TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SALDIRI TESPİT VE ENGELLEME SİSTEMLERİNİ ATLATMA SALDIRILARI

YÜKSEK LİSANS TEZİ

Hakan KILIÇ

Bilgisayar Mühendisliği Anabilim Dalı
Bilgi Güvenliği

Tez Danışmanı: Prof. Dr. Ali Aydın Selçuk

AĞUSTOS 2019

Fen Bilimleri Enstitüsü Onayı

……..
Prof. Dr. Adı SOYADI
Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksininlerini sağladığını onaylarım.

……
Prof. Dr. Oğuz ERGİN
Anabilimdalı Başkanı

TOBB ETÜ, Fen Bilimleri Enstitüsü'nün 171111140 numaralı Yüksek Lisans / Doktora Öğrencisi **Hakan KILIÇ**'ın ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı **"SALDIRI TESPİT VE SALDIRI ENGELLEME SİSTEMLERİNİ ATLATMA SALDIRILARI"** başlıklı tezi **Gün,Ay, Yıl** tarihinde aşağıda imzaları olan jüri tarafından kabul edilmiştir.

Tez Danışmanı : **Prof. Dr. Ali Aydın SELÇUK** .....
TOBB Ekonomive Teknoloji Üniversitesi

Jüri Üyeleri : **Prof. Dr. Suat ÖZDEMİR (Başkan)** .....
Gazi Üniversitesi

Prof. Dr. Kemal BIÇAKCI .....
TOBB Ekonomi ve Teknoloji Üniversitesi

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, alıntı yapılan kaynaklara eksiksiz atıf yapıldığını, referansların tam olarak belirtildiğini ve ayrıca bu tezin TOBB ETÜ Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırlandığını bildiririm.

.

Hakan KILIÇ

İMZA

ÖZET

Yüksek Lisans Tezi

SALDIRI TESPİT VE SALDIRI ENGELLEME SİSTEMLERİNİ ATLATMA SALDIRILARI

Hakan KILIÇ

TOBB Ekonomi ve Teknoloji Üniveritesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı
Bilgi Güvenliği

Danışman: Prof. Dr. Ali Aydın SELÇUK

Tarih: Ağustos 2019

Saldırı tespit ve engelleme sistemleri siber güvenlik mimarisinin ilk ve en önemli katmanıdır. Bu teknolojiler sistemlerde bulunan zafiyetleri sömürmeye ve sistemlerin içerisine yasa dışı olarak sızmaya yönelik gerçekleştirilen saldırıların tespit edilmesini ve engellenmesini sağlar. Saldırı tespit ve engelleme sistemleri sızma teşebbüslerini yakalamak ve engellemek için sınır güvenliğinde ve üç katmanlı mimari güvenliğinde kullanılmaktadır. Ek olarak saldırı tespit ve engelleme sistemleri merkezi bir noktadan yönetme ve saldırıları raporlama imkanı sağlamaktadır. Saldırı tespit ve engelleme sistemlerinin atlatılması arkasında koruduğu sistemlerin siber güvenliğinde büyük zafiyetler meydana getirmektedir. Araştırmada yedi adet saldırı tespit ve engelleme sistemlerini atlama tekniği ve bu tekniklerin saldırı tespit ve engelleme sistemlerini atlatma başarı oranları incelenmiştir. Araştırmanın başarı oranlarının ölçülmesi için ilk olarak saldırılar kurban makineye gönderilerek saldırı tespit ve engelleme sisteminde tetiklediği alarm sayısı ölçülmüştür. Daha sonra saldırılar atlatma teknikleri ile birleştirilmiş ve kurban makinelere gönderilerek saldırı tespit ve engelleme sisteminde tetiklediği alarm sayısı kaydedilmiştir. Bunların sonucunda saldırılar ve atlatma teknikleri kullanılarak gerçekleştirilen saldırılardan üretilen alarm sayıları arasındaki yüzdesel fark alınarak atlatma tekniklerinin başarı oranı belirlenmiştir. Araştırma

esnasında kullanılan yedi adet atlatma tekniği; TTL atlatma, MTU ile paket parçalama, zaman, ajan adı ve port numarası değiştirme, kodlama ve gizleme, sahte sağlama kodu, dosya başlığı değiştirme ve dosya ve dizin değiştirme teknikleridir. Alatma teknikleri gerçek saldırılar ile birleştirildiğinde kurban makinelere eşit derecede etki etmesi hedeflenmiş, kurban makineleri sömüremeyen atlatma denemelerine test sonuçlarında yer verilmemiştir. Testler sonucunda gözlemlenen negatif başarılar da test sonuçları tablolarında yer almaktadır. Araştırmanın son bölümünde ise yedi adet atlatma tekniğine karşı alınabilecek önlemler anlatılmaktadır. Önlemler üç adet atlatma tekniğini kapsarken dört adet atlatma tekniğine karşı saldırı tespit ve engelleme sisteminlerinde alınabilecek geçerli bir önlem bulunmamaktadır. Araştırma sırasında kullanılan tüm saldırı kümesi araştırma sırasında oluşturulmuş olup herhangi bir alıntı yapılan saldırı bulunmamaktadır. Testler esnasında Snort saldırı tespit ve engelleme sisteminin en güncel versiyonu olan 2.9.13 versiyonu kullanılmıştır. Araştırma verilerinin de gösterdiği üzere saldırı tespit ve engelleme sistemleri atlatma teknikleri sayesinde günümüzde hala atlatılanilir bir durumdadır.

Anahtar Kelimeler: Saldırı Tespit Sistemi, Saldırı Önleme Sistemi, Siber Saldırı, Siber Güvenlik, Atlatma Teknikleri

ABSTRACT

Master of Science

EVASİON TECHNİQUES EFFİCİENCY OVER THE INTRUSION PREVENTION AND DETECTION TECHNOLOGY

Hakan KILIÇ

TOBB University of Economics and Technology
Institute of Natural and Applied Sciences
Department of Computer Engineering
Information Security

Supervisor: Prof. Dr. Ali Aydın SELÇUK

Date: August 2019

Intrusion Prevention Systems (IPS) and Intrusion Detection Systems (IDS) are the first line of the defense of cyber-environment. This technology is made for capturing and preventing breaches and attacks for perimeter and three-tiered architecture security. Besides intrusion detection and prevention system provides an remarkable of centralized security management and reporting. Evading of an intrusion detection and prevention system creates a large gap in cyber-security. This research examines seven common evasion techniques and success rates of these over the intrusion detection and prevention system. The success rates was measured that firstly attacks without evasion was sent to the victim machines and observed the number of alarm that was triggered by Snort intrusion detection and prevention system. Then attacks were combined with the seven evasion techniques and they were sent to the victim machines and observed the number of alarm. The statistical difference between first and the second situation determines the success rate of the evasion technique. These techniques are TTL evasion, fragmentation with MTU modification evasion, tampering time – agent name and port name evasion, encoding and obfuscation evasion, bad checksum evasion, file header manipulation evasion, file and path change evasion. The test results are shared in tables to demonstrate the exact success rates. There are some trials that gives

negative success on evasion, these are also shared to evaluate the test results objectively. The last part include have suggestions to avoid the seven evasion techniques. There are some countermeasures to mitigate the seven evasion attacks in which there remarkable solutions only for three of them. The rest of evasion techniques have no feasible mitigations. The last version of Snort intrusion detection and prevention system was used to test attacks and evasion techniques. The whole attack and evasion dataset created by contemporary attack techniques during the research. Test results demonstrate that the intrusion detection and prevention system can be bypassed with evasion techniques.

Keywords: Intrusion Detection System, Intrusion Prevention System, Cyber Attack, Cyber Security, Evasion Technique

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren danışman hocam Prof. Dr. Ali Aydın SELÇUK'a, kıymetli tecrübelerinden faydalandığım TOBB Ekonomi ve Teknoloji Üniversitesi Bilgisayar Mühendisliği Bölümü öğretim üyelerine, test ortamının kurulumu ve testlerin gerçekleştirilmesinde büyük yardımları bulunan Sertaç KATAL'a, saldırı senaryolarının gerçekleştirilmesinde yardımcı olan Umut ERGİN'e, araştırmanın konusu ve içeriğini belirlemede verdiği bilgi fikirler nedeniyle Bahtiyar BİRCAN'a, maddi ve manevi destekleriyle her zaman yanımda olan aileme, çalışmalarım boyunca verdiği tüm desteklerden dolayı kız arkadaşım Nida DEMİR'e ve arkadaşım Mert ÇINAR'a çok teşekkür ederim.

İÇİNDEKİLER

ŞEKİL LİSTESİ

Sayfa

TABLO LİSTESİ

Sayfa

KISALTMALAR

IPS : Intrusion Prevention System
IDS : Intrusion Detection System
TTL : Time to Live
MTU : Maximum Transfer Unit
OSSTMM : Open Source Security Testing Methodology Manual
PHP : Hypertext Preprocessor
TCP : Transmission Control Protocol
HTTP : HyperText Transfer Protocol
HTTPS : Encrypted HyperText Transfer Protocol
SYN : SYNchronize
IP : Internet Protocol
OSI : Open System Interconnection
UDP : User Datagram Protocol
SQL : Structured Query Language
XSS : Cross-site Scripting
UTF : Unicode Transformation Format
BIT : Smallest Data Part
BYTE : 8 BIT
URL : Uniform Resource Locator
PCAP : Packet Capture Data
SMB : Server Message Block
APT : Advanced Persistent Threat

1. GİRİŞ

Saldırganlar ve sistem sahipleri arasındaki çekişme yıllardır devam etmektedir. Saldırganlar politik ve finansal sebepler veya repütasyon amacı ile sistemlere saldırmaktadırlar. Temel amaçları sistemden bilgi çalmak, bilgiyi değiştirmek veya bilgiye erişimi kısıtlamak olacak şekilde bilgi güvenliğinin üç temel unsuru olan gizlilik, bütünlük ve erişilebilirlikten herhangi birini sekteye uğratmaktır. Bunlara önlem olarak sistem sahipleri teknik ve idari tedbirler geliştirir. Bu noktada teknik tedbirlerden biri olan temel güvenlik duvarı ve erişim kısıtlama tedbirleri saldırganlar ile mücadele konusunda yetersiz kalmaktadır. Siber saldırıların önüne geçebilmek amacıyla saldırı tespit ve engelleme sistemleri bulunmuştur. Saldırı tespit ve engeleme sistemlerinin odak noktası haberleşme ve ağ sistemleri üzerinde yapılan sızma teşebbüslerini, servis dışı bırakma saldırılarını ve kritik değişiklikleri takip etmek, raporlamak ve engellemektir. Saldırı tespit ve engelleme sistemleri saldırganlara karşı en büyük engelleyici önlemdir [3], [5]. 2003'te CSI/FBI Computer Crime and Security anketine göre siber saldırıların %99'u saldırı tespit ve engelleme sistemleri tarafından tespit edilmiş ve engellenmiştir [4].

Saldırı tespit ve engelleme sistemleri ilk olarak 1980 yılları başlarında ABD Ulusal Güvenlik Ajansı çatısı altında geliştirilmeye başlanmıştır [1]. Bu teknolojinin çıkma nedeni, yetkili kullanıcı hesap aktivitelerini kayıt altında tutmak, erişilen dizin ve dosyaları kayıt altına almak ve denetim kayıtlarını saklamaktır. Bu çalışma 1986'da Dorothy E. Denning ve onun asistanı olan Peter G. Neumann tarafından ilerletilerek günümüzde kullanılan saldırı tespit ve engelleme sistemlerine benzer bir yapıya doğru evriltilmiştir. Dorothy E. Denning ve Peter G. Neumann'ın yapmış olduğu çalışmada, kullanıcıların ağ üzerinde yaptığı hareketler takip edilmiş, istatistiksel olarak farklı olduğu belirlenen kullanıcı hareketleri raporlanmıştır [7].

Günümüzde yaygın olarak kullanılan saldırı tespit ve engelleme sistemlerinin temellerini ise 1998 yılında Lawrence Berkley Laboratuvarları'ndan çıkan "BRO" saldırı tespit ve engelleme sistemleri oluşturmaktadır [8]. BRO'nun çıkışından 3 ay sonra Libcap paket dinleyicisiyle birlikte çalışan Snort saldırı tespit ve engelleme

sistemi Martin Roesch tarafından duyurulmuştur. Snort, bu tarihten günümüze kadar en yaygın saldırı tespit ve engelleme sistemi olarak kullanılmaktadır. 2007 yılına kadar kullanım sayısı 300.000 adeti geçmiştir [2].

Saldırı tespit ve engelleme sistemlerinin temel kullanım amacı, arkasında bulunan canlı sistemleri, ağ cihazlarını ve kullanıcı cihazlarını korumaktır [12]. Ek olarak, saldırıları tek bir noktadan tespit etme, engelleme ve raporlama avantajı sağlar. Bu nedenle saldırı tespit ve engelleme sistemleri ağ, donanım ve son kullanıcı sistemlerinin vazgeçilmez bir parçası haline gelmiştir [6]. Saldırı tespit ve engelleme sistemleri, çalışma prensibi gereğince, arkasında bulunan sistemi korumak için sisteme gelen ve sistemden giden tüm ağ trafiğini derinlemesine inceler. Bu inceleme sonucuna göre zararlı trafiği algılar, engeller ve raporlar. Saldırı tespit ve engelleme sistemleri tüm ağ trafiğini üzerinden geçirdiği için bazı kaynak kısıtlamaları ve açıkları mevcuttur [12]. Bunlara ek olarak, saldırı tespit ve engelleme sistemlerinde konfigürasyon yapılandırması sistemin nasıl çalışacağını ve nasıl tespit yapacağını belirler. Bu nedenle konfigürasyon hataları da saldırı tespit ve engelleme sistemlerini güçsüz kılan faktörler arasındadır [3]. Saldırganlar yukarıda belirtilen zafiyetleri kullanarak saldırı tespit ve engelleme sistemlerini atlatmaya çalışmaktadırlar. Saldırı tespit ve engelleme sisteminin aşılması, saldırganlar ile sistemler arasındaki güvenlik katmanın ortadan kalkarak sistemlerin savunmasız kalması demektir. Ayrıca saldırı tespit ve engelleme sistemleri, sistemlerde çıkan zafiyetlerin yamaları yayınlanana kadar geçen süre boşluğunu da tolere eder. Saldırı tespit ve engelleme sistemlerinin üzerinde bir zafiyetin imzasını yükleyip aktif hale getirmek, arkasında bulunan tüm sunucuları o zafiyetin yaması ile güncelleştirmekten daha kolay ve hızlıdır [9].

Saldırı tespit ve engelleme sistemlerinin imza tabanlı ve anomali tespiti tabanlı olmak üzere temelde iki farklı türü mevcuttur. Anomali tespiti tabanlı saldırı tespit ve engelleme sistemleri, sistemde belirli bir olağan durum belirler. Belirlemiş olduğu olağan durum dışına çıkan her türlü hareket veya sistemsel iletişimler alarm olarak ekrana yansır ve engellenir. Fakat yüksek oranda yanlış uyarı alınmasından dolayı anomali tabanlı saldırı tespit ve engelleme sistemlerinin kullanımı yaygın değildir [3].

İmza tabanlı saldırı tespit ve engelleme sistemleri ise önceden belirlenmiş bir açıklığın ya da saldırının tespit imzalarını içinde barındırır [3]. Sisteme gelen ve giden her paket incelenerek, bünyesinde bulunan imza kümeleri ile karşılaştırılır, eşleşen bir paket tespit edildiği zaman alarm üretilir ve saldırı engellenir. Bu sayede yanlış uyarı oranı

anomali tabanlı sistemlere göre oldukça düşüktür ve en yaygın olarak kullanılan sistemler olarak karşımıza çıkmaktadır [10].

İmza tabanlı saldırı tespit ve engelleme sistemleri en az yanlış uyarı üreten ve en yaygın kullanılan siber güvenlik ürünü olmasına rağmen geliştirilen teknikler sayesinde atlatılabilir ve arka taraftaki sistemlere ulaşılabilir olduğu görülmektedir. Saldırı tespit ve engelleme sistemleri, atlatma saldırılarına karşı her geçen gün daha da gelişse bile limitasyon, kaynak tüketimi gibi sebepler ve yeni geliştirilen teknikler nedeniyle atlatılamamayı garanti edememektedir [9].

Bu çalışmada, en güncel imza tabanlı saldırı tespit ve engelleme sistemi olan Snort'un, yedi adet atlatma tekniğine karşı başarı oranı ölçülmüştür. Bu yedi adet atlatma tekniği seçilirken günümüzde en sık rastlanan ve bahsi geçen teknikler göz önünde bulundurulmuştur. Bu teknikler sırasıyla:

- TTL atlatma
- MTU ile paket parçalama
- Zaman, ajan adı ve port numarası değiştirme
- Kodlama ve gizleme
- Sahte sağlama kodu
- Dosya başlığı değiştirme
- Dosya ve dizin değiştirme

Bu çalışmada yukarıda sıralanan atlatma teknikleri saldırılar ile birleştirilerek kullanılmıştır. Bu aşamadan sonra, her bir saldırının saldırı tespit ve engelleme sisteminde tetiklediği alarm sayısı ölçülmüştür. Ardından atlatma teknikleri kullanılarak bu saldırılar tekrar saldırı tespit ve engelleme sistemine gönderilmiş ve tetikledikleri alarm sayısı ölçülmüştür. Normal saldırılar ve atlatma teknikleri ile birleştirilmiş saldırıların ürettiği alarm sayıları arasındaki yüzdesel fark, atlatma tekniğinin başarı skorunu oluşturmaktadır. Bunun sonucu olarak, atlatma teknikleri kullanılarak bir saldırı tespit ve engelleme sisteminin ne kadar başarı ile atlatılabildiği ölçülecektir.

Bu araştırma esnasında tüm saldırı teknikleri OSSTMM açık kaynak sızma testi metodolojisine uygun yapılarak raporlanmıştır [26]. Saldırılar keşif/tespit

(scan/recon), istismar (exploit) ve istismar sonrası (post-exploit) olmak üzere temelde üç ana kategoride sınıflandırılmaktadır.

NOT: Saldırı tespit ve engelleme sistemi olarak Snort 2.9.13 kullanılmıştır. 11 Nisan 2019'daki son kayıtlı kural kümesi indirilmiş ve Snort'un içine yüklenmiştir. Sadece saldırı tespit ve engelleme sisteminin tespit ve engelleme yapabileceği saldırılar kullanılmıştır. Web uygulama güvenlik duvarı ya da veri sızıntısı engelleme cihazlarının kapsamına girilmemiştir. Saldırılar karşısında üretilen alarmlar ilk olarak Snort saldırı tespit ve engelleme sisteminin arayüz yazılımı olan Sguil de gözlemlenmeye çalışılmış, ardından sayısal verileri kaydetme zorluğundan ötürü alarmlar CLI üzerinden kayıt edilip kendi yazmış olduğumuz betik üzerinden saydırılarak sonuçlara ulaşılmıştır.

1.1 Tezin Amacı

Bu araştırmada amaç, saldırı tespit ve engelleme sistemlerini atlatma tekniklerinin gerçek bir senaryoda ne kadar etkili ve başarılı olabileceğini ölçmek, hem saldırgan bakış açısı ile sistemlere erişmek hem de savunan bakış açısı ile savunmadaki eksiklikleri tespit etmektir.

1.2 Literatür Araştırması

Literatürde, saldırı tespit ve engelleme sistemlerinin atlatılması ile ilgili yapılmış altı adet başarılı çalışma bulunmaktadır. İncelenmiş olan çalışmalar ve bu tezin konusu olan araştırmamızda ortak olarak, saldırıların saldırı tespit ve engelleme sistemlerine yakalanmadan atlatılması ve sistemin açıklıklarını sömürmek amaçlanmıştır. Bu araştırmalar sırasıyla aşağıda açıklanmaktadır.

1998 yılında ilk olarak bir saldırı tespit ve engelleme sisteminin atlatılabileceği fikri öne sürülmüştür. O zamanki saldırı tespit ve engelleme sistemleri ile günümüzde kullanılanlar arasında büyük farklar olsa da Newsham ve Timothy tarafından yapılan bu çalışmada sistemin insan kaynaklı yanlışlar yüzünden nasıl geçilebildiği gösterilmiştir. Temel amaç saldırı tespit ve engelleme sistemlerinde yapılan temel kurulum ve kurgu hatalarını insalara göstermek ve bunların nasıl kolayca sömürülebileceğini anlatmaktır [28].

2009 yılında Stonesoft firması kendi saldırı tespit ve engelleme sistemlerinde yaptığı yenilikleri test edebilmek için başlattığı çalışma 2010 yılında saldırı tespit ve engelleme sistemlerini atlatmaya yönelik program yazma ve makale çalışmasına dönüşmüştür. Bu betikte 23 adet atlatma tekniği kullanılarak piyasadaki ticari tüm saldırı tespit ve engelleme ürünleri üzerinde test edilmektedir. Bu test sonucunda kendi ürünlerinin en yüksek başarı elde ettiği iddia edilmiş ve test sonuçları yayınlanmıştır. Bu olaydan sonra saldırı tespit ve engelleme sistemleri, üreticileri tarafından bu çalışmada belirtilen 23 adet atlatma saldırısına karşı korunaklı hale getirilmiştir. Ticari ürünlerin yanı sıra Snort, Suricata ve BRO gibi açık kaynaklı ürünlerin de bu saldırıları engelleyebilmek için saldırı tespit motorlarında geliştirmeleri yapılarak güncellemeleri yayınlanmıştır. Stonesoft yapmış olduğu bu çalışmayı Eveader test aracı olarak ücretsiz bir şekilde piyasaya sürmüştür. Ancak kötü amaçlarla kullanımlar yaşanması nedeniyle aracını piyasadan geri çekmek zorunda kalmıştır ve günümüzde erişilebilir durumda değildir [27].

2011 yılında Çin'in National Chiao Tung Üniversitesinde yapılmış olan çalışmada saldırı tespit ve engelleme sistemlerini atlatmaya yönelik beş adet atlatma tekniği ele alınmış, tekniklerin Fortinet, ZyXEL ve Snort ürünlerinde ne kadar alarm ürettiğinin testleri yapılmıştır. Çalışmayı üniversitede araştırma görevlisi olan Tsung-Huan Cheng, Ying-Dar Lin, Yuan-Cheng Lai ve Po-Ching Li isimli akademiyenler icra etmişlerdir. Bu testlerde saldırı tespit ve engelleme sistemleri ilk kurulum halinde bırakılmış, üzerlerinde hiçbir sıkılaştırma veya ayarlama yapılmamıştır. Testlerde kullandıkları atlatma teknikleri günümüzde de etkili olan teknikler olmasına rağmen saldırı tespit ve engelle sistemlerinde konfigürasyon ayarları yapılmadan kullanıldığı için her bir teknik yüksek oranda başarılı olmuş şeklinde sonuçlar elde edilmiştir. Bu nedenle sonuçlar tam olarak gerçeği yansıtamamaktadır. Ek olarak, çalışmada atlatma tekniklerinden korunmak için tavsiyelere de yer verilmiştir. Testlerde örneklendirme ve denemelerin yetersiz olduğu ve sonuçların raporlanması kısmında eksiklikler olduğu düşünülmektedir. Ancak çalışmada bahsedilen atlatma ve korunma yöntemleri günümüzde kullanılan güncel saldırı tespit ve engelleme sistemleri için de hala geçerlidir [29].

Zhengzou Üniversitesinde 2013 yılında yapılan bir çalışmada saldırı tespit ve engelleme sistemlerini atlatma metodolojileri ele alınmış ve atlatma teknikleri ile ilgili bir araç tanımlanmıştır. Tanımlanan aracın durum bazlı nasıl çalışması gerektiği ile

ilgili bilgiler verilmiş ve saldırı tespit ve engelleme sistemlerinden gelen tepkilere göre davranışlarını ve saldırılarını nasıl değiştirmesi gerektiğinden bahsedilmiştir. Ek olarak saldırıları saldırı tespit ve engelleme sistemine gönderirken yapması gereken korelasyon ve normalizasyon algoritmaları da tanımlanmıştır. Betimlenen araç hiçbir zaman yapılamamış olsa da nasıl yapılacağı teorik olarak açıklanmıştır. Geliştirilecek olan aracın sadece IP başlığında, TCP başlığında ve http başlığında değişiklikler yapmasının daha yüksek başarılar getireceği iler sürülmüştür. Geliştirilecek aracın saldırı tespit ve engelleme türlerine göre (anormallik tabanlı ve imza tabanlı) farklı hareket etmesi ve işleyişini değiştirmesi öngörülmüştür [30].

2014 yılında gerçekleştirilen çalışmada saldırı tespit ve engelleme sistemlerinin atlatılmasında kullanılabilecek bir algoritma geliştirilebileceği ve bu algoritmanın Apriori algoritması esas alınarak yapılabileceği iddia edilmiştir. Yapılan çalışmada atlatma teknikleri belirlenerek Apriori algoritması ve alternatif olarak ADABOOST algoritması ile nasıl birleşeceği ve bu sayede saldırı tespit ve engelleme sistemlerinin nasıl atlatılacağı anlatılmıştır. Saldırılar değerlendirilirken 1999 yılında yayınlanan KD-99 saldırı veri tabanı kullanılmış, atlatma teknikleri KD-99 veri tabanındaki saldırılar ile birleştirilip algoritmalar yardımı ile sistemlere nasıl saldırılabileceği değerlendirilmiştir [31].

Saldırganlar açısından saldırı tespit ve engelleme sistemlerini inceleyen bir çalışma da M'hamed Chammem, Mohamed Hamdi ve Tai-Hoon Kim isimli akademisyenler tarafından icra edilmiştir. Çalışmanın amacı, günümüzde en büyük siber tehdit olarak nitelendiren APT (Gelişmiş inatçı tehdit) gruplarının saldırı tespit ve engelleme sistemleri tarafından ne kadar engellenebileceği üzerinedir. Araştırmada APT gruplarının saldırı tespit ve engelleme sistemlerini atlatmak için kullandığı ve kullanabileceği yöntemlere değinilmiş, EVADER aracı ile önceden kaydedilmiş atlatma tekniklerinin ne kadar tehlikeli olabileceğinden bahsedilmiştir. Saldırganların ağaç yöntemi ile kurban sistemlerde saldırı tespit ve engelleme sistemlerini geçerek nasıl ilerleyebileceği incelenmiştir [32].

Yapmış olduğumuz araştırmaya en fazla benzerlik gösteren çalışma ise 2016 yılında Helsinki Üniversitesinde Särelä, Kyöstilä, ve Kiravuo isimli araştırma görevlileri tarafından yapılmıştır. Bu araştırmada belirlenen sekiz adet atlatma tekniği piyasadaki en popüler on beş cihaz üzerinde test edilmiş ve sonuçları yayınlanmıştır. Bu araştırmada bahsi geçen bazı atlatma teknikleri yapmış olduğumuz araştırmada da yer

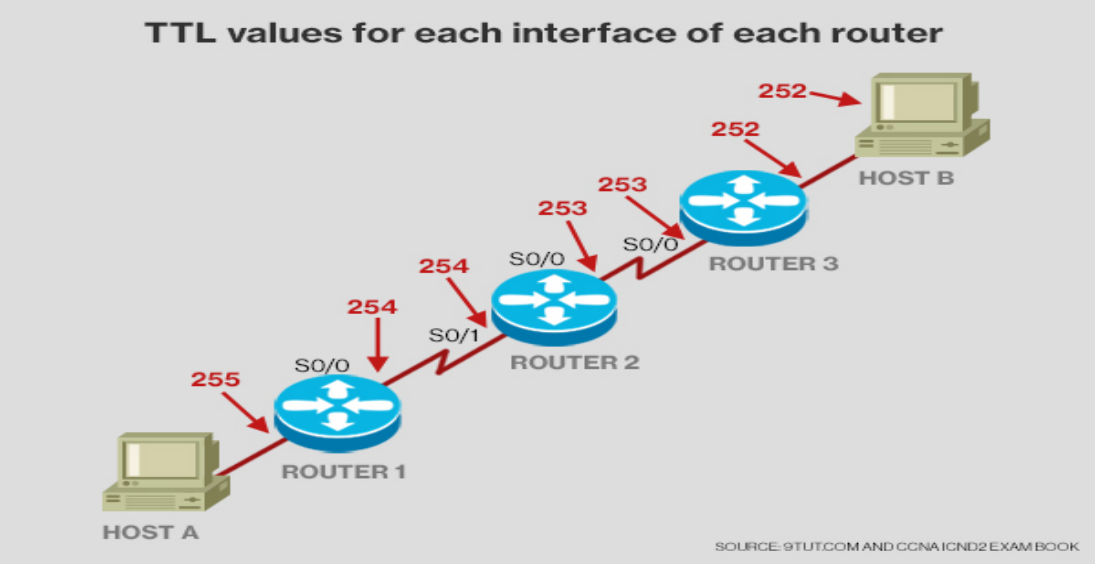
almaktadır. Araştırma esnasında açık kaynak kodlu hiçbir ürün kullanılmamıştır bu sebeple marka bağımlı bir çalışma olmuştur [33].

2. ATLATMA TEKNİKLERİ

Atlatma tekniklerinin uygulanma amacı saldırı tespit ve engelleme sistemlerini atlatarak arkasında koruduğu gerçek sistemlere ulaşmaktır. Her bir atlatma tekniğinin farklı bir uygulama yöntemi ve uygulanabileceği saldırı metodolojileri bulunmaktadır. Bu bölümde atlatma saldırılarının neler olduğundan bahsedilecektir.

2.1 TTL Atlatma

TTL değeri OSI ağ katmanının 3. katmanında bulunan bir paket başlık değeridir. Bu değer, ağ paketi yönlendiricilerinden (router) ve güvenlik cihazlarından geçerken "1" azaltılır. TTL değerinin "0" değerine ulaşması durumunda paket, alınan sistem tarafından düşürülür. Bu değerin var olma sebebi ağda dolaşan paketlerin yollarını kaybetmeleri durumunda sistemi gereksiz yere yormamasıdır. Aksi takdirde yolunu kaybetmiş ve ilerlemeye çalışan bir ağ paketi, ağda sonsuz döngüye sebebiyet verebilir. Bu da ağı erişilemez ve çalışamaz bir hale sokar. Bu ihtimallerden kaçınmak için her bir ağ paketinin TTL değeri olması zorunludur. TTL değeri ile saldırı tespit ve engelleme sistemlerini atlatmanın iki yöntemi mevcuttur [13].



Şekil 2.1 : TTL değerinin paket ağda ilerledikçe eksilme durumu.
(https://www.cbronline.com/what-is/what-is-ttl-4992801/2/)

Birinci yöntemde saldırı tespit ve engelleme sistemlerinin kaynaklarını verimli kullanmak amacıyla TTL değeri düşük olan paketlerin işlenmediği veya bu paketlerde imza kümesine bakılmadığı varsayılmaktadır. Sistemin çalışma mantığı, hedef

makineye ulaşamayacak veya ulaşma ihtimali düşük olan paketler ile sistemin kaynaklarını tüketmemek, daha verimli çalışmasını sağlamaktır [35]. Saldırı tespit ve engelleme sisteminde paketin hedefe ulaşma ihtimalinin düşük olması önceden belirlenen eşik değerine göre değişkenlik göstermektedir. Eşik değeri saldırı tespit ve engelleme sisteminin konfigürasyon dosyasında yer alan bir parametredir. TTL atlatma tekniğinin birinci yöntemini test etmek için TTL değeri sıfır değerine yaklaştırılır. Ancak, TTL değeri sıfır değerine doğru yaklaştırıldıkça paketlerin kurban makineye ulaşma ihtimalinin de düşüyor olmasından dolayı bu yöntemin gerçek bir senaryoda başarıya ulaşması güçtür [13].

İkinci yöntem mantıksal olarak birinci yöntemin tam tersidir. Bu yöntemde gönderilen paketlerin TTL değerine bakılmaksızın saldırı tespit ve engelleme sisteminde imza işletildiği varsayılır. Buradaki senaryoda saldırgan paketleri parçalar ve araya anlamsız ek paketler ekler. Bu paketlere son kurban sisteme ulaşamayacak kadar düşük TTL değerleri verilir. Saldırı tespit ve engelleme sistemi bu paketleri birleştirdiğinde zararsız bir bütün gibi dururken, anlamsız paketler iletilirken TTL değerleri sıfıra ulaşacağı için düşürüleceklerdir [25]. Düşürülen paketler kurban makineye hiçbir zaman ulaşamayacağı için sadece parçalanmış zararlı paketler hedefe ulaştırılır. Kurban makine kendine ulaşan paketleri birleştirdiğinde saldırı başarılı olacaktır. Örneğin; saldırgan kurban makineye bir “virüs” göndermeyi amaçlıyor olsun. Saldırı paketi “virüs” şekilde gönderildiği zaman paket saldırı tespit ve engelleme sistemine takılacak ve saldırı başarısız olacaktır. Saldırgan ilk önce paketi “v”-“i”-“r”-“ü”-“s” olarak parçalara ayırır, sonra paketlerin arasına “x” gibi bir veya birden çok anlamsız ve saldırıyı temiz gibi gösterecek paketler ekler. Sonradan eklenen saldırı paketlerinin TTL değerleri kurban makineye ulaşamayacak kadar düşük bir değerde gönderilir. Bundan sonra saldırı paketleri “v”-“x”-“i”-“r”-“x”-“ü”-“s” şeklini alır. Oluşturulan yeni paketler dizini saldırgan tarafından hedef makineye gönderilir. Öncelikli olarak saldırı tespit ve engelleme sistemine gelen paketler değerlendirilir ve herhangi bir saldırı imzasıyla örtüşmediği için paketler kurban makine tarafına herhangi bir alarm üretilmeden gönderilir. “x” paketleri kurban makineye iletilirken yolda düşürüleceği için kurban makine “v”-“i”-“r”-“ü”-“s” şeklindeki paket kombinasyonunu alır ve birleştirir, bu sayede atlatma saldırısı başarılı olur [44].

2.2 MTU İle Paket Parçalama

MTU değeri OSI ağ katmanının 3. katında bulunan bir değerdir. Bu değer, ağ paketinin içinde barındırabileceği maksimum byte miktarını belirler. Bir ağ paketinde gönderilmek istenen değer MTU değerinden büyük ise paket parçalanarak gönderilir. Bu işleyiş saldırganlar tarafından sömürülerek saldırı tespit ve engelle sistemlerini atlatma tekniği olarak kullanılabilmektedir [14].

MTU değerini değiştirerek paket parçalama yapılması saldırganlar tarafından uzun zamandır kullanılan bir yöntemdir. MTU ile paket parçalama atlatma tekniğinin temel amacı, ağ paketlerinin parçalanış şeklinden ötürü saldırı tespit ve engelleme sisteminin korelasyon motorunun bu paketleri düzgün sıraya dizememesini sağlamaktır. Burada bozulmak istenen düzen saldırı tespit ve engelleme sisteminin ara bellek eleman miktarının sayısını aşarak sistemi paketlerin bütününe bakamaz hale getirmektir. Saldırı tespit ve engelleme sisteminin ara bellek eleman miktarının düşük olmasının amacı servis dışı bırakma saldırılarına karşı cihazı korunaklı hale getirmektir [15].

2.3 Zaman, Ajan Adı ve Port Numarası Değiştirme

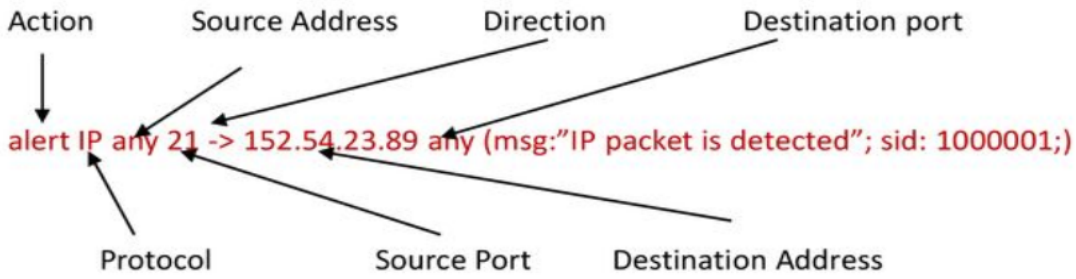
Belirli parametrelerin değiştirilmesi ile saldırı tespit ve engelleme sisteminin atlatılması, saldırı aşamasının keşif bölümünde uygulanabilen bir atlatma tekniğidir. Burada ki temel amaç saldırı tespit ve engelleme sisteminin baktığı ana saldırı bulgularını farklılaştırarak sistemi atlatmaktır. Ek olarak zaman, ajan adı ve port numarasına göre imzalar ve desenler (pattern) mevcuttur. Bu parametreler ile oynamak saldırı tespit ve engelleme sisteminin korelasyon ve tespit motorlarını yanıltmaktadır.

Zamanı değiştirme tekniği zamanı yavaşlatma olarak kullanılmaktadır. Bu teknikte saldırgan, saldırı paketleri arasında geçen zamanı arttırarak saldırı tespit ve engelleme sistemini yanıltmayı amaçlar. Saldırganlar bu atlatma tekniğini port tarama, ip tarama ve açık servis tespiti(banner grabbing) işlemleri esnasında kurban sistem hakkında bilgi toplarken kullanmaktadırlar. Örnek olarak; bir saldırgan açık portları keşfetmek istediğinde kurbanın IP adresinin tüm portlarına SYN isteği gönderir. Kurban makine bu isteğe karşılık olarak açık portlarından saldırgana cevap döner. Arada bir saldırı tespit ve engelleme sistemi var ise, sistem gelen SYN paketlerini arkaya geçirirken de bağlantının devamını kontrol edebilmek adına paketleri ara belleğinde tutar. Belirli bir sayıda SYN isteği belirli bir zaman aralığında tek bir IP adresinden geliyorsa saldırı

tespit ve engelleme sistemi bunu saldırı olarak algılar ve engeller. Bu noktada saldırgan saldırı tespit ve engelleme sisteminde ara bellekte görülen kaç adet SYN paketinden sonra alarm oluştuğunu tespit edebilirse, sistemin ara bellekte tutma süresi boyunca alarm oluşturacak sayının bir eksiği kadar paket gönderir. Belirli aralıklarla sistem ara belleğini boşaltmak zorunda kalacağı için bu saldırıyı zaman ile oynayarak yaptığında sistemi aşmış ve kurban makineye saldırısını gerçekleştirmiş olacaktır. Burada ki kritik nokta saldırganın göndereceği saldırı paketleri arasındaki zamanı sisteme göre ayarlaması ve sistemde alarm oluşturmayacak kadar yavaşlatmasıdır. Bu atlatma tekniğinin en zor kısmı ise saldırganın saldırıyı tamamlayabilmesi için fazla zamana ihtiyaç duymasıdır. Ek olarak saldırgan amacına ulaşabilmek için uzunca bir süre boyunca da kaynaklarının bir kısmını saldırıya harcamak zorunda kalacaktır [16].

Ajan adı, HTTP/HTTPS protokollerinde bulunması zorunlu bir parametredir. Ajan adı değiştirme atlatma tekniği HTTP/HTTPS protokollerinde uygulanan bir tekniktir. Saldırganlar, saldırı araç ve programlarını varsayılan ajan adı ile kullandığında saldırı tespit ve engelleme sistemleri sadece ajan adına bakarak saldırıyı kolayca tespit edebilir. Saldırı tespit ve engelleme sistemleri güncellenen imza kümeleri sayesinde her türlü saldırı aracının ajan bilgisini bulundurmaktadırlar. Saldırgan bu ismi değiştirmediği taktirde, saldırı aracıyla yasal bir istek yapsa bile sistem bunu saldırı olarak algılayıp engelleyecektir. Örnek olarak; Nikto web tarama aracı bir web sitesinin tüm alt dizinlerini ve oradaki dosyaları saldırgana gösterebilir. Bu keşif işlemi sırasında kurban sunucuya HTTP GET istekleri yaparak belirli parametreler ile dizinleri ve altında bulunan her türlü obje ve dizini çağırır, isteklere dönen cevaplara göre de çıktı üretir. Bu noktada isteklerini “Mozilla/5.0(Nikto/2.1.6)” ajan adıyla gerçekleştirir. Saldırı tespit ve engelleme sistemleri bu ajan adını gördükleri anda paketin geri kalanına ya da isteklerin oluşmasına bakmaksızın alarm üretir ve engelleme yapar. Bu noktada saldırgan sistemi atlatmak için normal bir web tarayıcının ajan adı ile Nikto’nun ajan adını değiştirirse, sistem saldırıyı tespit etmek için farklı yollar bulmak zorunda kalacaktır. Bu atlatma tekniği bir saldırı aşamasında yapması kolay ve etkili bir yoldur. Kullanılan saldırı aracı ajan adının değiştirilmesine izin vermiyorsa (bazı saldırı araçları kötü kullanımı engellemek için ajan adlarının değiştirilmesine izin vermez) saldırgan bir vekil sunucu yardımı ile bu atlatma saldırısını gerçekleştirebilir [17].

Farklı bir port kullanmak ya da diğer ismiyle port numarasını değiştirmek demek saldırı paketlerinde OSI ağ katmanın 4. katında değişiklikler yapmak demektir. Saldırı tespit ve engelleme sisteminde çoğu kural port bazlı yazılır ve tespit motoru da saldırıları kurallar sayesinde port bazlı olarak algılar. Bu noktada port numarası değiştirmek saldırı tespit ve engelleme sistemini atlatmanın etkili bir yönetimidir. Bu atlatma tekniği uygulama yöntemi açısından önceden belirtilen atlatma tekniklerinden farklıdır. Bu teknikte saldırgan kendi saldırdığı portu değil kurban makinenin portlarını değiştirmelidir. Çünkü saldırgan saldırı esnasında herhangi bir portunu kullanabildiğinden sistem burada kurban makinenin portunu baz alır. Örnek olarak; saldırgan kurban makineden kendisine bir kabuk açacağı zaman HTTP gibi zaten kendisine veri akışı olan bir port kullanarak sistemi atlatabilir. Çünkü saldırı tespit ve engelleme sistemi 80 portundan dışarı yönlü yapılan veri akışını normal karşılayacağı için saldırganın bu portu kritik verileri kaçırmak için kullandığını tespit edemeyecektir [18]. Saldırı tespit ve engelleme sistemleri için port numarasının önemi ve nasıl kullanıldığı Şekil 2.2'de gösterilmektedir.



Şekil 2.2 : Örnek bir Snort kuralı. (https://slideplayer.com/slide/13774900/)

2.4 Kodlama ve Gizleme

Kodlama tekniği ilk olarak verinin bozulmadan karşı tarafa iletilebilmesi için kullanılmak üzere ortaya çıkmış bir yöntemdir. Bu yöntem özellikle web protokolünde, kullanıcıların farklı dillerde kullandıkları sistemler arası iletişim sağlanırken kayıp veya hataların engellenmesi için ortaya çıkmıştır. Gönderilen url veya içerik istekleri diller arasında çevirilirken kodlama olmadan iletilirse kayıp veya bozulmaya uğrayabilirler. Bu yöntem bir soruna çözüm olarak geliştirilmiş olsa da, saldırganların elinde güçlü bir silaha dönüşebilmektedir.

Saldırganlar kodlama tekniğini saldırı tespit ve engelleme sistemlerini atlatmak için kullanmaktadırlar. Burada saldırganların fark edilmeden duvarları aşabilmesinin

nedeni saldırı tespit ve engelleme sistemlerinin tüm kodlama yöntemlerini bilemiyor olmasından kaynaklanmaktadır. Eğer saldırgan, saldırısının yük bölümünü saldırı tespit ve engelleme sisteminde mevcut olmayan bir kodlama yöntemi ile kodlarsa, sistemin bu saldırıyı yakalaması neredeyse imkansız bir hal almaktadır [21,22]. Ancak, sistemde bu kodlamayı açabilecek bir kod çözme yöntemi mevcut ise sistem saldırıyı tespit edebilir. Saldırı tespit ve engelleme sistemleri, kendi içlerinde bir eşleyici bulundurur. Bu eşleyici sayesinde sistem kodlanmış olarak gelen paketlerin nasıl açılacağını anlar ve paketlerin kodunu çözer. Kod çözme işleminden sonra da saldırı tespit motoru gelen pakete göre imza kümesini tarar [5].

Saldırganlar kodlama atlatma tekniğini kullanmak için sadece sistemde olmayan kodlama metodlarını kullanmak zorunda değildir. Ek olarak, paketler birden çok kez kodlanırsa, kod çözme işlemi zorlaşacak veya yapılamayacak olmasından dolayı saldırı tespit ve enhelleme sistemi saldırıya karşı bir alarm üretemeyecektir. Saldırı tespit ve engelleme sistemlerinin paketleri bir defadan fazla kodlandığında kod çözme işlemini yapamama nedenleri sınırlı kaynaklarını tüketmemek veya yüzlerce kez kodlanmış ve amacı sistemi servis dışı bırakmaya yönelik olan saldırılara mağruz kalmamaktır [22].

Günümüzde en yaygın kullanılan kodlama yöntemi UTF-8 kodlama yöntemidir. Bu yöntem ile "/etc/shadow" şeklinde bir dizini kodlamak istersek sonuç "\x2f\x65\x74\x63\x2f\x73\x68\x61\x64\x6f\x77" şeklinde görünecektir. Sistem kodlanmış olan "\x2f\x65\x74\x63\x2f\x73\x68\x61\x64\x6f\x77" paketini tekrar "/etc/shadow" şekline dönüştüremez ise, alarm üretmesi de olanaksız hale gelmektedir [24].

Gizleme tekniği kodlama yöntemi ile çok fazla benzerlik gösterse de en büyük farkı bu methodun yasal olarak hiçbir kullanımı mevcut değildir. Gizleme yöntemi, kodlama yönteminin zamanla kötü kullanım için evrilmiş bir halidir. Bu tekniğin ortaya çıkışındaki amaç, saldırganların güvenlik cihazlarından saklanabilmesidir. Örnek vermek gerekirse, bu teknikte "/etc/shadow" gibi bir dizini shikataganai gizleme yöntemi ile gizlersek ortaya "fzÐYžrJ{ßÇði¿□ŒxþŠ¤>r&%Ïžã(ákC"-Ý|Î_‰e T€" şeklinde bir metin çıkmaktadır. Gizleme yönteminin en güzel yanı saldırı kodunu herhangi bir kod açma işlemi gerektirmeden işlemci seviyesinde anlamlı hale getirilebilmesidir. Bu

saldırıların saldırı tespit ve engelleme sistemleri tarafından yakalanması oldukça zordur çünkü bu metni kod açma işlemini gerçekleştirmeden işlemcisinde işlemesi gerekmektedir. Bu tarz bir davranış sistemin kendisinin ele geçirilmesine olanak verebilir [23].

2.5 Sahte Sağlama Kodu

Bu tekniğin çıkış noktası OSI ağ katmanlarındaki bir durumu sömürmek üzerinedir. OSI ağ katmanında TCP paketlerinin karşı makineye ulaşırken bir değişikliğe uğrayıp uğramadığının kontrolü sağlama değeri ile yapılır. Paketin içindeki değerinin byte cinsinden toplam değeri sağlama bölümüne yazılır. Saldırganlar sağlama değerini sömürerek sistemi atlatmayı başarmaktadır [44].

Saldırı tespit ve engelleme sistemleri genellikle kaynaklarını verimli kullanmak adına gelen paketlerin sağlama değerine bakmayacak şekilde konfigüre edilmiştir. Bu konfigürasyon kurulum sonrasında değiştirilebilse de çoğunlukla kaynak tüketimi gerekçesi ile aktif hale getirilmezler. Saldırganlar bu zafiyeti sömürmek için sisteme sağlama değeri yanlış olan paketlerle karıştırılmış olarak saldırı paketlerini gönderir ve sistemi atlatırlar. Örnek olarak, saldırganın elinde ”virüs” şeklinde bir zararlı olduğunu varsayalım. Bu noktada saldırgan, paketleri “v”-”i”-”r”-”ü”-”s” şeklinde parçalar ve ardından paketlerin arasına rastgele sıra ile “x” paketleri ekler. Burada önemli olan nokta sonradan eklenen “x” paketlerinin sağlama değerlerinin hatalı oluşudur. Saldırı paketleri kurban makineye gönderilir. Saldırı tespit ve engelleme sistemi paketleri analiz etmek için alıp birleştirdiğinde karşısına “v”-”x”-”i”-”r”-”x”-”ü”-”s” şeklinde birleşen bir ağ paket kümesi çıkar. Bu paketler hiçbir saldırı imzası ile eşleşemeyeceği için sistemde herhangi bir alarm tetiklemeden arka tarafa geçmeyi başarırlar. Paketler kurban makineye ulaştığı zaman, kurban makine paketlerin sağlama değerlerini inceler ve yanlış değerde olan paketleri işlemeden düşürür. Bu noktada kurban makinenin birleştirdiği paketler tekrar “v”-”i”-”r”-”ü”-”s” şeklini almış olur ve saldırgan amacına ulaşmış olur. Bu atlatma tekniği parçalama saldırısına benzerlik gösterse de kurban makinenin tepkisinden dolayı ismi sahte sağlama kodu tekniği olarak adlandırılmıştır [43].

2.6 Dosya başlığı değiştirme

Dosyaların başlık bilgileri, işletim sistemlerinin dosyanın tipini belirlemek için başvurduğu bir bilgidir. Her türlü dosya tipinde dosya başlık bilgisi bulunmaktadır. İşletim sistemi, uzantıya bakmaksızın dosya başlık bilgisine göre dosyayı nasıl açacağını veya işleyeceğini belirler. Bu bilgiye "magic number" denir. "Magic number" adli analiz ve zararlı yazılım araştırmalarında da öncelikli olarak incelenen bilgiler arasıda yer alır. Örneğin; "MZ" ile başlayan bir dosyanın tipi ".exe", "JAR" ile başlayan bir dosyanın tipi de ".java" şeklindedir [36].

Saldırı tespit ve engelleme sistemleri, kendilerine bir dosya geldiğinde zamanı ve kaynaklarını verimli kullanabilmek adına sadece dosya tipine özel olarak yazılmış imzalara bakar ve işletir. Her dosya tipinin istismar edilebilir zafiyeti farklıdır. Bu nedenle de saldırı tespit ve engelleme sistemleri dosyaların tiplerine göre farklılık gösteren zafiyetleri dosya tiplerine göre ayırır. Saldırganlar da dosya tipi üzerine işletilen bu savunma mekanizmasını atlatmak için dosya başlık bilgisini değiştirip sistemleri atlatmaktadırlar [36].

Dosya başlık değiştirme atlatma tekniği günümüzde pek çok siber güvenlik sistemini atlatma tekniği olarak bilinmektedir. Çünkü sadece saldırı tespit ve engelleme sistemleri değil, diğer siber güvenlik sistemleri de dosyaları tipine göre işleterek bir sonuca varmaktadırlar. Örneğin, saldırgan istismar kodunun tipine karar verdikten sonra, kurbana paketi gönderir. Ancak kurban başlık bilgisi değiştirilmiş paketi otomatik olarak çalıştıramayacaktır. Bu nedenle saldırganın istismar kodunun kendisinin çalıştırması veya dosyayı gerçek başlık bilgisine çevirmesi gerekmektedir. Bu noktada, saldırgan kurban makinede dikkat çekmemek için çalıştırılmasında sorun olmayacak bir dosya seçer ve istismar kodunu dosya ile birleştirerek tek dosya haline getirir. Ardından saldırı tespit ve engelleme sistemini atlatabilmek için dosyanın başlık bilgisini değiştirir ve kurbana gönderir. Saldırı tespit ve engelleme sistemi gelen paketin dosya başlık bilgisine göre uygun imzalara bakar ve alarm üretemeden kurban sisteme gönderir. Bu noktada atlatma tekniği diğer tekniklerden farklılık göstermektedir. Çünkü burada saldırgan, saldırı paketleri kurban makineye geldikten sonra saldırı paketlerinin dosya başlık bilgilerini tekrar ilk haline gelecek şekilde düzenlemek zorundadır. Aksi takdirde dosya çalıştırılamayacak olduğu için kurban makine saldırıdan etkilenmeyecektir. Tekniğin gerçekleşebilmesi için saldırganın

etkileşimi şarttır. Bu tarz atlatma tekniklerini saldırganlar kurban makine ve sistemlerde düşük yetkilere sahip kullanıcılar ele geçirdiklerinde yetki yükseltmek için veya içeride bir makineyi ele geçirdikten sonra diğerlerine yatayda yayılabilmek için kullanmaktadırlar [3].

Dosya başlığı değiştirme tekniğinin bir diğer kullanım alanı ise web saldırılarıdır. Örnek olarak, web sayfalarının yükleme bölümlerinde dosya tipi kısıtlaması bulunmaktadır. Bu noktada saldırganlar yükleme kısıtlamasına uygun olarak değiştirilen dosya başlık bilgisine sahip saldırı dosyasını sisteme yükler. Web sitelerinde genel olarak php dili kullanıldığı için, saldırganlar dosya başlık bilgisi değiştirilmiş dosyaların içerisine kendilerine ters ya da direk kabuk açabilecekleri php kodları yerleştirebilir. Web sunucu kodlarında girdi denetimi yapılmamış ise web sitesi dosyayı işlemeye çalıştığı aşamada farkına varılmadan php kodlarını çalıştırır ve saldırgan amacına ulaşmış olur. Bu saldırıda dosyalar yüklenmesi gereken yere istenilen dosya başlık bilgisinde yükleneceği için saldırı tespit ve engelleme sistemi dosyaya alarm üretmeyecektir [5].

Dosya başlığı değiştirme tekniği siber saldırının istismar sonrası aşamasında kullanılır. Keşif ve istismar aşamalarında herhangi bir saldırı ile birleştirilemez.

2.7 Dosya ve Dizin Değiştirme

Saldırı tespit ve engelleme sistemine göre risk teşkil eden özel dosya ve dizinler sistemlerde önceden tanımlı olarak mevcuttur. Bu dosya ve dizinlere dışarıdan ulaşılması istila emaresi olarak değerlendirilir. “/etc/shadow” ve “/etc/passwd” bu dizin ve dosyalara birer örnektir. Eğer bir saldırgan “/etc/shadow” dizinine erişmeye veya bu dizinde bir işlem yapmaya çalışırsa, saldırı tespit ve engelleme sistemleri bu işlemi saldırı olarak değerlendirir, alarm üretir ve engeller. Saldırgan kritik bilgileri değiştirmek veya çalmak için sistemi alarm ürettirmeden atlatması gerekmektedir [38].

Bu teknikte saldırgan değiştirmek veya çalmak istediği dosya veya dizinleri başka bir uzantıya kopyalayabilir ya da isimlerini değiştirebilir. Bu sayede saldırgan ulaşmak istediği kritik yerleri ve dosyaları saldırı tespit ve engelleme sistemine takılmadan çalabilir veya değiştirebilir. Örneğin; “/etc/passwd” dizinini ve içindekileri “/usr/tmp” dizinine koplayalar ise, artık “/usr/tmp” dizini ile işlem yapacağı için sisteme

takılmadan saldırısını tamamamyabilecektir. Saldırganın "/etc/passwd" dizinini ve içindekileri "/usr/tmp" dizinine koplayalarken sisteme yakalanmaması için ise işlemini sistemin bakmadığı bir kanaldan gerçekleştirmesi gerekmektedir [38].

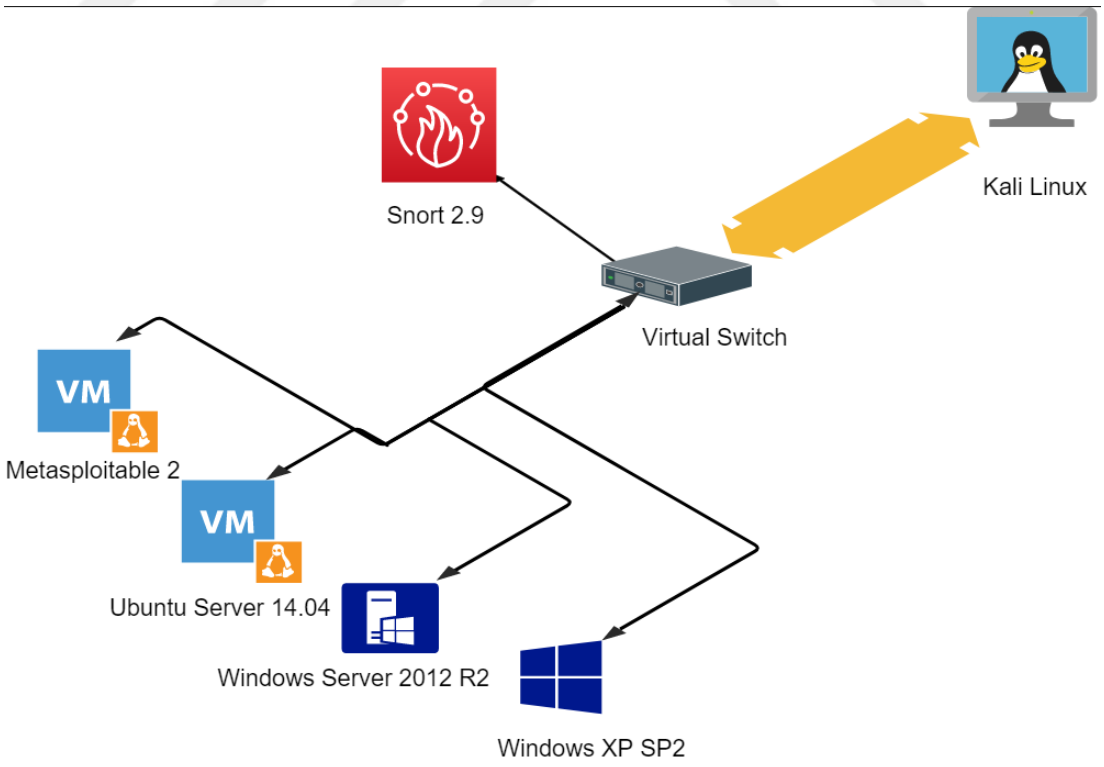
Dosya ve dizin değiştirme atlatma tekniği siber saldırının istismar sonrası aşamasında kullanılmaktadır. Keşif ve istismar aşamalarında herhangi bir saldırı ile birleştirilemez.

3. ATLATMA TEKNİKLERİ DENEME ORTAMI

Bu bölümde atlatma tekniklerini test etmek için oluşturulan test ortamında kullanılan sistemler, donanımlar, betik ve programlar ve testlerin hangi metodolojilere göre hangi amaçlarla yapıldığı anlatılacaktır.

3.1 Test Ortamı

Tüm testler sanallaştırma ortamı kullanılarak yapılmıştır. Sanallaştırma ortamı olarak VMware ürün ailesinden ESXI 6.5 platformu kullanılmıştır. Saldırgan makine olarak bir adet Kali Linux 2018.3 versiyonu kullanılmıştır. Saldırıları iletmesi için bir adet virtual switch ve kurban makine olarak da dört adet makine kullanılmıştır. Kurban makineler sırasıyla Metasploitable2, Ubuntu server 14.04, Windows Server 2012 ve Windows XP SP2 şeklindedir. Saldırılar tüm port aynalama alınarak saldırı tespit ve engelleme sistemi olarak kullanılan Snort 2.9.13 versiyonu yüklü bir makineye gönderilmiştir. Snort sistemi Ubuntu server 18.04 LTS üzerine kurulmuştur. Sistem sadece saldırı tespit etme sistemi olarak çalıştırılmıştır. Test ortamının mantıksal topolojisi Şekil 3.1'de gösterilmektedir.



Şekil 3.1 : Saldırı ve atlatma tekniklerinin test ortamın mantıksal topolojisi.

3.2 Kullanılan Betik ve Programlar

Test süreci boyunca çok sayıda betik ve program kullanılmıştır. Kullanılan betik ve programlar OSSTMM açık kaynak sızma testi metodolojisine göre anlatılacaktır.

OSTTMM açık kaynak sızma testi metodolojisine göre saldırının üç ana aşaması mevcuttur. Bu aşamalar sırasıyla; keşif (kurban makine veya sistemler hakkında aktif ve/veya pasif olarak bilgi toplama, açıklık ve port tarama), istismar (kurban makine veya sistemde hatalı bir portokol, kod veya işlemler üzerinden istismar) ve istismar sonrası (kurban makine veya sistemlerde yetki yükseltme, bilgi çalma, değiştirme ve kalıcılık) olarak tanımlanmaktadır.

Saldırılar ilk aşamada hiçbir atlatma yöntemi kullanılmadan icra edilmiş, trafik saldırı tespit ve engelleme sistemine gönderilmiş ve alarmlar not edilip kayıt altına alınmıştır. İkinci aşamada, atlatma teknikleri kullanılabilecek saldırılar ile birleştirilerek kurban makinelere gönderilmiş ve tüm trafik saldırı tespit ve engelleme sistemine gönderilerek alarmlar tekrar not edilip kayıt altına alınmıştır. Her iki aşamada da tüm saldırılar tam pcap olarak kaydedilmiştir. Saldırılar ve atlatma tekniklerinde, saldırının veya atlatma tekniğinin kurban makinelerde istenilen etkiyi yaratıp yaratmadığı gözlemlenmiş, başarısız saldırı ve atlatma teknikleri test sonuçlarına yansıtılmamıştır.

Keşif aşamasında kurban makineler hakkında bilgi toplama işlemleri gerçekleştirilmiştir. Keşif saldırıları sırasıyla ip tarama, port tarama, servis bulandırma ve açık servis tespiti şeklindedir. Keşif saldırıları uygulanırken Nmap, Burp Suite, Dirb, DNSLookUp, Nikto, Enum4Linux ve SMTPEnum betik ve programları kullanılmıştır. Bu betik ve programlar TTL atlatma, MTU ile paket parçalama ve zaman, ajan adı ve port numarası değiştirme atlatma tekniklerinde kullanılmıştır.

İstismar aşamasında yapılan saldırılar kod yükleme, kod çalıştırma, derin dizin gezinme, dosya içerme, kaba kuvvet, SQL enjeksiyonu ve XSS saldırıları şeklindedir. Bu saldırılarda kullanılan betik ve programlar JohnTheRipper, WFUZZ, Hydra, NetCat, BeefXSS, Burp Suite, SQL Map, Metasploit, Veil, Msfvenom, Havij ve Hping3 şeklindedir. Bu betik ve programlar TTL atlatma, MTU ile paket parçalama, zaman, ajan adı ve port numarası değiştirme, kodlama ve gizleme, sahte sağlama kodu ve dosya başlığı değiştirme atlatma teknikleri ile kullanılmıştır. Bu aşamadaki saldırı ve atlatma teknikleri betik ve programlara ek olarak el ile de test edilmiştir.

İstismar sonrası aşamasında yapılan saldırılar yetki yükseltme, aşamalı saldırı(pivoting), veri çalma ve kalıcılık saldırılarıdır. Bu metodda NetCat, Metasploit, Veil, Msfvenom, Powershell Empire, Mimikatz, Sshutle ve Lasagne betik ve programları kullanılmıştır. Bu betik ve programlar dosya ve dizin değiştirme, dosya başlığı değiştirme ve zaman, ajan adı ve port numarası değiştirme atlatma teknikleri ile kullanılmıştır. Bu aşamadaki saldırı ve atlatma teknikleri betik ve programlara ek olarak el ile de test edilmiştir.

Yukarıda bahsedilen betik ve programlara ek olarak el ile yapılan testler, bu araştırma için özel olarak geliştirmiş olduğumuz betikler, Fragroute, TCPreWrite ve TCPReplay programları aracılığıyla yapılmıştır.

3.3 Snort Konfigürasyonu

Saldırı tespit ve engelleme sistemi olarak açık kaynak kodlu Snort programının en güncel versiyonu olan 2.9.13 versiyonu kullanılmıştır. Bu programın seçilmesindeki sebep en yaygın kullanılan açık kaynak kodlu saldırı tespit ve engelleme sistemi olmasıdır. Snort'un sitesine kayıt yaptırarak community rules ve ek olarak emerging threat kuralları da indirilip eklenmiştir. Tüm saldırı ve atlatma tekniklerinin kesin başarmını görmek adına sistemde tüm imzalar aktif hale getirilmiştir. Ek olarak Barnyard2, BASE 1.4, Pulledpork ve OpenAppID araçları kurulmuştur. Ağ konfigürasyonunda Home_net ve External_net bölümleri any (tüm ip adreslerini kapsayacak şekilde) olarak tanımlanmıştır [37, 39].

Testte Kullanılan Saldırı	Saldırının Kısaltması
Agresif Port ve Zafiyet Tarama	APZT
Agresif Port, Zafiyet Tarama ve Linux Betiklerini Çalıştırma	APZT-L
Agresif Port, Zafiyet Tarama ve Windows Betiklerini Çalıştırma	APZT-W
Ping Atmaksızın Port Tarama	NPP
EternalBlue İstismar Kodu	EIK
PHP Ters Kabuk ile MeterPreter	PHPRS
Rastgele SQL Enjeksiyonu	RSQLI
Nmap Top 10.000 Port Tarama	NT10
Hydra ile FTP Kaba Kuvvet	HFBF
El ile Kod Çalıştırma	MCE
Dirbuster ile Alt Dizin ve Dosya Taranması	DIRB
SQLMap ile SQL Enjeksiyonu Taraması	SMS
SQLMap ile Kör SQL Enjeksiyonu Taraması	BSMS
Hydra ile SSH Kaba Kuvvet	HSBF
SMTP Servisi Dökümü	SMTE
Nikto ile Zafiyet ve Dizin Keşfi	NVDS
"/etc/shadow" ve "/etc/passwd" Dizinlerinin Çağırılması	CPC

Testte Kullanılan Saldırı	Saldırının Kısaltması
Apache HTTP Üzerinden PHP İstismarı	AHPE
XSS Veri Yükü Sırasıyla Gönderimi	XSP
TCP Ters Kabuk Alma	TRSH
PHP Ters Kabuk	PHPS
Dosya İçerme	FIA
Tomcat Yönetici İstismar Kodu	TAE
Bulandırma	FUZ
Linux Zafiyet Kodu	LEC
PHP Zafiyet Kodu	PEC
CMD Unix Zafiyet Kodu	CEC
Linux Shell Zafiyet Kodu	LSEC
SQL Enjeksiyonu Denemesi(100 adet)	SIT
"/etc/passwd" İçerisindeki Dosyanın Çekilmesi	EPFC
"/etc/apache/apache.conf" İçerisindeki Dosyanın Çekilmesi	EAFC
"/etc/resolve.conf" İçerisindeki Dosyanın Çekilmesi	ERFC
"/proc/self/environ" İçerisindeki Dosyanın Çekilmesi	PSFC
"/var/www/phpinfo.php" İçerisindeki Dosyanın Çekilmesi	VWFC

Tablo 3.1 : Saldırı isimleri ve kısaltmaları.

4. ATLATMA TEKNİKLERİNİN SONUÇLARI

Bu bölümde, test edilen yedi adet atlatma tekniğinde gözlemlenen başarı oranları ve atlatma tekniklerinin hangi saldırılar ile birlikte kullanıldığı aktarılacaktır. Testlerdeki başarı oranı saldırının ilk halinin kaç alarm tetiklediğinin ölçümünden sonraki alarm sayısı ile atlatma tekniği ile kaç alarm tetiklediğinin ölçümünden sonraki alarm sayısının arasındaki yüzdesel farka göre belirlenmiştir. Başarısız olan saldırılar ve atlatma teknikleri başarı oranı tablosuna etki etmemiştir.

Test sonuçları tablolarda gösterilmiş, isim bölümleri kısaltmalar kullanılarak adlandırılmıştır.

4.1 TTL Atlatma Sonuçları

TTL atlatma tekniğinde toplamda oniki adet deneme yapılmıştır. TTL atlatma tekniği keşif ve istismar aşamalarındaki saldırılar ile kullanılmıştır. İstismar kısmında baz alınan dört adet saldırı bulunmaktadır. Bu saldırılar sırasıyla kurban makineye gönderilen ms08_017 Eternalblue olarak bilinen SMB zafiyetinin istismar kod parçacığı ve php'de zafiyeti sömüren içerisine meterpreter ile ters kabuğa konulmuş kod parçacığıdır [18-20]. Keşif bölümünde ise sekiz adet deneme bulunmaktadır. Keşif bölümündeki saldırılar Nmap ile ve istismar bölümündeki istismar kod parçacıkları Metasploit üzerinde hazırlanmıştır. Keşif saldırılarında, saldırı paketlerinin ekranda aynı sonuçları gösterdiği gözlemlenmiştir. İstismar saldırılarında ise ters kabuk atlatma tekniği ile kurban makineye sızma işlemi başarılı olmuştur.

Tablo 4.1'de sonuçları gösterilen TTL atlatma tekniklerinin ilk dört adedi TTL değeri 1 olarak belirlenerek uygulanmış, ikinci dört denemede ise TTL değeri 2 olarak belirlenerek uygulanmıştır. TTL değerinin 1 ve 2 olarak kullanıldığı atlatma denemeleri büyük oranda benzer sonuçlar gösterse de TTL=1 olduğu durumda negatif başarı ile de karşılaşılmıştır. Saldırganların keşif saldırılarında TTL atlatma teniği kullandığında yüksek başarılar kazanabildiği gözlemlenmiştir.

Saldırı İsmi	Kullanılan Atlatma Tekniği	Saldırı Alarm Sayısı	Atlatmalı Alarm Sayısı	Başarı Oranı
APZT	TTL=1	193	2	99%
APZT	TTL=2	232	401	-73%
APZT-L	TTL=1	236	2	99%
APZT-L	TTL=1	12	11	8%
APZT-W	TTL=2	193	2	99%
APZT-W	TTL=1	232	83	64%
NPP	TTL=1	236	106	55%
NPP	TTL=2	12	10	17%

Tablo 4.1: Keşif saldırısı ile birleştirilen TTL atlatma test sonuçları.

Saldırı alarm sayısı kolonu baz alınan saldırının saldırı tespit ve engelleme sisteminde tetiklediği alarm sayısını, atlatmalı alarm sayısı kolonu atlatma tekniği saldırıyla birleştirildiğinde tetiklenen alarm sayısını göstermektedir.

Tablo 4.2'de EternalBlue saldırısı ile birleştirilen TTL atlatma denemeleri başarı sonuçları gösterilmektedir. Sonuçları ilk satırda gösterilen denemede saldırı paketlerinin arasına kurban makineye ulaşamayacak TTL değerleri eklenerek test yapılmıştır. İkinci denemede tüm paketlere düzensiz (rastgele) TTL değerleri verilerek test gerçekleştirilmiştir. Son denemede ise TTL değeri tüm paketlerde 1 olacak şekilde ayarlanarak test gerçekleştirilmiştir. Tablo 4.2'de görülen sonuçlara göre, hedefe ulaşamayan TTL değerli paketler eklenerek yapılan TTL atlatma tekniğinin en yüksek başarıya ulaştığı ve saldırganlar açısından en büyük avantaj kazandıran test olduğu gözlemlenmiştir.

Saldırı İsmi	Kullanılan Atlatma Tekniği	Saldırı Alarm Sayısı	Atlatmalı Alarm Sayısı	Başarı Oranı
EIK	Ulaşmayan Paketli TTL	189	0	100%
EIK	Düzensiz TTL	189	10	95%
EIK	TTL=1	189	10	95%

Tablo 4.2: EternalBlue istismar kodu ile birleştirilen TTL atlatma tekniği test sonuçları.

Tablo 4.3'de PHP ters kabuk saldırısı ile birleştirilmiş atlatma denemesinde kurban makineye ulaşamayan TTL paketlerinin eklendiği deneme sonuçları gösterilmektedir. Test sonucuna göre %100 başarıya ulaşılamamış, saldırı tespit ve engelleme sistemi

atlatma tekniği ile birleşik saldırıyı 1 adet alarm ile yakalamıştır. EternalBlue istismarına göre php istismarında TTL atlatma tekniğinin uygulanması daha düşük başarı ile sonuçlanmıştır.

Saldırı İsmi	Kullanılan Atlatma Tekniği	Saldırı Alarm Sayısı	Atlatmalı Alarm Sayısı	Başarı Oranı
PHPRS	Ulaşmayan Paketli TTL	9	1	89%

Tablo 4.3: PHP ters kabuk saldırısı ile birleştirilen TTL atlatma tekniği test sonucu.

Yukarıdaki tablolarda görüldüğü üzere en yüksek başarı %100 iken, en düşük başarı negatif olarak sonuçlanmıştır.

4.2 MTU ile Paket Parçalama Sonuçları

MTU ile paket parçalama atlatma tekniği 6 farklı senaryo ile test edilmiştir. Senaryolar keşif ve istismar aşamalarındaki saldırılar ile birleştirerek yapılmıştır. Senaryolarda paketler düzenli ve düzensiz olarak farklı MTU değerleri ile parçalanarak test edilmiştir. Düzenli MTU değeri ile parçalanan paketlerde sabit düşük bir MTU değeri seçilerek testler icra edilmiştir. Düzensiz MTU değeri ile parçalanan paketlere birbirinden farklı olacak şekilde MTU değerleri verilerek test icra edilmiştir. Sabit MTU değeri kullanılan atlatma denemelerinde TCP/IP protokolü gereğince MTU değerleri yalnızca dört sayısının katları olabilmektedir.

Yapılan testlerde saldırı tespit ve engelleme sisteminin atlatma tekniklerini ara bellekte düzgün veya tam olarak sıralayamaması amaçlanmıştır.

Port tarama ile belirli bir düzende olan MTU değerleri Tablo 4.4'de gösterilmektedir. İlk dört satırda NPP ile ifade edilen saldırı baz alınarak MTU değerleri 80 ile 8 arasında değiştirilerek paketler parçalanmıştır. Başarı oranı dört denemede de %17 olarak gözlemlenmiştir. Bu sonuca dayanarak ping atmaksızın tüm portları tarama saldırısıyla MTU değerleri bir değerde sabit kaldığı sürece saldırının başarı oranının değişmediği ve düşük olduğu gözlemlenmiştir. Son denemede ise aynı NPP saldırısı kullanıldığında ancak MTU değeri düzensiz (rastgele) verildiğinde başarı oranının %58'e yükseldiği görülmüştür. Sonuç olarak, keşif saldırılarında saldırganın daha yüksek başarı gösterebilmesi için düzensiz MTU değeri kullanması gerekmektedir.

Saldırı İsmi	Kullanılan Atlatma Tekniği	Saldırı Alarm Sayısı	Atlatmalı Alarm Sayısı	Başarı Oranı
NPP	MTU=80	12	10	17%
NPP	MTU=40	12	10	17%
NPP	MTU=20	12	10	17%
NPP	MTU=80	12	10	17%
NPP	Düzensiz MTU	12	5	58%

Tablo 4.4: Keşif saldırısıyla birleştirilen MTU ile paket parçalama atlatma tekniği başarı sonuçları.

Agresif port ve zafiyet tarama saldırısında her bir pakete farklı bir MTU değeri verilerek, saldırı paketlerinin farklı boyutlarda parçalanması sağlanmıştır. Test sonuçları Tablo 4.5'te gösterilmektedir. İlk denemede MTU değerleri 4 ile 20 arasında verilirken, ikinci denemede MTU değerleri 20 ile 40 arasında düzensiz (rastgele) olarak verilmiştir. Her iki denemede de başarı oranının %95 olduğu gözlemlenmiştir. Sonuç olarak, bu değer aralıklarında MTU değerinin artırılması saldırı tespit ve engelleme sistemlerinin atlatılma başarısını değiştirmemektedir.

Saldırı İsmi	Kullanılan Atlatma Tekniği	Saldırı Alarm Sayısı	Atlatmalı Alarm Sayısı	Başarı Oranı
APZT	Düzensiz MTU-1	193	9	95%
APZT	Düzensiz MTU-2	193	9	95%

Tablo 4.5: Keşif saldırısıyla birleştirilen MTU ile paket parçalama atlatma tekniğinde düzensiz MTU değeri denemeleri başarı sonuçları.

SQL enjeksiyonu (RSQLI) ve php ters kabuk alma (PHPRS) paketleri 20 ile 40 değerleri arasında düzensiz MTU değerleri ile parçalanarak kurban makineye gönderilmiştir. Test sonuçları Tablo 4.6'da gösterilmektedir. SQL enjeksiyonu saldırısı ile birleştirilen MTU ile paket parçalama atlatma tekniği %95 başarı oranına ulaşmışken, php zafiyeti ile ters kabuk alma ile birleştirilen MTU ile paket parçalama atlatma tekniği %78 başarı oranına ulaşabilmiştir. Sonuç olarak, istismar aşamasında MTU ile paket parçalama atlatma tekniğinin SQL enjeksiyonu ile birleştirilmesinin, istismar kodu ile birleştirilmesine oranla saldırı tespit ve engelleme sistemlerinin atlatılmasında yüksek oranda başarı sağladığı gözlemlenmiştir.

Saldırı İsmi	Kullanılan Atlatma Tekniği	Saldırı Alarm Sayısı	Atlatmalı Alarm Sayısı	Başarı Oranı
RSQLI	Düzensiz MTU	189	10	95%
PHPRS	Düzensiz MTU	9	2	78%

Tablo 4.6: Düzensiz MTU değerleri ile SQL enjeksiyonu ve php zafiyeti ile ters kabuk alma saldırılarının birleşik deneme sonuçları.

Sonuçları Tablo 4.5'in ikinci satırında ve Tablo 4.6'nın ilk satırında görülen denemelerde MTU değerleri 20 ile 40 arasında düzensiz olarak seçilerek MTU ile paket parçalama atlatma tekniği ile birleştirilmiştir. Bu aynı şartlar altında; tekniğin hem keşif aşamasında hem de istismar aşamasında eşit başarı gösterdiği gözlenlenmiştir.

MTU ile paket parçalama tekniğinde gözlemlenen en yüksek başarı %95 iken, en düşük başarı oranı %17 olarak gözlemlenmiştir.

4.3 Zaman, Ajan Adı ve Port Numarası Değiştirme Sonuçları

Zaman değiştirme ile saldırı tespit ve engelleme sistemini atlatma tekniğinde keşif saldırı paketlerinin kurban makinelere belirli aralıklarla yavaşlatılarak gönderimi sağlanmıştır. Saldırılar saldırı tespit ve engelleme sisteminin ara belleğinde belirli bir süre tutulacağından dolayı ne kadar yavaş gönderilirse o kadar az alarm tetikleneceği varsayılmaktadır.

Atlatma tekniği kullanılırken Nmap aracının zamanlaması kullanılmıştır. Saldırıda ilk olarak Nmap default ayarda (-T5) paket gönderim hızı ile çalıştırılmış ve saldırı NT10 olarak kaydedilmiştir ve tabloda gösterilmiştir. Bu saldırı en yaygın kullanılan ilk 10.000 portun SYN taraması ile keşfetmedilmesi şeklinde bir saldırıdır. Nmap aracının zamanlama fonksiyonu -T parametresi ile değiştirilebilmektedir. En hızlı tarama -T5 iken en yavaş tarama -T0 olarak gerçekleştirilmektedir [45].

Tablo 4.7'de ilk satırdan son satıra doğru gidildikçe zamanın yavaşlatılarak yapıldığı denemelerin sonuçları görülmektedir. Tablo 4.7'nin son satırında ise en yavaş gönderim hızı olan -T0 gönderimine ek olarak paketler daha da yavaşlatılabilmek amacıyla saldırgan makinenin ara belleğinde bekletilerek gönderilmiştir. Bu sonuçlara göre zaman değiştirme ile atlatma tekniğinin kullanım yoğunluğuyla test sonuçları arasında doğrudan bir oran kurulamasa bile, en yavaş şekilde tarama yapmanın

saldırgan açısından en az alarm tetikleyen ve zamanlama kategorisinde başarısı en yüksek olan deneme olduğu gözlemlenmiştir.

Saldırı İsmi	Kullanılan Atlatma Tekniği	Saldırı Alarm Sayısı	Atlatmalı Alarm Sayısı	Başarı Oranı
NT10	T4 Zamanına Yavaşlatma	72	9	88%
NT10	T3 Zamanına Yavaşlatma	72	19	74%
NT10	T2 Zamanına Yavaşlatma	72	19	74%
NT10	T1 Zamanına Yavaşlatma	72	19	74%
NT10	T0 Zamanına Yavaşlatma	72	17	76%
NT10	T0 Zamanına Yavaşlatma ve Ara Bellekte Bekletme	72	2	97%

Tablo 4.7: Zaman değiştirme ile birleştirilmiş port ve versiyon tarama saldırısı ile birleşik deneme sonuçları.

En kısa taramanın tamamlanması 1.2 saniye sürerken, en uzun tarama 14 saat 38 dakika 46 saniye sürmüştür. Zaman değiştirme atlatma tekniğinde gözlemlenen en yüksek başarı %97 iken, en düşük başarı oranı %74 olarak gözlemlenmiştir.

Ajan adı değiştirme atlatma tekniği basit ama saldırganların görünmezliğini arttıran bir tekniktir. Ajan adı değiştirme atlatma tekniğinin denemelerinde yüksek başarılar gözlemlense de negatif başarılar da görülmüştür.

Ajan adı değiştirme testinde kullanılan araçlar Hydra, DirBuster, SQLMap, SMTPEnum ve Nikto'dur. Ek olarak, el ile WEB üzerinden kod çalıştırma (command execution) saldırısı test edilmiş, ardından, Burb Suite (Vekil Sunucu) ile araya girilerek gönderilen isteğin ajan adı parametresi değiştirilmiştir. Tüm saldırıların ajan adı olarak en güncel Chrome Web tarayıcısının ajan adı kullanılmıştır.

Tablo 4.8'de yukarıda bahsedilen araç ve saldırılar kısaltmalarıyla gösterilmektedir. Her bir saldırının ajan adı kısmına Chrome tarayıcısının güncel ajan adı verilerek atlatma tekniği uygulanmış ve başarı oranları yan sütunda gösterilmiştir. Test sonuçlarına göre ajan adı değiştirme tekniği başarı oranı, kullanılan saldırı ve araca göre farklılık göstermektedir. Bazı denemelerde ise atlatma başarısı saldırı

başarısından daha düşüktür. Bu teknik yalnızca Hydra, Dirbuster, SQLMap ve Nikto araçlarıyla kullanıldığı zaman başarıya ulaşmıştır. Ajan adı değiştirme atlatma tekniğinin kullanımı saldırganlara her zaman avantaj sağlamadığı testlerde gözlemlenmiştir.

Saldırı İsmi	Kullanılan Atlatma Tekniği	Saldırı Alarm Sayısı	Atlatmalı Alarm Sayısı	Başarı Oranı
HFBF	Chrome Ajan Adı Kullanılması	2817	2	100%
MCE	Chrome Ajan Adı Kullanılması	0	170	-
DIRB	Chrome Ajan Adı Kullanılması	9201	1503	84%
SMS	Chrome Ajan Adı Kullanılması	465	3301	-610%
BSMS	Chrome Ajan Adı Kullanılması	3776	2492	34%
HSBF	Chrome Ajan Adı Kullanılması	2	2	0%
SMTE	Chrome Ajan Adı Kullanılması	0	5	-
NVDS	Chrome Ajan Adı Kullanılması	25096	935	96%

Tablo 4.8: Ajan adı değiştirme atlatma tekniği deneme sonuçları.

Ajan adı değiştirme atlatma tekniğinde en yüksek başarı istatistiksel hesaplama nedeni ile ilk denemedir ve oranı %99,5'un üzerinde olduğu için %100'e yuvarlanmıştır. En düşük başarı ise negatiftir ve tüm testler arasındaki en büyük negatif değişim bu test sırasında kaydedilmiştir.

Port numarası değiştirme atlatma tekniği, saldırgan açısından zor ama etkili bir tekniktir. Bu teknikte önemli olan nokta, saldırganın içeriden dışarıya çıkartacağı bilgi ve dosyalarda farklı portlar kullanma zorunluluğu olması veya saldırıyı farklı bir porta göndermesi gerekmesidir.

Tablo 4.9'un ilk üç satırında kısaltması CPC olan "/etc/shadow" ve "/etc/passwd" dizinlerinin farklı üç adet porttan çağırılması sonucu elde edilen başarı oranları gösterilmektedir. Sonuçlara göre aynı saldırının farklı portlardan gerçekleştirilmesinin saldırı tespit ve engelleme sistemlerinin atlatılmasında farklı başarılar gösterdiği gözlemlenmiştir. Tablonun üçüncü, dördüncü ve beşinci satırlarında farklı saldırıların aynı porttan çağırılması dendndiğinde elde edilen sonuçlar gösterilmiştir. Bu sonuçlara göre, çağırılan portun yanı sıra kullanılan saldırının da alarm tetiklemede başarı oranına etkisi olduğu görülmektedir. Saldırganın, saldırı tespit ve engelleme sistemini atlatabilmek için hem kullanacağı saldırıya hem de hangi portu seçeceğine dikkat etmesi gerekmektedir.

Saldırı İsmi	Kullanılan Atlatma Tekniği	Saldırı Alarm Sayısı	Atlatmalı Alarm Sayısı	Başarı Oranı
CPC	Port 4444 Kullanıldı	91	10	89%
CPC	Port 3367 Kullanıldı	91	10	89%
CPC	Port 8080 Kullanıldı	91	17	81%
AHPE	Port 8080 Kullanıldı	5	0	100%
XSP	Port 8080 Kullanıldı	23	4	83%

Tablo 4.9: Port numarası değiştirme atlatma tekniği test sonuçları.

Port numarası değiştirme atlatma tekniğinde gözlemlenen en yüksek başarı %100 iken, en düşük başarı oranı %83 olarak gözlemlenmiştir. Son üç testte 8080 portu kullanılarak farklılıklar gözlemlenmek istenmiştir.

4.4 Kodlama ve Gizleme Sonuçları

Kodlama ve gizleme atlatma teknikleri yapılan tüm testler arasında açık ara en yüksek başarı oranlarının gözlemlendiği testtir.

Kodlama testlerinde UTF-8 kullanılmış ve pakete 3 kez ve 5 kez kodlama uygulanmıştır. Gizleme testlerinde ise Shikatagai, XorDynamic ve Shellelfbinary yöntemleri kullanılmıştır.

Tablo 4.10'da TCP ters kabuk alma saldırısı (TRSH) baz alınarak gerçekleştirilen gizleme tekniğinin test sonuçları gösterilmektedir. İlk üç satırda görülen sonuçlar Shikataganai atlatma tekniğinin sırasıyla 3, 7 ve 21 kez uygulanarak denenmesinden elde edilmiş sonuçları göstermektedir. Sonuçlarda %100 başarı gözlemlenmiştir. Dördüncü satırdan altıncı satıra kadar gösterilen sonuçlarda XorDynamic gizleme tekniği sırasıyla 3, 7 ve 21 kez uygulanmıştır. Sonuçlarda %100 başarı gözlemlenmiştir. Tablonun son iki satırında ise ShellElfBinary gizleme tekniği sırarıyla 3 ve 7 kez uygulanması sonucu elde edilen sonuçları gösterilmektedir. Sonuçlarda %100 başarı gözlemlenmiştir. Tablo 4.10'da gösterilen test sonuçlarından anlaşılacağı üzere saldırı tespit ve engelleme sistemleri gizleme tekniğine karşı savunmasızdır.

Saldırı İsmi	Kullanılan Atlatma Tekniği	Saldırı Alarm Sayısı	Atlatmalı Alarm Sayısı	Başarı Oranı
TRSH	3 Kez ShikataGanai	5	0	100%
TRSH	7 Kez ShikataGanai	5	0	100%
TRSH	21 Kez ShikataGanai	5	0	100%
TRSH	3 Kez XorDynamic	5	0	100%
TRSH	7 Kez XorDynamic	5	0	100%
TRSH	21 Kez XorDynamic	5	0	100%
TRSH	3 Kez ShellElf	5	0	100%
TRSH	7 Kez ShellElf	5	0	100%

Tablo 4.10: Shikatanagai, XorDynamic ve ShellElf gizleme tekniği ile birleştirilen tcp ters kabuk saldırısı test sonuçları.

Tablo 4.11'de php sömürülerek kabuk alma saldırısı baz alınarak gerçekleştirilen kodlama tekniğinin testlerinin sonuçları gösterilmektedir. Tabloda sırasıyla UTF-8 kodlama tekniğinin 3 ve 5 kez uygulanması sonucu elde edilen sonuçlar gösterilmektedir. Kodlama tekniği uygulandığı durumda %92 başarı elde edilse de 3 ile 5 kez tekniğin uygulanması durumunda sonucun değişmediği gözlemlenmiştir. Sonuçlara göre saldırı tespit ve engelleme sistemleri saldırı paketleri ne kadar kodlansa da içerisindeki saldırı paternini tespit edebildiği ve alarm üretebildiği gözlemlenmiştir.

Saldırı İsmi	Kullanılan Atlatma Tekniği	Saldırı Alarm Sayısı	Atlatmalı Alarm Sayısı	Başarı Oranı
PHPS	3 Kez UTF-8	12	1	92%
PHPS	7 Kez UTF-8	12	1	92%

Tablo 4.11: UTF-8 kodlama tekniğinin üç kere ve yedi kere kullanıldığı denemelerde elde edilen test sonuçları.

Test sonuçlarında gözlemlenen en yüksek başarı %100 iken, en düşük başarı oranı %92 olarak gözlemlenmiştir. Bu test en yüksek başarılı denemelerin yapıldığı atlatma teniğidir.

4.5 Sahte Sağlama Kodu Sonuçları

Sahte sağlama kodu atlatma tekniği, tüm atlatma teknikleri arasında uygulaması en zor atlatma tekniğidir. Paketlerin aralarına dizi numarasını bozmadan sağlama değeri değişik bir paket eklemek oldukça güçtür.

Test esnasında gerçekleştirilen saldırılarda Hping3, Wfuzz, Metasploit, Msfvenom ve Meterpreter kullanılmıştır. Ek olarak, el ile de bir saldırı gerçekleştirilmiştir. Sahte sağlama kodu atlatma tekniğini test etmek için kaydedilen pcap dosyaları kullanılmıştır. Pcap dosyalarının üzerlerinde değişikliğe gidilerek tekrar gönderimi sağlanmıştır. Saldırı paketlerinin arasına yanlış sağlama değerine sahip paketler eklenmiştir. Aynı saldırının baz alındığı durumlarda, aralara eklenen paketlerde değişiklikler yapılarak testler icra edilmiştir.

Tablo 4.12'de sonuçları gösterilen ilk denemede APZT kısaltmalı saldırı ile yapılan her üç saldırı paketi arasına bir adet fazladan paket eklenmiş, FIA kısaltmalı saldırı ile yapılan ilk denemede her beş paket arasına bir adet fazladan paket ekleme yapılmış, ikinci denemede her iki saldırı paketi arasına bir adet fazladan paket eklenmiştir. TAE kısaltmalı saldırı ile yapılan denemede sırasıyla dokuz, yedi, beş ve iki adet fazladan paket eklemesi yapılmıştır. Tablodaki son denemede bulandırma saldırısı test edildiğinden dolayı, saldırının içerisinden rastgele paketlerin doğrulama değerleri bozulmuştur. Test sonuçlarına göre her bir paketin arasına sahte sağlama kodu değerli saldırı eklemek saldırgana %100 başarı kazandırmakta, daha seyrek sahte sağlama kodlu paket ekleme ise saldırı tespit ve engelleme sistemini tam olarak atlatamamaktadır.

Saldırı İsmi	Kullanılan Atlatma Tekniği	Saldırı Alarm Sayısı	Atlatmalı Alarm Sayısı	Başarı Oranı
APZT	Sahte Sağlama Kodu Her 3'lü de	193	17	91%
FIA	Sahte Sağlama Kodu Her 5'li de	6	5	17%
FIA	Sahte Sağlama Kodu Her 2'li de	6	0	100%
TAE	Sahte Sağlama Kodu Her 9'lu de	6	6	0%
TAE	Sahte Sağlama Kodu Her 7'li de	6	2	67%
TAE	Sahte Sağlama Kodu Her 5'li de	6	2	67%
TAE	Sahte Sağlama Kodu Her 2'li de	6	0	100%
FUZ	Sahte Sağlama Kodu Rastgele	686	82	88%

Tablo 4.12: Sahte sağlama kodu test sonuçları.

Test sonuçlarında gözlemlenen en yüksek başarı %100 iken, en düşük başarı oranı %0 şeklinde gözlemlenmiştir. Testlerde negatif başarı gözlemlenmemiştir.

4.6 Dosya Başlığı Değiştirme Sonuçları

Dosya başlığı değiştirme atlatma tekniği, iki aşamalı olarak değerlendirilir. İlk aşamada saldırgan dosyanın başlık bilgisini değiştirerek kurban makineye yollar. Bu aşama saldırgan açısından uygulanması kolay bir aşamadır. İkinci aşamada ise saldırgan hedefe ulaşan saldırı paketinin başlık bilgisini original haline döndürmek zorundadır. Saldırganın ikinci aşamayı gerçekleştirebilmesi için içeride belirli yetkilere sahip bir arka kapısı olması gerekmektedir. Dosya başlığı değiştirme atlatma tekniği ikinci aşamanın gereksinimlerinden ötürü saldırgan açısından uygulanması zor bir teknik olarak değerlendirilmektedir.

Test esnasında gerçekleştirilen saldırılarda kullanılan araçlar Metasploit, SQLMap, Nmap ve Meterpreter şeklindedir. Saldırı veri yüklerinin başlık bilgileri atlatma testlerinde değiştirilerek test edilmiştir.

Tablo 4.13'ün ilk dört satırında, yukarıda anlatılan dört sadırının yük bölümü alınarak dosya başlık kısmı var olmayan bir dosya başlığı şeklinde değiştirilmesi şeklinde gerçekleştirilen testlerin başarı sonuçları gösterilmektedir. Tablonun beşinci satırından

onuncu satırına olan kısmında ise birden fazla belirlenen SQL enjeksiyonu saldırılarının http yük kısımlarına farklı başlık bilgileri getirilmesi sonucu elde edilen sonuçlar gösterilmektedir. Getirilen başlık bilgileri beş ile yedinci satırlar arasındaki denemelerde rastgele ve anlamsız olarak seçilmiş, sekiz ile onuncu satırlar arasındaki denemelerde ise mevcut olan farklı dosya başlıkları seçilmiştir (.jar, Mz ve PDF). Son denemede ise nmap keşif saldırısının paketlerinin veri yüklerine rastgele dosya başlıkları verilmiştir. Son denemenin farkı, normalde yük kısmı olmayan bir saldırı yapılırken yük eklenerek bu yükün de yanlış başlık değerlerine sahip olduğu durum test edilmiştir. Tablo sonuçlarına göre en yüksek başarı oranları istismar kodunun dosya başlık bilgisinin değiştirildiğinde gözlemlenmektedir. En düşük başarılar ise SQL enjeksiyonu saldırısının başlık bilgilerini değiştirdiğinde görülmüştür. Bu durumda saldırı tespit ve engelleme sistemlerine yapılan atlatma deemelerinde SQL enjeksiyonu kullanılan durumlarda saldırganların dosya başlığı değiştirme tekniğini uygulamasının verimsiz olacağı gözlemlenmiştir. Yükü olmayan saldırılarda ise başarı kazanıldığı ama mutlak atlatmanın gerçekleşemediği görülmüştür.

Saldırı İsmi	Kullanılan Atlatma Tekniği	Saldırı Alarm Sayısı	Atlatmalı Alarm Sayısı	Başarı Oranı
LEC	Başlık Bilgisini Değiştirme	2	0	100%
PEC	Başlık Bilgisini Değiştirme	2	1	50%
CEC	Başlık Bilgisini Değiştirme	1	0	100%
LSEC	Başlık Bilgisini Değiştirme	2	0	100%
SIT	Başlık Bilgisini Değiştirme-1	1276	1171	8%
SIT	Başlık Bilgisini Değiştirme-2	1276	1214	5%
SIT	Başlık Bilgisini Değiştirme-3	1276	1138	11%
SIT	Başlık Bilgisini Değiştirme-4	1276	1268	1%
SIT	Başlık Bilgisini Değiştirme-5	1276	960	25%
SIT	Başlık Bilgisini Değiştirme-6	1276	1088	15%
NT10	Başlık Bilgisini Değiştirme	72	10	86%

Tablo 4.13: Dosya başlığı değiştirme tekniği testi başarı sonuçları.

Test sonuçlarında gözlemlenen en yüksek başarı %100 iken, en düşük başarı oranı %1 şeklinde gözlemlenmiştir. Testlerde negatif başarı gözlemlenmemiştir.

4.7 Dosya ve Dizin Değiştirme Sonuçları

Dosya ve dizin değiştirme atlatma tekniği, uygulaması kolay olmakla birlikte saldırganın içeriye müdahale etmesi gereksinimi olması nedeniyle gerçek ortamda hayata geçirilmesi zor bir tekniktir.

Test esnasında gerçekleştirilen saldırılarda kullanılan araçlar Netcat, Bash kabuğu ve Burbsuite şeklindedir. Testler esnasında web protokolü kullanılmış. Saldırılar ve atlatma teknikleri web üzerinde çağırılmış Netcat ile alınmış bir Bash kabuğu vasıtasıyla da istenilen değişiklikler saldırı tespit ve engelleme sisteminin bakmadığı bir kanaldan gerçekleştirilmiştir.

Atlatma teknikleri test edilirken saldırı tespit ve engelleme sisteminde imzası bulunan 5 adet kritik yol ve dosya seçilmiştir. İlk beş denemede dosya veya yollar /tmp/ dizinine kopyalanarak çağırılmıştır. İkinci beş denemede ise dosyalar, isimleri değiştirilerek çağırılmıştır. Test sonuçları Tablo 4.14'de gösterilmektedir.

Tablo 4.14'te gösterilen sonuçlar, sırasıyla, yukarıda anlatılan saldırıların ilk beş testte farklı dizine kopyalanması ve son beş testte dosya adının değiştirilmesi şeklinde icra edilmiştir. Testler aynı saldırılar baz alınarak yapılmış ve yüksek başarılar göstermiştir. Saldırganların dosya veya dizini değiştirerek saldırı tespit ve engelleme sistemlerini atlatabileceği test sonuçlarında görülmektedir. Saldırma tekniği uygulanırken dosyanın veya dizinin değiştirilmesi arasında bir başarı farkı görülmemektedir.

Saldırı İsmi	Kullanılan Atlatma Tekniği	Saldırı Alarm Sayısı	Atlatmalı Alarm Sayısı	Başarı Oranı
EPFC	Farklı Dizin(/tmp/)	2	1	50%
EAFC	Farklı Dizin(/tmp/)	1	0	100%
ERFC	Farklı Dizin(/tmp/)	2	0	100%
PSFC	Farklı Dizin(/tmp/)	2	0	100%
VWFC	Farklı Dizin(/tmp/)	6	3	50%
EPFC	Farklı Dosya Adı	2	1	50%
EAFC	Farklı Dosya Adı	1	0	100%
ERFC	Farklı Dosya Adı	2	0	100%
PSFC	Farklı Dosya Adı	2	0	100%
VWFC	Farklı Dosya Adı	6	2	67%

Tablo 4.14: Dosya ve dizin değiştirme denemeleri tekniği başarı sonuçları.

Test sonuçlarında gözlemlenen en yüksek başarı %100 iken, en düşük başarı oranı %50 şeklinde gözlemlenmiştir. Testlerde negatif başarı gözlemlenmemiştir.

5. ATLATMA TEKNİKLERİNE GÖRE YAPILABİLECEK İYİLEŞTİRMELER

Test skorlarından elde edilen verilere göre saldırı tespit ve engelleme sistemlerinde konfigürasyon ve ayarlamalar ile yapılabilecek iyileştirmeler mevcuttur. Bu iyileştirmeler sayesinde sistemin atlatma tekniklerine karşı başarı oranı artacaktır.

TTL atlatma tekniğini önlemek için saldırı tespit ve engelleme sisteminde konfigürasyon değişikliği ile sonuçları iyileştirmek mümkün değildir.

MTU ile paket parçalama atlatma tekniğini önlemek için saldırı tespit ve engelleme sistemi kurulduktan sonra konfigürasyon ayarlarında varsayılan arabellek büyüklüğü değerinin daha büyük bir değer ile değiştirilmesi gerekmektedir. Bu çözüm tam olarak atlatma tekniğinden koruma sağlamasa da MTU ile paket parçalama atlatma tekniği ile karşılaşıldığı zaman saldırı tespit ve engelleme sisteminin üreteceği alarm sayısının daha yüksek ve daha doğru olmasını sağlayacaktır [40].

Zamanlama atlatma tekniğini önlemek için saldırı tespit ve engelleme sisteminde arabellekte tutulan elemanların varsayılan tutulma zamanını ve varsayılan arabellek büyüklüğünü arttırmak gereklidir. Her ne kadar tutulma zamanı ve arabellek büyüklüğü arttırılsa da paketler saldırgan tarafından daha da yavaşlatılıp alınan önlemler yine de atlatılabilir. Ancak bahsedilen iki adet önlemin alınması saldırganların zamanlama atlatma tekniğini uygulamasını daha zor bir hale getirecektir [41].

Ajan adı değiştirme atlatma tekniğine karşı saldırı tespit ve engelleme sisteminde bir önlem almak mümkün değildir. Çünkü ajan adı ile saldırıyı tespit etmek saldırı tespit ve engelleme sisteminin saldırıları hızlı bir şekilde keşfedebilmesini sağlayan bir yöntemdir. Bu yöntemin haricinde hazırdaki imza kümesini de kontrol ettiği için bu tekniği önleyici bir savunma bulunmamaktadır [41].

Port numarası değiştirme atlatma tekniğine karşı saldırı tespit ve engelleme sistemlerinde yapılabilecek önlem, kuralların hepsinde port numarası bölümünü tüm portlar şeklinde yapmaktır. Fakat bu yöntem saldırı tespit ve engelleme sistemi için büyük bir handikap yaratacaktır çünkü imzaları işletme süresi uzayacak ve bu sebeple kapasitesinin çok altında hizmet verebilir hale gelecektir. Bu tekniğe karşı önlem almak sistemi daha zayıf bir hale getirecektir.

Kodlama ve gizleme atlatma tekniğine karşı saldırı tespit ve engelleme sisteminde alınabilecek önlem, Snort sistemlerde Unicode.map dosyasına mümkün olduğunca çok çeşitte kodlama metodlarının haritalarını yazmaktır. Gizleme için ise özel kurallar mevcuttur. Ancak her iki atlatma tekniğine karşı tam bir koruma önlemi bulunmamaktadır [40, 42].

Sahte sağlama kodu atlatma tekniğine karşı saldırı tespit ve engelleme sisteminde alınabilecek önlem, ilk kurulumda gelen sağlama değerinin her paket için kontrol edilmesi şeklinde olan ayarın aktif hale getirilmesidir. Bu ayar iki nedenden dolayı kapalı halde bırakılmaktadır. İlk neden; saldırı tespit ve engelleme sistemi her paketin sağlama değerine bakmaya başladığı zaman sistemin kaynak tüketiminde %50 seviyelerine kadar artış gösterebilmesidir. İkinci neden ise sistemin ilk kurulumda sağlama değerini kontrol özelliğinin kapalı olarak gelmesidir [40].

Dosya başlığı değiştirme atlatma tekniğine karşı saldırı tespit ve engelleme sisteminde alınabilecek önlem, standart imzaların yanı sıra saldırıların içerisindeki metadatalara göre de imza yazılmasıdır. Fakat bu yöntem sistemin verimliliğini düşürür ve sistemin yükünü artırır. Bu önlem gizlilik ve bütünlüğün erişilebilirlikten daha önemli olduğu yerlerde uygulanabilir bir önlemdir.

Dosya ve dizin değiştirme atlatma tekniği gerçek bir saldırı ve atlatma senaryosunda gerçekleştirilmesi oldukça güç bir tekniktir. Saldırı tespit ve engelleme sisteminde bu tekniğe karşı alınabilecek bir önlem mevcut değildir.

6. SONUÇ

Günümüzde saldırı tespit ve engelleme sistemleri kurum ve kuruluşların siber saldırılara karşı en önemli korunma kalkanıdır. Siber kalkanın düşmesi veya geçilmesi durumunda arkasında koruduğu sistemler korumasız ve saldırılara açık bir hale gelmektedir.

Yapılan testlerde saldırı tespit ve engelleme sistemlerinin atlatma saldırılarına karşı olan direnci ölçümlenmiştir. Araştırmanın başında önerilen yedi adet atlatma tekniğinden bir kısmı %100 başarı ile testleri geçerken bir kısmı ise bu başarı oranına ulaşamamıştır. Araştırma test sonuçlarına göre güncel bir saldırı tespit ve engelleme sisteminin atlatma tekniklerine karşı hala zayıf olduğu ve atlatılabildiği ispat edilmiştir.

Port numarası değiştirme, kodlama ve gizleme, sahte sağlama kodu, dosya başlığı değiştirme ve dosya ve dizin değiştirme atlatma teknikleri yapılan testlerde %100 başarı oranına ulaşan atlatma teknikleridir. Geriye kalan teknikler beklenen başarı oranına ulaşamamış olsa bile her biri belirli yüzdelerde başarılı olmuşlardır.

Test sonuçlarında beklenen pozitif başarıların yanı sıra negatif başarılar da kaydedilmiştir. Negatif başarı, saldırıların atlatma tekniği eklenmeden önce daha az alarm tetiklediği durumlar olarak adlandırılmıştır. Bu denemelerin önemi saldırganların atlatma tekniğini uyguladığı zaman mutlak suretle başarı kazanamayacağının da bir göstergesidir.

Yapılan testlerde saldırı tespit ve engelleme sistemlerinde tüm imzalar açık olarak denemeler yapılmıştır. Gerçek ortamda çalışan saldırı tespit ve engelleme sisteminde tüm imzalar açılmamakta, tüm imzalar yerine sistemin arkasındaki makineleri ilgilendiren olası imzalar açılmaktadır. Saldırı tespit ve engelleme sistemini yönetenler açısından; atlatma teknikleri kullanıldığı zaman saldırı denemesini yakalayabilecekleri imza sayısı testlerde denenen imza sayısından daha az olacaktır. Bu nedenle saldırının görünmez bir şekilde başarıya ulaşma olasılığı yüksektir. Savunanlar açısından saldırı tespit ve engelleme sistemlerinde kurban sistemlere göre atlatma tekniklerine karşı önlem almak oldukça önemlidir.

KAYNAKLAR

[1] https://searchsecurity.techtarget.com/definition/intrusion-detection-system/ alındığı tarih: 04.02.2019

[2] https://www.academia.edu/34024105/Snort_IDS_and_IPS_Toolkit/ alındığı tarih: 11.01.2019

[3] **Shon Harris, Fernando Maymi,** (2018). CISSP All-In-One Exam Guide, 8th ed. McGraw-Hill Education.

[4] http://www.lfca.net/Reference%20Documents/2003-CSI-FBI-Survey.pdf/ alındığı tarih:16.01.2019

[5] https://www.sans.org/reading-room/whitepapers/detection/ alındığı tarih: 11.02.2019

[6] **Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras and B. Stiller,** (2010). "An Overview of IP Flow-Based Intrusion Detection," IEEE Commun. Surveys Tutorials, vol. 12, issue 3, pp. 343-356.

[7] D. Denning and P. Neumann, (1986). An Intrusion-Detection Model, IEEE Symposium on Security and Privacy, Volume 1, sayfa 118, Amerika.

[8] https://bricata.com/blog/bro-ids-threat-detection/ alındığı tarih: 22.03.2019

[9] **A. Pathan** (2016) The State of the Art in Intrusion Prevention and Detection, 1th ed., Routledge.

[10] **A. Ghorbani, W. Lu and M. Tavallaee**, (2010) Network intrusion detection and prevention, 1st ed. New York: Springers.

[11] http://insecure.org/stf/secnet-ids/secnet ids.html/ alındığı tarih: 07.04.2019

[12] **M. Handley, V. Paxson, and C. Kreibich**, (2001) "Network Intrusion Detection: Evasion, Traffic Normalization, and End-to-end Protocol Semantics," In Proc. USENIX Security Symposium, Amerika, Ağustos 13-17.

[13] https://fortiguard.com/encyclopedia/ips/12934/ alındığı tarih: 07.03.2019

[14] https://www.giac.org/paper/gsec/589/ip-fragmentation-attacks-checkpoint-firewalls/101350 alındığı tarih: 15.04.2019

[15] https://security.radware.com/ddos-knowledge-center/ddospedia/http-fragmentation-attack/ alındığı tarih:20.05.2019

[16] https://www.secjuice.com/port-scanning-penetration-testing-part-three/ alındığı tarih: 22.04.2019

[17] https://www.cardinaleconcepts.com/add-custom-header-to-nikto-scan/ alındığı tarih: 17.04.2019

[18] https://www.cvedetails.com/cve/CVE-2017-0144/ alındığı tarih: 07.01.2019

[19] https://www.cvedetails.com/cve/CVE-2017-0143/ alındığı tarih: 07.01.2019

[20] https://www.cvedetails.com/cve/CVE-2018-17082/ alındığı tarih: 07.01.2019

[21] https://www.microsoft.com/en-us/wdsi/threats/malware-encyclopedia-description?Name=Worm%3AWin32/Dorkbot/ alındığı tarih: 08.02.2019

[22] **P. Fogla and W. Lee**, (2006) "Evading Network Anomaly Detection Systems: Formal Reasoning and Practical Techniques," In Proc. ACM Conference on Computer and Communications Security (CCS), Amerika.

[23] https://securityintelligence.com/an-example-of-common-string-and-payload-obfuscation-techniques-in-malware/ alındığı tarih: 18.01.2019

[24] **D. Watson, M. Smart, G. R. Malan and F. Jahanian**, (2004), "Protocol Scrubbing: Network Security Through Transparent Flow Modification," IEEE/ACM Trans. Netw., vol. 12, issue 2, pp. 261-273, April.

[25] **S. Dharmapurikar and V. Paxson**, (2005). "Robust TCP Stream Reassembly In the Presence of Adversaries," In Proc. USENIX Security Symposium, Hollanda.

[26] https://resources.infosecinstitute.com/penetration-testing-methodologies-and-standards/ alındığı tarih: 08.02.2019

[27] http://evader.stonesoft. com/assets/files/Evader_UsersGuide_20120905.pdf/ alındığı tarih: 13.05.2019

[28] **P. Thomas H and N. Timothy**, (1998). Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection. Adres: https://www.cs.unc.edu/~fabian/course_papers/PtacekNewsham98.pdf

[29] **T. Cheng, Y. Lin, Y. Lai, P. Lin**, (2011). Evasion Techniques: Sneaking through Your Intrusion Detection/Prevention Systems, IEEE Communications Surveys & Tutorials, Volume: 14, Issue: 4, sayfa 1011-1020.

[30] **D. Lipeng, C. Xingyuan, T. Huilin, S. Wang**, (2013). A generation framework of multiple evasions on IDS, Third International Conference on Instrumentation, Measurement, Computer, Communication and Control, Zhengzhou, China.

[31] **R. Rutuja ve P. R. Devale**, (2014). To Fınd New Evasıon Technıques On Network Intrusıon Detectıon System, Vol. 2, Issue 3, sayfa 129-140.

[32] **M. Chammen, M. Hamdi, T. Kim**, (2014). Extending Advanced Evasion Techniques Using Combinatorial Search, 7th International Conference on Security Technology, Amerika.

[33] **M. Särelä, T. Kyöstilä, T. Kiravuo and J. Manner**, (2017). "Evaluating intrusion prevention systems with evasions", International Journal of Communication Systems, vol. 30, no. 16.

[34] http://www.memecode.com/docs/linux-critical-path.html/ alındığı tarih: 27.01.2019

[35] **S. Huang, C. Blundo, S. Cimato, B. Masucci, D. MacCallum and D. Du**, (2010). Network Security, 1st ed., Springer.

[36] https://www.garykessler.net/library/file_sigs.html/ alındığı tarih: 11.03.2019

[37] **N. Khamphakdee, N. Benjamas and S. Saiyod**, (2014). "Improving Intrusion Detection System based on Snort rules for network probe attack detection", 2nd International Conference on Information and Communicaiton Technology (ICoICT), Malezya.

[38] https://bugs.launchpad.net/ubuntu/ alındığı tarih: 11.03.2019

[39] https://www.ibm.com/support/knowledgecenter/en/SSB2MG_4.6.0/com.ibm.ips.doc/tasks/configuring_snort.html/ alındığı tarih: 17.03.2019

[40] https://www.securityarchitecture.com/learning/intrusion-detection-systems-learning-with-snort/configuring-snort/ alındığı tarih:

[41] https://wiki.archlinux.org/index.php/Snort/ alındığı tarih: 16.02.2019

[42] https://resources.infosecinstitute.com/snort-rules-workshop-part-one/ alındığı tarih: 21.02.2019

[43] https://www.sans.org/security-resources/tcpip.pdf/ alındığı tarih: 08.03.2019

[44] https://www.giac.org/paper/gcia/615/intrusion-detection-evasion-trace-analysis/104437/ alındığı tarih: 09.01.2019

[45] https://svn.nmap.org/nmap/docs/nmap.usage.txt/ alındığı tarih: 09.01.2019

[46] https://www.exploit-db.com/exploits/40872/ alındığı tarih: 18.03.2019

EKLER

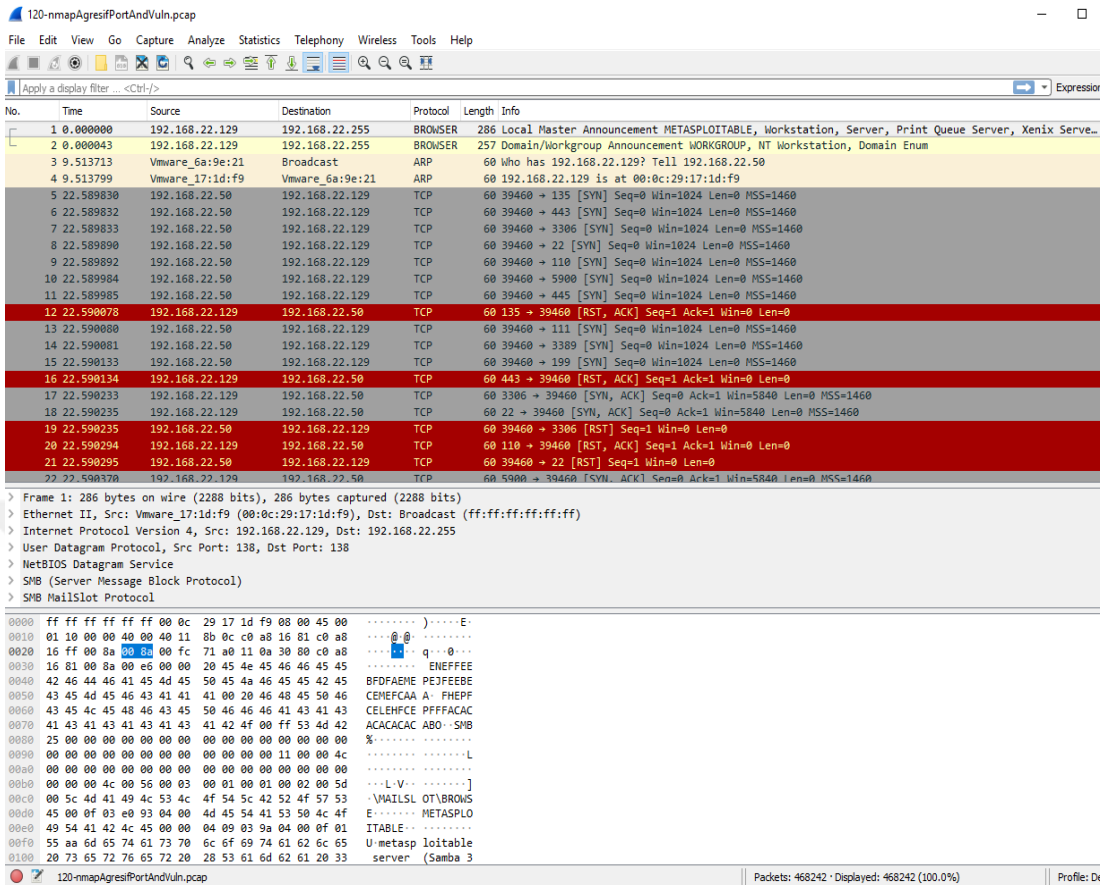
EK 1: APZT kısaltmalı saldırı dosyasının Wireshark ile gösterimi

EK 2: Nmap taramsının Sguil ile gözlemlenmesi

EK 3: Snort için indirilen community rules'un gösterimi

EK 4: Nmap taramasının Snort uyarılarının CLI gösterimi (APZT kısaltmalı saldırı için oluşan alarmlardan bir örnek)

EK 1



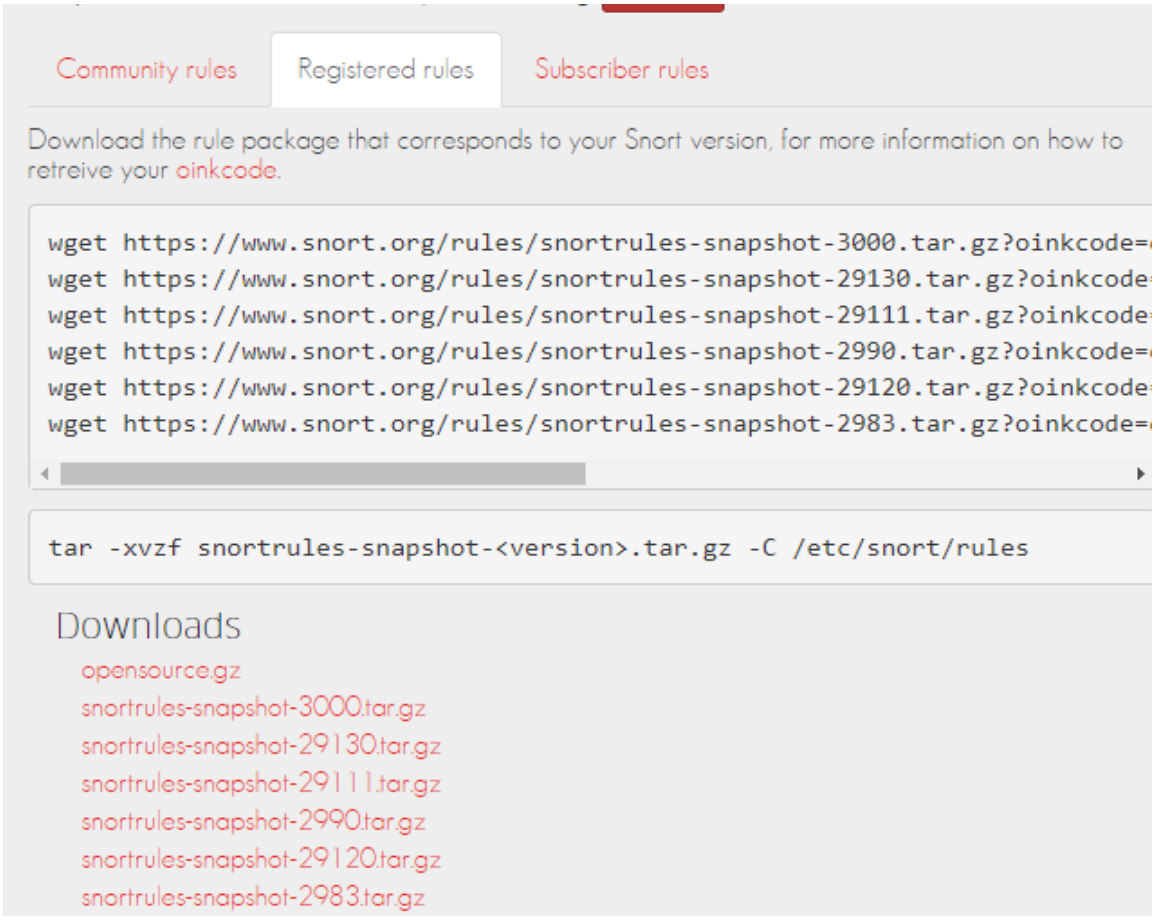
Şekil Ek.1: MTU ile paket parçalama ve sahte sağlama kodu tekniklerinde kullanılan APZT kısaltmalı saldırı Wireshark ile gösterimi.

Ek 2

ST	CNT	Sensor	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
RT	2	hakan...	3.31653	2019-02-18 13:56:50	192.168.22.50	57685	192.168.22.128	3306	6	ET SCAN Suspicious inbound to mySQL port 3306
RT	2	hakan...	3.31655	2019-02-18 14:16:06	192.168.22.50	57685	192.168.22.128	1521	6	ET SCAN Suspicious inbound to Oracle SQL port 1521
RT	2	hakan...	3.31657	2019-02-18 15:28:10	192.168.22.50	57685	192.168.22.128	1433	6	ET SCAN Suspicious inbound to MSSQL port 1433
RT	2	hakan...	3.31659	2019-02-18 22:20:56	192.168.22.50	57685	192.168.22.128	5432	6	ET SCAN Suspicious inbound to PostgreSQL port 5432
RT	8	hakan...	3.31661	2019-02-18 23:04:17	192.168.22.50	55715	192.168.22.128	39697	17	ET SCAN NMAP OS Detection Probe
RT	30	hakan...	3.31721	2019-02-18 23:16:19	192.168.22.50	52246	192.168.22.128	80	6	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting En...
RT	30	hakan...	3.31710	2019-02-18 23:16:19	192.168.22.50	52228	192.168.22.128	80	6	ET SCAN Possible Nmap User-Agent Observed

Şekil Ek.2 : Snort'un Sguil arayüzü ile örnek bir alarm kümesi gösterimi.

Ek 3



Şekil Ek.3 : Register olduktan sonra indirilen en son Snort kuralları.

EK 4

```
02/24-14:04:53.453506 [**] [119:228:1] "(http_inspect) server response before
client request" [**] [Priority: 3] {TCP} 192.168.22.129:80 ->
192.168.22.50:36264
02/24-14:04:53.488938 [**] [119:228:1] "(http_inspect) server response before
client request" [**] [Priority: 3] {TCP} 192.168.22.129:8180 ->
192.168.22.50:36562
02/24-14:04:53.599439 [**] [119:228:1] "(http_inspect) server response before
client request" [**] [Priority: 3] {TCP} 192.168.22.129:80 ->
192.168.22.50:36276
02/24-14:04:53.642031 [**] [119:201:1] "(http_inspect) not HTTP traffic" [**]
[Priority: 3] {TCP} 192.168.22.50:36580 -> 192.168.22.129:8180
02/24-14:04:53.692693 [**] [119:228:1] "(http_inspect) server response before
client request" [**] [Priority: 3] {TCP} 192.168.22.129:8180 ->
192.168.22.50:36586
02/24-14:04:53.749536 [**] [119:228:1] "(http_inspect) server response before
client request" [**] [Priority: 3] {TCP} 192.168.22.129:80 ->
192.168.22.50:36296
02/24-14:04:53.895305 [**] [119:228:1] "(http_inspect) server response before
client request" [**] [Priority: 3] {TCP} 192.168.22.129:8180 ->
192.168.22.50:36594
02/24-14:04:54.095966 [**] [119:228:1] "(http_inspect) server response before
client request" [**] [Priority: 3] {TCP} 192.168.22.129:8180 ->
192.168.22.50:36598
02/24-14:04:54.195997 [**] [119:228:1] "(http_inspect) server response before
client request" [**] [Priority: 3] {TCP} 192.168.22.129:8180 ->
192.168.22.50:36602
02/24-14:04:54.296667 [**] [119:228:1] "(http_inspect) server response before
client request" [**] [Priority: 3] {TCP} 192.168.22.129:8180 ->
192.168.22.50:36604
02/24-14:04:54.497302 [**] [119:228:1] "(http_inspect) server response before
client request" [**] [Priority: 3] {TCP} 192.168.22.129:8180 ->
192.168.22.50:36606
02/24-14:04:54.697055 [**] [119:228:1] "(http_inspect) server response before
client request" [**] [Priority: 3] {TCP} 192.168.22.129:8180 ->
192.168.22.50:36608
02/24-14:04:54.897538 [**] [119:228:1] "(http_inspect) server response before
client request" [**] [Priority: 3] {TCP} 192.168.22.129:8180 ->
192.168.22.50:36610
```

Şekil Ek.2 : Snort alarmlarının CLI'da gösterilmiş hali.

ÖZGEÇMİŞ

Ad-Soyad : Hakan KILIÇ

Uyruğu : T.C.

Doğum Tarihi ve Yeri : 1993, ANKARA

E-posta : hakank.kilic@gmail.com

ÖĞRENİM DURUMU:

- **Lisans** : 2017, İhsan Doğramacı Bilkent Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği
- **Yükseklisans** : 2019, TOBB ETÜ, Fen Bilimleri Fakültesi, Bilgisayar Mühendisliği, Bilgi Güvenliği

MESLEKİ DENEYİM VE ÖDÜLLER:

Yıl	Yer	Görev
2017-2018	Barikat Bilişim Güvenliği	Güvenlik Analisti ve Olay Müdahale
2018-Halen	Barikat Bilişim Güvenliği	Satış Öncesi Mühendisi

YABANCI DİL: İngilizce, Fransızca

TEZDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER:

- **Kılıç,H., Katal, S. and Selçuk, A. A**., 2019. Evasion Techniques Efficiency Over The IPS/IDS Technology, UBMK-19 Uluslararası Bilgisayar Bilimleri ve Mühendisliği Konferansı, 11-15 Eylül, Samsun, Türkiye.