinde kaydetmenin kolay bir yolunu sunmaktadır. Bu, makine öğrenimi eğitimi için mi yoksa performans testi amacıyla mı yapıldığına bakılmaksızın tüm verilerin aynı şekilde standartlaştırıldığı anlamına gelmektedir.

İlgili elemanlar seçildikten sonra F-skoru ölçümü hem hassasiyetten hem de hatırlamadan elde edilmektedir ve sınıflandırıcının genel karakterini daha iyi temsil eden bir sonuç vermektedir. Sonuç 1'e ne kadar yakınsa, hassasiyet ve hatırlama da o kadar yüksek olmaktadır ve sınıflandırıcı iyi sonuç veriyor anlamına gelmektedir.

6.2. Geliştirilen Uygulama

Uygulama 4 çekirdekli, 16 GB hafızalı ve 250 GB saklama kapasiteli bir Windows 10 işletim sistemine sahip istemci üzerinde sanal olarak çalışan 2 çekirdekli işlemci, 12 GB hafızalı ve 60 GB saklama kapasiteli Ubuntu işletim sistemine sahip bir istemci üzerinde Python 2.7 kodlama dili kullanılarak geliştirilmiştir.

Uygulama başlangıç olarak üç parametre (temiz trafik, tünelli trafik, test edilecek trafik) veya tek parametre (test edilecek trafik) alabilmektedir. Bu parametrelerin tipi PCAP dosya tipidir. Üç parametre ile uygulama başlatıldığında modelin hassasiyeti için ekstra kontroller yapılması için model yeniden eğitilmektedir. Temiz trafiğe sahip PCAP dosyası, tünelli olduğu bilinen trafiğe sahip PCAP dosyası ve test edilecek olan dosyası uygulamaya giriş parametresi olarak verilmektedir.

```
Parameters:  pcap/temiz.pcap pcap/tunelli.pcap pcap/test.pcap
```

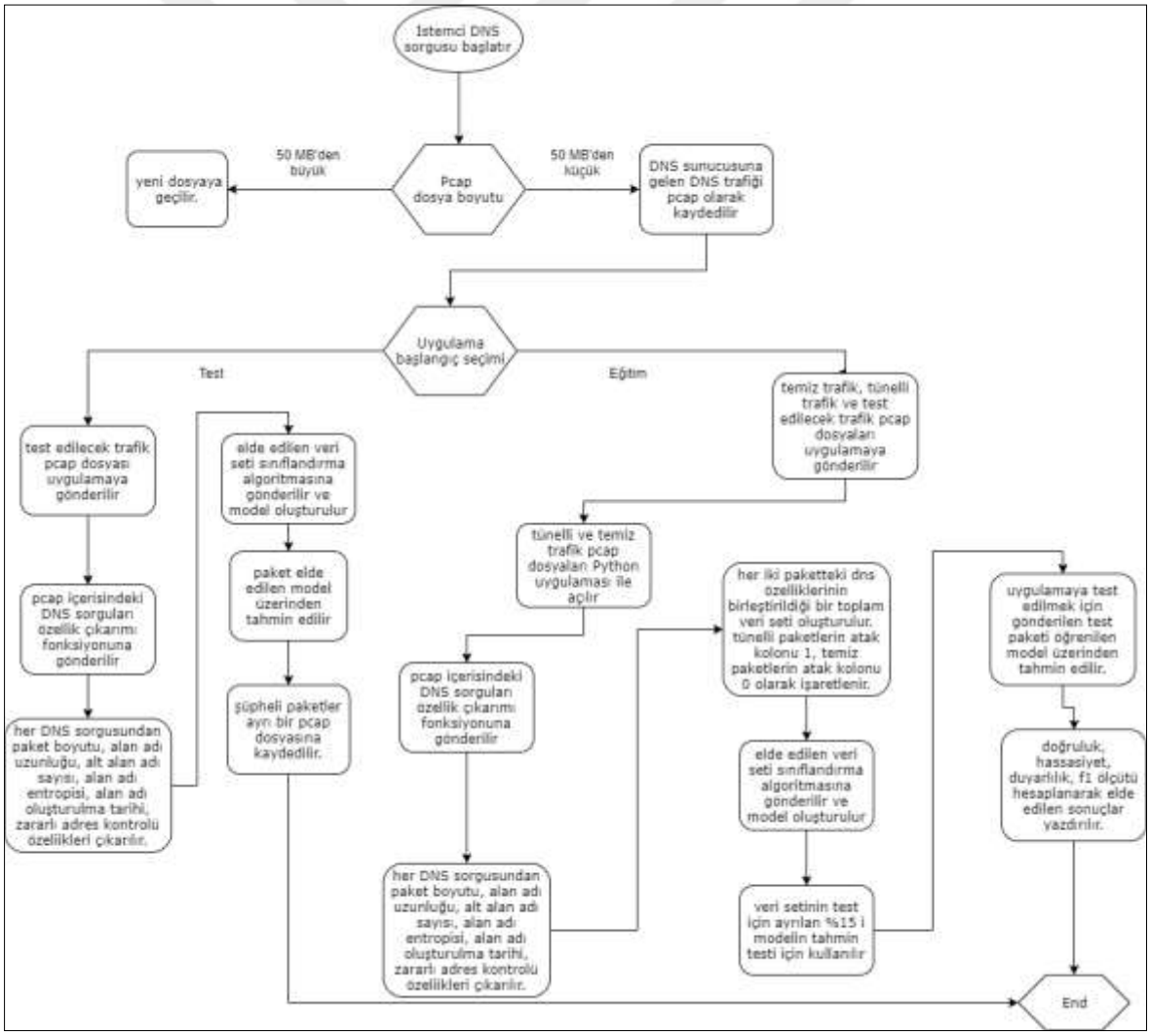
Resim 6.6. Uygulamanın öğrenme amacıyla çalıştırılması

Tek parametre ile uygulama başlatıldığında ise öğrenilmiş model üzerinden test işlemi gerçekleştirilmektedir. Bu durumda ise uygulamaya parametre olarak verilecek dosya ise test edilmesi istenen dosya olacaktır.

```
Parameters:  pcap/test.pcap
```

Resim 6.7. Uygulamanın test amacıyla çalıştırılması

Uygulamanın genel akış şeması ise Şekil 6.2'deki diyagramda gösterilmektedir:



Şekil 6.2. Uygulama akış şeması

Geliştirilen uygulamada tünellemenin tespit edilmesi için paketlerdeki bazı nitelikler incelenmiştir. Uygulamada kullanılacak ayrım niteliklerinden biri olan etki alanı adı entropi hesaplama kuralı olarak Eş. 6.1.'de gösterilen Shannon Entropy formülü kullanılmıştır:

$$Entropi = \sum_{k=0}^{N} (p_k \, ln \, p_k)$$

Eş. 6.1. Shannon Entropi formülü

p_k etki alanı adı içerisindeki k. sırada gelen sembolün tekrar sayısını göstermektedir.

Entropi hesaplama için kullanılan Python kod bloğu Resim 6.8'de yer almaktadır. Bu fonksiyona parametre olarak gönderilen değer ise sorgudaki QNAME alanıdır.

```
def entropiHesapla(text):
    if not text:
        return 0
    entropi = 0
    text = text.translate(None, '._-')
    for x in range(256):
        p_x = float(text.count(chr(x))) / len(text)
        if p_x > 0:
            entropi += - p_x * math.log(p_x, 2)
    return entropi
```

Resim 6.8. Entropi hesaplama kod parçacığı

Uygulamada kontrol edilmesi gereken niteliklerden biri olan sorgu yapılan alt etki alanı sayısı ise Resim 6.9'daki Python kod bloğu ile bulunmuştur:

```
def altDomainHesapla(domain):
    diziHesap = list(reversed(domain.split('.')))[1:]
    sayi = 0
    tmpPtr = root
    for eleman in diziHesap:
        for child in tmpPtr.children:
            if child.data == eleman:
                tmpPtr = child
                sayi += len(child.children)
                break
    altDomainDizi.append(count)
```

Resim 6.9. Alt etki alanı sayısı hesaplama kod parçacığı

Etki alanı adı uzunluğu bilgisi Resim 6.10'daki Python kod bloğu ile hesaplanmıştır:

```
def alanAdiUzunluguHesapla(domain):
    uzunlukDizi.append(len(domain))
```

Resim 6.10. Etki alanı adı uzunluğu hesaplama kod parçacığı

Etki alanı adının ne zaman oluşturulduğunu bulmak için öncelikle Python çalıştırılan Linux sisteme whois paketi kurulmuştur:

```
pip install whois
```

Python'da whois kütüphanesi import edilerek Resim 6.11'deki kod bloğu aracılığıyla domain isminin ne zaman oluşturulduğu bilgisi çıkarılmıştır:

```
def olusturulmaTarihiHesapla(domain):
    simdikiTarih = datetime.datetime.now().replace(microsecond=0)
    fld = tldextract.extract(domain)
    try:
        pwhois.net.socket.setdefaulttimeout(10.0)
        domainWhoIs = pwhois.get_whois(fld.registered_domain)
        olusturulmaTarih = domainWhoIs['creation_date'][0]
        fark = simdikiTarih - olusturulmaTarih
        fark2 = round(fark.total_seconds() / 31536000, 3)
        olusturulmaTarih.append(fark2)
        tarihler[fld.registered_domain] = fark2
    except:
        olusturulmaTarih.append(None)
        tarihler[fld.registered_domain] = None
```

Resim 6.11. Alan adı oluşturulma tarihi hesaplama kod parçacığı

Çıkarılan etki alanı bilgilerinden ‘creation_date’ bilgisi bir değişkende tutularak etki alanı isminin ne zaman oluşturulduğu elde edilmiştir. Bu hesaplamada oluşturulma tarihi ile bugün arasındaki yıl farkı bulunarak veri setinde yıl farkı değeri kullanılmıştır. Etki alanı adının oluşturulma değerinin hesaplanamadığı durumlarda ise “None” değeri eklenir. Veri işleme kısmında bu “None” değerlere ait paketler setten temizlenmektedir.

Sorgulanan etki alanına ait IP adresinin zararlı olarak işaretlenip işaretlenmediği kontrolü “isthisipbad” Python kod parçacığı [24] kullanılarak elde edilmiştir. Bu kod parçacığı tanımlı verilen etki alanı adına ait IP adresini bularak bu adresi popüler IP ve DNS engelleme listelerinde kontrol etmektedir. Kontrol edilen IP adresi bu listelerden birinde bulunursa bu kontrol “true” değer döndürecektir.

```
def badipHesapla(domain):
    if domain != "" and domain != " ":
        sonuc = ipbad.checkip(domain)
        badipSonuc.append(sonuc)
        badIpler[domain] = sonuc
```

Resim 6.12. Alan adı IP adresinin zararlı sitelerde kontrol edilmesi kod parçacığı

Uygulamada bu parametrelerin çeşitli kombinasyonları kullanılarak tünelli paket ve tünelsiz paketlerden öğrenme veri seti oluşturulmakta ve model öğrenme işlemi

gerçekleştirilmektedir. Veri setinin %85'i öğrenme amacıyla %15'i ise test amacıyla kullanılmaktadır. Sonrasında aynı şekilde uygulamanın üçüncü parametresi olan test edilecek paket öğrenilen bu model üzerinden test edilmektedir. Modelin hassasiyetinin artırılması için pakette boş paketler veri setinden çıkarılmakta ve normalizasyon sağlanmaktadır.

Uygulama sonuç çıktısı olarak doğruluk, hassasiyet, duyarlılık, f1 skoru ve eğri altı alan değerleri elde edilmektedir.

Uygulamada işlemler tamamlandığında tünelli olarak sınıflandırılan veri paketleri sonuc.pcap isimli bir dosyaya kaydedilmektedir. Sonuçlar içerisinde etki alanlarının dağılımı da çıktı olarak elde edilmektedir.

6.3. Uygulama Sonuçları

Önerilen çözümün başarısı öğrenme sürecinde kullanılan DNS tünelleme barındıran pcap dosyasının veya tüm trafiğinin DNS tünelleme içerdiğinden emin olunan yeni bir pcap dosyasının tekrar teste tabi tutulması sonucu elde edilen DNS tünelleme paketleri gözlemlenerek hesaplanabilir.

Yapılan testlerde elde edilen sonuçlara göre True Positive, True Negative, False Positive ve False Negative sonuçlar çıkarılmıştır.

TP: True Positive, DNS tünelleme barındırdığı sonucu çıkarılan paketin gerçekten DNS tünelleme barındırıyor olması durumundaki DNS paket sayısı.

FP: False Positive, Kernel SVM eğrisine göre DNS tünelleme olarak işaretlenen ancak gerçekte temiz olan DNS paket sayısı.

FN: False Negative, Kernel SVM eğrisine göre temiz olarak işaretlenen ancak gerçekte DNS tünelleme içeren DNS paket sayısı.

TN: True Negative, Kernel SVM eğrisine göre temiz olarak işaretlenip gerçekte de temiz olan DNS paketi sayısı.

İçeriği herkes için açık olan genel pcap dosyaları ve uygulama kapsamında çeşitli araçlarla oluşturulan 10 farklı pcap dosyası üzerinde yapılan testlerden en yüksek sonuç entropi, alt domain sayısı, sorgu uzunluğu ve alan adının oluşturulma tarihi bilgilerine bakıldığında elde edilmiştir. Paket boyutu, alan adı oluşturulma tarihi ve alan adının daha önce zararlı olarak işaretlenip işaretlenmediği kontrollerinin tamamı uygulandığında 0.96 başarım oranı elde edilmiştir. Başarım oranının hesaplanması aşağıdaki eşlenikte gösterilmiştir.

Çizelge 6.1. Hata matrisi

Sınıflandırma Sonucu / Gerçek durum	Gerçekte tünelleme içeren paket	Gerçekte temiz paket
Tünelleme içeriyor	TP:2088	FP:86
Temiz trafik	FN:46	TN:1100

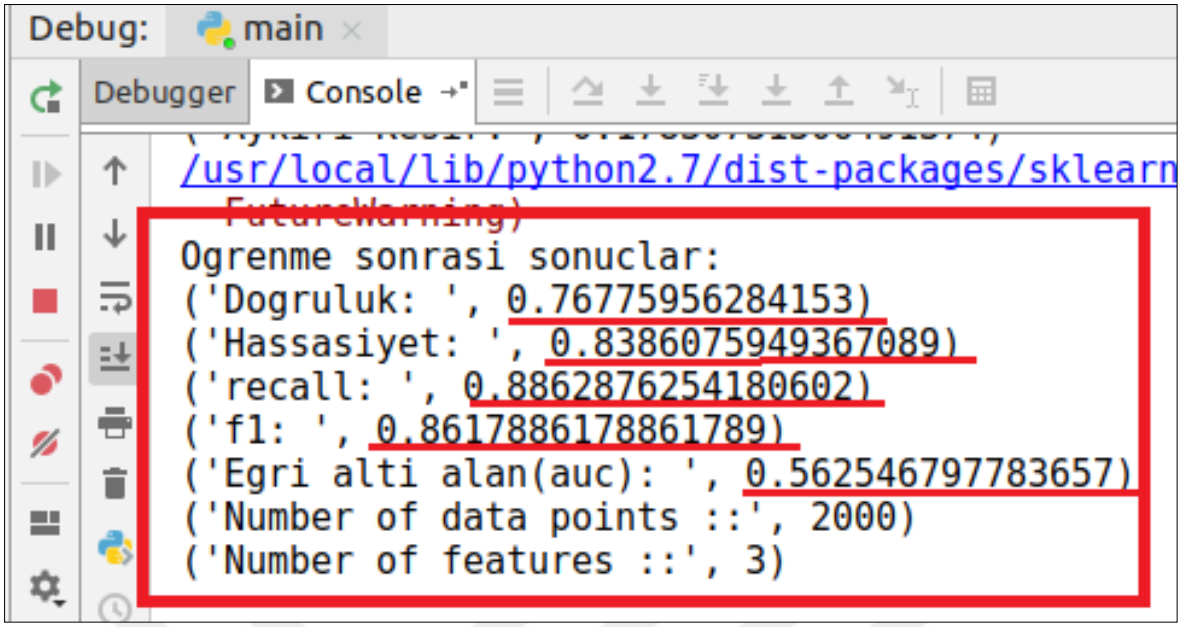
Bu değerlere göre önerilen çözümün başarım oranını Eş.6.2'deki formül ile hesaplayabiliriz:

$$Accuracy - Başarım = \frac{TP+TN}{Tüm\ Paketler} \rightarrow \frac{2088+1100}{2088+1100+86+46} = 0.96$$

Eş.6.2. Başarım oranı

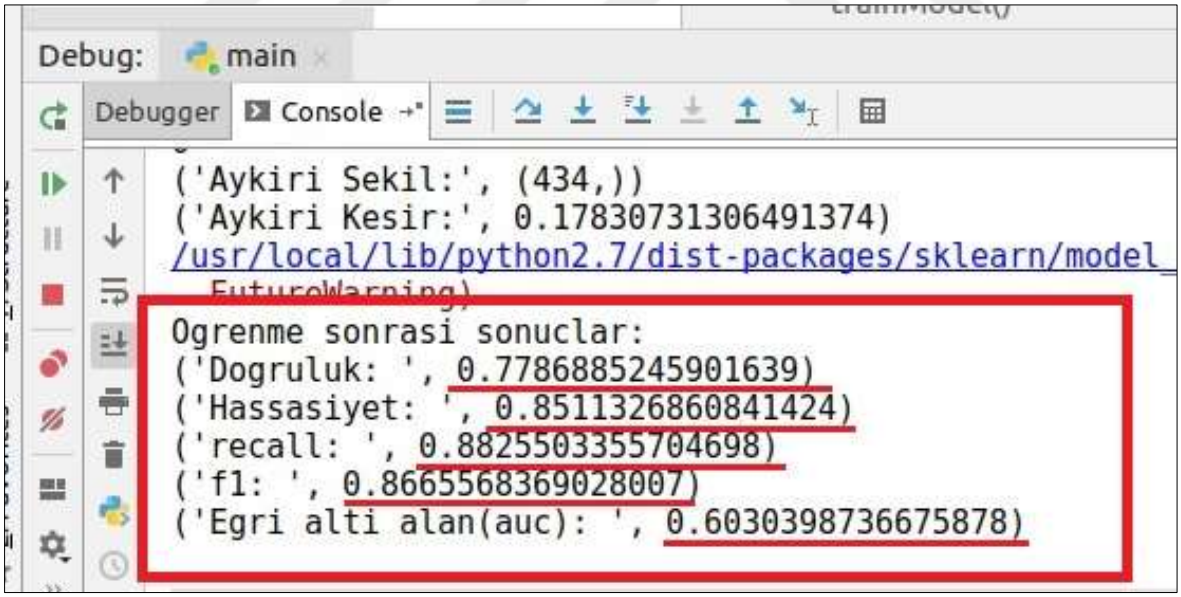
Bu sonuca göre önerilen çözümün başarım oranını %96 olarak hesaplanmıştır.

DNS tünelleme tespitinde şimdiye kadar yapılan çalışmalarda yoğunlukla entropi, alt alan adı sayısı ve sorgu uzunluğuna bakılmıştır. Bu tez çalışmasında geliştirilen uygulamada da ilk olarak bu parametrelerle test işlemi yapılmıştır. Bu işlemde elde edilen sonuçlar Resim 6.13' de gösterilmektedir.



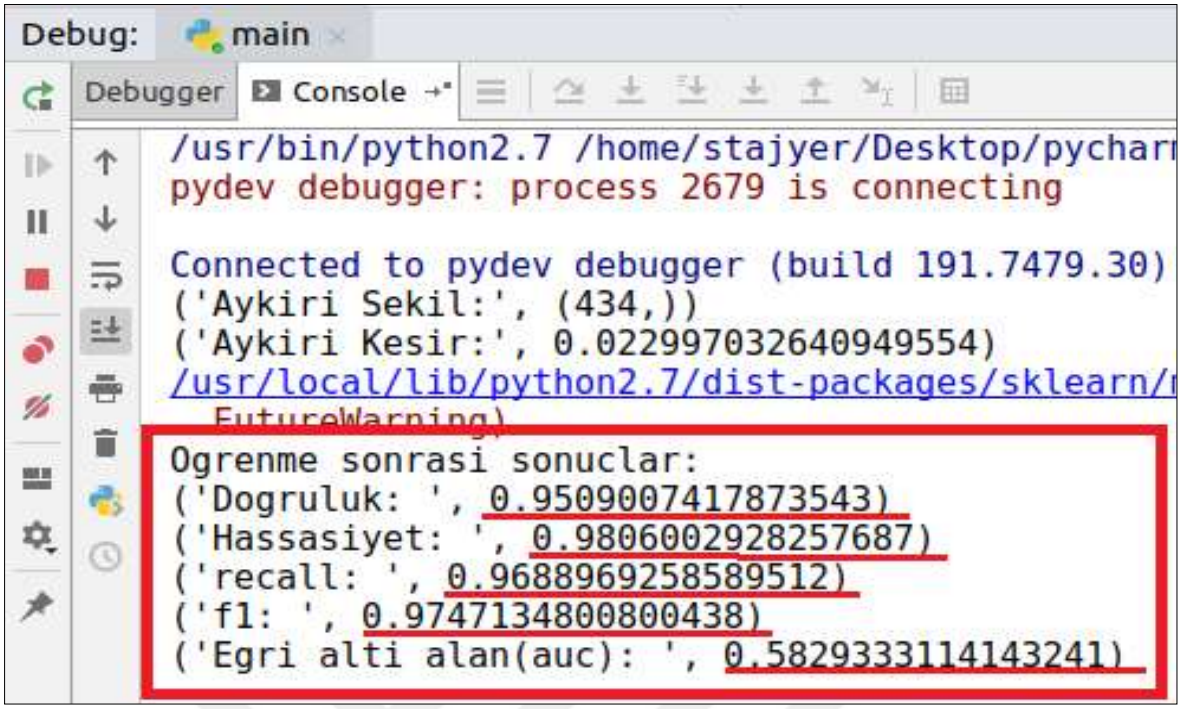
Resim 6.13. Entropi, alt alan adı sayısı ve sorgu uzunluğu sonuçları

İkinci çalışmada ise bu parametrelere ek olarak alan adının zararlı IP adresleri listelerinde yer alıp almadığı kriterine bakılmıştır ve Resim 6.14'teki sonuçlar elde edilmiştir.



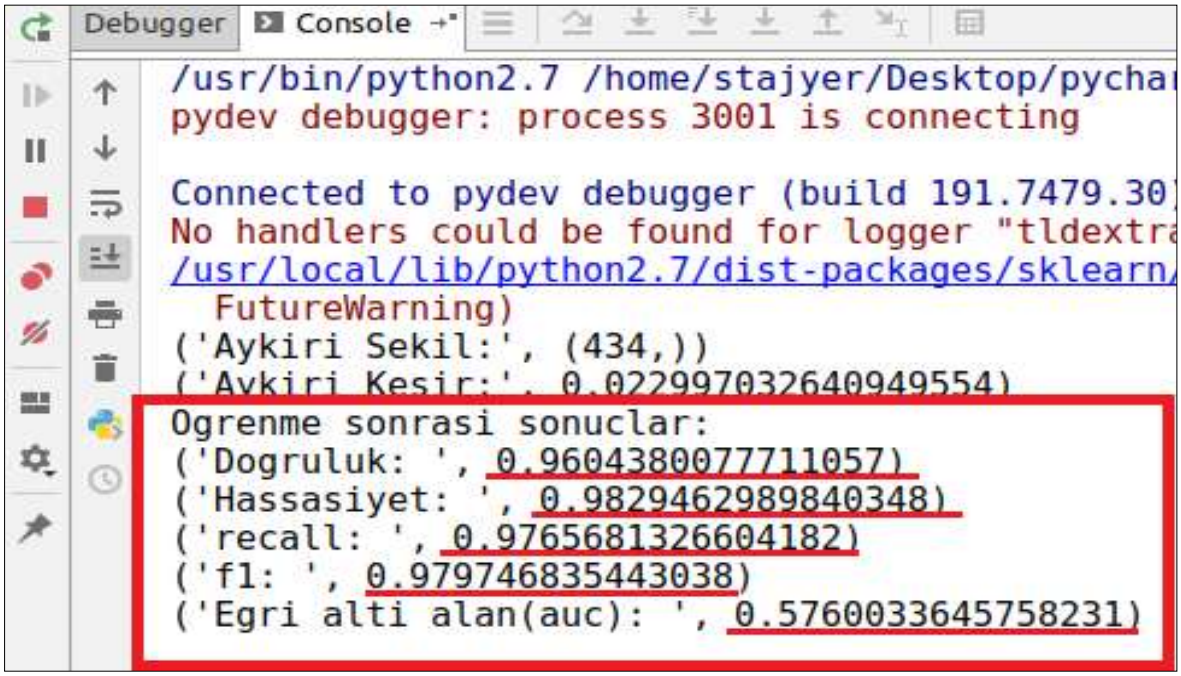
Resim 6.14. Entropi, alt alan adı sayısı, sorgu uzunluğu ve zararlı IP kriteri sonuçları

Üçüncü çalışmada ise entropi, alt alan adı sayısı ve sorgu uzunluğu parametrelerine ek olarak paket boyutu kriteri eklenmiştir ve Resim 6.15'teki sonuçlar elde edilmiştir.



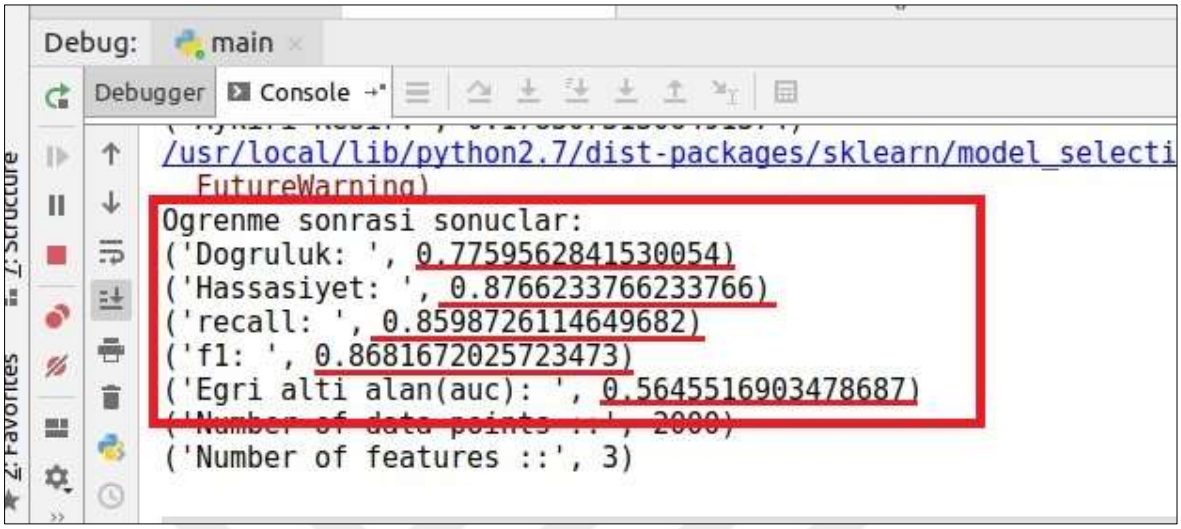
Resim 6.15. Entropi, alt alan adı sayısı, sorgu uzunluğu ve paket boyutu sonuçları

Dördüncü çalışmada ise entropi, alt alan adı sayısı, sorgu uzunluğu ve alan adının oluşturulma tarihi parametrelerine bakılmıştır. Elde edilen sonuçlar ise Resim 6.16'da gösterilmektedir.



Resim 6.16. Entropi, alt alan adı sayısı, sorgu uzunluğu ve alan adının oluşturulma tarihi sonuçları

Son çalışmada ise bu parametrelerin hepsi kullanılarak model oluşturulmuştur. Test sonucunda elde edilen sonuçlar ise Resim 6.17'de yer almaktadır.



Resim 6.17. Tüm parametrelerin kullanılması ile elde edilen sonuçlar

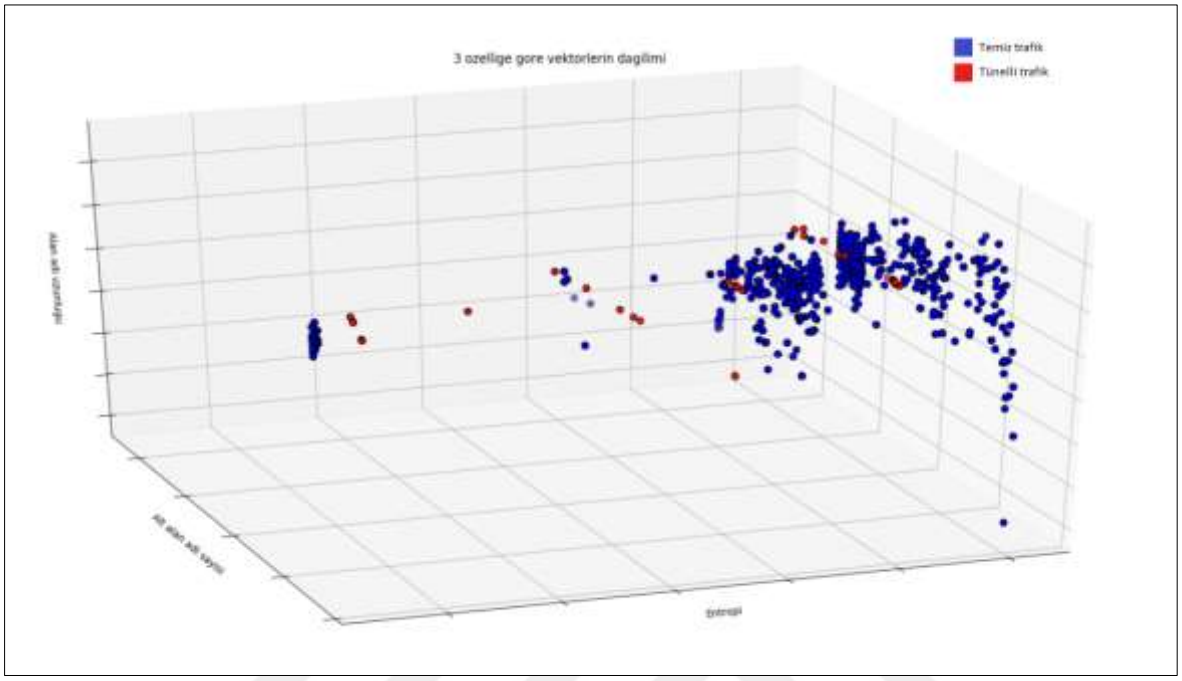
Elde edilen sonuçlara göre daha önce değerlendirilen entropi, alt alan adı sayısı ve sorgulanan alan adı uzunluğu özelliklerine paket boyutu eklendiğinde f1 ölçüm oranının 0.9747'ye yükseldiği, paket boyutu yerine alan adı oluşturulma tarihi eklendiğinde ise 0.9797'ye yükseldiği gözlenmiştir. Elde edilen tüm sonuçların karşılaştırılması Çizelge 6.2'de yer almaktadır.

Çizelge 6.2. Parametre ölçüm sonuçları

Ölçümler / **Parametreler**	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	f1 Ölçümü
Entropi + Alt alan adı sayısı + Alan adı uzunluğu	0.7677	0.8386	0.8862	0.8617
Entropi + Alt alan adı sayısı + Alan adı uzunluğu + paket boyutu	0.9509	0.9806	0.9688	0.9747
Entropi + Alt alan adı sayısı + Alan adı uzunluğu + oluşturulma tarihi	0.9604	0.9829	0.9765	0.9797
Entropi + Alt alan adı sayısı + Alan adı uzunluğu + zararlı IP kontrolü	0.7786	0.8511	0.8825	0.8665
Entropi + Alt alan adı sayısı + Alan adı uzunluğu + paket boyutu+ oluşturulma tarihi + zararlı IP kontrolü	0.7759	0.8766	**0.8598**	0.8681

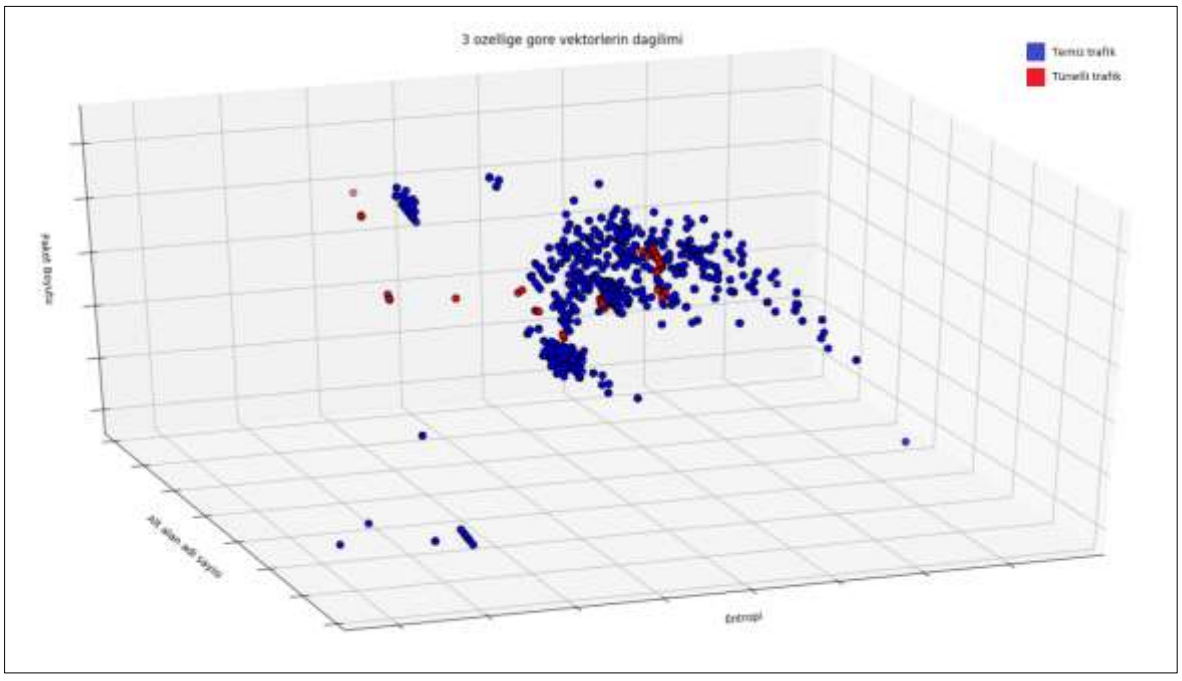
Sonuçların görselleştirilmesi için Python matplotlib kütüphanesi kullanılmıştır. Bu kütüphaneyi kullanarak grafik oluşturabilmek için üçerli parametre setleri seçilmiştir.

İlk sette alan adı uzunluğu, alt alan adı sayısı ve entropi değerleri kullanılmıştır. Elde edilen vektör dağılımı sonucu Resim 6.18'de yer almaktadır.



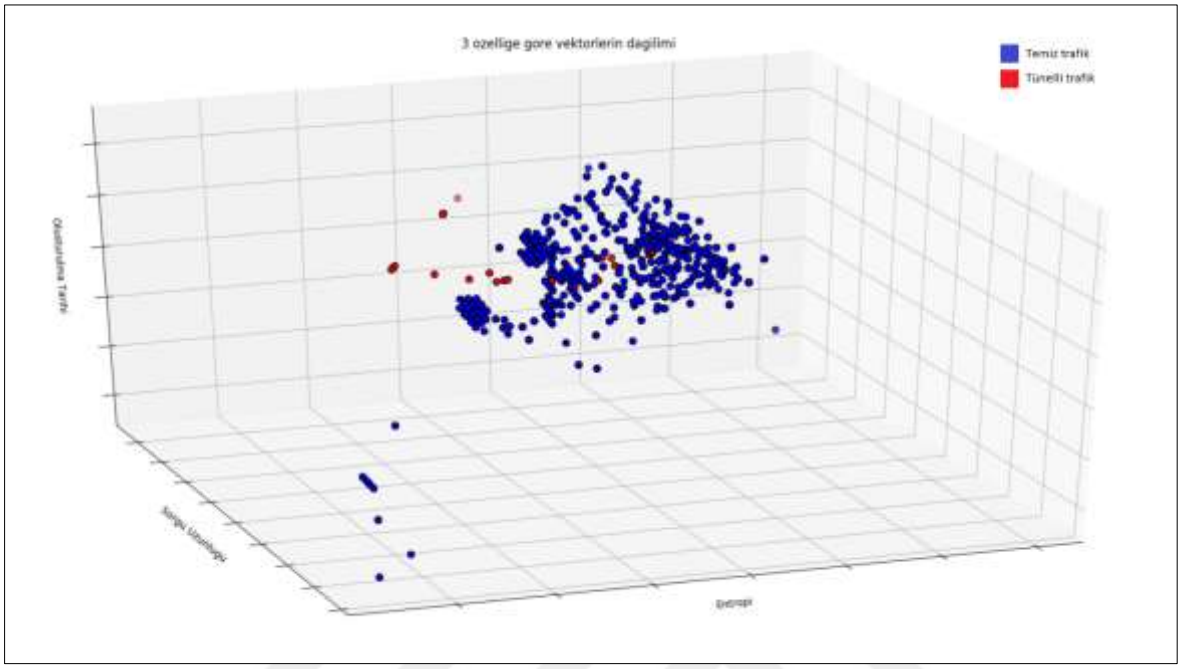
Resim 6.18. Alan adı uzunluğu, alt alan adı sayısı ve entropi değerleri vektör dağılımı

Paket boyutu, alt alan adı sayısı ve entropiye bakıldığında ise dağılım grafiği Resim 6.19'da yer almaktadır.



Resim 6.19. Paket boyutu, alt alan adı sayısı ve entropi değerleri vektör dağılımı

Sorgu uzunluğu, entropi ve oluşturulma tarihi kriterlerine bakıldığında ise Resim 6.20'deki vektör dağılımı elde edilmiştir.



Resim 6.20. Sorgu uzunluğu, entropi ve oluşturulma tarihi değerleri vektör dağılımı

7. SONUÇ VE ÖNERİLER

Önerilen çözümde Python programlama dili kullanılarak geliştirilen Support Vector Machine sınıflandırma algoritması kullanılmıştır. Uygulanan çözümde daha önce farklı çalışmalar içerisinde kullanılmış parametreler toplu ve bütünleşik bir şekilde kullanılmıştır. Daha önce çalışma gerçekleştirilen parametrelere ek olarak, etki alanı isminin ne zaman oluşturulduğu ve etki alanına ait çözümlenen IP adresinin zararlı olarak işaretlenip işaretlenmediği kontrolü eklenmiştir. Tüm bu kontroller Support Vector Machine sınıflandırma algoritması uygulanarak temiz ve tünelleme yapılmış pcap dosyalarından lineer SVM ayrımı hesaplanmıştır. Öğrenme süreci tamamlandığında test edilmek istenen Pcap dosyası uygulamaya sokularak içerisindeki daha önce öğrenilen lineer SVM eğrisine göre DNS tünelleme olması muhtemel paketler tespit edilmeye çalışılmıştır.

Elde edilen sonuçlara göre daha önce yapılan çalışmalara ek olarak kullanılan parametrelerin başarım oranını artırdığı görülmektedir. En etkin katkının ise alan adının oluşturulma tarihi kriterine bakıldığında elde edildiği gözlemlenmektedir.

Aşırı uyma/uyum gösterme probleminden kaçınma adına model, temiz ve tünelli olduğundan emin olunan farklı veri setleri ile denenmiştir. Bu problemden kaçınmaya devam etmek adına sürekli farklı ve daha geniş veri setleri üzerinde modeli uygulamanın SVM performansı düşürücü etkisi olduğu değerlendirilebilmektedir. Uygulama kapsamında gerçekleştirilen testlerde başarım oranının düştüğü gözlenmemiştir. SVM'in veri seti genişledikçe başarım oranın düşebileceği ihtimaline karşın uygulama alanının nitelikleri doğru ve tam bir şekilde belirlenerek bu niteliklerin tümünü kapsayan veri setleri oluşturmak faydalı olacaktır. SVM'de diğer sınıflandırma algoritmalarına oranla aşırı uyma/uyum gösterme problemi ile daha az karşılaşılmaktadır.

Örtüşme probleminden kaçınmak için ise gelecek çalışmalarda boyut azaltma konsepti ile ilgili bir teknik kullanılarak veri seti tekrar iki boyutlu hale getirilerek iki boyutlu veri üzerinde SVM uygulanabilir.

"Whois" sorgusu yaparak etki alanının ne zaman kaydedildiği bilgisi çıkarılmaktadır. Ancak whois sorgusu yapılan mevcut sunucuların çoğunda aynı ip'den yapılabilecek sorgu sayısını sınırlayan bir limit bulunmaktadır. Bunu aşmak için tüm whois sunucuları bir listede toplanarak her sorguda rastgele bir sunucu seçilerek whois sorgusu her seferinde

farklı bir sunucuya doğru yapılmıştır. Yine de whois sunucularının birbiriyle de veri paylaşabileceklerini göz önünde bulundurarak whois sorgusu yapmak için ücretli bir sunucu kullanmak değerlendirilebilir.

DNS sorgusu yapıldığında alınan cevaba göre ilgili alan adı sunucusu veya sorgulanan kayda ait asıl sunucunun IP adresinin daha önce herhangi bir kötücül amaç için kullanılıp kullanılmadığını tespit etmek üzere 73 adet kara liste üzerinde sorgulama yapılmıştır. Bu kara liste tutan adreslerin bazılarında robotik erişimleri engellemek üzere güvenlik kodu (captcha) kullanılmaktadır. Bu durumu çözümleyebilmek için ise kara liste sitelerinde yayınlanmış tüm adresler manüel bir şekilde belirli ve yeterli aralıklarla yerel bir adrese taşınarak IP adresinin kara listede yayınlanmış olup olmadığının kontrolü bu adres üzerinden gerçekleştirilebilir.

KAYNAKLAR

1. İnternet: IANA (2008). Domain Name System (DNS) parameter. URL: http://www.iana.org/assignments/dns-parameters, Son Erişim Tarihi: 09.11.2019.

2. Albitz, P. and Liu, C. (2001). *DNS and BIND*. (4th Ed). Massachusetts: O'Reilly.

3. İnternet: ICANN (2008). *Icann homepage*. URL: http://www.icann.org/, Son Erişim Tarihi: 09.11.2019.

4. İnternet: NSTX. (2002). *Tunneling network-packets over DNS – summary*. URL: http://savannah.nongnu.org/projects/nstx/, Son Erişim Tarihi: 09.11.2019.

5. Kozierok, C. (2005). *The TCP/IP guide: A comprehensive, illustrated internet protocols reference*. (1st Ed.). San Francisco: No Starch Press.

6. İnternet: "DNSSEC", URL: http://www.dnssec.net/, Son Erişim Tarihi: 09.11.2019.

7. Maloof, M. A. (2006). *Machine learning and data mining for computer security*. Berlin: Springer Verlag.

8. İnternet: Bcher, P., Holz, T., Ktter, M. and Wicherski, G. (2005). *Know your enemy: Tracking botnets*. The Honeynet Project & Research Alliance. URL: http://www.honeynet.org/papers/bots/, Son Erişim Tarihi: 09.11.2019.

9. Houle, K. J. and Weaver, G. M. (2008). *Trends in denial of service attack technology*. CERT, Carnegie Mellon University.

10. Modi, C., Patel, D., Borisaniya, B., Patel, H., Patel, A. and Rajarajan, M. (2012). A survey of ıntrusion detection techniques in cloud. *Journal of Network and Computer Applications*, 36(1), 42-57.

11. Tsai, C. F., Hsu, Y. F., Lin, C. Y. and Lin, W. Y. (2009). Intrusion detection by machine learning: A review. *Expert Systems With Applications*, 36(10), 11994–12000.

12. Kolias, C., Kambourakis, G. and Maragoudakis, M. (2011). Swarm intelligence in intrusion detection: A Survey. *Computers & Security*, 30(8), 625-642.

13. Gudekli, U. T. and Ciylan, B. (2019). DNS tunneling effect on dns packet sizes. *International Journal of Computer Science and Mobile Computing*, 8(1), 154-162.

14. Bubnov, Y. (2016). DNS tunneling detection. *Internatinal Journal of Recent Trends in Engineering & Research*, 2(4), 241-245.

15. Mockapetris, P. (1987). *Domain names - implementation and specification, the internet society.* RFC 882, USC/Information Sciences Institute.

16. İnternet: Berkeley University of California (2008). *Bind DNS software*. URL: http://www.isc.org/index.pl?/sw/bind/index.php, Son Erişim Tarihi: 09.11.2019.

17. Brandhorst, C. J. and Pras, A. (2005). *Dns: A statistical analysis of name server traffic at local network-to-internet connections*. Electrical Engineering, Mathematics and Computer Science, University of Twente.

18. Buczak, A. L., Hanke, P. A., Cancro, G. J., Toma, M. K., Watkins, L. A. and Chavis, J. S. (2016). *Detection of tunnels in PCAP data by random forests*. Proceedings of the 11th Annual Cyber and Information Security Research Conference, CISRC 2016.

19. Perdisci, R., Antonakakis, M. and Lee, W. (2008). *Solving the DNS cache poisoning problem without changing the protocol*. College of Computing, Georgia Institute of Technology, Atlanta.

20. Zhong, S., Khoshgoftaar, T. M. and Seliya, N. (2007). Clustering-based network intrusion detection. *International Journal of Reliability, Quality and Safety Engineering*, 14(2), 169-187.

21. Patel, R., Thakkar, A. and Ganatra, A. (2012). A survey and comparative analysis of data mining techniques for network intrusion detection systems. *International Journal of Soft Computing and Engineering*, 2(1), 265-271.

22. Nadiammai, G. V., Krishnaveni, S. and Hemalatha, M. (2011). A comprehensive analysis and study in intrusion detection system using data mining techniques. *International Journal of Computer Applications*, 35(8), 51-56.

23. Brugger, S. T. (204). *Data mining methods for network intrusion detection*. UC Davis Dissertation Proposal, Department of Computer Science, University of California.

24. Satam, P., Alipour, H., Al-Nashif, Y. and Hariri, S. (2015). *Anomaly behavior analysis of DNS protocol.* Department of Electrical and Computer Engineering, University of Arizona, Tucson.

25. Yamada, A., Miyake, Y., Terabe, M., Hashimoto, K. and Kato, N. (2009). *Anomaly detection for DNS servers using frequent host selection.* Proceedings of the 23rd International Conference on Advanced Information Networking and Applications (AINA'09), Bradford, UK.

26. Wang, Y., Hu, M.-Z., Li, B. and Yan, B.R. (2006). *Tracking anomalous behaviors of name servers by mining DNS traffic*. Proceeding of the ISPA 2006 Workshops, Sorrento, Italy.

27. Kambourakis, G., Moschos, T., Geneiatakis, D. and Gritzalis, S. (2007). *A fair solution to DNS amplification attacks*. Proceedings of the 2nd International Workshop on Digital Forensics and Incident Analysis (WDFIA'07), Samos, Greece.

28. Aiello, M., Mongelli, M. and Papaleo, G. (2015). DNS tunneling detection through statistical fingerprints of protocol messages and machine learning. *International Journal of Communication Systems*, 28(14), 1987–2002.

29. Kantardzic, M. (2003). *Data mining: Concepts, models, methods, and algorithms*. New Jersey: John Wiley & Sons.

30. Ozturk, F. and Subasi, A. (2010). *Comparison of decision tree methods for ıntrusion detection*. Proceeding of the 2nd International Symposium on Sustainable Development, 401-407.

31. Huang, J., Lu, J. and Ling, C. X. (2003). *Comparing naive bayes, decision trees and SVM with AUC and accuracy*. Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM'03).

32. Zhang, J. and Zulkernine, M. (2005). *Network intrusion detection using random forests*. Proceeding of the 3rd Annual Conference on Privacy, Security and Trust, Canada.

33. Jain, A., Sharma, S. and Sisodia, M. S. (2011). Network intrusion detection by using supervised and unsupervised machine learning techniques: A survey. *International Journal of Computer Technology and Electronics Engineering*, 1(3), 14-20.

34. Aiello, M., Mongelli, M. and Papaleo, G. (2013). *Basic classifers for DNS tunneling detection.* Proceedings of the 18th IEEE Symposium on Computers and Communications,Croatia.

35. Berkhin, P. (2006). A survey of clustering data mining techniques. (Edt. Kogan, J., Nicholas, C., and Teboulle, M.). In: *Grouping multidimensional data*. Berlin: Springer.

36. Plonka, D. and Barford, P. (2008). *Context-aware clustering of DNS query traffic.* Proceedings of the 8th ACM SIGCOMM Internet Measurement Conference, Greece.

37. Buczak, A. L., Hanke, P. A., Cancro, G. J., Toma, M. K., Watkins, L. A. and Chavis, J. S. (2016). *Detection of tunnels in PCAP data by random forests*. Proceedings of the 11th Annual Cyber and Information Security Research Conference, CISRC 2016.

38. Pearson, O. (1998). *Dns tunnel - through bastion hosts*. Bugtraq posting, Web: http://seclists.org/bugtraq/1998/Apr/0079.html, Son Erişim Tarihi: 09.11.2019.

39. Revelli, A. and Leidecker, N. (2009). *Introducing heyoka: DNS tunneling 2*.0

40. Bianco, D. (2006). *A traffic-analysis approach to detecting dns tunnels.*

41. İnternet: Pietraszek, "DNSCat", (2004). URL: http://tadek.pietraszek.org/projects/DNScat, Son Erişim Tarihi: 09.11.2019.

42. İnternet: Skoudis, "The six most dangerous new attack techniques and what's coming next?", (2012). URL: https://blogs.sans.org/pentesting/files/2012/03/RSA-2012-EXP-108-Skoudis-Ullrich.pdf, Son Erişim Tarihi: 09.11.2019.

43. Farnham, G. and Atlasis, A. (2013). *Detecting dns tunneling. SANS Institute InfoSec Reading Room 9.*

44. İnternet: Maarten Van Horenbeeck, "Dns tunneling", (2006). URL: http://www. daemon.be/maarten/dnstunnel, Son Erişim Tarihi: 09.11.2019.

45. Butler,P. (2011). “Quantitatively analyzing stealthy communication channels”, Computer Science, Virginia Tech, Blacksburg, VA.

46. Bilge, L., Kirda, E., Kruegel, C. and Balduzzi, M., (2011) “Exposure: finding Malicious Domains Using Passive DNS Analysis”, Boston. Northeastern University.

47. Born, K. (2010). *Detecting DNS tunnels using character frequency analysis*. Kansas State University, (2010b).

48. Born, K. and Gustafson, D. (2010a). NgViz: *Detecting DNS tunnels through ngram visualization and quantitative analysis.* Proceedings of the 6th Annual Workshop on Cyber Security and Information Intelligence Research, Tennessee, USA.

49. Born, K. and Gustafson, D. (2010b). *Detecting DNS tunnels using character frequency analysis.* Proceedings of the 9th Annual Security Conference, Las Vegas, Nevada.

50. Lockington, R.A., Rodbourn, L., Barnett, S., Carter, C.J. and Kelly, J.M. (2002). Regulation by carbon and nitrogen sources of a family of cellulases in Aspergillus nidulans. *Fungal Genet Biol, 37*(2), 190–196.

51. Shafieian, S., Smith, D. and Zulkernine, M. (2017). *Detecting DNS tunneling using ensemble learning*. USA: Springer International Publishing.

52. Homem, I., Papapetrou, P. and Dosis, S. (2016). *Entropy-based prediction of network protocols in the forensic analysis of DNS tunnels*. World Congress on Internet Security (WorldCIS), London.

53. Brownlee, N., Wessels, D. and Zdrnia, B. (2007). *Passive monitoring of DNS anomalies.* Auckland. University of Auckland.

54. Almusawi, A. (2017). *DNS tunneling detection method based on multilabel support vector machine.* M.S. Thesis, Ferdowsi University of Mashhad, Iran.

55. İnternet: CAIDA (2018). IPv4 Routed /24 DNS Names Dataset. URL: https://www.caida.org/data/active/ipv4_dnsnames_dataset.xml, Son Erişim Tarihi: 09.11.2019.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : GÜDEKLİ, Uğur Tanık

Uyruğu : T.C.

Doğum tarihi ve yeri : 30.05.1987, Ankara

Medeni hali : Evli

Telefon : 0 (312) 801 07 40

e-mail : ugurtanik.gudekli@gazi.edu.tr

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Yüksek lisans	Gazi Üniversitesi / Bilişim Sistemleri Ana Bilim Dalı	Devam ediyor
Lisans	Kocaeli Üniversitesi / Bilgisayar Mühendisliği	2009
Lise	Fahrettin Kerim Gökay Anadolu Lisesi	2005

İş Deneyimi

Yıl	Yer	Görev
2017-Halen	Microsoft	Uzman Saha Mühendisi
2013-2017	Türk Havacılık Uzay Sanayii	Bilgi Teknolojileri Uzman Mühendisi
2011-2013	Türk Telekomunikasyon A.Ş.	Çevresel Sistemler Mühendisi
2010-2011	Probilişim	Sistem Mühendisi
2007-2009	Bilgi Birikim Sistemleri	Sistem Destek Uzmanı

Yabancı Dil

İngilizce

Yayınlar

1. Güdekli, U.T. (2019). DNS Tunneling Effect On DNS Packet Sizes. *International Journal of Computer Science and Mobile Computing*, 8, 1.

Hobiler

Film, Seyahat, Yürüyüş

GAZİLİ OLMAK AYRICALIKTIR…