T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SİBER GÜVENLİK ANALİZİ İÇİN YENİ BİR SİBER SALDIRI SİMÜLATÖRÜ GELİŞTİRİLMESİ

DOKTORA TEZİ

Şahin KARA

Enstitü Anabilim Dalı	**:**	**BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ**
Tez Danışmanı	**:**	**Prof. Dr. Ahmet ZENGİN**

Ağustos 2022

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SİBER GÜVENLİK ANALİZİ İÇİN YENİ BİR SİBER SALDIRI SİMÜLATÖRÜ GELİŞTİRİLMESİ

DOKTORA TEZİ

Şahin KARA

Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ

Bu tez 17 / 08 /2022 tarihinde aşağıdaki jüri tarafından oybirliği/oyçokluğu ile kabul edilmiştir.

BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Şahin KARA
17.08.2022

TEŞEKKÜR

Doktora eğitimim boyunca değerli bilgi ve deneyimlerinden yararlandığım, her konuda bilgi ve desteğini almaktan çekinmediğim, araştırmanın planlanmasından yazılmasına kadar tüm aşamalarında yardımlarını esirgemeyen, teşvik eden, aynı titizlikte beni yönlendiren değerli danışman hocam Prof. Dr. Ahmet ZENGİN'e teşekkürlerimi sunarım.

Bugünlere gelmemi sağlayan ve her zaman yanımda olan aileme ve dostlarıma da destekleri için teşekkür ederim.

İÇİNDEKİLER

SİMGELER VE KISALTMALAR LİSTESİ

ACE	: Uygulama Karakterizasyon Ortamı (Application Characterization Environment)
ADEVS	: Ayrık Olay Sistem Simülatörü (A Discrete Event System Simulator)
ADS-B	: Otomatik Bağımlı Gözetim Yayını (Automatic Dependent Surveillance-Broadcast)
ARP	: Adres Çözümleme Protokolü (Address Resolution Protocol)
BRITE	: Boston Üniversitesi İnternet Topoloji Üreteci (Boston University Representative Internet Topology Generator)
BT	: Bilişim Teknolojileri
CSE-CIC	: İletişim Güvenliği Kurumu ve Kanada Siber Güvenlik Enstitüsü (Communications Security Establishment and the Canadian Institute for Cybersecurity)
CSRF	: Siteler Arası İstek Sahteciliği (Cross-Site Request Forgery)
CSV	: Virgülle Ayrılmış Değerler (Comma-Separated Values)
DDoS	: Dağıtılmış Hizmet Reddi (Distributed Denial of Service)
DES	: Ayrık Olaylı Sistem (Discrete Event System)
DEVS	: Ayrık Olaylı Sistem Tanımlama (Discrete Event System Specification)
DEVS-CAS	: Ayrık Olay Sistem Tanımlı Siber Saldırı Simülatörü (Discrete Event System Defined Cyber Attack Simulator)
DEVSJAVA	: Java Destekli DEVS Modelleme ve Simülasyon Ortamı(Java Supported DEVS Modeling and Simulation Environment)
DHCP	: Dinamik Ana Bilgisayar Yapılandırma Protokolü (Dynamic Host Configuration Protocol)

DNS : Alan Adı Sistemi (Domain Name System)
DoS : Hizmet Reddi (Denial-of-Service)
DSR : Dinamik Kaynak Yönlendirme (Dynamic Resource Routing)
DTE : Geliştirme ve Test Ortamı (Development and Test Environment)
DTN : Gecikme Toleranslı Ağ (Delay Tolerant Network)
FIN : TCP Bağlantı Bitiş İşareti (Finish)
FTP : Dosya Transfer Protokolü (File Transfer Protocol)
GPL : Genel Kamu Lisansı (General Public License)
ICMP : İnternet Kontrol Mesaj Protokolü (Internet Control Message Protocol)
IDS : Saldırı Tespit Sistemi (Intrusion Detection System)
INET : İnternet Topoloji Üreteci (Internet Topology Generator)
IP : İnternet Protokolü (Internet Protocol)
IT : Bilişim Teknolojileri (Information Technology)
LDAP : Basit Dizin Erişim Protokolü (Lightweight Directory Access Protocol)
MAC : Ortam Erişim Denetleyicisi (Media Access Controller)
MANET : Mobil Özel Ağlar (Mobile Adhoc Network)
MITM : Ortadaki Adam Saldırısı (Man in the Middle)
MVC : Model Görünüm Denetleyicisi (Model-View-Controller)
NTP : Ağ Zaman Protokolü (Network Time Protocol)
OSPF : İlk Açık Yöne Öncelik Yönlendirme Protokolü (Open Shortest Path First)
OTcl : Nesne Yönelimli Dinamik Programlama Dili (Object Tool Command Language)
PARSEC : Karmaşık Sistemler için Paralel Simülasyon Ortamı (Parallel Simulation Environment for Complex Systems)
PCAP : Ağ Paketi Veri Dosyası (Packet Capture File)
P-DEVS : Paralel DEVS (Parallel DEVS)
RST : TCP Bağlantı Kapalı İşareti (Reset)
RT-DEVS : Gerçek Zamanlı DEVS (Real Time-DEVS)
SCADA : Kontrol ve İzleme Sistemi (Supervisory Control and Data

	Acquisition)
SITL	: Sistem İçi Döngü (System In the Loop)
SMTP	: Basit Posta Aktarım Protokolü (Simple Mail Transfer Protocol)
SNMP	: Basit Ağ Yönetim Protokolü (Simple Network Management Protocol)
SNT	: Ölçeklenebilir Ağ Teknolojileri (Scalable Network Technologies)
SOAP	: Basit Nesne Erişim Protokolü (Simple Object Access Protocol)
SQL	: Yapılandırılmış Sorgu Dili (Structured Query Language)
SQLI	: SQL enjeksiyonu (SQL Injection)
SSH	: Uzak Sunucu Güvenli Bağlantı Protokolü (Secure Shell)
SYN	: TCP Bağlantı Başlatma İşareti (Synchronize)
TCP	: İletim Kontrol Protokolü (Transmission Control Protocol)
TL	: En Son Olay Zamanı (Last Event Time)
TN	: Bir Sonraki Olay Zamanı (Next Event Time)
UDDI	: Evrensel Tanımlama, Keşif ve Entegrasyon (Universal Description, Discovery, and Integration)
UDP	: Kullanıcı Datagram Protokolü (User Datagram Protocol)
USB	: Evrensel Seri Veri Yolu (Universal Serial Bus)
VANET	: Araçsal Tasarsız Ağlar (Vehicular Ad Hoc Network)
WAF	: Web Uygulaması Güvenlik Duvarı (Web Application Firewall)
WEP	: Kablosuz Ağ Güvenlik Algoritması (Wired Equivalent Privacy)
WiMAX	: Geniş Bant Kablosuz Erişim Teknolojisi (Worldwide Interoperability for Microwave Access)
WMN	: Kablosuz Örgü Ağları (Wireless Mesh Network)
WSDL	: Web Hizmetleri Açıklama Dili (Web Services Description Language)
WSN	: Kablosuz Sensör Ağı (Wireless Sensor Networks)
XMAS	: X-Işını Mikro Kırınım Analiz Yazılımı (X-Ray Microdiffraction Analysis Software)
XML	: Genişletilebilir İşaretleme Dili (Extensible Markup Language)
XSS	: Siteler Arası Komut Dosyası Çalıştırma (Cross Site Scripting)

ŞEKİLLER LİSTESİ

TABLOLAR LİSTESİ

ÖZET

Anahtar kelimeler: Siber güvenlik, siber saldırı deneyleri, modelleme ve simülasyon, ağ sınama ortamları, DEVS.

Kurumların ve bireylerin iş süreçlerini bilişim teknolojileri ile yürütme zorunluluğu beraberinde risk ve tehditleri de beraberinde getirmiştir. Siber saldırılar, telafisi zor sonuçlara yol açabilir. Bu saldırılara karşı pek çok saldırı tespit ve güvenlik sistemi geliştirilmiş olsa da, bilgi sistemlerine yönelik saldırılar ve güvenlik ihlalleri hızla artmaktadır. Bu çalışmada siber güvenliğin sağlanmasında en önemli konulardan biri olan güvenlik zafiyetlerinin anlaşılması ve siber saldırıların tespit edilmesi için yazılım tabanlı bir araç geliştirilmesi amaçlanmaktadır. Siber saldırı yöntemlerini test etmek için fiziksel ağları kullanmak çok maliyetli ve zaman alıcı bir süreçtir. Bu tez çalışmasında, siber saldırı senaryolarını simüle etmek, test etmek ve sonuçları değerlendirmek için DEVS modelleme yaklaşımı kullanılarak bir siber saldırı simülasyon modeli geliştirilmiştir. Saldırı modellerinin üzerinde çalıştıralacak model ağın topolojisi, BRITE topoloji üreteci tarafından oluşturulmuştur. Saldırı modellerinin özelliklerine göre saldırı senaryolarının parametre ve yapılandırma ayarlarının yapıldığı görsel arayüzler kullanılarak saldırı simülasyonu gerçekleştirilmiştir. Saldırı adımlarının etkilerinin ve sonuçlarının gözlemlenmesi ve değerlendirilmesi için simülasyon izleme çerçevesi kullanılmıştır. Böylece bir sanal ağda bir saldırı senaryosunu simüle eden ve uygun saldırı tespit sistemi uyarıları üreterek bu uyarıları değerlendiren bir uygulama geliştirilmiştir. Ayrıca, saldırı önleme amacıyla ağ trafiğinin incelendiği ve şüpheli etkinlik belirlendiğinde "refleks" tipi eylemlerin gerçekleştirildiği bir saldırı tespit sistemi geliştirilmiştir. Sistemin test edilmesi için Kanada Siber Güvenlik Enstitüsü tarafından paylaşılan CSE-CIC-IDS2018 veri seti kullanılmıştır. Geliştirme ortamı olarak DEVS-Suite simülasyon paketi kullanılmıştır. Farklı siber saldırı simülasyon uygulamaları ile karşılaştırmalar yapılmış ve farklılıkları ortaya konmuştur. Bu araç belirli saldırı türleri için uyarı verileri elde etmek maksadıyla kullanılsa da, sonraki çalışmalarda daha farklı saldırı türleri için de uyarı verileri oluşturmak üzere genişletilebilir bir altyapı sağlamaktadır. Bu çalışmada, büyük ölçekli kurumsal ağların kolaylıkla tasarlanabileceği ve geçerli düzeyde performans, ölçeklenebilirlik ve doğruluk ile siber güvenlik testlerinin kısa sürede yapılabileceği görülmüştür.

DESIGN AND IMPLEMENTATION OF A NEW CYBER ATTACK SIMULATOR FOR CYBER SECURITY ANALYSIS

SUMMARY

Keywords: Cyber security, cyber attack experiments, simulation and modeling, network testing environments, DEVS.

The fact that institutions and people have to deal with their work with information technologies has also brought risks and threats. Cyber attacks can have consequences that are difficult to recover. Although many intrusion detection and security systems have been developed against these attacks, attacks and security breaches against information systems are increasing rapidly. In this thesis, it is aimed to understand security vulnerabilities, which is one of the important issues in terms of cyber security, and to detect cyber attacks. Using real networks for cyber attack test runs is very costly and time consuming. In this thesis, a cyber attack simulation model has been developed using the DEVS modeling approach to simulate cyber attack scenarios, test the attacks and observe the results. The attack simulation was carried out by using the developed visual interfaces, in which the parameters and configuration settings of the attack scenarios were made according to the characteristics of the attack modes. A simulation and tracking framework was used to observe and evaluate the effects and consequences of attack steps. An application has been developed that simulates attack scenarios on a developed network and evaluates these warnings by generating appropriate attack alerts. In this study, a method similar to the methodology of intrusion detection systems is developed, in which network traffic is examined for intrusion prevention and "reflex" type actions are performed when suspicious activity is detected. CSE-CIC-IDS2018 dataset was used to test the developed system. DEVS-Suite simulation package was used as application development environment. The developed cyber attack simulator has been compared with other cyber attack simulation applications and its different aspects have been revealed. Although this cyberattack simulator is used to obtain alert data for certain types of attacks, it provides an extensible infrastructure to generate alert data for more different types of attacks in future studies. In this study, it has been seen that large-scale corporate networks can be designed easily and cyber security tests can be carried out in a short time.

BÖLÜM 1. GİRİŞ

1.1. Problem Tanımı

Teknoloji hayatımızı kolaylaştırmıştır ancak risk ve tehditleri de beraberinde getirmiştir. Bilgisayar ve ağlardaki açıklıklar siber saldırılar için uygun bir zemin haline gelmiştir. Gerçekleştirilen siber saldırılar çok farklı amaç ve motivasyonlarla yapılmaktadır [1].

Bilginin değiştirilmesi, çalınması, yok edilmesi kişi ve kuruluşlar için telafisi zor sonuçlar doğurabilmektedir. Siber saldırıları gerçekleştirmek için bilişim teknolojileri yoğun olarak kullanılmakta olup, kullandığımız bilgisayar sistemleri de bu zararlı saldırılara maruz kalmaktadır. Bilişim teknolojisi, bilişim güvenliği teknolojisinden daha hızlı ilerlemektedir. Her zaman güvenlik boşluklarından sızmaya çalışan bir tehdit kaynağı mevcuttur [2].

Bilgisayar ağ sistemlerinin güvenliği gün geçtikçe daha kritik bir güvenlik problemi olarak karşımıza çıkmaktadır. Siber saldırıları gerçekleştirmek için kullanılan uygulamalar kolaylıkla elde edilebilmektedir. Bu uygulamalarla acemi bilgisayar korsanları bile yıkıcı siber saldırılar gerçekleştirilebilmektedirler. Bu saldırıların zararlarından korunmak için pek çok saldırı tespit sistemi geliştirilmiştir. Bu çalışmalar, bilişim sistemlerine yönelik saldırı ve güvenlik ihlallerini engellemek için yeterli olamamaktadır [3].

Bireyler, kurumlar ve devletler, yüksek miktarda önemli verilerini siber ortamda bulundurmaktadır. Bu durum kötü niyetli siber saldırganları farklı amaçlarla harekete geçiren bir motivasyon kaynağı olmaktadır. Hedef alınan siber ortamlara yönelik siber saldırı gerçekleştirmek isteyen tarafların hedeflerine ulaşmak için başvurabilecekleri pek çok siber saldırı yöntemi mevcuttur. Saldırganlar, virüsler,

truva atları, DoS, DDoS, yemleme, ağ trafiği dinleme, araya girme, casus yazılımlar gibi daha pek çok yöntemi kullanarak amaçlarını gerçekleştirmektedirler.

Özel veya kurumsal bir ağda güvenliğin sağlanması, kuruluşların bilgi yönetimiyle ilgili olarak karşılaştığı önde gelen zorluklardan biridir. Kurumsal ve kişisel bilgiler, güvenilir kişiler tarafından erişilmesi amacıyla özel ağlarda saklanır. Bu bağlamdaki çoğu özel ağda internet bağlantısı vardır, bu durum ağ için web tabanlı saldırıları bir tehdit unsuru yapar. Bilgisayar korsanları bu tür ağları takip ederek bu ağlara erişim sağlamalarına yardımcı olacak istismarları araştırıp denemektedirler.

Bilgisayar ağlarındaki tüm güvenlik açıklarını bulmak ve çeşitli güvenlik önlemlerini güncel tutmak şirket ağ yöneticileri için zorlu bir süreçtir. Bu durum, ağa saldırmayı planlayanlar için bir avantaja dönüşmektedir [4].

Bu tez çalışmasında bilgisayar ağlarında güvenliği sağlanmak amacıyla modelleme ve simülasyon araçlarının kullanıldığı bir uygulama geliştirilmiştir. Siber güvenliğin sağlanması için en önemli aşama olarak güvenlik zafiyetlerinin ve açıklıkların kötü niyetli siber saldırganlardan önce bulunup önlem alınması sistemi de tasarlanmıştır.

1.2. Literatür İncelemesi

Bu bölüm, daha önce bu alanlarda gerçekleştirilen bazı önemli araştırmaları gözden geçirmektedir ve araştırmacılar tarafından geliştirilen farklı modelleme yaklaşımlarını içermektedir.

Siber saldırıları anlamak için etkili bir araç, saldırıların sınıflandırılması ve saldırı ilerlemesinin modellenmesidir. Bu, saldırganın davranışını anlamak için bilinen bir saldırıyla ilişkili eylemleri dikkatle gözlemeyi içerir. Dougherty ve Gonslaves [5], yazılım korumasının test edilmesine yardımcı olmak için uyarlanabilir bir siber saldırı sistemi geliştirmiştir. Yapılan araştırma, uygun siber saldırı modellerinin geliştirilmesinin, maliyeti ve zamanı önemli ölçüde azaltma potansiyeline sahip olduğunu göstermiştir. Bu modellemeyle, üç temel saldırı kategorisi belirlenmiştir.

Bunlar; web tabanlı uygulama, istemci-sunucu uygulama saldırıları ve tek başına sistem saldırılarıdır. Bu araştırma yoluyla yapılan modelleme, yalnızca yazılım korumasını test etmek için geliştirilmiştir, bağımsız sistem saldırılarını modellemede eksik kalmıştır.

Diğer araştırmacılar, atak adımlarıyla ilişkili yalıtılmış IDS uyarılarını analiz ederek saldırı modellemeye odaklanmıştır. Örneğin, Cheung ve ark. [6], saldırı senaryolarının gerçek IDS uyarılarını gözleyerek modellendiği "İlişkili Saldırı Modelleme" adlı bir projeyi geliştirmiştir. Bu IDS uyarıları genelde belirli bir saldırı adımıyla (genellikle bir takım sömürme) ilişkilendirilebilir. Bu, gerçek saldırıların saldırı modelleri geliştirilmesinde kullanılmasına izin verse de, birçok IDS uyarısı yanlış pozitif olabilir ve bu nedenle modelleme sürecini etkileyebilir. Dahası, bazı saldırı adımları tamamen atlanmış olabilir.

Ayrıca, bir bilgisayar korsanının uygulayabileceği farklı istismarların tipik sıralamasını belirlemek için daha üst düzey araştırmalar yapılmıştır. Holdender ve ark. [7], grafik teori tekniklerinden yararlanarak, bazı saldırı türlerinin yapılabilmesi için hangi saldırı eylemlerinin veya istismarın gerekli olduğunu belirleyen bir grafik tabanlı şablon geliştirmiştir. Bu gelişme ilk olarak bilinen sömürü türlerini, bu istismarlar meydana gelmeden önce nasıl bir faaliyetin gerekli olduğunu ve daha sonra ne tür bir faaliyette bulunulabileceğini temel alarak kategorilere veya aşamalara göre gruplandırmak suretiyle gerçekleştirilmiştir. Ardından, farklı aşamaları ilişkilendirmek için bir bitişiklik matrisi geliştirilmiş; belirli bir aşamadaki bir istismar, şablonun tüm aşamalarında gerçekleştikten sonra hangi aşamaların oluşabileceğini belirlenmiştir. Bu çalışmada bilinen güvenlik açıklarının olası istismarlarını aşamalardan birinde kategorize ederek saldırıyı modelleme süreci basitleşir, ancak yeni ortaya çıkan güvenlik açıklarından dolayı dikkate alınması gereken çok sayıda istismarın saldırı aşamalarının belirlenip modellenmesi gerekir.

Garg ve ark. [8], saldırıları tespit etmek amacıyla güvenlik mekanizmalarının yeteneklerini ölçmek için bir çerçeve geliştirmiştir. Saldırı önleme algılama sistemleri ve diğer güvenlik sistemleri arasındaki hataların, kuruluşlar açısından

hayati sonuçları olabilir. Böylelikle, güvenlik hatalarının bu tür sistemlerde nerede olduğunu tespit etmek çok önemlidir. Geliştirilen çerçeve, karmaşık saldırıları simüle etmek için bir platform ve güvenlik algılama mekanizmaları ve saldırılar için şablonlar içermektedir. Çerçeve güvenlik sistemleri arasındaki farklılıkların değerlendirilmesine odaklanılmıştır. Bu çalışmaya IDS'ler dahil edilmiş olsa da, IDS uyarı verileri sağlamaktan ziyade güvenlik sistemleri arasındaki farklılıkları değerlendirilmesine yoğunlaşılmıştır. IDS'ler ve diğer güvenlik sistemleri, özellikle uyarı verileri gerektiren bir simülasyon için gerekli olandan çok daha karmaşık bir şekilde modellenmiştir.

DeLooze ve ark. [9], siber saldırıların ve güvenlik sistemlerinin kombinasyonunu modellemek için bir simülasyon metodolojisi geliştirmiştir. Ağ güvenliği alanında kariyer yapmaya çalışan bireylerin eğitim ve öğretim kurslarına yardımcı olmak için "Sanal Ağ Simülasyonu" adlı bir simülasyon modeli geliştirilmiştir. Geliştirilen simülasyon modeli kullanıcının takdirine bağlı olarak ayarlanabilen büyük miktarda ağ cihazları ve bu ağlara simüle edilmiş saldırıları gerçekleştirebilir. Bu sistem, bir eğitim aracı olarak kurulduğundan, saldırılar ve IDS uyarılarıyla ilişkili veri üretimi için yeterli derecede iyi tasarlanmamıştır.

Kuhl ve Kistner [10], ticari simülasyon paketi ARENA'nın kullanımı ile bilgisayar ağlarının modellenmesine ve siber saldırıların ağlar içinde yapılmasına ve modellenmiş bilgisayar ağı içerisindeki ilgili IDS sensörleri için uyarıların üretilmesine olanak tanıyan bir simülasyon yapısı geliştirmiştir. Kistner [11], ağ aygıtları için daha ayrıntılı nitelikler geliştirmek ve bir dizi parametre temelinde saldırıları otomatik olarak üretmek için bir yöntem sağlamak maksadıyla bu çalışmayı daha da genişletmiştir.

F. Cohen [12] tarafından geliştirilen simülatör ilklerden biri olup sonraki bazı çalışmalara temel teşkil etmiştir. Simüle edilmiş siber saldırıların, düğüm ve bağlantılarla modellenen bir ağ üzerindeki etkileri için sayısal değerler üreten bir simülatör geliştirilmiştir. Simülatörde, 37 tehdit profili (davranış), 94 atak (fiziksel ve siber) ve 140 savunma mekanizmasından oluşan daha önceden geliştirilmiş bir

neden-sonuç modeli kullanılmıştır. Saldırganın beceri seviyesi ve gizliliği kullanıcı tarafından tanımlanan parametrelerdir. Simülasyon çıkışı, saldırganın hedefe ulaşması durumunda, saldırı süresi ve sonucu içermektedir.

Kotenko ve Mankov [13], tasarım ve dağıtım aşamalarında bilgisayar ağlarının güvenlik açığının aktif olarak değerlendirilmesi için tasarlanan "Attack Simulator" yazılım aracıyla ilgili uygulama sorunlarını ve deneylerini anlatmaktadır. Önerilen model varlıklardan oluşan saldırı yapılandırması ve saldırı senaryolarının durum makinelerinin tanımlamalarına dayanmaktadır. Saldırı simülatörünün genelleştirilmiş bir ajan tabanlı mimarisini karakterize etmektedir. Bilgisayar ağı modeli ve gerçek bir bilgisayar ağı saldırıları analiz edilmektedir. Farklı yapılandırmalara ve güvenlik politikalarına sahip bilgisayar ağlarına karşı çeşitli saldırı senaryoları üretmek için Attack Simulator'un etkinliğini gösteren deneyler yapılmaktadır. Bu çalışma bilgisayar ağı benzetiminde ağ ve ağ bileşenleri hakkında çok az ve sınırlı bilgi içermektedir.

Ulanov ve Kotenko [14] Internette yazılımsal ajanlardan oluşan ekiplerin ve aralarındaki siber savaş senaryolarının modelleme ve simülasyonunu gerçekleştirmiştir. Örnek bir DDoS saldırısı yapan ve buna karşı savunma yapan ekip simülasyonu gösterilmiştir. DDoS saldırılarına karşı savunma mekanizması çoklu-ajan simülasyonu için gerekli geliştirme ortamı tanıtılmıştır. Simülasyon için OMNeT++ INET Çerçevesi kullanılmıştır.

Kuhl ve Sudit [15], siber güvenlik yöntemlerinin test edilmesi için uzun zaman gerektiren ve oldukça maliyetli olan fiziksel ağlara alternatif olarak sanal bir simülasyon modelleme yaklaşımı sunmaktadır. Geliştirilen simülasyon yöntemi siber güvenlik için bilgi füzyon sistemlerini test etmek amacıyla tasarlanmıştır. Geliştirme ortamı olarak Arena simülasyon programı kullanılmıştır. Bu simülatör, bir ağdaki paket akışının ayrıntılarını modelleyemez, ancak kötü niyetli siber saldırıları ve kötü amaçlı olmayan ağ etkinliğini temsil eden simüle edilmiş uyarılar üreterek izinsiz giriş tespit sisteminin davranışını simüle edebilir.

Van Leeuwen ve ark. [16], ağ bilgi sistemleri ve iletişim ağlarının incelenmesi için bir siber güvenlik analiz ve deney ortamı geliştirmiştir. Geliştirilen modelde siber saldırının sanal ve gerçek etkisini ölçmek için sanal makineler, simülasyon ortamı ve gerçek cihazlardan oluşan hibrit bir yapı sunulmaktadır. Gerçek ve sanal cihazlar arasında iletişimde veri transferi için OPNET'in SITL(System In The Loop) aracı kullanılmıştır. Çeşitli ağ cihazlarını içeren yerel ve geniş alan ağlarından meydana gelen güvenli bilgi sistemi tanımlanmıştır. Bu simülasyon ortamı donanım destekli sanallaştırma gerektiren bir altyapı satın almayı ve kurmayı gerektirir; altyapı üzerinde tam kontrol, test ortamının özelliklerinin daha kolay devreye alınmasını sağlar, ancak aynı zamanda yüksek başlangıç maliyetlerine ve sınırlı genişleme esnekliğine yol açar.

Barreto ve ark. [17], Brezilya'nın okyanustaki geniş petrol arama sahasında çalışan helikopterlerin hava trafik kontrol sistemlerinde kullanılan otomatik gözlem yayını ADS-B teknolojisinin (Automatic Dependent Surveillance Broadcast) güvenlik açıklıklarının giderilmesi konulu bir çalışma yapmıştır. Bu makalede sözü geçen kritik altyapı sistemine yönelik siber saldırıların simülasyon araçları kullanılarak değerlendirilmesi ve ölçülmesinde bir vaka çalışması yapılarak sınama ve simülasyon ortamı geliştirilmiştir. Sınama ortamı için Cyber Exata, VR-Forces yazılımları kullanılmıştır.

Torres ve ark. [18], kablolu ve kablosuz tam ölçekli taktiksel sanal ağlarda siber saldırı ve güvenlik yöntemlerinin test ve analizini yapabilen, önceden yapılmış ilgili çalışmalar üzerine kurulu yeni bir simülasyon ortam modeli sunmaktadır. Geliştirilen model ile bir dizi siber saldırı yapılarak belli bir sanal ağ mimarisinin esneklik ve sağlamlığı test edilebilmektedir.

Norman ve ark. [19], siber alanda ağ sistemlerinin test ve deneylerinin geliştirilmesi için bir simülasyon modeli tasarlamıştır. Bu modelde karmaşık ağlarda hızlı ve düşük maliyetli analiz, siber saldırı/faaliyet etkilerini değerlendirilmesi, silah sistemleri üzerindeki siber etkinin değerlendirilmesi ve çeşitli tehdit ve hedef sistemlerin temsil edildiği simülasyon ortamı sunulmaktadır. Proje Amerikan Savunma Bakanlığınca

desteklenmiştir. Modeli geliştirmek için OPNET, Exata, GNS3/Dynamis, Lariat, Breaking Point, GOTS, Emulap, VMWare gibi araçlar kullanılmıştır.

Kotenko ve Chechulin [20], saldırganların tespiti ve bunlara karşı gerçek zamanlı önlemlerin belirlenmesi için siber saldırı grafikleri kullanarak güvenlik değerlendirmesi ve etki analizi sağlayan bir sistem olan CAMIAC'ı sunmuştur. Ancak sistem, saldırı projeksiyonunu daha büyük bir sistemin parçası olarak kullanmaktadır ve araştırma çalışmaları siber saldırılara odaklanmamaktadır.

Moskal ve ark. [21], saldırı davranışı modelleme ve simülasyon konusundaki mevcut çabaları gözden geçirmekte ve siber saldırı davranışı simülasyonu için modüler bir sistem olan CyberSim ile kurumsal ağlarda siber saldırı davranışlarının oluşturulması, simülasyonu ve analizi için uygun bir sistem geliştirmeyi amaçlamaktadır. Saldırgan davranışlarının benzerliklerini veya benzersizliğini daha iyi belirlemek amacıyla saldırı etkinlik dizileri oluşturmak için benzer ve gereksiz uyarıları toplayan bir yöntem sunulmuştur. Saldırılara ait IP adresleri tespit edilememekte ve saldırı engelleme mekanizması bulunmamaktadır.

Ekelhart ve ark. [22], güvenlik analizi neticesinde ortaya çıkan ve çeşitli düşmanlara karşı sistemin direncini deneysel olarak değerlendirmede nasıl kullanılabileceğini gösteren bir prototip uygulamayı tanımlamıştır. Simülasyon aracı, bir dizi saldırı simülasyonu gerçekleştirerek her replikasyon için seçilen sonuç değişkenlerini kaydetmek suretiyle model bir sistem üzerinde saldırı analizleri gerçekleştirir. Bu çalışmada saldırı modelleri yeterince iyi modellenmemiştir, sosyal mühendislik, ağ oluşturma gibi ek davranış modelleri ve saldırı kalıpları eksik kalmıştır.

Bergin [23], otonom araç sistemlerinde siber güvenliğin modellenmesi ve simülasyon desteği için bir siber saldırı ve savunma simülasyon yapısına olan ihtiyacı belirtmiştir. Bu otonom araç sistemleri insansız hava ve kara araçlarını kapsamaktadır. Örnek bir siber saldırı simülatör sistemi ile bu tip modelleri destekleyen bir yapı tanıtılmıştır.

Park ve ark. [24], siber saldırı simülasyon aracı SECUSIM, saldırı mekanizmalarını belirlemek, savunma mekanizmalarını doğrulamak ve sonuçlarını değerlendirmek için F. Cohen'in siber saldırı simülatörünü referans alarak yeni bir araç geliştirmiştir. SECUSIM VisualC++ temel alınarak uygulanmış ve yüzlerce ağ bileşenine karşı yirmi saldırı senaryosunun simülasyonunu gerçekleştirebilmiştir. Bu simülatörlerin her ikisi de saldırgan davranışı uygulamıştır; ancak sonuçları önceden tanımlanmış saldırı adımlarına dayanmaktadır ve simülatörler belirli güvenlik açıklarını hesaba katmamış ve uyarı çıktıları üretmemiştir.

Tablo 1.1. Siber saldırı simülatörlerinin karşılaştırılması

Siber Saldırı Simülatörleri	Kullanılan dil	Kullanılan simülatör	Senaryo sayısı	Modellenebilen düğüm sayısı	Kullanılan ağ türü
Kotenko [14]	C++	OMNET++	N/A	1000	Genel
Park et al. [24]	Visual C ++	SECUSIM	20	1000	Genel
Kotenko and Man'kov [13]	Visual C ++	MASDK	N/A	1000	Genel
Kuhl et al. [15]	Java	Arena	37	1500	Kurumsal
Dennis Lee Bergin [23]	Java	QualNet	6	1000	Mobil
DEVS-CAS (önerilen simülatör)	Java	DEVS-Suite	6	3.500	Kurumsal

Ağ simülasyon araçlarının karşılaştırılması üzerine yapılan çalışmalar, siber saldırı simülatörlerine nispeten daha fazladır. Bu konuda yapılan literatür araştırmasında farklı siber saldırı simülasyon araçlarının senaryo sayıları, modellenebilir düğüm sayıları ve ağ türleri genel olarak karşılaştırılmıştır. Bu tezde, DEVS tabanlı siber saldırı simülatörü (DEVS-CAS) tanıtılmaktadır. Ayrıca yapmış olduğumuz performans analizlerinde bir bilgisayardaki diğer simülatörler ile 1500 düğüme ulaşmak mümkünken, Tablo 1.1.'de görüldüğü gibi DEVS-CAS ile yaklaşık 3000 - 3500 düğüme çıkarılabilmektedir. Dolayısıyla DEVS-CAS'ın daha iyi bir ölçeklenebilirliğe sahip olduğu söylenebilir. Bu tez çalışmasında geliştirilen siber saldırı simülatörünün diğer simülatörlerden daha iyi olan yönleri DEVS-Suite simülasyon yazılımının sağladığı esnek özelliklere dayanmaktadır.

1.3. Motivasyon

Günümüzde bütün teknolojik donanımlar ve onları çalıştıran yazılımlar ile cihazları birbirine bağlayan iletişim ağlarından oluşan ortama “siber ortam” veya “siber uzay” (cyberspace) adı verilmektedir. İletişim ağlarının ve internetin hemen hemen her alanda kullanılması ve günlük hayatta kullandığımız çeşitli cihazların bu ağlarla birbirlerine bağlanabilmesi, farklı bir savaş türünün doğmasına neden olmaktadır. Bu yeni savaş türünde silah olarak bilgisayar virüsleri, yazılım ve donanımlar ve iletişim ağları kullanılmaktadır. Siber alan yaygınlaştıkça, güvenlik hususlarının önem kazanması kaçınılmaz bir durumdur. Güvenliğe verilen önem artmış olsa bile, sistemler bugün artık daha açık ve saldırıya maruz kalma ihtimalleri de daha yüksektir.

Milyarlarca bilgisayar ve bunun birkaç katı kadar cep telefonu bulunmaktadır; bunların çoğu internet üzerinden birbirine bağlıdır. Bilgisayar ve cep telefonları, siber uzaya açılan bir kapı niteliğindedir. Bilgisayar kullanıcılarının çoğunun güvenlikten haberi bile olmayabilmektedir. Bunun sonucunda da bugün milyonlarca bilgisayar, başkalarının bilgisayar veya sistemlerine saldırı yapmak için köle bilgisayar olarak kullanılmaktadır. Elektromanyetik alanın sağladığı imkanlarla birlikte ortaya yeni zafiyet ve tehditlerin çıkması, siber güvenlik risklerini daha da arttıran bir durum oluşturmuştur.

Günlük yaşamımızda her alanda pek çok zorlu iş ve işlemleri, gelişmiş teknoloji ürünlerini bilhassa bilişim teknolojilerini kullanarak kolay ve hızlı bir şekilde gerçekleştirebilmekteyiz. Bu durum teknolojiye olan bağımlılığımızı arttırmıştır. Hayat standartlarını yükselten bu durum beraberinde yeni risk ve tehditleri de getirmiştir. Hızla gelişen yeni teknolojilerden faydalanırken diğer taraftan bu gelişmelerin neden olduğu tehdit ve risklere karşı gerekli tedbirlerin alınması büyük önem arz etmektedir.

Dilimizde “siber alan” veya “siber ortam” olarak da kullanılmakta olan siber uzayın farklı zamanlarda farklı tanımı yapılmıştır. Yeni bir ortam olan siber uzayın kendine

has özellikleri vardır. Bu özelliklerin bilinmesi, bu ortamda siber saldırıların nasıl bir tehdit olduğu konusunda da ipuçları vermektedir. Siber uzaydaki elemanlara erişim ve bu ortamda bir noktadan bir noktaya ulaşım coğrafyadan bağımsız ve neredeyse ışık hızında meydana gelir. Bu yüksek hızdaki erişim düzeyi aynı hızda bir risk ve tehdit etkisine yol açmaktadır. Bu tehditler nükleer tesislerden uydu sistemlerine, sağlık ve ulaşım hizmetlerinden bankacılık hizmetlerine, elektrik, doğalgaz ve trafik hizmetlerine kadar daha pek çok alanı etkileyebilecek bir boyuttadır. Dünyada bu alanların her birinde gerçekleştirilmiş ve sonuç alınmış pek çok siber saldırı örnekleri mevcuttur. Örneğin 2010 yılında çok iyi korunan İran'daki nükleer tesisler Stuxnet adlı bir zararlı yazılım ile fiziksel olarak hasara uğratılmıştır [25].

Siber ortam güvenliğini sağlamak ve siber saldırılarla etkin bir şekilde mücadele edebilmek için siber saldırı mekanizmalarının iyi anlaşılması gerekmektedir. Siber saldırıların türleri ve nasıl yapıldığı ve bu saldırılara nasıl karşı konulacağı konusunda pekçok uygulama ve pratik çalışma yapmak gereklidir. Siber saldırı ve savunma deneylerini fiziksel bilgisayar ağ sistemlerinde gerçekleştirmek zaman alıcı, riskli ve maliyetli olmaktadır. Buna alternatif olarak modelleme ve simülasyon ile pek çok siber güvenlik testleri daha kısa sürede ve çok daha düşük maliyetle kolaylıkla gerçekleştirilebilmektedir. Bu tezde DEVS tabanlı modelleme ve simülasyon yöntemi kullanılarak yaygın olarak gerçekleştirilen bazı siber saldırıların simülasyonları yapılıp test sonuçları ortaya konmuştur.

1.4. Tezin Amacı

Bu tezin kapsamı saldırı tespit sistemlerinin performans ve doğruluğunu arttırmak için bir simülasyon modeli geliştirmek, geliştirilen model ile DEVS-Suite yazılımı altında simülasyon deneyleri yapmak, iyi yapılandırılmış ayrıntılı bir sanal ağ tanımlamak ve saldırı tespit sistemleri için gerçekçi uyarıları sağlamaktır. Böylece şirket ve kuruluşların hassas bilgi varlıklarını depolayan özel ağların güvenliğini sağlamaktır.

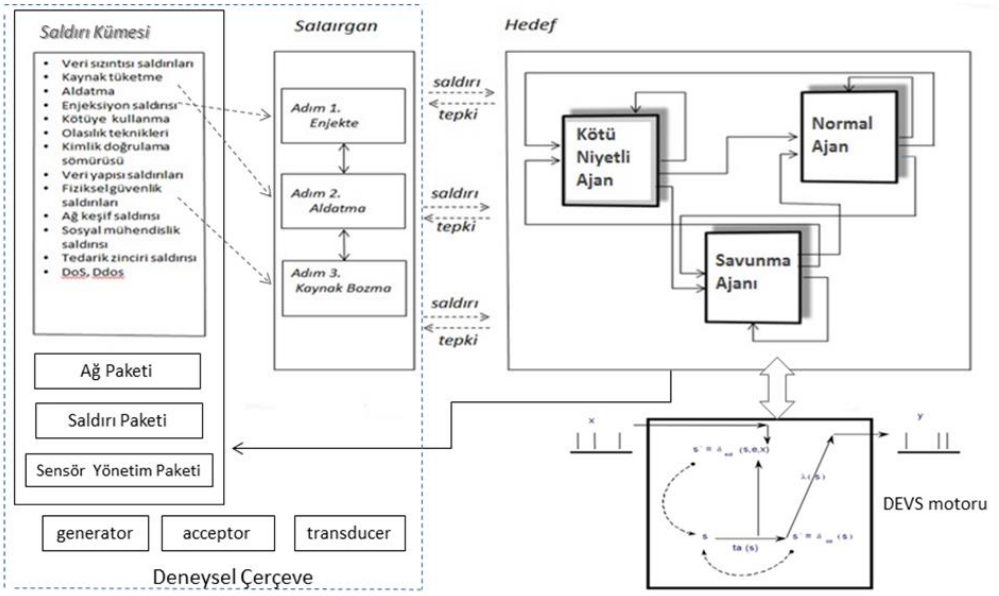
Bu tezin amacı, bilgisayar ağlarında veri güvenliğini sağlamak için siber saldırıların ve güvenlik tehditlerinin değerlendirilmesi ve yeni yöntemler geliştirilmesine olanak sağlayan bir simülasyon aracını geliştirmektir. Siber saldırı ve sensör verilerini elde ederek saldırı tespit sistemlerinin performans ve doğruluğunu arttırmak için bir simülasyon modeli geliştirmek bu tezin ana hedefleridir. Bu simülasyon aracını geliştirmek için Java, DEVS modelleme yaklaşımı ve DEVS-Suite yazılımı kullanılmıştır. Bu çalışmada siber güvenliğin sağlanması için en önemli aşama olarak güvenlik zafiyetlerinin ve açıklıkların kötü niyetli siber saldırganlardan önce bulunup önlem alınması amaçlanmaktadır. Geliştirilen siber saldırı simülatörü ile;

- kullanıcılar ağ topolojisini otomatik oluşturabilecek,
- oluşturulan topolojiler üzerinde saldırı senaryoları oluşturup çalıştırabilecek,
- ağın açıklarını belirleyebilecek,
- uyarı verilerini görüntüleyebilecek ve
- uyarıları doğru bir şekilde tespit edip bu ataklara karşı yeni yöntem ve aksiyonlar geliştirebilecektir.

1.5. Yöntem

Saldırı simülatörünün geliştirilmesi süreci belirli aşamalardan oluşmaktadır. Sistem tasarımı ve analizi için modelleme ve simülasyon hedeflerinin belirlenmesi kavramsal modelleme aşamasında gerçekleştirilmektedir. Temel ağ sentezi aşamasında geliştirilen varlıklar ve düğümler bağlanarak değişik topolojiler ve ağ konfigürasyonları oluşturulmuştur. Saldırı modelleme aşamasında saldırı modelleri kendi karakteristiklerine göre geliştirilerek deneysel çerçeveye eklenmiştir. Geliştirilen modellerin simülasyon deneyleri için DEVS deneysel çerçeve kavramı kullanılmıştır. Saldırı tespit ve engelleme aşamasında saldırı tespit ve engelleme mekanizması eklenerek simülasyonu yapılıp sonuçları gözlemlenmiştir. Saldırı simülasyonu sürecinde saldırı simülasyon testleri yapılıp, sonuçlar gözlemlenip analizleri yapılmış ve grafikler oluşturulmuştur.

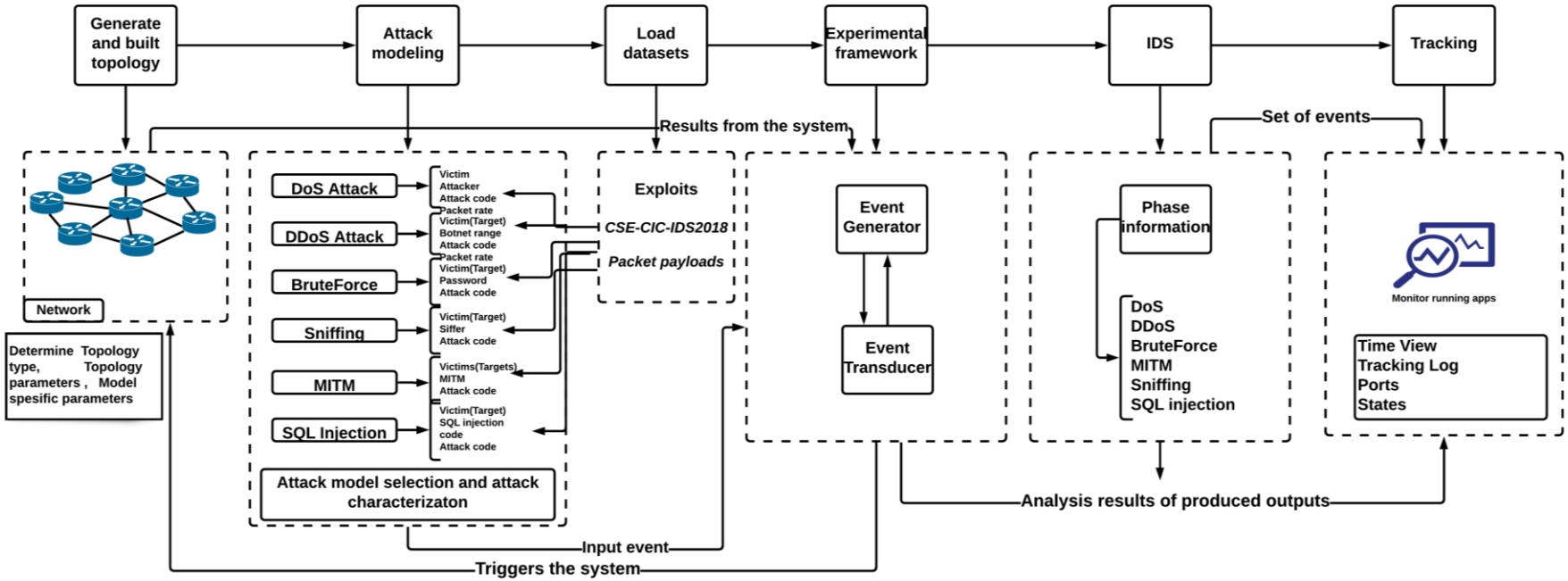
Nihai sonuca ulaşmak için saldırgan ajanın adım adım kullanabileceği pek çok saldırı mekanizması bulunmaktadır. Hedef varlığın kendi savunma yeteneğine veya zafiyetine göre her saldırıya tepkisi farklı olmaktadır. Bu nedenle bir saldırgan ve hedef arasındaki etkileşimleri gözlemlemenin yanı sıra çeşitli saldırı mekanizmaları ve hedeflerin durumunun gözlemlenebildiği Şekil 1.1.'de gösterilen bir simülasyon modeli gereklidir.



Şekil 1.1. Geliştirilen simülatör bileşenleri

Gerçek bir bilgisayar ağında, ağ trafiği, ağdaki aygıtlar arasında taşınan paketlerin tümünü temsil eder. Olası tüm ağ trafiğini modellemek, simülasyonun performansını önemli öçlüde azaltacak bir işlemdir. Ayrıca, bu trafiği modellemek, simülasyona ek bir değer katmaz. Bu nedenle modeller, saldırı ilerlemesinde veya saldırı tespit süreçlerinde yer alan ağ trafiğini içerir.

Geliştirilen saldırı simülasyon modelinin uygulama aşamaları Şekil 1.2.'de gösterilmiştir. Saldırı modellerinin çalıştırılacağı bir ağ modeli oluşturulmuştur. Gerekli ağ yapısı için bir ağ topolojisi üreteci kullanılmıştır.



Şekil 1.2. Siber saldırı geliştirme ortamı ve bileşenleri

DEVS-Suite kullanılarak geliştirilen siber saldırı uygulaması, DEVS-Suite çekirdeğinin üzerine inşa edilmiştir. DEVS formalizmi ve ileri yazılım mühendisliği teknikleri kullanılarak üst düzey performans, ölçeklenebilirlik, teorik sistem tasarımı ve kullanım kolaylığı sağlanmaktadır. Düğümler ve bağlantılar tarafından işlenen olayların durum çizelgeleri ve simülasyon modellerinin davranışları Bölüm 5'te detaylandırılacaktır.

1.6. Tezin Bilime Katkısı

Bu tez çalışmasında ayrık olay tabanlı olarak benzetimi yapılan farklı ölçekteki ağlarda, DEVS tabanlı modelleme ve simülasyon yaklaşımı ile birçok farklı saldırıları yapabilen bir siber saldırı simülasyonu uygulaması gerçekleştirilmiştir. Bu uygulamayı geliştirmek için Java dilinin nesne yönelimli ve gelişmiş yapısından faydalanılarak DEVS-Suite ortamı kullanılmıştır. Siber saldırı simülatörünün, ayrık olay tabanlı modelleme ve simülasyon yaklaşımı ile geliştirilen geniş ölçekli ağlar üzerinde çalışabilecek DEVS formalizasyonu ve sistem tasarımının uygun bir şekilde doğrulamasını ve onaylanmasını kolaylaştırması yönüyle özgün ve yeni bir çalışmadır.

Geniş alanlarda kullanılmasına rağmen ağ sistemlerinde kullanılan teknolojilerin güvenlik altyapısı tam bir koruma sağlayacak şekilde oluşmamıştır [13]. Bunun önemli bir nedeni kapalı kaynak kodlu sistemlerin kullanılmasıdır. Eğitim ve araştırma araçlarının yetersizliği ve kullanım zorlukları da teorik kavramların yerleşmesine olumsuz etki eden eksikliklerdir. Bu tezde bu eksiklikleri giderebilecek bir simülasyon aracı (DEVS-CAS), Java, DEVS modelleme yaklaşımı ve DEVS-Suite yazılımı kullanılarak açık kaynak kodlu olarak geliştirilmiştir. Bu çalışmayla DEVS yaklaşımı kullanılarak esnek, ölçeklenebilir, dağıtık ve paralel mimaride çalışan yeni bir siber saldırı simülatör aracı geliştirilmiştir.

1.7. Tez Planı

Bölüm 1'de problemin tanımı, literatür incelemesi, yapılan tez çalışmasının amacı, simülasyon ve modelleme yöntemi, geliştirilen siber saldırı simülatörünün diğer simülatörlerle karşılaştırması, bilime katkısı ve tez planlaması hakkında bilgi verilmektedir.

Bölüm 2'de siber güvenlik, bilgi teknolojileri güvenliği ile siber tehditler hakkında genel; siber saldırılar ve saldırı alt türleri hakkında ise detaylı bilgi verilerek, saldırılar ve saldırı alt türleri birlikte bir tabloda gösterilmektedir.

Bölüm 3'te bilgisayar ağlarının modelleme ve simülasyon süreçleri ana ve alt başlıklar altında değerlendirilmiştir. Modelleme ve simülasyon teknikleri hakkında bilgi verilerek, sürekli sistem ve ayrık sistem simülasyon modellerinde sistem davranışları açıklanmıştır. Ayrık olaylı sistem tanımlama (DEVS) yaklaşımının bilgisayar ağlarının modelleme ve simülasyonunda daha çok tercih edilmesinin nedenlerine değinilmiştir. Ayrık olaylı sistem tanımlama, temel olarak iki farklı seviyede; atomik DEVS ve birleşik DEVS ile birleşik modellerin oluşturduğu hiyerarşik model davranışları incelenmiştir. Ayrıca simülasyon modellemesinde nesne yönelimli yaklaşımın kullanılmasının prosedürel yaklaşıma göre sağladığı avantajlar hakkında bilgi verilmiştir.

Bölüm 4'te siber güvenlik simülasyon araçları ele alınmıştır. Bu bölümde bilgisayar ağlarını modellemek için akademik araştırmalarda yaygın kullanılan ağ simülatörleri, mimarileri, kullanılabilirlik, ölçeklenebilirlik, taşınabilirlik özellikleri ile istatistikleri ve sistem sınırlamalarının sonuçları incelenmiştir. Bu özellikler çerçevesinde bu simülatörlerin genel özellikleri ve karşılaştırma tablosu gösterilmiştir.

Bölüm 5'te ağ mimarisi hakkında bilgi verilerek siber saldırıları gerçekleştirmek için DEVS yaklaşımı kullanılarak hazırlanmış bir ağ simülasyon aracı (DEVS-CAS) tanıtılmıştır. DEVS-Suite simülasyon geliştirme ortamı ve ağ topolojileri oluşturmak

için kullanılan BRITE topoloji üreteci tanıtılmıştır. Saldırı modelleme yaklaşımı, saldırı simülasyon yöntemleri ve DEVS tabanlı saldırı senaryosu açıklanmıştır.

Bölüm 6'da bu tez çalışmasında simülasyonu gerçekleştirilen siber saldırılar detaylıca açıklanmıştır. Bu saldırıların başında DoS ve DDoS saldırılarının benzetimi yapılmıştır. Diğer bir saldırı türü olarak BruteForce olarak adlandırılan şifre tahmin (password attack) saldırısı simülasyonu gerçekleştirilmiştir. Veri tabanına dayalı uygulamalara saldırmak için kullanılan bir atak tekniği olarak yapılandırılmış sorgu dili enjeksiyonu (SQL Injection) saldırısı tasarlanmıştır. Ağdaki trafik akışındaki paketleri yakalamak ağ dinleme (Sniffing) ve iki taraf arasındaki iletişimi dinleyip iletişimi manipüle eden MITM (Man In The Middle) olarak anılan ortadaki adam saldırılarının benzetimi yapılmıştır. Ayrıca saldırı önleme amacıyla ağ trafiğinin incelendiği ve şüpheli etkinlik belirlendiğinde "refleks" tipi eylemlerin gerçekleştirildiği saldırı tespit sistemlerinin metodolojisine benzer bir yöntemin simülasyonu gösterilmiştir.

Bölüm 7'de simülasyon deney sonuçlarına yer verilmektedir. Bu bölümde DoS ve DDoS saldırı simülasyonlarının sonuçları detaylandırılmıştır. Karşılaştırma ve sonuç grafikleri gösterilip yorumlanmışır. Diğer saldırıların simülasyonlarına ait sonuçlar, "Simülasyonu yapılan saldırılar" konu başlığı olan 6. bölüm altında gösterildiği için bu bölümde ayrıca gösterilmemiştir.

Bölüm 8'de geliştirilen uygulamanın test sonuçları özetlenmekte ve gelecekte bu uygulamanın genişletilerek yapılabilecek çalışmalara fayda sağlayacak öneriler verilmektedir.

BÖLÜM 2. SİBER GÜVENLİK VE SİBER SALDIRI TÜRLERİ

2.1. Siber Güvenlik

Siber güvenlik, bilgisayar ağlarını, donanımları ve değerli verileri yetkisiz erişimlerden veya suç amaçlı kullanımlardan koruma çalışmaları ile bilgilerin gizliliğini, bütünlüğünü ve kullanılabilirliğini sağlama uygulamasıdır [26]. Siber alanda faaliyet gösteren bilgisayar ağları, sunucular, mobil cihazlar, elektronik sistemler veya veri ağları ile bu sistemlerde çalışan yazılımların siber tehditlere karşı korumak için alınan önlemleri kapsamaktadır. Bilgisayar sistemlerine sürekli artan güven, Bluetooth ve Wi-Fi gibi internet ve kablosuz ağ standartları ve akıllı telefonlar, televizyonlar ve çeşitli cihazlar dahil olmak üzere nesnelerin internetini oluşturan akıllı cihazların büyümesi nedeniyle bu alan giderek daha önemli hale gelmektedir. Siber güvenlik, hem politik kullanım hem de teknoloji karmaşıklığı nedeniyle çağdaş dünyadaki önemli zorluklardan biridir [27, 28].

Sistemlerden gerektiği şekilde yararlanabilmek için bilgi sistemlerinin bütünlüğünün sağlanması; sistemin istenen koşullarda çalışmasına bağlıdır. Siber güvenlik, gizliliğin sağlanması amacıyla sistemi oluşturan parçaların, yazılım, donanım ve sistemlerin savunulmasına bir bütün hâlinde katkı sağlar. Bilgisayar ağları dışarıdan erişime açık oldukça bu ağlarda güvenlik riski ortaya çıkmaktadır. Sistem zafiyetleri, teori ve pratik arasındaki farktan doğar. Bir sistem teoride tasarımcının istediği gibi, pratikte ise kodlandığı gibi çalışır. Zafiyetlerin sebebi genellikle bu farktan kaynaklanmaktadır. Bu bölümde, siber tehditler ve siber saldırılar hakkında bilgi verilmektedir.

2.2. Siber Saldırılar

Siber saldırılar, çeşitli teknikler ve araçlar ile güvenlik açıkları istismar edilerek bir sisteme zarar vermek veya normal işlemleri bozmak için gerçekleştirilen eylemlerdir. Saldırganların çok çeşitli amaçları vardır ve hedeflerine ulaşmak için saldırılar başlatırlar. Uzmanlık, kaynak ve motivasyon açısından bir saldırgan tarafından harcanması gereken çabanın ölçümü saldırı maliyeti olarak adlandırılır [29]. Saldırı aktörleri, dijital dünya için tehdit oluşturan kişilerdir [30]. Bu aktörler, bilgisayar korsanları, suçlular ve hatta hükümetler olabilirler [31].

Hükümet web siteleri, finansal sistemler, haber ve medya siteleri, askeri ağlar ve kamu altyapı sistemleri siber saldırıların ana hedefleridir. Literatürde siber saldırı kaynağı iç ve dış olmak üzere iki ana tipte sınıflandırılır. Dahili saldırılar, bir sunucuda hesabı olan veya ağa fiziksel erişimi olan ve bir sistemde ayrıcalıkları veya yetkili erişimi olan kullanıcılar tarafından gerçekleştirilir [32, 33]. Dış saldırılar, ağ etki alanına ait olmayan saldırganlar tarafından gerçekleştirilir. İzinsiz girişlerin ve yapılan saldırının etkisi, ulaşılacak hedeflere bağlıdır. Bireysel bir saldırganın küçük hedefleri olabilirken, organize saldırı gruplarının daha büyük amaçları olabilir. Bireysel bilgisayar korsanları, tek başına çalışan ve yalnızca düşük güvenlikli sistemleri hedefleyen profesyonellerdir [34]. Profesyonel bilgisayar korsanlığı ekiplerinin, organizasyonlarının kaynaklarından veya uzmanlığından yoksundurlar. Organize saldırı grupları, gelişen iletişim ve internet teknolojisine daha çok hakimdirler. Ayrıca bu gruplar teknolojik uygulamalarla, farklı ağların altyapı yönlendirme bilgilerine erişim imkanlarından daha fazla haberdardırlar. Bu grupların motivasyonları oldukça çeşitlidir; hedefleri tipik olarak intikam, ticari sırların çalınması, ekonomik casusluk ve ulusal bilgi altyapısını hedef alan belirli amaçları içerir. Ayrıca, finansal veriler gibi kişisel bilgilerin diğer suç örgütlerine, teröristlere ve hatta hükümetlere satılmasını da içerir. Bireysel hacker hedefleri, boyut veya çeşitlilik açısından nispeten küçüktür ve başlatılan saldırılar, organize gruplar tarafından başlatılanlardan nispeten daha düşük etkiye sahiptir. En etkili organize saldırılar, farklı ülkelerden istihbarat teşkilatları, endüstriyel, siyasi ve askeri casusluk gibi belirli amaçlar için diğer ülkelerin askeri sistemlerini araştırma

amacıyla gerçekleştirdikleri saldırılardır. Ajanlar, amaçlarına ulaşmak için, finansal ve insan kaynaklarının yanı sıra, araştırma ve geliştirme kuruluşlarından donanımsal veya yazılımsal teknolojiler ve metodolojiler sağlamaya kadar çok sayıda uzmana, altyapıya ihtiyaç duyarlar. Bu tür ajanlar, hedeflerini gerçekleştirmek için organize yapılara ve gelişmiş kaynaklara sahiptir ve ağlar için en büyük tehdittir. Bu bölümde gerçekleşen yaygın siber saldırı türleri incelecektir.

2.3. Siber Saldırı Türleri

Literatüre baktığımızda, siber saldırı türlerini açıklamak ve tanımlamak için farklı metodolojiler ve sınıflandırmalar sunumuştur ve siber güvenlik uzmanlarının gelecekteki siber saldırıları tespit etme çalışmaları önem kazanarak devam etmektedir. Bu kapsamda bazı araştırmacılar, ağ ve bilgisayar saldırıları arasındaki ilişkiyi açıklamış ve araştırmalarında ek olarak siber saldırıları; saldırı türü, saldırının hedefi, zararlı güvenlik açıkları ve faydalı yük saldırı türleri gibi dört boyuta ayırmıştır [35]. Burada faydalı yük, paketin, mesajın veya kodun verileri taşıyan kısmıdır. Bilgi güvenliğinde, faydalı yük terimi genellikle kötü amaçlı kodun yıkıcı işlemi gerçekleştiren kısmını ifade eder. Bazı araştırmacılar, saldırı türlerini, gelecek hedeflerini, sınıflandırmanın ölçümünü ve açıklamalarını anlatmak amacıyla siber saldırıların risk değerlendirmesine vurgu yaparak siber saldırıların bir sınıflandırmasını vermiş ve bilgisayar sistemlerindeki kusur ve zafiyetlerin kapsamlı bir analizini yapmışlardır [36]. Farklı araştırmacılar mobil bilgi işlem için güvenlik açıkları ve tehditlerin bir sınıflandırmasını, saldırının alt tipleri, işletim sistemi aygıtı üzerindeki etkisini, özel savunma tekniklerini ve verilen hasarları tanımlayan birçok farklı saldırı türünü açıklamıştır [37-40]. Virüsler, kötü amaçlı yazılımlar, tuş kaydediciler (keylogger), arka planda çalışan gizli programlar (rootkit), casus yazılımlar, solucanlar, truva atları, hizmet reddi (DoS), dağıtılmış hizmet reddi (DDoS), ağ açıkları, uygulama saldırıları, kablosuz saldırılar, sosyal mühendislik, arabellek taşması ve ağ dinleme (sniffing) dahil olmak üzere farklı siber saldırı türlerini ayrıntılı olarak sunan ve öneren çalışmalar da vardır [41-52].

Bu bölümde literatür çalışmalarından elde edilen sonuçlar incelendikten sonra çeşitli kötü niyetli siber saldırıların genel bir sınıflandırması yapılarak saldırı alt tipleri bölüm sonunda bir tabloda gösterilmiştir.

2.3.1. Bilgi toplama ve keşif saldırısı

Genel olarak keşif, bir saldırı başlatmadan önce bir hedef hakkında bilgi toplama eylemidir. Siber saldırganlar, bir ağa veya sisteme saldırmak ve kullanılabilecek bilgileri toplamak amacıyla ağlara ve sistemlere erişmek için hedef hakkında keşif çalışmaları gerçekleştiriler. Saldırganlar tarafından bir organizasyonun kullanılabilecek zayıflıkları, güvenlik açıklarını, etkinlikleri ve düğümleri bulmak için keşif yapılır. Keşfin en yaygın biçimi, e-posta mesajları, web siteleri, sosyal medya siteleri, mesajlaşma uygulamaları ve bir şirketin iç ağı gibi güvenlik açıklarını arayan ağları taramayı içerir ve eski yazılımlar veya saldırılara karşı savunmasız olabilecek güvenlik sistemleri çalıştıran sistemlere ve bilgisayarlara erişim yolları arar. IP adresi, Whois kayıtları, DNS bilgileri, kullanılan işletim sistemi, çalışan e-posta kimliği, telefon numaraları vb. bilgiler toplanır. Bir güvenlik açığı tespit edildiğinde, suçlular kötü amaçlı kod kullanarak bu güvenlik açığından yararlanır. Sistemden bilgi ve istihbarat çalmaya çalışabilirler veya bunu önemsiz (spam) e-postalar göndermek veya başka bir web sitesine karşı hizmet reddi saldırıları (DDoS) başlatmak için kullanabilirler. Bilgisayar korsanları, bu tür keşifleri gerçekleştirmek için genellikle otomatik araçlar kullanır. Saldırgan, daha sonra esas olarak başka bir kişinin bilgisayarına veya bir kuruluşun bilgisayar sistemine erişmek için bir parça kötü amaçlı yazılım kullanarak kodu silah haline getirecektir. Saldırgan daha sonra diğer sistemlere saldırı başlatmak veya kuruluştan veri çalmak için kurbanın bilgisayar sistemini kullanır [53].

2.3.2. Port/bilgi tarama saldırısı

Bilgi tarama saldırısı yalnızca belirli hedefleri tarar. Kullanılan IP adresleri, açılan TCP veya UDP portları, kullanılan işletim sistemlerinin versiyonu, platformu ve hedef ana bilgisayarda başlatılan hizmetler ve işlemler hakkında bilgi almak

amacıyla gerçekleştirilir [54]. Bilgisayar bağlantı noktaları, bir programdan veya internetten ağdaki bir aygıta veya başka bir bilgisayara bilgi akışı için merkezi bir noktadır. Bağlantı noktası numaraları tutarlılık ve programlama için kullanılır. Port numaraları 0 ile 65.536 arasında değişir ve temel olarak popülerliğe göre sıralanır. Bir örnek olarak önemli bazı port numaraları ve bunlara atanan hizmetler şunlardır: Port 20 (UDP), veri aktarımı için kullanılan dosya aktarım protokolünü (FTP) tutar. Port 22 (TCP), güvenli oturum açma, FTP ve port yönlendirme için güvenli kabuk (Secure Shell) (SSH) protokolüne sahiptir. Port 53 (UDP), adları IP adreslerine çeviren etki alanı adı sistemidir (DNS). Port 80 (TCP), World Wide Web HTTP'dir [55].

Belirli bir amaca bağlı olarak, bağlantı noktası taraması için birkaç teknik vardır:

Ping taramaları: En basit port taramalarına ping taramaları denir. Bir ağda, bir ağ veri paketinin hatasız bir IP adresine dağıtılıp dağıtılamayacağını doğrulamak için bir ping kullanılır. Ping taramaları, İnternet Kontrol Mesajı Protokolü (ICMP) istekleridir ve yanıtları tuzağa düşürmek için farklı sunuculara otomatik olarak birkaç ICMP isteği gönderir. BT yöneticileri, bir güvenlik duvarı kullanarak ping taramasını gidermek veya ping taramasını devre dışı bırakmak için bu tekniği kullanabilir. Bu da saldırganların ping yoluyla ağı bulmasını imkansız hale getirir.

Yarı açık veya SYN taramaları: Yarı açık tarama veya SYN (senkronizasyon) taraması, saldırganların tam bağlantı kurmadan bir bağlantı noktasının durumunu belirlemek için kullandıkları bir taktiktir. Bu tarama yalnızca bir SYN mesajı gönderir ve bağlantıyı tamamlamaz. Hedefi asılı bırakır. Hedef cihazlarda potansiyel açık portları bulmayı amaçlayan hızlı ve sinsi bir tekniktir.

XMAS taramaları: XMAS taramaları daha da sessizdir ve güvenlik duvarları tarafından daha az fark edilir. Örneğin, FIN paketleri genellikle TCP 3-yollu el sıkışma ve başarılı veri aktarımı kurulduktan sonra bağlantıyı sonlandırmak için sunucudan veya istemciden gönderilir ve bu, “göndericiden daha fazla veri yok” mesajı ile belirtilir. FIN paketleri genellikle güvenlik duvarları tarafından fark

edilmez çünkü SYN paketleri öncelikli olarak aranır. Bu nedenle, XMAS taramaları, FIN dahil olmak üzere tüm bayraklarla paketler gönderir ve yanıt beklemez, bu da bağlantı noktasının açık olduğu anlamına gelir. Bağlantı noktası kapatılırsa, bir RST yanıtı alınacaktır. XMAS taraması, izleme günlüklerinde nadiren görünür ve bir ağın koruması ve güvenlik duvarı hakkında bilgi edinmenin daha sinsi bir yoludur.

Port tarama sonuçları ağın veya sunucunun durumunu gösterir ve açık, kapalı veya filtrelenmiş olmak üzere üç kategoriden birinde tanımlanabilir.

Açık bağlantı noktaları: Açık bağlantı noktaları, hedef sunucunun veya ağın bağlantıları veya datagramları aktif olarak kabul ettiğini ve dinlediğini belirten bir paketle yanıt verdiğini gösterir. Ayrıca, tarama için kullanılan hizmetin de (tipik olarak TCP veya UDP) kullanımda olduğunu gösterir. Açık portları bulmak, tipik olarak port taramanın genel amacıdır ve saldırı yolu arayan bir siber suçlu için bir zaferdir. BT yöneticilerinin önündeki zorluk, meşru kullanıcıların erişimini sınırlamadan onları korumak için güvenlik duvarları kurarak açık bağlantı noktalarına barikat kurmaya çalışmaktır.

Kapalı bağlantı noktaları: Kapalı bağlantı noktaları, sunucunun veya ağın isteği aldığını, ancak bu bağlantı noktasında hizmet “dinlemediğini” gösterir. Kapalı bir bağlantı noktasına hâlâ erişilebilir ve bir ana bilgisayarın bir IP adresinde olduğunu göstermede yararlı olabilir. BT yöneticileri, açık duruma geçebilecekleri ve potansiyel olarak güvenlik açıkları oluşturabilecekleri için kapalı bağlantı noktalarını yine de izlemelidir. BT yöneticileri, daha sonra "filtrelenmiş" bağlantı noktaları haline gelecekleri kapalı bağlantı noktalarını bir güvenlik duvarı ile engellemeyi ihmal etmemelidir.

Filtrelenmiş bağlantı noktaları: Filtrelenmiş bağlantı noktaları, bir istek paketinin gönderildiğini, ancak ana bilgisayarın yanıt vermediğini ve dinlemediğini gösterir. Bu genellikle bir istek paketinin bir güvenlik duvarı tarafından filtrelendiği ve/veya engellendiği anlamına gelir. Paketler hedef konumlarına ulaşmazsa, saldırganlar daha

fazla bilgi bulamazlar. Filtrelenen bağlantı noktaları genellikle "hedefe ulaşılamaz" veya "iletişim yasak" yazan hata mesajlarıyla yanıt verir.

2.3.3. Hizmet reddi saldırısı (Denial of Service)

Hizmet reddi (DoS) saldırısı, bir saldırgan meşru kullanıcıların bilgisayar sistemlerine, ağlara, hizmetlere veya diğer bilgi teknolojisi (BT) kaynaklarına erişmesini imkansız hale getirdiğinde ortaya çıkan bir güvenlik tehdididir. DoS saldırılarında saldırganlar genellikle sunucuların, sistemlerin veya ağların kaynaklarını tüketen ve başka birinin (meşru kullanıcılar) bunlara erişmesini zorlaştıran veya imkansız hale getiren trafikle doldurur [56].

DoS saldırılarının kurbanları genellikle bankacılık, ticaret ve medya şirketleri gibi yüksek profilli kuruluşların veya devlet ve ticaret kuruluşlarının web sunucularıdır. DoS saldırıları genel olarak önemli bilgilerin veya diğer varlıkların çalınması veya kaybolması ile sonuçlanmasa da, kurbana çok fazla zaman ve paraya mal olabilir.

DoS saldırılarının iki genel yöntemi vardır: hizmetleri engelleme veya hizmetleri çökertme. Taşma (flood) saldırıları, sistem sunucusu arabelleğe çok fazla trafik aldığında, sunucunun yavaşlamasına ve sonunda durmasına neden olduğunda meydana gelir. Popüler taşma saldırıları; arabellek taşması, ICMP taşması ve SYN selini içerir

Arabellek taşması saldırıları, yazılım güvenlik açığının en iyi bilinen biçimlerinden biridir ve hala yaygın olarak kullanılan bir siber saldırıdır. Temel mantığı, bir ağ adresine normal kullanıcıların oluşturduğundan daha fazla trafik göndermektir. Bir arabellek taşması saldırısında, bir uygulama beklediğinden daha fazla girdi alır. Sonuç olarak, hata sistem belleğini kötü niyetli bir tehdide maruz bırakır. Bir arabellek taşması zarar vermese de, bir güvenlik açığı ortaya çıkarır. Tehdit aktörleri daha sonra uygulamanın arabelleğinin ötesindeki bellek konumlarına erişebilir ve bu da onların bu bellek alanına kötü amaçlı kod yazmalarını sağlar. Uygulama yürütüldüğünde kötü amaçlı kod başlatılır.

ICMP taşması, bir saldırganın hedeflenen sistemleri çökertmek için büyük boyutlu ping paketleri gönderdiği bir tür hizmet reddi saldırısıdır. İnternet Kontrol Mesajı Protokolü (ICMP), ağ boyunca paketlerin teslimini engelleyen ağ sorunları hakkında geri bildirim vermenin bir yoludur. TCP gibi üst protokoller, paketlerin teslim edilmediğini anlayabilir, ancak ICMP, "TTL aşıldı" ve "daha fazla parçaya ihtiyaç var" gibi daha farklı sorunları ayırt etmek için bir yöntem sağlar. ICMP protokolü ağ koşullarını bildirmek için çeşitli mesajları göndermek maksadıyla kullanılır. Yanlış yapılandırılmış veya savunmasız bir sistem, IPv4'te belirtilen maksimum 65.535 baytttan daha uzun bir pakette bir ping isteği aldığında, çökmesi muhtemeldir. ICMP taşması, bir saldırganın mevcut herhangi bir bant genişliğini tüketmek ve meşru kullanıcılara erişimi engellemek amacıyla hedef sunucuya büyük miktarda ICMP paketi göndermek için bir botnet kullandığında meydana gelir. Bu saldırı, çok sayıda kaynak kurbanın ağının mevcut tüm bant genişliğini tüketmek için yeterli ICMP trafiği gönderebildiğinde "başarılı" olarak kabul edilir. Bu saldırının bir örneği “ping” komutudur. "Ping" komutu, öncelikle, cihazınızın ağdaki başka bir cihaza, yani bir ağdaki iki nokta arasında veri gönderip alamayacağını kontrol ederek ağ bağlantısını test etmek için kullanılır. Ancak bu komut, ping'i boyut olarak büyütmek ve daha sık gerçekleşmesi için farklı değişkenlerle verilebilir. Bu tür parametrelerin ayarlarıyla oynanması mevcut sistem bant genişliğinin kullanılıp tüketilmesine yol açacaktır.

SYN taşması, bir sunucuya bağlanmak için bir istek gönderir, ancak el sıkışmayı asla tamamlamaz. Tüm açık bağlantı noktaları isteklerle dolana ve meşru kullanıcıların bağlanabileceği hiçbir bağlantı kalmayana kadar devam eder. Saldırgan, genellikle sahte bir IP adresi kullanarak hedeflenen sunucudaki her bağlantı noktasına tekrarlanan ilk bağlantı isteği (SYN) paketlerini gönderir. Sunucu her açık porttan bir SYN-ACK paketi ile her isteğe yanıt verir. Saldırgan ya beklenen ACK'yi göndermez ya da IP adresi sahteyse ilk etapta SYN-ACK'i asla almaz. Her iki durumda da, saldırı altındaki sunucu bir süre SYN-ACK paketinin onaylanmasını bekleyecektir. Bu süre boyunca sunucu, bir RST paketi göndererek bağlantıyı kapatamaz ve bağlantı açık kalır. Bağlantı zaman aşımına uğramadan önce başka bir

SYN paketi gelecek. Bu, giderek daha fazla sayıda bağlantıyı yarı açık bırakır, bu nedenle SYN taşkını saldırılarına "yarı açık" saldırılar da denir. Sonunda, sunucunun bağlantı taşma tabloları doldukça, meşru istemcilere hizmet reddedilir ve hatta sunucu arızalanabilir veya çökebilir. SYN sel saldırıları, bir TCP bağlantısının el sıkışma sürecinden yararlanarak çalışır. Normal şartlar altında TCP bağlantısı, bağlantı kurmak için üç farklı süreç sergiler. İlk olarak istemci, bağlantıyı başlatmak için sunucuya bir SYN paketi gönderir. Sunucu daha sonra iletişimi onaylamak için bu ilk pakete bir SYN/ACK paketi ile yanıt verir. Son olarak, istemci, paketin sunucudan alındığını onaylamak için bir ACK paketi gönderir. Bu paket gönderme ve alma dizisini tamamladıktan sonra, TCP bağlantısı açılır ve veri gönderip alabilir. Hizmet reddi oluşturmak için bir saldırgan, ilk SYN paketi alındıktan sonra sunucunun bir veya daha fazla SYN/ACK paketiyle yanıt vereceği ve el sıkışmasının son adımını bekleyeceği gerçeğinden yararlanır. Bunun için saldırgan, hedeflenen sunucuya genellikle sahte IP adresleriyle yüksek hacimli SYN paketleri gönderir. Sunucu daha sonra bağlantı isteklerinin her birine yanıt verir ve yanıtı almaya hazır bir açık bağlantı noktası bırakır. Sunucu hiç gelmeyen son ACK paketini beklerken, saldırgan daha fazla SYN paketi göndermeye devam eder. Her yeni SYN paketinin gelişi, sunucunun belirli bir süre için yeni bir açık port bağlantısını geçici olarak sürdürmesine neden olur ve mevcut tüm portlar kullanıldığında sunucu artık normal şekilde çalışamaz.

Diğer DoS saldırı türleri, hedefteki sistemlerin çökmesine veya hizmetlerin engellenmesine neden olan güvenlik zaaflarından yararlanır. Güvenlik açıkları, bir saldırgan tarafından bir sistemde yetkisiz eylemler gerçekleştirmek için kullanılabilecek yazılım, donanım yazılımı veya donanımdaki kusurlardır. Saldırganlar, sistemi istikrarsızlaştırmak veya başka kötü amaçlı etkinlikler gerçekleştirmek için bu hatalardan yararlanır, böylece sisteme erişilemeyecek veya kullanılamayacak şekilde girdi gönderilir.

2.3.4. Dağıtılmış hizmet reddi saldırısı

DDoS saldırıları, bir ağ kaynağını veya ağ cihazını normal kullanıcılar için kullanılamaz hale getirmeye çalışır. DoS saldırıları bir kişi veya sistem tarafından yapılırken, DDoS saldırıları iki veya daha fazla kişi veya botlar tarafından yapılır. Bot, kötü amaçlı yazılımlar tarafından oluşturulan, güvenliği ihlal edilmiş bir cihazdır. Ayrıca, bir DDoS saldırısı kurban bir sistemin veya ağ kaynaklarının hizmetlerinin sağlanmasına yönelik, internette çok sayıda güvenliği ihlal edilmiş bilgisayar (botnet) aracılığıyla dolaylı olarak başlatılan büyük ölçekli koordineli bir saldırı olabilir. Saldırgan, saldırı uygulamadan önce çok sayıda bilgisayarı internet üzerinden kontrolü altına alır ve bu bilgisayarlar savunmasız makinelerdir. Saldırgan, komutası altında faaliyete geçmesi için kötü amaçlı kod veya başka bir hackleme tekniği ekleyerek bu bilgisayar zayıflıklarından yararlanır. Gerçek saldıran tarafın belirlenmesi çok zordur, çünkü birçok (çoğunlukla güvenliği ihlal edilmiş) sistemlerin arkasına gizlenmiştir. Modern güvenlik teknolojileri, çoğu DoS saldırısı biçimine karşı savunma mekanizmaları geliştirmiştir, ancak DDoS'un benzersiz özellikleri nedeniyle, hala yüksek bir tehdit olarak kabul edilmektedir [57].

2.3.5. Oturum çalma saldırısı

Oturum ele geçirme, kullanıcının gizli verilerini çalmak ve tehlikeye atmak için önceden oluşturulmuş güvenilir iki sistem bağlantısı arasında izinsiz olarak geçerli oturumu devralma işlemidir [58]. Bu, ortadaki adam saldırısı olarak da bilinir. Herhangi bir web uygulamasında oturum açan biri, üç yönlü el sıkışmayı kullanarak bir istemci ile web sunucusu arasında güvenilir bir oturum oluşturur. Üç yönlü el sıkışma, istemci sistem ile bir web sunucusu arasındaki oturumu kullanarak güvenilir ve geçerli bir bağlantı oluşturmanın yolunu sağlayan güvenilir ve geçerli bir bağlantı kurulduktan sonra yalnızca istemci ve sunucunun birbirleriyle iletişim kurmaya başlaması ve alınan verileri göndermesini sağlayan bir işlemdir. Oturum çalma saldırısında saldırgan, geçerli güvenilir bağlantıyı devralır, paketleri bir sunucuya gerçek bir istemci olarak gönderir, paketi sunucudan alır ve istemciye gerçek bir sunucu olarak gönderir. Oturum çalma saldırısının saldırgan açısından en büyük

avantajı, herhangi bir savunma veya güvenlik duvarını kırmak zorunda olmadan, yalnızca ağı dinlemeye devam etmesi ve geçerli herhangi bir oturumu devralması yeterli olmaktadır.

2.3.6. Web sunucu saldırıları

Web siteleri web sunucularında barındırılır. Web sunucularının kendileri bir işletim sistemi çalıştıran bilgisayarlardır ve çeşitli uygulamaları çalıştıran arka uç veritabanına bağlıdır. Uygulamalarda, veritabanında, işletim sisteminde veya ağdaki herhangi bir güvenlik açığı, web sunucusuna saldırı yapılmasına neden olabilmektedir. Web uygulaması, web sunucusu ile istemci arasında iletişim kurmak için bir arayüz sağlar. Web sayfaları sunucuda oluşturulur ve tarayıcılar bunları istemci tarafında sunar. Veriler, istemci ve sunucu arasında HTTP protokolü aracılığıyla HTML sayfaları şeklinde iletilir. İstemci ve sunucu taraflarındaki güvenlik açıkları bir web uygulaması saldırısına yol açabilmektedir [59]. SOAP, WSDL, UDDI gibi web servis protokollerindeki güvenlik açıkları, SQL enjeksiyonu, XML zehirlenmesi vb. çeşitli saldırılar yapmak için kullanılabilir.

2.3.7. Kablosuz ağlara yönelik saldırılar

İnternetin yaygınlaşmasıyla birlikte iş süreçleri kablolara bağlı kalmadan online olarak yürütülebilmektedir. Kablosuz ağlar, internet teknolojilerinin getirdiği nispeten yeni teknolojilerden biridir. Yaygın olarak kablosuz ağ saldırıları olarak bilinen kablosuz ağları hedef alan sızma ve izinsiz giriş eylemleri ciddi tehditler oluşturmaktadır [60]. Kablosuz ağ saldırıları, ağ üzerinden gönderilen bilgileri yakalamayı ve/veya bilgi trafiğine izinsiz girmeyi amaçlar. Bu saldırıların çok sayıda alt tipleri bulunmaktadır, bunlar bölümün sonunda toplu bir listede gösterilmektedir.

2.3.8. Saldırı tespit sistemi, güvenlik duvarları ve bal tuzağı (honeypot) saldırıları

Saldırı tespit sistemleri, güvenlik duvarları ve bal tuzakları, siber saldırganlardan gelen kötü niyetli saldırıları tespit etmeyi, analiz etmeyi, durdurmayı ve önlemeyi amaçlayan yazılım programları veya donanım cihazlarıdır [61]. Güvenlik duvarı, güvenli olmayan Internet ve güvenli dahili ağ arasında bir ayrım duvarıdır. Güvenlik duvarı, çeşitli kurallar ve kalıplar için gelen ve giden bağlantıları izler ve bunlardan geçen bağlantıları filtreler. Güvenlik duvarını atlatmak için; parçalanmış paketler, açık bağlantı noktaları için güvenlik duvarının ötesini tarayan Firewalking ve güvenlik duvarı yolundan kaçınarak kaynak yönlendirme, HTTP ve ICMP tünelleme yöntemleri kullanılmaktadır. IDS, trafiği izleyen ve ilgili trafik konusunda yöneticiyi uyaran veya bilgilendiren güvenlik sistemleridir. Saldırıyı engellemezler, sadece yöneticiyi uyarırlar. İzinsiz giriş tespiti bir dizi zorlukla karşı karşıyadır. Bir izinsiz giriş tespit sistemi, bir ağdaki kötü niyetli faaliyetleri güvenilir bir şekilde tespit etmeli ve büyük miktarda ağ trafiğiyle başa çıkmak için verimli bir şekilde çalışmalıdır. Ağ tabanlı izinsiz giriş tespiti, en çok kullanılan IDS'lerdir. Bir IDS, kurulu bir yazılım parçası veya fiziksel bir cihaz olabilir. Birçok IDS aracı ayrıca, tespit edilen bir olayı daha sonra gözden geçirilmek üzere bir günlükte saklar. Politikalar veya hasar kontrolü ile ilgili kararlar almak için olayları diğer verilerle birleştirir. Saldırı alt türleri, IDS ekleme, hizmet reddi (DoS), yanlış-pozitif oluşturma ve oturum birleştirme, IDS parçalama, polimorfik kabuk kodu, IP adresi sahtekarlığı tekniklerini içermektedir. Tablo 2.1.'de saldırılar, saldırı alt türleri ile birlikte gösterilmektedir.

Tablo 2.1. Siber saldırı türleri

Saldırı Türü	**Saldırı Alt Türü**
Bilgi Toplama ve Keşif Saldırısı	İnternet bilgi toplama ve keşif saldırıları, Web sitesi bilgi toplama ve keşif saldırıları, DNS bilgi toplama ve keşif saldırıları, Google bilgi toplama ve keşif saldırıları.
Virüs saldırıları	Sistem ve önyükleme kaydı, polimorfik, küme, dosya, makro, kabuk, metamorfik, seyrek bulaştırıcı, dosya, müdahaleci, uzantı, tünelleme ve şifreleme virüsleri.

Tablo 2.1. (Devamı)

Saldırı Türü	**Saldırı Alt Türü**
Truva atı ve Arka kapı saldırıları	Kredi kartı saldırıları, e-bankacılık saldırıları, Http ve Https saldırıları, Botnet saldırıları, ICMP saldırıları, Mobil bilgi işlem saldırıları, Uzaktan oturum açma, erişim saldırıları.
Solucan saldırıları	Toplu posta solucanları ve ağa duyarlı solucanlar.
Port/Bilgi tarama saldırısı	TCP el sıkışma tarama saldırısı, Gizli tarama saldırısı, Xmas tarama saldırısı, FIN tarama saldırısı, Null tarama saldırısı, Idle(boşta) tarama saldırısı, UDP tarama saldırısı.
Numaralandırma saldırıları	Basit dizin erişim protokolü (LDAP) saldırısı, ağ temel giriş/çıkış sistemi(NetBios) saldırısı, basit posta aktarım protokolü (SMTP) saldırısı, basit ağ yönetimi protokolü (SNMP) saldırısı, ağ zaman protokolü (NTP) saldırısı, etki alanı adı sistemi (DNS) saldırısı.
Bilgi sistemi saldırıları	Parola saldırıları: Sözlük saldırıları, kaba kuvvet saldırıları, hibrit saldırılar, hece saldırıları, varsayılan parola saldırıları, elle parola tahmini saldırıları Keylogger saldırıları: Donanım tuş vuruşu saldırıları ve yazılım tuş vuruşu saldırıları Casus yazılım saldırıları: Multimedya casus yazılım saldırıları (ses ve video), masaüstü saldırıları, GPS saldırıları, yazdırma saldırıları, faks saldırıları, USB saldırıları, baskı ekranı yakalama saldırıları ve cep telefonu saldırıları. Rootkit saldırıları: Çekirdek, donanım ve uygulama düzeyinde Rootkit saldırıları.
Sniffer (koklama) saldırıları	Mac sel saldırıları, Mac adresi sızdırma ve çoğaltma saldırıları, Adres Çözümleme Protokolü (ARP) zehirlenmesi saldırıları, ARP sızdırma saldırıları, İnternet Protokolü (IP) sızdırma saldırıları, Alan Adı Sistemi (DNS) Spoofing saldırıları, DNS Önbellek Zehirlenmesi saldırıları, Sahte Dinamik Ana Bilgisayar Yapılandırma Protokolü (DHCP) saldırıları, DHCP aç bırakma saldırıları.
Tampon(arabellek) taşması saldırıları	Stack tabanlı arabellek taşması saldırıları ve Heap tabanlı arabellek taşması saldırıları.
Web sunucusu güvenlik açıkları saldırıları	Web önbelleği zehirleme saldırısı, Http yanıtı bölme saldırısı, Dizin geçişi saldırıları, Http yanıt kaçırma saldırıları, SSH kaba kuvvet saldırıları, Ortadaki adam saldırıları, Web sunucusu parola kırma saldırıları ve web uygulaması saldırıları

Tablo 2.1. (Devamı)

Saldırı Türü	**Saldırı Alt Türü**
İzinsiz giriş tespiti, IDS, Güvenlik Duvarları ve Honeypot saldırıları, arabellek taşması saldırılar	IDS ekleme saldırıları, Hizmet Reddi saldırıları (DoS) saldırıları, Yanlış-Pozitif Oluşturma saldırıları, Oturum Ekleme saldırıları, IDS Parçalama saldırıları, Polimorfik kabuk kodu saldırısı, IP Adresi sahtekarlığı, Ek Güvenlik Duvarından Kaçınma saldırısı
Sosyal mühendislik saldırıları	İnsan tabanlı saldırılar ve siber tabanlı saldırılar
Hizmet reddi saldırıları	Hizmet Reddi (DoS) saldırıları: Bant genişliği saldırıları, SYN taşma saldırıları, ICMP Flooding saldırıları, Peer-to-Peer saldırıları, Botnet saldırıları, Ping of death saldırıları, Teardrop saldırıları, Smurf saldırıları. Dağıtılmış Hizmet Reddi (DDoS) saldırıları: TCP sel saldırıları, UDP sel saldırıları, ICMP sel saldırıları, Amplifikasyon saldırıları, Protokol odaklı açıklardan yararlanma saldırıları, Smurf saldırıları ve Fraggle saldırıları.
Oturum çalma saldırıları	Tarayıcıda adam saldırısı, Oturum sabitleme saldırısı, Sıra numarası tahmini, IP sızdırma saldırıları, Sıfırlama (RST) paket saldırıları, UDP sızdırma saldırısı ve Blind saldırılar.
Web tabanlı uygulama saldırıları	Cross Site Scripting (XSS) saldırıları, SQL Injection saldırıları, Cookie Poisoning saldırıları, Yanlış yapılandırma saldırıları, Platform açıklardan yararlanma saldırıları, Parametre kurcalama saldırıları, Enjeksiyon hatası saldırıları, Komut enjeksiyon saldırıları, LDAP Enjeksiyon saldırıları, Gizli Dosya Manipülasyonu saldırılar, Siteler Arası İstek Sahteciliği (CSRF) saldırıları, DoS saldırıları, Buffer overflow saldırıları, Web hizmetleri saldırıları, XML Zehirlenmesi saldırıları
Kablosuz ağ saldırıları	Medya Erişim Kontrolü (MAC) Adres Sahtekarlığı saldırıları, Hileli Erişim Noktası saldırıları, WEP Enjeksiyon saldırıları, Veri Çerçevesi Enjeksiyon saldırıları, Cracking WEP anahtar saldırıları, Gizlice dinleme saldırıları, Maskeleme saldırıları, Beacon Flooding saldırıları, ARP Önbellek Zehirlenmesi saldırıları, Yönlendirme saldırıları, VPN Girişi kırma saldırılar, Paylaşılan Anahtar Tahmin saldırıları.

BÖLÜM 3. BİLGİSAYAR AĞLARININ MODELLENMESİ

Bilgisayar ağı tasarımları giderek daha karmaşık hale gelmektedir. Ağ simülatör araçlarına başvurmadan ağ performansını analitik olarak değerlendirmek çok zordur. Tasarımcılar bu zorluğu aşmak için, belli bir soyutlama seviyesinde modelleme ve simülasyon yöntemlerinden yararlanırlar. Ağ simülasyonu, gerçek dünya uygulaması dışında çeşitli ağ topolojilerini hesaplamak için kullanılan yaygın ve kullanışlı bir yöntemdir. Simülasyonların temel amacı, davranışları hakkında derinlemesine bir fikir edinmek için tasarım aşamasında gerçek sistemleri kurmak veya sistemleri taklit etmektir. Bir sistemi incelemek için, çeşitli deneysel test senaryolarını gerçekleştirmek için gerçek sistemin kendisi kullanılabilir. Bununla birlikte, birçok durumda, fiziksel sistem üzerinde test senaryolarının yeniden üretilmesi oldukça maliyetli veya tehlikeli olabilir. Diğer durumlarda, ya fiziksel sistemin test edilmesi mümkün değildir, çünkü hala tasarım aşamasındadır ya da testlerin bazı koşullarının gerçek dünyada yeniden üretilmesi mümkün değildir. Sistemin ölçekli bir versiyonunun incelenmesi, kesinlikle tam ölçekli muadili davranışı hakkında bir fikir verebilir. Ancak, ölçekli versiyonunu oluştururken gerçek fiziksel sistemin tüm detaylarının dikkate alındığından emin olunmalıdır. Bu, orijinal sistemin karmaşıklığına bağlı olarak çok fazla zaman ve çaba gerektirebilir. Gerçek sistemle veya bunun ölçekli bir versiyonuyla test yapmak mümkün olmadığında, sistemin bir modelinin kullanılması daha pratiktir.

3.1. Modelleme Süreci

Genellikle bir sistem modeli, sistemin parçalarını bazı varsayımlar altında nicel ve mantıksal olarak birbirine bağlayan bir dizi matematiksel ilişkiden oluşur. Bir sistem modeliyle çalışmak, fiziksel deney maliyeti olmaksızın sistemin değerlendirilmesine izin verme avantajına sahiptir. Ayrıca, dağıtımdan önce sistemin parçalarının yeniden tasarlanmasına da izin verir. Bir modelle aynı test koşullarını birçok kez yeniden

oluşturmak, sistemin kendisi kurmaktan daha kolaydır. Sistemin modeli yeterince basitse, sistemin davranışı, davranışını ve etkileşimlerini yöneten matematiksel denklemlerin analitik bir incelemesi yoluyla çıkarılabilir. Bununla birlikte, sistem çok karmaşık olduğunda, modelinin kendisi de karmaşıktır ve basitleştirme için güçlü varsayımlar kullanmadan veya çözümleri belirli vaka çalışmalarına kısıtlamadan analitik çözümler bulmayı imkansız olmasa da çok zorlaştırır.

3.2. Simülasyon Süreci

Bir sistem modelinden yararlanmanın alternatif bir yolu simülasyonlar gerçekleştirmektir. Analitik bir modeli doğrulamak için simülasyonlar kullanılabilir. Ancak, genellikle basitleştirici varsayımlar gerektiren analitik modelin çözülmesinin aksine, sistemin simülasyon dili kullanılarak tanımlanması, gerçek sistemle ilgili olarak çok ayrıntılı ve doğru olabilir. Simülasyonlar, gerçek sistem üzerinde test etmek için çok karmaşık veya çok pahalı olan senaryoları test etmeyi mümkün kılar. Simülasyonlar, aynı zamanda, gerçek sistemde mümkün olandan daha iyi bir test ortamında daha iyi bir kontrol ile tekrarlanabilir test durumlarına izin verme avantajına da sahiptir. Bununla birlikte, simülasyonlar dezavantajlar ve zorlukları da beraberinde getirir. İlk olarak, simülasyonların anlamlı olması için sistem modelinin simülasyon ortamında mümkün olduğunca yakın bir şekilde yeniden üretilmesi gerekir. Bu, zaman ve çaba açısından maliyetli olabilir. İkincisi, gerçek yaşam sisteminin çalışma ortamını yeniden üretirken de özel bir özen gösterilmesi gerekir. Bu genellikle, gerçek sistemin gelişeceği bir dizi girdi ve fiziksel kısıtlamaya dönüşür. Bu nedenle, çalışma ortamını yeniden üretmek de önemli ölçüde çaba gerektirebilir. Bu, özellikle gerçekçi girdi verileri simüle edilemediğinde ve simülasyonlara girdi olarak beslenecek örnek verileri elde etmek için deney yapılması gerektiğinde böyledir. Üçüncüsü, sistemlerin karmaşıklığı arttıkça, bir sistemi oluşturan bileşenlerin sayısı ve bu bileşenler arasındaki ve çalışma ortamlarıyla olan etkileşimlerin sayısı katlanarak artar. Bu nedenle, simülasyonlarla bu tür sistemlerin tam davranışını incelemek, önemli miktarda bilgi işlem belleği ve işlem süresi gerektiren karmaşık bir görev haline gelir. Son olarak, çoğu zaman sınırlı yetenekler sunabilecek mevcut bir simülasyon ortamıyla çalışılmalıdır. Bu tez

kapsamında modelleme aracı olarak Ayrık Olay Sistem Tanımlaması (DEVS) ve DEVS-Suite Simülasyon ortamı kullanılmıştır. DEVS-Suite Simülasyon çerçevesi 5. Bölümde detaylı olarak açıklanmıştır.

3.3. Modelleme ve Simülasyon Teknikleri

Modelleme ve simülasyon yaklaşımı genellikle durumu zaman içinde sürekli değişen sürekli sistemler ve durumu yalnızca ayrık zamanlarda değişen ayrık sistemler olmak üzere iki farklı sistem kategorisinde ele alınır. Ek olarak, bir sistem modeli deterministik veya stokastik, statik veya dinamik olabilir. Deterministik durumda, belirli bir girdi parametreleri seti benzersiz bir çıktı seti üretirken, stokastik durumda elde edilen çıktı, gerçek çıktının bir tahminidir. Çoğu zaman, stokastik sistemler, modeli bazı olasılıksal unsurlar içeren sistemlerdir. Stokastik sistemler için, bazı parametrelerin rastgeleliği, simülasyonlarda uygulanması gereken önemli bir husustur. Simüle edilmiş bir model, sistemin ilerlemesi zamana bağlı olduğunda dinamik, sistem temsili zamana bağlı olmadığında ise statik olduğu söylenir. Kablolu ve kablosuz bilgisayar ağları, bir mesajın alımı veya iletimi gibi bir olayın meydana gelmesi üzerine, durumların yalnızca ayrık anlarda değiştiği stokastik ayrık olay sistemlerinin önemli bir alanını oluşturur.

Simülasyon amacıyla, bir sistem, birbirleriyle iletişim kuran ve bazı görevleri veya olayları yürüten sanal varlıklardan oluşan bir mantıksal süreç ile modellenebilir. Sistemin durumundaki değişim, zaman içinde sanal varlık durumu değişikliklerinden veya olay yürütmesinden kaynaklanır. Her olayın, meydana geldiği zamanı gösteren (yani, yürütülmesi gereken) ilişkili bir zaman damgası vardır. Temel olarak, simülasyonlardaki bir olay, zaman damgası olarak adlandırılan belirli bir zamanda simülasyon sistemi durumuna yönelik bir güncellemenin bir göstergesini temsil eder.
Ayrık olay sistemlerinin yaygınlaşması ve bunların gerçek hayatta, özellikle bilgisayar ağlarında birçok sistemi temsil etmesi nedeniyle, bu sistemler daha detaylı açıklanacaktır. Bilgisayar ağları çoğunlukla ayrık olay simülasyonu ile simüle edilir. Bu yaygın kullanımın arkasındaki ana neden, bilgisayar ağ protokolleri sonlu durum makineleri olarak modellenebildiğinden, ayrık olay simülasyonunun

bilgisayar ağlarının davranışını temsil etmek için daha iyi uyarlanmasıdır. Bir bilgisayar ağında, iki ardışık olay arasında sabit bir durum vardır ve ayrık olay simülasyonu, bir sabit durumdan diğerine atlamaya izin vererek daha hızlı simülasyonlara yol açar. Ayrık olay simülasyonunun diğer avantajlı yönleri, esneklik ve daha düşük hesaplama yüküdür.

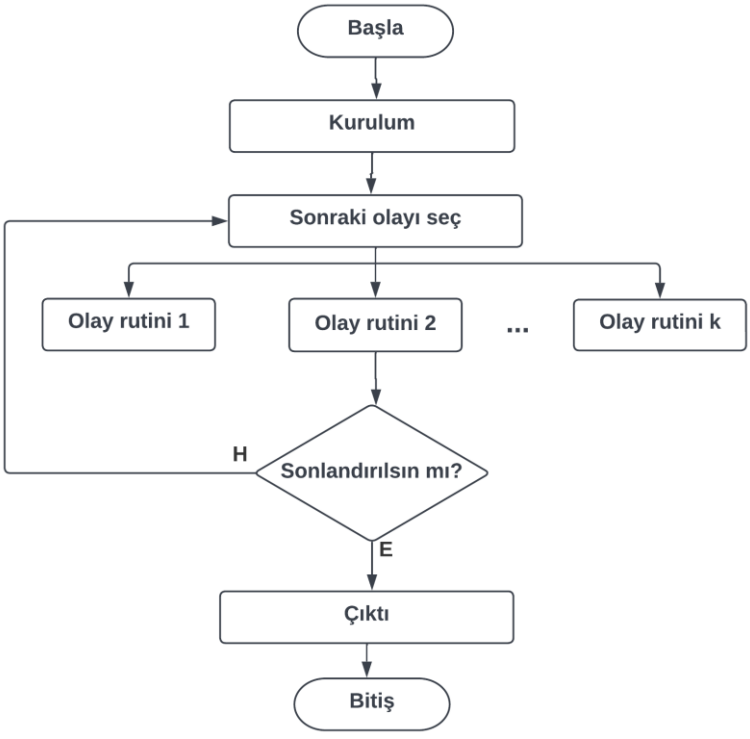
3.4. Sürekli Sistemlerin Simülasyonu

Simülasyon, gerçek fiziki dünyadaki bir sürecin veya sistemin zaman içindeki dinamiklerinin bir bilgisayar kullanılarak taklidini yapan ve sonuçlarını analitik olarak değerlendiren bir tekniktir. Simülasyonlar, gerçek bir sistemin zaman içindeki gelişimini temsil eder. Sistemdeki değişiklikler, gerçek dünyadaki etkilerini tahmin etmek için uygulamadan önce benzetimi yapılır. Sonuçta elde edilenler, istenen modelin özelliklerine ait birer tahmindir [62]. Simülasyon modelleri durum değişkenlerinin zaman içerisindeki değişimine göre sürekli sistem simülasyonu ve ayrık olay simülasyonu olmak üzere durum değişkeninin zaman içerisinde aldığı değerlere bağlı olarak ayrık veya sürekli sistemler olarak ikiye ayrılır. Ayrık olay simülasyonu modellerinde sistem durum değişkeni, olay olarak tanımlanan zamanın belirli noktalarında değişim gösterir. Sürekli sistem simülasyon modellerinde ise sistem davranışını temsil eden durum değişkeni zaman içerisinde sürekli olarak değişim gösterir. Simülasyon modelleri, ister ayrık, ister sürekli, isterse de melez bir yapı göstersin temelde tek bir amaca hizmet ederler. Bu amaç, sistem davranışının ve sistemi meydana getiren öğeler arasındaki ilişkilerin kestirilmesi yolu ile sistem performansının geliştirilmesidir. Ayrık sistemlerde, sistem durumunda değişikliğe neden olan prosesler birbirinden ayrı olaylarla tanımlanan noktalarda meydana gelirler. Sistemin durum değişkeni zamanın bir fonksiyonu olarak sürekli değişim gösteriyorsa, bu tip sistemlere sürekli sistemler denilmektedir. Sürekli sistemlerde önemli olan daha önce de tanımlandığı gibi düzgün ve sürekli değişimlerdir. Genellikle, sürekli sistemlerin tanımı diferansiyel denklemler kümesi gibi sürekli denklemler şeklinde olacaktır, bu denklemler sistem niteliklerinin zaman içerisinde nasıl değiştiklerini göstereceklerdir.

3.5. Ayrık Olaylı Sistem Tanımlama (DEVS) Yaklaşımı

Ayrık Olaylı Sistem Tanımlama (DEVS), yaygın olarak kullanılan kapsamlı bir simülasyon teknolojisidir. DEVS, fiziksel sistemlerin davranışını, yani bilgisayar ağlarında olduğu gibi durumları zamanla gelişen, zaman içinde etkileşime giren varlık koleksiyonlarını temsil etmeye ve incelemeye izin verir. DEVS simülasyonu nesneler ve olaylar olmak üzere iki temel yapı taşı üzerine kurulmuştur. Simülasyon nesneleri gerçek fiziksel nesnelere (varlıklara) karşılık gelmektedir. Olayların ise potansiyel olarak iki işlevi vardır, bunlar; simülasyon nesnesinin durumunu değiştirmek veya gelecekteki olayları programlamaktır. DEVS'te işlenecek olaylar bir liste veya olay takviminde tutulur [63].

Olay planlama zaman ilerleme algoritmasının akış diyagramı Şekil 3.1.'de gösterilmektedir. Algoritma, kurulum, olay işleme döngüsü ve çıktı olarak üç bölümden oluşur.



Şekil 3.1. Ayrık olay simülatörünün akış diyagramı [64]

Kurulum bölümünde saat, varlıklar ve durum değişkenleri kurulur. Ardından simülatör ikinci kısma girer. Olaylar bir döngüde işlenir. Bunun için gelecekteki olay listesinden bir sonraki olay alınır ve tipine bağlı olarak belirli bir olay rutini (işleyici) çağrılır. Olay rutini, durum değişkenlerini ve varlıkları değiştirebilir, istatistikleri güncelleyebilir ve yeni olay bildirimleri oluşturabilir. Döngünün sonlandırma koşulu geçerli olduğunda simülasyon son kısma girer. Çıktı kısmında ise nihayi istatistikler hesaplanır ve gerekirse dosyalara yazılır.

DEVS modelleme yaklaşımı, ayrık olay simülasyonu için yeni bir yaklaşım olarak Zeigler tarafından ortaya atılmıştır [65]. DEVS, bir ayrık olay soyutlaması kullanarak karmaşık dinamik sistemleri modellemek için geliştirilmiş popüler bir formalizmdir. Bu soyutlama düzeyinde, bir sisteme zamanlanmış ilgili "olaylar" girdi dizisi, sistemin durumunda ani değişikliklere neden olur. Bu olaylar harici olarak (yani başka bir model tarafından) veya dahili olarak (yani zaman aşımları nedeniyle modelin kendisi tarafından) oluşturulabilir. Sistemin bir sonraki durumu, sistemin önceki durumuna ve olaya göre tanımlanır. Olaylar arasında sistemin durumu değişmez. Harici geçiş işlevi, giriş bağlantı noktalarından mesajlar alır ve uygun durum geçişlerini başlatır. Dahili geçiş işlevi, mevcut duruma göre durum değişikliklerini tanımlar ve zaman ilerleme işlevi, simülasyon sırasında gerekli zamanlama yapılandırmasını kontrol eder. Ek olarak, çıkış işlevi, veri/kontrol paketlerini çıkış portları üzerinden gönderir. DEVS'e dayalı olarak geliştirilen sistem, atomik ve birleşik bileşenlerin bir bileşimi olarak temsil edilebilir.

Atomik modeller, sistemin davranışını temsil eden temel bloklardır ve düşük düzeyde ayrık olaylı sistemin otonom davranışını tanımlar. Birleşik DEVS, bir sistemi bileşenler ağı olarak tanımlar. Farklı DEVS modelleri daha karmaşık modeller üretmek için özellikle ağ simülasyonları için faydalı olabilecek diğer harici modellerle kolayca entegre edilebilir.

3.5.1. Atomik DEVS modelleme yaklaşımı

Atomik bir model, çevre ile tüm etkileşimini sahip olduğu giriş/çıkış portları ile sağlayan ve sistemin ayrık olaylı davranışının değişik yönlerini tanımlayan bir yapıdadır. Ayrık olay durumunda, olaylar bu tür bağlantı noktalarında görünen değerleri belirler. Modelin dışında ortaya çıkan dış olaylar giriş portlarında alındığında, model açıklamasının bunlara nasıl yanıt vereceğini belirlemesi gerekir. Ayrıca, model içinde ortaya çıkan dahili olaylar, durumunu değiştirmenin yanı sıra, diğer model bileşenlerine iletilmek üzere çıkış portlarında olaylar olarak kendini gösterir. Temel bir model aşağıdaki bilgileri içerir:

- Harici olayların alındığı giriş portları kümesi.
- Harici olayların gönderildiği çıkış portları kümesi.
- Durum değişkenleri ve parametreleri kümesi: iki durum değişkeni genellikle mevcuttur, “faz” ve “sigma” (harici olayların yokluğunda sistem “sigma” tarafından verilen süre boyunca mevcut “faz”da kalır).
- Dahili geçişlerin zamanlamasını kontrol eden zaman ilerleme fonksiyonu, “sigma” durum değişkeni mevcut olduğunda, bu fonksiyon sadece “sigma” değerini döndürür.
- Zaman ilerleme fonksiyonu tarafından verilen süre geçtikten sonra sistemin hangi duruma geçeceğini belirten dahili geçiş fonksiyonu.
- Bir girdi alındığında sistemin durumunu nasıl değiştirdiğini belirten harici geçiş fonksiyonu, sistemi yeni bir “faz”a ve “sigma”ya yerleştirmek ve böylece onu bir sonraki dahili geçiş için programlamaktır.
- Sonraki durum hesaplanırken harici geçiş fonksiyonunu uygulamadan önce dahili geçiş fonksiyonunu ve dahili bir geçiş gerçekleşmeden hemen önce harici bir çıktı üreten çıktı fonksiyonunu uygular.

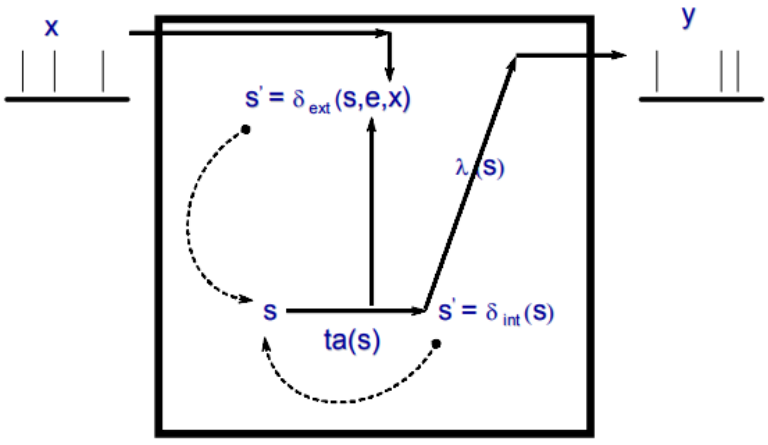
Paralel bir atomik DEVS modeli aşağıdaki yapıdadır:

$M= <X, S, Y, \delta_{int}, \delta_{ext}, \delta_{conf}, \lambda, ta>$, burada;

- X, giriş değerleri kümesi,
- S, durumlar kümesi,
- Y, çıkış değerleri kümesi,
- δint: S→S dahili geçiş fonksiyonu,
- δext: Q x X→S harici geçiş fonksiyonu, burada;

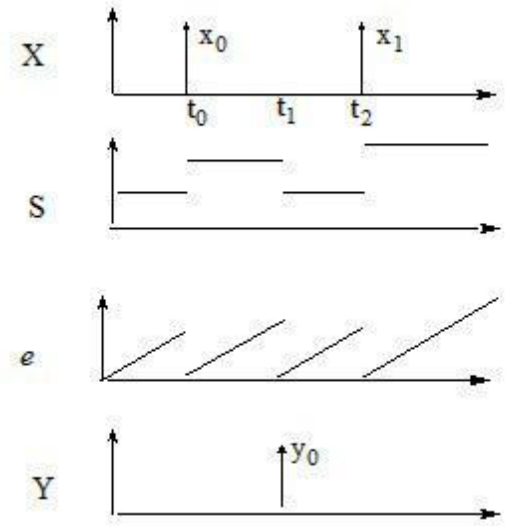
  Q = {(s,e)|s ∈ S, 0≤ e≤ta(s) toplam durum kümesi, e, en son olan geçişten bu yana geçen süre,
- δconf: Q x X→S çakışma (confluent) geçiş fonksiyonu,
- λ: S→Y çıkış fonksiyonu,
- ta: S→R + 0,∞ zaman ilerleme (time advance) fonksiyonu, 0 ve ∞ arasındaki pozitif reel sayılar kümesidir.

Şekil 3.2.'de temel bir DEVS atomik modelinin çalışma mantığı gösterilmektedir. Herhangi bir zamanda 's' durumundaki bir sistemde hiçbir harici olay meydana gelmemişse, sistem ta(s) zamanı süresince durumunu korur. Zaman ilerleme fonksiyonu ta(s) = 0 olduğunda, mevcut 's' durumunun süresi çok kısa olduğundan araya başka olaylar giremeyeceği için bu, geçici bir durum olarak görülebilir. (ta(s) = ∞ olduğunda), dışarıdan bir olay bu durumu değiştirmedikçe sonsuza kadar sistemin 's' durumu değişmez ve bu pasif bir durum olur. Bir durumun süresi dolduğu zaman, iç geçişlerden hemen önce sistem çıkış olarak λ(s) değerini verir ve δint(s) durumuna geçiş yapar [66].



Şekil 3.2. DEVS işleyiş mekanizması [67]

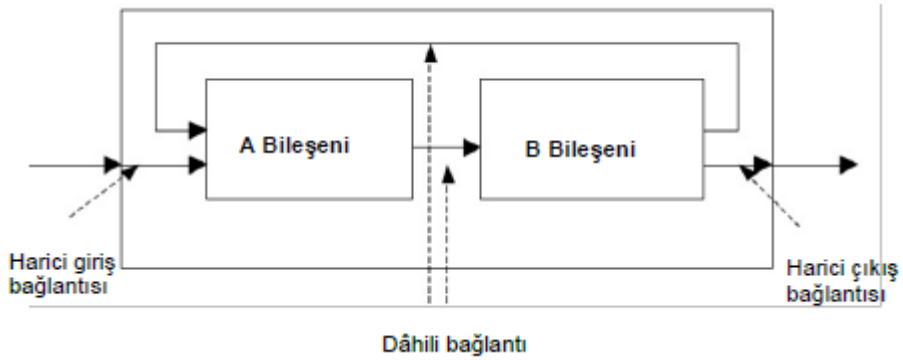
Eğer bir x∈ X dış olayı bitiş zamanından önce oluşursa (sistem (s,e) durumundaysa), sistem δext(s,e,x) durumuna geçiş yapar. Bu nedenle iç geçiş fonksiyonu, son geçişten itibaren herhangi bir olay olmadığı zaman sistemin başka bir duruma geçişine neden olur. Bu durum, giriş, mevcut (şimdiki) durum ve sistemin bu durumunda geçen süre tarafından belirlenir. Şekil 3.3.'te ayrık olaylı sistemde giriş, durum, geçen süre ve çıkışların zamana bağlı değişimleri grafik halinde görülmektedir.



Şekil 3.3. Ayrık olaylı sistemde girişler(X), durum(s), geçen süre(e) ve çıkışlar(Y)

3.5.2. Birleşik DEVS modelleme yaklaşımı

Birleşik model, "atomik" veya "birleşik" modeller ile ve bu modellerin birbirleriyle olan bağlantılarından meydana gelir. Oluşan modelin davranışı ise bileşenlerin bağlantısıyla ve davranışıyla tanımlanabilir. Üç türe ayrılan bağlantılar Şekil 3.4.'de gösterilmiştir.



Şekil 3.4. Birleşik DEVS yaklaşımında bağlantılar

Birleşik bir DEVS modelleme yaklaşımı aşağıdaki yapıdadır;

CM = < X, Y, D, {Mi}, EIC, EOC, IC, Select >, burada;

- *X, Y*: giriş ve çıkış kümeleridir
- *D: birleşik* modelin bileşenler kümesidir
- Her *i*∈ *D* için, *Mi: atomik* veya birleşik olabilen bir bileşenin DEVS modelidir;
- *EIC* ⊆ *X* × ∪i *Xi, harici* giriş bağlantı ilişkisi;
- *EOC* ⊆ ∪i *Yi* × *Y, harici* çıkış bağlantı ilişkisi;
- *IC* ⊆ ∪i *Yi* × ∪j *Xj, dâhili* bağlantı ilişkisi;
- *Select: 2 {Mi}* − ∅ → *{Mi}*, eşitlik fonksiyonudur.

DEVS formalizmi, sistem teorisi temeli üzerine kurulmuş ayrık olaylı sistemlerin genel model tanımlanmasında kullanılır. DEVS formalizmi, sıralı ve paralel/dağıtılmış platformlarda gerçekleştirilebilen karakteristik bir soyut simülasyon motoru mimarisine sahiptir. Akıllı bileşenlere sahip karmaşık teknik ve doğal sistemlerin simülasyon çalışmaları ve uzun vadeli model geliştirme süreçleri için uygundur.

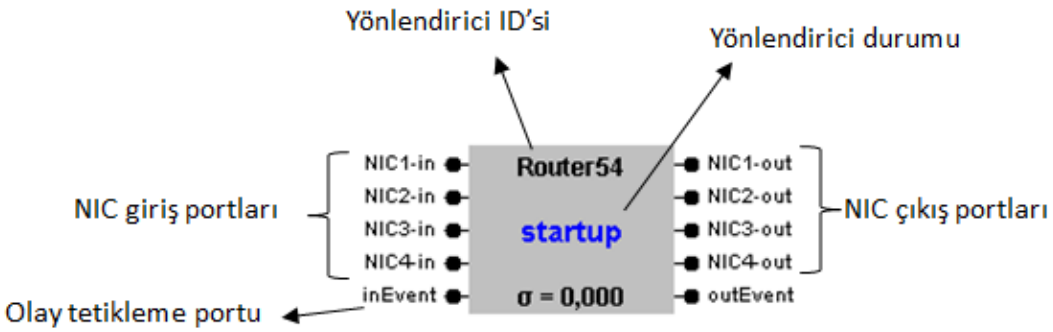
3.6. Nesne Yönelimli Modelleme ve Simülasyon

Simülasyon modellemesinde nesne yönelimli bir yaklaşımın kullanılması, prosedürel yaklaşıma göre birçok avantaja sahiptir. Nesneleri yeniden kullanma ve genişletme özelliği, nesne yönelimli bir çerçevenin geliştirilmesinde merkezi bir özellik ve avantajdır. Ayrıca, nesneler modüler olduğundan, bir nesnenin her bir örneklemeyle ilişkilendirilen bilgiler, gerektiğinde nesnenin konumuna referans verilerek tek bir yerde tutulur [68]. Ayrıca nesneleri temsil etmeye odaklanan bir nesne yönelimli simülasyon ile hesaplamalar ve modelleme mantığı nesneler (sınıflar) arasında bölünebilir. Modelleme fonksiyonlarının bu delegasyonu prosedür bazlı modellemeyle mümkün olandan daha organize bir yapı sağlar [69].

Nesne yönelimli bir simülasyon modelinin geliştirilmesinde ilk adım, uygun yazılım mimarisi kavramını tasarlamaktır. Bu, modelin ve simülasyon yönlerinin ve denetimlerinin üst düzeyde nasıl etkileşim kuracağı konusunda geniş bir tanım içerir. Sarjoughian ve Singh [70], gerekli etkileşimi sağlamak için simülasyon mimarisinde üç ayrı çerçevenin kullanılmasını önermektedir. Bunlara bir model, kontrol ve izleme çerçevesi dahildir. Bu durumda model çerçevesi uygulama işlevselliğini oluşturur ve temsil edilen nesnelerin durumlarını işler. Kontrol, kullanıcı eylemlerini model değişikliklerine bağlar ve kullanıcıya sunmak için belirli "görünümler" veya sistemin durumlarını seçer. Görünüm, modelin oluşturulduğu ve kullanıcı hareketlerinin sağlandığı mimarinin kullanıcı tarafını temsil eder. Ayrık olay simülasyon mimarisi geliştirirken, Brunner ve Schriber [71], ayrık bir trafik veya işlem miktarının sistemin durumunu değiştirmek için bir sistem içinde noktadan noktaya (veya nesneden nesneye) hareket ettiği "işlem akış dünyası görünümü" sağlamanın önemini belirtmektedir. Bu tür bir etkileşimi sağlamak için çok sayıda yaklaşım yapılabilir, ancak kilit nokta bu işlem akışını açıkça tanımlayacak bir çerçeve oluşturmaktır.

Bu çalışmada siber saldırıları gerçekleştirmek için DEVS yaklaşımı kullanılarak hazırlanmış olan ağ benzetim aracı kullanılmıştır [67]. DEVS modelleme yaklaşımının, modüler bir yapıyı ve dağıtık çalışmayı desteklemesi, atomik ve bileşik modellerden oluşan geniş ölçekli karmaşık sistemlerin modellenmesinde kolaylıklar sağlamaktadır. Şekil 3.5.'de simülasyon ortamında gösterilen bir atomik model,

dahili ve harici durum geçiş fonksiyonlarına, giriş-çıkış portlarına, başlangıç durumuna ve durum değişkenlerine sahiptir. Atomik modeller, bir DEVS modelinin temel yapı taşlarıdır. Bir atomik modelinin davranışı, durum geçiş fonksiyonları (dahili, harici ve birleşik), çıkış fonksiyonu ve zaman ilerleme fonksiyonu ile tanımlanır.



Şekil 3.5. Atomik DEVS modeli

Paralel DEVS yaklaşımı ile klasik DEVS yaklaşımında önemli bir problem olan aynı anda meydana gelen olayları yönetememesi durumu çözülmüştür. Bundan dolayı Paralel DEVS (P-DEVS), klasik DEVS yaklaşımının gelişmiş bir sürümüdür denebilir. Paralel DEVS atomik modelinin klasik modelden farkı, çakışma geçiş fonksiyonuna, değerler kümesine ve çoklu giriş / çıkış portlarına sahip olmasıdır. Klasik birleşik DEVS modelinin paralel DEVS modelinden farkı ise seçim fonksiyonuna sahip olmasıdır [72]. 5. Bölümde siber saldırı senaryolarının gerçekleştirildiği DEVS-Suite ortamı ayrıntılı olarak incelenecektir.

BÖLÜM 4. SİBER GÜVENLİK SİMÜLASYON ARAÇLARI

Ağ simülasyonu, bilgisayar ağları alanında tartışmasız en önde gelen değerlendirme metodolojilerinden biridir. Yeni ağ protokollerinin ve iletişim mimarilerinin geliştirilmesi için ana akımdır [66]. Ağ simülatörleri, hem iletişim kanallarını hem de ağ düğümlerinin davranışını tanımlayarak rastgele bir bilgisayar ağının modellenmesini destekler. Örneğin, yeni bir yönlendirme protokolünün özelliklerini araştırmak için genellikle bir ağ simülatörü kullanılır ve yönlendirme davranışı çeşitli topolojilerde incelenebilir. Ağ simülasyonu, ağ protokollerinin işlevsel ve performans analizi için önemli bir kaynak olmuştur. Günümüzde yaygın olarak kullanılan ağ simülatörlerinin sayısı oldukça fazladır ve çeşitli problemlerin ve işlev bozukluklarının üstesinden gelmek için her gün yeni araçlar ve sistemler geliştirilmektedir. Simülatör mimarisi, karmaşık simülasyon modellerinin oluşumunu sağlayan temel bir bileşendir. Simülatörlerin farklı odak noktaları vardır ve araştırma alanları için farklı işlevlerle tasarlanmıştır; bu nedenle mimaride farklılık gösterirler. Çok çeşitli ağ simülatörleri geliştirmiştir. Ancak, belirli bir amaca uygun bir simülatörün seçilmesi, ağ simülatörlerinin kapsamlı bir şekilde incelenmesini gerektirir. Ağ simülatörleri hakkındaki mevcut literatürün sınırlamaları vardır. Sınırlı sayıda simülatör, karşılaştırmaya uygun fonksiyonel ve performans kriterleri ile çalışmalara dahil edilmiş ve uygun simülatör seçimi için makul bir seçim modeli sunulmamıştır. Bu sınırlamaların üstesinden gelmek için, sınıflandırmalar, ek karşılaştırma parametreleri, sistem sınırlamaları ve çeşitli kriterler kullanarak karşılaştırmalar yapmak gerekmektedir. Bu kapsamda simulator mimarisi, dokümantasyon ve kullanım kolaylığı, simüle edilebilecek düğüm sayısı açısından simülatörün ölçeklenebilirliği, simülatörü analiz etmek için sağladığı istatiksel çıktılar, yeniden kullanılabilmek için simülasyon kodu taşınabilirliği ve simülatörün mevcut bilgi işlem kaynaklarını kullanma yeteneği öne çıkan kriterlerdir.

Çeşitli ağları modellemek için çok sayıda simülatör mevcut olsa da, bu bölümde akademik araştırmalarda yaygın kullanılan ağ simülatörleri, mimarileri, kullanılabilirlik, ölçeklenebilirlik, taşınabilirlik özellikleri ile istatistikleri ve sistem sınırlamalarının sonuçları incelenmiştir.

4.1. Simülatörler ve Özellikleri

4.1.1. NS2

NS2 (Ağ Simülatörü Sürüm-2), günümüzde kullanılan en yaygın ağ simülatörlerinden biridir. Özellikle ağ araştırması için tasarlanmış açık kaynaklı bir nesne yönelimli ayrık olay ağ simülatörüdür. NS2, TCP, FTP, UDP, HTTPS ve DSR gibi fonksiyonların ve protokollerin hem kablolu hem de kablosuz simülasyonu için destek sağlar. NS2 iki anahtar dilden oluşur; C++ ve nesneye yönelik araç komut dili (OTcl). C++, simülasyon nesnelerinin dahili mekanizmasını (yani bir arka uç) tanımlarken, OTcl, ayrı olayları zamanlamanın yanı sıra nesneleri bir araya getirerek ve yapılandırarak simülasyonu kurar. C++ ve OTcl, TclCL kullanılarak birbirine bağlanır. NS2 ile protokol etkileşimi, tıkanıklık kontrolü, ağ dinamiklerinin etkisi, ölçeklenebilirlik gibi farklı konuları araştırmak amacıyla simülasyonlar gerçekleştirilebilmektedir. Topoloji boyutu, yoğunluk dağılımı, trafik üretimi, üyelik dağılımı, gerçek zamanlı üyelik değişimi gibi ağ dinamikleri içeren senaryolar çalıştırılabir ve metin tabanlı veya animasyon tabanlı simülasyon çıktıları alınabilir [73].

4.1.2. NS3

NS3 simülatörü, ağ araştırmaları ve eğitimi için açık, genişletilebilir bir ağ simülasyon platformu sağlamak üzere geliştirilmiştir [74]. NS3 ayrık olaylı bir ağ simülatörüdür. GNU GPLv2 altında lisanslanarak araştırma ve geliştirme için kullanılmaktadır. NS3, paket veri ağlarının nasıl çalıştığına dair modeller ve kullanıcıların simülasyon deneyleri yapması için bir simülasyon motoru sağlar. NS3 kullanım nedenlerinden bazıları, gerçek sistemlerle gerçekleştirilmesi daha zor veya

mümkün olmayan çalışmaları gerçekleştirmek, yüksek kontrollü, tekrarlanabilir bir ortamda sistem davranışını incelemek ve ağların nasıl çalıştığını öğrenmektir. Bazı simülasyon platformları, kullanıcılara tüm görevlerin gerçekleştirildiği tek bir entegre grafik kullanıcı arayüzü ortamı sağlarken, NS3 bu konuda daha modülerdir. NS3 ile çeşitli harici animatörler ve veri analizi ve görselleştirme araçları kullanılabilir. Ancak, kullanıcılar komut satırında ve C++ ve/veya Python yazılım geliştirme araçlarıyla çalışabilmeleri gerekmektedir.

4.1.3. QualNet

QualNet(Quality Networking), Scalable Network Technologies (SNT) tarafından C++ ile yazılmış GloMoSim'in ticari bir versiyonudur. QualNet, protokoller tasarlamak, ağ senaryoları oluşturmak ve performanslarını analiz etmek için kapsamlı bir ortam sağlamaktadır. Bir ağ değerlendirme yazılımı olan QualNet, sonlu durum makinesi olarak modellenmiştir [75]. QualNet, katmanlı bir mimari üzerine tasarlanmıştır. Uygulama, taşıma, MAC ve fiziksel katmanlardan oluşur. Hem kablolu hem de kablosuz ağları ve karışımını simüle edebilir. NS simülatörünün aksine Qual-Net, 802.11s taslağı dahil olmak üzere birden fazla kablosuz teknolojiyi simüle edebilir. Grup hareketlilik modeli, rastgele yol noktası modeli ve iz tabanlı modeller dahil olmak üzere birçok mobilite modelini destekler. QualNet, WiFi, Sensör ağları, MANET ve WiMAX gibi çeşitli ağları simüle etmek için kapsamlı bir kütüphaneye sahiptir.

4.1.4. NetSim

NetSim, ağ laboratuvarı deney ve araştırmaları için kullanılan stokastik bir ayrık olay ağı simülasyon aracıdır [76]. Protokol modelleme ve simülasyon için bir ağ simülasyon yazılımı olup, bilgisayar ağlarını analiz etmeye olanak tanır. Çok yönlü özellik ve işlevselliğe sahip olan NetSim, kullanıcı kodu ile NetSim'in protokol kitaplıkları ve simülasyon çekirdeği arasında arayüz görevi gören yerleşik bir geliştirme ortamı ile birlikte kullanılmaktadır. Ağ, alt ağ, düğüm ve ayrıntılı bir paket izleme gibi çeşitli soyutlama seviyelerinde ağ performans ölçümleri sağlar.

NetSim'in standart ve akademik versiyonları mevcuttur ve yüksek seviyeli mimari ve kodun ortak bir tasarım çerçevesi üzerine inşa edilmiştir. Kurumsal ağ topolojisinin bir simülasyonunu oluşturmak veya tasarımı yapılan ağdaki cihazları kullanmadan sorun giderme pratiğine yardımcı olmak için kullanılabilir. Grafiksel seçeneklerle performans karşılaştırması sağlayan yerleşik bir analiz çerçevesine sahiptir. Veri ve kontrol paketi akışı, NetSim'in yerleşik paket animatörü aracılığıyla görselleştirilebilir ve kullanımı kolaydır.

4.1.5. OMNeT++

OMNeT++, hem kablolu hem de kablosuz ağları simüle etmek için açık kaynaklı, genişletilebilir, modüler, bileşen tabanlı bir ayrık olay simülatör aracıdır [77]. Tamamen C++ ile yazılmıştır. Çoğunlukla araştırma ve eğitim amaçlı ve bilimsel çalışmalar için kullanılmaktadır. Eclipse tabanlı bir IDE, bir grafik çalışma ortamı ile birlikte birçok araç sunar. Birbiriyle etkileşime giren cihazlardan oluşan herhangi bir sistemi simüle edebilen genel amaçlı bir simülatördür. OMNeT++, bileşen tabanlı, hiyerarşik, modüler ve genişletilebilir bir mimari sağlar. Bileşenler ve modüller C++ ile programlanır ve daha sonra yüksek seviyeli bir dil (NED) kullanılarak daha büyük bileşenler ve modellerde birleştirilir. OMNeT++, geniş GUI desteğine sahiptir ve modüler mimarisi sayesinde simülasyon çekirdeği (ve modelleri) uygulamalara kolayca yerleştirilebilir. Simülasyon çekirdeği kütüphanesinin yanı sıra simülasyon ortamı, simülasyon için bir grafiksel ağ düzenleyicisi (GNED), bir NED derleyicisi, grafiksel ve komut satırı arayüzlerini içerir. OMNeT++ ayrıca paralel dağıtık simülasyonu desteklemektedir.

4.1.6. OPNET

OPNET, iletişim ağlarını, ağ cihazlarını, protokolleri ve uygulamaları incelemek için esnek bir şekilde farklı ağ topolojileri oluşturmaya ve simülasyona olanak tanır [78]. Kullanıcılar için nispeten daha güçlü grafiksel destek sunar. Grafik düzenleyici arayüzü, uygulama katmanından fiziksel katmana kadar ağ topolojisi ve varlıkları oluşturmak için kullanılabilir. Topolojilerin konfigürasyon ve simülasyon sonuçlarını

sezgisel ve görsel olarak sunulabilmektedir. OPNET, ayrık olay simülasyon mekanizmasına dayanmaktadır. Ağları düzenlemek için hiyerarşik bir yapı kullanılır. OPNET, ağ yöneticilerin ağlarını ve gelecekte yapmak istedikleri uygulamaları analiz etmelerini sağlayan farklı araçlara (NetDoctor, ACE ve MVI) sahiptir.

4.1.7. J-SIM

J-SIM, nicel sayısal modeller oluşturmak ve bunları deneysel referans verilerine göre analiz etmek için geliştirilmiş Java tabanlı bir ayrık olay simülasyon sistemidir [79]. Otonom bileşen programlama modeli kavramı üzerine inşa edilmiştir. J-Sim'in Java'da otonom bileşen mimarisiyle birlikte uygulanması, J-Sim'i platformdan bağımsız, genişletilebilir ve yeniden kullanılabilir bir ortam haline getirir. J-Sim, Perl, Tcl veya Python gibi farklı komut dosyası dilleriyle entegrasyonuna izin veren bir komut dosyası arabirimi sağlar.

4.1.8. GloMoSim

Küresel mobil bilgi sistemi simülatörü (Global Mobile Information System Simulator - GloMoSim), hem kablosuz hem de kablolu ağları simüle etmek için kullanılabilen parallel ayrık olaylı bir ağ protokolü simülatörüdür. Simülasyon protokolleri Parsec derleyicisi kullanılarak derlenir. Parsec, ayrık olay simülasyon modellerinin sıralı ve paralel yürütülmesi için geliştirilen C tabanlı bir simülasyon dilidir. GloMoSim, Java ile yazıldığı için platformdan bağımsız bir görselleştirme aracına sahiptir. MANET (Proaktif, reaktif ve hibrit yönlendirme protokolleri), WSN, VANET, DTN (Gecikme toleranslı Ağ) ve WMN'yi içeren ad hoc ağlar için geniş modellere sahiptir [80].

4.1.9. DEVS-Suite

DEVS-Suite, DEVS formalizmine dayalı, ayrık olaylı ve açık kaynak kodlu olarak geliştirilen genel amaçlı bir simülasyon ortamıdır [81]. DEVS-Suite, modelleri ve etkileşimlerini grafiksel, etkileşimli bir ortamda temsil etmek için kullanılan Java tabanlı bir uygulamadır. DEVS Suite, dağıtık ve paralel çalışmayı destekleyen,

simülasyon sonuçlarının daha iyi gözlenebilmesi için bazı eklentiler içeren DEVSJAVA benzetim aracının yeni bir sürümüdür. Bu tez çalışmasında kullanılan DEVS-Suite simülasyon ortamı 5. Bölümde ayrıntılı olarak incelecektir.

4.2. Simülatörlerin Karşılaştırılması

Farklı özelliklere sahip ağları modellemek için burada anlatılanların dışında çok sayıda simülatör mevcuttur, bu bölümde akademik araştırmalarda yaygın kullanılan ağ simülatörleri, mimarileri, kullanılabilirlik, ölçeklenebilirlik, taşınabilirlik özellikleri ile istatistikleri ve sistem sınırlamalarının sonuçları incelenmiştir ve bu özellikler çerçevesinde bu simülatörlerin karşılaştırma tablosu Tablo 4.1.'de gösterilmiştir. Ayrıca bu simülatörlerin genel özellikleri de Tablo 4.2.'de gösterilmiştir.

Tablo 4.1. Özelliklerine göre simülatörlerin karşılaştırma tablosu

Simülatör Adı	Simülasyon Olay Türü	Mevcut Modül	Ölçeklenebilirlik	Düğüm sayısı	Paralellik	Açıklama
NS2	Ayrık olay	Kablolu, Kablosuz, Ad Hoc ve Kablosuz Sensör Ağları	Sınırlı	3000'e kadar	Yok	Protokol simülasyonu ve ağ araştırması için özel olarak tasarlanmıştır.
NS3	Ayrık olay	Kablolu, Kablosuz, Ad Hoc ve Kablosuz Sensör Ağları	Sınırlı	-------	-------	Eğitim ve araştırma amaçlı kullanım için tasarlanmıştır.
QualNet	Ayrık olay	Kablolu ve Kablosuz ağ (WiFi, Sensör ağı, MANET, WIMAX, vb.)	Geniş	500-20000	Var	Yeni protokol tasarlamak ve performanslarını analiz etmek için tasarlanmıştır.
GloMoSim	Ayrık olay	Kablolu, Kablosuz ve Geçici Ağlar	Geniş	10.000'e kadar	Var	Protokol yığını için modüler simülasyon sağlar.
NetSim	Ayrık Olay	Kablolu, Kablosuz sensör ağı (kablosuz LAN, WiMAX)	Geniş	-------	-------	Bilgisayar ağlarını analiz eden protokol modelleme ve simülasyon yazılımıdır.

Tablo 4.1. (Devamı)

Simülatör Adı	Simülasyon Olay Türü	Mevcut Modül	Ölçeklenebilirlik	Düğüm sayısı	Paralellik	Açıklama
OMNET++	Ayrık Olay	Kablolu, Kablosuz, Ad-hoc ve Kablosuz Sensör Ağları.	Orta	--------	Var	Akademik ve araştırma alanlarında genişletilebilir ve bileşen tabanlı bir analiz ve ayrık olay simülasyon sağlar
OPNET	Ayrık Olay	Kablolu, Kablosuz, Geçici ve Kablosuz Sensör Ağları.	Geniş	290 düğüme kadar	Var	Karmaşık senaryoları kolaylaştırmak için esnek GUI sağlar
JSim	Sayısal metodlar	Kablolu ve kablosuz sensör ağları.	Orta	1000 düğüme kadar	Var	Genel ağ bileşenleri için özel olarak tasarlanmıştır.
DEVS-Suite	Ayrık Olay	LAN, MAN, WAN, Kablolu, Kablosuz, Ad-hoc ve Kablosuz Sensör Ağlarının modellenme ve simülasyonu.	Çok büyük	100.000'e kadar	Var	Büyük ölçekli dinamik ağlarda yönlendirme şemalarının hızlı simülasyonunu sağlar.

Tablo 4.2. Genel özelliklerine göre simülatörlerin karşılaştırma tablosu

Simülatör Adı	Lisans türü	Dil	Desteklenen İşletim Sistemi	GUI Desteği	Belgeleme	Kullanım kolaylığı
NS2	Açık kaynak	C++ ve OTCL	GNU/Linux,FreeBSD, Mac OS X, Windows	Sınırlı	Çok iyi	Zor
NS3	Açık kaynak	C++ ve Python bağlamaları	GNU/Linux,FreeBSD, Mac OS X, Windows	Var	Çok iyi	Zor
GloMoSim	Açık kaynak	C	Windows, Linux, Sun SPARC Solaris	Sınırlı	Zayıf	Zor

Tablo 4.2. (Devamı)

Simülatör Adı	Lisans türü	Dil	Desteklenen İşletim Sistemi	GUI Desteği	Belgeleme	Kullanım kolaylığı
QualNet	Ticari (Akademisyenler ve diğerleri için ayrı lisans)	C++	UNIX, Windows, MAC, Linux	Var	Çok iyi	Orta
NetSim	Tescilli	C ve Java	Windows	Var	Çok iyi	Kolay
OMNET++	Açık kaynak	C++	Windows, Linux, Mac OS X,	Var	İyi	Kolay
OPNET	Ticari (endüstriyel amaçlar için)	C ve C++	Windows	Var	İyi	Kolay
J-SIM	Açık kaynak	Java	Windows, MAC OS X, Linux.	Var	Orta	Kolay
DEVS-Suite	Açık kaynak	Java	Windows, MAC OS X, Linux.	Sınırlı	İyi	Orta

BÖLÜM 5. YENİ BİR SİBER GÜVENLİK SİMÜLATÖRÜ TASARIMI (DEVS-CAS)

5.1. Ağ Mimarisi ve DEVS-Suite Simülasyon Ortamı

Bilgisayarın haberleşebilmesi, bilgisayar ağlarının ortaya çıkmasını sağlamıştır. Bilgisayar ağlarının boyutu, yönetilmesinin güçlüğü ve kurulumunun zaman alıcı ve maliyetlerinin yüksek olması nedeniyle bu sistemler üzerinde farklı amaçlarla testler yapmak zor ve riskli olduğundan çeşitli denemeler için bilgisayar ağlarının modellenmesi yoluna gidilmektedir. Gerçek bir sistemi temsil etmek için modelleme yöntemi kullanılmaktadır. Bilgisayarların modellenmesi, bilgisayar ortamında, gerçek bir sistemin bilgisayar aracılığıyla benzetiminin yapılmasıdır. Böylece mevcut sistem üzerinde istenilen herhangi bir çalışma, sisteme zarar vermeden mümkün olacaktır.

Bir ağın birçok giriş noktası vardır. Bu giriş noktaları, ağ geçidi olarak kabul edilebilecek cihazlara ek olarak, ağı oluşturan donanım ve yazılımları içerir. Ağı savunmak için bu giriş noktalarını dikkate almak zorunludur. Bu amaçla, ağ trafiğini izlemek ve ağa yetkisiz erişimi engellemek için güvenlik cihazlarına ihtiyaç vardır. Bu cihazlar ile ağ farklı seviyelere ayrılmalı ve dış tehditler mümkün olduğunca en aza indirilmelidir.

Özel bir bilgisayar ağının yapısı çeşitli şekillerde olabilir. En sade haliyle tüm ağ aygıtlarının birbirleriyle doğrudan iletişim kurmalarını sağlayan tek bir merkezi ağ cihazına bağlanabilmesidir. Ancak böyle bir ağdaki herhangi bir cihazda güvenlik riski oluştuğunda ağın genel güvenliği tehlikeye girer. Bir ağ kurmaya yönelik modern yaklaşım, ağ içinde birkaç farklı düzey veya katman oluşturmaktır. Bu düzeylerin her biri belirli bir ağ aygıtı grubundan oluşur ve ağın daha derin seviyelerine harici bir kaynaktan erişmek daha zordur.

Benzetim ortamı içerisinde cihazları temsil etmek için nesneler kullanılır ve bunlar saldırganların yararlanmaya çalıştığı hedeflerdir. Her makinenin, işlevselliğini ve erişilebilirliğini yansıtan önemli sayıda öznitelikleri olabilir. Bu özelliklerin hepsini modellemek çok zaman alıcı olacaktır. Bu nedenle, model, üretilen saldırı ve saldırı tespiti uyarılarıyla ilişkili veya önemli bir etkiye sahip olan temel makine özelliklerine odaklanır.

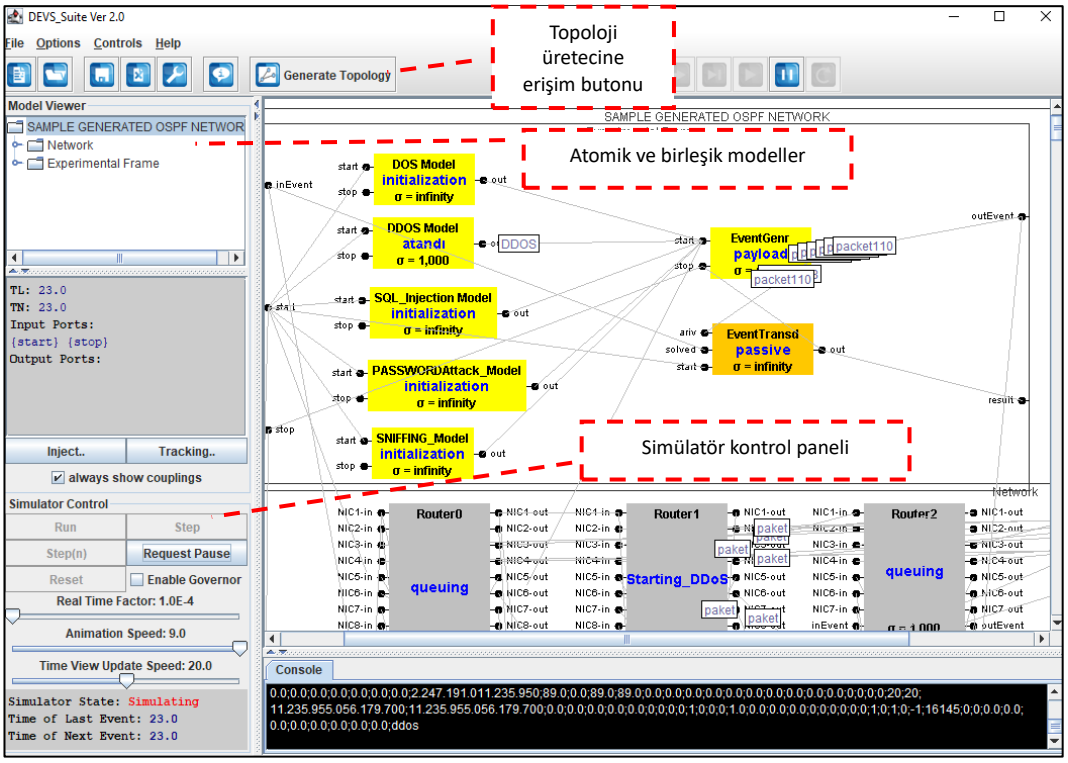
Bir ağın genel yapısının, ağ trafiğinin ve siber saldırıların nesne yönelimli programlama yaklaşımıyla simülasyon modellemesi, nesne sınıflarını yeniden kullanabilme ve kolay geliştirme yeteneği sayesinde geliştiricilere büyük kolaylık sağlar. Nesneler modüler olduğundan, kodları başka bir projede eklemek ve yeniden kullanmak kolaydır [82]. Her fonksiyonun nesne olarak soyutlandığı bir simülasyon ile hesaplamalar ve modelleme fonksiyonları nesneler arasında bölünerek daha organize bir yapı sağlanır.

Bu çalışmada, siber saldırıları gerçekleştirmek için DEVS yaklaşımı kullanılarak geliştirilmiş bir ağ simülasyon aracı kullanılmıştır. DEVS formalizmi kullanılarak geliştirilen paralel ve dağıtık benzetim algoritmalarına sahip DEVS tabanlı ağ simülasyon aracı sistemin geliştirilmesinde kullanılmıştır [83].

Ayrık Olay Sistem Tanımlama (DEVS) yaklaşımı, sistem adı verilen matematiksel bir nesneyi tanımlamanın bir yoludur. DEVS yaklaşımı, ayrık olay sistemlerinin modellenmesi ve analizi için ilk olarak 1976 yılında Dr. Bernard P. Zeigler tarafından 'Theory of Modeling and Simulation' adlı kitabında tanıtıldı . DEVS, ayrık olay tabanlı, modüler ve hiyerarşik bir simülasyon yaklaşımı olarak son zamanlarda diğer yaklaşımlardan daha fazla öne çıkmıştır [84].

DEVS yaklaşımının nesne yönelimli programlama teknikleri kullanarak geliştirilmiş çok sayıda yazılım uyarlaması vardır. DEVS-Suite, Java programlama dilinin gelişmiş özelliklerini kullanarak, DEVS yaklaşımı ile gerçekleştirilen karmaşık ağ sistemlerinin davranışını görüntüleyen genel bir modelleme ve simülasyon aracıdır.

DEVS-Suite ortamında ağ çerçevesi ve saldırı modelleri arayüzü Şekil 5.1.'de görüldüğü gibi DEVSJAVA simülasyon aracının yeni versiyonunda gösterilmiştir. DEVS-Suite benzetim aracını diğer araçlara göre öne çıkaran en önemli etkenlerden biri Java programlama dilinin esnek yapısı ile tasarlanmış olmasıdır. Ağ topolojileri oluşturmak için BRITE topoloji oluşturma aracı uygulamaya entegre edilmiştir.



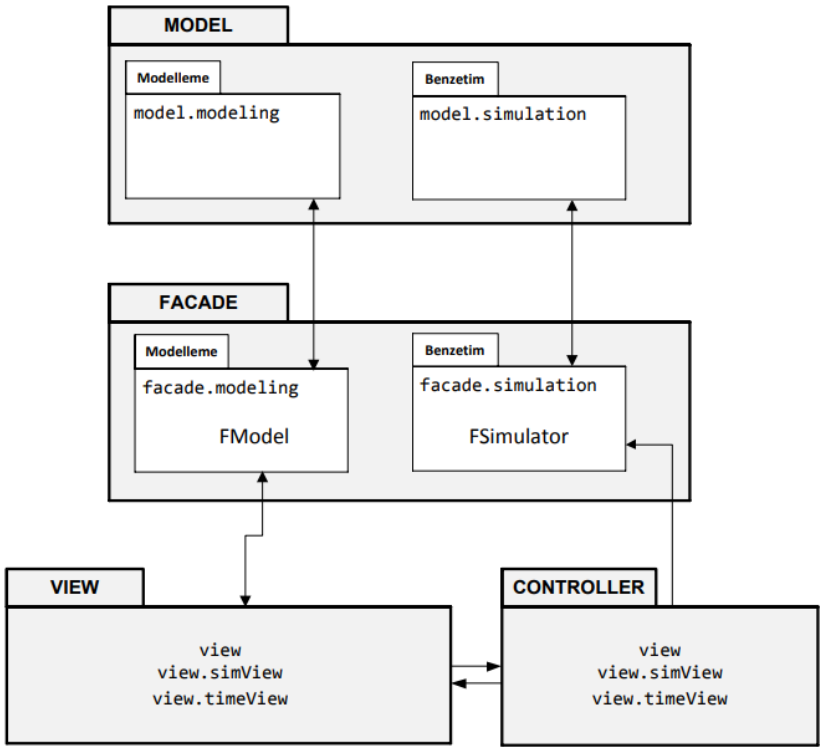
Şekil 5.1. DEVS-Suite ortamında ağ çerçevesi ve saldırı modelleri arayüzü

DEVS-Suite arayüz ekranında model görüntüleyici, simülatör kontrolü, SimView ve izleme penceresi olarak dört ana bölüm vardır. Bir model yükledikten sonra, sol üst köşedeki model görüntüleyici, bu modelde bulunan hem atomik hem de bağlı bileşenlerin bir listesiyle doldurulur. Bileşen listesinin hemen altında, kullanıcı tarafından seçilen modele ait önceden tanımlanmış değişkenleri listeleyen bir kutu bulunur. Bu kutu, kullanıcı başka bir bileşen seçtiğinde güncellenecektir. Hemen altında iki düğme vardır: “Inject”, simülasyonda rastgele zaman noktalarında el ile veri sağlamak için, “Tracking” ise seçilen düğümlerin veri görselleştirme pencerelerini açmak için kullanılır. DEVS-Suite ile gerçekleştirilen benzetim, adım

adım veya sürekli çalıştırılabilir ve bir sonraki (TN) veya en son (TL) olay zamanı izlenebilir.

DEVS formalizminin ADEVS, DEVS-Suite, DEVS/C++, DEVSJAVA, JAMES II, SmallDEVS, DESS/DEVS, P-DEVS, RT-DEVS, Cell-DEVS, Fuzzy-DEVS olmak üzere birçok uyarlaması mevcuttur [85,86].

Şekil 5.2.'de görüleceği üzere DEVS-Suite benzetim ortamı, MVC mimarisine ek bir tasarım şablonu eklenerek geliştirilmiştir. DEVS-Suite paket yapısı Model, Controller, View ve Façade olmak üzere birbiri ile etkileşen sınıfları barındıran paket ve alt paketlerinden oluşmaktadır.

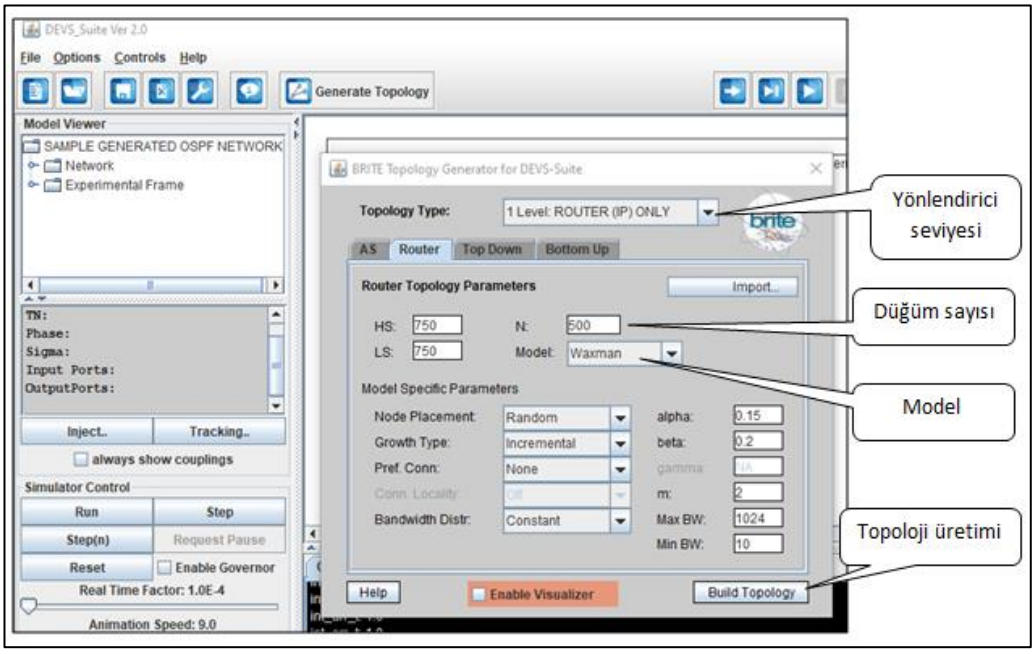


Şekil 5.2. DEVS-Suite MFVC paket yapısı

Façade; bileşenlerin dış dünya ile iletişimini ve giriş / çıkış portlarını kullanmasını sağlayan pakettir. Model; uygulamanın iş katmanını temsil eder. Kod sayfasında görünen veriler modelde tutulmaktadır. Görünüm; uygulamanın sunum katmanını temsil eder. Modelin içerdiği verileri görselleştirmek için kullanılır. Controller, hem

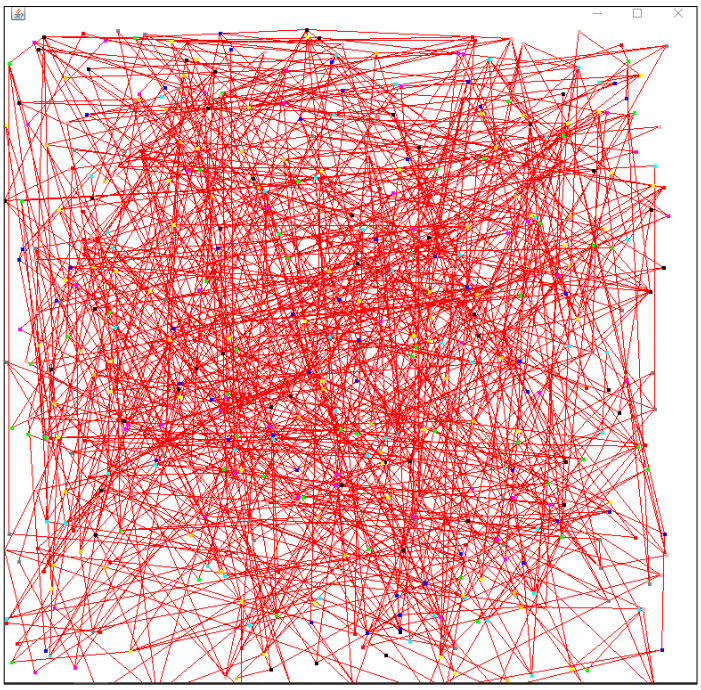
modelde hem de görünümde çalışır. Uygulama akışını, yani model nesnesindeki veri akışını yönetmek ve veriler değiştiğinde görünümü güncellemek için kullanılır.

Ağ simülasyon çalışmaları, ağ topolojisinin temel özelliklerini yeniden üreten iyi topoloji oluşturma modelleri gerektirir. Bu tür modellerin simülasyonlarda kolay ve etkin bir şekilde kullanılabilmesi de bir gerekliliktir [87]. Bu çalışmada, evrensel bir topoloji oluşturma aracı olan BRITE kullanılmıştır. BRITE topoloji üretim aracının ekran görüntüsü Şekil 5.3.'te verilmiştir. Bu arayüz aracılığıyla kullanıcı, model parametreleri, girdi dosyaları ve dışa aktarma formatlarını belirleyerek üretim sürecini yönetebilir. 500 düğümlü bir ağ topolojisinin BRITE görüntüleyicisindeki ekran görüntüsü ise Şekil 5.4.'de gösterilmiştir.



Şekil 5.3. BRITE topoloji üreteci ekran görüntüsü

BRITE, herhangi bir topoloji üretme yöntemiyle sınırlı olmayan ve çoklu nesil modelleri destekleyen esnek bir topoloji üreteci olacak şekilde tasarlanmıştır [88]. BRITE, kullanıcı tarafından elle yazılabilen veya BRITE'ın GUI'si tarafından otomatik olarak oluşturulabilen bir yapılandırma dosyasından üretim parametrelerini okuyarak topolojileri oluşturur.



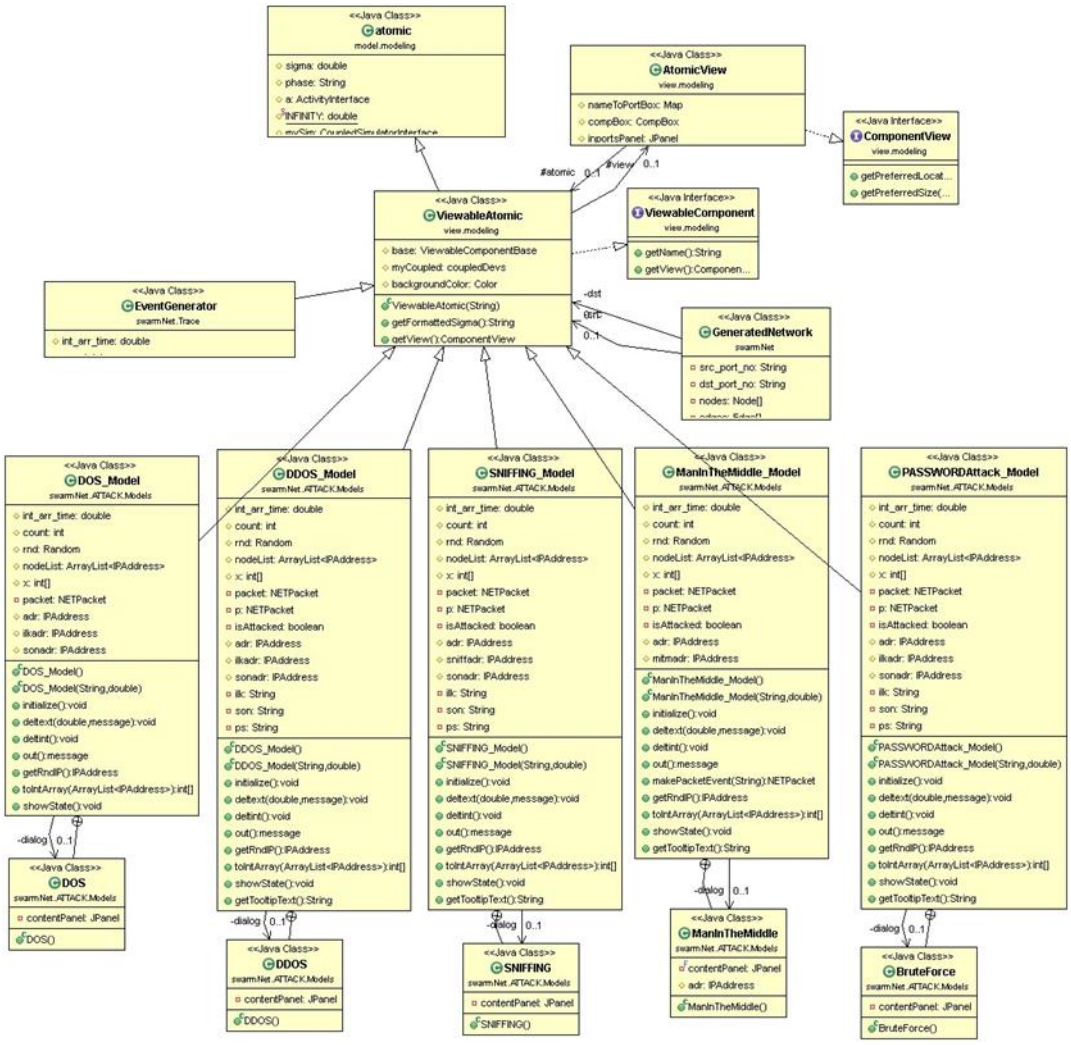
Şekil 5.4. 500 düğüm için BRITE görüntüleyici

5.2. Saldırı Modelleme

Bir saldırı simülasyon modelinin geliştirilmesi bu modelin ağ yapısının ve saldırı senaryolarının geliştirilmesi, bir dizi giriş ve çıkış seçeneklerini ve ağları tanımlama ve saldırı senaryolarına özgü özelleştirilmiş bir arabirimi içerir. Model, genellikle gerçekçi ortamları modellemede istenen özelliklere sahip nesne yönelimli programlama dilleri kullanılarak geliştirilmektedir. Siber saldırı simülatörü ile bir kullanıcı belirli bir ağ topolojisi oluşturabilir veya yükleyebilir, ağın güvenlik açıklarını belirleyebilir, saldırı senaryoları oluşturabilir, çalıştırabilir ve üretilen algılayıcı uyarı verilerini görüntüleyebilir. Simülatörün işlevselliği için birçok giriş ve çıkış gereklidir.

Simülasyonlar genellikle mevcut sistemleri değerlendirmek için incelenebilecek bazı gerçek dünya durumunun doğru bir modelini sağlamaya çalışmaktadır. Dolayısıyla, bir simülasyon çerçevesinin geliştirilmesinde nesne yönelimli programlamanın kullanılması, simülasyon modellerinin gerçek nesneler arasındaki etkileşimi temsil etmeye eğilimli olduğu göz önüne alındığında birçok yararı vardır. Nesne yönelimli programlama, nesneleri temsil etmede sınıfların kullanımına ek olarak, modülerliği

ve tekrar kullanılabilirliği sağlamada yardımcı olan miras, kapsülleme ve polimorfizm gibi temel özelliklere sahiptir. Nesneler modüler olduğundan, bir nesnenin her bir örneklemeyle ilişkilendirilen bilgileri, gerektiğinde nesnenin konumuna referans verilerek tek bir yerde tutulabilir ve simülasyon modelleme mantığı sınıflar arasında uygun şekilde bölünebilir. Modelleme işlevleri prosedür tabanlı dillerle mümkün olandan daha organize bir yapı sağlar.



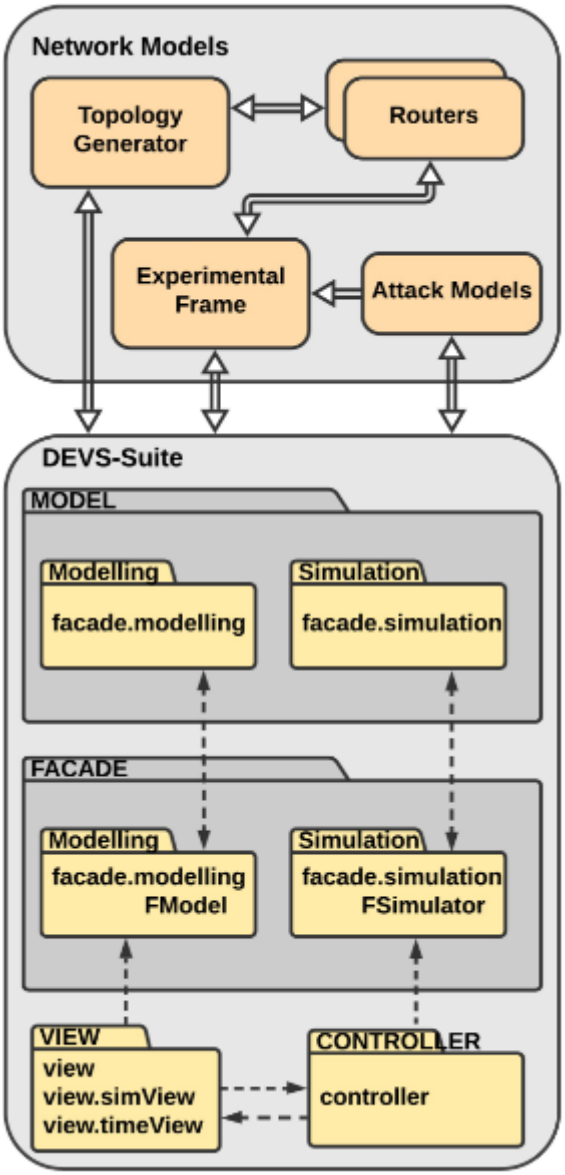
Şekil 5.5. Saldırı modelleri sınıf diyagramı

Siber saldırı simülatör sınıfları, Şekil 5.5.'deki gibi sınıfların etkileşimine dayalı olarak belli sayıdaki paketler halinde organize edilmişlerdir. Bunlar genel olarak; simülasyon paketi, ağ paketi, saldırı paketi, görsel paket, izleme ve yönetim paketleridir. Amaca göre farklı paketler de eklenebilmektedir. Saldırı paketi, saldırı

olaylarını simüle eden sınıfları içerir. Ağ paketi, sanal bir bilgisayar ağın topolojisini ve ağ cihazlarını tanımlamak için kullanılan sınıfların kümesini içerir. Ağ paketi fiziksel ağ cihazlarını ve bu cihazlarda yazılım ayarlarını kapsayan sınıflardan oluşmaktadır. Saldırı paketi, ağ üzerinde bir dizi saldırı senaryoları oluşturmak için gerekli sınıfları içerir. Aynı zamanda mevcut saldırı eylemleri ve saldırı uyarıları için veri tabanı olarak hareket eden sınıfları da içerir. Görsel paket, simülatör kullanıcısına bir ara yüz sağlamak için kullanılan sınıfları içerir. Bu sınıflar grafik kullanıcı ara yüzü oluşturmak, ağı görselleştirme, veri giriş formları ve sonuçların görüntülenmesine yardım etmektedir. İzleme yönetim paketi, performans parametrelerini tanımlayan ve uygulayan sınıfları içerir.

5.3. Saldırı Simülasyonu Metodolojisi

Modellenen ağdaki belli bir süre boyunca bir dizi saldırıyı simüle etmek için ilk olarak o ağ için bir saldırı senaryosu oluşturmak gerekir. Simülasyon modelleri, kullanıcı tanımlı siber saldırı oluşturma yöntemlerini içerir. Her model ağ, özellikle o ağ için tanımlanmış bir dizi saldırı senaryosu içerebilir; Ancak, simülasyon yalnızca bir seferde seçilen bir senaryoyu işler. Saldırı paketi, uygulamada ayrıntılı saldırıların belirlenmesi ve simüle edilmesi için araçlar sağlamaktadır. Nihai sonuca ulaşmak için saldırgan ajanın kullanabileceği pek çok saldırı mekanizması bulunmaktadır. Hedef varlığın kendi savunma yeteneğine veya zafiyetine göre her saldırıya tepkisi farklı olmaktadır. Bu nedenle bir saldırgan ve hedef arasındaki etkileşimleri gözlemlemenin yanı sıra çeşitli saldırı mekanizmaları ve hedeflerin özelliklerini temsil edebilecek bir simülasyon modeli geliştirilmiştir. Gerçek bir bilgisayar ağında, ağ trafiği, ağdaki aygıtlar arasında taşınan paketlerin tümünü temsil eder. Olası tüm ağ trafiğini modellemek, simülasyonun performansını önemli ölçüde azaltacak bir işlemdir ve tüm bu trafiği modellemek, simülasyona ek bir değer katmaz. Bu nedenle modeller, saldırı ilerlemesinde veya saldırı tespit süreçlerinde yer alan ağ trafiğini içerir.

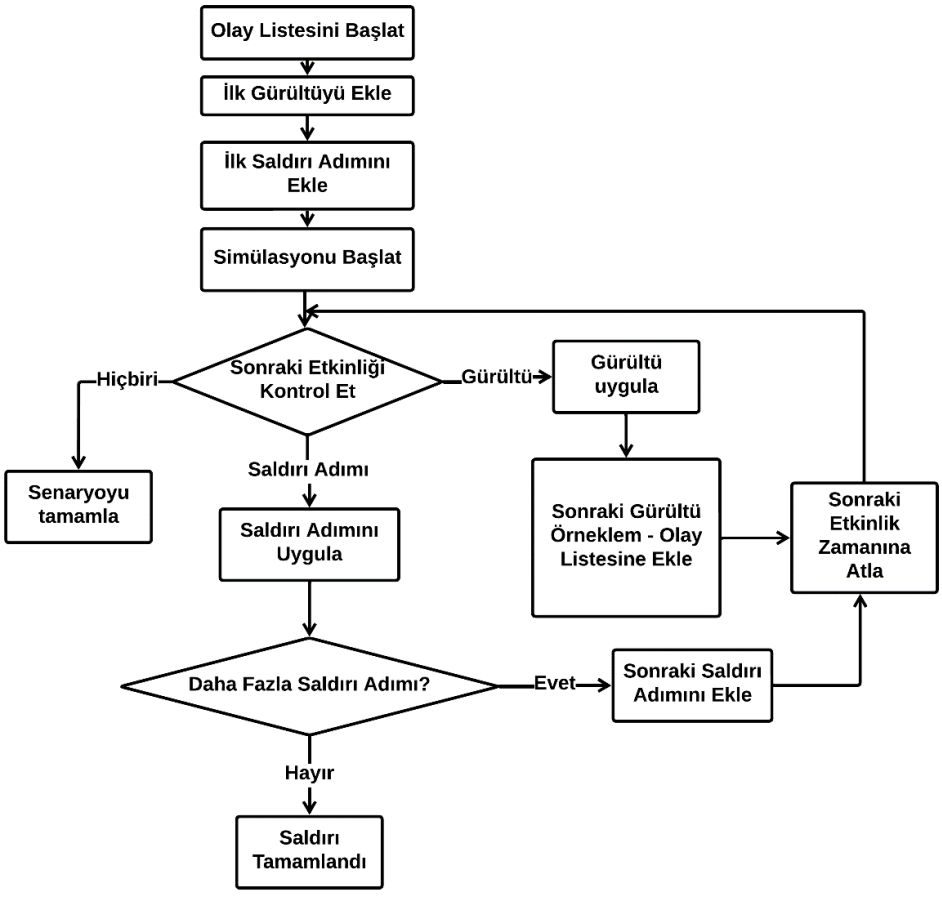


Şekil 5.6. Kavramsal modeller ve modelleme metodolojisi

Bu çalışmada, DEVS-Suite altında topolojik olarak yapılandırılmış ve tasarlanmış sanal büyük ölçekli ağ sistemine karşı siber saldırı gerçekleştirmek için yaygın olarak kullanılan siber saldırı türleri kullanılmıştır. Bu kapsamda, Şekil 5.6.'da görüldüğü gibi DEVS tabanlı dağıtık büyük ölçekli ağ simülasyon modeline saldırı modelleri entegre edilmiştir. Bu modeller; DoS, DDoS, BruteForce, SQL Enjeksiyonu, ortadaki adam (Man in the Middle) ve dinleme (Sniffing) saldırı modelleridir. Geliştirilen saldırı simülatörü, birçok saldırı türünü simüle edebilecek bir altyapı sağlayacak şekilde yapılandırılmıştır. Daha fazla saldırı simülasyonu, gelecekteki tehditlere karşı

daha etkili önlem alınmasına neden olacaktır. Geliştirilen aracın atakları daha kısa sürede tespit etme olasılığı ortaya çıkmaktadır [89].

Simülatör, saldırı senaryolarını yürütmek için ayrık olay simülasyonu kullanır. Saldırı adımları ve gürültü de dahil olmak üzere bir dizi trafik nesnesi, simülasyon için olay listesi olarak kullanılır. Bu trafik nesneleri dizisi, diziye olay ekleme ve kaldırma yöntemlerini içeren bir olay sırası sınıfının parçasıdır. Bir simülasyonun başlatılması sırasında, boş bir trafik nesneleri dizisini içeren yeni bir olay sırası nesnesi oluşturulur. Senaryodaki her saldırının ilk saldırı adımı ve gerekirse gürültü uyarısı, artan başlama zamanı sırasına göre olay sırası dizisine yerleştirilir. Şekil 5.7.'de, bir saldırı senaryosunun ayrık olay simülasyonunu yönetmek için kullanılan modelleme mantığı gösterilmektedir.

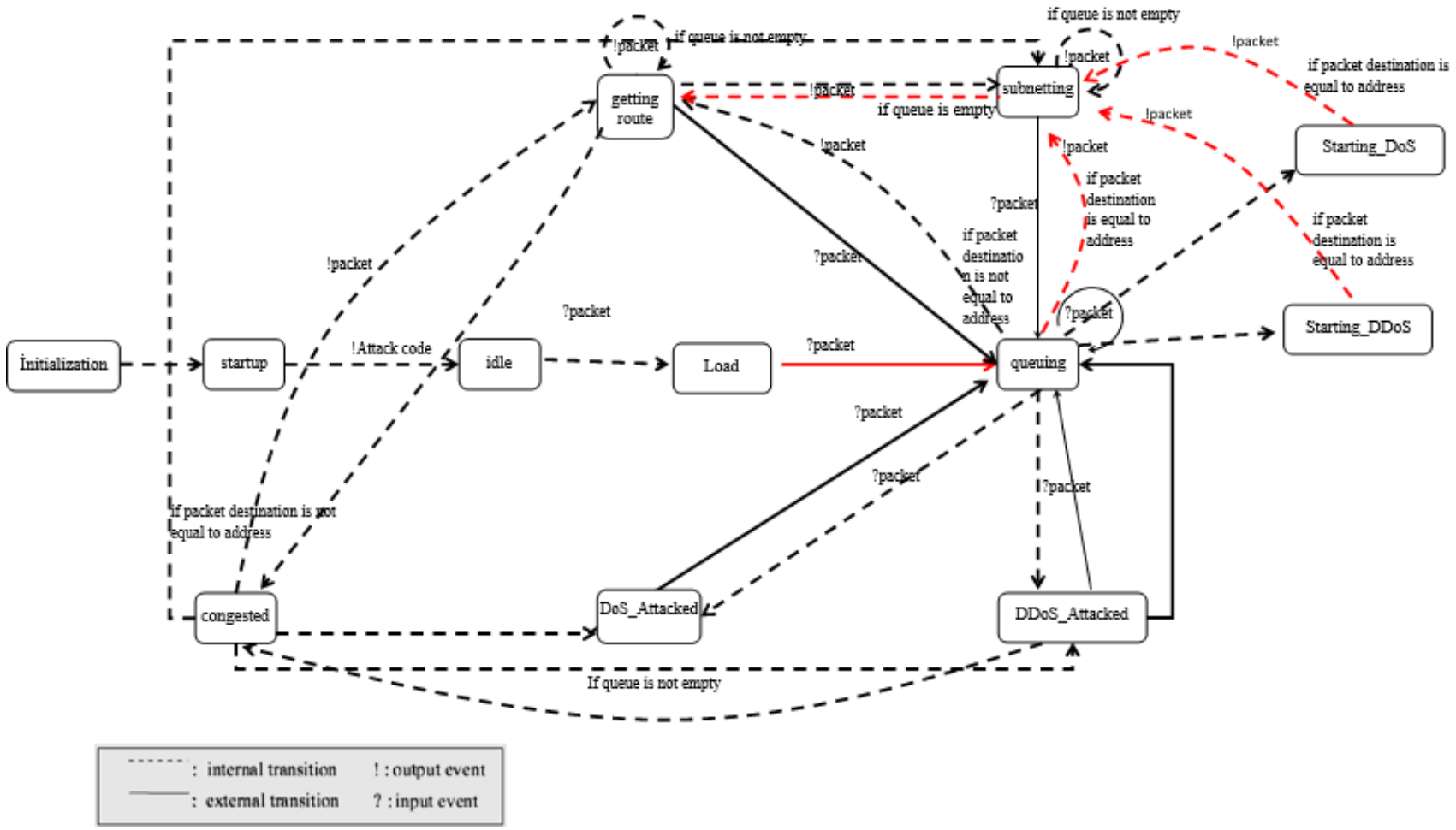


Şekil 5.7. Saldırı senaryosu

Simülasyon, olay sırası dizisinde bulunan her bir trafik nesnesi (olay) sırayla işlenerek yürütülür. Simülasyon zamanı başlangıçta 0'da başlar ve simülasyon saati her zaman olay tablosundaki bir sonraki olayın başlangıç zamanına atlar. Olay işlendiğinde olayın trafik nesnesi uyarılar oluşmasına neden olabilir veya olay tablosuna başka bir olayın eklenmesine neden olabilir. Örneğin, olay tablosundaki bir sonraki olay bir saldırı adımıysa, bu adım uyarılar oluşmasına neden olabilir. Ek olarak, bu saldırı adımının tamamlanması, saldırıdaki bir sonraki adımın olay dizisine eklenmesine izin verir. Olaylar, olay dizisinde hiçbir olay kalmayana kadar işlenir, bu da saldırı senaryosunun simülasyonunun tamamlandığını gösterir.

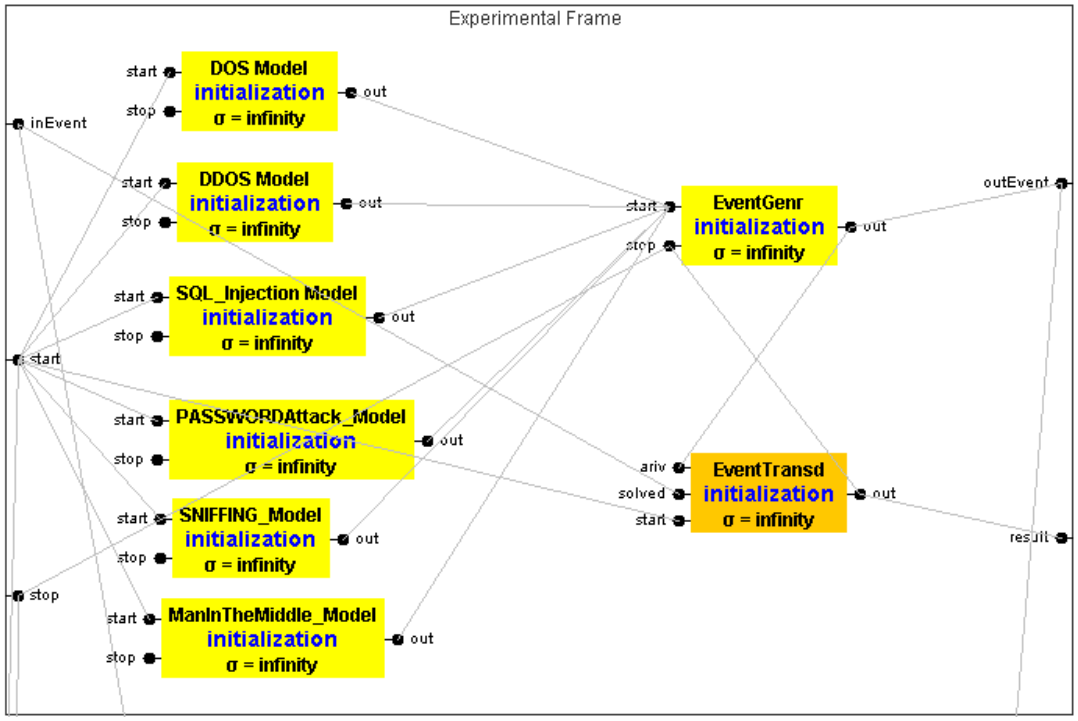
DEVS-Suite siber saldırı uygulaması (DEVS-CAS), DEVS-Suite çekirdeğinin üzerine inşa edilmiştir. Düğümler ve bağlantılar tarafından işlenen olaylar, Şekil 5.8.'de gösterildiği gibi durum çizelgeleri şeklinde tanımlanabilir. Simülasyon modellerinin davranışını görselleştirmek için, iç ve dış olaylara tepki olarak durum değişikliklerini göstermek gerekir. Bir düğüm atom modeli, bir "ilk" aşamada oluşur ve diğer model bileşenleri için herhangi bir bilgiye sahip değildir. Her düğüm komşularına bir merhaba mesajı gönderdikten sonra tablolar oluşturmaya başlar ve ağda neler olduğunu öğrenir. Sistemdeki olaylar, seçilen saldırı tipine göre seçilir. Saldırı mantığını anlamak için yeterli sayıda olay kullanılır. Olay sayısını artırmak performansı düşürmeye neden olsa da bu, doğruluğu arttırmaya yardımcı olan bir durumdur. Yalnızca harici ve dahili geçiş işlevleri, bir düğümün yeni olaylar için durumunu değiştirmesine neden olabilir.

Yazılım ortamında geliştirilen modelin test edilebilmesi için DEVS-Suite içerisinde deneysel çerçeve modeli oluşturulmalıdır. Girdileri enjekte ederek ve çıktıları yorumlayarak senaryoları yönlendirmek için DEVS tabanlı simülasyonlarda deneysel çerçeveler kullanılır. Bu tasarım geleneksel olarak farklı rollere sahip üreteç, alıcı ve dönüştürücü modelleri gerektirir. Model testi gibi belirli kontrollü deneylerde, sıralı programlama, özellikle kod azaltma, test senaryosu geliştirme çıktısı ve başarısız testler için tanılama gibi birçok faydası olan daha basit bir tasarım sunar. Bu araştırma, komut dosyası oluşturma yoluyla test etmeyi kolaylaştıran atomik DEVS'den türetilen bir test çerçevesi sunar [90].



Şekil 5.8. Hizmet reddi saldırısında hedef düğüm durumları, durum geçişleri

Deneysel çerçeve, Şekil 5.9.'da görüldüğü gibi saldırı modellerine sahip iki ana bileşenden oluşur. Olay üreteci (Enent Generator); sisteme tetik sinyali vermek için sistemin giriş terminallerine bağlanan bir üreteçtir. Olay dönüştürücü (Event Transducer); sistem modelinden gelen sonuçları değerlendirmek için modelin çıkış uçlarına bağlanan bir dönüştürücüdür. Olay dönüştürücü, simülasyon çalışmasının sonuçlarının değerlendirilmesinde ve analizinde kullanılan bir araçtır.



Şekil 5.9. Geliştirilen deneysel çerçeve

BÖLÜM 6. SİBER SALDIRILAR VE SALDIRI ÖNLEME YÖNTEMLERİNİN MODELLENMESİ

DEVS-Suite altında yapılandırılan ve topolojik tasarımı yapılan sanal geniş ölçekli ağ sistemine yönelik siber saldırılar gerçekleştirmek üzere yaygın kullanılan siber saldırı türleri kullanılmıştır. İlk olarak web sunucularını hedef alan DoS ve DDoS saldırılarının benzetimi yapılmıştır. Diğer bir saldırı türü olarak parolaya ulaşmak için ardarda veri yüklenmesi veya veri göndermesi yapılarak gerçekleştirilen ve BruteForce olarak adlandırılan şifre tahmin saldırısı (password attack) simülasyonu gerçekleştirilmiştir. Veri tabanı temelli uygulamalara saldırmak için kullanılan bir atak tekniği olarak SQL Enjeksiyon saldırısı tasarlanmıştır. Ağdaki trafik akışındaki paketleri yakalamak için dinleme (Sniffing) ve iki taraf arasındaki iletişimi dinleyip iletişimi manüpüle eden ortadaki adam (Man In The Middle - MITM) saldırılarının benzetimi yapılmıştır. Bu saldırılara ait yapılandırma girişi, simülatör açıldığında yapılandırma formları kullanılarak yüklenmektedir.

6.1. Kullanılan Veri Seti

Bu tez çalışmasında veri değerini oluşturan paketler, Kanada İletişim Güvenliği Kurumu tarafından PCAP ve CSV formatında paylaşılan ve gerçek hayattaki modern ağ trafiğini içeren CSE-CIC-IDS2018 veri seti kullanılarak hazırlanmıştır. CSE-CIC-IDS2018 veri seti, erişime açık en son veri kümelerinden biridir. Gerçekçi arka plan trafiğinin ve çeşitli saldırı senaryolarının test edildiği Amazon Web Hizmetleri (AWS) üzerinde kontrollü bir ağ ortamında 10 gün boyunca tüm ağ trafiğini kaydederek oluşturulmuştur. Sonuç olarak, veri kümesi hem iyi huylu ağ trafiğini hem de popüler ağ saldırılarının örneklerini içerir [91].

Tablo 6.1. Veri setinde etiketlere ait örneklem sayıları

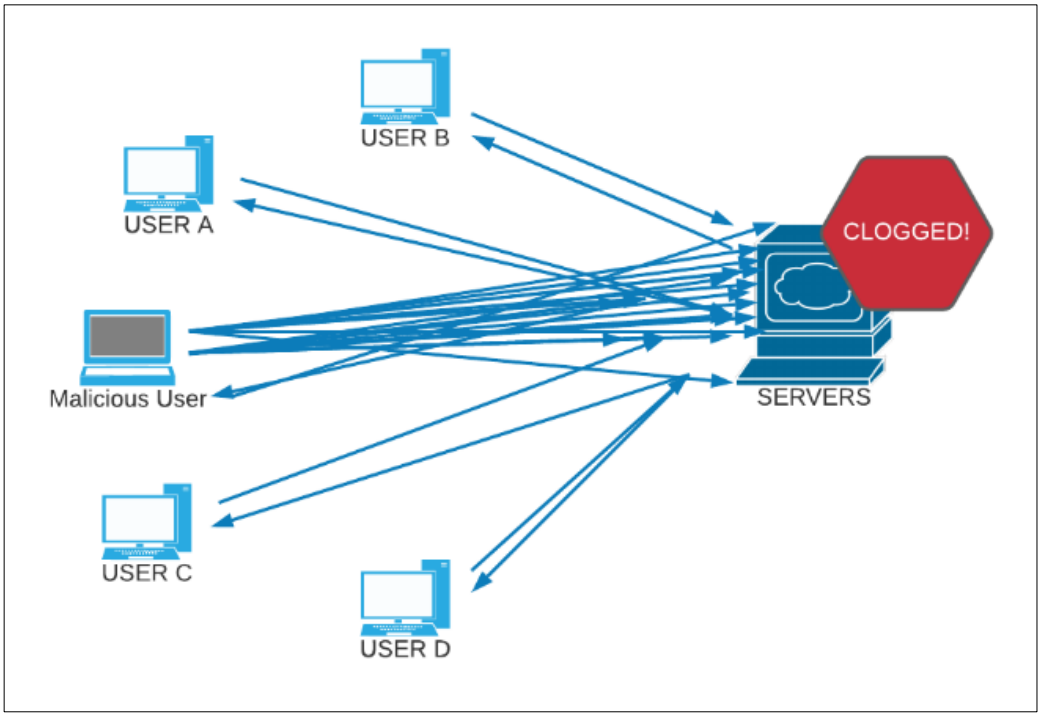
ETİKET	ÖRNEKLEM SAYISI
Benign (zararsız)	6000000
Bot	290000
BruteForce-\Web	612
BruteForce-XSS	231
DDOS	690000
DoS Attacks-Slowloris	11000
DoS Attacks-GoldenEye	41500
DoSattacks-Hulk	462000
DoS Attacks-Slow HTTPTest	140000
FTP-BruteForce	196000
lnfilteration	161000
SQL Injection	90
SSH-Bruteforce	188000

Bu çalışmada bu veri setinin CSV formatı kullanılmıştır. Bu veri setindeki vektörler 79 öznitelik içermektedir, saldırı türüne göre farklı öznitelikler seçilip kullanılabilmektedir. Bu örneklerin özellikleri saldırı türüne bağlı olarak gerekirse ön işlemlerden geçirilip dönüştürülerek daha farklı veriler de elde edilebilmektedir. Tablo 6.1.'de veri seti içerisinde yer alan etiketlere göre örneklem sayıları görülmektedir.

6.2. DoS Saldırısı

DoS saldırısı, bilgisayar korsanının sistemi ve ağı tahrip edebilecek bazı farklı saldırı yöntemleri kullanarak hizmeti ele geçirmesi ve ayrıca CPU, ram, arabellek, ağ bant genişliği gibi bilgisayar kaynaklarını işgal edebilmesi nedeniyle normal kullanıcının hizmeti alamaması olarak tanımlanır. Bir DoS saldırısında, siber saldırganlar, sunucunun bant genişliğini aşırı yüklemek için bu hedef sunucuya hızlı ve sürekli istekler göndermek maksadıyla tipik olarak internet bağlantısı olan bir cihaz kullanır.

DoS saldırganları, sistemdeki bir yazılım güvenlik açığından yararlanır ve sunucunun RAM veya CPU'sunu tüketmeye devam eder. Bir DoS saldırısının temel amacı, Şekil 6.1.'de gösterildiği üzere ağ bağlantılı bir hizmeti aşırı yükleyerek kullanılamaz hale getirmektir. Servis sağlayıcıya gönderilen bu kadar çok sayıda kötü niyetli istek, sunucuyu bir noktadan sonra yanıt veremez duruma getirir [92].

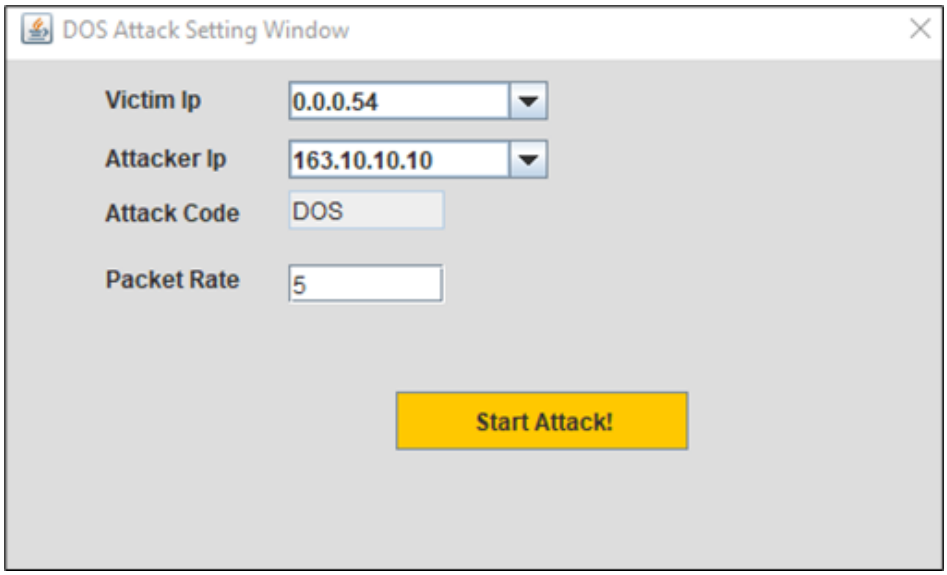


Şekil 6.1. DoS saldırı mekanizması [93]

DoS saldırılarının hedefi genellikle finans kurumları olduğu için hizmet kesintisi, ilgili ağ sistemine zarar vermenin yanı sıra önemli mali kayıplara da neden olabilmektedir [94].

Günümüzde DoS saldırılarını önlemek için pek çok yazılım ve donanım çözümleri geliştirmesine rağmen güvenlik zafiyetleri bulunan cihaz veya web siteleri DoS saldırılarından etkilenmektedirler [95]. Bunun sebeplerinin başında güncellenmemiş ağ cihazları, tecrübesiz güvenlik veya IT çalışanları ve yanlış güvenlik politikaları olarak görülebilir.

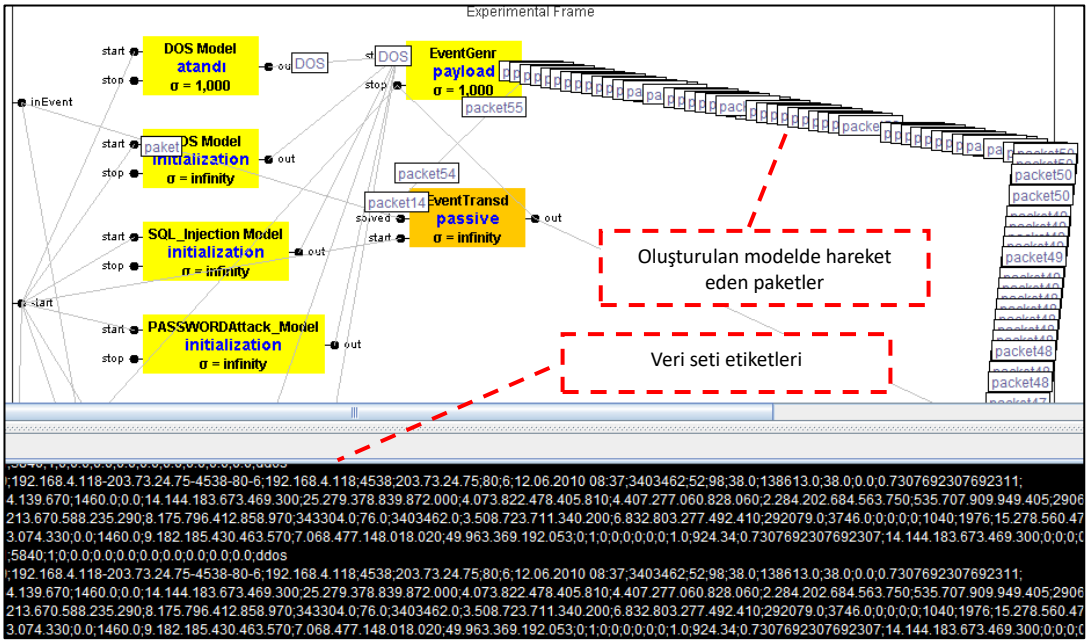
Bu çalışmada ağ bileşenleri tanımlanıp bir topoloji üreteci ile farklı ölçeklerde ağ simülasyonu başlatıldıktan sonra kontrol ara yüzünde çalıştırılıp OSPF protokolü ile yönlendiricilerde yönlendirme tabloları oluşturulmaktadır. Simülasyon ekranında saldırı modeli olarak DoS modeli seçilmesi durumunda Şekil 6.2.'deki DoS saldırısı yapılandırma penceresi açılmaktadır.



Şekil 6.2. DoS saldırısı yapılandırma penceresi

DoS saldırısı ayarlarının yapıldığı formda kurban bilgisayarın ve saldırıyı gerçekleştirecek bilgisayarın IP numaraları ile saldırı kodu ve her adımda gönderilecek paket sayısı bilgisi ayarlanmaktadır. Bu verilerle simüle edilen DoS saldırı modeli tetiklenmiş olur. Deneysel çerçevedeki olay üreteci atomik modelinde olaylar otomatik olarak veya giriş bağlantı noktalarına manuel olarak bir giriş olayı oluşturulabilir. Bir girdi olayı, bir bağlantı noktası adını(port adı), veri değerini (paket) ve geçen süreyi içerir. Geçen zaman, ilişkili olayın bir zaman damgasıdır ve belirli bir olayı belirli bir sonlu, gelecek zaman örneğinde planlamak ve enjekte etmek için kullanılır. Geçen zaman, simülatör saati ile ilişkili zaman birimleri cinsinden sağlanır.

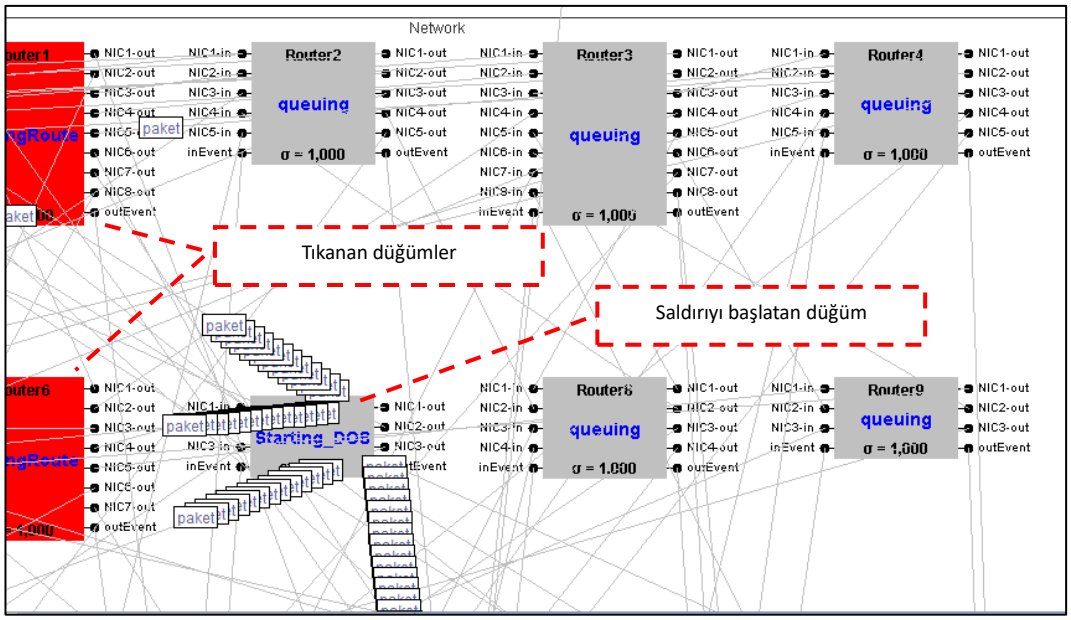
DoS simülasyonunda DoS saldırı modelinin çıkışları Şekil 6.3.'deki gibi olay üreteci (event generator) atomik modelinin giriş bağlantı noktasına bağlanarak giriş olayı otomatik oluşturulmuştur. Bundan sonra durumların ve çıktıların değişimini görmek ve değerlendirmek için kontrol bölümündeki "step" veya "run" düğmesine basılarak saldırı sürdürülür. Her adımda saldırgan olan cihazın çıkış portlarından hedefi kurban bilgisayar olan ve sayısı saldırı başlangıcında ayarlanan paketler gönderilir.



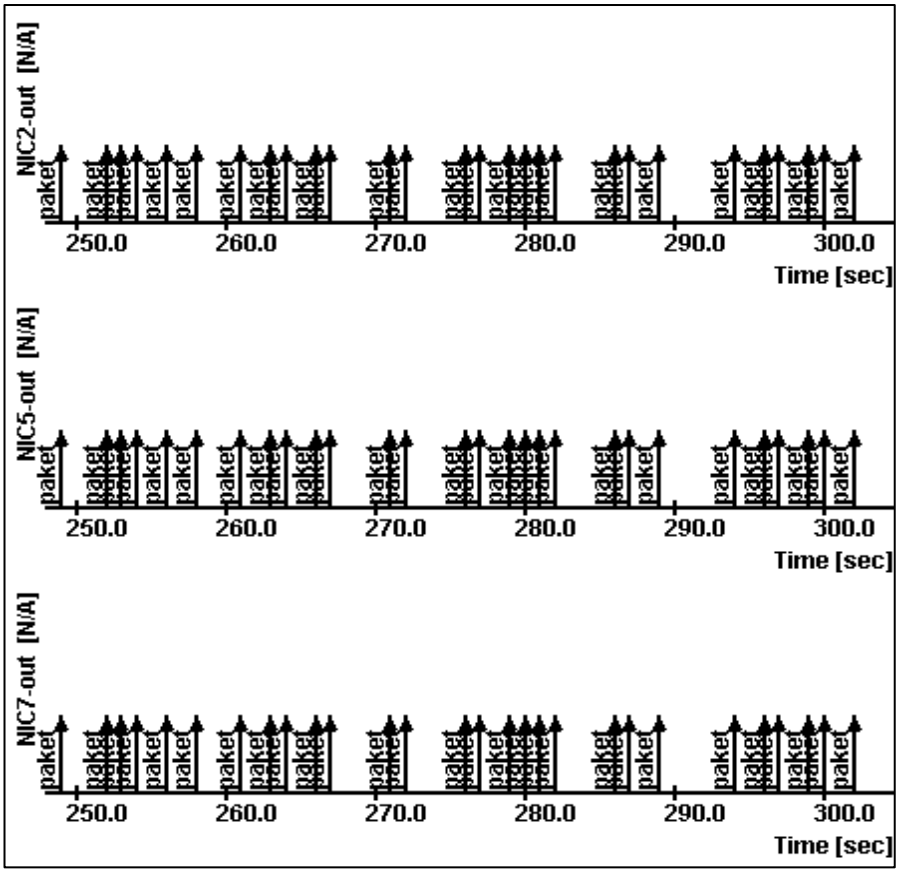
Şekil 6.3. Saldırı modeli ve deneysel çerçeve bağlantı görünümü

Saldırı paketleri oluşturulurken paketlere eklenen saldırı kodları aracılığıyla saldırgan bilgisayarın durumu Şekil 6.4.'de görüldüğü gibi Starting_DOS olarak değişmekte böylece saldırgan cihaz simülasyon ortamında gözlemlenebilmektedir.

DoS saldırısını başlatan düğümde oluşturulan ve çıkış portlarından gönderilen paketler Şekil 6.4.'te yoğun bir paket çıkışı şeklinde görülebilmektedir. Aynı zamanda Şekil 6.5.'te kayıt izleme penceresinden de saldırıyı başlatan cihazın çıkış portlarında görülen paket yoğunluğu zamana bağlı bir grafik ile izlenebilmektedir.

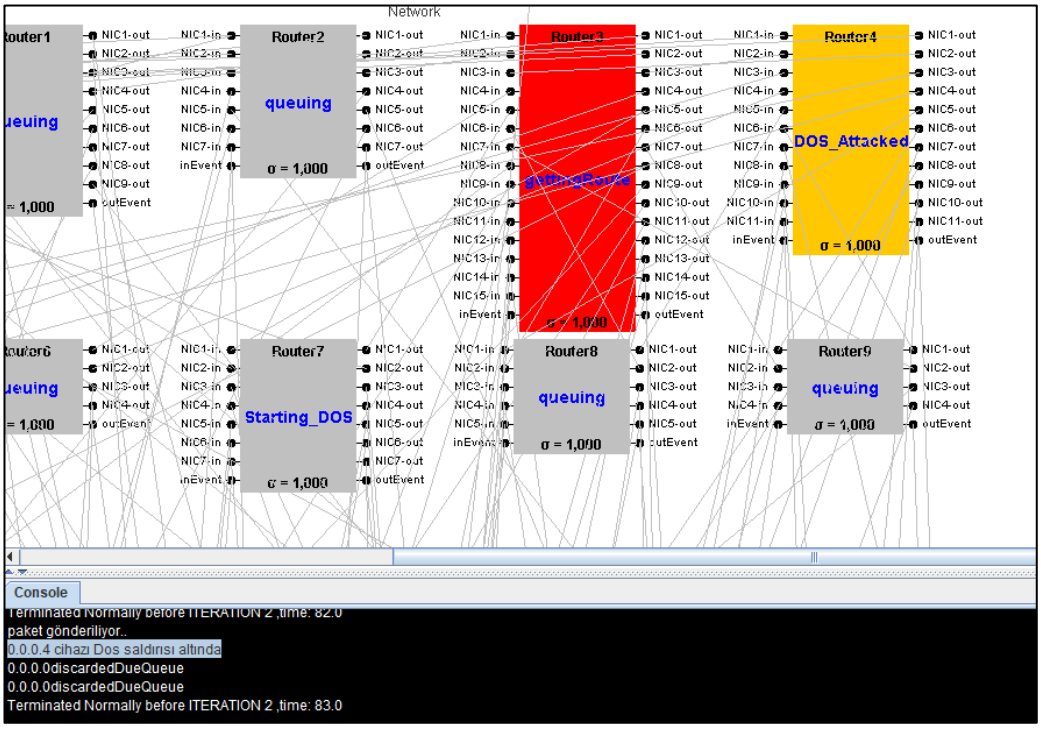


Şekil 6.4. Saldırgan bilgisayardan hedefe yönelik saldırı görüntüsü



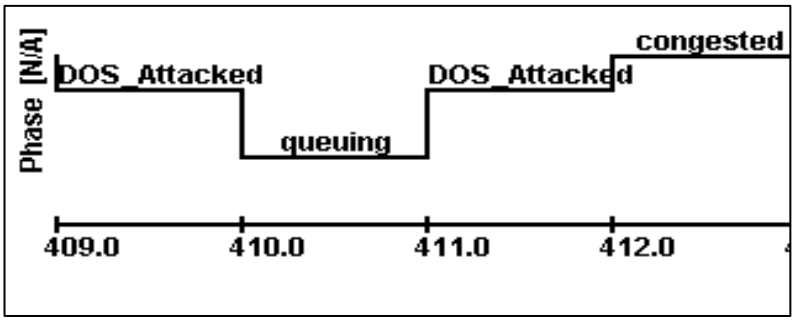
Şekil 6.5. Çıkış portları izleme penceresi

Saldırgan tarafından kurban cihaza yönelik yoğun bir şekilde başlatılan paket trafiği ağda yönlendirme tabloları ve yönlendirme algoritmalarına göre farklı yollarla hedefe ulaşmaktadır. Simülasyon anında paket trafiğindeki yoğunluktan dolayı yol üstündeki bazı yönlendirici düğümlerde tıkanıklıklar yaşanabilmektedir. Kurban olarak seçilen hedef bilgisayarın giriş portlarına gelen paketler belirli bir sayıya ulaştıktan sonra bilgisayar Şekil 6.6.'de görüldüğü gibi "DOS_Attacked" durumuna geçmekte ve ilgili cihaza yönelik DoS saldırısının yapıldığı konsol penceresinden kurbanın IP numarası belirtilerek durumu bir uyarı mesajı ile belirtilmektedir.



Şekil 6.6. DoS saldırısı altındaki düğümün görünümü

İlk DoS saldırı uyarısı yapıldığı zaman cihaz tamamen servis dışı kaldığı anlamına gelmez, saldırı devam ettiği süre içerisinde hedefe ulaşan paket sayısına bağlı olarak bu uyarı belirli aralıklarla tekrar etmektedir. Bir süre sonra saldırı kesilmezse tamamen tıkanma gerçekleşecek ve Şekil 6.7.'de grafikten anlaşıldığı üzere kırmızı seviye olan "congested" durumuna geçilecektir ve servis engellenmiş olacaktır.



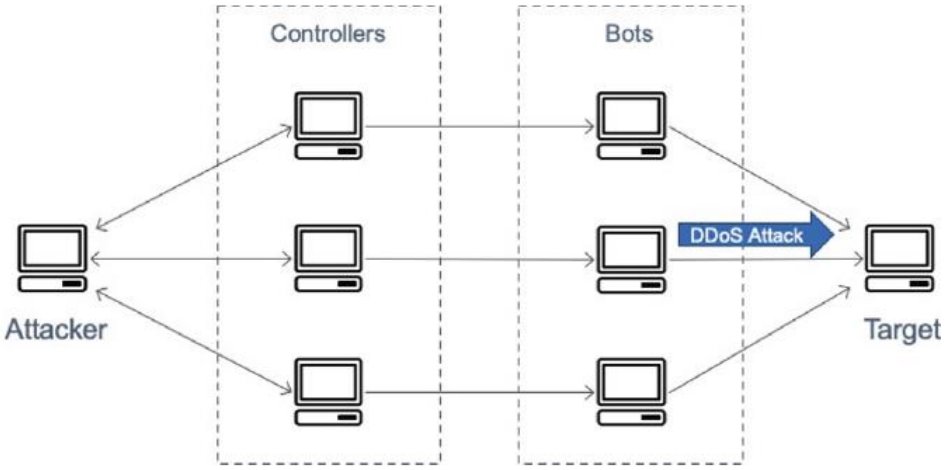
Şekil 6.7. DoS saldırısında servis dışı kalma durumu

Saldırı devam ettirildiği takdirde bir süre sonra kurban bilgisayar tamamen tıkanıp trafik akışı tamamen durmaktadır ve saldırı amacına ulaşmaktadır. Saldırı başlangıcında saldırı yapılandırma ayarları yapılırken her adımda gönderilecek paket sayısını arttırılırsa saldırı ve servis dışı kalma süresi kısalacaktır. DoS saldırısı altındaki hedef düğümün durum değişim grafiği ile portlarında oluşan paket trafiği Şekil 6.5. ve Şekil 6.7.'de gösterilmektedir. DoS saldırısına ait simülasyon deney sonuçları ve grafikleri Bölüm 7'de gösterilecektir.

6.3. DDoS Saldırısı

DDoS saldırısı, en az bir hedefe karşı bir DoS saldırısı başlatmak amacıyla birçok bilgisayarın kullanıldığı koordineli bir DoS saldırısıdır. Saldırgan, birden fazla bilgisayarın farkında olmadan kaynaklarını kullanarak saldırı başlatır. İstemci/sunucu teknolojisi kullanılırsa saldırgan sayısı arttırılabilir. Bir DDoS saldırısı, Şekil 6.8.' de gösterildiği gibi dört kısımdan oluşur, bunlar:

- Saldırıyı başlatan gerçek saldırgan.
- Zombi bilgisayarları kontrol edebilen, güvenliği istismar edilmiş ana bilgisayarlar.
- Zombi bilgisayarlar (Botnet).
- Hedef bilgisayar.



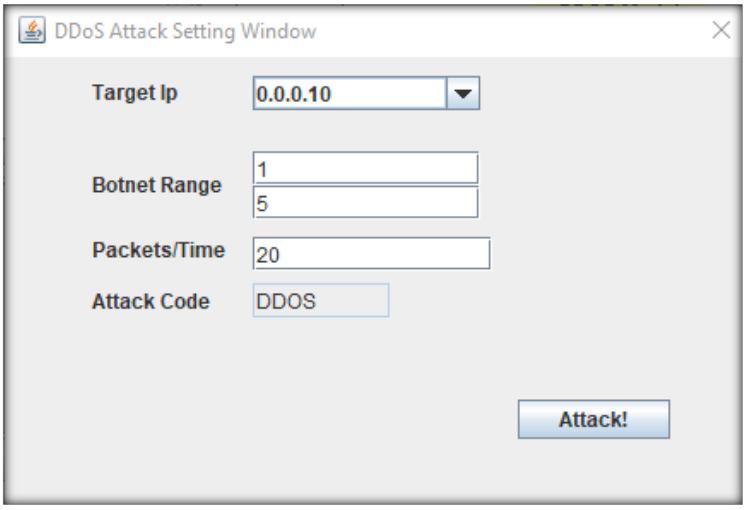
Şekil 6.8. DDoS saldırısı mekanizması [96]

Bir DDoS saldırısı dört aşamada gerçekleşir. İşe alma aşamasında saldırgan, saldırıyı gerçekleştirmek için savunmasız aracıları seçer. Uzlaşma aşamasında saldırgan, aracıların güvenlik zafiyetlerini istismar ederek saldırı kodunu yerleştirir, ayrıca keşif ve devre dışı bırakılmalarını engeller. İletişim aşamasında botnetler hazır olduklarını saldırgana işleyiciler aracılığıyla duyurur. Son olarak saldırı aşamasında saldırgan saldırının başlama emrini verir.

Dağıtılmış Hizmet Reddi (DDoS) saldırısında, bir saldırgan, bir hedefe yönelik saldırıyı düzenlemek için birden çok kaynak kullanır. Bu kaynaklar, kötü amaçlı yazılım bulaşmış bilgisayarların, yönlendiricilerin, IoT cihazlarının ve diğer uç noktaların dağıtılmış gruplarını içerebilir. Şekil 6.8.'de, saldırıya katılan ve hedefi hizmet dışı bırakmak için bir paket akışı veya istek üreten güvenliği ihlal edilmiş bir ana bilgisayar ağını göstermektedir

Saldırgan uzaktan her bota güncellenmiş talimatlar göndererek makineleri yönetebilir. Bir kurbanın IP adresi hedeflendiğinde, her bot, hedefe istekler gönderir ve hedeflenen sunucunun veya ağın kapasitesinin dolmasına ve normal trafiğin hizmet reddi ile sonuçlanmasına neden olur. Her bot aynı zamanda normal bir internet cihazı olduğundan, normal trafik ile saldırı trafiğini ayırmak zordur.

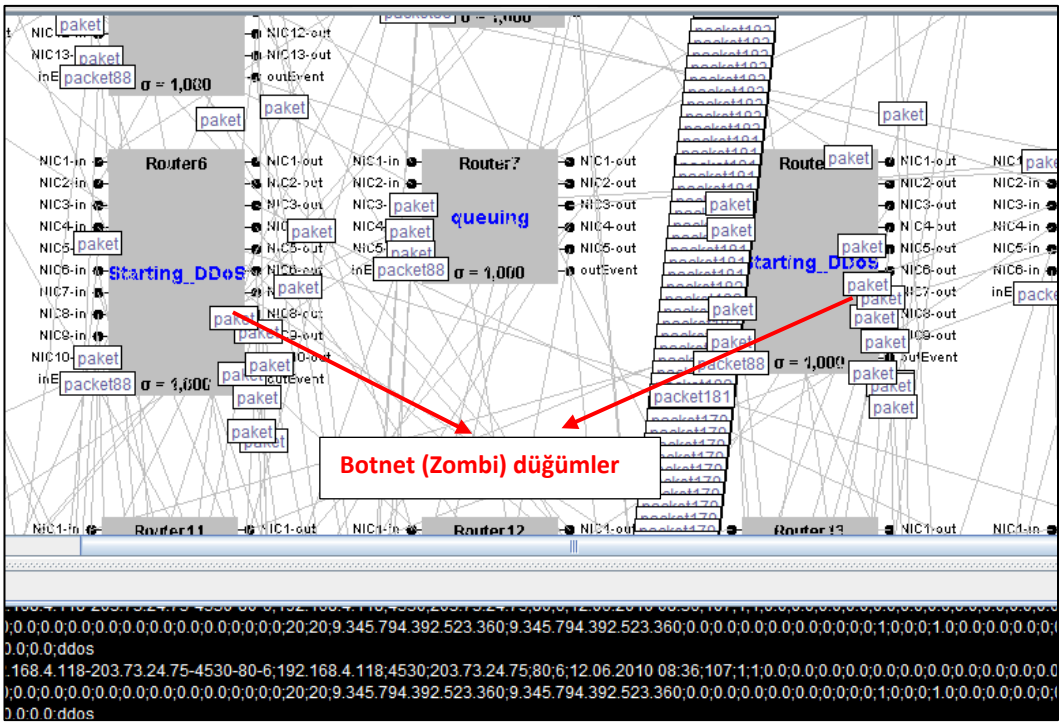
Simülasyon ekranında DDoS model seçilmesi durumunda Şekil 6.9.'daki DDoS saldırısı yapılandırma penceresi açılmaktadır.



Şekil 6.9. DDoS saldırısı yapılandırma penceresi

DDoS saldırı ayarlarının yapıldığı formda kurban bilgisayarın IP numarası belirtilmektedir. Saldırıyı gerçekleştirecek olan botnet veya zombi bilgisayarların sayısı Botnet Range ile belirtilmektedir. Zombi cihazların IP numaraları botnet sayı aralığı ile ilişkili IP numaraları olarak ayarlanmaktadır. Saldırı kodu ve her adımda gönderilecek paket sayısı bilgisi ayarlanmaktadır. Bu verilerle simüle edilen DDoS saldırı modeli yapılandırılmış olur. Bu veriler deneysel çerçevedeki üreteç atomik modelinin giriş portlarına otomatik olarak bir giriş olayı oluşturmaktadır. Bir girdi olayı, bir bağlantı noktası adını (port adı), veri değerini (paket) ve geçen süreyi içerir.

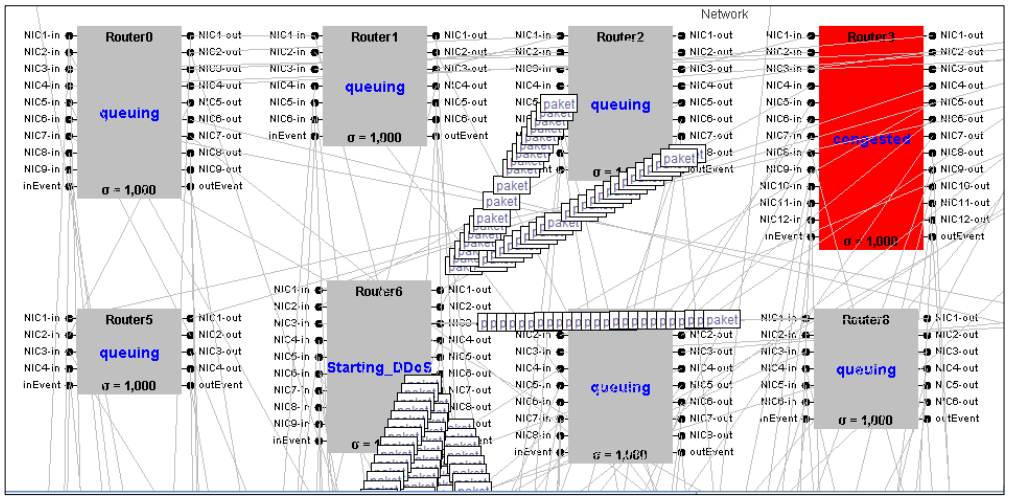
Simülasyondaki saldırgan ve kurban düğümlerin durumlarını ve çıktıların değişimini görmek ve değerlendirmek için kontrol bölümündeki panel yardımıyla saldırı sürdürülür. Her adımda saldırgan olan cihazın çıkış portlarından, hedefi kurban bilgisayar olan ve sayısı saldırı başlangıcında ayarlanan paketler gönderilir. Saldırı paketleri oluşturulurken paketlere eklenen saldırı kodları aracılığıyla saldırgan bilgisayarların durumu Şekil 6.10.'da görüldüğü gibi "Starting_DDoS" olarak değişmekte ve gözlemlenebilmektedir.



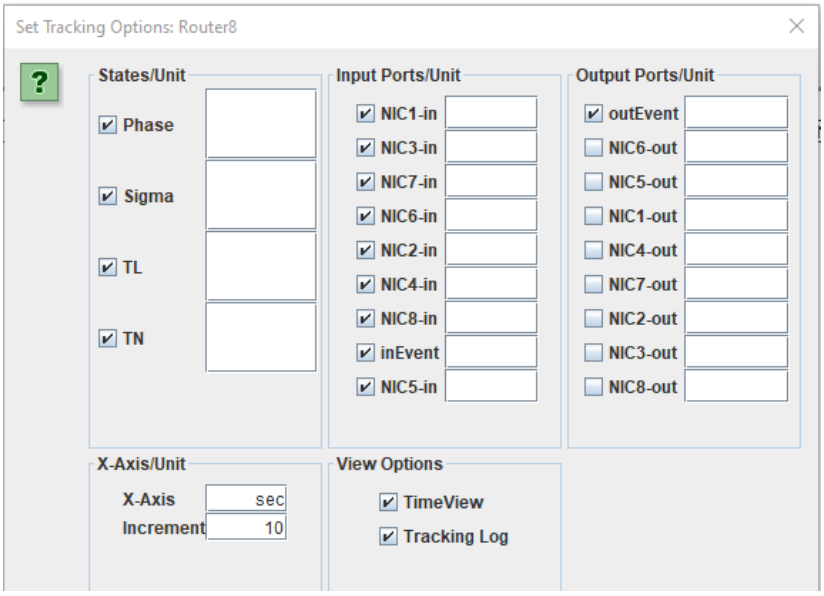
Şekil 6.10. DDoS saldırı aşamasındaki Botnetler

Saldırı emri alan botnetler Şekil 6.11.'de görüldüğü gibi hedefe yönelik yoğun bir paket trafiği başlatmaktadır. Her adımda üretilecek/gönderilecek paket sayısı saldırı başlamadan önce parametre ayarlarının yapıldığı formda belirtilmektedir. Hedefe yönelik başlatılan yoğun trafik birden çok botnet(zombi) kaynaktan yapıldığı için hedefe giden yol üstündeki düğümlerde geçici tıkanıklıklara neden olmaktadır. Hedef düğüm yoğun paket akışına hedef oluğunu, düğüme gelen paket sayısı belli bir sayıyı geçtiğinde bunu "DoS_Attacked" durumuna geçerek göstermektedir. Hedef makinanın tampon alanı kısa sürede dolacaktır. Saldırı devam ettiği müddetçe tıkanıklığı sürecektir ve bu cihaz servis dışı kalacak, böylece DDoS saldırısı amacına ulaşmış olacaktır.

DDoS saldırısına ait durum geçişlerini ve log kayıtları izleyebilmek için istenilen düğümün Şekil 6.12.'deki izleme arayüzünden gerekli seçenekler ayarlanabilmektedir. Bu pencereden giriş portları, çıkış portları, olay zamanları, zaman eksen ayarları ve görünüm seçenekleri seçilebilmektedir.



Şekil 6.11. DDoS saldrı aşamasındaki paket trafiği görünümü

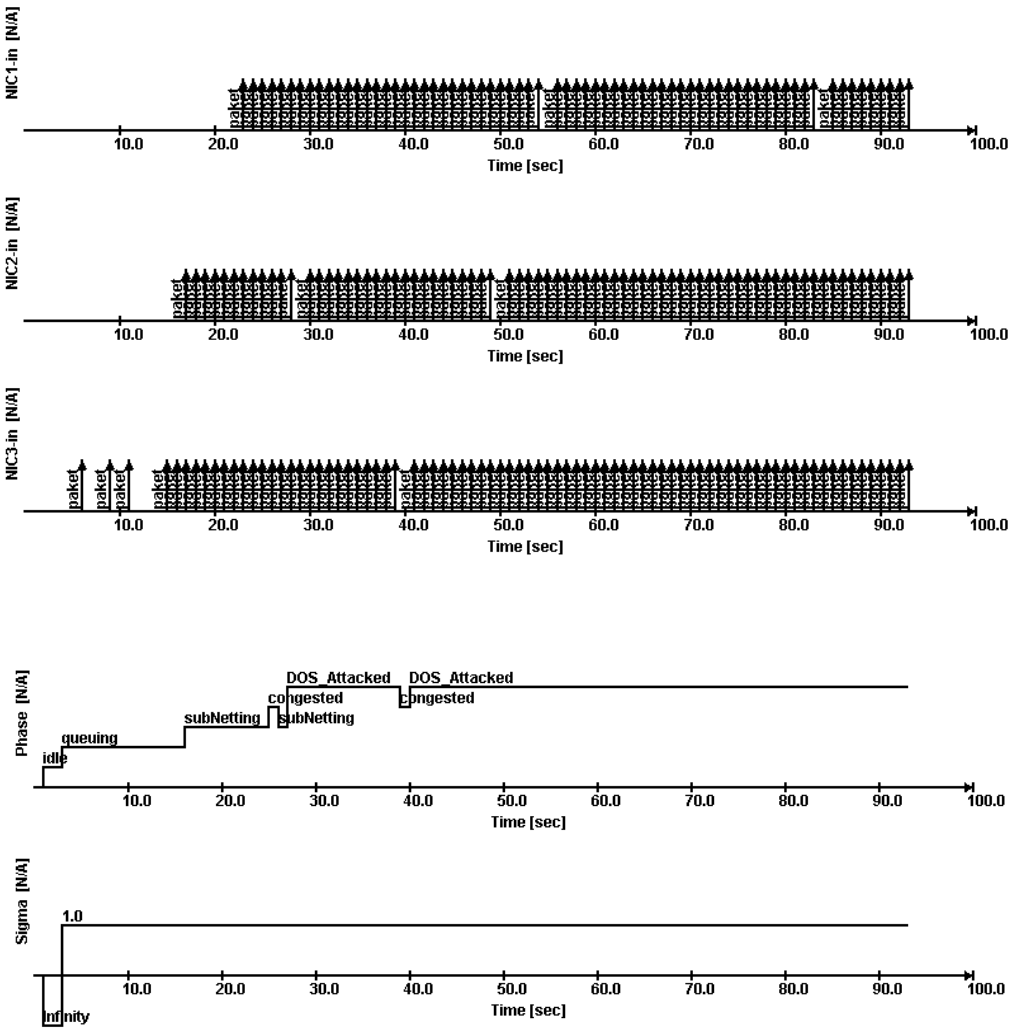


Şekil 6.12. Ağ izleme seçenekleri penceresi

Ağ trafiği normal akışını sürdürürken saldırı başladıktan yaklaşık 20 saniye sonra hedef bilgisayarın giriş portlarında sıra dışı bir paket yoğunluğu oluştuğu görülmektedir. Hedef düğüm, giriş portlarındaki yoğun paket akışına hedef oluğunu, düğüme gelen paket sayısı belli bir sayıyı geçtiğinde "DoS_Attacked" durumuna geçtiğini Şekil 6.13.'te görüldüğü gibi göstermektedir. Hedef makinanın tampon alanı kısa sürede dolmakta ve tıkanma durumuna(congested) geçtiği görülmektedir.

Saldırı devam ettiği müddetçe tıkanıklığı sürmektedir ve bu cihaz servis dışı kalmaktadır, bu DDoS saldırısı amacına ulaşmıştır.

Geliştirilen uygulamaların test edilmesi için Kanada Siber Güvenlik Enstitüsü tarafından paylaşılan CSE-CIC-IDS2018 veri seti kullanılmıştır.



Şekil 6.13. DDoS saldırısına ait hedef düğümün giriş portlarının izleme penceresi ve durum değişim grafiği

6.4. BruteForce (Kaba Kuvvet/Şifre) Saldırısı

Kaba kuvvet saldırısında saldırganlar, hedefledikleri kullanıcı hesaplarına sızmak ve hesapla bağlantılı bilgileri ele geçirmek için genellikle otomatik (amaca yönelik siber korsanlık yazılımları üzerinden) deneme-yanılma yöntemi uygulamaktadır. Bu

yöntem, çeşitli güvenlik unsurları içeren karmaşık kombinasyona sahip (ya da kriptografik) şifrelerin kırılmasını mümkün kılmaktadır. Aynı kapsamdaki farklı bir yöntem de manuel olarak rastgele karakterler kullanılarak parola tahmini yapılmaya çalışılmasıdır. 12345678 ya da Qwerty gibi çok basit ve güvensiz şifrelerin denenmesinin haricinde sosyal mühendislik teknikleri kullanılarak kullanıcıların sosyal medya hesapları, vb. incelenerek elde edilen bilgilerden yola çıkılarak yine deneme-yanılma şeklinde hesaba girilmeye çalışılabilmektedir. Ancak genellikle otomatik deneme-yanılma metodunun uygulandığının altını çizmek mümkündür.

Otomatik deneme süreci üzerinde durulması gereken bir konudur. Türkçe'de ö, ü gibi özel karakterlerin çıkarıldığı a-z aralığındaki harfler ile 0-9 arası rakamlardan oluşmuş 5 haneli bir şifre için 52 milyon 521 bin 875 adet kombinasyon bulunmaktadır. Bu kombinasyonlara *=!'^ gibi karakterlerin bulunma ihtimali de eklendiğinde ihtimaller seti çok daha yüksek sayılara ulaşmaktadır. Şifrelerin açığa çıkarılması için bu alana özel yazılımların tercih edilmesinin arkasında yatan neden budur.

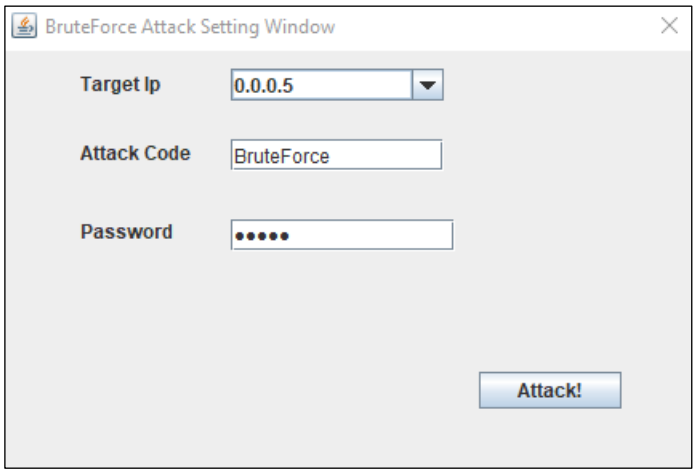
Çoğu web sitesine girişlerde büyük-küçük harf, rakam, alfanümerik kombinasyonlar gibi öğelere sahip karmaşık şifrelerle girişler söz konusu olduğunda siber korsan açısından süreç çok daha uzun olabilmektedir. Ancak en nihayetinde gelişmiş ya da farklı yazılımsal araçlar, sosyal medya ve benzer kanallar üzerinden toplanan kullanıcı bilgileri ile kullanıldığında, bir de siber korsanın kaba kuvvet alanına hakimiyeti eklendiğinde ne kadar zor olursa olsun şifrelerin ele geçirilmesi, dolayısıyla kritik verilere izinsiz kimseler tarafından erişimi mümkün hale gelebilmektedir.

Bu çalışmada kaba kuvvet saldırısını simülasyonunda şifre ihtimallerini denemek için BruteForce.java sınıfında Şekil 6.14.'deki karakter seti kullanılmıştır. Uzun ve karmaşık şifrelerin bulunması uzun sürmektedir. Özel karakterlerin bulunmadığı bu karakter setinde bile 5 haneli bir şifre için 50 milyondan fazla kombinasyon denemek gerekmektedir.

```
{'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j',
 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't',
 'u', 'v', 'w', 'x', 'y', 'z', 'A', 'B', 'C', 'D',
 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N',
 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X',
 'Y', 'Z', '1', '2', '3', '4', '5', '6', '7', '8',
 '9', '0'};
```

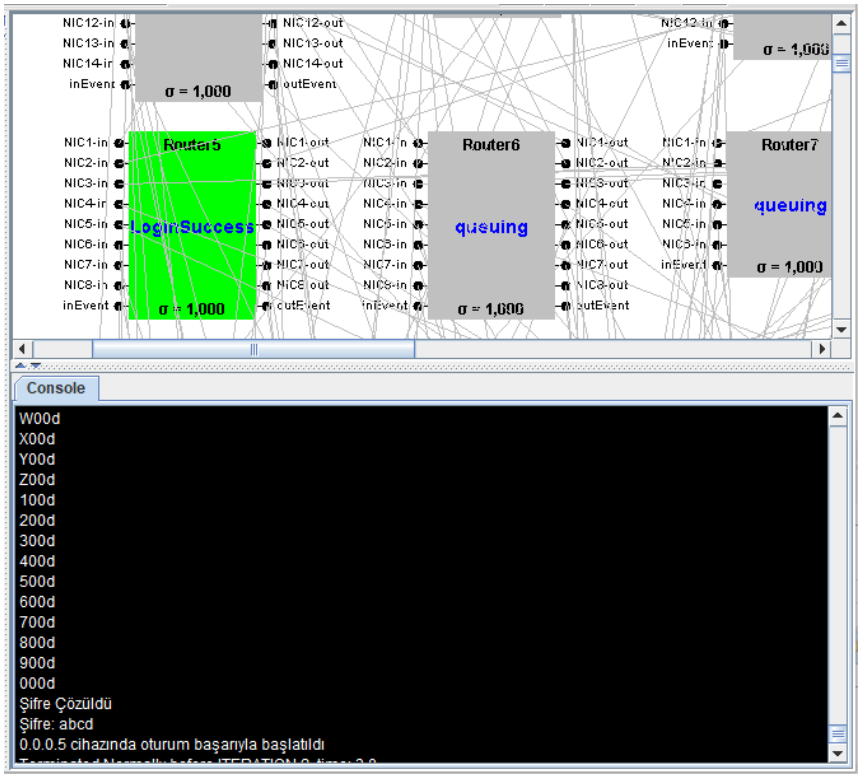
Şekil 6.14. BruteForce için kullanılan karakter seti

BruteForce yapılandırma penceresinde hedef bilgisayarın IP numarası ve saldırı kodu ile birlikte hedef bilgisayarın oturum açma şifresi Şekil 6.15.'deki pencereden girilebilmektedir. Şifreyi saldırı simülasyonunda yapılandırma penceresinden belirlemek sadece sistemin doğru sonuç ürettiğini doğrulamak maksadıyladır.



Şekil 6.15. BruteForce saldırısı yapılandırma penceresi

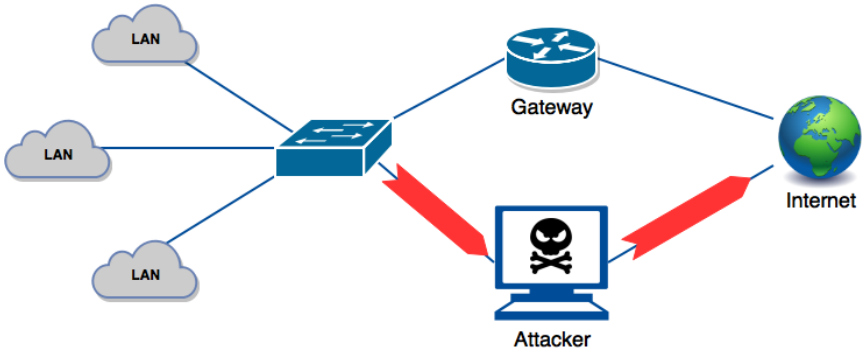
Hedef düğümde BruteForce nesnesi oluşturulup şifre çözüldükten sonra cihazın durum bilgisi Şekil 6.16.'daki gibi "LoginSuccess" olarak değişecektir. Bu da saldırının başarıya ulaştığını gösterecektir. Yapılandırma girişinde seçilecek uzun ve karmaşık bir şifre bu süreyi oldukça uzatacaktır. Sonucu hızlı test edebilmek açısından kısa şifreler kullanmak bekleme süresini azaltacaktır.



Şekil 6.16. BruteForce saldırısı başarılı olma durumu

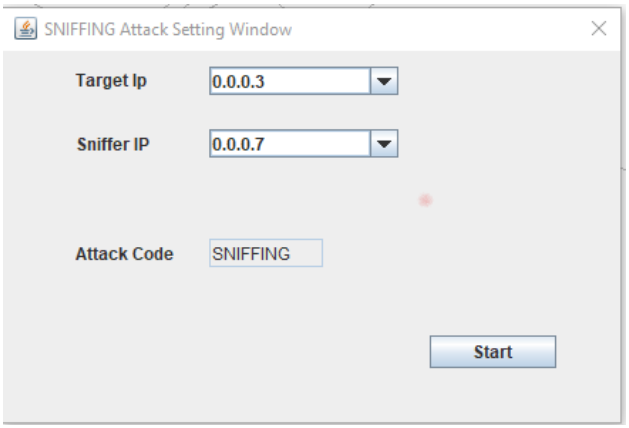
6.5. Dinleme Saldırısı

Bilgi güvenliği perspektifinden bakıldığında, paket dinleme (sniffing), trafiği yakalanabileceği, analiz edebileceği ve izlenebileceği bir hedefe yönlendirmek anlamına gelir. Ağ trafiğinin dinlenmesinde temel mantık, Şekil 6.17.'de görüldüğü üzere ağ geçidi cihazına gelen her paket kabul edildiği için iki bilgisayar arasındaki tüm verilerin yakalanarak saklanması olarak tanımlanabilir. Düz metin olarak bilgi içeren herhangi bir ağ paketi, saldırganlar tarafından ele geçirilebilir ve okunabilir. Bu bilgiler, kullanıcı adları, şifreler, gizli kodlar, bankacılık detayları veya saldırgan için değerli olan herhangi bir bilgi olabilir. Bilgisayarlar arasındaki bağlantıların şifreli olması bu saldırıya karşı alınabilecek en önemli önlemdir. Şifreli paketler yakalanabilse bile içeriği anlaşılmayacaktır. Şifreleme algoritmasının da saldırılara karşı dayanıklı ve uygun performans sağlayan yapıda olmalıdır.



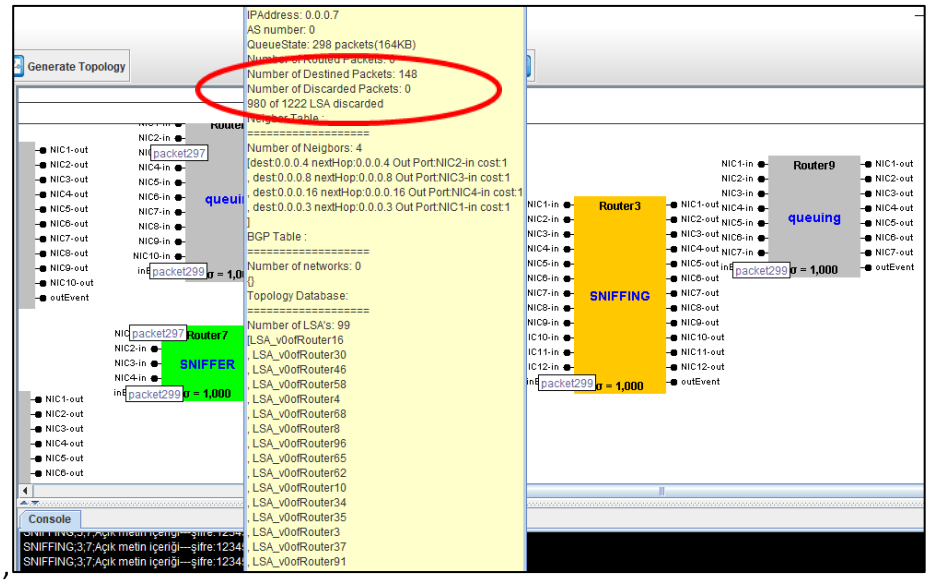
Şekil 6.17. Ağ trafiğinin dinlenmesi

Genel olarak pasif ve aktif olarak nitelenen iki dinleme türü vardır. Dinleme yapan cihazın ağ kartının karışık (promiscius) modda çalışması sağlanmalıdır. Böylece portlarına gelen her paketi kabul edebilecektir.



Şekil 6.18. Dinleme yapılandırma penceresi

Ağ trafiğini dinlemek yani ağda taşınan paketleri yakalamak için Şekil 6.18.'deki konfigürasyon arayüzünde dinleme yapılacak hedef cihaz ve saldırgan düğümün IP numaraları belirlenmektedir. Bütün ağ trafiğinin dinlenmesine karşın sadece bir cihazın hedef gösterilmesinin nedeni ağ üzerinde başkaca trafik oluşmasa bile simülasyon çalışma zamanında bu hedef düğüme yönelik ağ trafiği oluşturularak gelen bütün paketlerin dinleyici cihaza yönlendirilmesidir. Saldırgan düğüm tüm paketleri yani farklı IP adreslerine sahip paketleri de kabul edecek şekilde yapılandırılabilmektedir. Şekil 6.19.'daki örnekte test amacıyla sadece belirli bir hedef düğüm dinlenmiştir.



Şekil 6.19. Dinleyici cihazın durumu ve paket istatistiği

Simülatörde dinleme saldırı modelini çalıştırdığımızda bir süre sonra dinleme yapılanan düğümün durumu "SNIFFING" olarak değişmektedir, saldırgan konumundaki paket dinleyici düğümün durumu "SNIFFER" olarak izlenebilmektedir ve fare işaretçisini düğümün üzerine getirdiğimizde anlık istatistik listesi görülmektedir. Bu listede bu düğüme o ana kadar gelen paket sayısı görülmektedir. Test amacıyla hedef düğüme gönderilen paketlerin içerisinde şifrelenmemiş açık metin içeriğine örnek olacak bir metin yerleştirilmiştir. Dinlenen düğümden elde edilen paketlerin içeriği okunarak konsol penceresinde yazdırılmıştır ve konsolda pakette bulunan saklı metin görülebilmektedir. Paket dinleyiciler ancak açık metinleri okuyabilirler, şifrelenmiş mesajlar yakalansa bile içeriği okunamaz kabul edilmektedir.

6.6. Ortadaki Adam Saldırısı

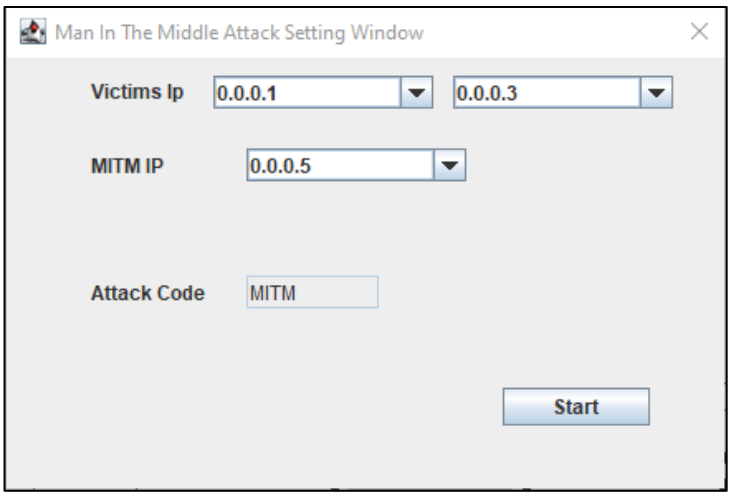
Ortadaki adam (Man In The Middle - MITM) saldırısında, saldırgan iki hedef arasındaki iletişimi gizlice dinler ve ardından birbirleriyle doğrudan iletişim kurduklarına inanan iki taraf arasındaki mesajları gizlice aktarır veya değiştirir. Ortadaki adam saldırılarının bir örneği, saldırganın kurbanlarla bağımsız ilişkiler kurduğu ve kurbanların birbirleriyle özel bir ilişkisi üzerinden doğrudan

konuştuklarına güvenmelerini sağlamak için aralarındaki mesajları aktardığı dinamik bir gizli dinlemedir. Tüm iletişim saldırgan tarafından kontrol edilir. Saldırgan, iki taraf arasında geçen her önemli mesajı engelleme ve yenilerini enjekte etme kapasitesine sahip olmalıdır. Bu, birçok koşulda doğrudandır; örneğin, şifrelenmemiş bir kablosuz erişim noktasının kapsamı içindeki bir saldırgan, kendisini ortadaki adam olarak ekleyebilir.

Araya girme yöntemlerinin başında ARP yanıltma (Spoofing), IP yanıltma ve DNS yanıltma sayılabilir. Bu yöntemlerden ARP yanıltma; saldırgan, MAC adresini yasal sunucu IP'sine bağlamak ve MAC adresini istemci IP'sine bağlamak için ARP mesajlarını taklit eder. IP yanıltma; saldırgan çekirdek ağdaki trafiği keser ve paketlerin kaynak veya hedef IP adresini saldırganınkiyle değiştirir. DNS yanıltma; saldırgan, belirli bir hizmeti sağlayan yasal sunucuların IP adresini saldırganın IP'si ile değiştirerek bir DNS sunucusunu zehirler.

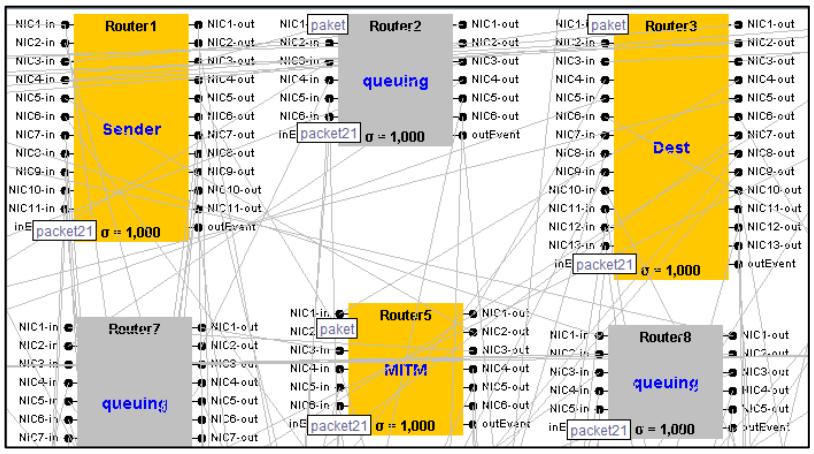
Ortadaki adam saldırısı yaygındır ve yeterli bir kimlik doğrulama güvenliğine sahip olmayan kriptografik sistemlerin çoğu, ortadaki adam saldırısına uğrama tehdidi altındadır.

Bu çalışmada ortadaki adam saldırı simülasyonunu gerçekleştirmek için Şekil 6.20.'de görülen saldırı yapılandırma arayüzü açılarak işleme başlatılır.



Şekil 6.20. Ortadaki adam saldırısı yapılandırma penceresi

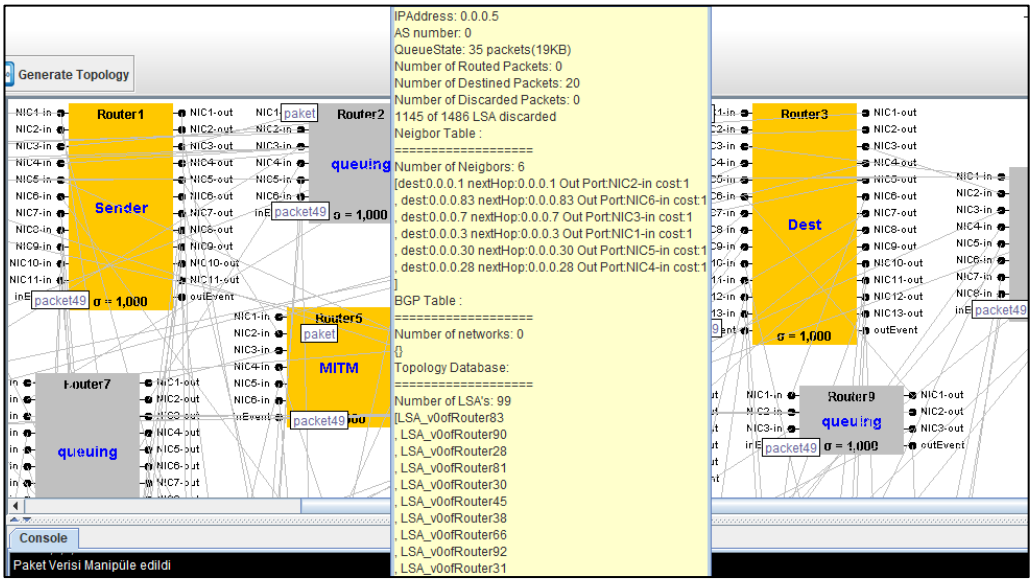
Araya girilerek dinlenecek kurban seçilen cihazlar ile saldırgan düğümün IP numarası bu arayüzde tanımlanmaktadır. Saldırı kodları her saldırı için saldırı imzası niteliğindedir, atomik düğümlerde saldırılara ait durum geçişleri bu saldırı kodları kullanılarak yapılmaktadır. Simülasyon başladığında iletişimde olan kurban düğümlerden olan mesaj gönderen kaynak düğümün durumu "Sender", mesajı alan hedefin durumu "Dest." ve araya girip paketleri üzerinden geçiren saldırgan düğümün durumu "MITM" olarak Şekil 6.21.'de gösterilmektedir.



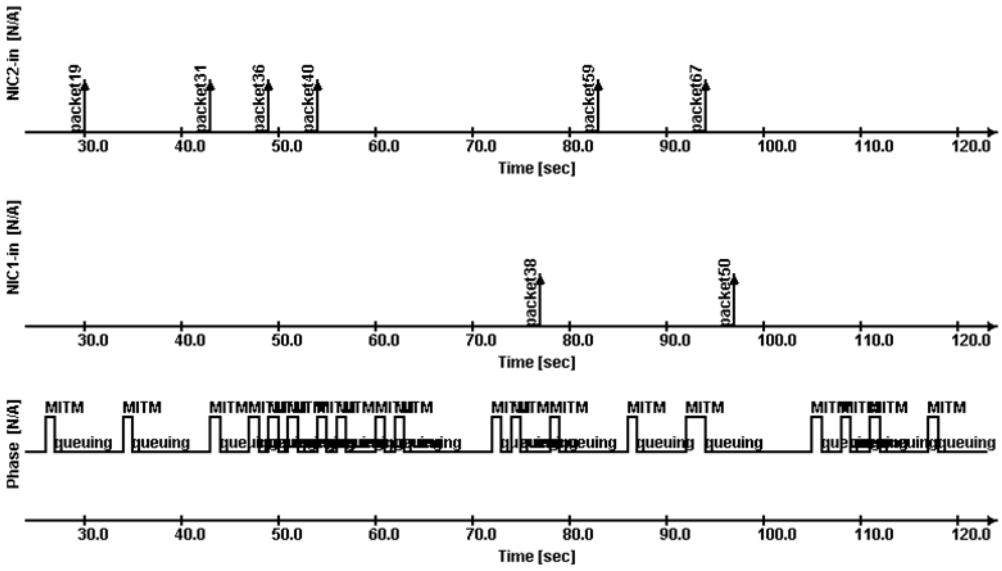
Şekil 6.21. Hedef, kaynak ve saldırgan(ortadaki adam) düğümlerin durumu

Kurban olarak seçilen cihazlar arasındaki bütün paket trafiği araya girip paketleri almak suretiyle dinlenmektedir. Bu duruma ait istatistikler ve elde edilen paketlere ait port izleme ekranı görüntüsü Şekil 6.23.'de görülmektedir. Saldırgan durumundaki düğümün üzerine fare işaretçisi odaklandığı zaman görünen yardımcı ileti penceresinden süzülen paket miktarı ve kuyrukta bekleyen paketler Şekil 6.22.'de görülmektedir.

Saldırgan düğüm yakaladığı paketleri istediği gibi manipüle ederek alıcı düğüme iletebilmektedir. Bu testi doğrulamak için pakete "Paket verisi manipüle edildi" verisi eklenerek konsol penceresinde paket içeriğindeki mesaj gösterilmiştir. Ortadaki adam saldırısına ait paket ve durum değişim grafikleri Şekil 6.23.'te gösterilmiştir.



Şekil 6.22. Araya girip dinleme yapan düğümün durumu ve verileri



Şekil 6.23. Ortadaki adam saldırısına ait paket ve durum değişim grafikler

6.7. SQL Enjeksiyonu

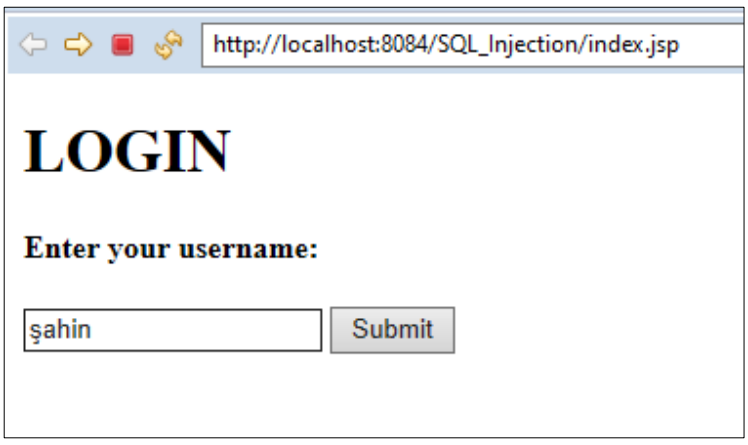
SQL, yapılandırılmış sorgu dili anlamına gelir. SQL, bir veritabanına bağlanmak ve iletişim kurmak için kullanılır. İlişkisel veritabanı yönetim sistemleri için standart bir dildir. SQL sorguları, veri girişi, güncellemeler ve kayıt silme gibi komutları yürütmek için kullanılır. SQLI olarak da bilinen SQL enjeksiyonu, görüntülenmesi

amaçlanmayan bilgilere erişmek için veritabanına yönelik kötü amaçlı SQL kodu kullanan yaygın bir saldırı türüdür. Bu bilgiler, kişisel veriler, hassas şirket verileri, müşteri bilgileri veya kullanıcı listeleri dahil olmak üzere pek çok öğeyi içerebilir. SQL enjeksiyon, web uygulamalarından alınan kullanıcı girdileri ile oluşturulan SQL sorgularının manipülasyonu olarak da tanımlanabilir [97].

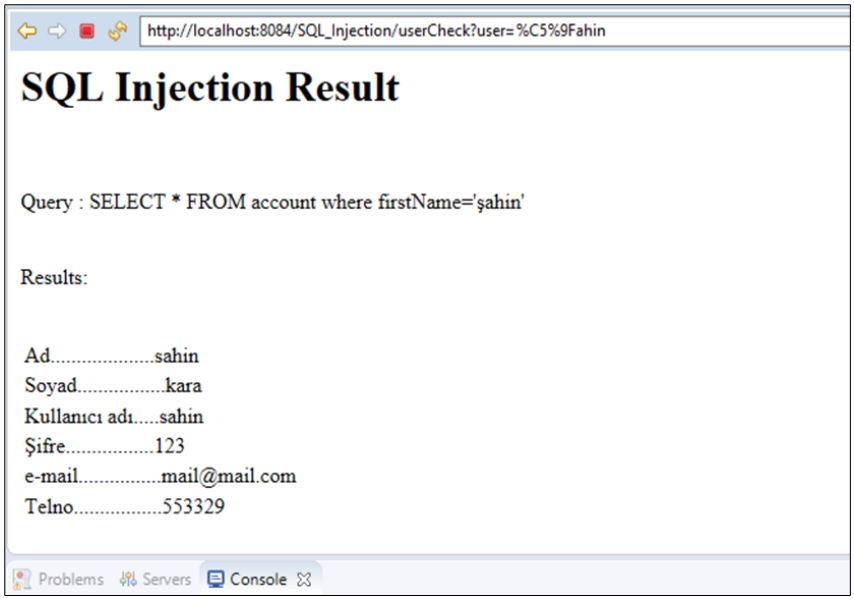
SQL enjeksiyonunun bir işletme üzerindeki etkisi geniş kapsamlıdır. Başarılı bir SQLI saldırı, kullanıcı verilerinin yetkisiz olarak görüntülenmesine, saldırganın bir veri tabanında yönetici yetkilerini kazanmasına ve tüm tabloların silinmesine sebep olabilir ve bunların tümü bir işletmeye büyük zarar verir.

Bir SQL enjeksiyonun potansiyel maliyetini hesaplarken ve kredi kartı bilgileri, adresler, telefon numaraları, gibi kişisel bilgilerin çalınması durumunda yaşanacak kayıplar göz önünde bulundurmak önemlidir. SQL enjeksiyonu herhangi bir SQL veritabanına saldırmak için kullanılabilirken, en çok hedefler web siteleri olmaktadır.

Bu çalışmada, web sayfalarında Java dilini kullanarak dinamik web sayfaları oluşturmamızı sağlayan bir Java teknolojisi olan Jsp- MySql tabanlı web uygulaması üzerinde, bir SQL enjeksiyonu saldırı örneği ve analizi sunulmuştur.



Şekil 6.24. Login sayfası



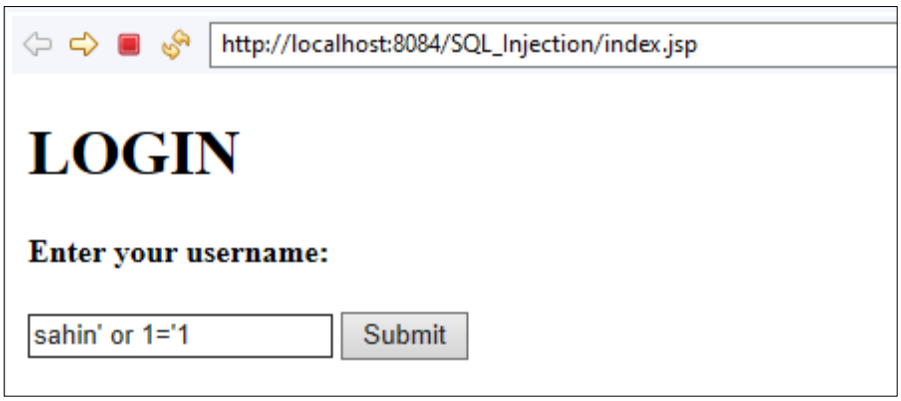
Şekil 6.25. Sql sorgu sonucu

Sql sorgularının manipülasyonunu göstermek için login sayfası olan index.jsp basit tutulmuştur. Şekil 6.24.'de "Login" sayfasından girilen kullanıcı adına göre önceden Mysql Workbench ile hazırladığımız veritabanına erişebilmek için gerekli sürücüler çalıştırılıp bağlantı kurulduktan sonra bu kullanıcıya ait kayıt varsa Şekil 6.25.'deki "Sql injection result" sayfasında listelenecektir.

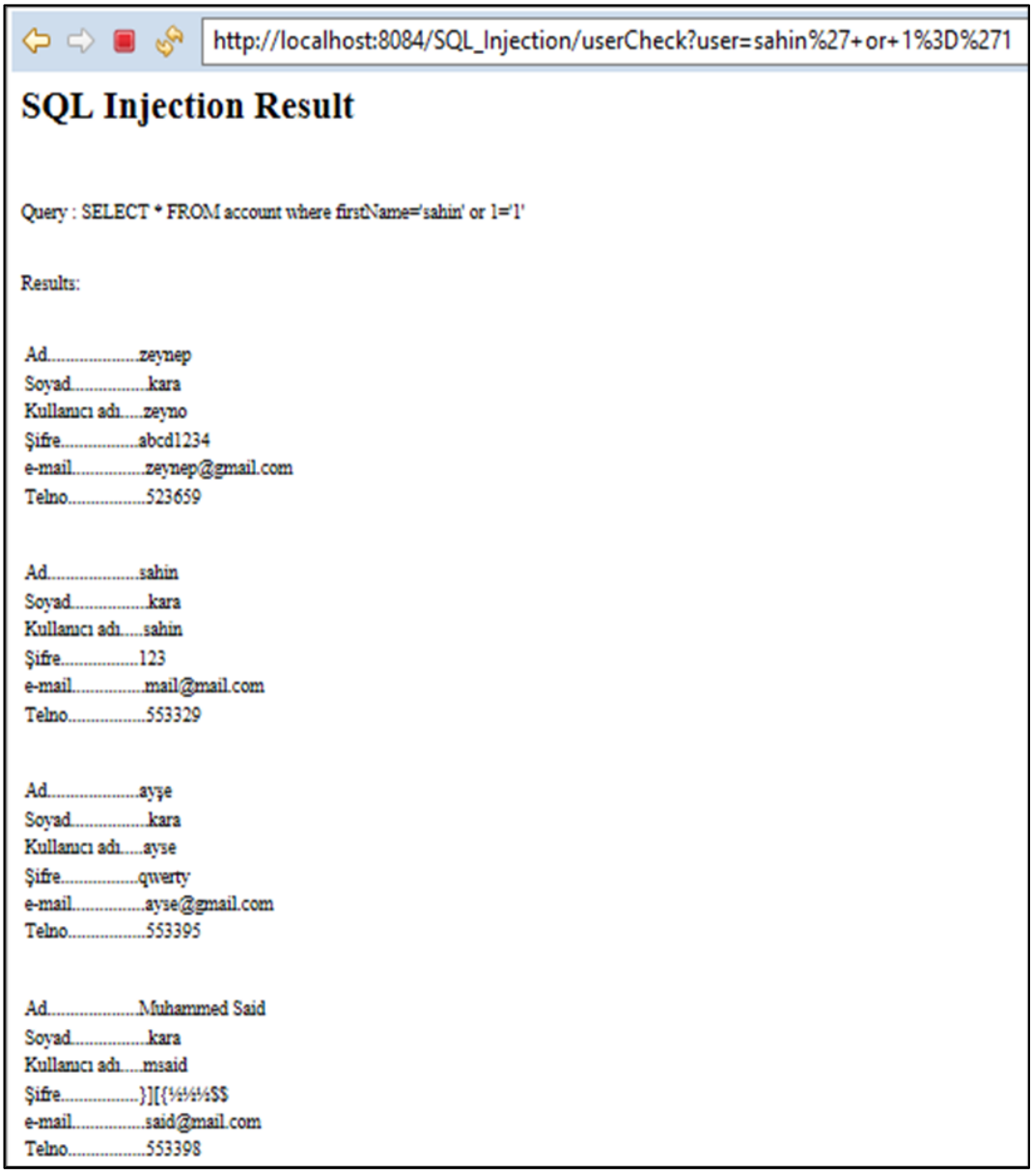
Şekil 6.25.'te sorgulama kullanıcı adına göre yapılmakta ve sadece girilen kullanıcı adına ait kayıt listelenmektedir.

Şekil 6.24.'de ve Şekil 6.25.'te yapılan işlemler normal kullanıcı davranışıdır ve normal kullanıcı adı girişi yapılmaktadır. Şekil 6.26.'da yine login sayfasında kullanıcı girişi yaparken kullanıcı adı yerine basit bir sql enjeksiyon ifadesi yazıyoruz. Bu yazılan ifade sorguyu manipüle edecek ve bütün kullanıcılara ait kayıtları listeleyecektir. SQL enjeksiyonunu yürütmek isteyen bir saldırgan, bir veritabanındaki doğrulanmamış giriş güvenlik açıklarından yararlanmak için standart bir SQL sorgusunu Şekil 6.26.'daki gibi manipüle eder.

SQL enjeksiyon ifadesi girilip çalıştırıldığında aşağıdaki Şekil 6.27.'de görüldüğü gibi veritabanında bulunan bütün kullanıcılara ait kayıtlar listelenmektedir. SQL enjeksiyon yöntemleri çoktur, burada sadece basit bir yöntem gösterilmiştir.



Şekil 6.26. Sql enjeksiyonu kod girişi



Şekil 6.27. Sql enjeksiyonu sonucu elde edilen kayıtlar

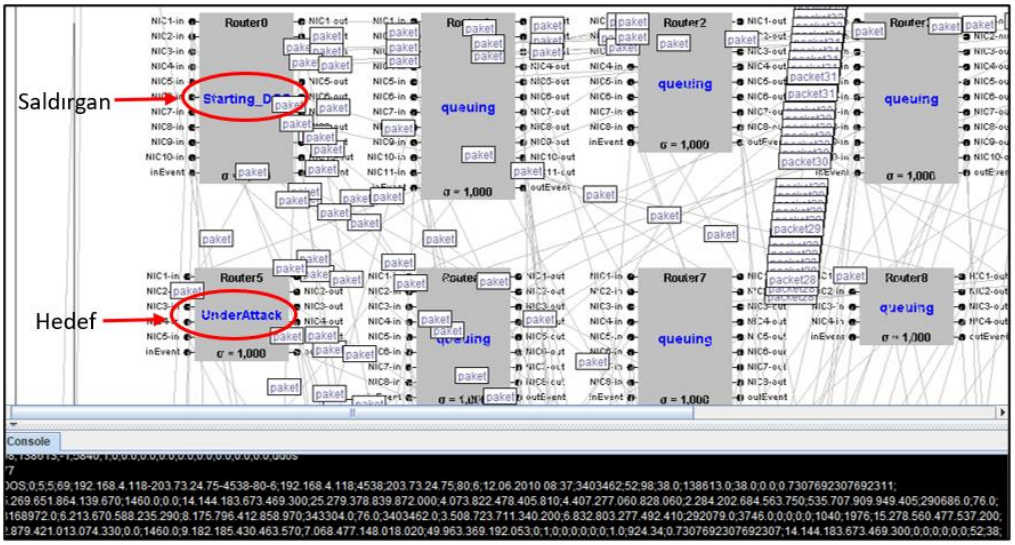
SQLI'yi filtrelemek için daha çok bir web uygulaması güvenlik duvarı (WAF) kullanılır. Bunu yapabilmek için, bir WAF tipik olarak, kötü niyetli SQL sorgularını detaylı olarak ayıklamasına izin veren büyük ve sürekli güncellenen bir imza listesini kullanmaktadır. Genellikle, böyle bir liste belirli saldırı vektörlerini ele almak için imzalar içerir ve yeni keşfedilen güvenlik açıklarını bertaraf eden engelleme kuralları getirmek için düzenli olarak yamalanır [98].

6.8. Yeni Bir Tespit ve Saldırı Önleme Yönteminin Uygulanması

Saldırı tespit sistemi, yüksek doğruluk oranında işlem yapabilse de, bir ağı saldırılara karşı fiilen savunmak için gereken işlemlerin yalnızca bir kısmını oluşturur. Saldırı tespit sistemi bir saldırının nedenini veya amacını belirleyebilir, ancak tek başına bir saldırıyı durdurmak için yeterli olmaz. Bu işlevi gerçekleştirmek için, algılanan kötü amaçlı etkinliğe tepki vermek amacıyla saldırı önleme sistemi geliştirilmiştir. Bu tezde, saldırı önleme amacıyla ağ trafiğinin incelendiği ve şüpheli etkinlik belirlendiğinde "refleks" tipi eylemlerin gerçekleştirildiği saldırı tespit sistemi geliştirilmiştir. Bu nedenle, saldırı simülasyonunda yapılan saldırılara tepki verebilecek bir yol izlenmiştir. Saldırı senaryosunu gerçekleştirmek için önceden tanımlanmış bir hedef IP adresine ve ilgili cihazın portlarına belirli bir kaynaktan paketlerin gönderimi sağlanır. Atomik model konfigürasyonunda normal ağ trafiği olarak kabul edilebilir durum dışında gerçekleşen anormal duruma göre saldırı uyarı alarmı verilir. Simülasyon deney sonuçlarının gösterildiği 7. Bölümde saldırı uyarıları ve risk seviyeleri detaylıca açıklanmıştır. Bu bölümde üretilmiş olan saldırı alarmına karşı sistemin karşı tepki yöntemi izah edilecektir.

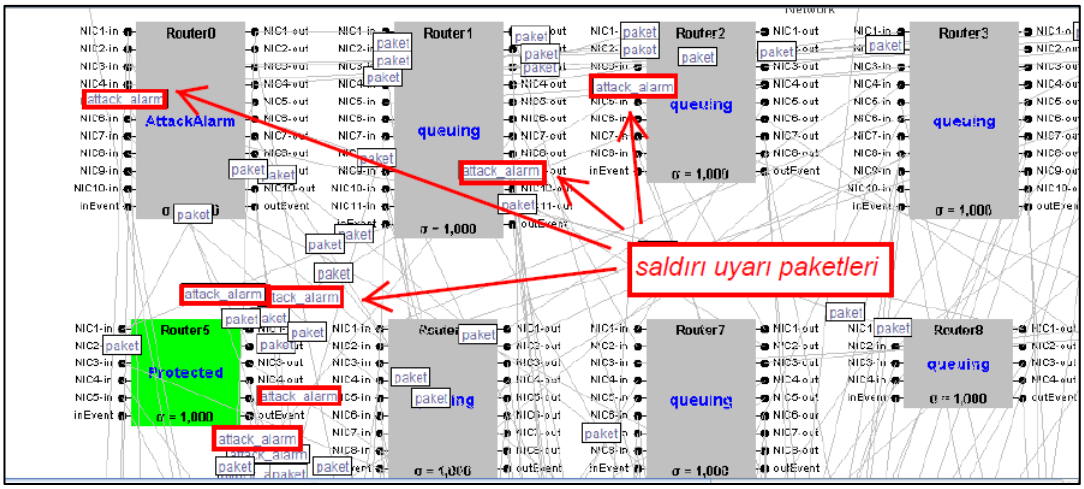
Trafiği gerçek zamanlı olarak analiz etmek için sürekli izlemeyi kullanmak, anormal durumları erken fark etmek, bilhassa DDoS etkinliğinin izlerini tespit etmek için önemli bir yöntemdir. Gerçek zamanlı izleme, saldırı tüm hızıyla devam etmeden önce bir saldırı girişimini tespit etmeyi ve erken önlem almayı sağlar. Bu çalışmada hazırladığımız DEVS tabanlı siber saldırı simülatöründe ağ trafiği gerçek zamanlı olarak izlenmektedir. Anormal durumlar için üretilen saldırı uyarı mesajları,

benzetim ağının sürekli izlenmesi ile gerçek zamanlı analiz neticesinde saldırı türüne göre Şekil 6.28.'deki gibi üretilen saldırı uyarılarıdır. Saldırı tespit sistemi herhangi bir saldırı tespit ettiğinde uygun saldırı uyarısı yayınladıktan sonra ağ trafiğini izlemeye ve saldırının ileri aşamalarını ve bu aşamalara ait uyarıları üretmeye devam etmektedir. Simülasyon ortamındaki saldırı altındaki çihazlarda saldırı seviyesi ile ilgili olarak gözlemciyi bilgilendirmek için farklı renklendirmeler kullanılmaktadır. Saldırı tespit sistemi bu aşamalarda saldırıyı engellemek noktasında herhangi bir etkinlik gerçekleştirmez.



Şekil 6.28. Hedef ve saldırgan düğüm saldırı başlangıcı durumları

Saldırı önlemek amacıyla gerçekleştirilecek işlemler saldırı engelleme mekanizması tarafından sağlanmaktadır. Bunun için saldırı engellemeye yönelik ilk adım olarak ağda saldırı tespit sisteminden farklı bir saldırı alarm mesajı yayınlanmaktadır. Bu alarm mesajı, saldırıya uğrayan cihaz tarafından Şekil 6.29.'da gösterildiği gibi kendi yönlendirme tablosundaki tüm komşu düğümlere duyurulur.



Şekil 6.29. Hedef düğümde ilk saldırı alarm uyarısı yayınımı

Saldırı alarmı alan düğümler de Şekil 6.30.'daki gibi kendi komşu düğümlerine bu saldırı alarmını duyuracak şekilde atomik model yapılandırılmıştır.

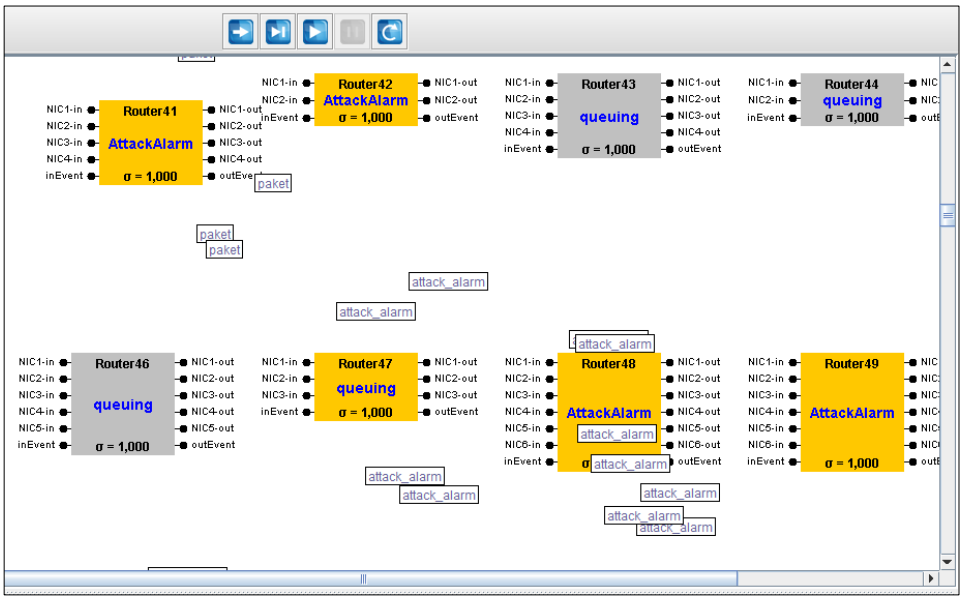
```
if(phaseIs("AttackAlarm") ) {

 for (int i = 0; i < getNumOutports(); i++){
      m.add(makeContent((String)
(getOutportNames().get(i)),
makePacketEvent("attack_alarm")));
                }
                return m;
        }

public  NETPacket makePacketEvent(String name){

                NETPacket p= new NETPacket(name);
                p.setId((byte) 19);
                p.setSource(address);
                p.setLength(20);
                p.setPriority((byte) 7);
                return p;
        }
```

```
IPAddress: 0.0.0.5
AS number: 0
QueueState: 35 packets(19KB)
Number of Routed Packets: 0
Number of Destined Packets: 20
Number of Discarded Packets: 0
1145 of 1486 LSA discarded
Neigbor Table :
====================
Number of Neigbors: 6
[dest:0.0.0.1 nextHop:0.0.0.1 Out Port:NIC2-in cost:1
, dest:0.0.0.83 nextHop:0.0.0.83 Out Port:NIC6-in cost:1
, dest:0.0.0.7 nextHop:0.0.0.7 Out Port:NIC3-in cost:1
, dest:0.0.0.3 nextHop:0.0.0.3 Out Port:NIC1-in cost:1
, dest:0.0.0.30 nextHop:0.0.0.30 Out Port:NIC5-in cost:1
, dest:0.0.0.28 nextHop:0.0.0.28 Out Port:NIC4-in cost:1
]
BGP Table :
====================
Number of networks: 0
{}
Topology Database:
====================
Number of LSA's: 99
[LSA_v0ofRouter83
, LSA_v0ofRouter90
, LSA_v0ofRouter28
, LSA_v0ofRouter81
, LSA_v0ofRouter30
, LSA_v0ofRouter45
, LSA_v0ofRouter38
, LSA_v0ofRouter66
, LSA_v0ofRouter92
, LSA_v0ofRouter31
```

(a) (b)

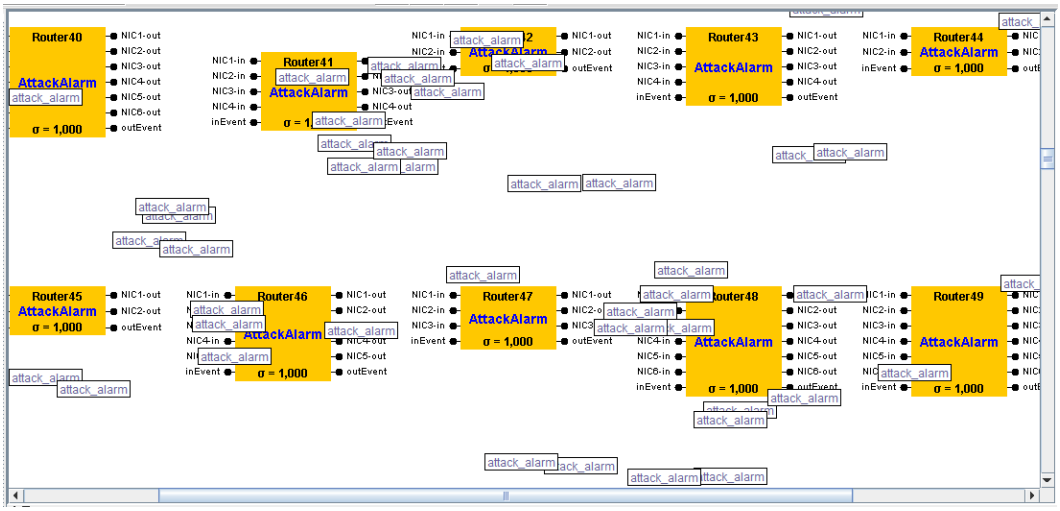
Şekil 6.30. (a) Atomik modelin saldırı alarm paketleri üreten kod kısmı (b) Hedef düğümün komşuluk tablosu

Saldırı altındaki cihaz Şekil 6.30.'daki kodlar ile üretilen saldırı alarm paketlerini Şekil 6.31.'de görüldüğü gibi çıkış portlarından komşuluk tablosundaki düğümlere ileterek saldırıyı bildirmektedir.



Şekil 6.31. Ağda saldırı alarmı yayınımı

Çok kısa bir süre içerisinde saldırı alarm mesajı ağdaki tüm düğümlere iletilmektedir. Bu saldırımı alarmını alan her düğüm Şekil 6.32.'de görüldüğü gibi "AttackAlarm" durumuna geçiş yapmaktadır.



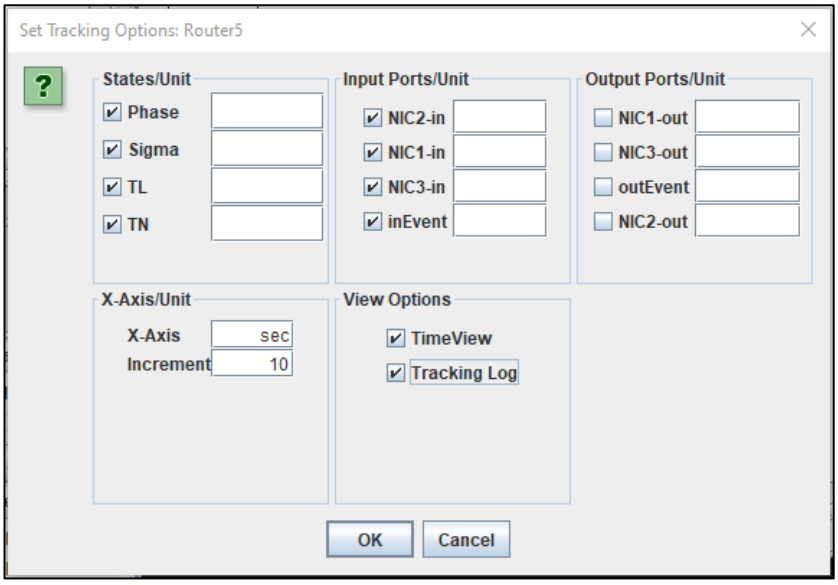
Şekil 6.32. Bütün düğümlerin saldırı alarm mesajlarını aldığı durum

Bundan sonraki aşamada ağ içerisindeki düğümlerin alınan saldırı alarmına karşı gösterecekleri reaksiyon başlatılmaktadır. Saldırı engellemede DoS ve DDoS saldırıları için güvenlik duvarı mantığı ile belirlediğimiz kriterlere göre süzme ve filtreleme yapılabilmektedir. Güvenlik duvarı mantığında UDP, ICMP, SYN gibi paketlere limit koymanın yanı sıra paket büyüklüğüne göre de filtreleme yapılmaktadır. Yaptığımız çalışmada siber saldırı simülasyonu testlerinde daha çok belirli bir kaynaktan gönderilen paket sayısındaki artışa göre yapılan saldırıya karşı ağdaki düğümlerin saldırgan düğümden gelen paketleri ağdan düşürmek suretiyle saldırıyı engelleme yöntemi kullanılmıştır. Yine başka bir yöntem olarak paket büyüklüğü de paket düşürme kriteri olarak testlerde kullanılmıştır. BruteForce saldırılarında engelleme kriteri olarak şifre deneme sayısı limiti kullanılmaktadır.

BÖLÜM 7. SİMÜLASYON DENEY SONUÇLARI

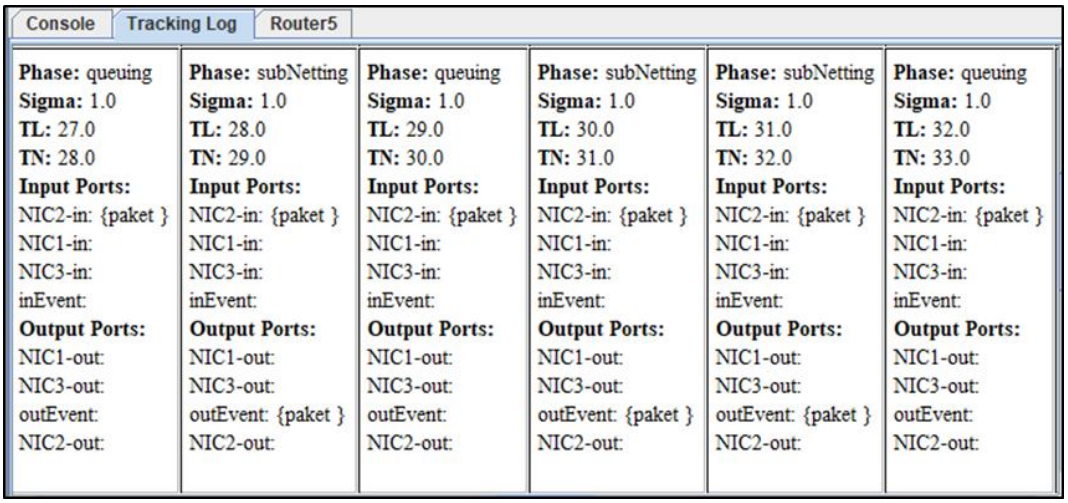
Saldırı simülasyonunda, saldırı modellerinin çıkışları olay üreteci atomik modelinin giriş portuna bağlanarak giriş olayı oluşturulur. Her adımda, saldırıyı gerçekleştiren cihazın çıkış portlarından, hedefi kurban bilgisayar olan ve saldırı yapılandırmasında sayısı belirlenen paketler gönderilir. Saldırı tipine bağlı olarak saldırı durumundaki saldırgan düğümün portlarındaki paket iletim yoğunluğu ve düğümün durum değişimi simülasyon görüntüsü Bölüm 6'da gösterilmiştir. Bu bölümde saldırı hedefindeki cihazda gerçekleşen olaylar, durum değişimleri, saldırı uyarıları ve portlarındaki trafik yoğunluğu grafiksel olarak gösterilerek elde edilen sonuçlara göre saldırının başarımı değerlendirilecektir.

İlk olarak DoS saldırı simülasyonunda hedef cihazın portlarında ve durumundaki değişimi değerlendirebilmek için saldırı altındaki hedef cihazın durum değişimi ve giriş-çıkış portlarındaki paket yoğunluk grafiklerinin izlenebilmesini sağlayan Şekil 7.1.'deki izleme seçenekleri yapılandırma arayüzünden gerekli seçimler yapılmaktadır.



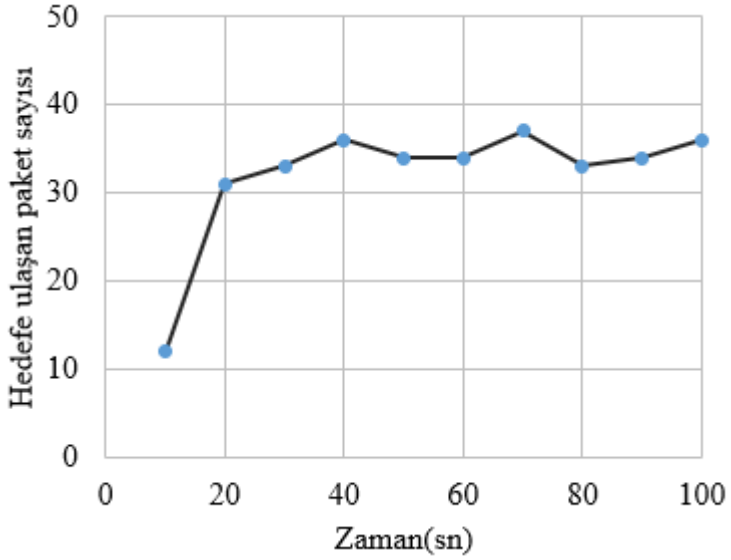
Şekil 7.1. İzleme seçenekleri yapılandırma penceresi

Bu arayüzde saldırıya ait log dosyasının oluşturulması için de "Tracking Log" görüntüleme seçeneği seçilmelidir. Saldırıya ait istatiksel grafiklerin elde edilmesi için Şekil 7.2.'de gösterildiği gibi tutulan log kayıtları kullanılacaktır. Saldırı senaryosu tamamlandıktan sonra saldırıya ait kayıtların kaybedilmemesi için simülasyon penceresi kapatılmadan önce log kaydının bir dosyaya kaydedilmesi gerekmektedir, aynı durum diğer saldırı türleri için de geçerlidir.



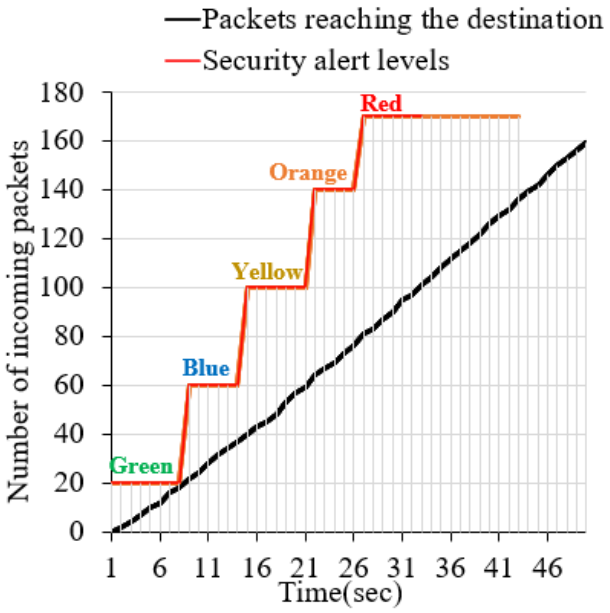
Şekil 7.2. Hedef cihazın log kaydı

Ağ simülasyonu bir DoS saldırısı ile başlasa bile ağ trafiği belirli bir yoğunluğa kadar bir süre normal şekilde işlemeye devam edecektir. Normal ağ trafiği devam ederken ve hizmet reddi saldırısı olmadığında bile düğüm tıkanıklığı oluşabilir. Ancak bunun bir DoS saldırısı olduğu anlamına gelmez. Belirli bir düğüme ulaşan paketler incelendiğinde, aynı kaynaktan gelen paketlerin fark edilebileceği şekilde yapılandırılır. Aynı kaynaktan gelen paket sayısı belirli bir değeri aştığında anormal bir durum olduğuna karar verilir ve DoS saldırı uyarısı verilir. DoS saldırısı başladıktan sonra, saldırgan kaynağından hedef düğüme birim zamanda geldiği tespit edilen paket sayısı Şekil 7.3.'de gösterilmiştir. Trafik yoğunluğuna veya kuyrukta bekleme sürelerine bağlı olarak anlık gelen paket sayısı değişmektedir.



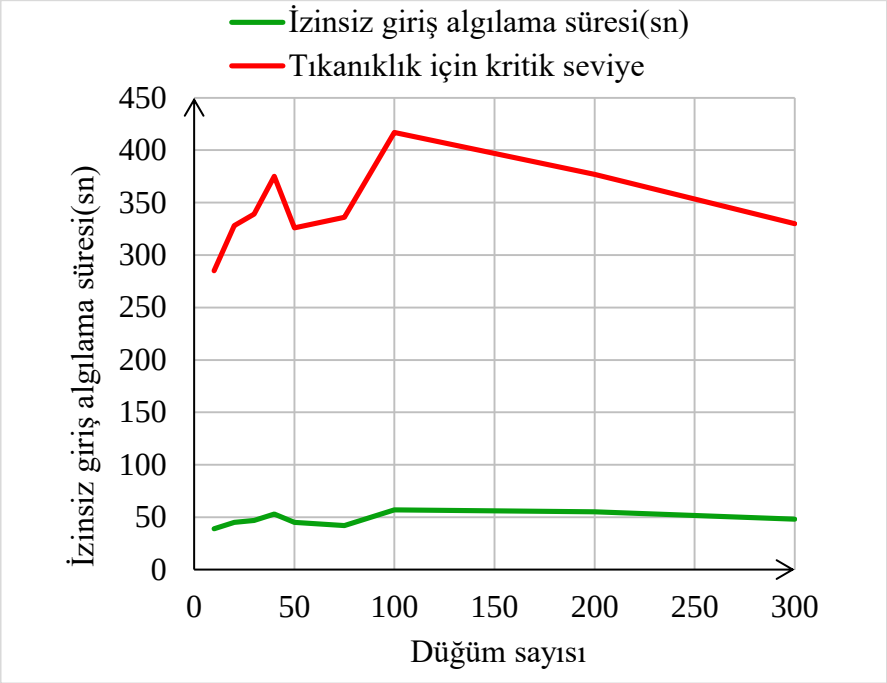
Şekil 7.3. Başarılı saldırı grafiği

Zamana bağlı olarak artan paket sayısına göre farklı güvenlik risk seviyeleri Şekil 7.4.'de belirlenmiştir. Belirlenen birim zamanda hedefe ulaşan paket sayısının 160 olduğu kırmızı seviye tıkanmanın başladığı seviyeyi göstermektedir. Simülasyon ortamındaki hedef cihazın rengi de grafikteki renklere göre değişmektedir. Bu da tehlikeyi fark etmek için gözlemciye kolaylık sağlamaktadır.



Şekil 7.4. Güvenlik uyarı seviyeleri

Belirli bir kaynaktan gönderilen paketler, belirli bir sayıya kadar herhangi bir anormalliğe neden olmaz. Bu durum yeşil seviye olarak gösterilmiştir. Atomik model konfigürasyonunda normal ağ trafiği olarak kabul edilebilir seviye için simülasyon süresi ile 10 saniyede 20 paket olarak konfigüre edilmiştir. Bu seviye aşıldıktan sonra anormal durum olarak kabul edilir ve bu anormal duruma göre DoS saldırı uyarı alarmı verilir. Ağdaki düğüm sayısına bağlı olarak uyarı alarm zamanlarının alt ve üst seviyeleri Şekil 7.5.'de gösterilmiştir. Saldırı tespit sürelerinin düğüm sayılarından fazla etkilenmediği grafikten anlaşılmaktadır.

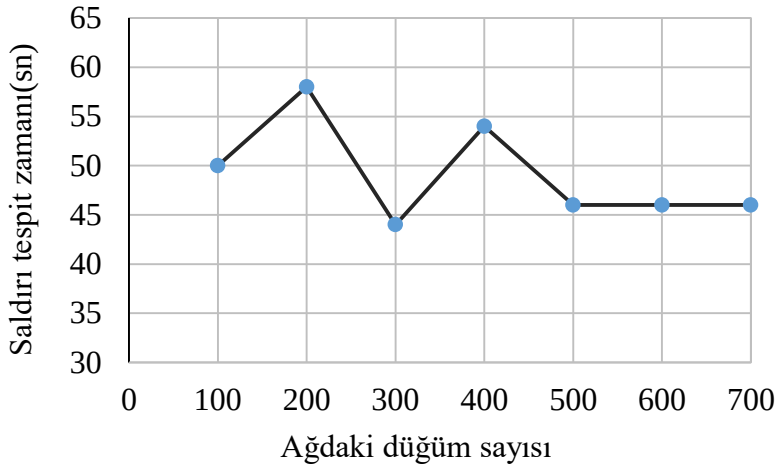


Şekil 7.5. Ağın boyutuna göre saldırı tespit zamanı

Saldırgan tarafından kurban cihaza yönelik yoğun olarak başlatılan paket trafiği, ağdaki yönlendirme tablolarına ve yönlendirme algoritmalarına göre farklı yollardan hedefe ulaşır. Simülasyon sırasında, paket trafiğinin yoğunluğu nedeniyle yoldaki bazı yönlendirici düğümlerinde tıkanıklık meydana gelebilir.

Bu çalışmada DoS saldırısı simülasyonu farklı büyüklükte ağlarda test etmek için saldırı simülasyonunun gerçekleştirileceği farklı ağ modelleri bir topoloji üreteci ile üretilmiştir. Saldırı simülatörüne entegre edilmiş olan topoloji üreteci ile küçük bir

ağdan çok büyük ölçekli ağlara kadar farklı boyutlarda ağların benzetimi yapılabilmektedir. Büyük ölçekli ağların benzetimi işlemci ve bellek kaynaklarını yüksek oranda kullanmaktadır. Binlerce yönlendiricinin bulunduğu bir ağ benzetim ortamında DoS saldırısı gerçekleştirildiği zaman uygulamayı barındıran bilgisayarın kaynakları yüksek oranda kullanıldığı için bekleme veya kilitlenmeye neden olabilmektedir. Farklı sayıda düğümü içeren ağlarda gerçekleştirilen DoS saldırısında saldırı tespit zamanları Şekil 7.6.'da gösterilmiştir.

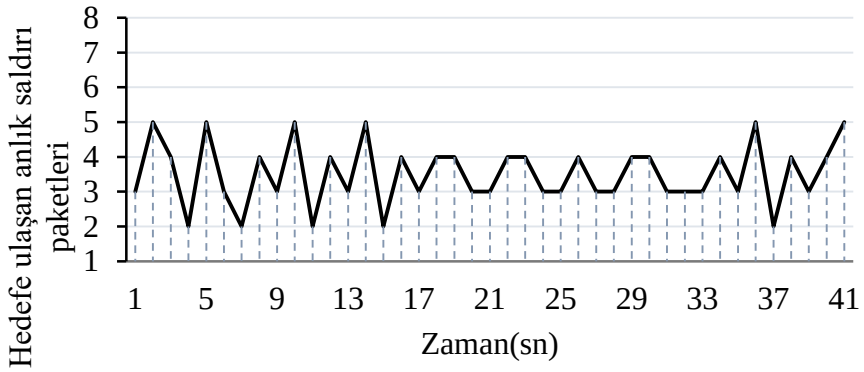


Şekil 7.6. Ağ boyutuna göre saldırı algılama süresi grafiği

DoS saldırısında saldırı tespiti, belirli bir süre aralığındaki hedefe ulaşan paket sayısı ile orantılı olarak hesaplanmaktadır. Oluşturulan yoğun ağ trafiği bazı düğümlerde tıkanmalara neden olmakta ve bu durumda paketler hedefe ulaşmak için farklı yollara yönlendirilmektedir. Bu duruma rağmen grafiklerden anlaşılacağı üzere saldırı tespit süreleri kararlı bir duruma gelip ağın büyüklüğünden fazla etkilenmemektedir. Bu sonuç, benzetimi yapılan ağlarda kullanılan yönlendirme protokollerinin ve algoritmalarının yüksek başarım oranına sahip olduğunu göstermektedir.

DDoS saldırıları, DoS saldırılarının bir alt sınıfıdır. Bir DDoS saldırısı, toplu olarak botnet olarak bilinen ve sahte trafikle hedef düğümü hizmet veremez duruma getirmek için kullanılan birden çok cihazı içerir. Saldırı emri alan botnetler hedefe yönelik yoğun bir paket trafiği başlatmaktadır. Her adımda üretilecek/gönderilecek paket sayısı saldırı başlamadan önce parametre ayarlarının yapıldığı formda

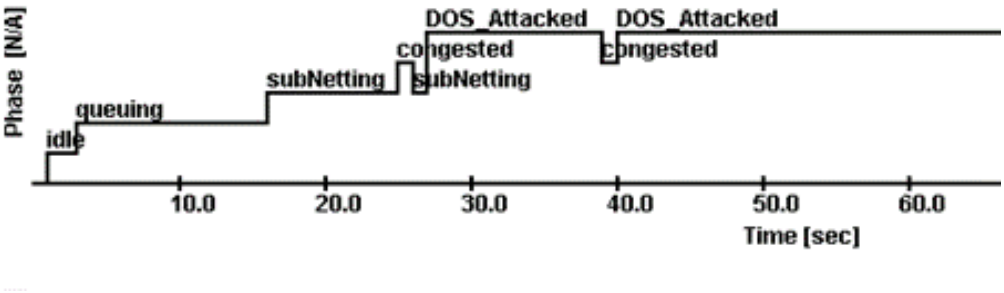
belirtilmektedir. Hedefe yönelik başlatılan yoğun trafik birden çok botnet(zombi) kaynaktan yapıldığı için hedefe giden yol üstündeki düğümlerde de tıkanıklıklara neden olmaktadır. Hedef düğüm yoğun paket akışına hedef olduğunu, düğüme gelen paket sayısı belli bir sayıyı geçtiğinde bunu "DoS_Attacked" durumuna geçerek göstermektedir. Hedef makinanın tampon alanı kısa sürede dolacaktır. Saldırı devam ettiği müddetçe tıkanıklığı sürecektir ve bu cihaz servis dışı kalacak, böylece DDoS saldırısı amacına ulaşacaktır. DoS saldırıları uyarı seviyeleri DDoS için de geçerlidir. DDoS saldırıları dağıtık botnet kaynaklarından yapıldığı için saldırı alarmı ve tıkanma durumu daha erken oluşmaktadır. Şekil 7.7.'de ve Şekil 7.8.'de hedef



Şekil 7.7. Anlık başarılı DDoS saldırısı

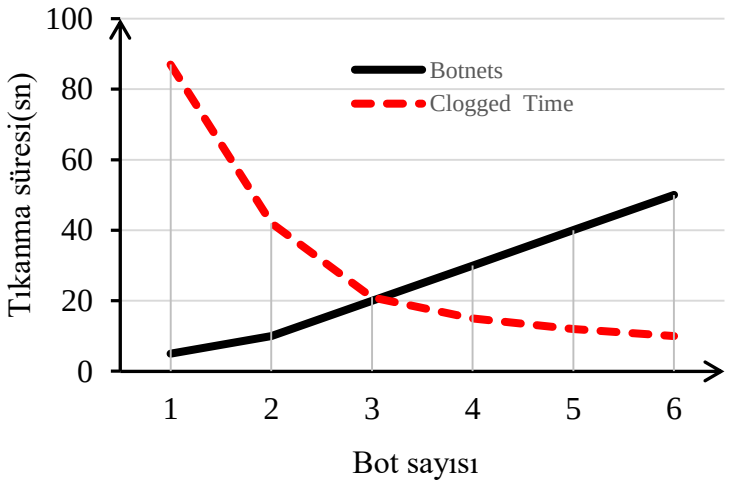
düğümde saldırıya ait grafiksel gösterimler görülmektedir. Ağ trafiği normal akışını sürdürürken saldırı başladıktan yaklaşık 20 saniye sonra hedef bilgisayarın giriş portlarında sıra dışı bir paket yoğunluğu oluştuğu görülmektedir. Hedef düğüm yoğun paket akışına hedef oluğunu, düğüme gelen paket sayısı belli bir sayıyı geçtiğinde bunu "DoS_Attacked" durumuna geçerek göstermektedir. Hedef makinanın tampon alanı kısa sürede dolmakta ve tıkanma durumuna(congested) geçtiği görülmektedir. Saldırı devam ettiği müddetçe tıkanıklığı sürmektedir ve bu cihaz servis dışı kalmaktadır ve DDoS saldırısı amacına ulaşmıştır.

Botnetlerin kullanıldığı bir DDoS saldırısının etkileri bot sayısına bağlı olarak değişmektedir. Bir hizmet reddi saldırısının temel amacı tıkanıklık oluşturup sunucuyu servis veremez duruma getirmek olduğu için aynı anda ne kadar çok istek gönderilirse tıkanma o kadar çabuk gerçekleşir



Şekil 7.8. DDoS saldırısına ait hedef düğüm durum geçiş grafiği

DDoS saldırılarının başarılı olabilmesi için olabildiğince farklı kaynaklardan hedefe mesaj/paket isteği göndermek gerekir. Botnetler daha çok bu amaçla kullanılmaktadır. Saldırıda kullanılan bot sayısı arttıkça tıkanma süresi kısalacaktır. Bu durum, simülasyon sonucunda elde edilen verilerle Şekil 7.9.'daki grafikle gösterilmiştir. Grafikten botnet sayısı ile tıkanma süreleri arasında bir ters orantı olduğu görülebilmektedir.

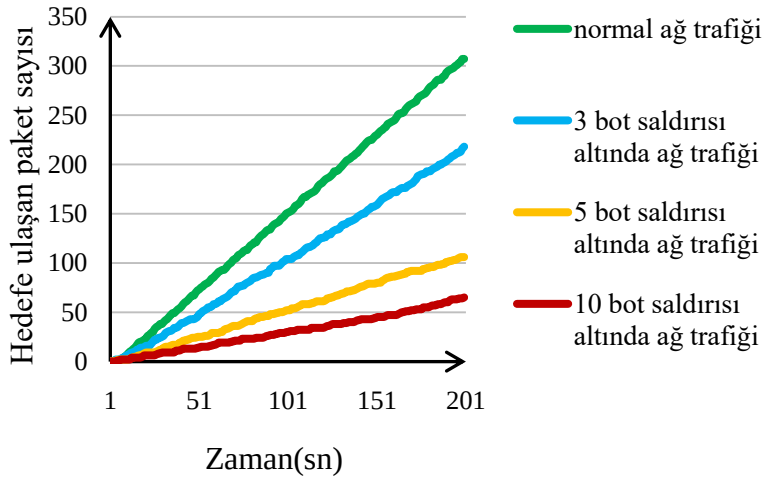


Şekil 7.9. Botnet sayısına karşı tıkanma zamanı

Şekil 7.9.'da 100 düğümlü bir ağda değişen sayıdaki zombi içeren botnetler ile yapılan DDoS saldırısında hedef düğümde bot sayısına bağlı olarak tıkanmanın gerçekleştiği zaman grafiği gösterilmektedir. Tek saldırgan ile tıkanma zamanı 90 sanniyede gerçekleşmesine karşın 6 bot ile bu süre 10 saniyeye kadar düşmektedir. Saldırı simülasyonunda bot sayısı teorik olarak ağdaki düğüm sayısından az

olmalıdır. Bu kurala bağlı kalarak ağdaki bot sayısı adım adım arttırılarak benzetimi yapılan ağda ağ trafiğinin nasıl etkilendiği gözlemlenmiştir.

Normal ağ trafiğinin DDoS saldırısından nasıl etkilendiğini göstermek için belli sayıdaki botnetlerle yapılan saldırıda elde edilen trafik verileri ile normal trafik verileri Şekil 7.10.'da birlikte gösterilmiştir. Grafikte görüldüğü üzere sabit bir ağdaki botnetlerin sayısı arttırıldıkça ağ trafiği orantılı olarak yavaşlamaktadır. Ağ trafiği hedefe ulaşan paket sayıları ile orantılı olarak gösterilmiştir.



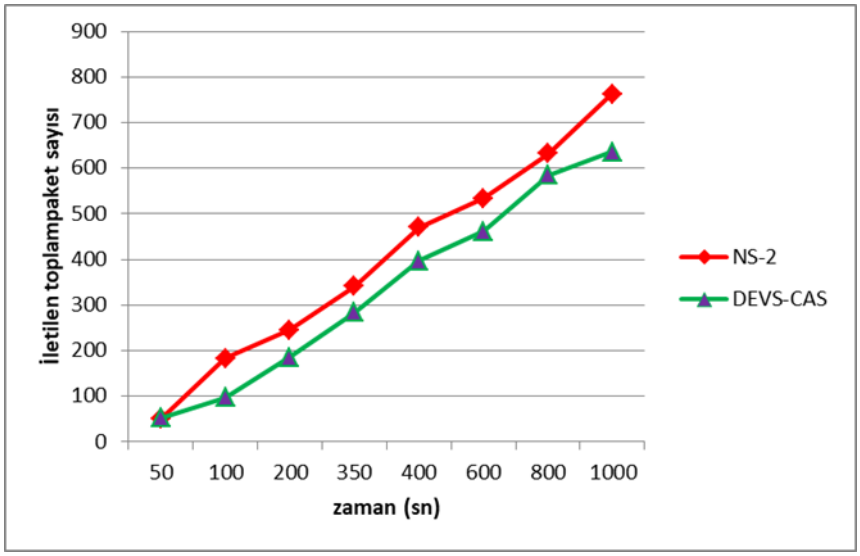
Şekil 7.10. Farklı bot sayıları ve DDoS saldırısı altında ağ trafiği çıkışı

.

7.1. Model Davranış Doğrulaması

Model davranış doğrulaması için yaygın olarak kullanılan NS2 simülasyon aracı kullanılmıştır. NS2'de yönlendiricinin bir paketi aldığı zamandan, paketin bağlantıda kullanıma sunulduğu zamana kadar hiçbir zaman harcanmaz, diğer tabirle yönlendirici işlemcisinin işlem yükünün neden olduğu yönlendirme gecikme süresi hesaba katılmaz. Sonuç olarak, gecikme farkı, seçilen soyutlamalardan ve varsayımlardan kaynaklanmaktadır. Örneğin simülatörlerdeki yönlendiricilerin işlem süresi değeri farklıdır. Bu değer NS2 simülatöründe sıfır iken DEVS simülatöründe sıfırdan büyük bir değerdir. Bu nedenle, ortalama uçtan uca gecikmeler farklıdır. DEVS-CAS ve NS2'nin modellenmesi arasındaki temel yapısal farklılıkları

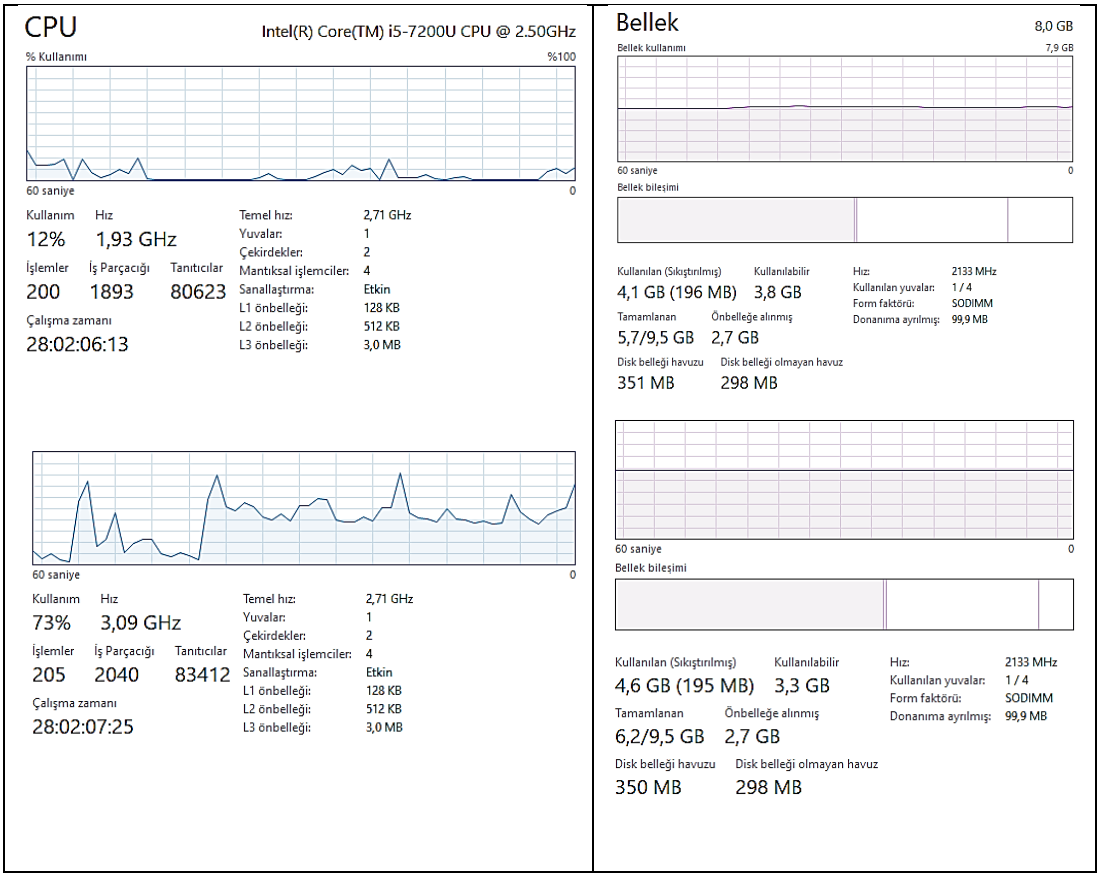
göstermek için deneylerde aynı konfigürasyonlar kullanılmıştır. Zamana bağlı iletilen paket sayıları Şekil 7.11.'de gösterilmiştir. Bir stabilizasyon aşaması süresinden sonra çıktı eğrileri neredeyse aynı ortalama değerlere yakınsar. Bu, her iki simülatörün aynı trafik konfigürasyonu için yakın çıktıları ürettiğini gösterir.



Şekil 7.11. İki simülatörün ağ çıkışlarının karşılaştırılması

7.2. DEVS-CAS Simülatöründe CPU ve Bellek Kullanımı

Şekil 7.12.'de 100 düğümlü bir ağ ile simülasyon çalıştırılmadan önce ve çalışması sırasında işlemci ve bellek kullanımı gösterilmiştir. Bu işlemlerin gerçekleştirilmesi sırasında Windows 10 sistem gözlemcisinden yararlanılmıştır. Grafikte simülasyon süresince işlemciye ait kullanım oranı ve bellek kullanım miktarına ait bilgiler görülmektedir. DEVS-CAS simülatöründe işlem zamanı boyunca işlemcinin %73 oranında kullanıldığı tespit edilmiştir. Simülatörün bellek kullanımı %65 oranında gerçekleştiği tespit edilmiştir.



Şekil 7.12. DEVS-CAS ile simülasyon başlangıcında ve simülayon sırasındaki sistemin durumları

BÖLÜM 8. SONUÇLAR VE DEĞERLENDİRME

Bir kurumsal ağı fiziksel olarak gerçekleştirmek ve bu ağlarda yeni siber güvenlik yöntemlerini test etmek maliyetlidir ve test verilerinin elde edilmesi de çok zaman alıcıdır. Kurumsal ağ tasarımı aşamasında ise güvenilir bir simülasyon aracı ile ağ tasarımının oluşturulması, güvenlik simülasyonlarının yapılması ve ağ tasarımlarının doğrulanması maliyet ve zaman tasarrufu sağlamaktadır. Siber ortamın güvenliğinin sağlanabilmesi için simülasyon ortamlarında sürekli olarak güvenlik testlerinin yapılması gerekmektedir.

Bu bağlamda, siber saldırılara karşı güvenliği etkin bir şekilde sağlamak maksadıyla kullanılan saldırı önleme uygulamaları için saldırı tespit uyarı verileri kritik öneme sahiptir. Bu tez kapsamında geliştirilen siber saldırı simülatörü, simülasyon ortamında tasarlanan ağdaki belirli siber saldırılara ilişkin uyarı verilerinin verimli bir şekilde elde edilmesi için bir araç sunmaktadır. Bu araç belirli saldırı türleri için uyarı verilerini elde etmek maksadıyla kullanılsa da, daha farklı saldırı türlerinin uyarı verileri oluşturmak için genişletilebilir bir altyapı sağlar. Geliştirilen uygulama, simüle edilen ağ üzerinde saldırı modelleri çalıştırabilme ve sonuçlarını izleyebilme özelliğine sahiptir. Bu tezde, büyük ölçekli kurumsal ağların kolaylıkla tasarlanabileceği ve geçerli düzeyde performans, ölçeklenebilirlik ve doğruluk ile siber güvenlik testlerinin kısa sürede yapılabileceği görülmüştür.

Uygulama, ağ benzetiminin oluşturulmasına izin veren ağ modelleme yetenekleri ile ayrıntılı saldırı senaryoları oluşturmak ve benzetimi yapılan ağ modeli üzerinde saldırı eylemlerini simüle etmek için kullanılan saldırı modelleme yetenekleri sağlar. DEVS-CAS ek olarak saldırı eylemleri ile ilişkili ağ trafiğinin modellenmesini, saldırıların algılanması ve uygun saldırı uyarılarının üretilmesini içermekltedir. Oluşturulan saldırı türü ve bir saldırının ağ üzerinden ilerleme biçimi üzerinde önemli bir kontrol sağlayan ve pek çok simülasyon parametrelerini belirleyebilen

görsel bir arayüzü tanımlanmıştır. Ağ modelinin işlevselliği ve saldırı simülasyonu, bazı farklı yaklaşımlarla doğrulanır. Bağlantılar, IP adresleri, log kayıtlarını da içeren ağ modelleme özelliklerinin birçoğu ağ modelinin gözlemlenmesi yoluyla görsel olarak doğrulanır. Saldırı parametreleri tarafından sağlanan kontrol seviyeleri ve diğer etkiler simülasyon gözlemlenerek doğrulanır. Saldırı uyarılarının üretimi, simüle edilen belirli saldırı eylemlerine karşı saldırı hedefindeki cihazın log çıktıları kontrol edilerek doğrulanır.

Bu tezde geliştirilen DEVS-CAS çerçevesi, hem modellenmiş bir ağ üzerinden saldırıların ilerlemesini hem de bu tür saldırılar sonucunda doğru IDS uyarılarının oluşturulmasını başarılı bir şekilde simüle etmektedir. Ağ modelleri ve saldırı senaryoları çok detaylı bir şekilde oluşturulabilir ve kontrol edilebilir. Ek olarak, saldırı senaryoları kolayca çalıştırılabilir ve değiştirilebilir.

Artan siber tehditlere karşı sürekli olarak yeni araçlar ve yöntemler geliştirilmektedir. Mevcut siber güvenlik araçlarının ve geliştirilen yöntemlerin test edilmesi için bilimsel araştırmalara ihtiyaç duyulmaktadır. Sanal test ortamlarında test sonuçlarının gerçekliğini artırmak için simülasyon araçlarının yetenekleri detaylı olarak incelenmelidir. Siber güvenlik araştırmalarının daha iyi yapılabilmesi için üniversitelerde siber güvenlik uygulama laboratuvarlarının açılmasının teşvik edilmesi ve eğitim süreçlerine siber güvenliğin eklenmesi ile mümkün olacaktır.

.

KAYNAKLAR

[1] Undercoffer, J., Joshi, A., & Pinkston, J., Modeling computer attacks: An ontology for intrusion detection. In International Workshop on Recent Advances in Intrusion Detection. Springer, Berlin, 113-135, 2003.

[2] Scarfone, K., Souppaya, M., Cody, A., & Orebaugh, A., Technical guide to information security testing and assessment. NIST Special Publication, 800(115): 2-25, 2008.

[3] Rai, M., & Mandoria, H., A study on cyber crimes cyber criminals and major security breaches. Int. Res. J. Eng. Technol., 6(7): 1-8, 2019

[4] McClure, S., Scambray, J., & Kurtz, G., Network Security Secrets And Solutions, 5th Edition (Hacking Exposed). McGraw-Hill Osborne Media, 1-692, 2005.

[5] Dougherty, E.T., & Gonslaves, P.G., Adaptive cyber-attack modeling system. Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense., SPIE, Bellingham, 6201: 620104, 2006

[6] Cheung, S., Lindqvist, U., & Fong, M. W. (2003, April). Modeling multistep cyber attacks for scenario recognition. IEEE 1:284-292, 2003

[7] Sudit, M., Stotz, A., & Holender, M., Situational awareness of a coordinated cyber attack. SPIE, 5812: 114-129, 2005

[8] Ashish Garg, Shambhu Upadhyaya and Kevin Kwiat, Attack Simulation for Measuring Detection Model Effectiveness, Second Secure Knowledge Management Workshop (SKM 2006), Polytechnic University, Brooklyn, NY, 28 - 29, 2006.

[9] DeLooze, L.L., Graig, C., McKean, P., & Mostow, J.R. (2004). Incorporating simulation into the computer security classroom. In: 34th Annual Frontiers in Education. FIE 2004, IEEE, 3: 13-18, 2004.

[10] Kuhl, M., & Kistner, J., Generation of synthetic cyber attack data using simulation. Final report for AFRL/IF VFRP Program, Rome, NY, 25-37, 2005

[11] Kistner, J., Cyber Attack Simulation and Information Fusion Process Refinement Optimization Models for Cyber Security. Rochester Institute of Technology, Kate Gleason College ofEngineering, Department of Industrial & Systems Engineering, Masters Thesis, 2006.

[12] Cohen, F., Simulating cyber attacks, defences, and consequences. Computers and Security, 18(6): 479–518., 1999.

[13] Kotenko, I.; Man'kov, E., Experiments with simulation of attacks against computer networks, International Workshop on Mathematical Methods, Models, and Architectures for Computer Network Security, Springer, 2776: 183–194, 2003.

[14] Kotenko, I., Ulanov, A., Agent-based simulation of DDOS attacks and defense mec-hanisms. Journal of Computing, 4(2): 16–37, 2005.

[15] Kuhl, M. E., & Sudit, M., Cyber Attack Modeling and Simulation for Network Security Analysis. IEEE Simulation Conference, 1180–1188, 2007

[16] Van Leeuwen, B., Urias, V., Eldridge, J., Villamarin, C., & Olsberg, R., Performing cyber security analysis using a live, virtual, and constructive (LVC) testbed. Proceedings - IEEE Military Communications Conference MILCOM, 1806–1811, 2010.

[17] Barreto, A. B., Hieb, M., & Yano, E., Developing a Complex Simulation Environment for Evaluating Cyber Attacks Developing a Complex Simulation Environment for Evaluating Cyber Attacks, 12248: 1-9, 2012.

[18] Torres, G., Smith, K., Buscemi, J., Doshi, S., Duong, H., Xu, D., Pickett, H. K., Distributed StealthNet (D-SN): Creating a live, virtual, constructive (LVC) environment for simulating cyber-attacks for test and evaluation (T&E). Proceedings - IEEE Military Communications Conference MILCOM, 1284–1291, 2015

[19] Norman, Mr Ryan, and Mr Christopher E. Davis., Cyber Operations Research and Network Analysis (CORONA) Enables Rapidly Reconfigurable Cyberspace Test and Experimentation, Modelıng & Simulation Coordination Office Publication, 15-24, 2013.

[20] Kotenko, I., & Chechulin, A., A Cyber Attack Modeling and Impact Assessment Framework, 5th International Conference on In Cyber Conflict, IEEE, 1–24, 2013.

[21] Moskal, S., Kreider, D., Hays, L., Wheeler, B., Yang, S. J., & Kuhl, M. Simulating attack behaviors in enterprise networks. IEEE Conference on Communications and Network Security, 359–360, 2013.

[22] Ekelhart, A., Kiesling, E., Grill, B., Strauss, C., & Stummer, C. Integrating attacker behavior in IT security analysis: a discrete-event simulation approach. Information Technology and Management, 16(3): 221–233, 2015.

[23] Bergin, D. Cyber-attack and defense simulation framework. Journal of Defense Modeling and Simulation, 2(4): 383–392, 2015.

[24] Park, J. S., Lee, J.-S., Kim, H. K., Jeong, J.-R., Yeom, D.-B., & Chi, S.-D. Secusim: A tool for the cyber-attack simulation. Information and Communications Security, Springer, 471–475, 2001.

[25] Langner, R., Stuxnet: Dissecting a cyberwarfare weapon. IEEE Security & Privacy, 9(3), 49-51, 2011

[26] Schatz, Daniel; Bashroush, Rabih; Wall, Julie. Towards a More Representative Definition of Cyber Security, Journal of Digital Forensics, Security and Law, 12 (2): 53-72, 2017.

[27] Kianpour, Mazaher; Kowalski, Stewart; Øverby, Harald, Systematically Understanding Cybersecurity Economics: A Survey, 13 (24): 13677, 2021.

[28] Stevens, Tim, Global Cybersecurity: New Directions in Theory and Methods, Politics and Governance. 6 (2): 1–4, 2018.

[29] Rainer, R.K., Cegielski, C.G, Ethics, privacy, and information security. Introduction to Information Systems: Supporting and Transforming Business, 3: 70–121, 2010.

[30] B. Schneier, Secrets And Lies: Digital Security in A Networked World. John Wiley & Sons, 2011.

[31] Kizza, J. M., Kizza, W., & Wheeler, Guide to computer network security, 387-411, Springer, 2013.

[32] M. Rudner, Cyber-threats to critical national infrastructure: An intelligence challenge, International Journal of Intelligence and CounterIntelligence, 26(3), 453–481, 2013.

[33] A. J. Duncan, S. Creese, and M. Goldsmith, Insider attacks in cloud computing, Trust, Security and Privacy in Computing and Communications, IEEE 11th International Conference on. IEEE, 857–862, 2012.

[34] J. Sheldon, State of the art: Attackers and targets in cyberspace, Journal of Military and Strategic Studies, 14(2): 1-19, 2012.

[35] S. Hansman and R. Hunt, A taxonomy of network and computer attacks, Computers Security, 24(1): 31–43, 2004.

[36] Igure, V. M., & Williams, R. D., Taxonomies of attacks and vulnerabilities in computer systems. IEEE Communications Surveys & Tutorials, 10(1):6-19, 2008.

[37] Friedman J., and Hoffman D. V., Protecting data on mobile devices: A taxonomy of security threats to mobile computing and review of applicable defenses. Information, Knowledge, Systems Management,7(1): 159-180, 2008.

[38] Myers C., Powers S., and Faissol D., Taxonomies of cyber adversaries and attacks: a survey of incidents and approaches. Lawrence Livermore National Laboratory, 7: 1-22, 2009.

[39] Singh P. K., Vatsa A. K.,Sharma R., & Tyagi P., Taxonomy based intrusion attacks and Detection management scheme in peer-to-peer network, International Journal of Network Security & Its Applications (IJNSA), 4(5): 167-179, 2012.

[40] Ye N., Newman C., and Farley T., A system-fault-risk framework for cyber attack classification. Information, Knowledge, Systems Management, 5(2): 135- 151, 2006.

[41] Avizienis, Algirdas,. Laprie JC., Randell Brian., and Landwehr Carl., Basic concepts and taxonomy of dependable and secure computing. Dependable and Secure Computing, IEEE Transactions,1(1): 11-33, 2004.

[42] Bråthen A., Correlating IDS alerts with system logs by means of a network-centric SIEM solution, Master's Thesis, Department of Computer Science and Media Technology, Gjøvik University, 2011.

[43] Collins M., Gates C., and Kataria G., "A model for opportunistic network exploits: The case of P2P worms." In Workshop on the Economics of Information Security (WEIS), University of Cambridge, 2006.

[44] Dodiya, B., & Singh, U. K., Identification of Taxonomic Features through Assessment ofExisting Taxonomies for Vulnerabilities Identification. International Journal of Computer Applications, 174(31), 14–22, 2021

[45] Kjaerland M., "A taxonomy and comparison of computer security incidents from the commercial and government sectors." Computers & Security, 25(7): 522-538, 2006.

[46] Lough D. L., A taxonomy of computer attacks with applications to wireless networks. PhD thesis, Virginia Polytechnic Institute and State University, 2001.

[47] Mishra, B. K., & Saini, H., Cyber attack classification using game theoretic weighted metrics approach. World Applied Sciences Journal, 7: 206-215, 2009

[48] Monahan-Pendergast, MaryTheresa., Attack Evolution: Identifying Attack Evolution Characteristics to Predict Future Attacks, Institute of Systems Research University of Maryland, PhD Thesis, 2006.

[49] Nasr K., El Kalam A. A., and Fraboul., Generating Representative Attack Test Cases for Evaluating and Testing Wireless Intrusion Detection Systems. International Journal of Network Security & Its Applications (IJNSA), 4(3): 1-19, 2012.

[50] Nunes S. R., Web attack risk awareness with lessons learned from high interaction honeypots, PhD thesis, Carnegie Mellon University, 2009.

[51] Rutkowska J., Introducing stealth malware taxonomy, COSEINC Advanced Malware Labs, 1(1), 1-9, 2006.

[52] Saber M., Bouchentouf T., Benazzi A., and Azizi M., Amelioration of attack classifications for evaluating and testing intrusion detection system. Journal of Computer Science, 6(7): 716-722, 2010.

[53] Shanto Roy, Nazia Sharmin, Jaime C. Acosta, Christopher Kiekintveld, and Aron Laszka., Survey and Taxonomy of Adversarial Reconnaissance Techniques. ACM Comput. Surv. https://doi.org/10.1145/3538704, 2022

[54] M.U. Nisa, Mehr, and Kashif Kifayat, Detection of slow port scanning attacks, International Conference on Cyber Warfare and Security (ICCWS). IEEE, 1-7, 2020.

[55] List of TCP and UDP port numbers https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers, Erişim Tarihi: 21.07.2022.

[56] Gunasekhar, P. T Gunasekhar, K.Thirupathi Rao and P. Lakshmi, A survey on denial of service attacks, International Journal of Computer Science & Information Technologies (IJCSIT), 5(2), 2373–2376, 2014.

[57] Mahjabin, T., Xiao, Y., Sun, G., & Jiang, W., A survey of distributed denial-of-service attack, prevention, and mitigation techniques, International Journal of Distributed Sensor Networks, 13(12), 1550147717741463, 2017.

[58] Baitha, A. K., Vinod, S., Session hijacking and prevention technique, International Journal of Engineering and Technology, 7(2.6), 193-198, 2018.

[59] Watson, D., The evolution of web application attacks, Network Security, Elsevier, 2007(11), 7-12, 2007.

[60] Kanawat, S. D., & Parihar, P. S., Attacks in wireless networks, International Journal of Smart Sensors and Adhoc Networks, 1(1), 113-116, 2011.

[61] Kaur, T., Malhotra, V., and Singh, D., Comparison of network security tools-firewall, intrusion detection system and Honeypot, International Journal of Enhanced Research in Science Technology & Engineering, 3(2), 200-204, 2014.

[62] Sharma A., Kalbarczyk Z., Iyer R., and Barlow J., Analysis of credential stealing attacks in an open networked environment. In Proc. Of the Fourth International Conference on Network and System Security. Washington, DC, USA: IEEE Computer Society, 144-151, 2010

[63] Robert E. Shannon. Simulation modeling and methodology, SIGSIM Simul. Dig., 8(3): 33–38, 1977.

[64] Ndih, Eugene David Ngangue, and Soumaya Cherkaoui., Methods, techniques and tools of computer systems and networks. Modeling and simulation of computer networks and systems, Morgan Kaufmann, Burlington, 485-504, 2015

[65] Zeigler, B. P., Theory of discrete event specified models: Modularity, hierarchy, experimental frames, International Journal Of General System, 10(1), 57-84, 1984.

[66] Antoine-Santoni, T., Santucci, J. F., De Gentili, E., & Costa, B., Discrete event modeling and simulation of wireless sensor network performance, Simulation, 84(2-3), 103-121, 2008.

[67] Zengin, A., Dağıtık Simulasyon Sistemleri İçin Yeni Bir Yönlendirme Algoritması ve Uygulaması, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği Bölümü, Doktora Tezi, 2004.

[68] Bischack, D. P., and Roberts, S. D., Object-Oriented Simulation. Proceedings of the 1991 Winter Simulation Conference, North Carolina, 194-203, 1991.

[69] Joines, J. A., and Roberts, S. D., Object-Oriented Simulation. Handbook of Simulation, Banks, J. John Wiley & Sons, New York, 397-427, 1998.

[70] Singh, R., & Sarjoughian, H. S., Software Architecture for Object-Oriented Simulation Modeling and Simulation Environments: Case Study and Approach. Computer Science & Engineering Dept., Arizona State University, Tempe, AZ, Tech. Rep., 2003

[71] Brunner, D. T., and Schriber, T. J., Inside Discrete-Event Simulation Software: How it Works and Why it Matters. Proceedings of the 2005 Winter SimulationConference, New Jersey, 167-177, 2005.

[72] Bisgambiglia, P. A., & Bisgambiglia, P., DecPDEVS: New Simulation Algorithms to Improve Message Handling in PDEVS. Open Journal of Modelling and Simulation, 9(2), 172-197, 2021.

[73] Musa, Ahmad, and Irfan Awan., Functional and Performance Analysis of Discrete Event Network Simulation Tools, Simulation Modelling Practice and Theory, Elsevier, 102470, 2021.

[74] Çavuşoğlu, Ü., Zengin, A., NS-2 ve NS-3 ağ simülatörlerinin ölçeklenebilirlik analizi ve karşılaştırma. Bilişim Teknolojileri Dergisi, 5(3), 41-50, 2012

[75] A. Hassan, "VANET Simulation," Masters Thesis in Electrical Engineering, School of Information Science, Computer and Electrical Engineering, Halmstad University, 2009.

[76] Kabir, M. H., Islam, S., Hossain, M. J., and Hossain, S., Detail comparison of network simulators. International Journal of Scientific & Engineering Research, 5(10), 203-218, 2014.

[77] Gupta, S. G., Ghonge, M. M., Thakare, P. D., and Jawandhiya, P. M., Open-source network simulation tools: An overview. International Journal of

Advanced Research in Computer Engineering & Technology (IJARCET), 2(4), 1629-1635, 2013.

[78] Lee, S., Ali, J., and Roh, B. H., Performance comparison of software defined networking simulators for tactical network: Mininet vs. OPNET, International Conference on Computing, Networking and Communications (ICNC), IEEE, 197-202, 2019.

[79] Chaudhari, K., Karule, P. T., Information about Simulation Software for Testing of Wireless Network. Journal of Information, 1(1), 12-22, 2015.

[80] Dorathy, I., & Chandrasekaran, M., Simulation tools for mobile ad hoc networks: a survey. Journal of applied research and technology, 16(5), 437-445, 2018.

[81] Orooji, F., Sarjoughian, H. S., and Taghiyareh, F., Modeling & simulation of educational multi-agent systems in DEVS-suite, 5th International Symposium on Telecommunications, IEEE, 956-961, 2010.

[82] Object-Oriented Simulation, https://repository.lib.ncsu.edu/bitstream/handle /1840.4/5531/1991_0032.pdf? sequence=1, Erişim Tarihi: 17.01.2022.

[83] Cobanoglu, B., Zengin, A., Ekiz, H., Celik, F., Kiraz, A., Kayaalp, F., Implementation of DEVS based distributed network simulator for large-scale networks. International Journal of Simulation Modelling IJSIMM, 13(2): 147–158, 2014.

[84] Park, S., Kim, S., Hunt, C., Park, D., DEVS peer-to-peer protocol for distributed and parallel simulation of hierarchical and decomposable DEVS models, 2007 International Symposium on Information Technology Convergence, 91–95, 2007.

[85] Franceschini, Romain, et al., A survey of modelling and simulation software frameworks using Discrete Event System Specification, 2014 Imperial College Computing Student Workshop. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Dagstuhl, 40-49, 2014.

[86] Wainer, G. A., Mosterman, P. J., Discrete-Event Modeling and Simulation: Theory and Applications, Taylor and Francis, CRC Press, London 120-200, 2010

[87] Haddadi, H., Rio, M., Iannaccone, G., Moore, A., & Mortier, R., Network topologies: inference, modeling, and generation. IEEE Communications Surveys & Tutorials, 10(2), 48-69, 2008

[88] Medina, Alberto, BRITE: An approach to universal topology generation, MASCOTS 2001, Proceedings Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems. IEEE, 346-353, 2001.

[89] Myllyla, J., Costin, A., Reducing the Time to Detect Cyber-attacks: Combining At-tack Simulation With Detection Logic, Proceedings of the 29th Conference of Open Inno-vations Association FRUCT, 465-474, 2021.

[90] McLaughlin, M., Sarjoughian, H., DEVS-scripting: a black-box test frame for DEVS models, 2020 Winter Simulation Conference, 2196–2207, 2020.

[91] CSE-CIC-IDS2018 on AWS https://www.unb.ca/cic/datasets/ids-2018.html, Erişim Tarihi: 08.08.2022.

[92] Understanding Denial-of-Service Attacks, https://www.cisa.gov/uscert/ncas /tips/ST04-015, Erişim Tarihi: 10.02.2022.

[93] Roy, S., & Khatwani, C., Cryptanalysis and improvement of ECC based authentication and key exchanging protocols. Cryptography, 1(1): 9-20, 2017

[94] Hasan, M. F., & Al-Ramadan, N. S., Cyber-attacks and Cyber Security Readiness: Iraqi Private Banks Case. Soc. Sci. Humanit. J, 5(8): 2312-2323, 2021.

[95] Lubna Fayez Eliyan, Roberto Di Pietro, DoS and DDoS attacks in Software Defined Networks: A survey of existing solutions and research challenges, Future Generation Computer Systems, 122(1), 149-171, 2021.

[96] Introduction: Denial of Service Attacks https://docs.aws.amazon.com /whitepapers/latest/aws-best-practices-ddos-resiliency/introduction-denial-of-service-attacks.html., Erişim Tarihi: 21.06.2022.

[97] Anley, C., Advanced SQL Injection In SQL Server Applications, Next Generation Security Software Publication, Surrey, 2002.

[98] Pantoulas, E., Description, analysis and implementation of a Web Application Firewall (WAF). Creation of attack scenarios and threats prevention, University of Piraeus, School of Information and Communication Technologies, Department of Digital Systems, Master's thesis, 2022.

ÖZGEÇMİŞ

Adı Soyadı **:** Şahin KARA

ÖĞRENİM DURUMU

Derece	Eğitim Birimi	Mezuniyet Yılı
Doktora	Sakarya Üniversitesi / Fen Bilimleri Enstitüsü / Bilgisayar ve Bilişim Mühendisliği	Devam ediyor
Yüksek Lisans	Fırat Üniversitesi / Fen Bilimleri Enstitüsü / Elektronik ve Bilgisayar Eğitimi	2013
Lisans	Süleyman Demirel Üniversitesi / Teknik Eğitim Fakültesi / Elektronik ve Bilgisayar Eğitimi	2002
Lise	Bitlis Anadolu Lisesi	1996

İŞ DENEYİMİ

Yıl	Yer	Görev
2002-2008	Milli Eğitim Bakanlığı	Öğretmen
2009-2011	Bitlis Eren Üniversitesi	Öğretim Görevlisi
2012-2018	Sakarya Üniversitesi	Öğretim Görevlisi
2019-Halen	Sakarya Uygulamalı Bilimler Üniversitesi	Öğretim Görevlisi

YABANCI DİL

İngilizce

ESERLER (makale, bildiri, proje vb.)

1. Daş, R., Kara, Ş., & Gündüz, M. Z., Casus Yazılımların Bilgisayar Sistemlerine Bulaşma Belirtileri ve Çözüm Önerileri, 5. In Uluslararası Bilgi Güvenliği ve

Kriptoloji Konferansı (5th International Conference on Information Security and Cryptology), ODTÜ, Ankara, 2012

2. Kara, S., Hizal, S., & Zengin, A., DESIGN AND IMPLEMENTATION OF A DEVS-BASED CYBER-ATTACK SIMULATOR FOR CYBER SECURITY. International Journal of Simulation Modelling (IJSIMM), 21(1), 2022.