T.C.

TRAKYA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

OPENFLOW PROTOKOLÜ KULLANAN KONTROL SUNUCULARI ARASINDA GÜVENLİK POLİTİKALARI PAYLAŞIM PROTOKOLÜ TASARIMI

GÖKHAN AKIN

DOKTORA TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Tez Danışmanı: Prof. Dr. Erdem UÇAR

EDİRNE-2021

Gökhan AKIN'ın hazırladığı "**Openflow Protokolü Kullanan Kontrol Sunucuları Arasında Güvenlik Politikaları Paylaşım Protokolü Tasarımı**" başlıklı bu tez, tarafımızca okunmuş, kapsam ve niteliği açısından Bilgisayar Mühendisliği Anabilim Dalında bir **Doktora tezi** olarak kabul edilmiştir.

Jüri Üyeleri	İmza
Prof. Dr. Erdem UÇAR	……
Prof. Dr. Tahir ALTINBALIK	……
Doç. Dr. İlhan UMUT	……
Doç. Dr. Erdinç UZUN	……
Dr. Öğr. Üyesi Hüseyin YÜCE	……

Tez Savunma Tarihi: 03/09/2021

Bu tezin Doktora tezi olarak gerekli şartları sağladığını onaylarım.

	İmza
Prof. Dr. Erdem UÇAR Tez Danışmanı	……
Dr. Öğr. Üyesi Enis KARAARSLAN İkinci Tez Danışmanı	……

Trakya Üniversitesi Fen Bilimleri Enstitüsü onayı

……

Prof. Dr. Hüseyin Rıza Ferhat KARABULUT
Fen Bilimleri Enstitüsü Müdürü

T.Ü.FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ DOKTORA PROGRAMI
DOĞRULUK BEYANI

Trakya Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında, tüm verilerin bilimsel ve akademik kurallar çerçevesinde elde edildiğini, kullanılan verilerde tahrifat yapılmadığını, tezin akademik ve etik kurallara uygun olarak yazıldığını, kullanılan tüm literatür bilgilerinin bilimsel normlara uygun bir şekilde kaynak gösterilerek ilgili tezde yer aldığını ve bu tezin tamamı ya da herhangi bir bölümünün daha önceden Trakya Üniversitesi ya da farklı bir üniversitede tez çalışması olarak sunulmadığını beyan ederim.

03 / 09 / 2021

Gökhan AKIN

Doktora Tezi

Openflow Protokolü Kullanan Kontrol Sunucuları Arasında Güvenlik Politikaları Paylaşım Protokolü Tasarımı

T.Ü. Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

ÖZET

Çevrimiçi verilen hizmetlerin gün geçtikçe kullanım oranının artması ile beraber, siber güvenlik tehditleri daha da kritik duruma gelmiştir. Yapılacak bir DDoS saldırısı sadece maddi zarara sebep olmaz, ülke çapında bazı hizmetlerin aksamasına da sebep olabilir. Mevcut çözümler, bu tip atakları hedef kurumda ya da ISP'de durdurmaktadır. Çalışma dahilinde; bu tip ulusal tehditleri kaynağında durdurmak amaçlı, Saldırı Bloklama Sunucu (SBS) yazılımı, Ulusal Siber Güvenlik Merkezi (USGM) sunucu yazılımı ve bu sunucular arasında atak bilgilendirmeleri için kullanılan bir protokol geliştirilmiştir. Sistem sadece saldırı engelleme üzerine oluşturulmuştur. Saldırı tespit verileri haricen sisteme bildirilmektedir. Oluşturulan sistem, Yazılım Tanımlı Ağları (YTA) destekleyen altyapılar için optimize edilmiştir. Saldırı bilgilerini paylaşmak için geliştirilmiş protokol, TCP 9090 ve 9191 portlarını kullanmaktadır. Sunucular, YTA kontrolcüsü ile REST API kullanarak, kontrolcü de anahtar cihazlar ile Openflow protokolünü kullanarak haberleşmektedir. Sistem; YTA destekli kurumlar arasında atak bilgisi doğrulaması, bloklama ve atak bittiğinde otomatik blok kaldırma süreçlerini gerçekleştirebilmektedir. YTA desteği olmayan ağlarda, sistem atak bilgilendirme amacıyla kullanılabilmektedir. Çalışmada, YTA desteği olmayan altyapılara eklenecek bir YTA cihazı ile atak bloklama özelliğinin kullanabileceği bir model de önerilmiştir.

Yıl : 2021

Sayfa Sayısı : 169

Anahtar Kelimeler : DDoS, Openflow, YTA, Flowspec

Doctoral Thesis

A Security Policy Sharing Protocol Design Between Control Servers Using Openflow Protocol

Trakya University Institute of Natural Sciences

Department of Computer Engineering

ABSTRACT

With the increasing use of online services, cyber security threats have become even more critical. A DDoS attack will not only cause financial burden, it may also cause disruption of some services across the country. Existing solutions stop such attacks at the target institution or ISP. Within the study; In order to stop such national threats at their source, Attack Blocking Server (SBS) software, National Cyber Security Center (USGM) server software and a protocol used for attack notifications between these servers have been developed. The system was created only for attack prevention. Intrusion detection data is reported to the system externally. The created system is optimized for infrastructures supporting Software Defined Networks (SDN). The protocol, developed to share attack information, uses TCP ports 9090 and 9191. The servers communicate with the SDN controller using REST API, and the controller communicates with the switch devices using the Openflow protocol. System can perform attack information verification, attack blocking and automatic block removal processes when the attack is over at the SDN supported institutions. In networks without SDN support, the system can only be used for attack notification purposes. In the study, a model that can use attack verification and attack blocking features is also proposed by adding an SDN supported device at non SDN infrastructures.

Year : 2021

Number of Pages : 169

Keywords : DDoS, Openflow, SDN, Flowspec

ÖNSÖZ

Doktora tez çalışma sürecimde bana her konuda destek olan danışmanım Prof. Dr. Erdem UÇAR’a çok teşekkür ederim.

Doktora başvuru mülakatı sırası da beklerken yanımda olandan, tezi tamamladığım son güne karar yanımda olan herkese teşekkürler. Desteğini hiç esirgemeyen ilk eşim Emine KÖKSOY'a teşekkür ederim.

Son olarak bu çalışma sürecinde, doktoraya başladığım ilk günden son gününe kadar umudunu hiç yitirmeyen babama ve anneme, sonsuz desteğini gördüğüm eşim Aydan NERGİZ AKIN'a, kızım Defne ve sabahları beni çok erken kaldırıp çalışmama destek olan oğlum Özgür Deniz’e de çok teşekkürler.

Gökhan AKIN

İÇİNDEKİLER

ÖZET....iv
ABSTRACT....v
ÖNSÖZ....vi
KISALTMALAR....x
ŞEKİLLER....xii
ÇİZELGELER....xvii
BÖLÜM 1: GİRİŞ....1
BÖLÜM 2: KULLANILAN TEKNOLOJİLER VE YAPILMIŞ ÇALIŞMALAR....7
2.1. Yazılım Tanımlı Ağlar (YTA)....7
2.1.1 İnternet Haberleşmesinde Mevcut Durum....7
2.1.2 Ağ Cihazlarının Gelişimi....8
2.1.3 Yazılım Tanımlı Ağların Çalışma Mekanizması....10
2.2. Openflow Protokolü....14
2.2.1 Openflow Protokolünün Tarihçesi....14
2.2.2 Openflow Protokolünün Yapısı....15
2.2.3 Openflow Mesajlaşma Tipleri....25
2.3 Yapılmış Çalışmalar, Endüstriyel Standartlar ve Karşılaştırma....28
BÖLÜM 3: USBS SİSTEMİ HEDEFLERİ VE SBP PROTOKOLÜNÜN YAPISI....36
3.1 Ulusal Saldırı Bloklama Sistemi (USBS) Hedefleri....36
3.2 Sistemin ve Protokolün Yapısı....38

3.3 Sistemin Operasyon Detayları 45
3.3.1 Saldırı Bloklama Sistemi 45
3.3.2 Ulusal Saldırı Bloklama Sistemi 47
BÖLÜM 4: USBS SİSTEMİNİN DİZAYN DETAYLARI 52
4.1. Saldırı Bloklama Sistemi (SBS) Detayları 52
4.1.1 SBS Sunucusunun Yapılandırılması 58
4.1.2 SBS Sistemine Atak Bilgisi Girme Süreci 59
4.1.3 SBS Atak Bilgilendirme Süreci 61
4.1.4 SBS Atak Bloklama Süreci 68
4.1.4.1 Saldırı Kaynak IP Adresinin Hangi Anahtarda Olduğu Tespiti 68
4.1.4.2 Bloklanması İstenen Saldırı Hakkında Bilgi Amaçlı Akış Eklenmesi 69
4.1.4.3 Trafik İstatistiğin Oluşması İçin Belirli Süre Bekleme 74
4.1.4.4 İlgili Anahtardan Trafik İstatistiklerinin Çekilmesi 75
4.1.4.5 İlgili Atak Trafiğinin Bloklanması 76
4.1.4.6 Bloklanan Trafik Bilgisinin Akış Veri Tabanına İşlenmesi 80
4.1.5 SBS Atak Devam Kontrolü 80
4.2 Ulusal Siber Güvenlik Merkezi (USGM) Sunucusu Detayları 85
4.2.1 USGM Sunucusunun Yapılandırılması 87
4.2.2 Ulusal Saldırı Bloklama ve Blok Kaldırma Süreci 87
4.2.2.1 SBS Sunucusunda Ulusal Atak Bloklama Süreci 93
4.2.2.2 SBS Sunucusunda Ulusal Atak Blok Kaldırma Süreci 94
BÖLÜM 5: UYGULAMA TEST ORTAMI VE ULAŞILAN HEDEFLER 97
5.1 Test Ortamını Oluşturmakta Kullanılan Araçlar 97
5.1.1 YTA Destekli Ağ Ortamı 97
5.1.1.1 Mininet Ağ Ortamının Kurulması 98

5.1.1.2 Mininet Komutları 99
5.1.1.3 Mininet Grafik Ortamı 101
5.1.1.4 Mininet'de Haberleşme Testlerinin Yapılması 103
5.1.2 YTA Kontrolcü 105
5.2 Test Ortamın Detayları 107
5.3 USGM Sistemi Üzerinde Yapılan Testler 112
5.3.1 Harici Bir SBS'den Atak Bilgisi Gelmesi (Test 1) 113
5.3.2 Atak Durduğunda Atak Bloklanmasının Otomatik Kaldırılması (Test 2) 115
5.3.3 Atağın Yerel Bir IP Adresinden Gelmesi (Test 3) 117
5.3.4 Belirtilen Atak Trafiğin Olmaması (Test 4) 118
5.3.5 Atak Kaynak IP Adresinin Kullanılmıyor Olması (Test 5) 120
5.3.6 Kendisine Ait Olmayan Bir IP Adresi için Atak Bilgilendirmesi (Test 6) 122
5.3.7 YTA Desteği Olmayan Şubeye Atak Bilgilendirmesi Gelmesi (Test 7) 124
5.3.8 YTA Desteği Olmayan Şubeden Atak Bilgilendirmesi Yapılması (Test 8) 126
5.3.9 USGM Sunucusuna Erişilememe Durumu (Test 9) 128
5.3.10 USGM Sunucusundan Ulusal Atak Bloklama Emrinin Gelmesi (Test 10) 129
5.3.11 USGM Sunucusunda Atak Blok Kaldırma Emrinin Gelmesi (Test 11) 131
5.4 Ulaşılan Hedefler 132
BÖLÜM 6: SONUÇLAR VE TARTIŞMA. 135
KAYNAKLAR 138
ÖZGEÇMİŞ 151
TEZ ÖĞRENCİSİNE AİT TEZ İLE İLGİLİ BİLİMSEL FAALİYETLER 152

KISALTMALAR DİZİNİ

ABS	Attack Blocking System
API	Application Programming Interface,
ARP	Address Resolution Protocol
ASIC	Application Specific Integrated Circuit
BGP	Border Gateway Protocol
CPU	Central Processing Unit
D/RTBH	Destination-based Remotely Triggered Black Hole
DDoS	Distrubuted Denial of Service Attack
DoS	Denial of Service Attack
FPGA	Field Programmable Gate Array
http	Hyper-Text Transfer Protocol
ICMP	Internet Control Message Protocol
IoT	Internet of Things
IP	Internet Protocol
IPS	Intrusion Prevention System
ISP	Internet Service Provider
MAC	Media access control
MTD	Moving Target Defense
NAT	Network Address Translation
NFC	Near Field Communication

ONF	Open Network Foundation
RFC	Request For Comments
S/RTBH	Source-based Remotely Triggered Black Hole
SBP	Saldırı Bloklama Protokolü
SBS	Saldırı Bloklama Sistemi
SDN	Software Defined Network
SNAP	Sub-network Access Protocol
SNMP	Simple Network Management Protocol
SSH	Secure Shell
STP	Spanning Tree Protokolü
STS	Saldırı Tespit Sistemi
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
ULAKNET	Ulusal Akademik Network
URL	Uniform Resource Locator
USBS	Ulusal Saldırı Bloklama Sistemi
VLAN	Virtual Local Area Network
YTA	Yazılım Tanımlı Ağ

ŞEKİLLER DİZİNİ

Şekil 1.1 Günlük yetkisiz SSH girişimi (2017) 3
Şekil 1.2 Ataklarda kullanılan kullanıcı adları dağılımı 4
Şekil 1.3 Günlük yetkisiz SSH girişimi (2019) 4
Şekil 2.1 Geleneksel yönlendirici 9
Şekil 2.2 İşletim sistemi modeli 11
Şekil 2.3 YTA Modeli 12
Şekil 2.4 YTA kuzey ve güney köprüsü 13
Şekil 2.5 Geleneksel ağ anahtar yapısı 16
Şekil 2.6 Openflow destekli ağ anahtar yapısı 17
Şekil 2.7 YTA anahtar çalışma süreci 22
Şekil 2.8 Çoklu akış tablosu 25
Şekil 2.9 Örnek YTA haberleşme süreci 27
Şekil 3.1 Merkezi mimari 39
Şekil 3.2 Dağıtık mimari 40
Şekil 3.3 ABS sistemin çalışma yapısı (Akin vd., 2020) 41
Şekil 3.4 Sistemin kötü amaçlı kullanımı 41
Şekil 3.5 Komşu ABS'nin tanımı ve ağ bilgilerinin kontrolü 42
Şekil 3.6 Saldırı bloklama protokolü talep paket yapısı 44
Şekil 3.7 Saldırı bloklama protokolü cevap paket yapısı 44
Şekil 3.8 Saldırı bloklama sistemi mimarisi 49

Şekil 3.9 a)YTA desteği olmayan ağlarda kullanım b) YTA desteği olmayan ağlarda tavsiye edilen kullanım 50

Şekil 3.10 Ulusal Saldırı Bloklama Sistemi mimarisi 51

Şekil 4.1 SBS sunucusu iş parçacıkları çalışma süreci 53

Şekil 4.2 SBS sunucusu ana yapılandırma dosyası 59

Şekil 4.3 SBS sunucusu yönetilen ağlar dosyası 59

Şekil 4.4 SBS sunucusu ana ekranı 60

Şekil 4.5 Atak bilgisi giriş ekranı 61

Şekil 4.6 USGM saldırı bloklama süreci 62

Şekil 4.7 SBS sunucusun atak kaynak IP adresini öğrenme süreci 63

Şekil 4.8 SBS sunucusun atak kaynak IP'nin kontrolünü sağlayan SBS sunucusuna yaptığı bloklama talebi süreci 64

Şekil 4.9 Saldırı bloklama sistemi sunucusu bloklama talep algoritması 65

Şekil 4.10 Saldırı bloklama sistemi sunucusu bloklama talebine cevap algoritması 66

Şekil 4.11 USGM sunucusunun SBS sunucusunu bilgilendirme algoritması 67

Şekil 4.12 YTA modülü algoritması 83

Şekil 4.13 YTA atak devam kontrol algoritması 84

Şekil 4.14 USGM çalışma süreci 85

Şekil 4.15 USGM sunucusu yönetilen ağlar dosyası 87

Şekil 4.16 USGM sunucusu ana ekranı 88

Şekil 4.17 USGM sunucusu atak bilgisi giriş ekranı 89

Şekil 4.18 USGM sunucusu ulusal bloklama ve blok kaldırma paketleri 90

Şekil 4.19 USGM sunucusu ulusal bloklama ve blok kaldırma süreci 91

Şekil 4.20 USGM sunucusu ulusal bloklama ve blok kaldırma talep algoritması 92

Şekil 4.21 USGM sunucusu ulusal bloklama ve blok kaldırma cevap algoritması 96

Şekil 5.1 Mininet ortamı 98
Şekil 5.2 Mininet'in başlatılması 99
Şekil 5.3 Mininet hazır topoloji şablonları 100
Şekil 5.4 Konsol yazılımında X11 özelliğinin aktive edilmesi 101
Şekil 5.5 Mininet istemcilerine doğrudan erişim 102
Şekil 5.6 Wireshark GUI arayüzüne erişim 102
Şekil 5.7 İstemciler arasında Ping uygulaması 103
Şekil 5.8 Web uygulamasının aktive edilmesi ve testi 104
Şekil 5.9 Iperf test aracının kullanımı 104
Şekil 5.10 Opendaylight'ın çalıştırılması 106
Şekil 5.11 Opendaylight arayüzü 106
Şekil 5.12 Rest API ile sorgunun yapılması 107
Şekil 5.13 Test ortamı 108
Şekil 5.14 Aktive edilen sistemde Şube 1 ile Şube 2'nin erişim testi 112
Şekil 5.15 İstemci 11_1'in ping trafiğini başlatması 113
Şekil 5.16 SBS2'den İstemci 11_1'in ping trafiğinin bloklanma talebi 113
Şekil 5.17 USGM sunucusunun sorgulara cevap vermesi 114
Şekil 5.18 SBS1'de gelen atak uyarısı üzerine yapılan bloklama işlemleri 115
Şekil 5.19 İstemci 11_1'de ping trafiğinin durdurulması 116
Şekil 5.20 SBS1 sunucusunun daha önce eklediği bloklamayı otomatik kaldırması ... 116
Şekil 5.21 İstemci 11_1'den aynı ağdaki istemci 11_2'ye doğru ping trafiği 117
Şekil 5.22 SBS1 sunucusunun atağın yerel olduğunu tespiti ve atağı bloklanması 117
Şekil 5.23 SBS1 sunucusunun daha önce eklediği bloklamayı otomatik kaldırması ... 118
Şekil 5.24 SBS2 sunucusuna olmayan bir atak bilgisi girilmesi 119
Şekil 5.25 Atak kaynak SBS sunucuları için gereken sorguların USGM'ye yapılması 120

Şekil 5.26 SBS1 sunucunun belirtilen atağın gerçekleşmediğinin tespiti 120
Şekil 5.27 SBS2 sunucusunda aktif olmayan bir IP adresi için atak bilgisi girilmesi .. 121
Şekil 5.28 Atak kaynak SBS sunucuları için gereken sorguların USGM'ye yapılması 121
Şekil 5.29 SBS1'de atak kaynak IP adresinin kullanılmadığı tespiti 122
Şekil 5.30 SBS2 sunucusunda kendisine ait olmayan bir IP adresi için atak bilgisi girilmesi 123
Şekil 5.31 Atak kaynak SBS sunucuları için gereken sorguların USGM sunucusuna yapılması 123
Şekil 5.32 SBS1'de atak hedef IP adresinin SBS2'ye ait olmadığı tespiti 124
Şekil 5.33 SBS2'den YTA desteği olmayan bir şubeye atak uyarısı yapması 125
Şekil 5.34 Atak kaynak SBS sunucuları için gereken sorguların USGM sunucusuna yapılması 125
Şekil 5.35 YTA desteği olmayan SBS3 sunucunun atak bilgilendirmesini yapması ... 126
Şekil 5.36 İstemci 11_1'den YTA desteği olmayan bir ağda olan istemci 13_1'e başlatılan ping trafiği 126
Şekil 5.37 YTA desteği olmayan SBS3'den atak bilgilendirmesi yapılması 127
Şekil 5.38 Atak kaynak SBS sunucuları için gereken sorguların USGM sunucusuna yapılması 128
Şekil 5.39 SBS1'de gelen atak uyarısı üzerine yapılan bloklama işlemleri 128
Şekil 5.40 SBS2'in USGM sunucusuna erişememe durumu 129
Şekil 5.41 USGM sunucusundan ulusal atak bloklama talebi yapılması 130
Şekil 5.42 SBS1'de USGM sunucusundan gelen bloklama talebinin uygulanması 130
Şekil 5.43 YTA desteği olmayan SBS3'de USGM sunucusundan gelen bloklama talebi ile ilgili bilgilendirme 131
Şekil 5.44 USGM sunucusundan ulusal atak blok kaldırma talebi yapılması 131

Şekil 5.45 SBS1'de USGM sunucusundan gelen blok kaldırma talebinin uygulanması 132

Şekil 5.46 YTA desteği olmayan SBS3'de USGM sunucusundan gelen blok kaldırma talebi ile ilgili bilgilendirme.....132

ÇİZELGELER DİZİNİ

Çizelge 2.1 Openflow v1.0'da eşleşen kısımlar 18
Çizelge 2.2 Openflow v1.0'da tutulan istatistik değerleri 19
Çizelge 2.3 Örnek akış tablosu 24
Çizelge 2.4 Karşılaştırma tablosu 33
Çizelge 3.1 Emir kodları tablosu 44
Çizelge 3.2 Cevap kodları tablosu 45
Çizelge 4.1 Kullanılan hazır modüller 53
Çizelge 4.2 Oluşturulan modüller 54
Çizelge 4.3 Akış durumları 55
Çizelge 4.4 Yaygın kullanılan protokol numaraları 56
Çizelge 4.5 Yaygın kullanılan ICMP tip ve kod numaraları 57
Çizelge 4.6 Yaygın kullanılan protokoller ve port numaraları 57
Çizelge 4.7 Kullanılan hazır modüller 86
Çizelge 4.8 Oluşturulan modüller 86
Çizelge 5.1 Test ortamı detayları 108
Çizelge 5.2 Şube 1'i oluşturmak için kullanılan betik (script11.py) 109
Çizelge 5.3 Şube 2'nin oluşturulması için kullanılan betik (script12.py) 110
Çizelge 5.4 Şube 3'ün oluşturulması için kullanılan betik (script13.py) 110

BÖLÜM 1

GİRİŞ

İnternet üzerinden sağlanan kamusal ve özel sektör hizmetlerinin artması ile birlikte, İnternet erişiminin sürekliliği daha da kritik hale gelmiştir. İnternet erişiminde oluşan kesintilerin sonuçları sadece yüksek maddi kayıplara yol açmakla kalmamakta, gününüzde sağlık, güvenlik, adalet vb. tüm kamusal hizmetlerin kesintiye uğramasına da sebep olmaktadır. Özellikle mobil kullanıcılar ile daha da hızlı artan İnternet kullanıcısı sayısı, hem yönetimi zorlaştırmakta hem de siber saldırıların artması ile güvenliğin sağlanmasını da zorlaştırmaktadır (Skowyra, Bahargam & Bestavros, 2013; Zhang, Shou, Hu & Guo, 2013).

Ayrıca "Internet of Things" (IoT) kavramı ile İnternet üzerinden sağlanan hizmetler farklı bir boyuta taşınmaktadır. Akıllı şehirler, akıllı evler vb. gibi kavramlarla, her cihaza İnternet üzerinden erişilmesi ve yönetilmesi söz konusu olmaktadır. Bu hem cihaz sayısını arttırmakta hem de tehdidi daha da büyütmektedir (Zargar, Joshi & David, 2013).

Günümüzde siber güvenlik tehditleri hem ulusal hem de global bir güvenlik sorunudur. Son 4 yıldır Dünya Ekonomi Forum'u raporlarında siber güvenlik, dünyayı tehdit eden riskler listesinde ilk 10'da yer almaktadır. 2021 global riskler raporunda, yakın zamandaki en büyük riskler sıralamasında dördüncü sırada siber güvenlik başlığı geçmektedir (World Economic Forum, 2021). İnternet altyapısı için en kritik tehditlerden biri olarak "Distributed Denial of Service" (DDoS), dağıtık servis reddi saldırılarını sayabiliriz (Cui vd. , 2016; Mirkovic & Reiher, 2004). Bu ataklar ile herhangi bir önceden tespit veya uyarı olmadan ağ ve bilgisayar kaynakları tüketilebilmektedir (Douligeris &

Mitrokotsa, 2004). Bu da, verilen hizmetin durmasına sebep olmaktadır. Yüksek hacimli DDoS saldırılarının, hedef ağın sınırında durdurulması, sunucuya atağın gelmesini engelleyebilir ama ağın İnternet erişimini durduracağı için, hedef sistemin servis dışı kalmasını engelleyemez. İnternet erişim bant genişliğinin doldurulması ile atak hedefine ulaşır ve servis ulaşılamaz olur.

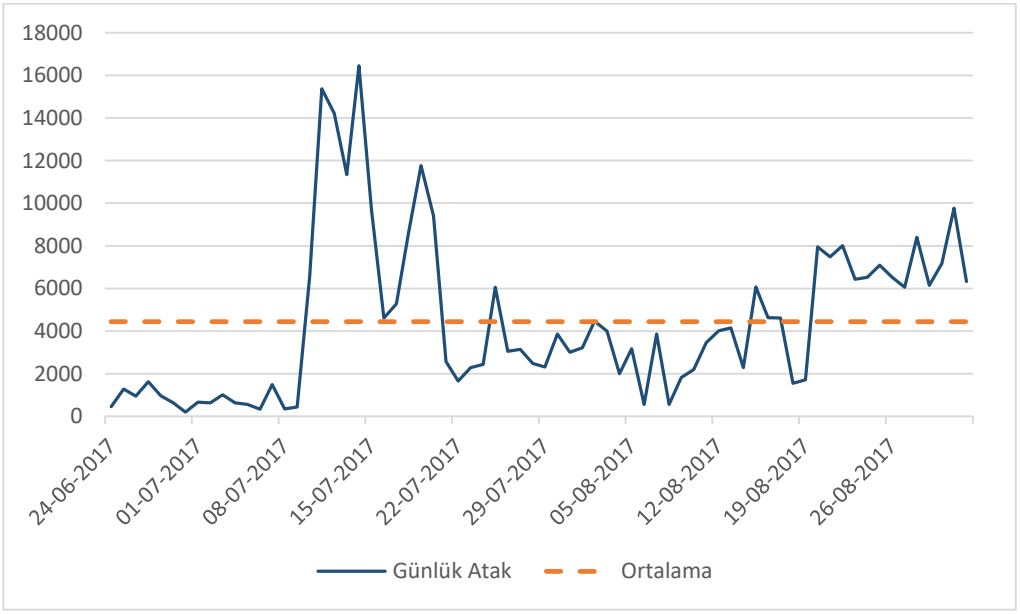
İnternet Servis Sağlayıcıları (Internet Service Provider, ISP) genellikle özel firmaların çözümleri ile atağı bloklamayı tercih etmektedirler (Arbor Networks, 2012). Bu sistemler, hedef ağın ISP erişiminin dolmasını engeller ancak bağlı olunan ISP'ye kadar olan hattın dolmasını engelleyemez.

Şubat 2016'da Türkiye'nin DNS sunucularına bu tip bir atak gerçekleştirildi. Bu sunucular ULAKNET'den (Ulusal Akademik Ağ, Türkiye'deki üniversitelerin ISP'si) üzerinden hizmet almaktaydı. Atak ULAKNET tarafından bloklandı. Ancak, ISP'nin İnternet erişiminin dolmasından dolayı ".tr" uzantılı siteler servis dışı kaldı. Bu da, sadece atağın hedef aldığı kuruma hizmet veren ISP tarafından alınan bu tip çözümlerin yetersiz kalabileceğini göstermektedir.

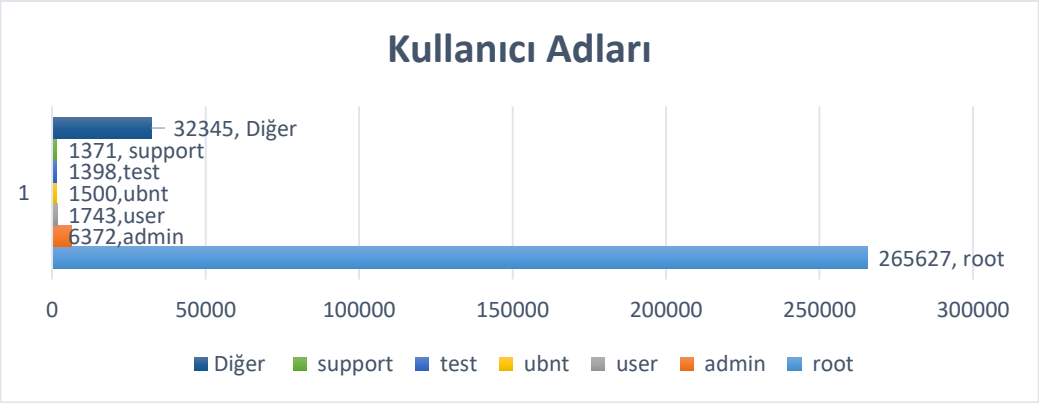
Bu tip ataklara endüstri standardı olan engelleme çözümleri ise BGP (Border Gateway Protocol) protokolünü kullanmaktadırlar (RFC3882, RFC5575, RFC5635). BGP protokolü ile hedef ağ, hizmet aldığı ISP'sine bu atağı engellemesi talebinde bulunur. Bu talep, BGP protokolü için manuel olarak bir yapılandırma ile gerçekleştirilir. Bu yapılan konfigürasyon, atak bittikten sonra yine manuel olarak kaldırılır. Ancak birçok kurumsal şirket BGP protokolünü kullanmamaktadır. Bu da, bu çözümlerin tüm ağ altyapılarında kullanılmasını engellemektedir. Bu tip durumlarda, ISP ile telefon veya eposta yolu ile iletişime geçilip, DDoS atağının engellenmesi için talepte bulunulur. Bu da atağa karşı alınacak müdahale sürecini uzatmaktadır. Hem ISP'nin hem de müşterinin BGP kullandığı durumlarda ise, atağın gerçekten olup olmadığı tespit edilmeden, müşterinin talebine güvenmek durumunda kalınmaktadır.

Gerek özel DDoS çözümleri, gerekse endüstri standardı DDoS engelleme çözümleri, atak kaynağı sistemleri yaptıkları atak ile ilgili otomatik bir uyarı sistemine sahip değildirler. Atak zararlı yazılım bulaşmış sistemler üzerinden yapılıyorsa, bu sistemler diğer ağlara saldırmaya devam edebilirler.

Ayrıca, şu ana kadar bahsedilen çözümler genel olarak DDoS ataklarına karşıdır. Bu tip sistemler, düşük hacimli ve servis dışı bırakmak amaçlı olmayan ataklara karşı çözüm sunmazlar. Çalışma dahilinde, bu tip tekil ataklara örnek olan SSH (Secure Shell) servisine yapılacak bir parola atağını inceledik. İlk olarak, 24 Haziran 2017 ile 1 Eylül 2017 (70 Gün) tarihleri arasında yerel bir ISP'den izin alarak, gerçek IP adresine sahip bir cihazı, SSH servisi tüm İnternet üzerinden erişilebilecek şekilde bağladık. Bu süre zarfında, yetkisiz SSH erişim ataklarını kayıt altına aldık. Toplamda 310.356 erişim atağı, günlük ortalamada 4.434 atak gerçekleşti (Şekil 1.1). Bir günde gelen en yüksek atak teşebbüsü 16.542 adet olarak gerçekleşti. Bu çalışmada, ataklarda kullanılan en çok kullanıcı adı verisini de analiz etme şansımız da oldu. Buna göre "root" kullanıcı adı, 265.627 kere ile en çok denenen kullanıcı adı oldu (Şekil 1.2).

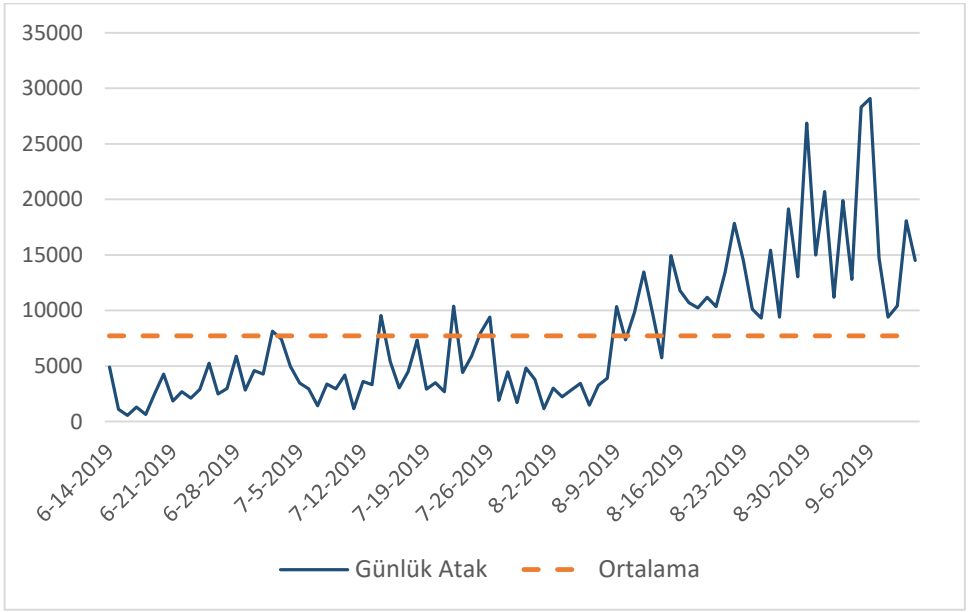


Şekil 1.1 Günlük yetkisiz SSH girişimi (2017).



Şekil 1.2 Ataklarda kullanılan kullanıcı adları dağılımı.

Aynı çalışmayı, 14 Haziran 2019 - 11 Eylül 2019 tarihleri arasında toplam 90 gün olacak şekilde bu kez bir üniversitede gerçekleştirdik (Akin, Buk & Ucar, 2020). Toplamda 693.597, günlük ortalamada 7.707 yetkisiz SSH erişimi gerçekleştirildi (Şekil 1.3). Bir günde gerçekleşen en yüksek yetkisiz erişim sayısı 29.081 oldu. Sadece fikir vermesi amacı ile gerçekleştirilen bu çalışma ile seneler içerisinde bu atak tipi için gerçekleşen teşebbüslerin arttığını ve küçümsenmeyecek bir boyutta olduğunu gördük.



Şekil 1.3 Günlük yetkisiz SSH girişimi (2019).

Saldırı engelleme sistemleri (Intrusion Prevention System, IPS) ile bu tip tekil ataklar, hedef ağ yönetimi tarafında engellenebilmektedir. Saldırı hedef ağa kadar ulaşır ve burada bloklanır. Ancak saldırı kaynağı olan ağın yöneticileri, bloklanma sonrasında bilgilendirilmez. Saldırı kaynağı zararlı yazılım bulaşmış bir bilgisayar ise faaliyetlerine devam edecektir.

Bu çalışmada; gerek DDoS ataklarına, gerekse tekil ataklara karşı kullanılabilecek Ulusal Saldırı Bloklama Sistemi (USBS) oluşturulmuştur. Bu sistemle amaçlanan, saldırıların kaynağında bloklanması ve atak etkilerinin minimum düzeye indirilmesidir. Sistem dahilinde oluşturulmuş Saldırı Bloklama Sistemi sunucuları (SBS'ler) birbirleri arasında kendilerine gelen saldırı bilgilerini, Saldırı Bloklama Protokolü (SBP) ile paylaşmaktadırlar. Bu protokol ile hedeflenen, bu sistemi kurmuş ağ yöneticileri arasında bir atak bilgilendirme sistemi oluşturmaktır. Bu sistemi kullanmak için BGP protokolünü kullanmaya gerek yoktur ve Software Defined Network (SDN) olarak da bilinen Yazılım Tanımlı Ağ (YTA) mimarisi ile beraber kullanılması durumunda otomatik saldırı bloklama, saldırı kontrolü ve saldırı bitince bloklamanın kaldırılmasını sağlamaktadır.

Yönlendiriciler günümüzde çoğunlukla dağıtık karar veren cihazlar olarak çalışır ve merkezi bir karar verme mekanizmasına sahip değildirler. Ağ altyapısını merkezileştirmek için geliştirilen Openflow protokolü ile çalışan sistemlerde, mevcut ağ cihazlarının sahip olduğu kontrol katmanı (ağ uygulaması) ve veri katmanı (anahtarlama uygulaması) mimarileri farklı şekilde değerlendirmektedir (Mckeown vd., 2008). Kontrol katmanı işlemleri genel amaçlı bir CPU (Central Processing Unit) ile gerçekleştirilirken, veri katmanı paket anahtarlama işlemlerini ASIC (Application Specific Integrated Circuit) ile gerçekleştirilir. Yazılım Tanımlı Ağ dizaynında, kontrol katmanı operasyonunu, "YTA Kontrolcüsü" olarak adlandırılan harici bir bilgisayar gerçekleştirir. YTA kontrolcü yazılımı, birden çok ağ cihazına hizmet verebilir. Bu sistemde uygulama katmanı amaçlı geliştirilen yazılımlar ile YTA kontrolcüsü merkezi bir noktadan yönetilebilir (Farhady, Lee & Nakao, 2015). YTA kullanımı ile ağların yönetimi ve izlenmesi çok daha kolay hale gelmiştir (Ding, Jon, Sasu & Flinck, 2014; Sezer vd., 2013; Vissicchio, Vanbever & Bonaventure, 2014). YTA ayrıca birçok gelişmiş güvenlik

çözümünün de kolayca uygulanabilmesini sağlamıştır (Yoon vd., 2015). YTA ve Openflow protokolünün detayları, ikinci bölümde daha detaylı olarak işlenmektedir.

YTA mimarisini kullanan USBS sistemi, saldırıları kaynağında bloklayabilmektedir. Ancak YTA desteği olmayan ağ altyapılarında, sisteme eklenecek tek bir YTA anahtar ile otomatik bloklama özelliği devreye alınabilir. Bu işlemin detayları üçüncü bölümde tarif edilmiştir. Böyle bir ekleme yapılmaması durumunda ise USBS sistemi sadece atak bilgilendirme sistemi olarak da kullanılabilir.

BÖLÜM 2

KULLANILAN TEKNOLOJİLER VE YAPILMIŞ ÇALIŞMALAR

2.1. Yazılım Tanımlı Ağlar (YTA)

2.1.1 İnternet Haberleşmesinde Mevcut Durum

Günümüzde kullanılan ağ işletim sistemleri ve kullanılan protokoller 1970'li yıllarda belirlenmiştir. O yıllarda İnternet Protokolü (IP) için yaklaşık 16 milyon IP adresine sahip olan 32 bitlik adres yapısının yeterli olduğu düşünülmüştür. Halbuki günümüzde İnternet'te kullanılan cihaz sayısına göre bu adres aralığı çoktan yetersiz kalmıştır (Morreale & Anderson, 2015). İnternet kullanımının artmasının yanı sıra erişim hızlarında sürekli artmaya devam etmektedir. IEEE 802.3bs standardı ile beraber kablolu olarak 400Gbit, kablosuz yerel alan ağı olarak da 9.6Gbit (802.11ax) erişim hızlarına ulaşmıştır. Bununla birlikte yaygın olarak kabul edilen gözlem, İnternet altyapısı ile ilgili gelen yeni teknolojilerin ne kadar iyi olursa olsun mevcut yapıya uygulanmasının uzun zaman almasıdır. Bunun en iyi örneklerinden bir tanesi IPv6 protokolüne geçiş sürecidir. IPv6 protokolüne, 2000'li yılların başında standart olmasına rağmen, halen daha tüm dünya çapında geçilebilmiş değildir. Ağ alt yapıları üzerine yeni teknolojiler geliştirilmesinin ve uygulamaya geçilmesinin önündeki en büyük engellerden bir tanesi, yönlendirici cihaz yazılımlarının donanıma gömülü olması gösterilmektedir (McCauley, Harchol, Pand, Raghavan & Shenker, 2019).

Günümüz ağ altyapılarında haberleşmenin kaynaktan hedefe ulaştırılması için birçok ağ cihazı kullanılmaktadır. Yönlendiriciler, anahtarlama cihazları, güvenlik duvarları vb. cihazları örnek olarak gösterebiliriz. Bu cihazlardan yönlendiriciler, İnternet

altyapısının ana yapı taşıdır ve ağlar arasında haberleşmeyi sağlar. Yönlendiriciler halen şu 3 prensip çerçevesine çalışmaktadırlar.

1. Her yönlendirici kendi kararını sahip olduğu yönlendirme tablosu çerçevesinde, kendisi verir.
2. Bir yönlendiricinin oluşturduğu yönlendirme tablosu, tüm yönlendiricilerin aynı tabloyu oluşturacağını göstermez.
3. Bir ağdan diğer ağa doğru belirlen bir rota bilgisi, geri dönüş trafiğinin de aynı rotadan geri döneceğini göstermez (Zinin, 2002).

Yaygın kullanılan yönlendiriciler, kapalı kaynak kodlu bir yazılımdan ve özel yapılandırılmış bir donanımdan oluşmaktadır. Bu cihazlar, komut satırı veya grafik arayüzleri ile tek tek yapılandırılmakta ve yönetilmektedirler. Artan kullanıcı sayısı paralelinde bu ağ cihazlarının sayıları da hızla artmaktadır. Alex Zinin'in prensiplerinden de anlaşılacağı gibi, dağıtık şekilde çalışan ağ cihazlarının tek tek yönetilmesi ve problemlerinin giderilmesi gerekmektedir. Ağ cihazlarında ki trafik yönlendirme tabloları, güvenlik önlemleri ve servis kalitesi ayarları gibi işlemler sebebi ile ağ yöneticileri üzerindeki yük gün geçtikçe artmaktadır. Bu da hatalı yapılandırma riskini artırmaktadır (Azodolmolky, 2013).

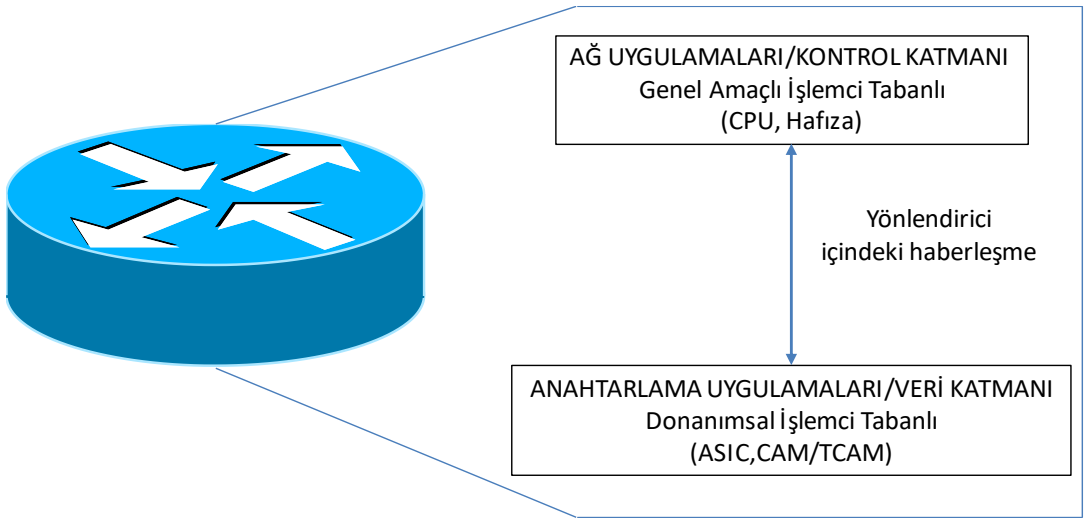
2.1.2 Ağ Cihazlarının Gelişimi

Ağ teknolojilerinin gelişim sürecinde yönlendirme işlemleri, başlangıçta genel amaçlı işlemciye sahip bilgisayarlar tarafından gerçekleştirilmiştir. İlk yönlendiricilerden biri, Honeywell DDP-516 marka bir mini bilgisayar üzerinde çalıştırılmıştır (Heart, Kahn, Ornstein, Crowther & Walden, 1970). Bu cihazlar, üzerlerindeki statik yönlendirme tablolarına göre genel amaçlı işlemci gücünü kullanarak, yazılım tanımlı anahtarlama yapacak şekilde yapılandırılmıştır.

Erişim hızlarının artması ile artan paket anahtarlama ihtiyacının yazılımsal çözümler ile sağlamaya devam etmek, pahalı işlemcilerin kullanılmasını gerektirmeye başlamıştır. Maliyetin yanı sıra enerji tüketimi ve soğutma ihtiyacı da artmış, bu da tasarım zorluklarını beraberinde getirmiştir.

Çözüm olarak, yüksek paket anahtarlama amaçlı donanım tabanlı mimariler geliştirilmiştir (Casado, Koponen, Monn & Shenker, 2008). Donanım tabanlı mimariler yüksek paket anahtarlama kapasitesini çok daha ucuz maliyetlere sağlayabilmektedir. Ancak zaman içinde ortaya çıkan yeni ihtiyaçlara göre bu donanımlara özellik eklenmesi, yeni bir donanım tasarlanması ve üretilmesi ile mümkün olmaktadır. Bu da zaman alan ve maliyetli bir süreçtir.

Günümüz yönlendirme cihazlarının mimarisi kontrol katmanı ve veri katmanından oluşmaktadır (Nadeau & Gray, 2013). Kontrol katmanı cihazların yazılımsal olarak yönettiği yönlendirme protokolleri ve ARP (Address Resolution Protocol) mekanizması gibi işlemleri üstlenir. Veri katmanı ise, kontrol katmanı süreçleri sonrasında belirlenmiş hedef arayüze paketlerin donanımsal olarak anahtarlanmasını sağlar (Longo, Distefano, Bruneo & Scarpa, 2015). Kontrol katmanı ile veri katmanı arasındaki haberleşme ise, cihazın içerisinde yapılmakta ve firmaların kendilerine özel geliştirdikleri protokoller ile gerçekleşmektedir (Şekil 2.1).



Şekil 2.1 Geleneksel yönlendirici.

Ancak hızla artan kullanıcı sayıları, erişim hızları ve yeni ihtiyaçların çıkması mevcut cihazların kontrol katmanının yetersiz kalmasına sebep olmaktadır. Veri katmanı halen daha yeterli olsa bile sadece kontrol katmanı yetersiz olduğu için çözüm ancak

cihazın komple değiştirilmesi ile mümkün olmaktadır. Bu da periyodik olarak ciddi maliyetlere sebep olmaktadır.

Büyük ağların temel ihtiyaçlarını trafik mühendisliği, güvenlik, servis kalitesi, yönlendirme, anahtarlama, sanallaştırma, monitör etme ve yük paylaşımı olarak sıralayabiliriz. Şekil 2.1'de gösterilen cihazlardan oluşturulan geleneksel ağ altyapılarında, her cihaz bir paketin ulaştırılacağı portu kendisi belirlemektedir (Zinin, 2002). Güvenlik ve diğer tüm ihtiyaçlar da her cihazda tek tek uygulanmaktadır. Bu şekilde yönetilen ağlarda, doğası gereği merkezi bir kontrol mekanizması bulunmamaktadır, yani ağ sistemleri dağıtık olarak yönetilmektedir. SNMP (Simple Network Management Protocol) ve benzeri protokoller ile merkezi bir monitör sistemi oluşturulması söz konusu olabilmektedir (Case, Rose, McCloghrie, Waldbusser & Preshuhn, 2002). Ancak bu tip uygulamalarla bile ağdaki trafiğin görünürlüğü kısıtlı kalmaktadır.

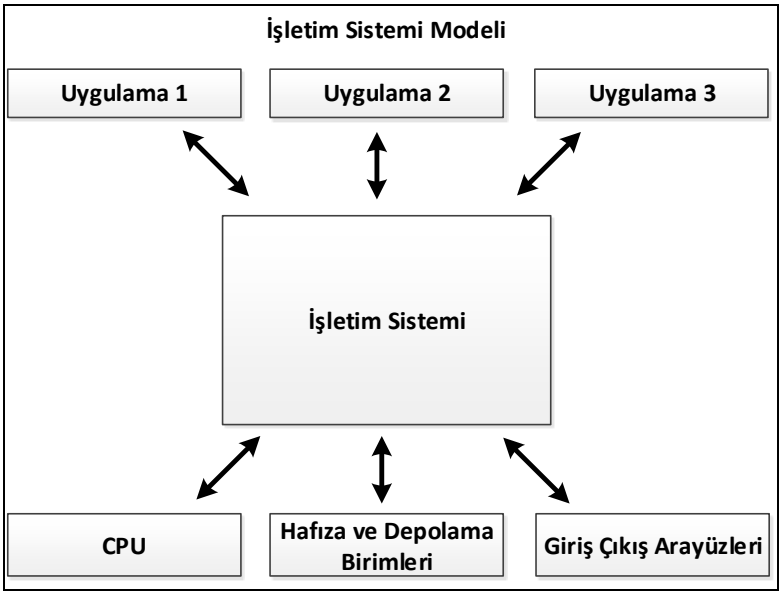
Bunun yanı sıra YTA öncesinde ağ cihazları üzerine yapılan araştırma çalışmaları mevcut bilgisayarlar ile yapılan yazılımlar ile gerçekleştirilebilmekteydi. Ancak bu tür çalışmalarda, yazılımsal anahtarlamanın sorunu olan yüksek bant genişliği ve yüksek paket anahtarlama ihtiyacı üzerine yoğunlaşılamamaktadır. Buna çözüm olarak geliştirilmiş programlanabilir Net-FPGA kartları, donanımsal anahtarlama üzerine araştırma yapılabilmesinin önünü açmıştır ama bu kartlar port sayısı ve hızı olarak gerçek bir ağ ortamını emüle etmek için yetersiz kalmaktadır. (Mckeown vd., 2008).

Mevcut yüksek port sayısına ve anahtarlama kapasitesine sahip donanımsal çalışan anahtar cihazların üzerinde çalışan yazılımlar, firmaların kendi geliştirdikleri kapalı platformlardır. Bu sebepten araştırmacılar, bu yüksek port sayısına ve anahtarlama kapasitesine sahip cihazlar üzerinde yazılım geliştirememektedirler.

2.1.3 Yazılım Tanımlı Ağların Çalışma Mekanizması

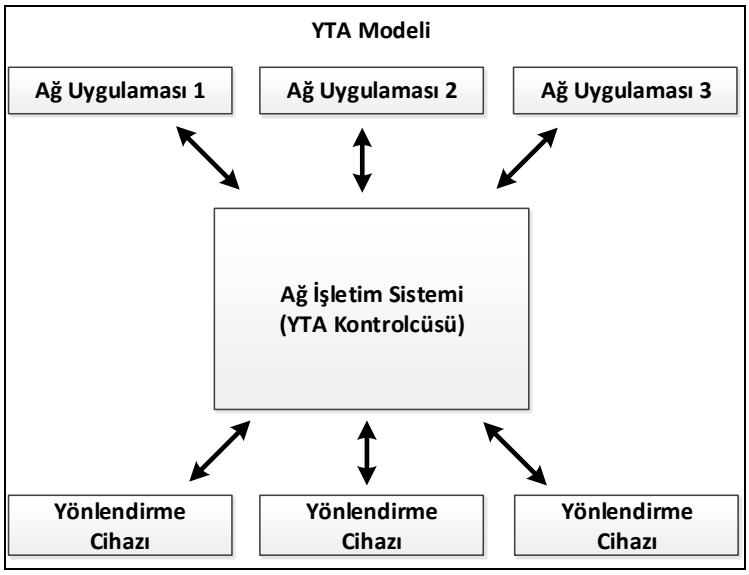
YTA'yı anlamak için bir bilgisayarın çalışma mantığını inceleyebiliriz. Bilgisayarları oluşturan birleşenler, donanım, işletim sistemi ve üzerinde çalışan uygulamalardır. İşletim sistemi, uygulamaların CPU, hafıza, depolama ünitesi ve giriş çıkış birimleri gibi donanımsal öğelerini kullanabilmesi için bir arayüz sağlamaktadırlar,

yani işletim sistemi, geliştirilen uygulamalar ile donanım birimleri arasında köprüyü sağlamaktadır (Şekil 2.2).



Şekil 2.2 İşletim sistemi modeli.

SDN yapısının çekirdeğinde aynen bir bilgisayarda olduğu gibi bir ağ işletim sistemi bulunmaktadır (Şekil 2.3). Ağ işletim sistemine YTA kontrolcüsü de denmektedir. YTA kontrolcüsü, ağ uygulamalarının yönlendirme cihazlarına erişimi için gereken köprüyü sağlamaktadır. Ağ uygulamaları, günümüz network ihtiyaçlarını sağlamak amacı ile geliştirilen yazılımlardır.



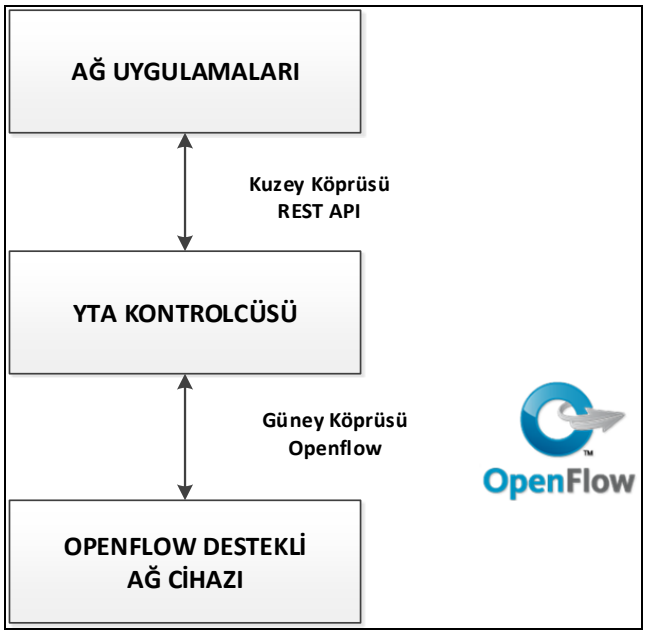
Şekil 2.3 YTA Modeli.

Geleneksel ağ cihazlarındaki (Şekil 2.1) genel amaçlı işlemci ve hafıza biriminden oluşan kontrol katmanından, donanımsal işlemci (ASIC) ve özel hafıza birimleri olan CAM ve TCAM'den oluşan veri katmanından bahsetmiştik. Kontrol katmanı, cihazın üzerindeki yapılandırmaya bağlı olarak yönlendirme ve ARP tablolarının hafıza biriminde oluşturulmasını sağlar. Daha sonra da bu tabloların cihazın içerisindeki donanımsal hafıza birimleri tarafından öğrenmesi sağlanır. Bu tablolar ile ağ cihazının ASIC'lerle paketleri hedefe yönlendirilmesi sağlanır (Kreutz vd., 2015).

YTA modelinde ise, cihazın beyni sayılabilecek kontrol katmanı, artık ağ cihazlarının içerisinde yer almamaktadır. Kontrol kararları geliştirilen ağ yazılımları tarafından gerçekleştirilmektedir. Bu yazılımların verdiği kararlar, YTA kontrolcü yazılımına REST API vb. arayüzler ile iletilmektedir. Bu iletişim kanalı kuzey köprüsü olarak isimlendirilmektedir (Şekil 2.4) (Qi & Li, 2016). Kuzey bağlantısına örnek olarak dinamik yönlendirme protokollerini çalıştıran Quagga sunucu verilebilir (Nascimento vd., 2010).

YTA kontrolcü yazılımı da, bu kararları Openflow protokolü ile ağ cihazlarına iletir. Bu iletişim kanalı güney köprüsü olarak adlandırılmaktadır (Şekil 2.4). Bu sistem marka bağımlılığı olmadan, Openflow protokolünü destekleyen yazılımsal veya

donanımsal tüm ağ cihazları ile çalışabilmektedir (Hu, 2014). Bu cihazlar; yönlendiriciler, anahtarlama cihazları, kablosuz erişimi cihazları vb. ağ cihazları olabilmektedir**.** Kontrol sunucu ile diğer kontrol sunucuları arasında oluşturulacak bağlantılara ise doğu-batı bağlantısı denmektedir (Hakiri, Gokhale, Berthou, Schmidt & Gayraud, 2014). Bu bağlantı tipi ile ağırlıkla ARGE için kullanılan NetFPGA kartları arasında kullanılmaktadır (Lai vd., 2014).



Şekil 2.4 YTA kuzey ve güney köprüsü.

Marka bağımlılığı olmayan Yazılım Tanımlı Ağlar kavramının ortaya atılması ile beraber, ağ cihazlarının çalışma mimarilerinde devrimsel nitelik olarak kabul edilebilecek bir değişiklik süreci başlamıştır (Azodolmolky, 2013). Böylece dağıtık değil merkezi bir karar mekanizması ile kararlar verilmekte ve tüm yapılandırmalar tek bir arayüz üzerinden sağlanabilir hale gelmiştir.

2.2. Openflow Protokolü

2.2.1 Openflow Protokolünün Tarihçesi

Klasikleşmiş ağ mimarisinde devrimsel bir yapı değişikliği sağlayan YTA mimarisi ile birlikte bütün cihazlar, bir kontrol sunucusu yardımı ile tek noktadan yapılandırılabilir ve monitör edilebilir hale gelmiştir (Monsanto, Reich, Foster, Rexford & Walker, 2013). YTA sistemi, sadece ağ yönetiminde kemikleşmiş olan bu sorunlara çözüm üretmek ile kalmamış, bilgisayar ağlarını kolaylıkla programlanabilir bir yapıya dönüştürmüştür. Son yıllarda YTA'nın tanınırlığı, bilgisayar ağları ile ilgilenen topluluklarda iyice artmış olsa da; bu kavramın dayandığı temeller geçtiğimiz yirmi yılda ortaya çıkmıştır (Feamster, Rexford & Zegura, 2013). Daha önceleri OPENSIG ile bu şekilde bir arayüz sağlanmaya çalışılmış, ancak yaygın bir kullanımı olmamıştır (Farhady, Lee & Nakao, 2015). 2004 yılında IEFT, RFC 3746 ile cihazların kontrol katmanı ve veri katmanı işlevlerini ayırmak amacı ile ForCES adlı bir arayüz standardı yayınlamıştır (Yang, Dantu, Anderson & Gopal, 2004). ForCes, kontrol katmanını ve veri katmanını aynı ağ cihaz içeresinde gösteriyordu.

Kontrol katmanının harici bir noktada olmasının öncüsü Stanford Üniversitesi'nin Ethane(2007) projesidir (Sezer vd., 2013). Bu kavramın somut olarak kullanıma geçmesini sağlayan, bu çalışmada ortaya çıkan Openflow protokolüdür ve Stanford Üniversitesi (Stanford University) ile Berkeley Kaliforniya Üniversitesi'nin (University of California at Berkeley) birlikte yürüttüğü bir araştırma projesinin ürünüdür. Openflow protokolü, kontrol sunucusu ile ağ anahtarı arasında bulunan güney bağlantısında taşınan trafik akış bilgilerinin aktarılmasını sağlamaktadır. Akış (flow) bilgisi, ağ cihazlarının paketleri doğru hedefe anahtarlanması için kullandıkları veridir (Tiwari, 2013). Bu verinin farklı kısımlarına müdahale ederek artık bir kenar anahtarı, yönlendirici veya güvenlik duvarı olarak kullanmak mümkün olmaktadır.

Bu çalışma sonrasında oluşturulan Openflow protokolü üzerine geliştirme çalışmaları, Standford Üniversitesi'nde akademik olarak devam etti. Bu çalışmalara donanımsal anahtar cihazları ile endüstriden şirketler de destek oldu (Koponen vd., 2010).

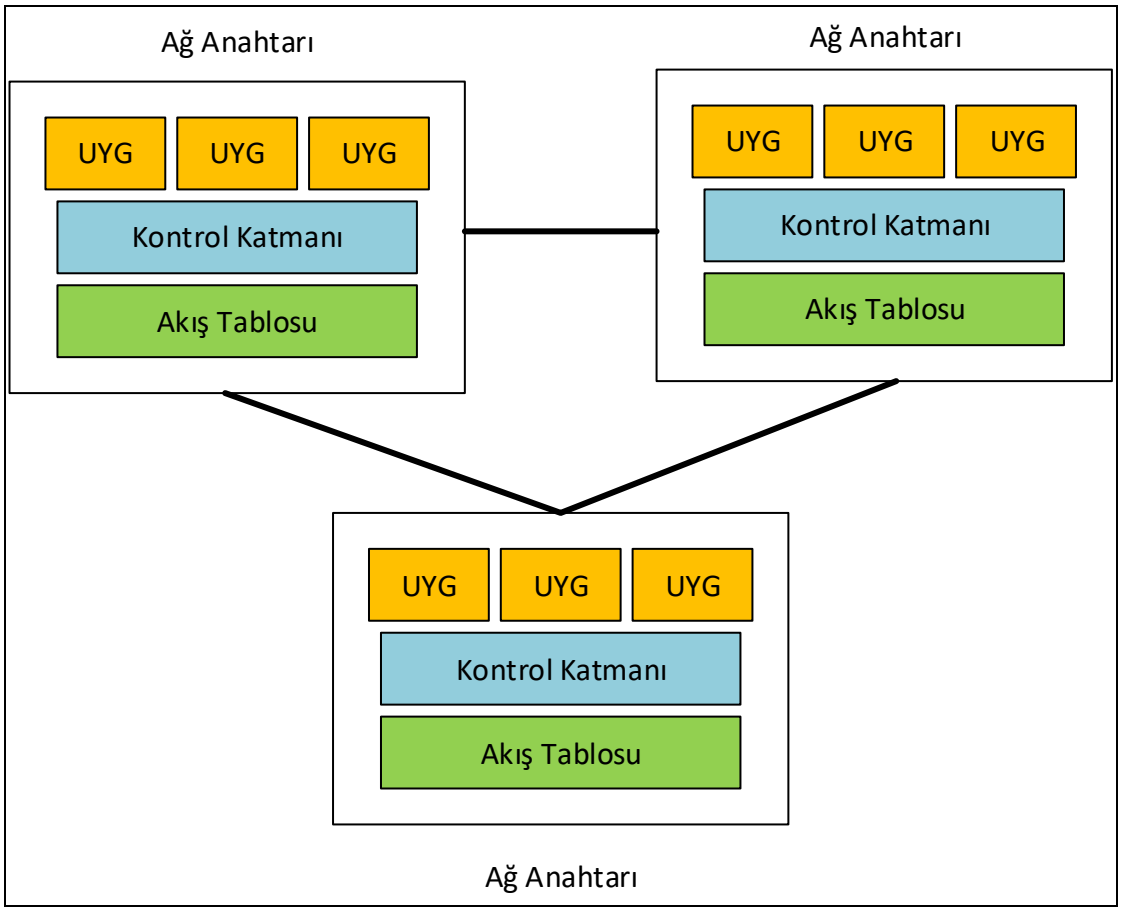
Openflow protokolünün ilk versiyonu olan Openflow 1.0, Standford Üniversitesi'nde 2009 yılında tanımlanmıştır (ONF, 2009). Günümüzde Openflow

protokolünün geliştirilmesi görevini üstlenen Open Network Foundation (ONF), 2011 Mart ayında kurulmuştur. ONF, 2011 Aralık ayında v1.1, 2012 Şubat ayında v1.2, 2012 Haziran ayında v1.3, 2013 Ekim ayında v1.4, 2014 Aralık ayında v1.5 versiyonunu yayınlamıştır. Ağ cihazları üreticilerinin büyük bir kısmı şu an ONF üyesidir.

2.2.2 Openflow Protokolünün Yapısı

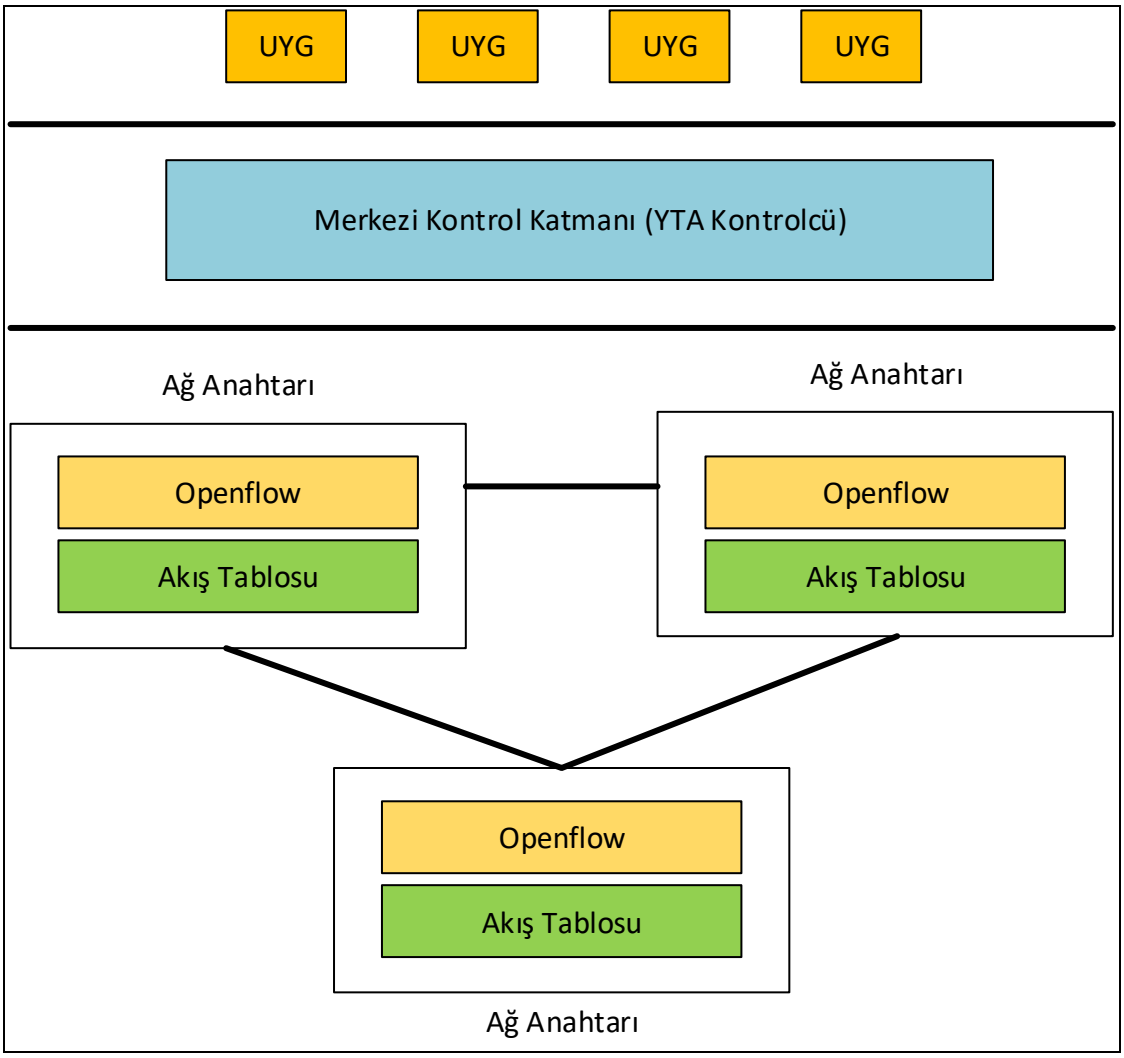
YTA mimarisinin güney köprüsü, Openflow destekli anahtar cihazlarından oluşur. Bu anahtarlar, sadece YTA kontrolcüsü ile yönetilebileceği gibi, hibrit olarak da çalışabilmektedir. Yani hem geleneksel anahtarlama mekanizmaları ile hem de YTA ile yönetilebilirler. Bu ağ cihazları donanımsal olabileceği gibi yazılımsal da olabilir.

Akış, bir anahtar cihaza giren pakete ne yapılacağını belirten bir tanımlamadır. Bu akış satırları güvenlik duvarlarındaki erişim kural listeleri gibidir. Ancak sadece pakete izin verilmesi veya yasaklanması üzerine karar vermezle, paketin nereden gideceği gibi kararları da belirlerler (Naous, Erickson, Covington & McKeown, 2008).



Şekil 2.5 Geleneksel ağ anahtar yapısı.

Geleneksel ağ cihazlarında akış tabloları, her cihazın içinde bulunan kontrol katmanı tarafından veri katmanına iletilir. (Doherty, 2016). Bu mimariyi kullanan anahtarlarda, akış tablosu cihazın içindeki uygulamalar ile yönetilir (Şekil 2.5). Openflow destekli anahtar cihazlarının akış tabloları ise YTA kontrolcüsü tarafından veri katmanına iletilir. Akış bilgilerinin, ağ cihazına iletilmesinde Openflow protokolü kullanılır. YTA kontrolcüsü, bu tabloları ya kendisi üzerinde bulunan yazılımla ya da harici geliştirilmiş ağ yönetim uygulamaları ile belirlerler (Şekil 2.6). İstenirse bir anahtar cihazda, birden fazla akış tablosu oluşturulabilir. Birden fazla akış tablosu oluşturulmasının sebebi; geleneksel cihazlarda yapılan yönlendirme fonksiyonu, erişim kural listesi fonksiyonu ve Network Address Translation(NAT) fonksiyonu gibi farklı misyonları olan işlemlerin farklı tablolarda tutulabilmesi içindir.



Şekil 2.6 Openflow destekli ağ anahtar yapısı.

Akış tablosunda ki her satır 3 tip veriyi tutar. Bunlar:

1. Paket başlık bilgileri: Burada Openflow anahtar cihaza giren paketlerin karar verilmesi öncesi kontrol edilecek kısımlar belirtilir.
2. Sayaçlar: Tabloda tutulan bu kısımda, akış ile ilgili istatistik tutulması sağlanılır. Bunlar bu satırla eşleşen paket sayısı veya bu paket tipinde geçen toplam trafik hacmi olabilir.
3. Aksiyon: Eşleşen paketlere ne yapılacağını belirten kısımdır.

Bu üç veririnin dışında, akış tablosunda hangi satırın önceliği olduğunu belirtmek için tabloya öncelik (priority) numarası eklenir. Ayrıca Openflow kontrolcüsüne erişilememe durumunda yani acil durumlarda devreye giren "Emergency" biti de akışa eklenebilir. Bu bit set edilmiş satırlar, acil durumda uygulanacak akış kurallarını göstermektedir.

Openflow 1.0 destekli anahtar cihazlarında eşleştirilen kısımları Çizelge 2.1'de bulabilirsiniz (ONF, 2009). İlerleyen Openflow versiyonlarında bu eşleşen kısımlar arttırılmıştır. Örneğin Openflow 1.1'e MPLS ve QinQ desteği de gelmiştir (ONF, 2011). Bu çalışmada Openflow 1.0 kullanılmıştır.

Çizelge 2.1 Openflow v1.0'da eşleşen kısımlar.

Alan	**Bits**	**Ne Zaman Uygulanabilir**	**Notlar**
Cihaza Giriş Port Numarası	Uygulanan cihaza bağlı değişmektedir	Tüm Paketlere	Numara olarak belirtilebilir ve 1'den başlar
Ethernet Kaynak Adresi	48	Etkinleştirilmiş portlardaki tüm paketlere	
Ethernet Hedef Adresi	48	Etkinleştirilmiş portlardaki tüm paketlere	
Ethernet Tipi	16	Etkinleştirilmiş portlardaki tüm paketlere	Openflow anahtar hem Ethernet hem de SNAP başlığına 802.2 standardını destekler
VLAN ID	12	Ethernet tipi 0x8100 olan tüm paketlere	
VLAN Öncelik Numarası	3	Ethernet tipi 0x8100 olan tüm paketlere	VLAN PCP kısmı
IP Kaynak Adresi	32	Tüm IP ve ARP paketleri	Subnet maskesi eklenebilir

IP Hedef Adresi	32	Tüm IP ve ARP paketleri	Subnet maskesi eklenebilir
IP Protokol Tipi	8	Tüm IP ve IP üzeri Ethernet paketleri, ARP paketleri	Sadece ARP op kod numarasının alt 8 biti kullanılır
IP ToS Bitleri	6	Tüm IP Paketleri	8 bit bellirtilen ToS bitlerinin üst 6 biti
Taşıma Katmanı kaynak port numarası / ICMP Tip Numarası	16	Tüm TCP,UDP ve ICMP paketleri	Sadece alt 8 biti
Taşıma Katmanı hedef port numarası / ICMP Kod Numarası	16	Tüm TCP,UDP ve ICMP paketleri	Sadece alt 8 biti

Openflow v1.0 destekli cihazlarda tutulan sayaçlara Çizelge 2.2'den ulaşabilirsiniz. Bu çalışma dahilde "Ulaşan Paket Sayısı" ve "Ulaşan Paketlerde Düşürülen Sayısı" atak kontrol sürecinde kullanılmıştır.

Çizelge 2.2 Openflow v1.0'da tutulan istatistik değerleri.

Sayaç	**Bits**
Tablo Başına	
Aktif Giriş	32
Paket Aramaları	64
Eşleşen Paket	64
Akış Başına	
Ulaşan Paket Sayısı	64
Ulaşan Byte Sayısı	64
Süre (saniye)	32
Süre (nanosaniye)	32
Port Başına	
Ulaşan Paket Sayısı	64

Yollanan Paket Sayısı	64
Ulaşan Byte Sayısı	64
Yollanan Byte Sayısı	64
Ulaşan Paketlerde Düşürülen Sayısı	64
Yollanan Paketlerde Düşürülen Sayısı	64
Ulaşan Paketlerde Hata Sayısı	64
Yollanan Paketlerde Hata Sayısı	64
Gelen "Frame Alignment" Hataları	64
Gelen "Overrun" Hataları	64
Gelen "CRC" Hataları	64
Çakışma	64
Kuyruk Başına	
Yollanan Paket Sayısı	64
Yollanan Byte Sayısı	64
Yollanan "Overrun" Hataları	64

Openflow v1.0 cihazlarda uygulanabilecek aksiyonlar iki tiptir. İlki tüm 1.0 destekli anahtarların desteklemesi gereken aksiyonlar, ikincisi opsiyonel olan aksiyonlar. Buna göre uygulanabilecek aksiyonlar aşağıda sıralanmıştır.

1. İlet (Forward): Tüm Openflow destekli anahtar cihazları aşağıdaki hedeflere doğru ilet aksiyonunu alabilirler.
    a. Tüm (All): Paketleri geldiği port hariç tüm portlara yönlendirmek için
    b. Kontrolcü (Controller): Paketi kapsülleyip kontrolcüye yollamak için
    c. Yerel (Local): Paketi yerel ağ yığınındaki anahtarlara göndermek için
    d. Tablo (Table): Akış tablosundaki aksiyonu uygulamak için (Sadece packet-out paketlerine uygulanır)
    e. In_Port: Geldiği porta paketi yollamak için
    f. Normal: Paketin, anahtar cihazın desteklediği geleneksel teknik ile yönlendirilmesi için (Opsiyonel bir aksiyondur)
    g. Flood: Spanning Tree Protokolü'ne (STP) uygun bir şekilde geldiği port hariç tüm portlardan yollamak için (Opsiyonel bir aksiyondur)
2. Düşür (Drop): Bu aksiyonda eşleşen tüm paketler düşürülür.

3. Kuyruğa Ekle (Enqueue): Bağlı olduğu portun kuyruğuna iletir. Bu işlem temel QoS desteğini sağlamak için kullanılır. Opsiyonel bir aksiyondur.
4. Alanı Değiştir (Modify-Field): Bu aksiyon ile paketin aşağıda listelenmiş kısımları değiştirilebilir. Opsiyonel bir aksiyondur.
    - VLAN ID
    - VLAN öncelik numarası
    - VLAN etiketinin kaldırılması
    - Kaynak MAC adresi
    - Hedef MAC adresi
    - Kaynak IP adresi
    - Hedef IP adresi
    - IPv4 ToS bitleri
    - Kaynak taşıma katmanı port numarası
    - Hedef taşıma katmanı port numarası

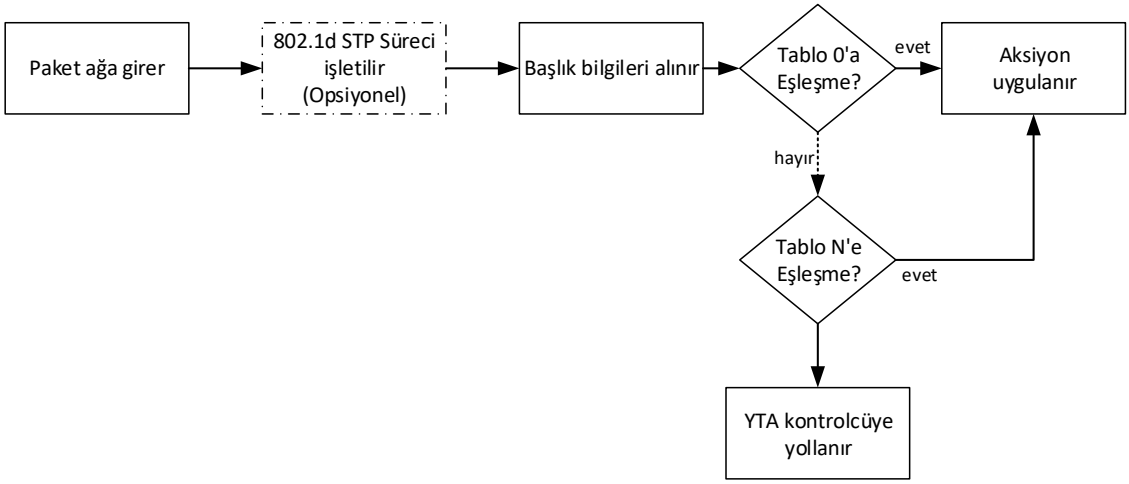
YTA destekli anahtara gelen bir paketin eşleşme süreci Şekil 2.7'de gösterildiği gibidir. Buna göre sisteme dahil olan bir paket, opsiyonel olarak öncelikle STP sürecine tabi tutulabilir. Daha sonra başlık bilgilerini alınır ve 0 numaralı akış tablosu ile karşılaştırılır. Eğer eşleşme yoksa bir sonraki akış tablosu kontrol edilir. Eşleşememe durumu devam ederse son akış tablosuna kadar iletilir ve hiçbir satır ile eşleşmiyorsa var sayılanda paket ile ilgili ne yapacağını sormak için paket YTA kontrolcüsüne iletilir.

Bir veri trafiği, bir Openflow anahtara geldiği zaman, en küçük numaralı akış tablosunun ilk satırından itibaren eşleşip eşlemediği kontrol edilir. Eşleşme gerçekleşmesi durumunda o akış kuralının aksiyonu gerçekleştirilir ve sayaç değerleri buna bağlı olarak arttırılır. Şayet hiçbir akış satırına eşleşmez ise yapılandırmaya bağlı olarak üç tane süreç yürütülebilir.

1. Paketi düşürmek
2. Paketi tüm arayüzlerden yollamak

3. Openflow kontrolcüsüne sormak

Eğer gelen trafik birden fazla akış satırı ile eşleşiyor ise önceliklendirmesine göre hangisinin uygulanacağına karar verilir. En öncelikli satır, gelen paket başlık bilgisine kesin eşleşen satırdır. Yani hiçbir wildcard (*) ifadesi barındırmayan satırdır. Burada ki "*" ifadesi, bu kısım ne olursa olsun anlamına gelmektedir. Eğer wildcard ifadesi bulunan birden fazla satır ile eşleşme gerçekleşirse, akış şemasın da yönetici tarafından atanan öncelik numarasına bakılır. Numarası büyük olan satır uygulanır. Şayet öncelik numaraları da eşit ise anahtar cihazın kendisi karar verir.



Şekil 2.7 YTA anahtar çalışma süreci.

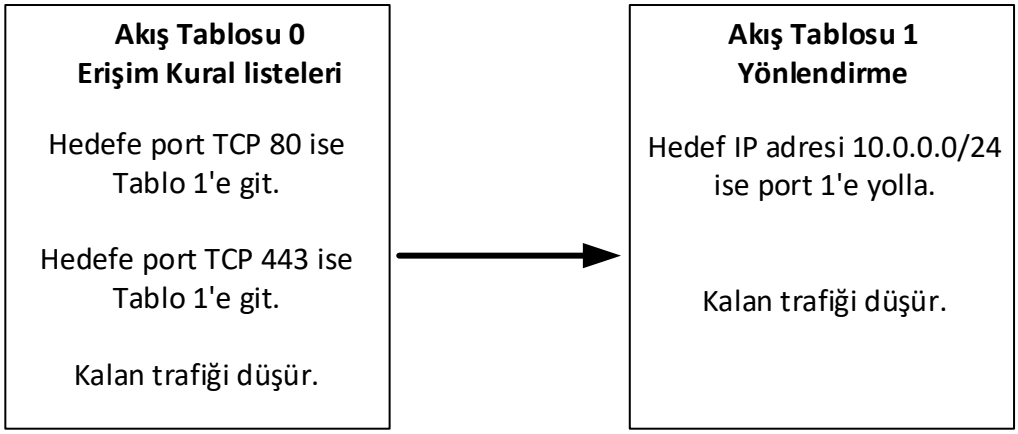
Tanımlanan akış tablosu satırına göre bir Openflow anahtar, farklı ağ cihazı rollerini üstlenebilir. Çizelge 2.3'de gösterilen birinci akış satırında sadece hedef MAC adresi belirtilmiş ve diğer ifadeler "*" olarak bırakılmış. Buna göre Openflow anahtar cihazı, aynen bir ikinci katman anahtar cihazı gibi sadece hedef MAC adresine bakarak karar verir. Çizelgede aksiyon kısmında görülebileceği gibi paket birinci porttan yollanır. İkinci akış satırında ise sadece hedef IP adresine göre bir tanımlama yapılmıştır. Bu ifade çerçevesinde, cihaz bir yönlendirici gibi sadece hedef IP adresine göre karar vermektedir. Üçüncü ifade de hem hedef MAC adresi hem de VLAN etiket numarasına bakılmaktadır. Buna göre cihaz VLAN destekli bir anahtar cihaz gibi karar vermektedir. Dördüncü ifade de protokol numarasının 6 (TCP) ve hedef port numarasının 80 olması durumunda paketin

düşürülmesi aksiyonu tarif edilmektedir. Bu durumda ise cihaz bir güvenlik duvarı gibi çalışmaktadır. Burada bu ifade diğer ifadeler ile kolaylıkla çakışabilir. Yani birden fazla satır ile eşleşebilir. Verilen örnekte anahtar cihaz, öncelik numarası en yüksek olan güvenlik duvarı tanımı satırını uygulayacaktır. Son ifade de ise hiçbir satıra eşleşmeyen bir trafik gelirse, varsayılanda ne yapılacağı tanımlanmıştır. Buna göre eşleşmeyen tüm trafik düşürülecektir.

Çizelge 2.3 Örnek akış tablosu.

Satır No	Anahtar Port No	Kaynak MAC	Hedef Mac	Ether Type	VLAN ID	VLAN ID Öncelik no	Kaynak IP	Hedef IP	Protokolü No	IP ToS Bitleri	Kaynak Port No	Hedef Port No	Aksiyon	Paket ve Byte Sayacı	Öncelik Numarası	Gerçekleştirilen Cihaz Tipi
1	*	*	00:10:20:30:40:50	*	*	*	*	*	*	*	*	*	Port 1		10	2. Katman Anahtar
2	*	*	*	*	*	*	*	10.0.1.0/24	*	*	*	*	Port 2		20	Yönlendirici
3	*	*	00:11:20:30:40:50	*	vlan 10	*	*	*	*	*	*	*	Port3		30	Vlan Bazlı Anahtar
4	*	*	*	*	*	*	*	*	6	*	*	80	Düşür		50	Güvenlik Duvarı
5	*	*	*	*	*	*	*	*	*	*	*	*	Düşür		0	

Bir cihazda birden fazla akış tablosu oluşturulması durumunda ise, öncelikle en küçük numaralı tablo kontrol edilir. Eğer akış, bu tabloda hiçbir satıra eşleşmez ise varsayılan eylem uygulanır. Şekil 2.8'de birden fazla akış tablosu içeren bir örnek verilmiştir.



Şekil 2.8 Çoklu akış tablosu.

Buna göre cihaz öncelikle 0 numaralı akış tablosunu inceler. Bu tabloya göre hedef port numarası TCP 80 ve TCP 443 olan trafiğe izin verir ve 1 numaralı akış tablosuna yönlendirir. 1 numaralı akış tablosunda ise hedef IP adresi 10.0.0.0/24 olan trafik port 1'e yönlendirilir. Bunun dışındaki tüm trafik düşürülür.

2.2.3 Openflow Mesajlaşma Tipleri

Openflow protokolü üç tip mesajlaşma tipi ile çalışmaktadır.

1. YTA kontrolcüsü ile anahtar arası mesajlaşmalar: Bu mesajlar, kontrolcü tarafından anahtar cihazı yönetmek ve denetlemek amacı ile kullanılır. Anahtar cihaz, mesaj tipine bağlı olarak bu mesaja cevap verebilir ya da vermeyebilir. Aşağıda bu şekildeki haberleşme tiplerini görebilirsiniz.

Features Mesajları: Kontrolcü ile bağlantı kurulduktan sonra anahtar cihazın desteklediği özellikleri öğrenmek için kullanılır. Anahtar cihaz buna cevap vermek zorundadır.

Configuration Mesajları: Bu mesaj tipi ile kontrolcü, anahtar cihaza bazı parametreleri atayabilir ya da bunları sorgulayabilir. Anahtar cihaz sadece sorgulama mesajlarına cevap vermek zorundadır. Bu mesaj tipi ile yapılan işlemler aşağıda sıralanmıştır.

Modify-State: Bu mesaj tipi ile kontrolcü, anahtar cihazın akış tablosuna akış satırı ekleyebilir, akış satırını değiştirebilir veya silebilir. Bu mesaj tipinin alt mesaj tipleri aşağıda sıralanmıştır.

- ADD: Bu mesaj tipi ile anahtar cihaza yeni akış satırı eklenir.
- MODIFY: Bu mesaj tipi ile anahtar cihazda bulunan akış satırı revize edilebilir.
- DELETE: Anahtarda bulunan bir akış satırının silinmesi için kullanılır.

Read-State: Bu mesajlar ile anahtar cihazdan istatistiksel bilgiler sorgulanır.

Send-Packet: Kontrolcü bu mesaj tipi ile anahtar cihaza, paketleri belirli bir porttan yolla emrini verir.

Barrier: Bu mesaj tipi ile kontrolcü bağımlılıkları veya tamamlanan işlemler ile ilgili bilgi sorgular.

2. Senkron mesajlar: Bu mesajlar hem kontrolcü hemde anahtar cihazdan yollanabilir. Karşılıklı durum kontrolü için kullanılır. Aşağıda bu şekildeki haberleşme tiplerini görebilirsiniz.

Hello: Bu mesaj tipi kontrolcü ve anahtar cihaz arasında ilk bağlantı kurulumunda kullanılır.

Echo Request/Reply: Hem kontrolcü hemde anahtar cihazın kullanabildiği bu mesajlar ile ikisi arasındaki haberleşmedeki gecikme, bant genişliği ve en önemlisi halen daha cihaz aktif mi bilgisi paylaşılmış olur. Yani “heartbeat” mesajlaşmasıdır.

Vendor: Anahtar cihazın ek özelliklere sahip olup olmadığını belirtmesi için kullanılır. Bu da yeni çıkmış Openflow standartlarına sahip olup olmadığının anlaşılmasını sağlar.

3. Asenkron mesajlar: Anahtar cihaz, kontrolcüye yeni gelen paketler ile ilgili sorguda bulunabilir ve kendisi ile ilgili son durum güncellemesi yapabilir. Aşağıda bu şekildeki haberleşme tiplerini görebilirsiniz.

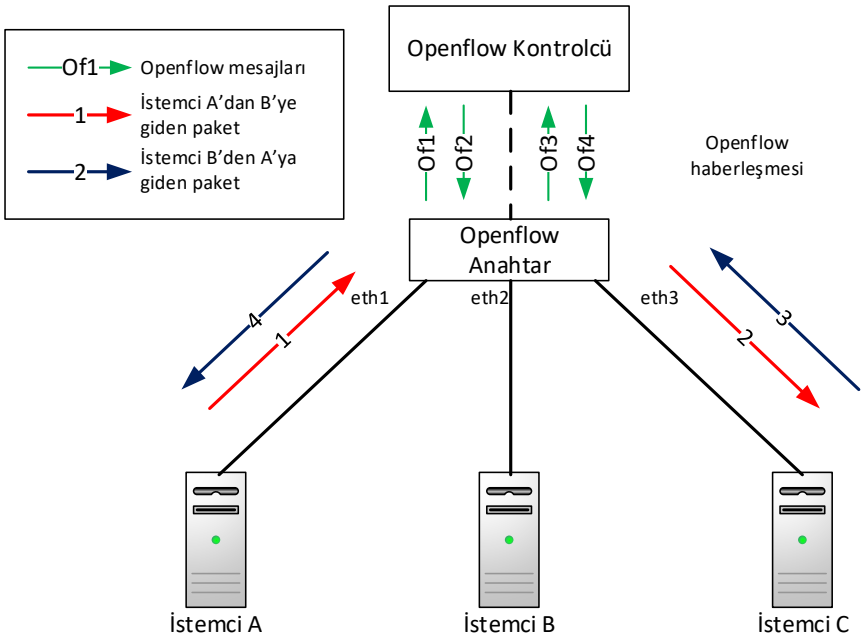
Packet-In/Packet-Out: Akış tablosunda satırlara eşleşmeyen bir trafik ile ilgili kontrolcüye ne yapılacağı hakkında bilgi sorgumak ve bu soruya cevap vermek amacı ile kullanılır.

Flow-Mod: Bu mesaj tipi, Packet-Out mesajı gibi bir paketin nereden gideceği bilgisinin kontrolcüden anahtara iletilmesinin yanı sıra paket iletilmeden önce pakette veya akış satırında yapılacak değişiklikleri de iletilir.

Flow-Removal: Bir akış tablosu satırının zaman aşımı süresinin dolması durumunda kontrolcüye haber vermek amacı ile iletilir.

Port-Status: Anahtar cihaz, port durumlarında bir değişiklik olması durumunda bunun bilgilendirmesini yapar.

Error: Anahtar cihaz bir hata yaşadığında kontrolcüyü bilgilendirmek için kullanır.



Şekil 2.9 Örnek YTA haberleşme süreci.

Şekil 2.9'daki YTA altyapısına sahip bir anahatar cihazda, İstemci A ve İstemci C arasındaki haberleşme incelenirse, haberleşme süreci şu şekildedir.

Adım 1: İstemci A, İstemci B'ye bir trafik yollar (Paket 1).

Adım 2: Paket Openflow destekli anahtar cihaza ulaşır.

Adım 3: Anahtar cihaz, bu haberleşme ile ilgili bir akış bilgisine sahip değilse ne yapacağını Openflow kontrolcüye Packet-In mesajı ile sorar (Of1 paketi).

Adım 4: Kontrolcü bu paketin ne yapılacağını Packet-Out mesajı ile anahtar cihaza iletir. Bu ağ yapısına göre, mesajı eth3 portuna yönlendir aksiyon bilgisi öğretilir (Of2 paketi). Yönlendirici bu paket yollanmadan önce revize edilmesini talep ediyor ise Flow-Mod paketi ile anahtar cihaza paketi yönlendirmesinin yanı sıra değişiklikte talep edebilir. Örnek olarak kaynak IP adresini, ToS bitlerini ya da akış zaman aşımı sürelerini revize ettirtebilir.

Adım 5: Akış bilgisine sahip anahtar cihaz, paketi eth3 portuna yönlendirir (Paket 2).

Adım 6: İstemci 3 paketi alır ve geri cevap yollar (Paket 3).

Adım 7: Anahtar bu haberleşme ile ilgili bir akışa sahip değil ise Packet-In mesajı ile kontrolcüye ne yapacağını sorar (Of3 paketi).

Adım 8: Kontrolcü, Packet-Out veya Flow-Mod paketi ile anahtara paketi eth1'den yollamasını iletir (Of4 Paketi).

Adım 8: Anahtar cihaz da İstemci C'den gelen paketi İstemci A'ya iletir (Paket 4).

Burada reaktif olarak çalışan bir Openflow dizaynı incelenmiştir. İstenirse haberleşme gerçekleşmeden önce hangi paketin nerden gideceği bilgisi anahtar cihaza öğretilebilir. Yani sistem proaktif olarak da oluşturulabilir.

2.3 Yapılmış Çalışmalar, Endüstriyel Standartlar ve Karşılaştırma

Saldırı ile mücadele çözümleri genel olarak saldırı tespit sistemleri ve saldırı engelleme sistemleri olarak görev yapmaktadır. Çalışmaların bir kısmı bu amaçlardan sadece biri için oluşturulmuşken, bazıları ise iki fonksiyonu da icra edecek şekilde yapılmıştır. Bunun yanı sıra yine saldırı çözümleri üzerine oluşturulmuş sistemler, genel amaçlı değil özel bir protokole ya da özel bir ağ tipine göre oluşturulmuştur.

Buna göre, bir sisteme özel çalışılmış yayınları incelersek, Wang (2019) DNS altyapısını savunmak amaçlı bir sistem ortaya koymuştur. El Houda, Khoukhi ve Hafid (2020), WisdomSDN ismi ile duyurdukları çalışma ile DNS servisi ile atakları engelleme

üzerine bir çalışma gerçekleştirmişler. K.Singh, P.Singh ve Kumar (2017), HTTP-GET ile yapılan sel DDoS atakları üzerine çalışmışlar. Sanjeetha, Shastry, Chetan ve Kanavalli (2020), YTA kullanarak aynı tip ataklar için bir engelleme sistemi geliştirmişler. Kurt, Yildiz, Certli, Sankur ve Cemgil (2018), SIP bazlı DDoS ataklarının tespiti üzerine çalışmışlar. Saravanan, Shanmuganathan ve Palanichamy (2016), uygulama katmanı üzerine yapılacak DDoS ataklarının tespiti üzerine çalışmışlar. Bulut bilişim altyapılarına yapılacak saldırı tespit ve engelleme üzerine Yusop ve Abawajy (2014) ve Miao vd. (2014), sis bilişim altyapıları üzerine ise Zhou, Guo ve Deng (2019), Priyadarshini ve Barik (2019) çalışmışlar. Agrawal ve Tapaswi (2018) ve Bhushan ve Gupta (2018) ise bulut sistemelere gelebilecek düşük hızlı DDoS atakları üzerine çalışmışlar. Sahoo vd. (2018), Carvalho, Abrao, Mendes ve Proença (2018) ve Yuan, Zou, Jin, Yu ve Yang(2017), çalışmalarında YTA ile veri merkezleri için DDoS atak tespiti ve engellemesi için çalışmalar yapmıştır. Wang, Zheng, Lou ve Hou (2015), YTA ile bulut bilişim sistemlerini DDoS ataklarına karşı korumak amaçlı bir sistem geliştirmişlerdir.

Mamolar vd. (2019), 5G ağlarını UDP sel DDoS ataklarına karşı korumak amaçlı bir çalışma yapmışlardır. Monge vd. (2019), 5G ağlarda trafik akış analizi ile DDoS tespiti üzerine çalışmışlar. Mamolar, Pervez, Calero ve Khattak (2018), enine DDoS algılama üzerine çalışmışlar. Sattar ve Matrawy (2019), 5G ağlarında "Slice Isolation" tekniği ile DDoS ataklarını engelleme üzerine çalışmışlardır. Chaudhary ve Harshan (2020), full duplex radyo haberleşmesi üzerine gerçekleştirilebilecek atakları engellemek üzerine bir engelleme sistemi oluşturmuşlardır. Thorpe, Tobin ve Murphy (2020), Near Field Communication (NFC) altyapılarında uygulama katmanı yaklaşımı ile atak engelleme sistemi üzerine çalışmışlardır.

YTA ağların getirdiği avantajları kullanan STRIDE isimli çalışmada istemcilere ağ aktivitelerine göre güven değerleri atayarak DDoS engelleme sistemi oluşturmuşlardır (Jantila & Chaipah, 2016). Aynı teknik ile akıllı ev altyapılarında kullanılan IoT sistemlerinden oluşturulabilecek BOTNET'leri tespit ve engelleme sistemi üzerine de bir çalışmada yapılmıştır (Abbas, Zahid, Hussain, Shah & Husnain 2020). Costa, Silveira, Araujo ve Gomes (2020), makine öğrenmesi ile IoT cihazlarına zarar verecek DDoS atak trafiğinin belirlenmesi için bir çalışma gerçekleştirmişler. C.Chen, L.Chen, Cai ve Tsai (2020), YTA alt yapısını kullanarak IoT cihazlarına gelen DDoS saldırılarını tespit amaçlı

bir çalışma gerçekleştirmişler. IoT sistemlerini korumak üzerine ise "Joint Defense Framework" isimli çalışma ile Li, Liu, Jin ve Yu (2021) bir atak engelleme sistemi oluşturmuşlar.

Nyasore, Zavarsky, Swar, Najyeju ve Dabra (2020), endüstriyel ağlarda kullanılan Modbus sistemlerinin TCP zafiyetlerine karşı derin paket incelemesi ile atak engelleme üzerine çalışmışlar. Javed vd. (2020), ODPV isimli akıllı haberleşme sistemlerine gerçekleştirilebilecek atakları engelleme üzerine bir çalışma gerçekleştirmişler.

Bilgisayar donanımları üzerine geçekleştirilebilecek ataklar üzerine çalışan Omar ve Khan (2020), IROHIDE ismi ile oluşturdukları sistem ile çok çekirdekli mikro mimarilere yapılabilecek atakları engellemek üzerine çalışmışlar. Wang, Yuan, Hou, Ji ve Meng (2020), yaptıkları çalışma ile tampon bellekler üzerine gerçekleştirilebilecek atakları engellemek amaçlı bir çalışma yapmışlardır. Reshikeshan ve Illindala (2020) ise Ethernet haberleşmesi üzerine yapılabilecek ve akıllı elektronik cihazların ele geçirilebilmelerine sebep olan ataklara karşı bir tespit ve engelleme sistemi oluşturmuşlar.

Toapanta, Caiced, Sanchez ve Gallegos (2020), e-ticaret altyapılarına gerçekleştirilebilecek atakları engelleme üzerine bir çalışma gerçekleştirmişler. Gunawan (2020), veri anonimleştirme yöntemi ile veri tabanlarına yapılabilecek saldırıları engellemek üzerine çalışmıştır. Wahida, Sattar ve Hossain (2020), aynı kişi hakkında anonimleştirmiş birden fazla veri setinin paylaşılması durumunda oluşabilecek veri mahremiyeti riskine, deneysel bir yaklaşım ile bir engelleme sistemi oluşturmuşlardır. Faruq ve Sattar (2020) ve Wahida, Sattar ve Hossain (2021), birden fazla noktada paylaşılan verilerin mahremiyeti üzerine bir model oluşturmuşlardır. Toapanta, Quimis, Gallegos ve Arellano (2020), yaptıkları çalışmada veri tabanları için bir güvenlik değerlendirme modeli prototipi oluşturmuşlardır. Oluşturdukları prototip, blok zinciri tekniği ile siber atakları engelleme üzerinedir.

Moving Target Defense (MTD) sistemi, istemcilere rastgele IP adresi atayan Openflow (YTA) tabanlı bir güvenlik çözümüdür (Jafarian, AL-Shaer & Duan, 2012). Bu sistem keşif ataklarına karşı koruma sağlayan bir çözümdür. Giotis, Argyropoulos, Androulidakis ve Kalogeros (2014), başka bir SDN tabanlı çözüm üzerine çalışmıştır. Bu sistem YTA'ın sağladığı kolay trafik yönetimi, monitör etme ve güvenlik duvarı

oluşturma özelliklerini kullanan bir çalışmadır. Bu çalışma dahilinde oluşturulmuş sistem, kampüs ağlarında Openflow ve Sflow kullanılarak bir anormallik ve atak engelleme mekanizmasıdır. Lai vd. (2020) ise 40 Gbps hızında NetFPGA tabanlı bir DDoS tespit sistemi oluşturmuşlardır.

Irenee, Hei, Wang, Ji ve Jiang (2020), Optimized K-Nearest Neighbor (OKNN) almoritması ile ağ akış analizi yaparak DDoS atak tespiti üzerine çalışmışlar. Cui, Zhang, He, Zhong ve Lu (2020), K-Means algoritmasını kullanarak YTA tabanlı bir DDoS tespit ve engelleme sistemi oluşturmuşlar. Firdaus, Munadi ve Purwanto (2020), K-Means++ ve Random Forest algoritmalarını kullanarak YTA ağlarında, DDoS atak tespiti üzerine çalışmışlar. Chen, Hou, Li ve Long(2020), Random Forest tekniğini kullanarak DDoS atak tespiti üzerine çalışmışlar. Lucky, Jjunju ve Marshall (2020), Lightweight Decision-Tree algoritmasını kullanarak DDoS sel ataklarının tespiti üzerine bir atak tespit sistemi geliştirmişler. Smiesko ve Uramova (2020), One-Parameter metodu ile DDoS atak algılaması üzerine çalışmışlar. Sumadi ve Aditya (2020), Ashi, Aburashed, Al-Fawa'reh ve Qasaimeh (2020) ve Sudugala, Chanuka, Eshan, Bandara ve Abeywardena (2020) makine öğrenmesi tekniği ile DDoS tespit sistemi geliştirmişler. Luong, Tran ve Le (2020), makine öğrenmesi ile YTA tabanlı ağlarda DDoS tespit ve engelleme sistemi geliştirmişler.

Mantur, Desai ve Nagegowda (2015) tarafından oluşturulmuş çalışma ise SDN altyapılarda imza bazlı güvenlik duvarı ve istatistiksel bazlı ağ saldırı tespit sistemidir. Yine Erhan ve Anarim (2020), %98 doğrulukla istatistiksel değişiklerden yararlanarak DDoS atak tespiti üzerine çalışma yapmışlar. Joldzic, Djuric ve Vuletic (2016) tarafından yapılmış çalışmada, yerel DoS ataklarını tespit metodu üzerine yoğunlaşmışlar. ArOMA ise YTA tabanlı başka bir DDoS engelleme sistemidir. Bu çalışma, atağa maruz kalan yerde ve yerel ISP'de atağın durdurulması üzerinedir (Sahay, Blanc, Zhang & Debar 2017).

Woodpecker isimi ile oluşturdukları çalışmada Wang, Li, Jiang, Jia ve Wu (2018), düşük hacimli sel ataklarına karşı bir tespit ve engelleme sistemi oluşturmuşlar. Nugraha ve Murthy (2020), derin öğrenme tekniklerini kullanarak YTA tabanlı ağlarda yavaş DDoS ataklarının tespiti üzerine bir çalışma yapmışlar.

Akin vd. (2020), yaptıkları çalışmada, Attack Blocking System (ABS) isimli bir atak engelleme sistemi oluşturmuşlardır. Bu sistem BGP protokolünden bağımsız atak yapan tarafa uyarıda bulunabilen bir sistemdir. Ancak merkezi kontrol ve merkezi atak bloklama mekanizmasına sahip değildir.

DDoS saldırılarını azaltmak için oluşturulmuş endüstriyel standartlardan bir tanesi RFC 3882'dir (Turk, 2004). Bu teknik Destination-based Remotely Triggered Black Hole (D/RTBH) olarak da adlandırılır. Bu sistem de, mağdur kurum BGP community değeri olarak 666 atadığı bir BGP anonsu yapar. Kurum bu anons ile bağlı olduğu ISP'ye, DDoS atağı gelen kendi sunucu IP adresinin trafiğini kara deliğe yöndirilmesini talep eder. Bu sayede kuruma doğru gelen atak trafiği yerel ISP'de bloklanır. Ancak bu durum, kurumun hizmet verdiği sunucunun erişimi de durur. Hizmetin devam edilebilmesi için ilgili sunucunun DNS kaydı değiştirilmek zorundadır.

Diğer endüstriyel çözüm ise Source-based Remotely Triggered Black Hole (S/RTBH) olarak da adlandırılan ve RFC 5635'de tarif edilen tekniktir (Kumar & McPherson, 2009). Bu teknik ile atağın hedefi olan kurumun bağlı olduğu ISP, atağı kaynak IP adresi ile bloklayabilmektedir. Ancak bu atamalar, atağa maruz kalan kurumun, BGP'de yapacağı manuel konfigürasyon ile gerçekleştirilebilir. Atak kaynağı ISP'ye bir bilgilendirme yapılmamaktadır.

Flowspec olarak isimlendirilen diğer bir endüstriyel çözümde, RFC 5575 ile tanımlanmıştır (Marques vd., 2009). Bu teknik, ISP'ler arasında BGP protokolü kullanarak saldırı bilgisi paylaşan bir sistemdir. Ancak büyük ISP'ler bu tipten bilgileri rakip ISP'ler ile paylaşmayı tercih etmezler (Chen, Hwang & Ku, 2007). Böyle bilgileri paylaşmak, özel bilgilerin sızdırılması olarak görülmektedir (Zhu, Tang, Shen, Du & Guizani, 2018). Bu sistem, saldırının atak kaynağına mümkün olabilen en yakın ISP'de erişim kural listeleri ile bloklanmasını sağlar. Atak kaynak kontrolü de yönlendirme tablosuna bakarak kontrol edilebilir. Ancak bu kontrol gerçekte böyle bir atağın olduğunu göstermez.

Bu üç standartta, ISP üzerinden çözüm sağlamaktadır. Atak kaynağı kuruma, bir bloklama yada bilgilendirme yapmaz. Ayrıca BGP ile çalıştıkları için atağa maruz kalan kurum BGP desteklemiyorsa, bu teknikler ile atak bloklama talebi yapamaz.

Bu çalışma ile sadece saldırı engelleme süreci üzerine çalışan bir sistem geliştirilmiştir. Sistem tüm ağ tipleri ve tüm trafik tipleri için kullanılabilir. Yapılmış çalışmalar ve konu ile alakalı standartlar ile bu çalışmada oluşturulan tekniğin karşılaştırmasını Çizelge 2.4'de bulabilirsiniz.

Çizelge 2.4 Karşılaştırma tablosu.

Referans	**Trafik tipi**	**Tespit/ Engelleme**	**BGP bağımsız**	**Başka Bir AS'de Engelleme**	**Merkezi Bilgi Temini**	**Merkezi Bloklama**	**Atak Kaynağını Uyarma**	**Atak Kaynağında Atak Kontrolü**
D/RTBH (RFC 3882)	Tüm	Engelleme	Hayır	Hayır	Hayır	Hayır	Hayır	Hayır
S/RTBH (RFC 5635)	Tüm	Engelleme	Hayır	Hayır	Hayır	Hayır	Hayır	Hayır
Flowspec (RFC 5575)	Tüm	Engelleme	Hayır	**Evet**	Hayır	Hayır	Hayır	Hayır
Wang (2019) El Houda vd. (2020)	DNS Trafiği	Tespit/ Engelleme	Evet	Hayır	Hayır	Hayır	Hayır	Hayır
K.Singh vd. (2017) Sanjeetha vd. (2020)	HTTP-GET trafiği	Tespit/ Engelleme	Evet	Hayır	Hayır	Hayır	Hayır	Hayır
Kurt vd. (2018)	SIP trafiği	Tespit	Evet	Hayır	Hayır	Hayır	Hayır	Hayır
Saravanan vd. (2016)	Uygula-ma katmanı	Tespit	Evet	Hayır	Hayır	Hayır	Hayır	Hayır
Yusop vd. (2014) Miao vd. (2014) Wang vd. (2015) Agrawal vd. (2018) Bhushan vd. (2018)	Bulut Bilişim için	Tespit/ Engelleme	Evet	Hayır	Hayır	Hayır	Hayır	Hayır
Zhou vd. (2019) Priyadarshini vd. (2019)	Sis Bilişim için	Tespit/ Engelleme	Evet	Hayır	Hayır	Hayır	Hayır	Hayır
Sahoo vd. (2018) Carvalho vd. (2017)	Veri Merkez-leri için	Tespit/ Engelleme	Evet	Hayır	Hayır	Hayır	Hayır	Hayır
Mamolar vd. (2018) Mamolar vd. (2019) Monge vd. (2019) Sattar vd. (2019)	5G Mobil Ağlar İçin	Tespit/ Engelleme	Evet	Hayır	Hayır	Hayır	Hayır	Hayır
Chaundhary vd. (2020)	Radyo Haber-leşmesi için	Tespit/ Engelleme	Evet		Hayır	Hayır	Hayır	Hayır
Thorpe vd. (2020)	NFC için	Tespit/ Engelleme	Evet		Hayır	Hayır	Hayır	Hayır

Abbas vd. (2020) Costa vd. (2020) Chen vd. (2020) Li vd. (2021)	IoT için	Tespit/ Engelleme	Evet		Hayır	Hayır	Hayır	Hayır
Nyasore vd. (2020)	Modbus için	Tespit/ Engelleme	Evet		Hayır	Hayır	Hayır	Hayır
Javed vd. (2020)	ODPV için	Tespit/ Engelleme	Evet		Hayır	Hayır	Hayır	Hayır
Omar vd. (2020)	Çok çekir-dekli İşlemcil er için	Tespit/ Engelleme	Evet		Hayır	Hayır	Hayır	Hayır
Wang vd. (2020)	Tampon bellek-ler için	Tespit/ Engelleme	Evet		Hayır	Hayır	Hayır	Hayır
Reshikeshan vd. (2020)	Ethernet için	Tespit/ Engelleme	Evet		Hayır	Hayır	Hayır	Hayır
Toapanta vd. (2020)	E-ticaret sistemle ri için	Engelleme	Evet		Hayır	Hayır	Hayır	Hayır
Gunawan (2020) Toapanta vd. (2020)	Veri tabanla-rı için	Engelleme	Evet		Hayır	Hayır	Hayır	Hayır
Wahida vd (2020 Faruq vd. (2020) Wahida vd. (2021)	Veri mahre-miyeti için	Engelleme	Evet		Hayır	Hayır	Hayır	Hayır
Jafraian vd. (2012)	İstemci bazlı	Engelleme	Evet		Hayır	Hayır	Hayır	Hayır
Gioris vd. (2014)	Tüm flow	Tespit/ Engelleme	Evet		Hayır	Hayır	Hayır	Hayır
Joldzic vd. (2016)	Yerel DoS için	Tespit	Evet		Hayır	Hayır	Hayır	Hayır
Sahay vd. (2017)	Yerel ISP için	Tespit/ Engelleme	Evet		Hayır	Hayır	Hayır	Hayır
Wang vd. (2018) Nugraha vd. (2020)	Düşük hacimli ataklar için	Tespit/ Engelleme	Evet		Hayır	Hayır	Hayır	Hayır
Lai vd. (2020)	Tüm	Tespit/ Engelleme	Evet		Hayır	Hayır	Hayır	Hayır
Mantur vd. (2015) Jantila vd. (2016) Irenee vd. (2020) Cui vd. (2020) Firdaus vd. (2020) Chen vd. (2020) Lucky vd. (2020) Smiekso vd. (2020) Sumadi vd. (2020) Ashi vd.	Tüm	Tespit/ Engelleme	Evet		Hayır	Hayır	Hayır	Hayır

(2020) Sudugala vd. (2020) Luong vd. (2020) Erhan vd. (2020)								
Akın vd. (2020)	Tüm	Engelleme	Evet	**Evet**	Hayır	Hayır	**Evet**	**Evet**
USBS	Tüm	Engelleme	Evet	**Evet**	**Evet**	**Evet**	**Evet**	**Evet**

BÖLÜM 3

USBS SİSTEMİ HEDEFLERİ VE SBP PROTOKOLÜNÜN YAPISI

3.1 Ulusal Saldırı Bloklama Sistemi (USBS) Hedefleri

Bu çalışmanın asıl amacı ağ saldırıların kaynağında bloklanması ve atak etkilerinin minimuma indirilmesidir. Bunun içinde ülke dahilinde bu sistemi kurmuş ağ yöneticileri arasında bir atak bilgilendirme sistemi oluşturmaktır. Bir atağın bloklanmasını sağlamak için bu sistem atağın gerçekten olup olmadığını kontrol edebilmekte ve yanlış bir karar alınmasını da engellemektedir. Bu çalışma dahilinde bu sistem için belirlenmiş hedefler aşağıda belirtilmiştir.

1. Sistem tüm yönlendirme protokollerinden bağımsız olarak çalışmalıdır. Mevcut durumda bu şekilde dağıtık çalışan benzeri sistemler BGP protokolünü kullanmaktadır. Bu da BGP kullanmayan bir çok kurumun o sistemlere dahil olmasını engellemektedir.
2. Mevcut sistemler ISP yardımı ile yüklü atakları engellemeye çalışmaktadır. Geliştirilen sistem ile kurumlar, ISP'lerden bağımsız olarak atakların kaynağında kesilmesine yardımcı olabilmelidir. Popüler olarak ISP'ler tarafından kullanılan "Source-based Remotely Triggered Black Hole" tekniği atak trafiğini NULL 0 arayüzüne yollayarak engellemektedir. Bu da atak altındaki sunucusunun servis dışı kalmasına sebep olmaktadır. Bu sistem dahilinde atak bloklanma sürecinde sunucu servis dışı kalmayıp, hizmetine devam edebilmelidir.

3. Böyle bir sistem ile atak engelleme artık sadece ISP'lerin üzerine yoğunlaştığı bir sorun olmayacaktır. Tüm paydaşların siber güvenlik sorununa çözüm sağladığı bir sistem oluşturulmaldır.
4. Kurumun hizmet aldığı ISP, bu sisteme dahil olmasa bile kurum bu sisteme dahil olabilmelidirler.
5. Hem ISP hem de ondan hizmet alan kurum bu sisteme dahil ise ve atak o kurumdan geliyorsa, uyarı sadece ilgili kuruma verilebilmelidir. Bu sayede ISP bu gibi atak bilgilendirmesi ile meşgul edilmeyecektir.
6. Bu sistemin YTA destekli bir ağda kullanılması durumunda ve atak var uyarısı geldiğinde atağın gerçekten var olup olmadığı tespit edilebilmelidir.
7. Atak trafiğinin gerçekten var olduğu tespit edilir ise YTA kontrolcüsü yardımı ile atak kaynağında bloklanabilmelidir. Atak trafiğinin olmadığı tespit edilir ise de boşuna bir müdahale yapılmamalıdır.
8. Sistem tespit edilmiş atak trafiğini bloklayabileceği gibi, atak sonladığında bloklamayı otomatik olarak kaldırabilmelidir. Bunun için atak bloklandıktan sonra belirli periyotlarla bloklanan trafiğin paket istatistiği takip edilip ve paket sayısında artış durduğunda otomatik bloklama kaldırılmalıdır.
9. Tespit edilen atağın kendi ağında olması durumunda, sistem tüm fonksiyonları ile çalışabilmelidir.
10. Eğer kurulduğu ağ, YTA desteği bulunmayan bir ağ alt yapısına sahip ise bu sistem atak bilgilendirme sistemi olarak kullanılabilmelidir. Bu bilgilendirme, sistemin arayüzü, SMS veya eposta gibi çözümlerle gerçekleştirilebilir.
11. Kurumsal ağ yöneticileri kendisinde bulunan ve botnete dahil olan PC'leri bu sistem sayesinde tespit edebilmelidir. Bu sayede zararlı yazılım bulaşmış bilgisayarların temizlenmesini sağlanabilecektir. Bu da uzun vadede botnet'lerin küçülmesinin de önünü açacaktır.
12. Mevcut saldırı engelleme sistemleri çoğunlukla DDoS saldırılarına çözümler sunar. Bu sistem ile sadece yüklü trafik yaratan ataklar değil tek kaynaktan gelen ve yüklü bir trafik yaratmayan saldırı girişimleri de kaynağında (ör. SSH kolay şifre saldırıları) engellenebilmelidir.
13. Sistem İnternet'e özel teknolojilerden bağımsız olmalıdır. Bu sayede askeriye ağ altyapıları gibi kapalı sistemlere de uygun olabilmelidir.

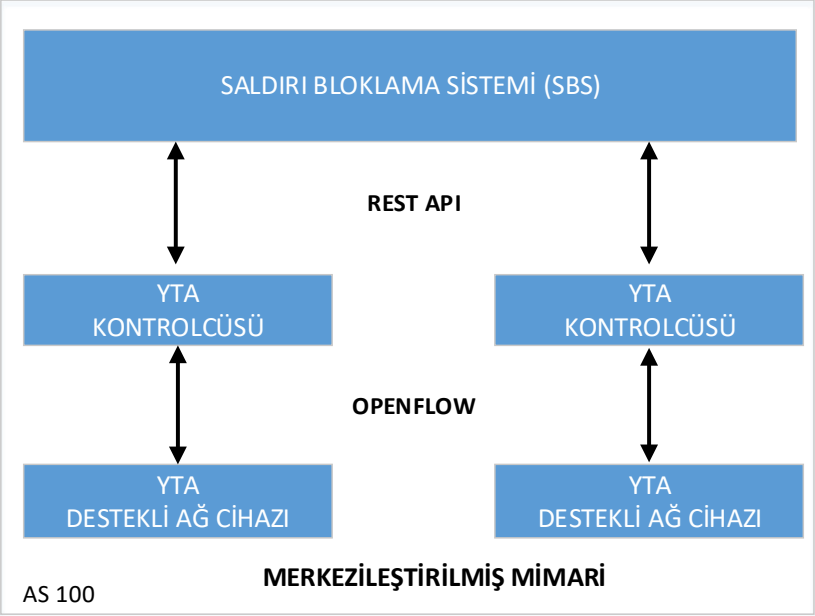
14. Bu sistem ile global çapta olan ataklarda sadece ISP'ler değil, kurumlar da atakların engellenmesi sürecine dahil olabilmelidir. Bu sistemden atak sürecine dahil olan kurumların tüm siber güvenlik ekipleri de bilgi alabilmeldir.
15. Sistem, ülke çapında kullanılabilmesi için, ekstra merkezi ağ altyapısı bilgilendirme sistemine sahip olmalıdır. Oluşturulacak bu sistem, yetkisiz bloklama taleplerinin önlenmesi yani sistemin güvenliği için çok önemlidir.
16. Merkezi bilgilendirme sistemi sadece bilgilendirme görevi için değil, aynı zamanda ulusal bir atak bloklama sistemi olarak da çalışabilmelidir. Bu amaç ile merkezde girilecek bir hedef IP adresi için sisteme dahil olan tüm ağlarda bloklama işlemi yapılabilmelidir. Yine aynı şekilde bloklama işlemi merkezi olarak belirtilmesi durumunda tüm sisteme dahil ağlarda kaldırılabilmelidir.

3.2 Sistemin ve Protokolün Yapısı

Saldırı kontrol sisteminin amaçlarından biri, saldırıya uğrayan tarafın saldırıya sebep olan kaynağı bilgilendirmesidir ve mümkünse saldırı kaynağını ağda kaynağa en yakın noktada bu atağın kesilmesini sağlamasıdır. Bu sistem bir saldırı tespit sistemi değildir yani saldırı tespiti üzerine çalışmamaktadır. Sadece dağıtık olarak saldırı bilgilendirme ve engelleme sistemidir. Saldırı tespiti üzerine birçok çalışma ve çözüm hali hazırda mevcuttur (Porras vd., 2012). Bu sistem Snort gibi mevcut bir saldırı tespit sistemlerine (STS) entegre olabilir (Xing, Huang, Xu, Chung & Khatkar, 2013). Bir honey pot sisteminden de bilgi alabilir(Li, Sun & Zhang, 2011). YTA ile paket istatistik bilgilerinin kolaylıkla alınabilmesi sayesinde anormalliğin algılaması ve bunun USBS sistemine belirtmesi ile de atak bloklanabilir. Zaten bu sistem, YTA bir sistemde daha başarılı olacağı düşünülerek tasarlanmıştır. Yukarda belirtildiği gibi farklı saldırı tespit kaynağından alınacak bu bilgiler bir API (Application Programming Interface) aracılığı ile bu sisteme iletilebilir. Bunların yanı sıra ağ/güvenlik yönetim birimi bizzat kendi tespit ettiği bir saldırı bilgisini özel bir arayüz üzerinde de sisteme girebilir. Bu çalışma dahilde bu şekilde sisteme girilen saldırıların engellenmesi üzerine çalışılmıştır.

Yukarıda da belirttiğimiz üzere sistemin ana çalışma düzeni Yazılım Tanımlı Ağ (YTA) altyapısına sahip yerlerde kullanılması için düşünülmüştür. Buna göre yazılım, YTA kontrolcüsünden ilgili bilgileri almak ve bloklama emrini verecek şekilde

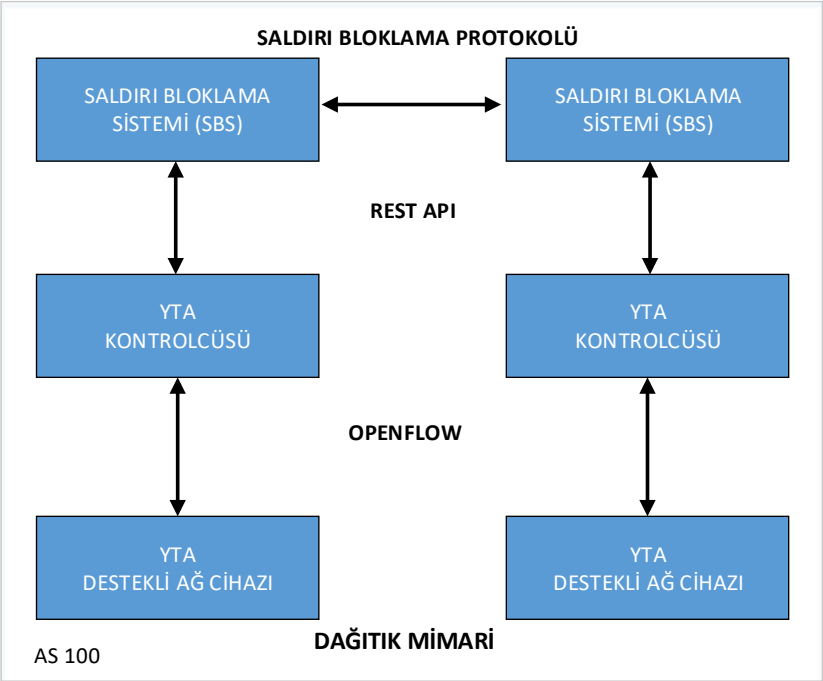
tasarlanmıştır. USBS, bir kurum içerisinde birden fazla YTA kontrolcü ile de çalışabilir. Yani bir kurumda bir tane bloklama sunucusu ve birden fazla ona bağlı YTA kontrolcüsü olabilir. [Merkezi Mimari, Şekil 3.1]



Şekil 3.1 Merkezi mimari.

Bu tarz bir kullanım daha çok kurumsal ağlarda kullanılabilecek bir modeldir. Ayrıca daha büyük bir ağ altyapısına sahip olan İnternet servis sağlayıcılarında ya da global şirketlerde birden fazla bloklama sunucusu kurularak da sistem devreye alınabilir (Dağıtım Mimari, Şekil 3.2).

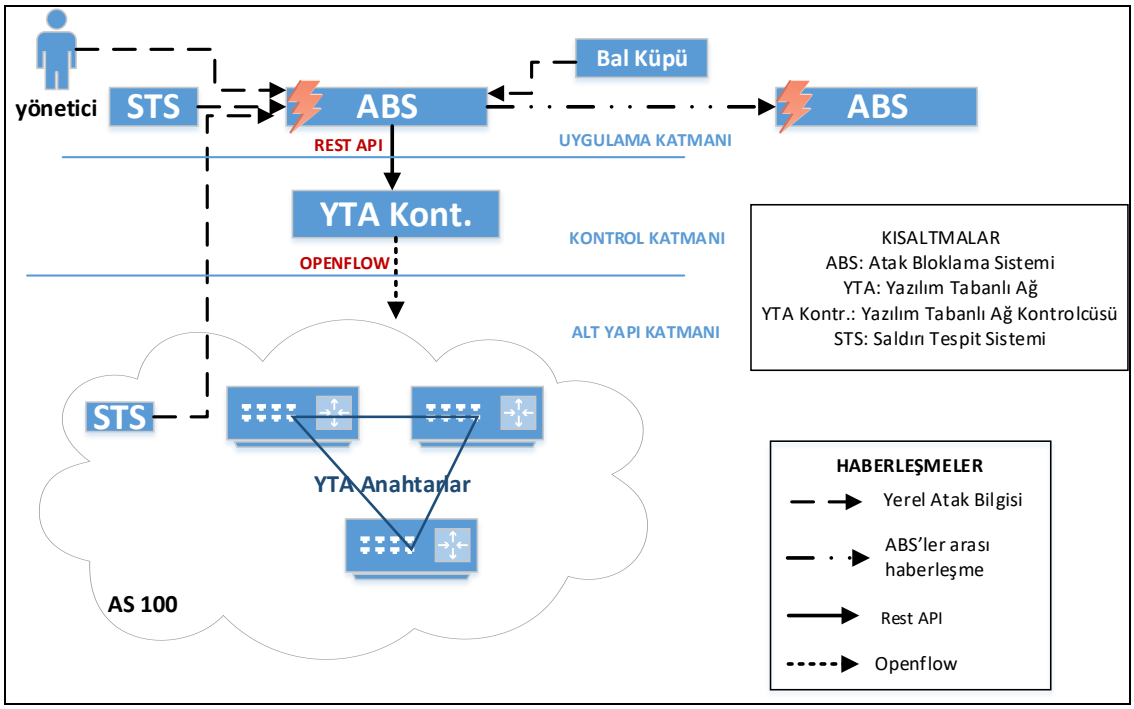
Geliştirilen yazılım direk tek bir bloklama sunucusuna göre oluşturulmuştur ancak bu hali ile dağıtık mimari olarak kullanmak da mümkündür. Aynı kurum altında tek bloklama sunucusu ve birden fazla YTA barınabilecek merkezi mimarilerde çalışabilmesi yazılama yapılacak bir eklenti ile mümkün olabilecektir.



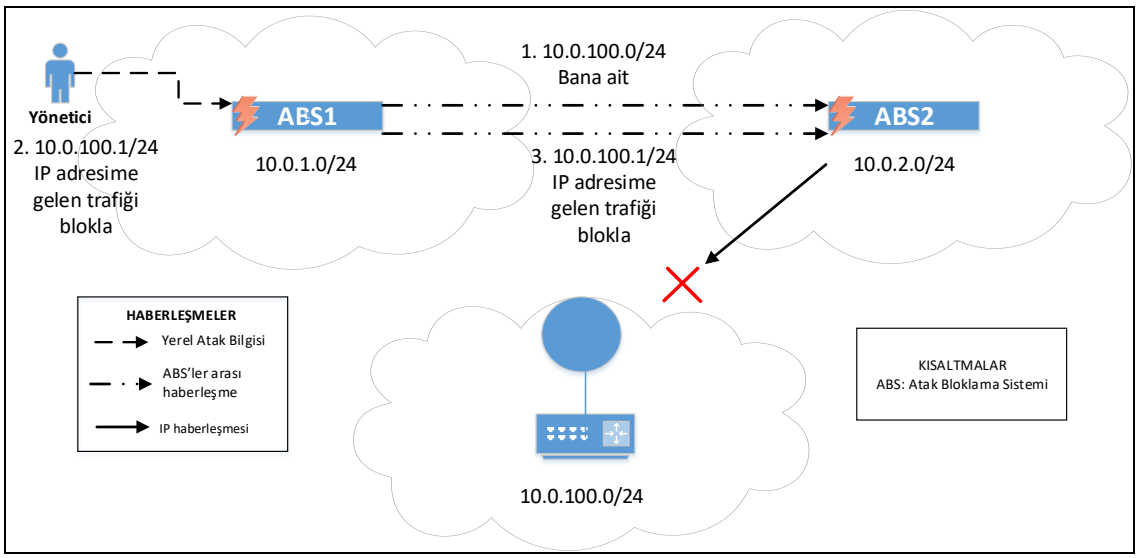
Şekil 3.2 Dağıtık mimari.

Doktora çalışması dahilde yapılan yayın ile oluşturulmuş ABS sistemi sadece ABS sunucularından oluşan ve dağıtık sistem olarak çalışan bir yapı idi (Şekil 3.3). Yayının sonuçlar kısmında da belirtildiği gibi bu sistemin en büyük sorunu bir atak bilgisi geldiği zaman, atağın kaynak IP adresinin hangi ABS sunucusunun sorumluluğu altında olduğunun belirlenmesi idi.

Bu çalışmada sunucular arasında kurulacak komşuluk ile birbirlerine sorumlu oldukları IP bloklarını birbirlerine öğretebilir şekilde belirtilmişti. Ancak buda güvenlik riskini beraberinde getirmekte idi. Şekil 3.4'de görüldüğü üzere ABS1 sunucusu kendisine ait olmamasına rağmen 10.0.100.0/24 ağ bloğunun kendisine ait olduğunu ABS2'ye öğretilebilmekte ve peşinden ABS2'den 10.0.100.0 ağına giden bir trafik için bloklama talep edip haberleşmede kesintilere yol açabilmekteydi.



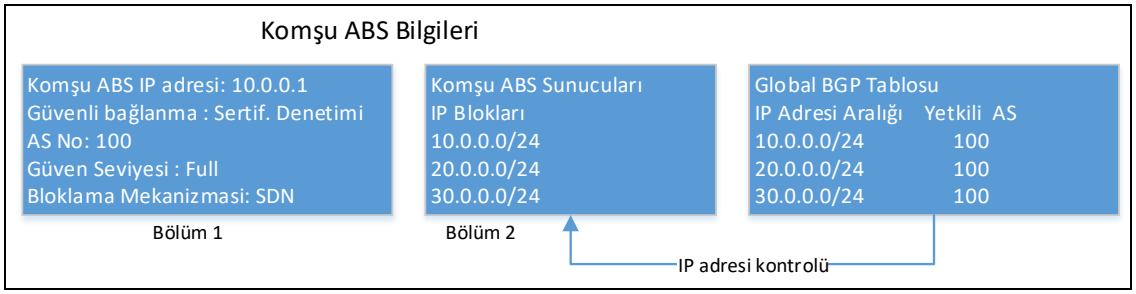
Şekil 3.3 ABS sistemin çalışma yapısı (Akin vd., 2020).



Şekil 3.4 Sistemin kötü amaçlı kullanımı.

Yine aynı çalışmada buna çözüm olarak öğrenilen bu bilginin RIPE gibi bölgesel IP dağıtıcılarından sağlanacak veri tabanları ile bilgi onaylaması yapılabileceği

belirtilmiştir. Yada global BGP routing tablosundan da elde edilecek AS bilgileri ile kontrol yapılabilecek şekilde düşünülmüştür (Şekil 3.5).



Şekil 3.5 Komşu ABS'nin tanımı ve ağ bilgilerinin kontrolü.

Ancak BGP protokolü kullanmayan şirketler genellikle bu veri tabanında kendilerine ait bir kayıt yapmadıkları için çoğu IP bloğunun kontrol edilmesi mümkün olmamaktadır. Çalışmada buna çözüm olarak kurumun kendi yerelinde hangi IP bloğu, hangi ABS sunucusu tarafından yönetildiği veri tabanlarını oluşturabileceği ve bu veri tabanlarının kurumlar arasında paylaşılabileceği üzerinde durulmuştur. Ancak buda sistemin çok büyümesi durumunda yönetilmesi pek mümkün olmayacak bir hal oluşturacaktır. IP adresi aralıkları, kurumların İnternet servis sağlayıcılarını değiştirmesi durumunda değişmektedir ve bu değişikliğin tüm sunucularda güncellenmesi bile başlı başına bir sorundur. Bu durum bazı sunucularda güncellenmiş bazılarında güncellenmemiş veri tabanlarına sebebiyet verebilir. Bu da sistemin yanlış karar verme riskini arttırır. Aynı yayının öneriler kısmında bu sisteme atanmış bir merkezi sunucu oluşturulabileceği ve bu veri tabanının onun sorumluluğunda olabileceği belirtilmiştir.

Bu amaçla bu çalışmada tek bir bloklama sunucusu değil, iki sunucu tipi oluşturulmuştur. Sunucuların bir birbirleri arasında haberleşmesi amacı ile de Saldırı Bloklama Protokolü (SBP) adında bir protokolü geliştirilmiştir. Oluşturulan USBS sisteminde görev yapan sunucular şu şekildedir:

1. Saldırı Bloklama Sistemi (SBS) Sunucusu: Sisteme dahil olan her ağda kurulacak olan bu sunucu, saldırı bloklama talebini, saldırı kontrolünü, saldırı bloklanmasını ve saldırı bloğunun kaldırılmasını gerçekleştirir.

2. Ulusal Siber Güvenlik Merkezi (USGM) Sunucusu: Hangi SBS'nin, hangi IP bloğunu yönettiği ile ilgili veri tabanını tutar. Bu sayede her bir SBS'nin kendisi üzerinde böyle bir veri tabanı oluşturma zorunluluğu ortadan kalkar. SBS'lerin merkezi veri tabanından bilgi almak istemeleri durumunda USGM'ye sormaları yeterli olacaktır. Tek bir noktada bulunacak bu ulusal ağ veri tabanının yönetimi ve güncel tutulması da daha kolay olacaktır. Türkiye'de küçük büyük tüm ISP'ler zaten bu bilgileri gerçek zamanlı olarak Bilgi Teknolojileri Kurumu (BTK) ile paylaşmaktadır. Bu bilgiler çerçevesinde BTK tüm kurumsal ağlarının IP adresi bloklarına ve kullanan kurum hakkında bilgiye sahiptir. Bu veri tabanının bu sistem ile kullanılabilmesi için gereken tek şey bu bilgilere birde ilgili kurumda kurulacak olacak SBS sunucusunun IP adresini eklemektir. USGM sunucusu ayrıca gerekmesi durumunda ulusal olarak emir verip tüm kurumlarda belirlediği hedef IP adresinin bloklanmasını da sağlar. Bu sayede geniş çaplı bir atağın engellenmesi sağlanabilir.

Bu sunucuların haberleşmesinde kullanılan SBP protokolü bir yedinci katman protokolü olup, dördüncü katmanda TCP protokolünü kullanmaktadır. Saldırı bloklama sistemi sunucuları birbirleri ile haberleşirken 9090 portunu, yine bu sunucular ulusal siber güvenlik merkezi sunucusu ile haberleşirken ise 9191 portu kullanmaktadır. Protokolün sahip olduğu iki tip paket vardır. Bu paketler:

1- Emir paketleri (Şekil 3.6)

2- Cevap paketleri (Şekil 3.7)

Atak kaynak IP adresi
Atak hedef IP adresi
Atak yapılan protokol numarası
Kaynak port numarası / ICMP ise ICMP tip numarası

Hedef port numarası / ICMP ise ICMP kod numarası
Emir kod numarası
Cevap kodu veya sorumlu SBS IP adresi

Şekil 3.6 Saldırı bloklama protokolü talep paket yapısı.

Cevap kodu veya sorumlu SBS IP adresi

Şekil 3.7 Saldırı bloklama protokolü cevap paket yapısı.

Emir paketlerinin içerisinde atak kaynak IP adresi, atak hedef IP adresi gibi atak detayları bulunduğu gibi emir kodu da bulunmaktadır. Emir kodları, talep edilen işleme göre sayısal bir değer almaktadır. Sistemde toplam 4 adet emir kullanılmaktadır. Bu emirler ve kısa açıklamalarını Çizelge 3.1'de bulabilirsiniz. Cevap paketinde ise cevap kodu (CK) numarası bulunur. Toplamda 8 adet cevap kodu bulunmaktadır ve Çizelge 3.2'de bu kodları ve kısa açıklamaları mevcuttur. Emir ve cevap kodlarının kullanım detayları 4. bölümde bulunmaktadır.

Çizelge 3.1 Emir kodları tablosu.

Emir Kodları
[EK:0] : USGM'den belirtilen IP adresinin sorumlusu SBS'nin IP adresi talep edilir.
[EK:1] : SBS saldırı kaynak IP adresi için bloklama talebi yapar.

[EK:2]: USGM saldırı kaynak IP adresi için ulusal bloklama talebi yapar.
[EK:3]: USGM saldırı kaynak IP adresi için ulusal bloklama kaldırma talebi yapar.

Çizelge 3.2 Cevap kodları tablosu.

Cevap Kodlar
[CK:10] : Bloklama talebi onaylandı.
[CK 11] : USGM'de sorulan IP'ne ait kayıt bulunmamaktadır.
[CK 12] : Bloklanması talep edilen SBS'den, ilgili IP adresine sahip olmadıkları bilgisi döndü.
[CK 13] : USGM, Saldırı hedef IP adresinin bloklama talebi yapan SBS sunucusuna ait olmadığı bilgisi döndü.
[CK 14] : USGM, ulusal ağ veri tabanına ulaşamadığını belirten bir cevap döndü.
[CK 15] : USGM'ye ulusal bloklama yetkisi olmadığı bilgisi döndü.
[CK 16] : SBS'nin, USGM'den sorgu yapma yetkisi olmadığı bilgisi döndü.
[CK 20] : Bu mesaj ile sorumlu SBS'nin IP adresi geri döndü.

3.3 Sistemin Operasyon Detayları

3.3.1 Saldırı Bloklama Sistemi

Bu sistem 2 adet sunucu tipi ile çalışmaktadır. İlk sunucu olan saldırı bloklama sistemi (SBS) sunucusu tüm USBS altyapısına dahil olan kurumlara kurulacaktır. Bu

sunucunun başlıca amacı kendi ağına gelen saldırıları, saldırı kaynağı kuruma iletmek ve atağın kaynağa en yakın noktada sonlandırılmasını sağlamaktır. Bunun sağlanması için kullanılan haberleşme tipleri şu şekildedir:

Haberleşme tipi 1: Saldırı Bloklama Protokolü

Haberleşme tipi 2: YTA Kontrolcüsü ile REST API haberleşmesi

Haberleşme tipi 3: Openflow protokolü

Bu sürecin başlaması ilk olarak SBS1 sunucusuna atak bilgisinin gelmesi ile gerçekleşmektedir (Şekil 3.8, Adım 1). Bu bilgilendirme (Bal kovanı veya STS gibi) harici bir kaynaktan gelecek ise sisteme eklenebilecek bir REST API modülü ile bu işlem gerçekleştirilebilir. Ya da bu çalışma dahilinde olan kurumun bilişim yönetimi tarafından doğrudan arayüz üzerinde girilmesi şeklinde olur.

İkinci adımda, atak bilgisini alan sunucu, atak kaynağının sorumlusu olan SBS2 sunucusunun IP adresini belirlemek amacı ile USGM sunucusuna SBP protokolü (Haberleşme tipi 1) ile sorgulama yapar (Şekil 3.8, Adım 2). Sorguya olumlu cevap dönülmesi durumunda sistem bir sonraki adıma geçer.

Üçüncü adımda SBS1 sunucusu (atak alan sistem) atağın bloklaması için SBS2 sunucusuna SBP protokolü ile talepte bulunur (Şekil 3.8, adım 3). SBS2 sunucusu atak talebinde bulunan sunucunun talep ettiği bloklanacak trafiğin hedef IP adresinin sorumlusu olup olmadığını belirlemek amacı ile USGM sunucusuna sorguda bulunur. Bu amaçla yine SBP protokolü ile saldırı hedef IP adresinin sorumlu SBS IP adresi sorgulanır (Şekil 3.8, adım 4). Güvenlik amaçlı ile yapılan bu kontrol süresince adım 3'de SBS2'ye gelen talep bekletilir. USGM sunucusundan gelen bilgiye göre yine adım 3 çerçevesinde SBS1 sunucusuna olumlu ya da olumsuz geri mesaj döndürülür (Şekil 3.8, adım 3). Atak bloklanmasını talep eden SBS1 sunucusun talebi onaylanırsa bloklama sürecine geçer.

SBS2 sunucusu öncelikle YTA kontrolcüsü ile REST API (haberleşme tipi 2) ile haberleşerek, böyle bir atak var mı yok mu kontrol eder (Şekil 3.8, adım 5). Eğer gerçekten böyle bir atak var ise yine REST API ile bloklama emrini verir. Ayrıca YTA sunucusu atağı blokladıktan sonra belirli sürede bir atağın devam edip etmediğini sorgulayarak eğer atak sonlandı ise bloklama işlemeni kaldırmasını YTA kontrolcüye iletir.

YTA kontrolcüsü gerek atak kontrolü gerçekleştirmek, gerek atak bloklamasını yapmak gerekse atak bloklamasını kaldırma amacı ile kurumun ağ cihazları ile Openflow protokolünü (Haberleşme tipi 3'ü) kullanarak haberleşir.(Şekil 3.8, adım 6)

Ancak USBS sistemine dahil olan kurumlar illa YTA destekleyen ağlar olmak zorunda değildir. Bu durumda atak uyarısı gelen kurumun yöneticileri kendi imkanları çerçevesinde atak kontrolünü yapabilir ve atağı bloklayabilirler. (Şekil 3.9a) Gerek YTA destekli bir ağ olsun, gerekse YTA desteklemeyen ağ olsun bu sistem sayesinde kendisinde gerçekleşen saldırılar hakkında bilgi sahibi olup, saldırı kaynağını sorgulama şansına sahip olabilirler. Bu sayede saldırıya sebep olan kötü niyetli bir kişi ise onun tespitini, eğer saldırı bir zararlı yazılım ile gerçekleştiriliyorsa enfekte olmuş makinanın tespitini ve temizlenmesi sağlayabilirler. YTA desteklemeyen bir ağ alt yapısı olmasına rağmen bu sistemin bloklama özelliklerinin kullanılması istenirse sistemin girişine konumlanacak bir YTA destekli anahtar eklenmesi yeterli olacaktır. Atak sistem içerisinde tam kaynağında kesilmese bile ağa girişi çıkış noktalarında kesme olanağı sağlayacaktır (Şekil 3.9b).

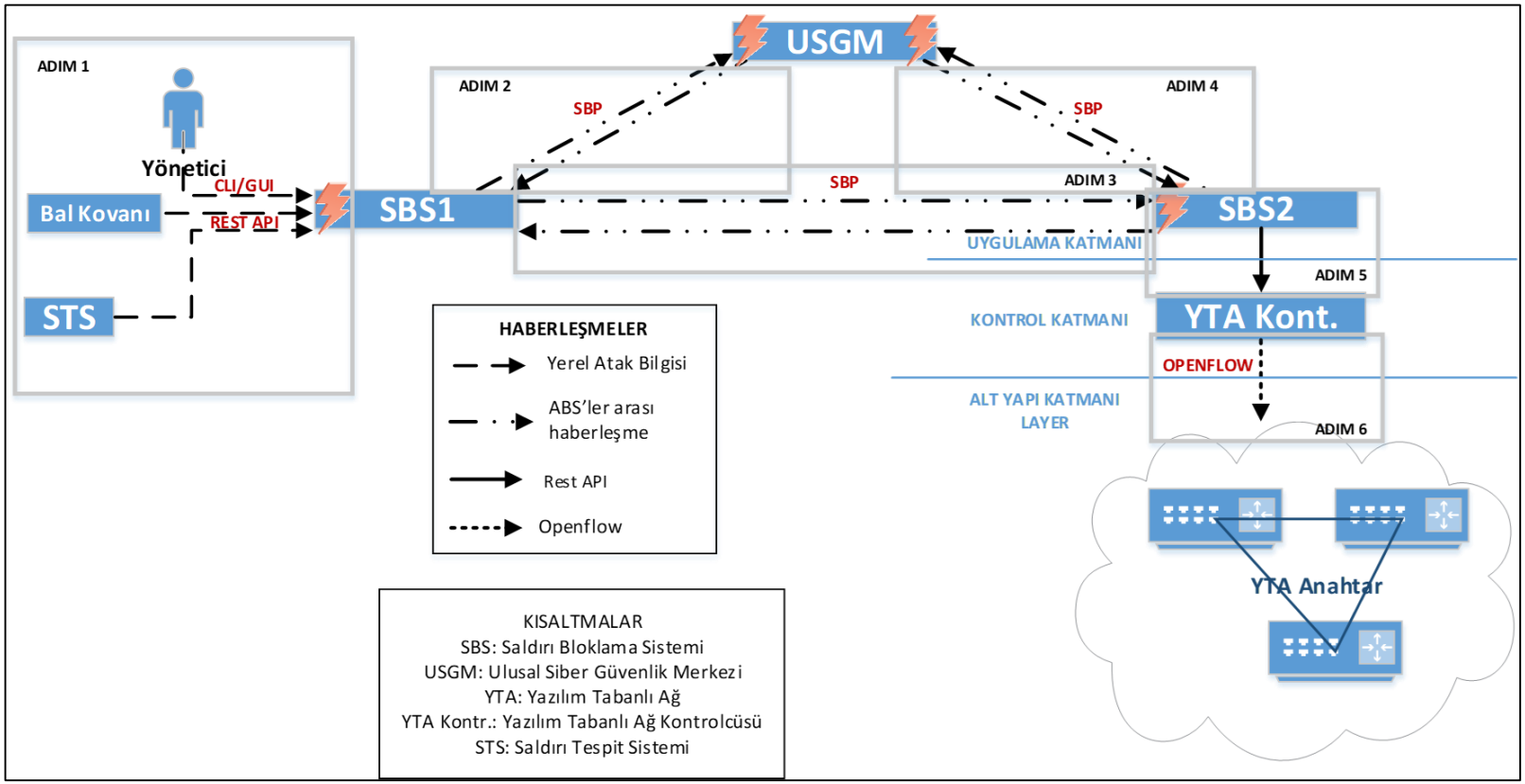
3.3.2 Ulusal Saldırı Bloklama Sistemi

Ulusal saldırı bloklama sisteminin diğer bir kullanılma amacı, USGM tarafından belirlenen bir hedef IP adresinin tüm ülkedeki ağlarda bloklanması yani yasaklanmasının sağlanmasıdır. Süreç ilk olarak USGM'deki yöneticilerin sisteme arayüzden bloklanmasını istedikleri IP adresini girmesi ile başlar (Şekil 3.10, adım 1).

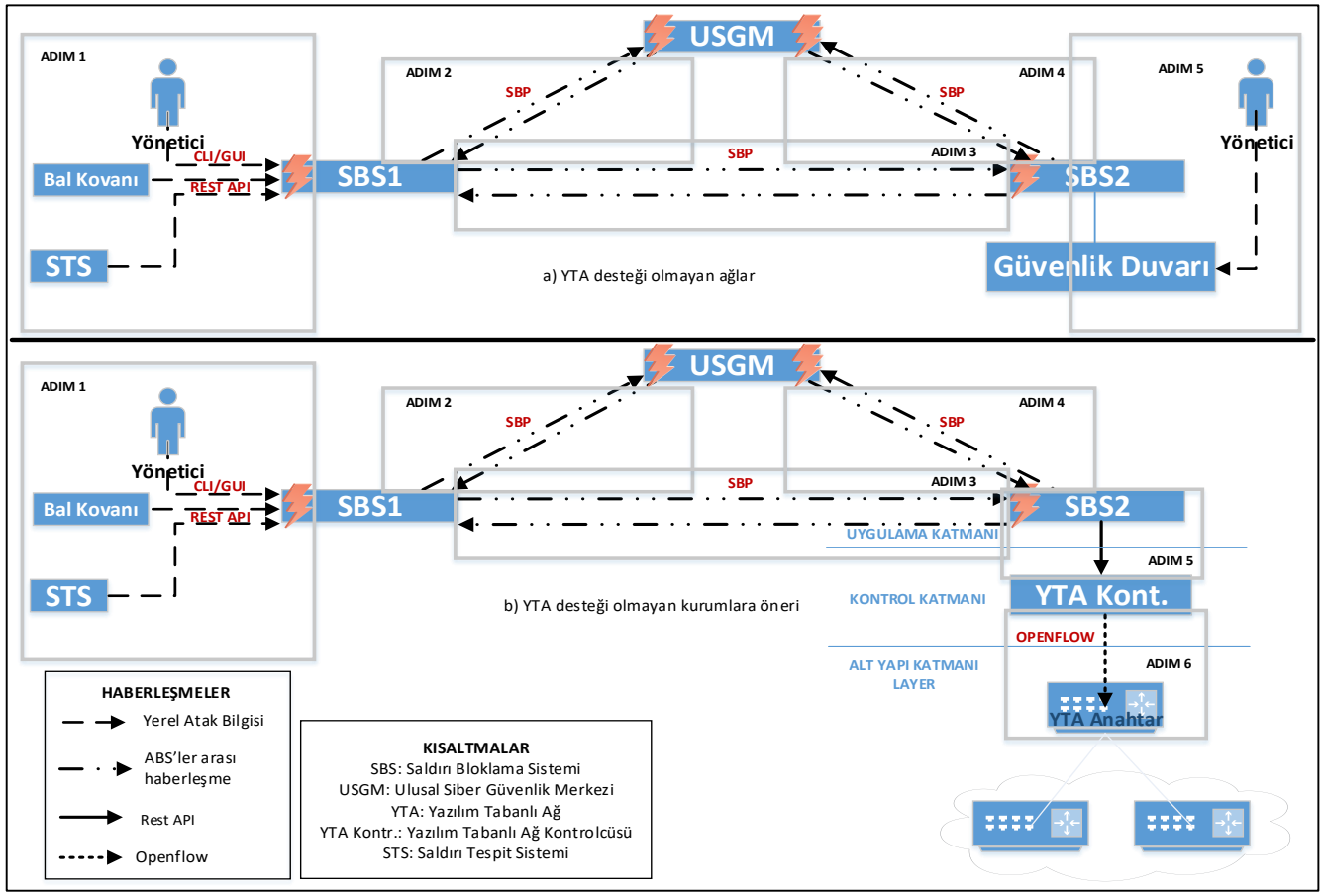
Bloklanacak hedef IP adresi belirtildikten sonra USGM sunucusu SBP protokolünü kullanarak bu emiri tüm SBS sunucularına iletir (Şekil 3.10, adım 2). SBS sunucuları emirin doğru kaynaktan gelip gelmediğini kontrol eder ve kontrol sonucunda USGM'ye olumlu veya olumsuz cevap dönerler.

Eğer SBS sunucusu, emri onaylarsa ve YTA desteğine sahip ise YTA kontrolcüsü ile iletişime geçip, hiçbir kontrol yapmadan bloklama emirini uygulamaya geçer (Şekil 3.10, adım 3). YTA kontrolcü de emiri Openflow protokolünü kullanarak ilgili ağ cihazına ulaştırıp, trafiğin bloklanmasını sağlarlar (Şekil 3.10, adım 4).

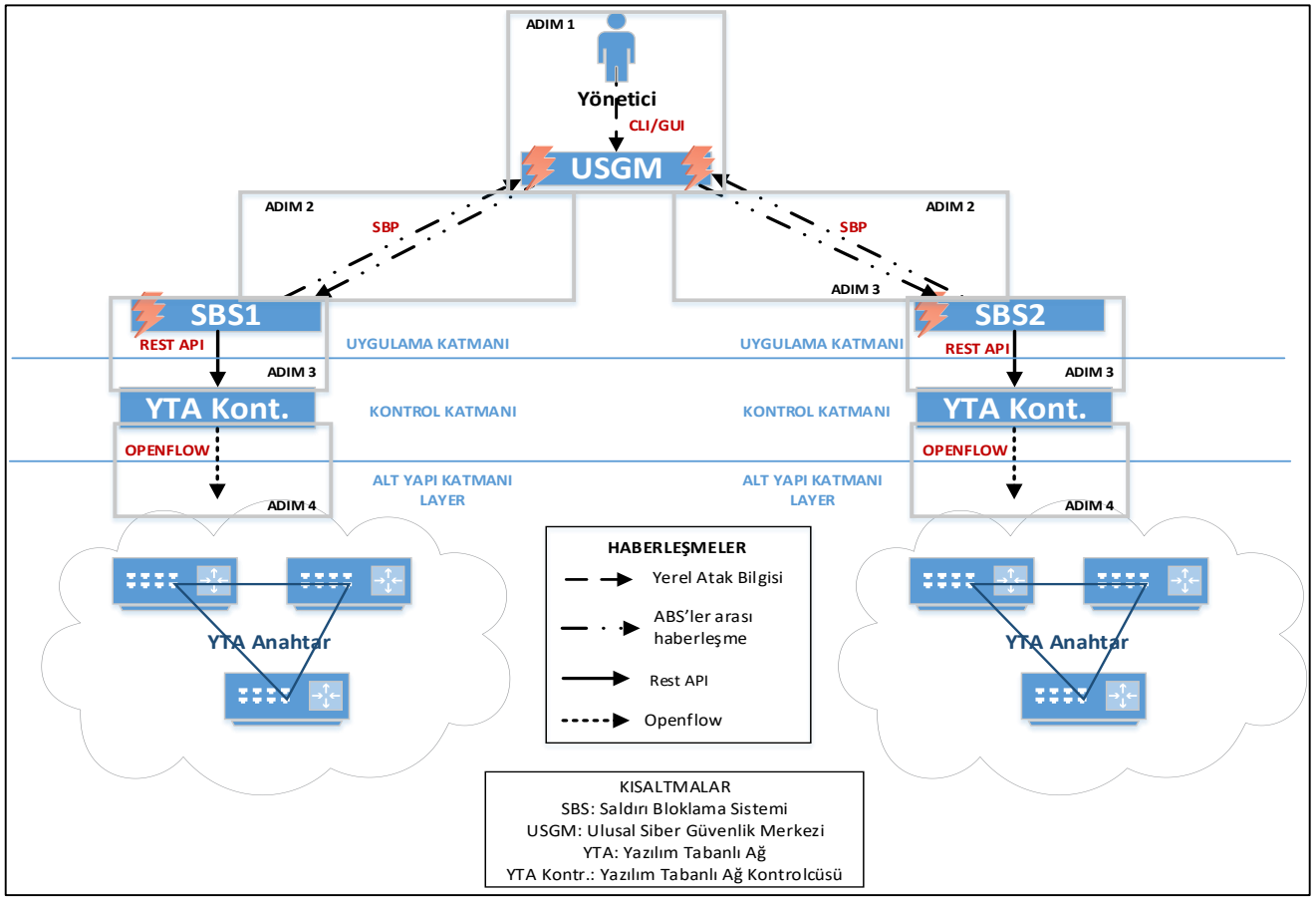
Bloklama sonrası USGM sunucusu bloklamanın kaldırılmasını talep edebilir. Bu talep sonucunda yine aynı süreç ile tüm SBS sunucularında bloklama kaldırılır. İlgili k urumun ağı YTA desteğine sahip değil ise bu işlemleri elle gerçekleştirebilir ya da yukarıda da belirttiğimiz gibi sistemin girişine YTA destekli bir anahtar ekleyerek ilgili bloklama işlemini gerçekleştirebilir.(Şekil 3.9b)



Şekil 3.8 Saldırı bloklama sistemi mimarisi.



Şekil 3.9 a)YTA desteği olmayan ağlarda kullanım b) YTA desteği olmayan ağlarda tavsiye edilen kullanım.



Şekil 3.10 Ulusal Saldırı Bloklama Sistemi mimarisi.

BÖLÜM 4

USBS SİSTEMİNİN DİZAYN DETAYLARI

Python programlama dile ile geliştirilmiş olan Ulusal Siber Güvenlik Sistemi 2 adet ana programdan oluşmaktadır. İki yazılımda da çoklu iş parçacığı tekniği kullanılmıştır. Bu iki program şunlardır:

1. Saldırı Bloklama Sistemi (SBS) Sunucusu
2. Ulusal Siber Güvenlik Merkezi (USGM) Sunucusu

4.1. Saldırı Bloklama Sistemi (SBS) Detayları

SBS yazılımı 3 adet çoklu iş parçacığından oluşmaktadır. (Şekil 4.1)

Bunlar:

1- Ana iş parçacığı (SBS istemci kısmı)

Bu iş parçacı yazılımın genel bir döngü dahilinde çalışmasını sağlamaktadır. Bu iş parçacığının amaçları, SBS sistemini ayağa kaldırmak yani tüm diğer iş birimlerini aktive etmek ve saldırı oluştuğu anda saldırı bilgilerinin sisteme girilmesidir.

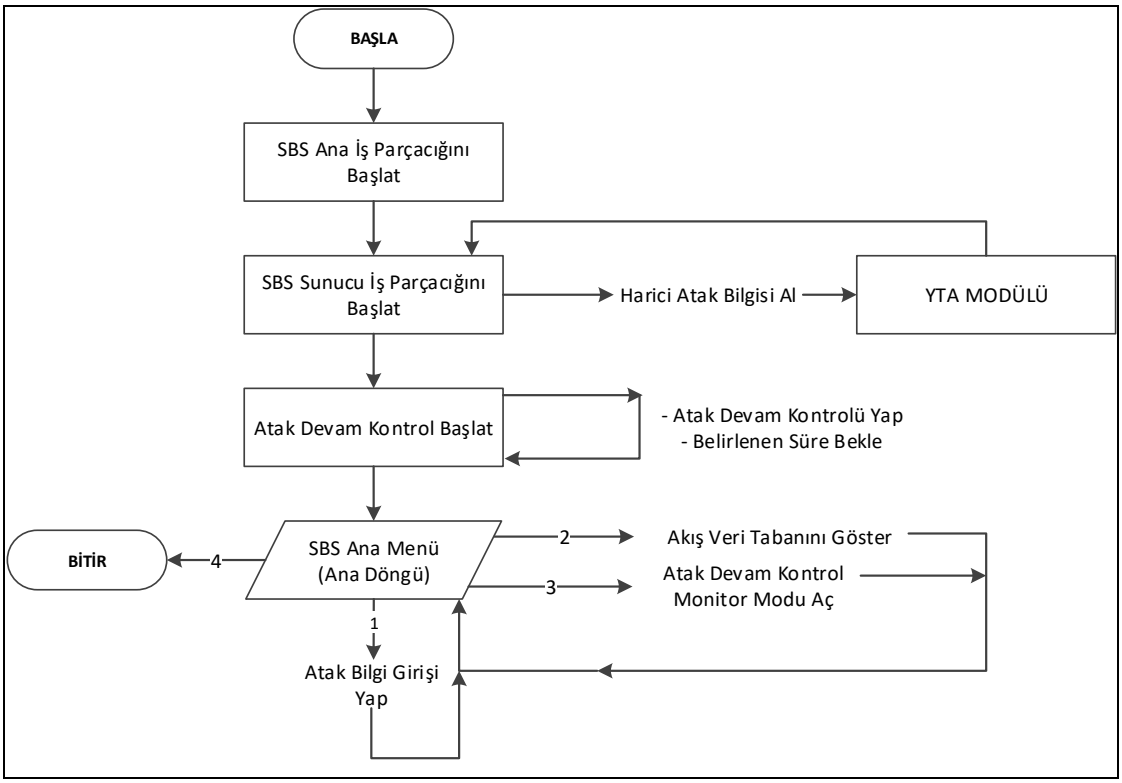
2- Sunucu iş parçacığı (SBS sunucu kısmı)

Bu iş parçacı USGM veya diğer SBS sunucularından gelen talepleri karşılamak amaçlı genel bir döngü dahilinde çalışmaktadır.

3- Atak kontrol iş parçacığı (SBS'nin atak devamını kontrol eden kısım)

Bu iş parçacığının amacı belirlenmiş sürelerde bir atak devam kontrolünü yapmaktadır. Bu süre çalışma dahilinde 60 saniye seçilmiştir.

SBS sunucusunun iş parçacıkları çalışma süreci detayına Şekil 4.1'den ulaşabilirsiniz.



Şekil 4.1 SBS sunucusu iş parçacıkları çalışma süreci.

SBS sunucusu dahilde kullanılan hazır modüllere Çizelge 4.1'de, oluşturulup kullanılan modülleri de Çizelge 4.2'de bulabilirsiniz.

Çizelge 4.1 Kullanılan hazır modüller.

"socket" Modülü: Bu modül ile programın TCP/IP sunucu ve istemci olarak çalışması sağlanmaktadır.

“threading” Modülü: Bu modül ile birden fazla iş parçacığının çalıştırılabilmesi sağlanmaktadır. “os” Modülü: Bu modül ile programın yönetimini sağlayan komut satırı arayüzünün şekillendirilmesi sağlanmıştır. “time” Modülü: Bu modül ile istenmesi durumunda programın belirtilen süre kadar bekletilmesi sağlanmaktadır. “ipaddress” Modülü: Bu modül ile sisteme girilen IP adresi detaylarının kontrolü sağlanmaktadır. “requests” Modülü: REST API ile YTA sunucusun haberleşmesi amacı ile kullanılmıştır. “re” Modülü: Düzenli ifadeler (Regular Expressions) modülü, metinlerin içinden istenilen ifadenin çekilmesi amaçlı kullanılmıştır.

Çizelge 4.2 Oluşturulan modüller.

“SBS_giris” Modülü: Bu modül dahilinde “SBSgirisyaz” fonksiyonu bulunmaktadır. Bu fonksiyon ile SBS sunucunun ana menüsünün gösterilmesi sağlanır. “SBS_atak_giris” Modülü: Bu modül dahilinde “ekle” fonksiyon bulunmaktadır. Bu fonksiyon ile tespit edilmiş bir atağın detayları sisteme girilir. “SBS_kaynak_kontrol” Modülü: Bu modül dahilinde “atak_ip_kontrol” fonksiyonu bulunmaktadır. Bu fonksiyon ile girilmiş olan atak kaynak IP adresinin kendi ağında olup olmadığı kontrol edilir. “loglama” Modülü: Bu modül dahilinde sunucuya erişim yapan istemcilerin IP adresi ve port numaraları “network_baglantileri_SUNUCUIPADRESİ.txt” dosyasında loglanır. “socket_istemci” Modülü: Bu modül dahilinde “veri_yollama” fonksiyonu bulunur. Bu modül diğer SBS’ler ve USGM sunucusu ile haberleşmek amacı ile kullanılır. “YTA” Modülü: Bu modül “atakekle” fonksiyonuna sahiptir ve bir atak bilgisi sisteme eklendiğin de ilgili tüm kontrol ve bloklama sürecini yönetir. Bu modül aynı zamanda aşağıda belirtilmiş olan diğer YTA modüllerini aktif olarak kullanır.

“YTA _switch_bulma” Modülü: “switchbul” fonksiyonuna sahip bu modül atak bilgisi geldiğinde, atak kaynak IP adresinin hangi anahtar cihazında olduğunu tespit eder. “YTA_flow_ekle” Modülü: Bu modül “put_drop_flow” ve “put_normal_flow” fonksiyonlarına sahipdir. “put_drop_flow” fonksiyonu, belirtilen akış detaylarına uyan trafiği izlemek amacı ile izin verilecek şekilde ilgili anahtara akış olarak ekler. “put_drop_flow” fonksiyonu ise belirtilen akış detaylarına sahip trafiğin bloklanmasını sağlayacak şekilde de anahtar cihaza akış ekler. “YTA_istatistik_alma” Modülü: “get_stat” fonksiyonuna sahip bu modül, daha önce “normal” veya “drop olarak eklenmiş akış bilgisine eşleşen trafik istatistik bilgisini alır. “YTA_flow_silme” Modülü: “delete_flow” fonksiyonuna sahip olan bu modül, talep edilen akışın anahtar cihazdan silinmesini sağlar. “YTA_atak_kontrol” Modülü: Bu modül “atak_devam_kontrol” isimli fonksiyona sahiptir. Atak kontrol iş parçacığı tarafından kullanılan bu modül belirtilen sıklıklarla daha önce bloklaması yapılmış atağın devam edip etmediğini kontrol eder.

SBS sunucusu ayrıca kendisine gelen saldırı bilgilerini daha sonra saldırı devam kontrolü yapabilmek adına oluşturulan bir akış kontrol veri tabanında tutmaktadır. Tek tablodan oluşan bu veri tabanı hafızada bulunacak şekilde yapılandırılmıştır. Bu tabloda bulunan veriler sırası ile şu şekildedir:

Akış durumu: Eklenmiş olan akışın güncel durumunu tutmaktadır. Akış durumların açıklamaları Çizelge 4.3’de bulmaktadır.

Çizelge 4.3 Akış durumları.

Akış durumu 1: Atak bloklaması akış kuralı şu anda aktiftir. Akış durumu 2: Bloklaması talep edilen kaynak IP adresi talep anında aktif bir IP adresi değildir. Akış durumu 3: Bloklanaması talep edilen kaynak IP adresi talep anında bahsi geçen trafiği yapmamaktadır.

Akış durumu 0: Atak trafiği durmuş ve sistem tarafından bloklama kaldırılmıştır.

Akış veri tabanında ayrıca aşağıdaki bilgiler bulunur.

Anahtar No: Atak kaynak IP adresinin tespit edildiği anahtar numarasıdır.

Eşleşen Paket Sayısı: Atak trafiğine eşleşen toplam paket sayısı, bu sayı artmaya devam ediyorsa atağın devam ettiğini, artışı durduğunda ise atağın durduğu tespit edilebilmektedir.

HIP: Atak Hedef IP adresidir.

KIP: Atak Kaynak IP adresidir.

ProtNo: Atak protokol numarasını gösterir. (Çizelge 4.4)

Çizelge 4.4 Yaygın kullanılan protokol numaraları.

Protokol Adı:	Numarası
ICMP (Internet Control Message Protocol)	1
IGMP (Internet Group Management Protocol)	2
TCP (Transmission Control Protocol)	6
UDP (User Datagram Protocol)	17
GRE (Generic Routing Encapsulation)	47
OSPF (Open Shortest Path First)	89

ICMP Tip Numarası: ICMP protokolü için yasaklama yapılır ise ICMP mesaj tip numarası (Çizelge 4.5), eğer ICMP atağı söz konusu değil veya tip numarası belirtilmek istenmiyor ise -1 olarak atanır.

ICMP Kod Numarası: ICMP protokolü için yasaklama yapılır ise ICMP kod numarası (Çizelge 4.5), eğer ICMP atağı söz konusu değil veya kod numarası belirtilmek istenmiyor ise -1 olarak atanır.

Çizelge 4.5 Yaygın kullanılan ICMP tip ve kod numaraları.

ICMP Tip No	ICMP Kod No	Açıklama
0	0	Ping (Eko) cevap mesajı
3	0	Hedef ağa ulaşılamıyor mesajı
3	1	Hedef cihaza ulaşılamıyor mesajı
3	7	Hedef protokole ulaşılamıyor mesajı
8	0	Ping (Eko) mesajı
11	0	TTL(Time to Live) değeri sonlandı

Kaynak Port Numarası: Atak kaynak port numarası (Çizelge 4.6), eğer TCP/UDP atağı söz konusu değil ise veya belirtilmek istenmiyor ise bu değer -1 olarak atanır.

Hedef Port Numarası: Atak hedef port numarası (Çizelge 4.6), eğer TCP/UDP atağı söz konusu değil ise veya belirtilmek istenmiyor ise bu değer -1 olarak atanır.

Çizelge 4.6 Yaygın kullanılan protokoller ve port numaraları.

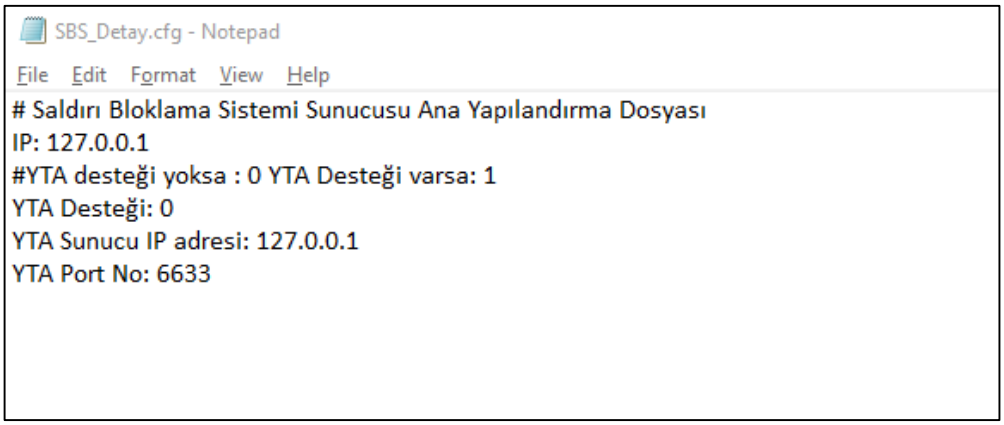
Protokol Adı:	Port Numarası
FTP (File Transfer Protocol)	TCP 20,21

Telnet	TCP 23
SSH (Secure Shell)	TCP 22
SMTP (Simple Mail Transfer Protocol)	TCP 25
DNS (Domain Name System)	TCP,UDP 53
DHCP (Dynamic Host Configuration Protocol)	UDP 67,68
HTTP (Hyper Text Transfer Protocol)	TCP 80
POP3 (Post Office Protocol)	TCP 110
HTTPS (Hyper Text Transfer Protocol Secure)	TCP 443

Atak Bilgisinin Kaynağı: Bu değer eğer USGM'den gelen bir atak bloklama talebi ise USGM değerini alır, kendisi üzerinden tanımlanmış bir atak ise "LOKAL" değerini alır, eğer atak bilgisi harici bir SBS'den gelmişse atak bilgilendirmesini yapan SBS'nin IP adresini alır.

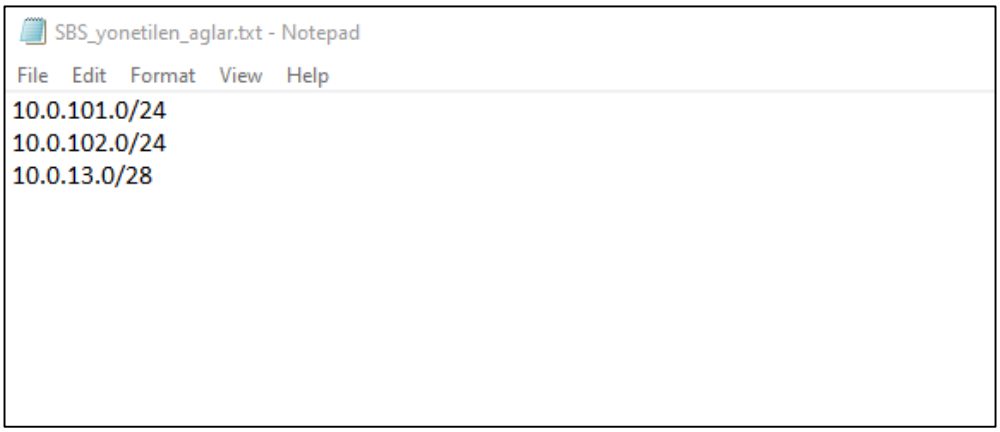
4.1.1 SBS Sunucusunun Yapılandırılması

Sistem başlatılmadan önce gerekli tanımlamaların yapılması gerekmektedir. Öncelikle SBS_Detay.cnf dosyasına sunucusunun ip adresinin ve YTA desteği olup olmadığının belirtilmesi gerekmektedir (Şekil 4.2). Eğer YTA desteği varsa YTA Kontrol sunucusunun IP adresi ve erişim için kullandığı port numarası girilmelidir. Eğer bir değişiklik yapılmazsa varsayılanda SBS sunucusu, loopback arayüzünün IP adresini kullanır ve YTA desteğini yok olarak kabul eder.



Şekil 4.2 SBS sunucusu ana yapılandırma dosyası.

İlk yapılandırma tamamlandıktan sonra ikinci yapılandırılması gereken sunucu dosyası "SBS_yonetilen_aglar.txt" dosyasıdır(Şekil 4.3). Bu dosyaya, SBS sunucusunu kuran kurumun sahip olduğu ağ adresleri, "Ağ adresi/alt ağ maskesi olarak" tanımlaması gerekmektedir. Bu tanımlamalar tamamlandıktan sonra SBS sunucusu başlatılabilir.

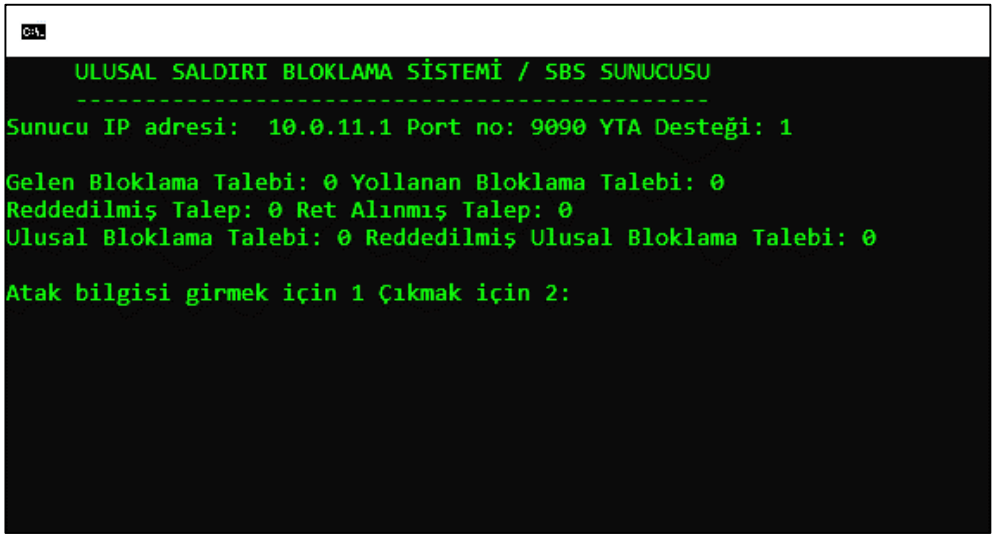


Şekil 4.3 SBS sunucusu yönetilen ağlar dosyası.

4.1.2 SBS Sistemine Atak Bilgisi Girme Süreci

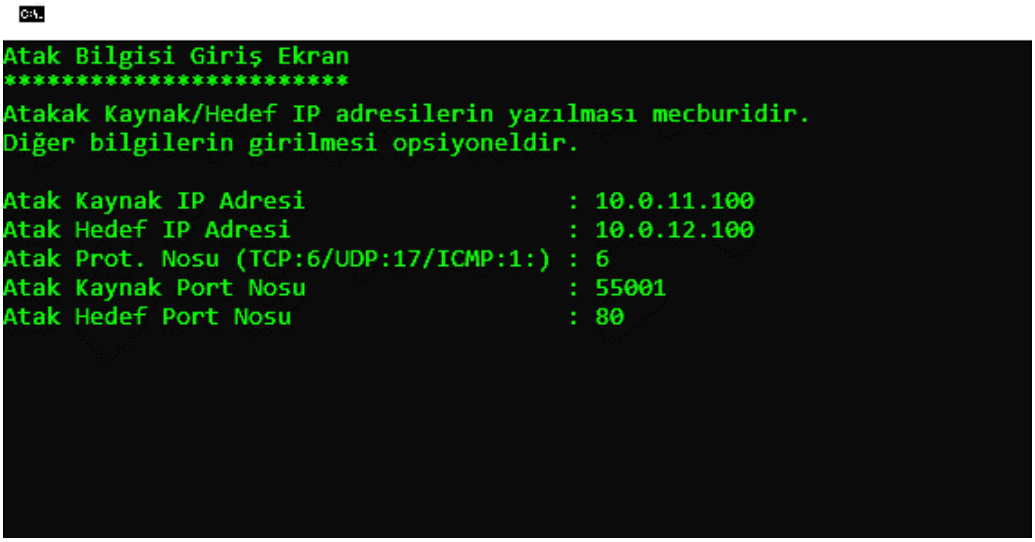
SBS sistemi sadece atak engelleme sistemi olduğu için atak bilgilerinin harici bir kaynaktan sisteme girilmesi gerekmektedir. Çalışma dahilinde bu bilgiler doğrudan

yöneticinin elle sisteme atak bilgilerini girmesi şeklinde yapılandırılmıştır. Bunun için SBS sunucusunun ana ekranında "1" tuşuna basılarak atak bilgisi ekleme ekranına girilir (Şekil 4.4).



Şekil 4.4 SBS sunucusu ana ekranı.

Bu ekranda saldırı kaynak IP adresi/hedef IP adresi, protokol numarası, kaynak/hedef port numarası olarak girilir (Şekil 4.5). Hedef ve kaynak IP adresinin girilmesi mecburidir. Ancak diğer bilgiler girilmeyebilir, yani boş olarak bırakılabilir. Bu durumda YTA sistemi atak kontrolünü ve bloklamayı protokol ve kaynak/hedef port numarasına bakmaksızın sadece kaynak/hedef IP adresine bakarak kontrol eder. Ayrıca protokol numarası 1 olarak girilmesi durumunda yani ICMP protokolü ile yapılan bir atak söz konusu olduğunda, ICMP tip ve ICMP kod numarası sisteme girilir.



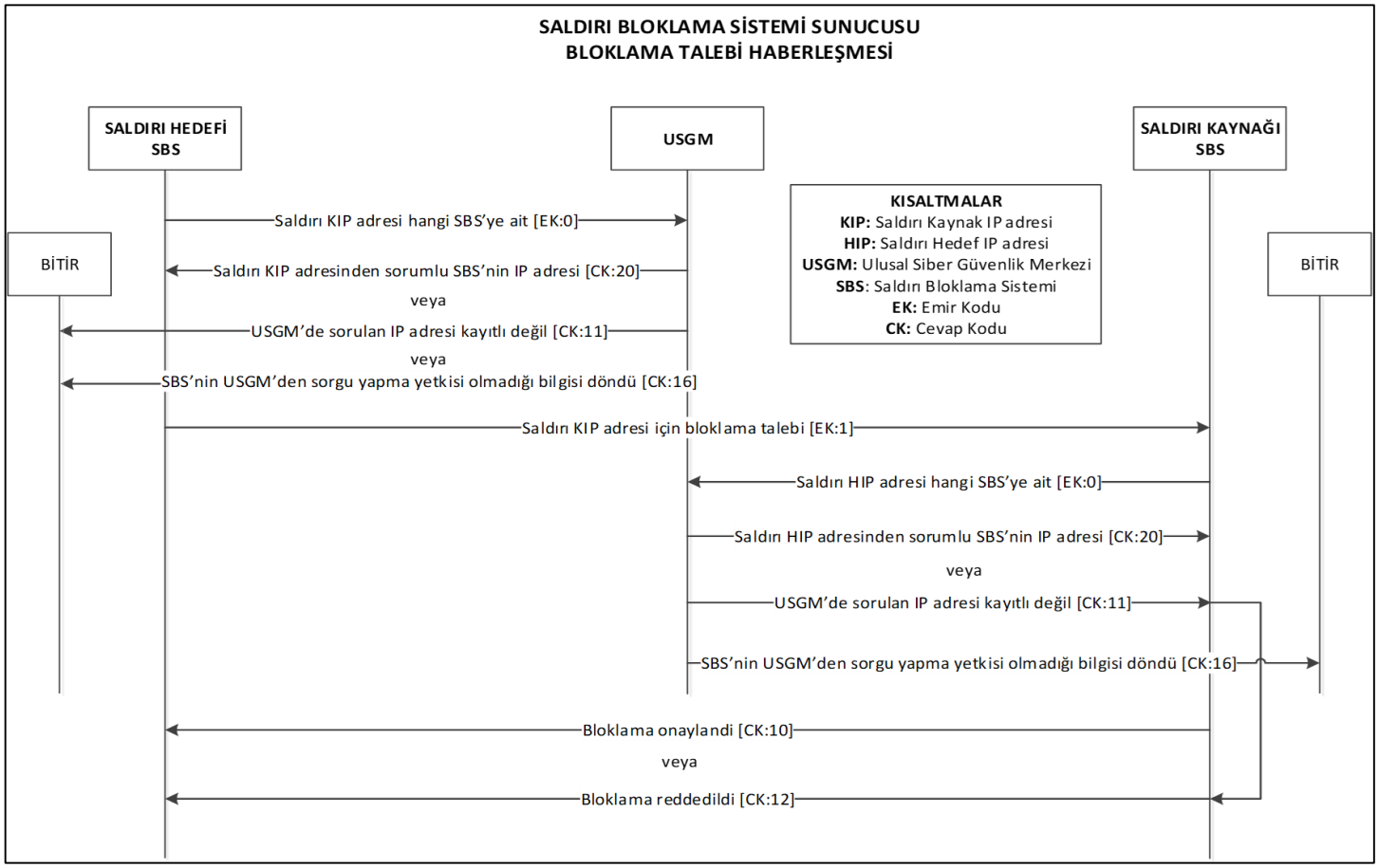
Şekil 4.5 Atak bilgisi giriş ekranı.

4.1.3 SBS Atak Bilgilendirme Süreci

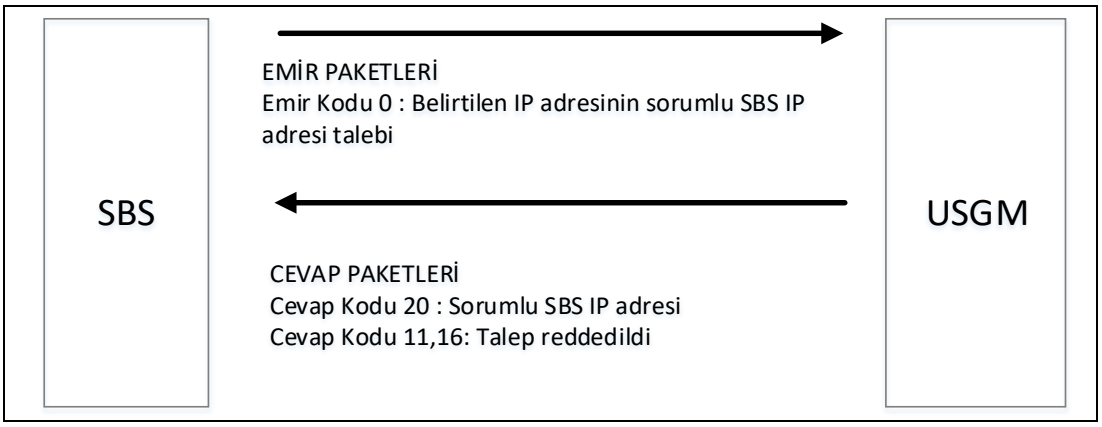
Atak verisi girilmesinin ardından, SBS sunucusu önce atağın kaynak IP adresi kendisine mi ait kontrolünü gerçekleştirir. Bu süreç çerçevesinde yapılan tüm haberleşme detayları Şekil 4.6'da gösterilmektedir.

Eğer IP adresi, sahip olduğu IP blokları içinde ait ise YTA kontrolcüsü aracılığı ile atak trafiğinin halen devam edip etmediğini kontrol mekanizması yardımı ile kontrol eder. Şayet atak trafiği halen daha devam ediyorsa atak trafiği için YTA kontrolcüsü üzerinden bloklama işlemi gerçekleştirir. Bu sürecin detaylarında "Atak Bloklama Süreci" bölümünde bahsedilecektir.

Atak kaynak IP adresi kendisine ait değil ise atak yapan ağa kurulmuş olan SBS sunucusunun IP adresini öğrenmek amacı için emir kodu 0 ile USGM sunucusuna SBP protokolü ile sorguda bulunulur (Şekil 4.7).



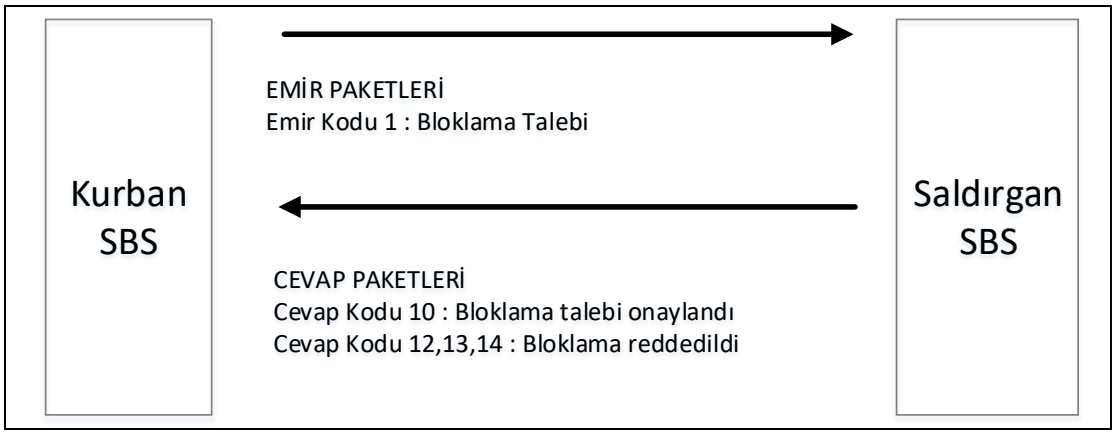
Şekil 4.6 USGM saldırı bloklama süreci.



Şekil 4.7 SBS sunucusun atak kaynak IP adresini öğrenme süreci.

USGM sunucusu böyle bir sorgu geldiği zaman öncelikle sorguyu yapan SBS sunucusunun IP adresinin sisteme dahil olup olmadığını kontrol eder. Bunun amacı, yetkisi olmayan kişilerin sorgu yapmasını engellemektir. Sisteme dahil olamayan bir sunucu ise yetkisi olmadığını belirten cevap kodu 16 olarak geri döner. Eğer yetkisi varsa sorgulanan saldırı kaynak IP adresine ait ağın kendi sisteminde kayıtlı olup olmadığını kontrol eder. Eğer sistemine kayıtlı değil ise cevap kodu 11 olarak yani olumsuz geri dönüş yapar. Eğer kaynak IP adresi kendi bünyesinde tanımlanmış ise onunla ilgili SBS sunucusunun IP adresini, sorgu yapan sunucuya döner.

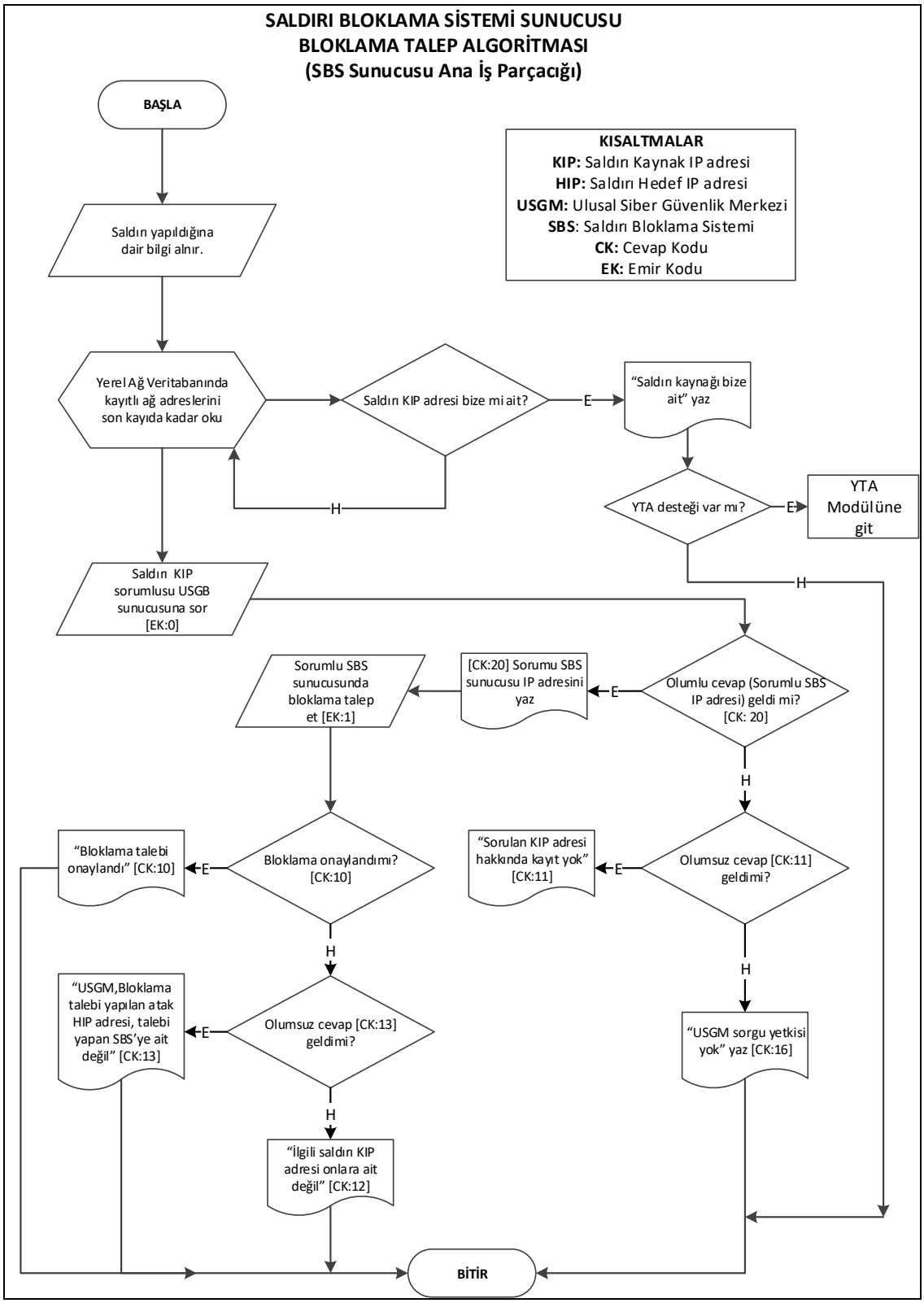
USGM'den ilgili SBS IP adresinin dönülmesi üzerine artık sistem atak kaynağı SBS sunucusu ile iletişime geçer (Şekil 4.8). Bu amaçla hedef SBS sunucusuna emir kodu 1 ile ilgili trafik detaylarını (Saldırı Kaynak IP adresi/Hedef IP adresi, protokolü numarası ve kaynak/hedef port numarası) yollayarak bloklama talebinde bulunur. Saldırgan SBS sunucusu önce saldırı hedef IP adresinin kendisine ait olup olmadığını kontrol eder. Eğer kendisine ait değil ise cevap kodu 12 olarak yani bu IP adresi bana ait değil diye geri dönüş yapar ve bloklama talebini reddeder.



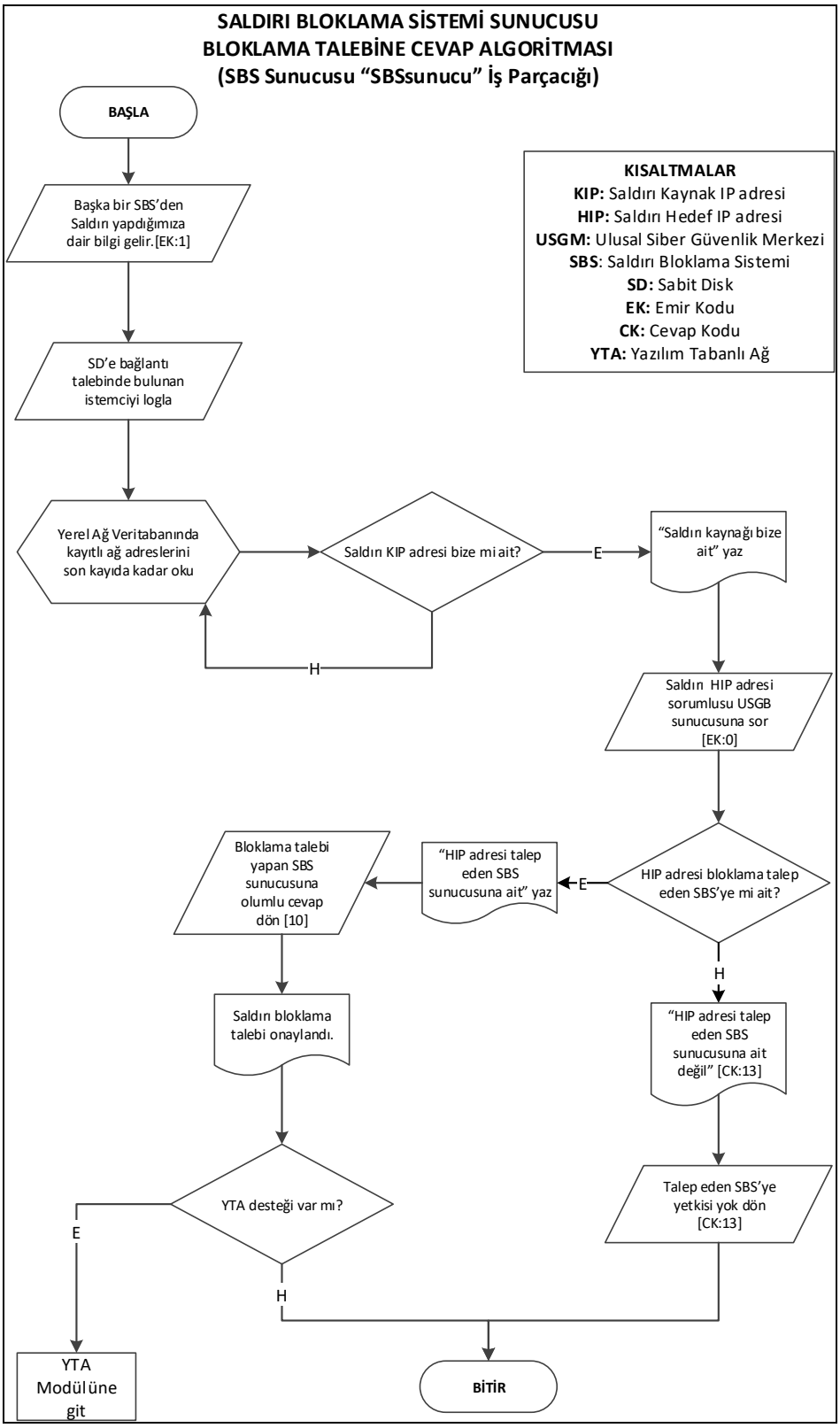
Şekil 4.8 SBS sunucusun atak kaynak IP'nin kontrolünü sağlayan SBS sunucusuna yaptığı bloklama talebi süreci.

Saldırı kaynak IP adresi kendisine ait ise öncelikle sorguda bulunan SBS sunucusunun bloklanmasını istediği hedef IP adresinin kendisine ait mi belirlemek amacı ile USGM sunucusuna sorumlu SBS IP adresini sorar. Bu haberleşme Şekil 4.7'deki süreç ile aynı gerçekleşir. USGM sunucusu gerekli kontrolleri yapıp kendisine sorumlu SBS sunucusunun IP adresini geri döner. Saldırgan durumda olan SBS sunucusu gelen IP adresinin bloklama talebini yapan SBS sunucusu ile aynı IP adresine sahip olup olmadığını kontrol eder. USGM'nin döndüğü IP adresi eğer bloklama talebi olan saldırı hedefi SBS'nin sorumluluğunda değil ise cevap kodu 13 olarak geri dönüş yapar ve bloklama talebini reddeder.

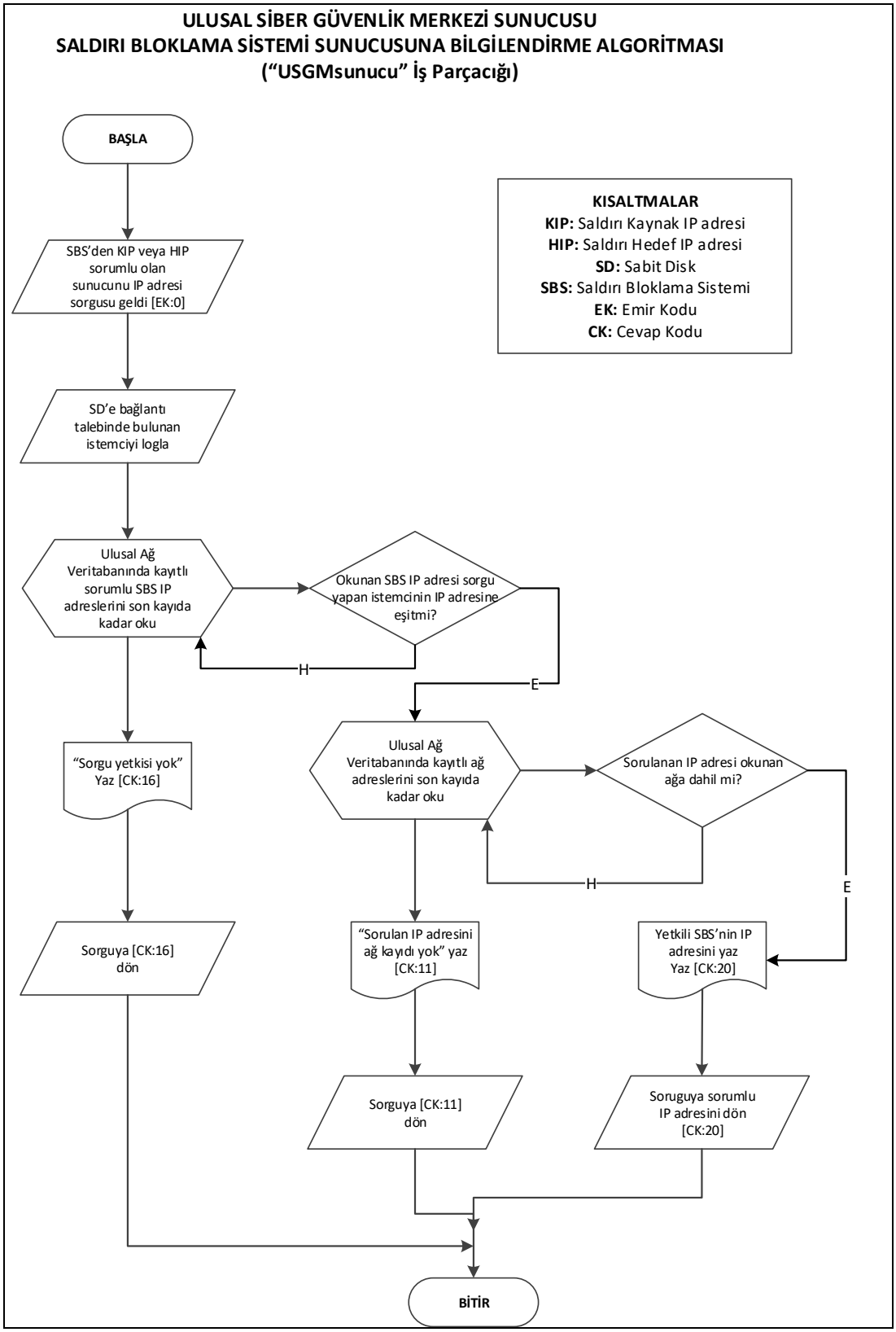
Eğer USGM sunucusundan, olumlu ya da olumsuz bir cevap geri dönülmez ise USGM sunucuna ulaşamadığı için gereken kontrolleri gerçekleştiremediğini ve talebi reddettiğine dair cevap kodu 16 ile geri dönüş yapar. Şayet tüm kontroller olumlu geçerse yani saldırı hedef IP adresi talepte bulunan SBS sunucusunun sorumluluğu altında ise bloklama sürecine geçilir. Bu süreç eğer saldırgan SBS sunucusu YTA desteğine sahip ise otomatik olarak gerçekleşirken, YTA desteğine sahip değil ise kurumun ağ yöneticileri tarafından manuel olarak gerçekleşmek durumunda kalınır. Bu süreç ile ilgili algoritma diyagramlarını Şekil 4.9, Şekil 4.10 ve Şekil 4.11'de bulabilirsiniz.



Şekil 4.9 Saldırı bloklama sistemi sunucusu bloklama talep algoritması.



Şekil 4.10 Saldırı bloklama sistemi sunucusu bloklama talebine cevap algoritması.



Şekil 4.11 USGM sunucusunun SBS sunucusunu bilgilendirme algoritması.

4.1.4 SBS Atak Bloklama Süreci

Önceki süreçte atak kaynak IP adresinin kendi ağında olduğu kesinleşmiş olup, talepte olan SBS'nin de atak talep yetkisi olduğu belirlenmişti. Bu aşamada son adım olan atak bloklanmasını süreci başlatılır. Bu amaçla YTA isimli modül geliştirilmiştir. Bu modül, bir atak emri geldiğinde, sırası ile aşağıda işlemleri gerçekleştirir.

1- Saldırı kaynak IP adresinin hangi anahtar üzerinde bulunduğu tespit edilir.

2- Bahsedilen atak trafiğin halen daha devam edip etmediğini belirlemek amacı ile atak trafiği detayları ile ilgili bir akış, belirlenen anahtara eklenir. Ancak bu ekleme bloklama olarak değil trafik geçsin şeklinde yapılır.

3- Trafik oluşup oluşmadığını anlamak için belirlenen süre kadar beklenir. Bu süre ne kadar uzun olursa saniyedeki paket sayısı az olan yavaş ataklarda tespit edilebilir. Projede bu süre 3 sn olarak atanmıştır. Bu süre kısılabilir ancak bu durumda sadece yüksek hacimli DoS atakları belirlenebilir.

4- Süre sonunda anahtardan eklenen akış ile ilgili istatistik bilgisi talep edilir. Gelen bilgiden eşleşen paket sayısına bakarak halen daha böyle bir trafik olup olmadığı anlaşılır.

5- Şayet eşleşen paket sayısı var ise atağın devam ettiği anlaşılıp bloklama sürecine geçilir.

6- Bloklama süreci tamamlanınca da eklenen akış detayları, SBS sunucusunun hafızasında tutulan akış veri tabanına eklenir.

Tüm bu adımların detayları aşağıdaki bölümlerde incelenmiştir.

4.1.4.1 Saldırı Kaynak IP Adresinin Hangi Anahtarda Olduğu Tespiti

Bu amaçla "YTA_switch_bulma" modülü yazılmıştır. Bu modül YTA Kontrolcü IP adresini ve port numarasını, o sistemde kullanılan ilk anahtar cihazın numarasını ve saldırı kaynak IP adresini haricen alır. REST API ile YTA kontrolcüsüne sorguda bulunabilmek için ayrıca "requests" isimli modülü kullanmaktadır. Aldığı parametreler ile YTA kontrolcüsüne sorguda bulunacağı URL adresini oluşturur. Adresin formatı şu şekildedir:

http://YTA_KONTROLCU_IP:PORT/restconf/operational/opendaylight-inventory:nodes/node/openflow:ANAHTAR_NO

Örneğin kontrolcü IP adresi 10.0.11.200, port numarası 8181 ve anahtar numarası 1 olan anahtar için sorgunda kullanılacak URL aşağıdaki gibidir.

http://10.0.11.200:8181/restconf/operational/opendaylight-inventory:nodes/node/openflow:1

Öncelikle bu URL adresi ve "response" modülü ile "GET" sorgusunda bulunur. Amacı toplamda kaç anahtar cihazının sistem de olduğunu tespit etmektir. Cevap olarak "<Response [200]>" mesajı alması ilgili numaralı anahtarın aktif olduğunu göstermektedir.

Toplamda aktif olan anahtar sayısı belirlendikten sonra yani ikinci aşamada belirlediği bu anahtarların detaylarını öğrenmek için tekrar sorgu yapar. Sorgu sonrasında ilgili anahtar cihazından tüm detayları içeren cevap gelir. Gelen cevap içerisinde saldırı kaynak IP adresinin kaydı olup olmadığı tespit edilir. Eğer varsa anahtar numarası YTA modülüne geri dönülür. Bu durumda, bir sonraki adım olan saldırı olarak bildirilen trafiğin halen daha devam edip etmediğinin kontrol edildiği aşamaya geçilir.

Eğer ilgili atak kaynak IP adresi yok ise bir sonraki anahtar cihazına geçilir. Hiçbir anahtar cihazında ilgili kayıt bulunmuyorsa YTA modülüne "0" değeri dönülür. YTA modülü de bloklama sürecini şu anda bu IP adresi aktif kullanımda değil diye durdur. Oluşturulmuş olan akış veri tabanına akış durumu olarak "3" olarak atanır. Tüm akış durum kodlarına 4.3 numaralı çizelgeden ulaşabilirsiniz.

4.1.4.2 Bloklanması İstenen Saldırı Hakkında Bilgi Amaçlı Akış Eklenmesi

Bu aşamada saldırı kaynak IP adresinin hangi anahtar üzerinde bulunduğu tespit edilmiştir. Ancak bu IP adresinin bir anahtar üzerinde bulunması, atağın bu kaynak IP adresi tarafından devam ettirildiği anlamına gelmez. Atağın devam edip etmediğini anlamak amacı ile tespit edilmiş anahtar cihaza atak detaylarını içeren bir akış bilgisi, trafik devam edecek şekilde işlenir. YTA kontrolcüsünde buna normal mod denir. Bu amaçla "YTA_flow_ekle" modülü geliştirilmiştir. Bu modül 2 fonksiyona sahiptir. Bunlardan ilki, normal modda akış eklenmesi olan "put_normal_flow" fonksiyonudur.

YTA modülü bu aşamada bu fonksiyonu çağırır. Bu fonksiyona eklenmek istenen tüm akış detayları (YTA kontrolcü IP adresi, port numarası, anahtar numarası, tablo numarası, akış numarası, saldırı kaynak IP adresi, saldırı hedef IP adresi, protokol numarası, ICMP tip numarası, ICMP kod numarası, saldırı kaynak port numarası, saldırı hedef port numarası) haricen belirtilir. Bu bilgiler eklenecek akışta kullanılacaktır.

Bu amaçla anahtar cihaza REST API ile ilgili mesaj, PUT mesajı formatında yollanır. Bunun için öncelikle kullanılacak URL adresi oluşturulur. Bu adresin içeresinde YTA kontrolcüsünün IP adresi, port numarası, anahtar numarası, tablo numarası ve akış numarası yer almalıdır. Adresi formatı şu şekildedir:

http://YTA_KONTROLCU_IP:PORT/restconf/operational/opendaylight-inventory:nodes/node/openflow:ANAHTAR_NO/table/TABLO_NUMARASI/flow/AKIŞ_NUMARASI

Örnek bir URL aşağıdaki gibidir:

http://10.0.11.200:8181/restconf/config/opendaylight-inventory:nodes/node/openflow:111/table/0/flow/201

Daha sonra bu URL adresine PUT mesaj tipi ile takip edilmek istenen trafiğin detayları XML formatında yollanır. Şayet bu bir TCP (no 6) veya UDP (no 17) protokolünü kullanan bir atak ise aşağıdaki mesaj yollanır.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<flow xmlns="urn:opendaylight:flow:inventory">
   <instructions>
      <instruction>
         <order>0</order>
      </instruction>
   </instructions>
   <table_id>TABLO NUMARASI</table_id>
   <id>AKIŞ NUMARASI</id>
   <match>
      <ethernet-match>
```

```
        <ethernet-type>
          <type>2048</type>
        </ethernet-type>
      </ethernet-match>
      <ipv4-destination>HEDEF IP ADRESİ/32 </ipv4-destination>
      <ipv4-source>KAYNAK IP ADRESİ/32</ipv4-source>
      <ip-match>
        <ip-protocol>PROTOKOL NUMARASI</ip-protocol>
      </ip-match>
      <tcp-destination-port>HEDEF PORT NUMARASI</tcp-destination-port>
      <tcp-source-port>KAYNAK PORT NUMARASI</tcp-source-port>
   </match>
      </match>
   <hard-timeout>0</hard-timeout>
   <idle-timeout>0</idle-timeout>
   <cookie>AKIŞ NUMARASI</cookie>
   <flow-name>SBS-ADDED-FLOW</flow-name>
   <priority>200</priority>
</flow>
```

Bir atak için yollanan mesajın örneğini aşağıda bulabilirsiniz.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<flow xmlns="urn:opendaylight:flow:inventory">
   <instructions>
      <instruction>
         <order>0</order>
         <apply-actions>
            <action>
               <order>0</order>
               <output-action>
                  <output-node-connector>NORMAL</output-node-connector>
```

```
                </output-action>
            </action>
        </apply-actions>
    </instruction>
</instructions>
<table_id>0</table_id>
<id>202</id>
<match>
    <ethernet-match>
        <ethernet-type>
            <type>2048</type>
        </ethernet-type>
    </ethernet-match>
    <ipv4-destination>10.0.12.1/32 </ipv4-destination>
    <ipv4-source>10.0.11.1/32 </ipv4-source>
    <ip-match>
        <ip-protocol>6</ip-protocol>
    </ip-match>
    <tcp-destination-port>80</tcp-destination-port>
    <tcp-source-port>1500</tcp-source-port>
</match>
<hard-timeout>0</hard-timeout>
<idle-timeout>0</idle-timeout>
<cookie>202</cookie>
<flow-name>SBS-ADDED-FLOW</flow-name>
<priority>200</priority>
</flow>
```

Şayet belirtilen atak da kaynak port numarası belirtilmemişse “<tcp-source-port>KAYNAK PORT NUMARASI</tcp-source-port>” kısmı yollanmaz. Ya da hedef port numarası belirtilmemişse “<tcp-destination-port>HEDEF PORT NUMARASI</tcp-destination-port>” kısmı yollanmaz. Hem hedef hem de kaynak port numarası belirtilmemiş ise iki kısımda REST API ile yollanan mesaj içerisinde yer almaz.

Şayet atak bir ICMP atağı ise bu durumunda aşağıdaki şekilde bir mesaj yollanır. Burada fark olarak “<icmpv4-type>TİP NO</icmpv4-type> ve “<icmpv4-code>KOD NO</icmpv4-code>” ifadeleri yer almaktadır. Şayet talep tüm ICMP trafiğinin tip ve kod numarasına bakmazsınız bloklanması şeklinde ise, ICMP tip ve ICMP kod bilgileri girilmeyebilir. Bu durumda eklenmiş olan bu satırlar yollanacak REST API ifadesi içerisinde yer almayacaktır. ICMP için tip numarası 8 ve kod numarası 0 olan yani Ping eko mesajı için bir XML örneğini aşağıda bulabilirsiniz.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<flow xmlns="urn:opendaylight:flow:inventory">
   <instructions>
      <instruction>
         <order>0</order>
         <apply-actions>
            <action>
               <order>0</order>
               <output-action>
                  <output-node-connector>NORMAL</output-node-connector>
               </output-action>
            </action>
         </apply-actions>
      </instruction>
   </instructions>
   <table_id>0</table_id>
   <id>201</id>
   <match>
      <ethernet-match>
         <ethernet-type>
            <type>2048</type>
         </ethernet-type>
```

```
      </ethernet-match>
      <ipv4-destination>10.0.11.1/32 </ipv4-destination>
      <ipv4-source>10.0.12.1/32 </ipv4-source>
      <ip-match>
         <ip-protocol>1</ip-protocol>
      </ip-match>
      <icmpv4-match>
         <icmpv4-type>8</icmpv4-type>
         <icmpv4-code>0</icmpv4-code>
      </icmpv4-match>
   </match>
   <hard-timeout>0</hard-timeout>
   <idle-timeout>0</idle-timeout>
   <cookie>201</cookie>
   <flow-name>SBS-ADDED-FLOW</flow-name>
   <priority>200</priority>
</flow>
```

Bu bilgiler YTA kontrolcüsü üzerinden atak kaynak IP adresinin yer aldığı anahtar cihaza işlenir. Burada yer alan "<output-node-connector>NORMAL</output-node-connector>" ifadesi ile trafiğin normal şekilde devam etmesi sağlanır. Yani trafik devam edecektir. Burada ki amaç anahtara işlenen bu akış ifadesi sayesinde bu trafik hakkında istatistik elde etmektir.

4.1.4.3 Trafik İstatistiğin Oluşması İçin Belirli Süre Bekleme

Bu aşamada trafik istatistiğinin oluşabilmesi için bir süre beklenir. Eğer tespit edilmesi istenen yüksek hacimli ataklar ise (saniyede binlerce paket gibi) bu süre bir saniye gibi kısa bir süre belirlenebilir. Burada YTA kontrolcüsünün talep edilen işlemleri gerçekleştirme süreside göz önüne alınmalıdır. Şayet daha az yoğunluktaki tekil atakların tespit edilmesi isteniyor ise bu süre daha uzun tutulmalıdır. Bu çalışma dahilde bu süre 3

saniye olarak belirlenmiştir. Bu sayede basit bir ping trafiği bile algılanabilinir. Süre dolduktan sonra bir sonraki adıma geçilir.

4.1.4.4 İlgili Anahtardan Trafik İstatistiklerinin Çekilmesi

Burada bir önceki adımda anahtar cihaza işlenen akış bilgisi sayesinde atak trafiği olarak belirtilen trafik ile ilgili istatistik çekilir. Bu amaçla "YTA_istatistik_alma" modülü geliştirilmiştir. Bu modüle haricen YTA kontrolcü IP adresi, port numarası, anahtar numarası, tablo numarası ve akış numarası bilgileri iletilir. Bu bilgilerin yardımı ile YTA kontrolcüsüne REST API'ı kullanarak aşağıdaki URL ile GET sorgusu gerçekleştirilir.

http://YTA_Kontrolcu_IP_Adresi:Port_No/restconf/operational/opendaylight-inventory:nodes/node/openflow:ANAHTAR_NO/table/TABLO_NO/flow/AKIŞ_NO/opendaylight-flow-statistics:flow-statistics/

Aşağıda örnek URL adresini bulabilirsiniz.

http://10.0.11.200:8181/restconf/operational/opendaylight-inventory:nodes/node/openflow:111/table/0/flow/201/opendaylight-flow-statistics:flow-statistics/

Bu URL ile yapılan sorguya cevap aşağıda gösterilen gibi gelecektir.


```
{
   "opendaylight-flow-statistics:flow-statistics": {
      "duration": {
         "nanosecond": 847000000,
         "second": 20
      },
      "byte-count": 2548,
      "packet-count": 26
   }
}
```

Gelen cevapta ki “packet-count” değeri okunur ve bu YTA modülüne paket sayısı bilgisi olarak dönülür. YTA modülüne şayet sıfır paket verisi dönülür ise buradan tarif edilen trafiğin halen olmadığı anlaşılır ve boşuna bir bloklama işlemi gerçekleştirilemez. Bu durumda YTA modülü akış veri tabanına atak trafik detaylarını ve akış durumunu işler. Atak trafiğinin halen devam etmediğini belirmek amacı ile akış durum kodunu veri tabanına “3” olarak işler ve istatistik çekmek amaçlı oluşturulan akış bilgisini ilgili anahtar cihazından silme işlemini gerçekleştirir. Bunun içinde “YTA_flow_silme” modülü kullanılır. Bu modüle haricen YTA kontrolcü IP adresi, port numarası, anahtar numarası, tablo numarası ve akış numarası bilgileri iletilir. Bu bilgilerin yardımı ile YTA kontrolcüsüne REST API’ı kullanarak aşağıdaki URL ile DELETE emrini yollar.

http://YTA_Kontrolcu_IP_Adresi:Port_No/restconf/operational/opendaylight-inventory:nodes/node/openflow:ANAHTAR_NO/table/TABLO_NO/flow/AKIŞ_NO/opendaylight-flow-statistics:flow-statistics/

Aşağıda örnek URL adresini bulabilirsiniz.

http://10.0.11.200:8181/restconf/operational/opendaylight-inventory:nodes/node/openflow:111/table/0/flow/201/opendaylight-flow-statistics:flow-statistics/

Eğer YTA modülüne dönülen paket sayısı sıfırdan büyük ise trafiğin devam ettiği anlaşılır. Bu durumda son aşama olan trafiğin bloklanması sürecine geçilir.

4.1.4.5 İlgili Atak Trafiğinin Bloklanması

Şayet istatistik dönüşünde, paket sayısının artığı gözlemlendi ise bu aşama başlar. YTA modülü bu amaçla yine “YTA_flow_ekleme” modülünü kullanır ancak bu sefer bu modüldeki “put_drop_flow” fonksiyonu çağırılır. Bu fonksiyona haricen YTA kontrolcü IP adresi, port numarası, anahtar numarası, tablo numarası, akış numarası, saldırı kaynak IP adresi, saldırı hedef IP adresi, protokol numarası, ICMP tip numarası, ICMP kod numarası, saldırı kaynak port numarası ve saldırı hedef port numarası bilgileri verilir. Bu bilgilerin yardımı ile aşağıdaki URL adresi ile PUT mesaj tipi kullanılarak YTA kontrolcüsüne bloklama emri verilir.

http://YTA_KONTROLCU_IP:PORT/restconf/operational/opendaylight-inventory:nodes/node/openflow:ANAHTAR_NO/table/TABLO_NUMARASI/flow/AKIŞ_NUMARASI

Örnek bir URL aşağıdaki gibidir:

http://10.0.11.200:8181/restconf/config/opendaylight-inventory:nodes/node/openflow:111/ table/0/flow/201

Aşağıda örnek verilen PUT mesajında, 10.0.11.200 adresli YTA kontrolcüsün sahip olduğu, 111 numaralı anahtar cihazına 201 numaralı akışı ekleme emri verilmektedir. Bu emir, bloklanacak trafik detaylarını XML formatında iletmektedir. İlgili XML formatı aşağıdaki gibidir.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<flow xmlns="urn:opendaylight:flow:inventory">
   <instructions>
      <instruction>
         <order>0</order>
      </instruction>
   </instructions>
   <table_id>0</table_id>
   <id>202</id>
   <match>
      <ethernet-match>
         <ethernet-type>
            <type>2048</type>
         </ethernet-type>
      </ethernet-match>
      <ipv4-destination>10.0.12.1/32 </ipv4-destination>
      <ipv4-source>10.0.11.1/32 </ipv4-source>
      <ip-match>
         <ip-protocol>6</ip-protocol>
```

```
        </ip-match>
        </match>
    <hard-timeout>0</hard-timeout>
    <idle-timeout>0</idle-timeout>
    <cookie>202</cookie>
    <flow-name>SBS-ADDED-FLOW</flow-name>
    <priority>200</priority>
</flow>
```

Verilen örnekte protokol numarası 6 olarak belirtilmiştir. Yani tüm TCP trafiği için bloklama emri verilmiştir. Şayet atak, kaynak ve hedef port numaraları da belirtilmiş şekilde daha spesifik olarak tanımlanmış ise "<tcp-source-port>KAYNAK PORT NUMARASI</tcp-source-port>" ifadesi ile atak kaynak port numarası ve "<tcp-destination-port>HEDEF PORT NUMARASI</tcp-destination-port> ifadesi ile de atak hedef port numarası ile bloklanacak trafik belirtilebilir. Bu durumda tüm TCP trafiği için değil belirtilen port numaralarına sahip TCP trafiği bloklanır.

ICMP protokolü yani protokol numarası 1 olacak şekilde bloklama işlemi gerçekleştirilecek ise aşağıdaki XML ifadesi YTA kontrolcüye yollanacaktır.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<flow xmlns="urn:opendaylight:flow:inventory">
    <instructions>
        <instruction>
            <order>0</order>
        </instruction>
    </instructions>
    <table_id>0</table_id>
    <id>201</id>
    <match>
        <ethernet-match>
            <ethernet-type>
```

```
            <type>2048</type>
        </ethernet-type>
    </ethernet-match>
    <ipv4-destination>10.0.11.1/32 </ipv4-destination>
    <ipv4-source>10.0.12.1/32 </ipv4-source>
    <ip-match>
        <ip-protocol>1</ip-protocol>
    </ip-match>
    <icmpv4-match>
        <icmpv4-type>0</icmpv4-type>
        <icmpv4-code>8</icmpv4-code>
    </icmpv4-match>
</match>
<hard-timeout>0</hard-timeout>
<idle-timeout>0</idle-timeout>
<cookie>201</cookie>
<flow-name>SBS-ADDED-FLOW</flow-name>
<priority>200</priority>
```

Burada tanımlanmış olan “icmpv4-type”ifadesinin 8 ve “icmpv4-code” ifadesinin de 0 olması, bunun ICMP echo, yani ping aracının ilk paketi olduğunu belirtmektedir. Şayet tüm ICMP trafiğinin bloklanması isteniyor ise bu ifadeler yollanan XML’den çıkartılır.

Burada dikkat edilecek diğer husus ise istatistik amaçlı eklenen akış ifadesinde “<instructions>”(talimatlar) kısmının altında “<output-node-connector>NORMAL </output-node-connector>” ifadesinin bu yollanan XML’de yer almamasıdır. Şayet YTA kontrolcüsüne talimat olarak hiçbir şey belirtilmiş ise kontrolcü bu akışı “drop” yani trafiği düşür olarak işler. Bu sayede atak trafiği durdurulur.

4.1.4.6 Bloklanan Trafik Bilgisinin Akış Veri Tabanına İşlenmesi

Bloklama işlemini tamamlayan "YTA_flow_ekleme" modülü YTA modülüne geri döner ve YTA modülü akışın trafik detayları yer alacak şekilde akış veri tabanına bilgileri işler. Burada akış durum kodu olarak 1 olarak eklemektedir. Akış durum kodu 1, atak bloklama isteği kontrol edildi ve atağın halen devam ettiği tespit edildiğini belirtir. Sonrasında da atak durduruldu anlamını taşımaktadır. Bu aşama ile atak bloklama süreci tamamlanır.

4.1.5 SBS Atak Devam Kontrolü

Bu sistem daha önceden bloklanmış trafiğin, atak sonlandığı zaman bloklanmasının kaldırılmasını da desteklemektedir. Bu süreci "atak_kontrol" isimli iş parçacığı yönetmektedir. Bu iş parçacığı dahilinde kontrol sürecini yönetmek amacı ile "atak_devam_kontrol" fonksiyonu yazılmıştır. Bu fonksiyona, global değişken ataması ile SDNControllerIP, SDNControllerPort, tableNo, flowNo değişkenlerini SBS programından almaktadır.

Fonksiyon belirlenen sürelerde bir atak devam ediyor mu diye kontrol eder. Bu süre proje dahilinde 60 saniye olarak belirlenmiştir. Yani kontrol işlemi 60 saniyede bir yapılabilir. Bu süre daha yüksek tutulabilir. Ancak deneysel amaçlı geliştirilmiştir bu projede bu fonksiyonun çalışma testlerinin çok beklemeden gerçekleştirilebilmesi amacı ile kısa tutulmuştur.

Fonksiyon, öncelikle eklenmiş bir akış var mı diye kontrol etmek amacı ile akış veri tabanını kontrol eder. Şayet yok ise bekleme sürecine geçer. Eğer akış veri tabanında eklenmiş akış varsa tüm eklenmiş akışları tek tek ele alır. Öncelikle akışın durumunu kontrol eder, eğer akış durum kodu 0, 2 veya 3 ise yani akışın şu anda aktif almaması durumunda veri tabanında bir sonraki kayda geçer.

Eğer akış durum kodu 1 ise yani halen daha aktif bloklama yapan bir akış olduğunu görürse, o akış ile ilgili istatistik çeker. Bu amaçla veri tabanından akış numarasını alır ve onla ilgili istatistiği REST API'ı kullanarak aşağıdaki URL ile GET sorgusu gerçekleştirilir.

http://YTA_Kontrolcu_IP_Adresi:Port_No/restconf/operational/opendaylight-inventory:nodes/node/openflow:ANAHTAR_NO/table/TABLO_NO/flow/AKIŞ_NO/opendaylight-flow-statistics:flow-statistics/

Aşağıda örnek URL adresini bulabilirsiniz.

http://10.0.11.200:8181/restconf/operational/opendaylight-inventory:nodes/node/openflow:111/table/0/flow/201/opendaylight-flow-statistics:flow-statistics/

Bu URL ile yapılan sorguya cevap aşağıdaki gibi gelecektir.


```
{
  "opendaylight-flow-statistics:flow-statistics": {
    "duration": {
      "nanosecond": 847000000,
      "second": 20
    },
    "byte-count": 5512,
    "packet-count": 42
  }
}
```


Bu istatistik sonrasında elde edilen “packet-count” yani paket sayısı, akış veri tabının da bu akış ile ilgili işlenmiş olan son paket sayısı ile karşılaştırılır. Eğer paket sayısında artış gözlemlenir ise bloklama korunur. Ayrıca elde edilen son paket sayısı bu akış ile ilgili akış veri tabanın da güncellenir.

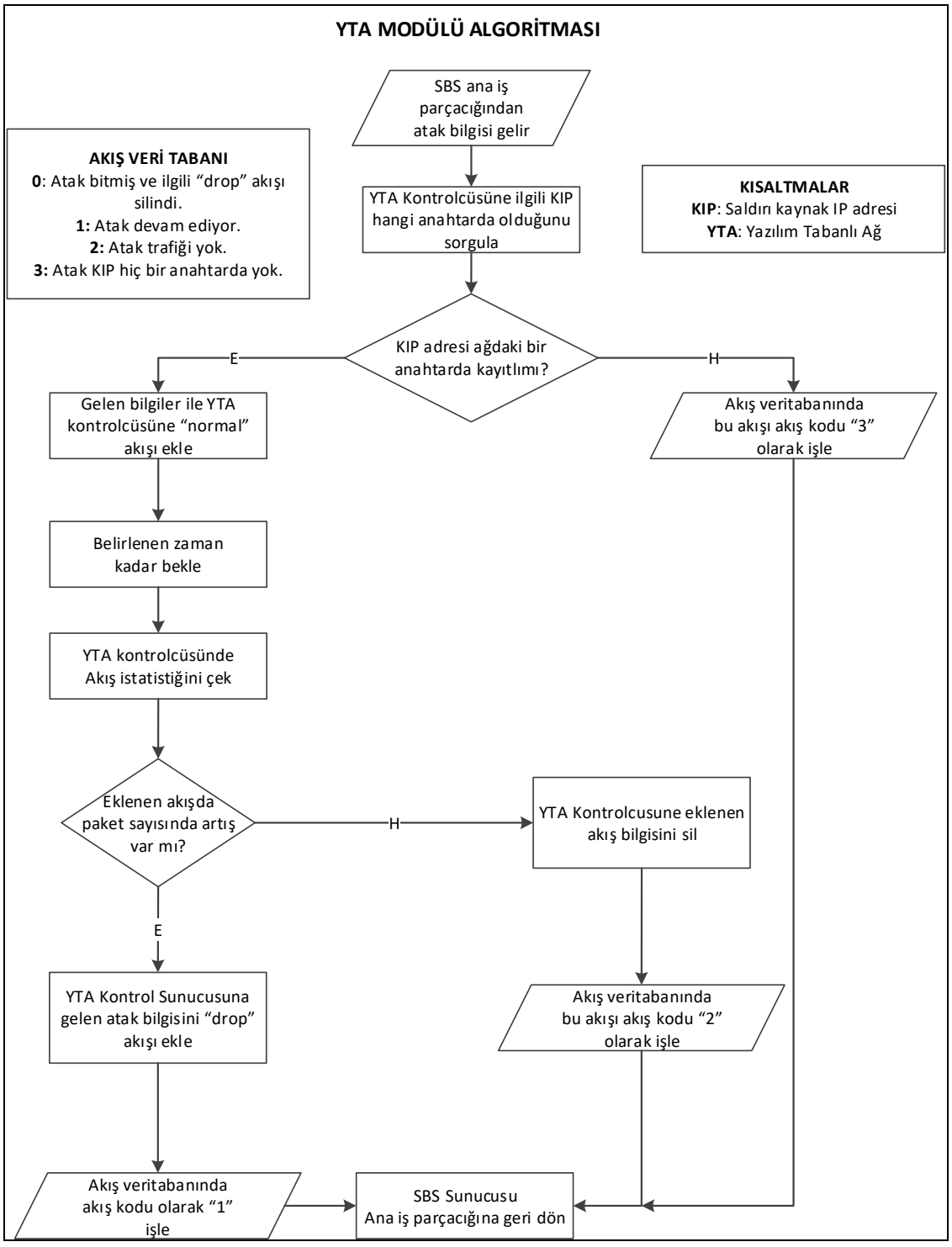
Şayet çekilen paket sayısı değeri veri tabanındaki ile aynı ise yani beklenen süre sonrasında paket sayısı artmamış ise atağın durduğu kabul edilir ve ilgili akışın kaldırılması sürecine geçilir. Bunun içinde “YTA_flow_silme” modülünü kullanır. Bu modüle YTA kontrolcünün IP adresi, port numarası, anahtar numarası, tablo numarası ve akış numarası bilgileri iletilir. Bu bilgilerin yardımı ile YTA kontrolcüsüne REST API’ı kullanarak aşağıdaki URL ile DELETE emrini yollar.

http://YTA_Kontrolcu_IP_Adresi:Port_No/restconf/operational/opendaylight-inventory:nodes/node/openflow:ANAHTAR_NO/table/TABLO_NO/flow/AKIŞ_NO/opendaylight-flow-statistics:flow-statistics/

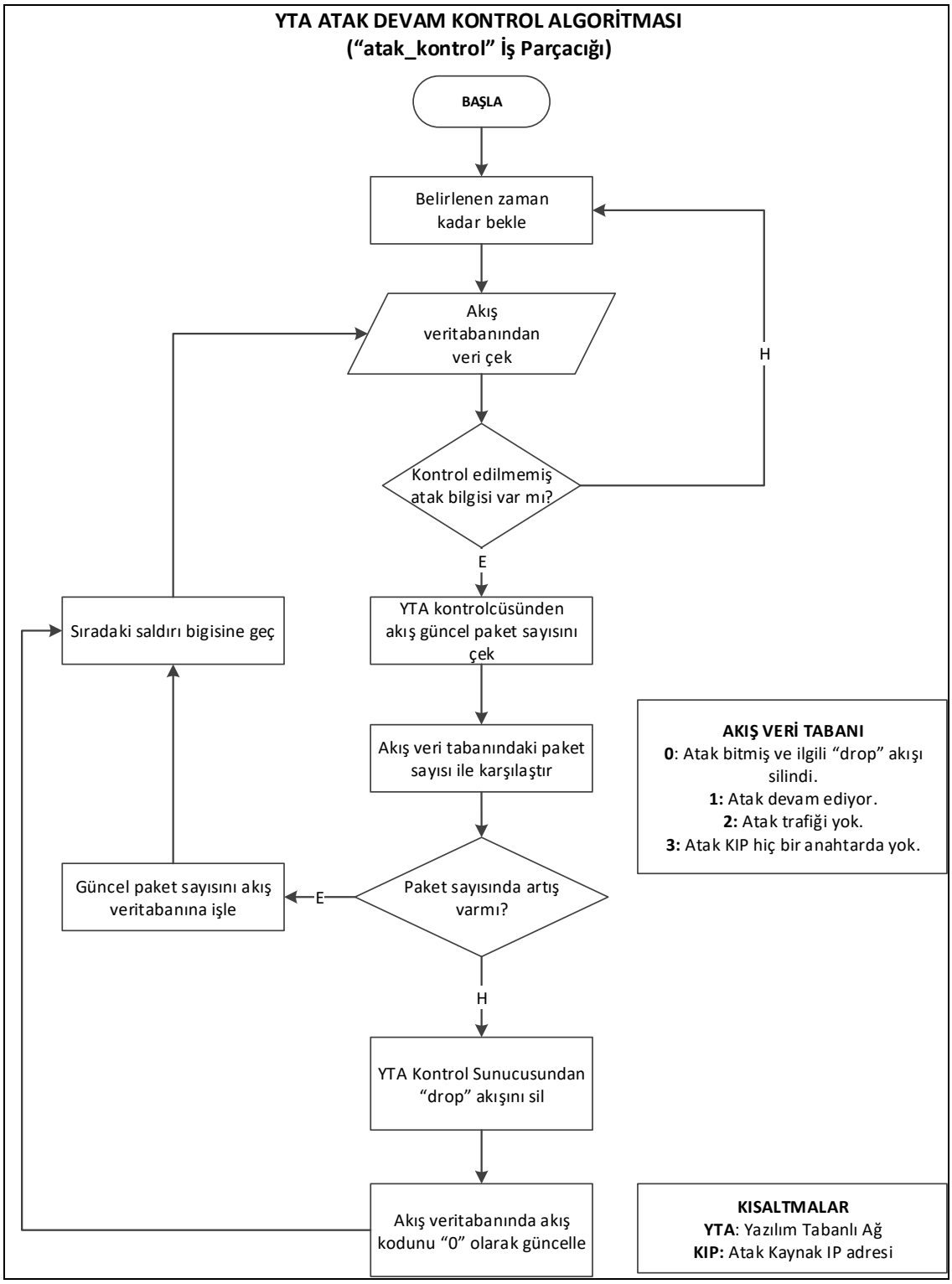
Aşağıda örnek URL adresini bulabilirsiniz.

http://10.0.11.200:8181/restconf/operational/opendaylight-inventory:nodes/node/openflow:111/table/0/flow/201/opendaylight-flow-statistics:flow-statistics/

Silme işlemi gerçekleştirildikten sonra akış veri tabanında bu akışın durum kodu 0 olarak değiştirilir. Yani atağın bittiği ve akışın kaldırıldığı belirlenmiş olur. Şekil 4.12'de ve Şekil 4.13'de YTA süreci ile ilgili algoritma diyagramlarını bulabilirsiniz.



Şekil 4.12 YTA modülü algoritması.



Şekil 4.13 YTA atak devam kontrol algoritması.

4.2 Ulusal Siber Güvenlik Merkezi (USGM) Sunucusu Detayları

USGM sunucusu yazılımı 2 adet çoklu iş parçacığından oluşmaktadır.

Bunlar:

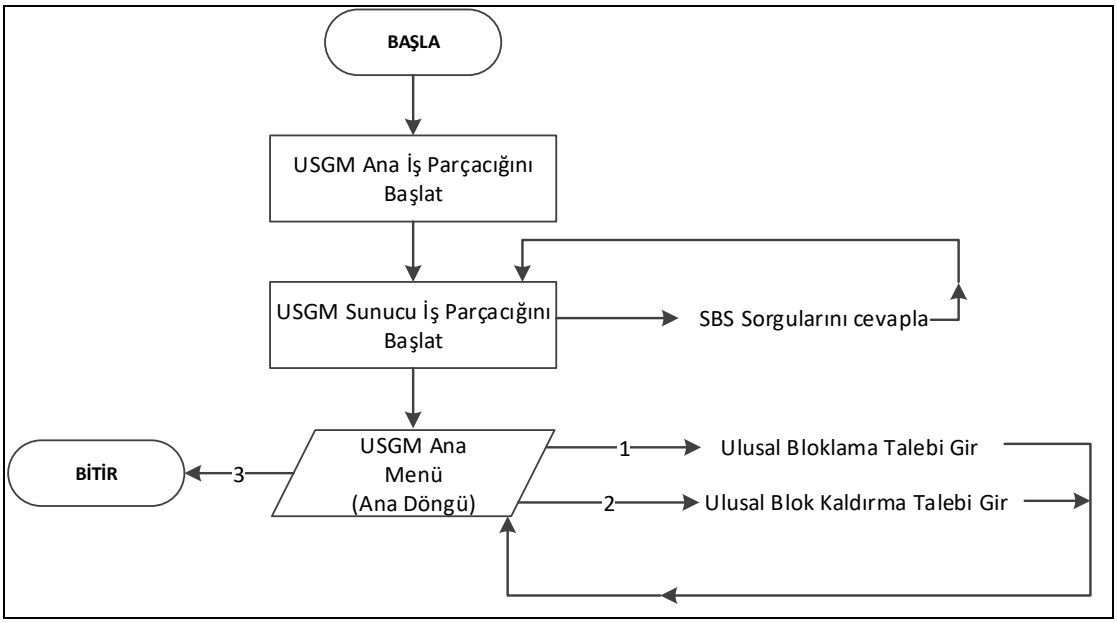
1- Ana iş parçacığı (USGM istemci kısmı)

Bu iş parçacığının amacı, USGM sistemini ayağa kaldırmaktır. Ayrıca gerekmesi durumda ulusal bir saldırı bloklama veya saldırı blok kaldırma talebi girilmesini sağlalar. Sonrasında da bu talebi SBS sunucularına iletir.

2- Sunucu iş parçacığı (USGM sunucu kısmı)

Bu iş parçacı SBS sunucularından gelen bilgilendirme taleplerini karşılamak amaçlı genel bir döngü dahilinde çalışmaktadır.

USGM sunucusu iş parçacıklarının çalışma süreci detayına Şekil 4.14'dan ulaşabilirsiniz.



Şekil 4.14 USGM çalışma süreci.

USGM sunucusu dahilde kullanılan hazır modülleri Çizelge 4.7'de, oluşturulup kullanılan modülleri de Çizelge 4.8'de bulabilirsiniz.

Çizelge 4.7 Kullanılan hazır modüller.

"socket" Modülü: Bu modül ile programın TCP/IP sunucu ve istemci olarak çalışması sağlanmaktadır.
"threading" Modülü: Bu modül ile birden fazla iş parçacığının çalıştırılabilmesi sağlanmaktadır.
"os" Modülü: Bu modül ile programın yönetimini sağlayan komut satırı arayüzünün şekillendirilmesi sağlanmıştır.
"ipaddress" Modülü: Bu modül ile sisteme girilen IP adresinin detaylarının kontrolü sağlanmaktadır.

Çizelge 4.8 Oluşturulan modüller.

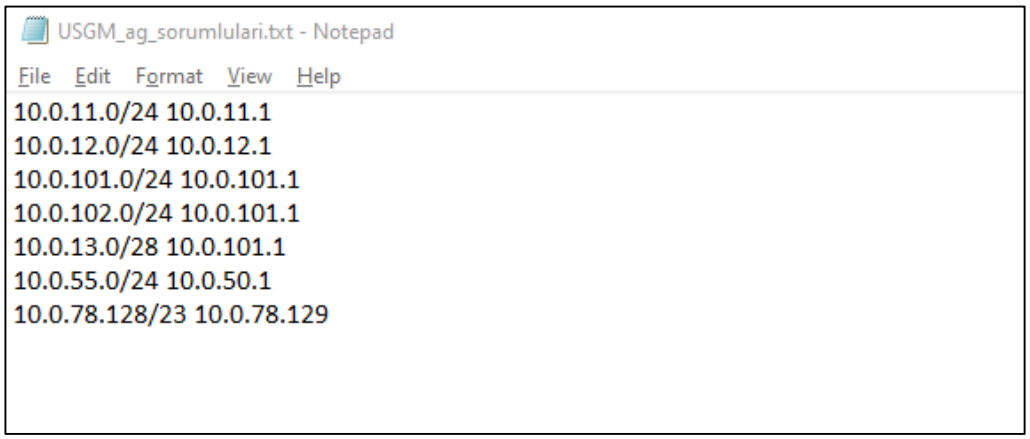
"USGM_sorumlu_kontrol" Modülü: Bu modül dahilinde "sorumlu_ip_kontrol" fonksiyonu ve "yetki_kontrol" fonksiyonu bulunmaktadır. "sorumlu_ip_kontrol" fonksiyonu "saldirgan_ip" değişkenini alır, yani saldırı kaynak IP adresinin alır ve sorumlu SBS sunucusunun IP adresini geri döner. "yetki_kontrol" fonksiyonu ise "soran_ip" değişkenini alır, yani USGM sunucusuna sorgudan bulunan SBS sunucusunun IP adresini alır ve kendisine kayıtlı bir SBS sunucusu olup olmadığını tespit eder.
"USGM_merkezi_bloklama" Modülü: "blokla_emri" fonksiyonun sahip bu modül ulusal saldırı bloklama işlemi için gerekli hedef IP adresi değişkenini alır ve SBS'lere bu emrin ulaştırılmasını sağlar. Bunun için aşağıda tanımladığımız "socket_istemci" modülünü de kullanır.
"socket_istemci" Modülü: Bu modül dahilinde "veri_yollama" fonksiyonu bulunur. Bu modül diğer SBS'ler ve USGM sunucusu ile haberleşmek amacı ile kullanılır.

“ip_adres_kontrol” Modülü: Bu modül dahilinde “ip_kontrol” fonksiyonu bulunmaktadır. Bu fonksiyon ile girilmiş olan IP adresinin doğru formatta olup olmadığı kontrol edilir.

“loglama” Modülü: Bu modül, sunucuya erişim yapan istemcilerin IP adresini ve port numaralarını “network_baglantileri_SUNUCUIPADRESİ.txt” dosyasına loglar.

4.2.1 USGM Sunucusunun Yapılandırılması

Bir sonraki adımda USGM sunucusunun yapılandırılması yapılır. Bunun için “USGM_ag_sorumlulari.txt” isimli dosya da ülke dahilinde kullanılan tüm ağ adresleri, alt ağ maskeleri ve bunları kullanan kurumların SBS sunucusu IP adresleri tanımlanır. Tanımlanma formatını Şekil 4.15’de bulabilirsiniz.

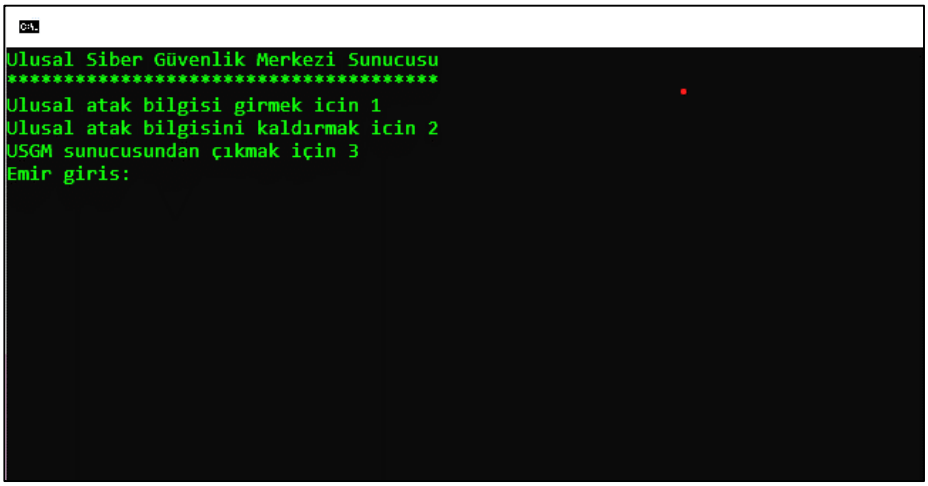


Şekil 4.15 USGM sunucusu yönetilen ağlar dosyası.

4.2.2 Ulusal Saldırı Bloklama ve Blok Kaldırma Süreci

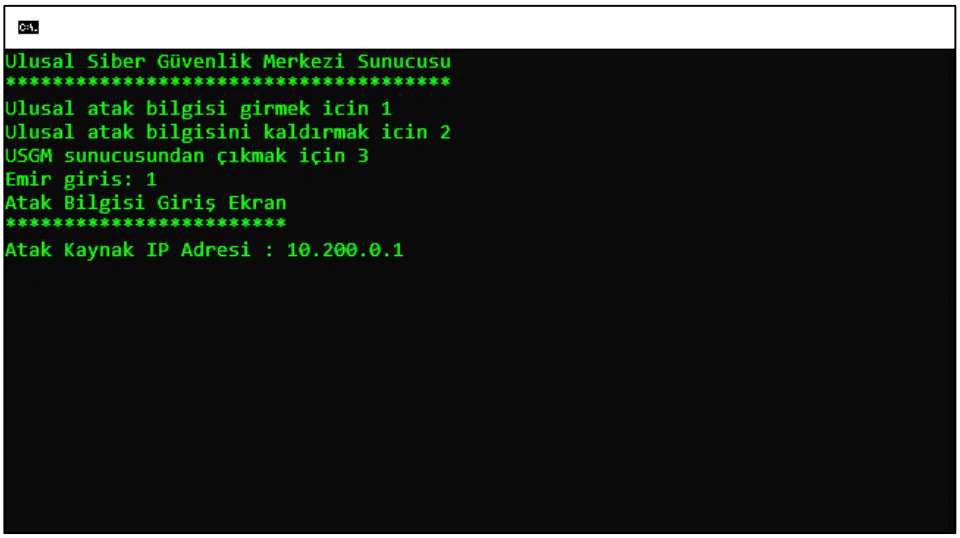
USBS sistemi Ulusal Siber Güvenlik Merkezi tarafından tüm SBS sistemine dahil olan ağlarda belirlenmiş bir hedef IP adresini bloklama talebi yapma yetkisine sahiptir. Ulusal bir atağın kesilmesi durumu gibi hallerde kullanılabilecek bu süreç USGM sunucusunda ilgili talebin girilmesi ile başlar. Bunun için USGM sunucusu ana ekranında

“1” tuşuna basılarak atak bloklama bilgisi girme ekranına girilir. Önceden verilmiş olan bloklama talebi kaldırılmak isteniyorsa yine aynı ekrandan “2” tuşuna basılır. (Şekil 4.16)



Şekil 4.16 USGM sunucusu ana ekranı.

Atak bilgisi giriş ekranında yapılması gereken sadece bloklanması talep edilen hedef IP adresinin sisteme girilmesidir (Şekil 4.17). Burada tek kontrolü yapılan doğru formatta bir IP adresinin sisteme girilip girilmediği bilgisinin kontrolüdür. Önceden bloklama talebinde bulunulan IP adresinin bloklama talebinin kaldırılması için de aynı ekran kullanılmaktadır. Yapılması gereken yine ilgili IP adresinin doğru formatta sisteme girilmesidir.



Şekil 4.17 USGM sunucusu atak bilgisi giriş ekranı.

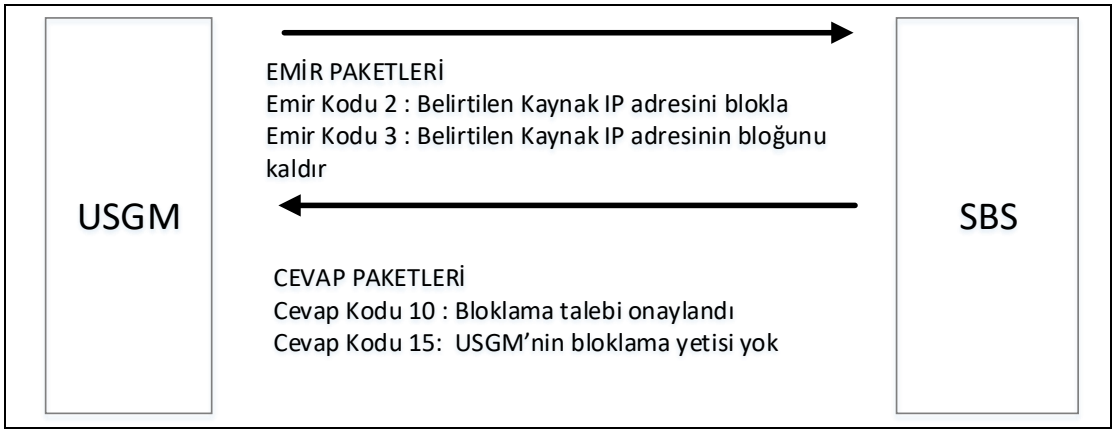
IP adresi girilir girilmez, USGM sunucusu kendisinde tanımlı tüm SBS sunucularına bloklama talebini iletmeye başlar. Şekil 4.12'de, bu süreçte yollanan paketlerin detayı verilmektedir.

Bu amaç ile önce emir kodu 2 ile tüm SBS sunucularına yollanan paket içerisinde belirtilen saldırı hedef IP adresi için bloklama emiri verilir. Emiri alan SBS sunucuları öncelikle talebi yollayan USGM sunucunun IP adresi kendisinde tanımlı USGM sunucusu ile aynı IP adresine sahip mi kontrolünü yapar. Şayet talepte bulunan sunucu tanınmayan bir sunucu ise talep reddedilir ve cevap kodu 15 ile geri dönüş yapılır. Eğer sunucu IP adresi, kendisinde tanımlı bir USGM sunucu IP adresi ise cevap kodu 10 ile geri dönülür yani talep onaylanır.

Onaylanma sonrasında talep edilen IP adresi için bloklama sürecine geçilir. Eğer sistem YTA destekli bir alt yapı ise YTA kontrolcü ile iletişime geçilip gereken bloklama işlemi gerçekleştirilir. YTA destekli olamayan bir ağ ise ilgili ağın yöneticileri manuel olarak bu bloklama işlemini gerçekleştirir. Bu sürecin detaylarını bir sonraki konuda bulabilirsiniz.

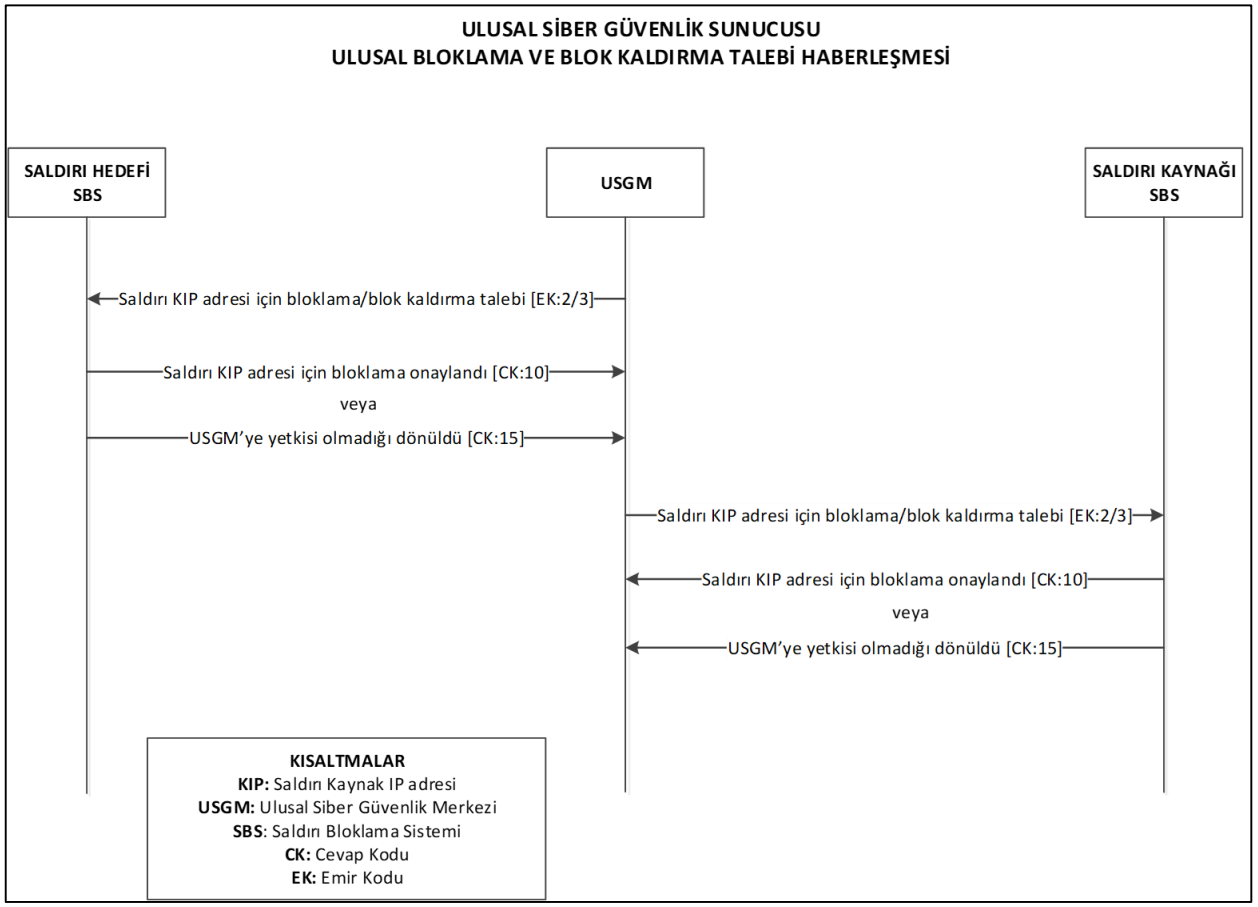
Benzer şekilde USGM sunucusu daha önce bloklama emiri verdiği bir IP adresi için bloklamanın kaldırılmasını da talep edebilir (Şekil 4.18). Bu amaçla önce sunucu tüm SBS sunucularına emir kodu 3 ile belirtilen IP adresi için geçerli bloklamanın

kaldırılmasını talep eder. Talebi alan SBS sunucusu, önce talepte bulunan USGM sunucusu IP adresininin, yetkisi olan bir sunucu olup olmadığını kontrol eder. Eğer yetkisi olmayan bir sunucu ise talebi cevap kodu 15 (yetin yok) ile geri mesaj yollayarak reddeder. Eğer sunucu IP adresi, sisteme kayıtlı bir IP adresi ise talebi onayladığını belirtmek amacı ile cevap kodu 10 ile geri döner ve YTA kontrolcüsü ile bu işlem için iletişime geçer. Eğer sistem YTA sistemine sahip değil ise o ağın sorumlusu personel tarafından bu işlemi elle gerçekleştirilebilir. Bu sürecin detaylarını bir sonraki konuda bulabilirsiniz.



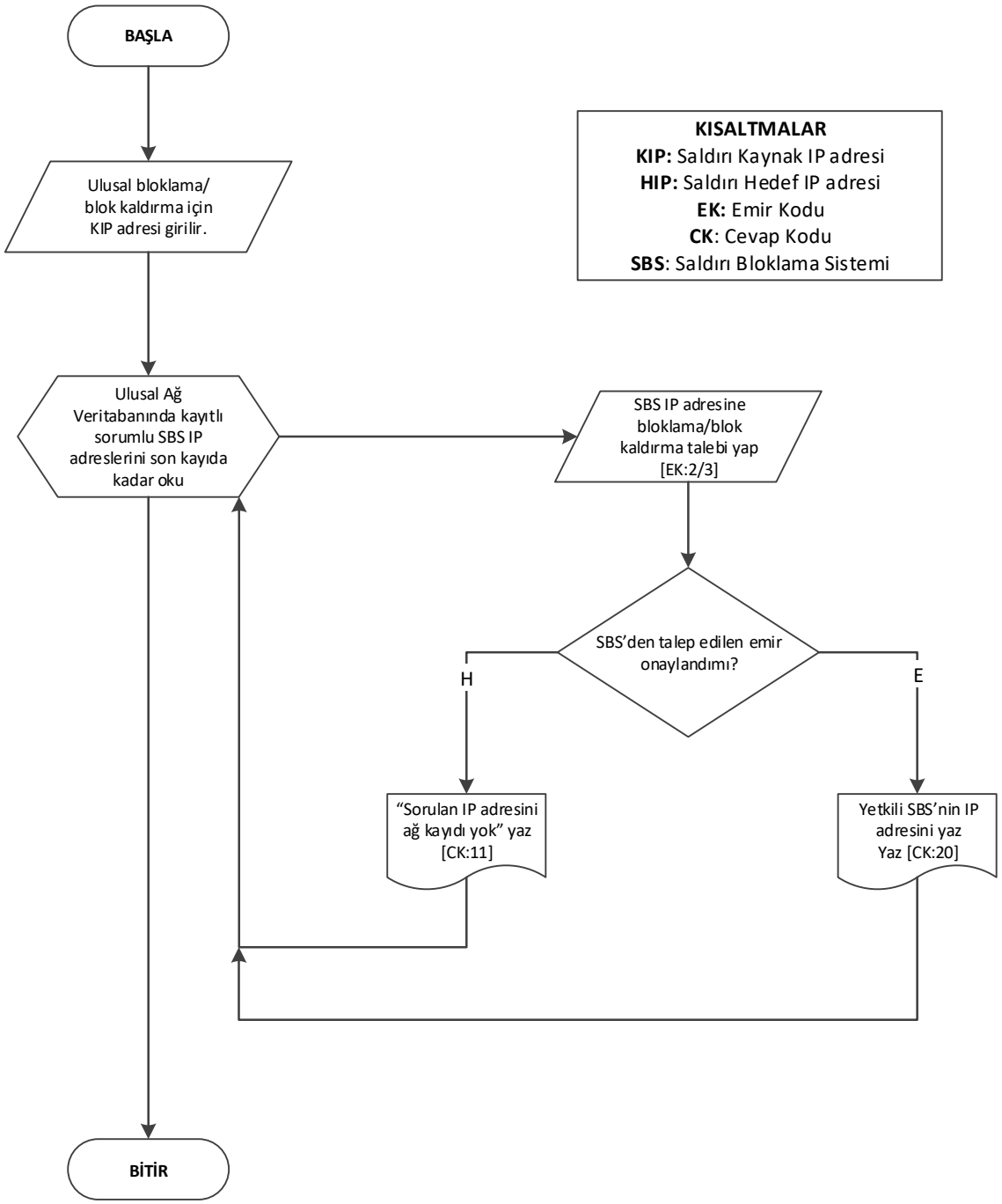
Şekil 4.18 USGM sunucusu ulusal bloklama ve blok kaldırma paketleri.

Bu süreçte yapılan tüm haberleşme detayları Şekil 4.19'da gösterilmektedir. Ayrıca Şekil 4.20'de USGM sunucunun ulusal bloklama işlemi algoritma diyagramını da bulabilirsiniz.



Şekil 4.19 USGM sunucusu ulusal bloklama ve blok kaldırma süreci.

ULUSAL SİBER GÜVENLİK SUNUCUSU
ULUSAL BLOKLAMA VE BLOK KALDIRMA TALEP ALGORİTMASI



Şekil 4.20 USGM sunucusu ulusal bloklama ve blok kaldırma talep algoritması.

4.2.2.1 SBS Sunucusunda Ulusal Atak Bloklama Süreci

Önceki konuda, bir hedef IP adresi için USGM sunucusun tüm ülke çapında bloklama talebi süreci talebi anlatılmıştı. Böyle bir talep geldiğinde, önce SBS sunucusu USGM sunucusu için yetki kontrolünü yapar. Talepte bulunan USGM sunucusunun yetkisi var ise ilgili hedef IP adresi için bloklama talebini gerçekleştirmek adına YTA modülü devreye girer. YTA modülünün "atakekle" fonksiyonu ile ilgili atak bilgisi eklenir.

Bu fonksiyona haricen YTA kontrolcüsünün IP adresi, port numarası, anahtar numarası, tablo numarası ve akış numarası dışında sadece hedef IP adresi verilir. Diğer tüm trafik verileri "-1" olarak atanır. Çünkü USGM sadece hedef IP adresine doğru yasaklama talebinde bulunabilmektedir. YTA modülü hedef IP adresi hariç tüm diğer trafik bilgilerinin "-1" olduğunu görünce bloklama talebinin USGM'den geldiğini anlar ve trafik aktif mi değil mi kontrol işlemini gerçekleştirmeden direk bloklama işlemine geçer.

YTA modülü bu amaçla yine "YTA_flow_ekleme" modülünün "put_drop_flow" fonksiyonu çağır. Hedef IP adresi için bloklama emrini REST API yardımı ile aşağıdaki URL adresine PUT mesaj tipi kullanarak yollar. Bu sayede ilgili bloklama gerçekleşir.

http://YTA_KONTROLCU_IP:PORT/restconf/operational/opendaylight-inventory:nodes/node/openflow:ANAHTAR_NO/table/TABLO_NUMARASI/flow/AKIŞ_NUMARASI

Örnek bir URL aşağıdaki gibidir:

http://10.0.11.200:8181/restconf/config/opendaylight-inventory:nodes/node/openflow:110/ table/0/flow/202

Bu emir ayrıca bloklanacak trafik detaylarını, XML formatında iletmektedir. İlgili XML formatı örneği aşağıdaki gibidir. Bu örnekte hedef IP adresi 50.0.0.1 olan trafik bloklanmaktadır.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<flow xmlns="urn:opendaylight:flow:inventory">
   <instructions>
      <instruction>
         <order>0</order>
      </instruction>
   </instructions>
   <table_id>0</table_id>
   <id>202</id>
   <match>
      <ethernet-match>
         <ethernet-type>
            <type>2048</type>
         </ethernet-type>
      </ethernet-match>
      <ipv4-destination>50.0.0.1/32 </ipv4-destination>
      </match>
   <hard-timeout>0</hard-timeout>
   <idle-timeout>0</idle-timeout>
   <cookie>202</cookie>
   <flow-name>SBS-ADDED-FLOW</flow-name>
   <priority>200</priority>
</flow>
```

4.2.2.2 SBS Sunucusunda Ulusal Atak Blok Kaldırma Süreci

USGM sunucusundan merkezi blok kaldırma talebi geldiğinde, tüm akış veri tabanında önceden verilmiş merkezi bloklama talepleri taranır. Şayet akış durum kodu 1 yani halen daha ilgili bloklama aktif olan bir USGM'den gelmiş kayıt tespit edilirse blok kaldırma işlemine geçilir. Bu amaçla "YTA_flow_silme" modülündeki "delete_flow" fonksiyonu çağırılır.

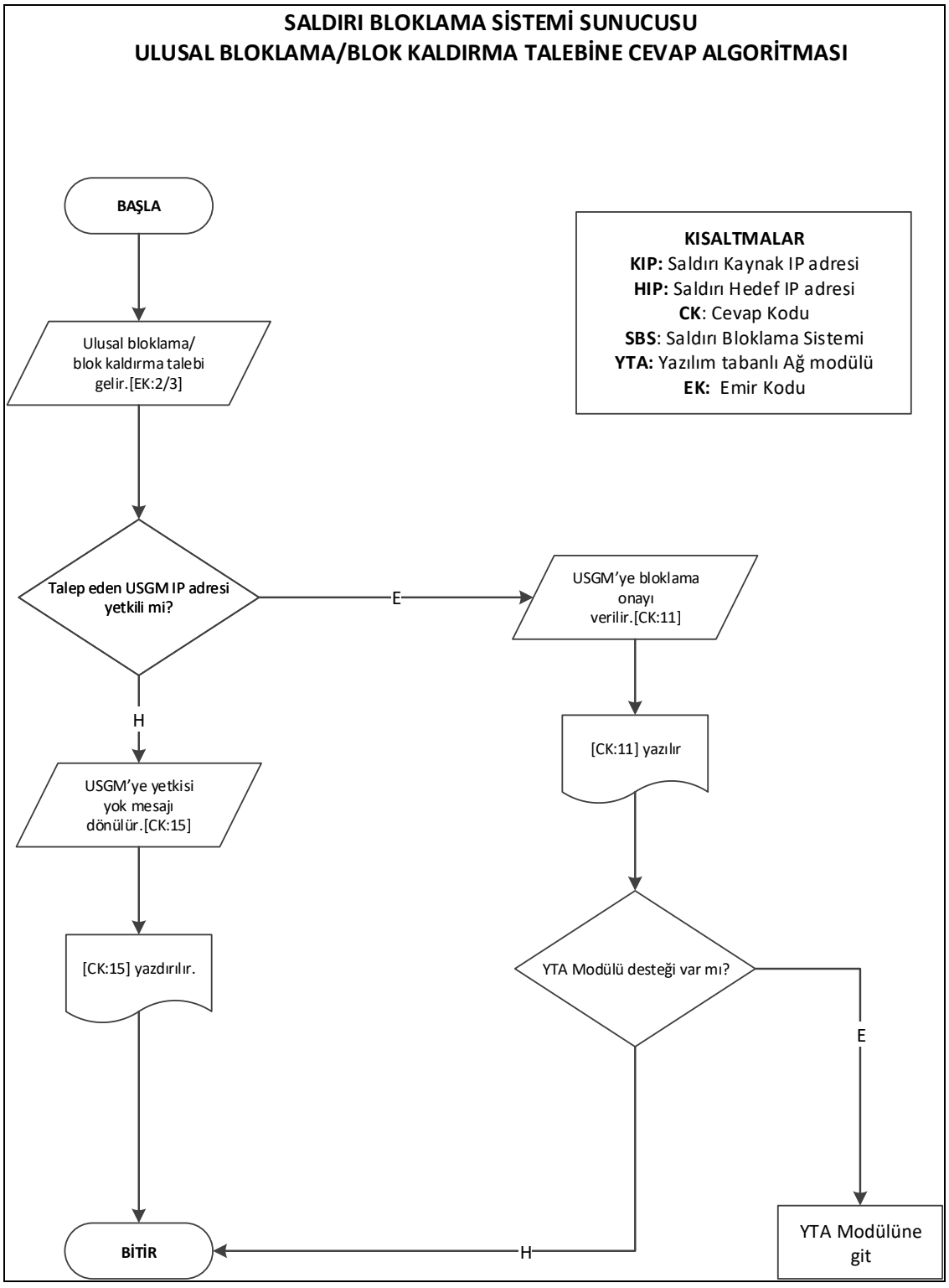
Bu fonksiyona YTA kontrolcü IP adresi, port numarası, anahtar numarası, tablo numarası ve akış numarası bilgileri iletilir. Bu bilgilerin yardımı ile YTA kontrolcüsüne REST API'ı kullanarak aşağıdaki URL ile DELETE emri yollanır.

http://YTA_Kontrolcu_IP_Adresi:Port_No/restconf/operational/opendaylight-inventory:nodes/node/openflow:ANAHTAR_NO/table/TABLO_NO/flow/AKIŞ_NO/opendaylight-flow-statistics:flow-statistics/

Aşağıda örnek URL adresini bulabilirsiniz.

http://10.0.11.200:8181/restconf/operational/opendaylight-inventory:nodes/node/openflow:110/table/0/flow/202/opendaylight-flow-statistics:flow-statistics/

Silme işlemi gerçekleştirildikten sonra akış veri tabanında bu akışın durum kodu 0 olarak değiştirilir. Yani ilgili bloklama işlemin kaldırıldığı belirtilmiş olur. Şekil 4.21'de SBS sunucunun ulusal bloklama işlemi algoritma diyagramını bulabilirsiniz.



Şekil 4.21 USGM sunucusu ulusal bloklama ve blok kaldırma cevap algoritması.

BÖLÜM 5

UYGULAMA TEST ORTAMI VE ULAŞILAN HEDEFLER

5.1 Test Ortamını Oluşturmakta Kullanılan Araçlar

Hedeflenen amaçları ve geliştirilen sistemi test etmek için bir adet YTA destekli atak yapan, bir adet YTA destekli ve bir adette YTA desteği olmayan atağa maruz kalan ağ oluşturulmuştur. Toplam 3 adet ağın yanı sıra Ulusal Siber Güvenlik Merkezi (USGM) sunucusunun bulunacağı dördüncü bir sistemin de devreye alınmıştır. Bu sistemin oluşturulmasını sağlayan araçlar bu bölüm dahilinde ele alınmıştır.

5.1.1 YTA Destekli Ağ Ortamı

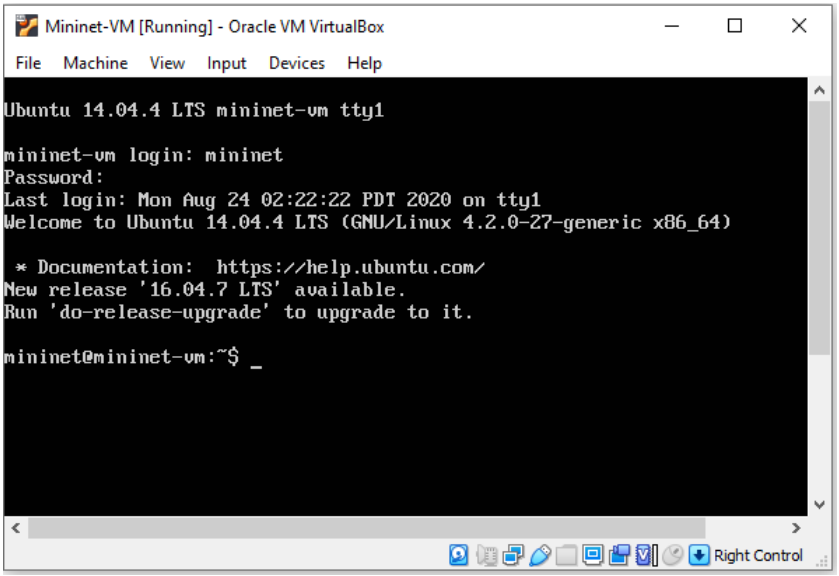
Çalışma da öncelikle fiziksel YTA anahtar cihazları kullanılmıştır. Ancak daha geniş çaplı testlerin gerçekleştirilebilmesi için çok sayıda YTA destekli anahtar cihazı ihtiyacı doğmuştur. Bu amaçla yazılım tanımlı sanal ağ anahtarlama cihazlarının kullanımına geçilmiştir. Bu sanal ağ ortamını oluşturmak için Mininet yazılımı ile çalışmalar devam ettirilmiştir (Mininet, 2021). Mininet, içerisinde sanal istemci, Openflow destekli sanal ağ anahtarı ve YTA kontrolcüsü barındıran, özel oluşturulmuş bir Linux dağıtımıdır. Mininet, özellikle Openflow uygulamaları geliştirmek amaçlı bir platformdur. Python API desteği ile istenilen ağ topolojisi oluşturulabilmektedir. Yine içerisine yerleştirilmiş test araçları yardımıyla üzerinde çalışılan proje ile ilgili testlerin yapılmasına da olanak sağlamaktadır.

Mininet içerisinde OVS (Open Virtual Switch) sanal ağ anahtarları kullanmaktadır. OVS, Apache Yazılım Vakfı tarafında geliştirilmiş olan bir projedir. Çalışma dahilinde OVS sürüm 2.0.2 kullanılmıştır (OpenVirtualSwitch, 2021).

Mininet fiziksel bir bilgisayar üzerinde çalışabildiği gibi sanal bir bilgisayar üzerinde de çalışabilmektedir. Birden fazla ağı aynı bilgisayarda oluşturmak amacı ile sanal bilgisayar üzerinde çalışan Mininet'in 2.2.2 sürümü tercih edilmiştir. Sanal işletim sistemini kullanmak amacı ile de Oracle şirketi tarafından geliştirilen ücretsiz ve açık kaynaklı hipervizör yazılımı olan Virtual Box'ın 6.1 sürümü tercih edilmiştir (VirtualBox, 2021).

5.1.1.1 Mininet Ağ Ortamının Kurulması

Çalışma dahilinde kullanılan Mininet sanal makine sürümü, Ubuntu Linux'un 64bitlik dağıtımı üzerine yapılandırılmış bir sürümdür. 1 Gigabyte hafıza, 16 Megabyte ekran kartı hafızası, 8 Gigabyte depolama alanı Mininet için yeterlidir (Şekil 5.1). Ağ yapılandırması olarak "Host-only Adapter" seçeneği yapılandırılmıştır. Bu seçenek sayesinde aktive edilen sanal bilgisayarlar ve ana bilgisayar rahatlıkla diğer ağlardan izole olarak haberleşebilmektedir.

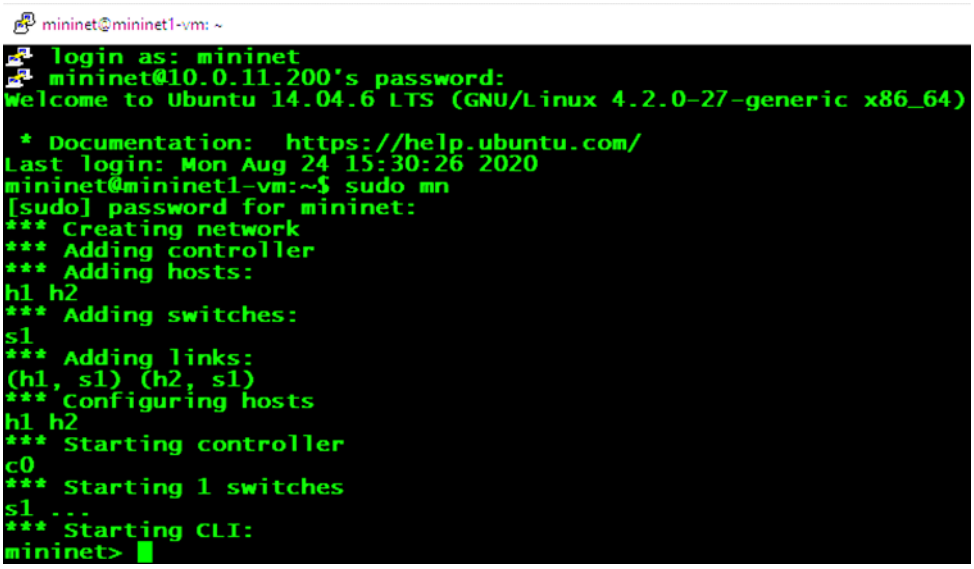


Şekil 5.1 Mininet ortamı.

Mininet işletim sistemine varsayılanda kullanıcı adı ve parolası “mininet” olarak giriş yapılabilmektedir. Bu aşamadan sonra Mininet’i aktive etmek için gereken komutlar bir sonraki bölümde ele alınmıştır.

5.1.1.2 Mininet Komutları

Mininet’i başlatmak için “sudo mn” komutunu yazmak yeterlidir (Şekil 5.2). Bu işlem ile S1 isminde bir anahtar cihazı ve üzerinde iki adet istemci ile temel bir ağı oluşturulur. Oluşan bu topoloji hakkında bilgi almak için “net” komutu, oluşturulan topolojinin bağlantı detaylarını görmek için “links” komutu kullanılabilir. Ayrıca ağa bağlı istemciler hakkında bilgi almak için de “net” komutu kullanılabilir. Mininet içinde barındırdığı referans YTA kontrolcü ise ile çalışır. Kontrolcü ve diğer cihazlar hakkında topluca bilgi almak için “dump” komutunu çalıştırmak yeterli olacaktır.



Şekil 5.2 Mininet’in başlatılması.

İstemciler üzerinde komut çalıştırmak için istemci adı ve çalıştırılmak istenen komut yazılır. Örneğin “h1 ifconfig” komutu ile “h1” isimli istemcinin arayüzleri ve bu arayüzlere atanmış IP adresleri görülebilir. “h1” istemcisinde çalışan işlemleri listelemek

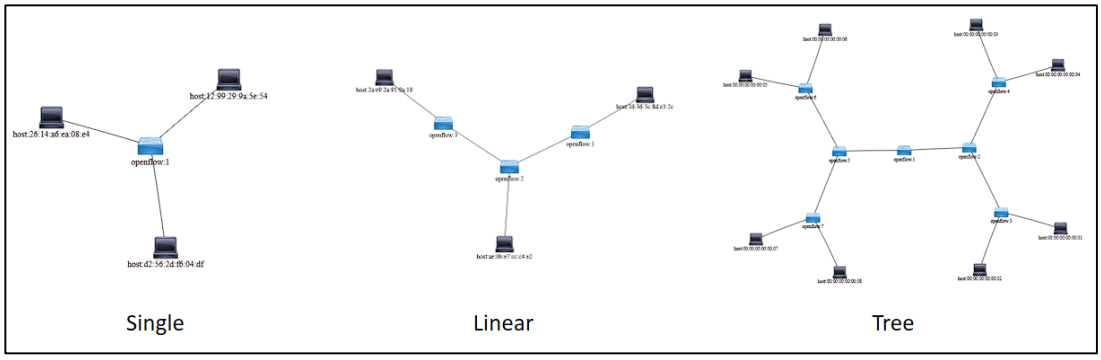
için “h1 ps -a” komutu çalıştırılır. Benzer şekilde sistemdeki OVS anahtarda çalışan işlemleri listelemek, “s1 ps –a” komutu ile mümkündür. Oluşturulan bu ağı kapamak için de “exit” komudu çalıştırılır. Mininet ile farklı topolojiler oluşturmak için kullanılacak komutlar şunlardır:

“sudo mn --topo single,X” : Tek bir anahtarın üzerinde X adet istemciden oluşan bir ağ oluşturur. Örneğin X yerine 3 yazılması durumunda 3 adet istemciye sahip bir topoloji oluşturulabilir.

“sudo mn --topo linear,X,Y” : X adet anahtarın doğrusal (bus) topolojisi şeklinde bağlanarak, her bir anahtarda da Y adet istemci olacak şekilde bir ağ oluşturulur. Buradaki Y değeri belirtilmez ise her anahtar üzerinde 1 bir istemci olacak şekilde bir ağ oluşturulur.

“sudo mn --topo tree,depth=X” : Bu komut ile X adet uç uca bağlanmış ağaç yapısı şeklinde bir topoloji oluşturulur.

Tüm bu topoloji örneklerini Şekil 5.3’de görebilirsiniz.



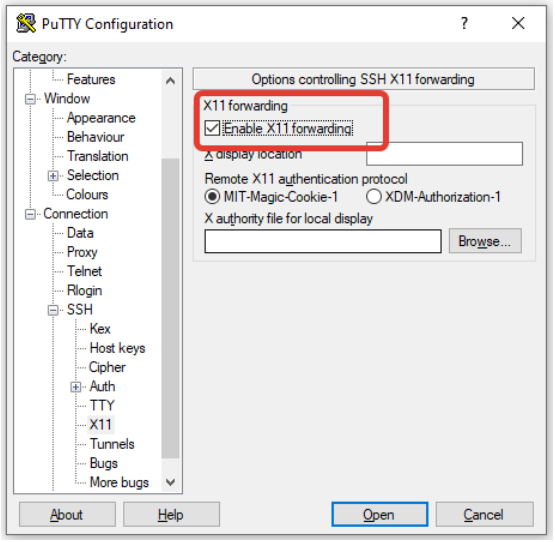
Şekil 5.3 Mininet hazır topoloji şablonları.

Bu komutlar ile Mininet kendisi istemciler IP ve MAC adreslerini atar. Şayet “--mac" şeklide bir parametre eklenirse, istemcilerin MAC adresleri daha okunaklı olan “00:00:00:00:00:01” adresinden başlayacak şekilde atanır. Her istemciye isten IP adresi ve MAC adresi atanacak şekilde bir topoloji kısa bir betik ile yazılıp, “mn --custom

./script11.py --topo mytopo” komutu ile oluşturulabilir. Bu konuda oluşturulan test ortamı için oluşturulmuş bir betik örneği bulunmaktadır.

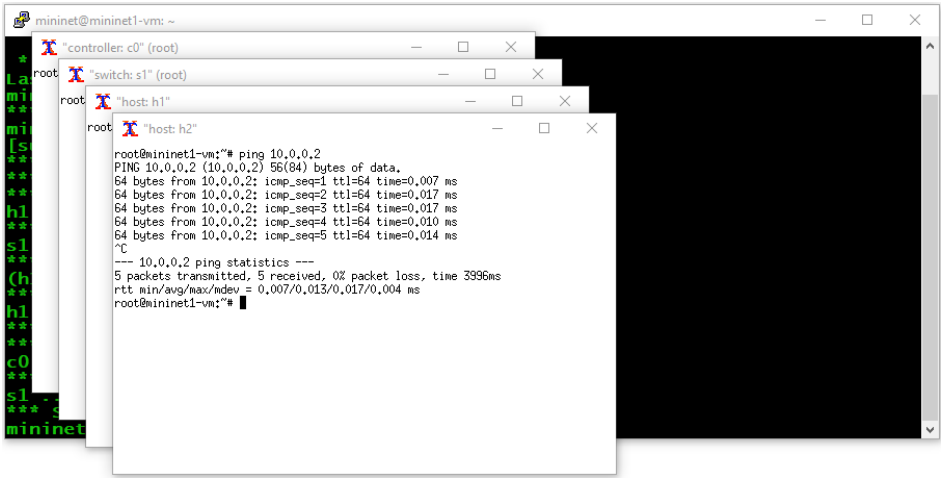
5.1.1.3 Mininet Grafik Ortamı

Mininet ile oluşturulmuş ağ ortamındaki istemcilerin komut arayüzlerine doğrudan erişebilmek ve bu ağ içerisinde akan trafiği gözlemlemek amacı ile “Wireshark” benzeri grafik arayüzlerini kullanabilmekte mümkündür. Bunun için öncelikle sanallaştırma yapılan bilgisayara, X11 grafik arayüzü sunucusunun yüklü olması gerekmektedir (Xming, 2021). Sunucu yüklendikten sonra Mininet sunucusuna bağlantı için kullanılan konsol yazılımında X11 özelliğinin de açık olduğu kontrol edilmelidir (Şekil 5.4).



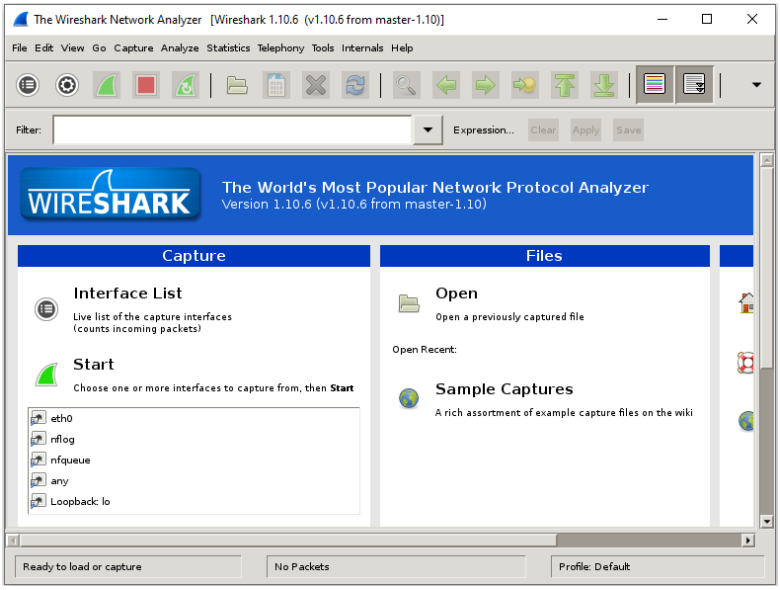
Şekil 5.4 Konsol yazılımında X11 özelliğinin aktive edilmesi.

Artık SSH ile bağlanılan Mininet sunucusunda, grafik arayüzü ile yönetim araçları aktive edilebilir. Öncelikle oluşturulmuş ağdaki istemcilere doğrudan ulaşmak için Mininet başlatılırken “-x” parametresi eklenmelidir. Bu aşamadan sonra istemciler üzerinden doğrudan komut çalıştırılabilmektedir (Şekil 5.4).



Şekil 5.5 Mininet istemcilerine doğrudan erişim.

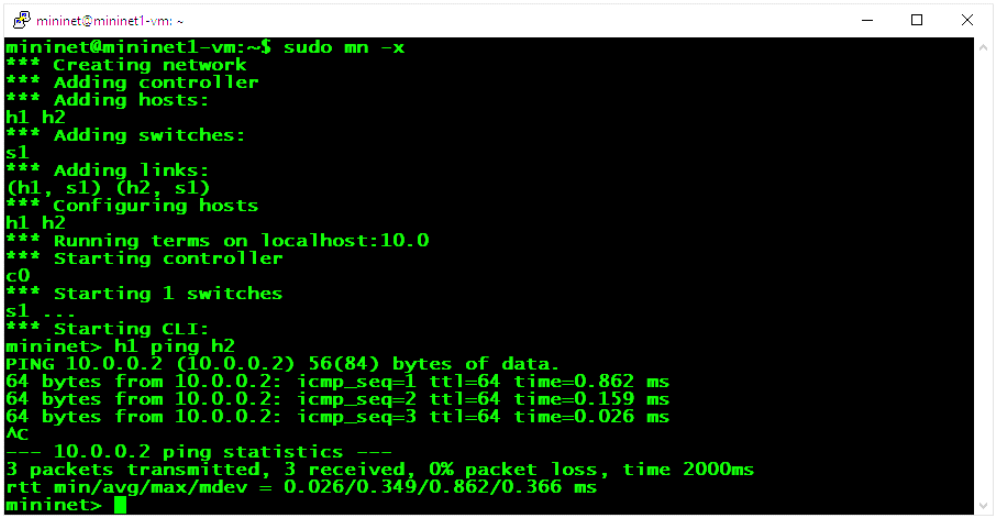
Ağ içindeki haberleşmeyi gözlemleyebilmek içinde ise Mininet başlatılmadan önce “sudo wireshark &” komutunun çalıştırılması yeterli olacaktır (Şekil 5.6). Bu aşamadan sonra Mininet başlatılınca paket izleme yazılımda beraberinde başlayacaktır.



Şekil 5.6 Wireshark GUI arayüzüne erişim.

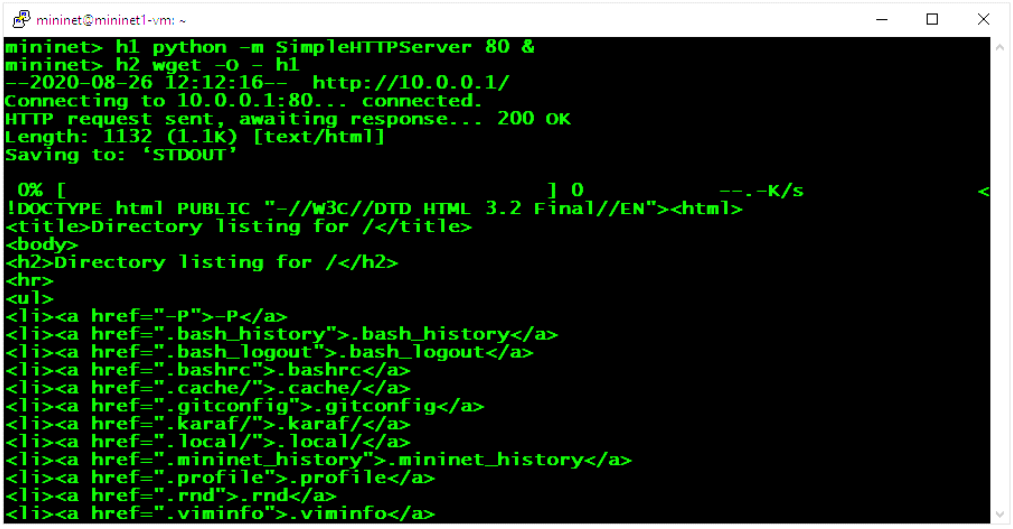
5.1.1.4 Mininet'de Haberleşme Testlerinin Yapılması

Oluşturulan sanal ağda, en temel yapılacak test "ping" aracını kullanmaktır. "pingall" komdu ile oluşturulmuş ağdaki tüm istemciler birbirlerini pinglemektedir. Sadece bir istemci pinglenmek istendiğinde ise kaynak bilgisayar adı, "ping" komutu ve hedef bilgisayar adının yazılması yeterli olacaktır (Şekil 5.7).



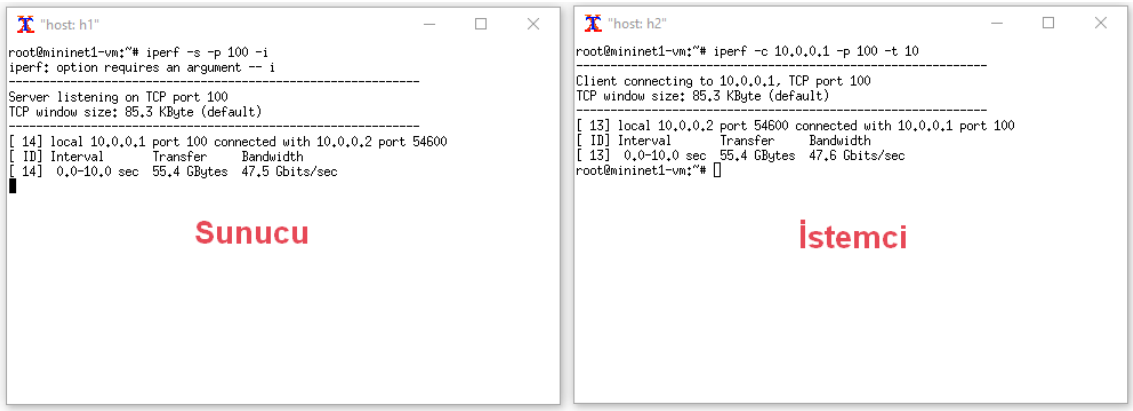
Şekil 5.7 İstemciler arasında Ping uygulaması.

İstemciler arasında haberleşme amaçlı diğer bir test ise istemcilerden birinde aktive edilecek basit bir web sunucusu ile gerçekleştirilir. Bunun için Mininet ortamında "h1 python -m SimpleHTTPServer 80 &" komutu çalıştırılır. Bu komutla beraber "h1" isimli istemcide 80 nolu porttan hizmet veren bir web sunucusu aktive edilmiş olur. Daha sonra diğer istemciler "h2 wget -O - h1" komutu ile bu web sunucuna erişim testleri gerçekleştirilebilir (Şekil 5.8). Benzeri bir test şayet sanal makinaya "curl" uygulaması yüklenmiş ise "curl" ile de "h1 curl h2" komutu ile gerçekleştirilebilir.



Şekil 5.8 Web uygulamasının aktive edilmesi ve testi.

Son olarak, Mininet üzerinden “iperf” paket oluşturma uygulaması ile istenen trafik tipi oluşturulup trafik testleri de gerçekleştirilebilir. Bunun için önce istemcilerden birinde iperf sunucusunun aktive edilmesi gerekir (IPerf, 2021). Artık başka bir istemciden, oluşturulmuş bu sunucuya erişim hızı testi yapılabilir. Şekil 5.9’da “h2” istemcisinde TCP 100 portundan yapılan haberleşme testi görülmektedir. Bu testte yapılan 10 saniyelik haberleşme sonucunda erişim hızının 47.6Gbit/s olduğu gözlemlenmektedir.



Şekil 5.9 Iperf test aracının kullanımı.

5.1.2 YTA Kontrolcü

YTA kontrolcüsü, uygulama katmanında çalışan yazılımlardan gelen emirleri ağ cihazlarına aktaran yazılımdır. YTA kontrolcü yazılımlarının ticari versiyonları olduğu gibi, özgür yazılım ve açık kaynak kodlu olanları da mevcuttur. Çalışma dahilinde bir özgür yazılım olan Opendaylight YTA kontrolcü yazılımı tercih edilmiştir (OpenDaylight, 2021).

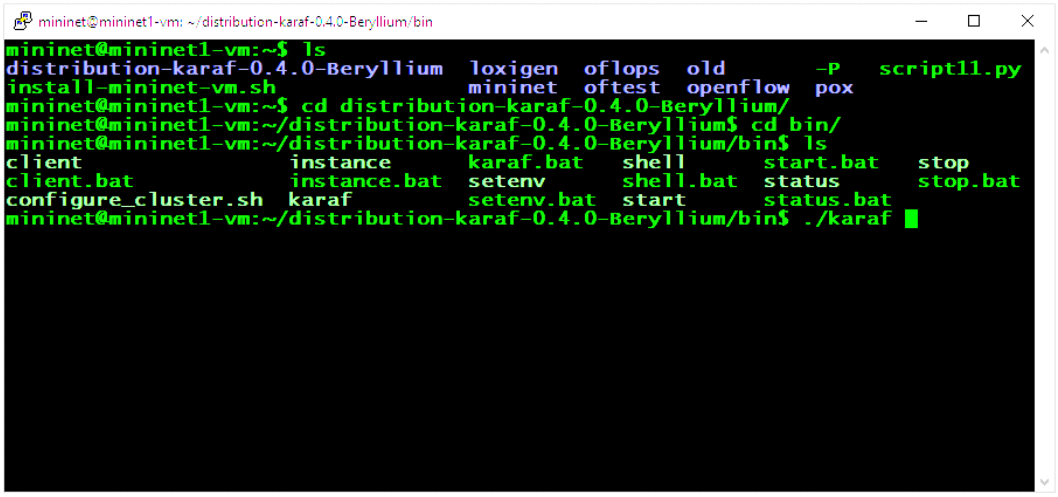
Opendaylight, Linux Vakfı tarafında geliştirilmektedir. Bu yazılım her boyutta ağı özelleştirmek ve otomatikleştirmek amacı ile oluşturulmuş modüler bir açık platformdur. Opendaylight yazılımına REST API ile istenen emirler yollanabilir ve bu yazılım da bu emirleri Openflow protokolünü kullanarak ağ cihazlarına iletebilmektedir. Proje dahilinde Opendaylight'ın Beryllium diye isimlendirilen versiyonu kurulmuştur. Java tabanlı geliştirilmiş olan bu yazılım, Linux ve Windows platformlarına kurulabilmektedir.

Opendaylight yazılımınının kendi resmi sitesinden indirildikten sonra, özel bir kurulum süreci yoktur. İndirilen yazılımın içinde bulunan "bin" klasörününde ki "karaf" yazılımının çalıştırılması Opendaylight'ın çalışması için yeterli olacaktır (Şekil 5.10). Çalıştırılması sonrasında Opendayligth'ın komut arayüzü devreye girer. Bu arayüz ile kendisine ait ek modüller yüklenebilmektedir (Şekil 5.11). Bunun için "feature:install <modül adı>" komutunun çalıştırılması yeterlidir. Çalışma dahilinde aşağıdaki modüller kurulmuştur.

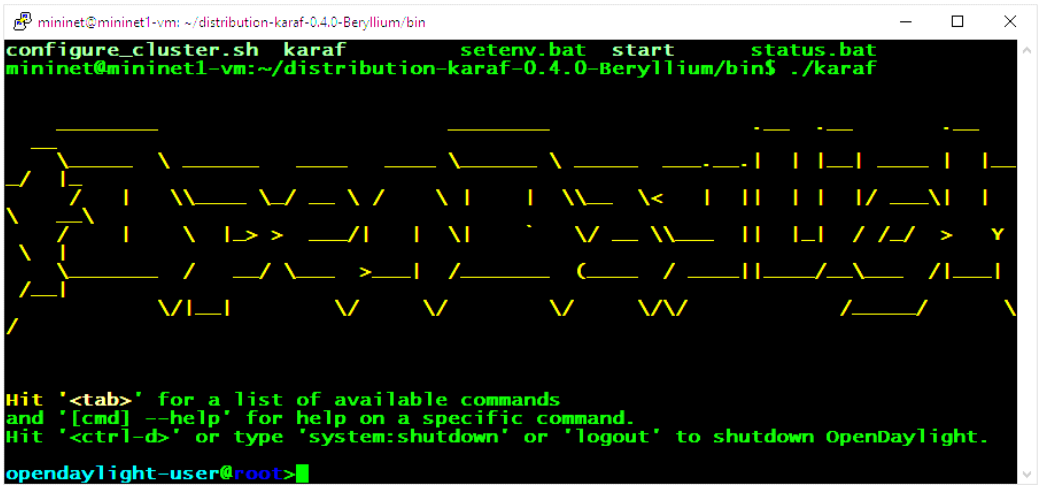
odl-restconf: Rest API ile Opendaylight üzerinden haberleşmeyi sağlar.

odl-l2switch-switch: Opendaylight'ın temel 2. Katman anahtarın paket yönlendirme kararlarını vermesini sağlar.

odl-dlux-all: Mininet ile oluşan topolojilerin "dlux" isimli grafik arayüzü ile kontrol edilebilmesini sağlar.



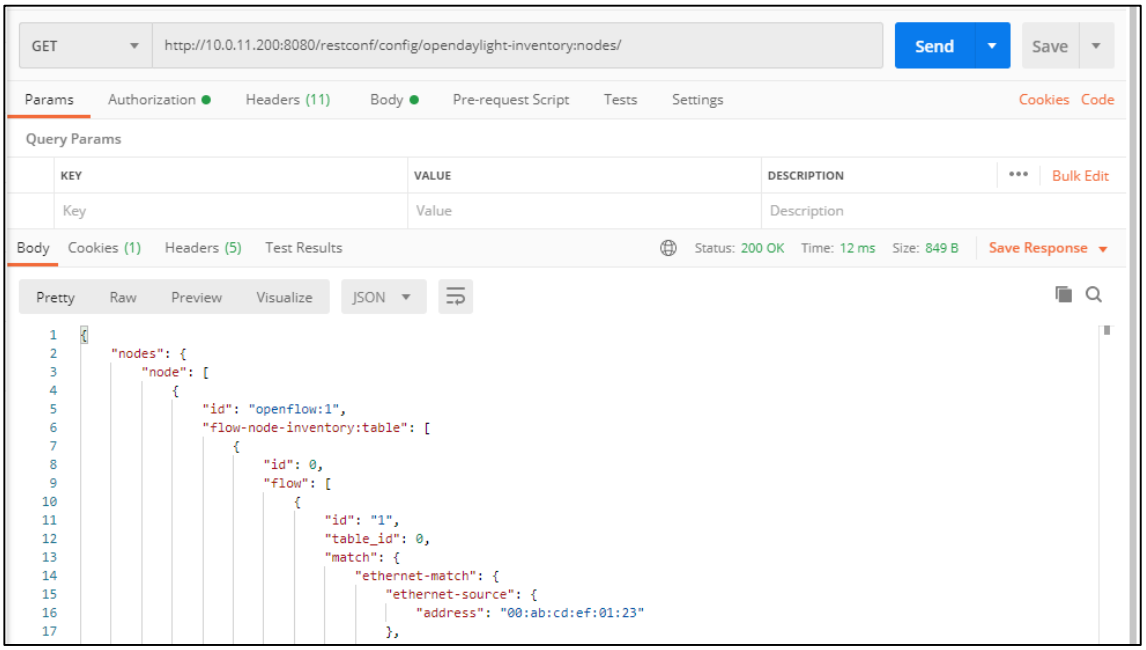
Şekil 5.10 Opendaylight'ın çalıştırılması.



Şekil 5.11 Opendaylight arayüzü.

Mininet ağ simülasyon platformu varsayılanda, kendi içinde olan referans YTA kontrolcüsünü kullanmaktadır. REST API ile erişim ve grafik arayüzünün kullanılabilmesi için Mininet, harici bir YTA kontrolcüsü ile çalıştırılabilmektedir. Bunun için "sudo mn --controller=remote,ip=YTA_Kontrolcüsü_IP adresi" komutunun çalıştırılması yeterli olacaktır. Opendaylight ile entegre olan Mininet platformuna artık RESTAPI ile istenen sorgu veya emir verilebilir. Şekil 5.12'de Postman yazılımı ile YTA

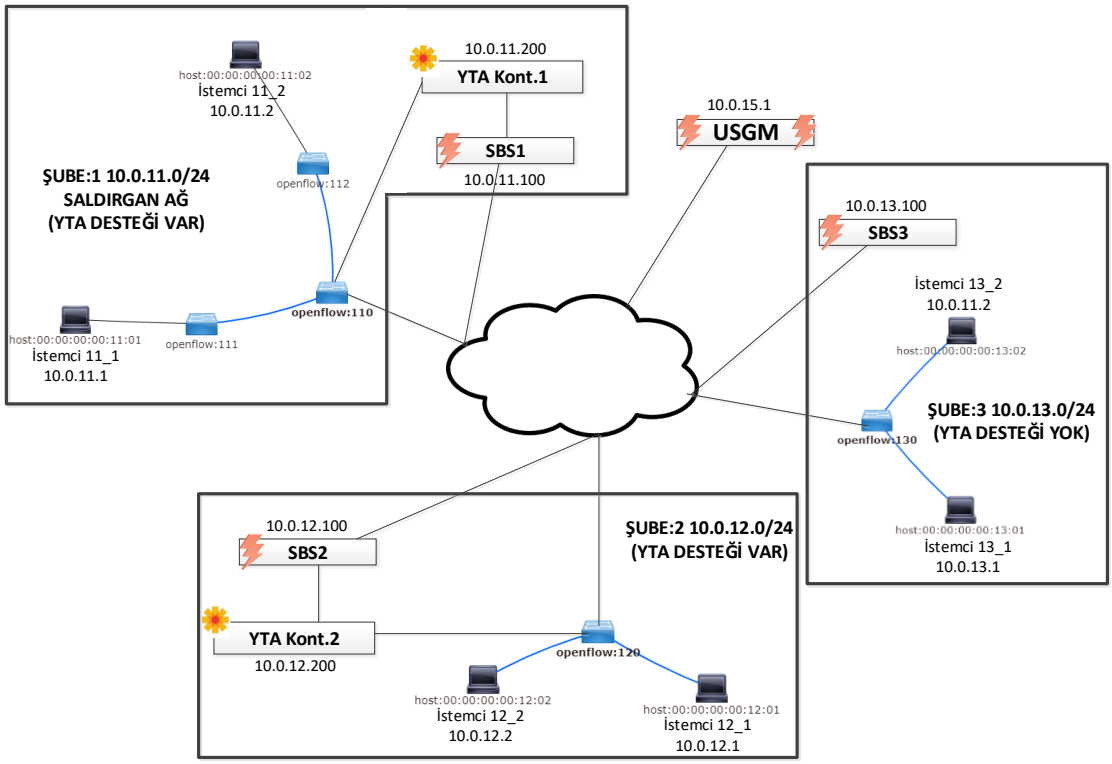
kontrolcüsüne “GET” mesaj tipi ile yapılmış bir RESTAPI sorgusunu bulabilirsiniz (Postman, 2021).



Şekil 5.12 Rest API ile sorgunun yapılması.

5.2 Test Ortamın Detayları

Test ortamı olarak 3 farklı ağ ve bir adet ulusal siber güvenlik merkezi oluşturulmuştur (Şekil 5.13). Oluşturulan ilk ağ şube 1 olarak isimlendirilmiştir ve saldırı kaynak ağı görevini üstlenmiştir. YTA destekli bu ağ, toplamda 3 adet ağ anahtarı ve 2 istemciden oluşturulmuştur. Şube 2’de YTA desteği olan bir ağ olarak yapılandırılmıştır. Bu ağda bir adet anahtar cihazı ve iki adet istemci yer almaktadır. Şube 3 ise YTA desteği olmayan bir ağ olarak planlanmıştır. Şube 3 dahilinde 1 adet ağ anahtarı ve 2 istemci yer almaktadır. Buradaki Şube 2 ve Şube 3 saldırıya maruz kalan ağlar olarak oluşturulmuştur. Saldırıların bloklanması için gereken bilgilendirmeyi sağlayan USGM birimi yani ulusal siber güvenlik merkezi birimi de ayrı bir IP adresi ile çalıştırılmıştır. Çizelge 5.1’de bu ağların ve bağlı cihazların IP adreslerine ulaşabilirsiniz.



Şekil 5.13 Test ortamı.

Çizelge 5.1 Test ortamı detayları.

ŞUBE 1 (Kırmızı)	**MININET ADI**	**NUMARASI/MAC ADRESİ**	**IP ADRESİ**
Anahtar Cihazları (Merkez)	s110	openflow: 110	-
Anahtar Cihazları 1	s111	openflow: 111	-
Anahtar Cihazları 2	s112	openflow: 112	-
İstemci 11_1	h11_1	00:00:00:00:11:01	10.0.11.1
İstemci 11_2	h11_2	00:00:00:00:11:02	10.0.11.2
SBS1 Sunucusu	-	-	10.0.11.100
YTA1 Kontrolcüsü	-	-	10.0.11.200
ŞUBE 2 (Yeşil)	**MININET ADI**	**NUMARASI/MAC ADRESİ**	**IP ADRESİ**
Anahtar Cihazları (Merkez)	s120	openflow: 120	-
İstemci 12_1	h12_1	00:00:00:00:12:01	10.0.12.1
İstemci 12_2	h12_2	00:00:00:00:12:02	10.0.12.2
SBS2 Sunucusu	-	-	10.0.12.100

YTA2 Kontrolcüsü	-	-	10.0.12.200
ŞUBE 3 (Gri)	**MININET ADI**	**NUMARASI/MAC ADRESİ**	**IP ADRESİ**
Anahtar Cihazları (Merkez)	s130	openflow: 130	-
İstemci 13_1	h13_1	00:00:00:00:13:01	10.0.13.1
İstemci 13_2	h13_2	00:00:00:00:13:02	10.0.13.2
SBS3 Sunucusu	-	-	10.0.13.100
YTA3 Kontrolcüsü	-	-	10.0.13.200
USGM Sunucusu:			10.0.15.1

Bu ortamın Mininet ile oluşturulması için özel betik dili kullanılmış ve cihazların tüm IP/MAC adresleri ve bağlantı detayları tarif edilmiştir. Çizelge 5.2, Çizelge 5.3 ve Çizelge 5.4'de sırası ile tüm şubeler için oluşturulmuş betikleri bulabilirsiniz.

Çizelge 5.2 Şube 1'i oluşturmak için kullanılan betik (script11.py).

```
# Python Betigi Ilk Ag icin: SITE 1
from mininet.topo import Topo

class MyTopo( Topo ):
"Ag altyapisi 1'in olusturulmasi."

def __init__( self ):
"Topolojinin olusturulmasi"

# Topolojinin baslatilmasi.
Topo.__init__( self )

# Istemci ve Anahtarin eklenmesi
H11_1 = self.addHost(  'h11_1', ip= '10.0.11.1/21', mac= '00:00:00:00:11:01' )
H11_2 = self.addHost ( 'h11_2', ip= '10.0.11.2/21', mac= '00:00:00:00:11:02' )
S111 = self.addSwitch( 's111' )
S112 = self.addSwitch( 's112' )
S110 = self.addSwitch( 's110' )

# Baglantilarin eklenmesi
self.addLink( H11_1, S111 )
self.addLink( H11_2, S112 )
```

```
self.addLink( S110, S111 )
self.addLink( S110, S112 )

topos = { 'mytopo': ( lambda: MyTopo() ) }
```

Çizelge 5.3 Şube 2'nin oluşturulması için kullanılan betik (script12.py).

```
# Python Betigi Ilk Ag icin: SITE 12
from mininet.topo import Topoclass MyTopo( Topo ):
   "Ag altyapisi 2'in olusturulmasi."

   def __init__( self ):
      "Topolojinin olusturulmasi"

      # Topolojinin baslatilmasi.
      Topo.__init__( self )

      # Istemci ve Anahtarin eklenmesi
      H12_1 = self.addHost(  'h12_1', ip= '10.0.12.1/21', mac= '00:00:00:00:12:01' )
      H12_2 = self.addHost ( 'h12_2', ip= '10.0.12.2/21', mac= '00:00:00:00:12:02' )
      S120 = self.addSwitch( 's120' )

      # Baglantilarin eklenmesi
      self.addLink( H12_1, S120 )
      self.addLink( H12_2, S120 )

topos = { 'mytopo': ( lambda: MyTopo() ) }
```

Çizelge 5.4 Şube 3'ün oluşturulması için kullanılan betik (script13.py).

```
# Python Betigi Ilk Ag icin: SITE 13
from mininet.topo import Topo

class MyTopo( Topo ):
   "Ag altyapisi 3'un olusturulmasi."

   def __init__( self ):
      "Topolojinin olusturulmasi"

      # Topolojinin baslatilmasi.
      Topo.__init__( self )

      # Istemci ve Anahtarin eklenmesi
```

```
        H13_1 = self.addHost(  'h13_1', ip= '10.0.13.1/21', mac= '00:00:00:00:13:01' )
        H13_2 = self.addHost ( 'h13_2', ip= '10.0.13.2/21', mac= '00:00:00:00:13:02' )
        S130 = self.addSwitch( 's130' )

        # Baglantilarin eklenmesi
        self.addLink( H13_1, S130 )
        self.addLink( H13_2, S130 )

topos = { 'mytopo': ( lambda: MyTopo() ) }
```

Bu betikler ile kurulmuş 3 adet YTA kontrolcünün çalıştırılması için aşağıdaki Mininet başlatma komutları çalıştırılır.

Şube 1:

sudo mn --custom ./script11.py --topo mytopo1 --controller=remote, ip=10.0.11.200, port=6633

Şube 2:

sudo mn --custom ./script12.py --topo mytopo2 --controller=remote, ip=10.0.12.200, port=6633

Şube 3:

sudo mn --custom ./script12.py --topo mytopo2 --controller=remote, ip=10.0.12.200, port=6633

Bu komutlar ile beraber 3 ayrı ağ aktive olmuştur. Ancak bu ağlar arası haberleşme olabilmesi için köprü kurulması gerekmektedir. Mininet, GRE tünelleme tekniği kullanarak ağlar arası haberleşmenin gerçekleştirilmesine izin vermektedir. Bunun için şubelerde çalıştırılacak komutlar aşağıdaki gibidir.

Şube 1-2 Arası:

sh sudo ovs-vsctl add-port s110 s110-gre12 -- set interface s110-gre12 type=gre options:remote_ip=10.0.12.200

Şube 1-3 Arası:

sh sudo ovs-vsctl add-port s110 s110-gre13 -- set interface s110-gre13 type=gre options:remote_ip=10.0.13.200

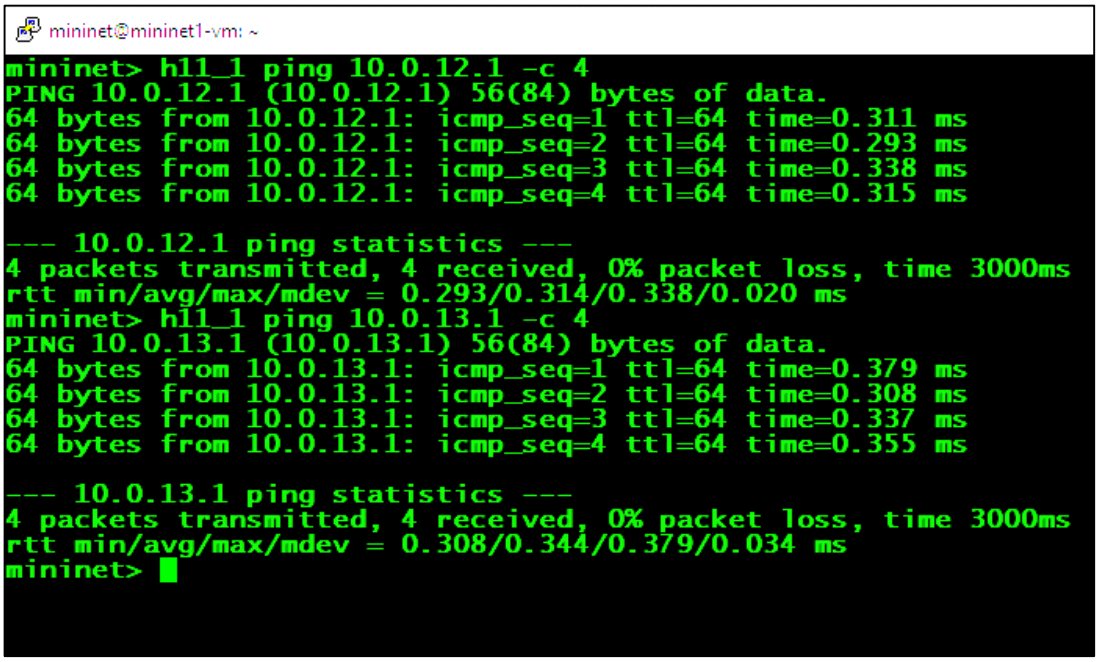
Şube 2-1 Arası:

sh sudo ovs-vsctl add-port s120 s120-gre21 -- set interface s120-gre21 type=gre options:remote_ip=10.0.11.200

Şube 3-1 Arası:

sh sudo ovs-vsctl add-port s130 s130-gre31 -- set interface s130-gre31 type=gre options:remote_ip=10.0.11.200

Bu aşamada; Şube 1, Şube 2 ve Şube 3 birbirine erişebilir hale gelmiştir (Şekil 5.14)



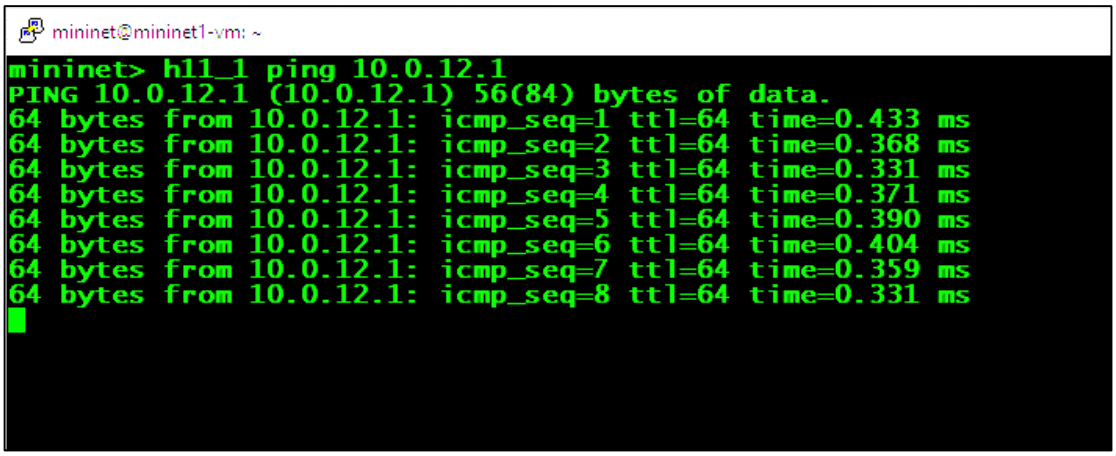
Şekil 5.14 Aktive edilen sistemde Şube 1 ile Şube 2'nin erişim testi.

5.3 USGM Sistemi Üzerinde Yapılan Testler

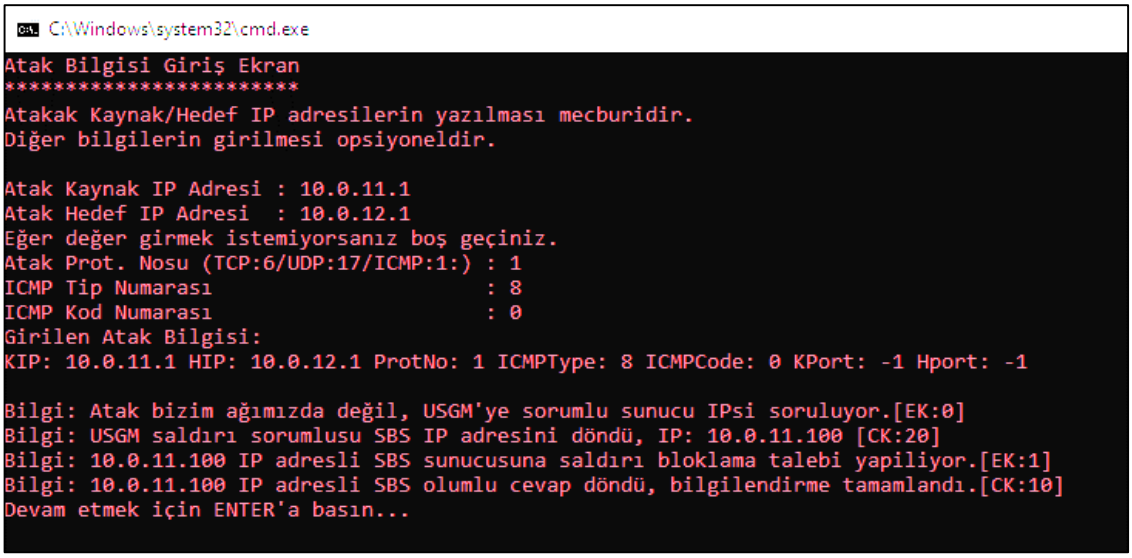
Bu aşamada geliştirilen sistemin, hedeflenen amaçları sağlayıp sağlamadığının belirlenmesi için testler uygulanmıştır.

5.3.1 Harici Bir SBS'den Atak Bilgisi Gelmesi (Test 1)

İlk testte Şube 1'deki istemci 11_1'den, istemci 12_1'e ping atağı başlatıldı (Şekil 5.15). SBS2'ye bu atağın detayları girildi ve SBS2 sunucusu atağın bloklanması ile ilgili süreci başlattı (Şekil 5.16).

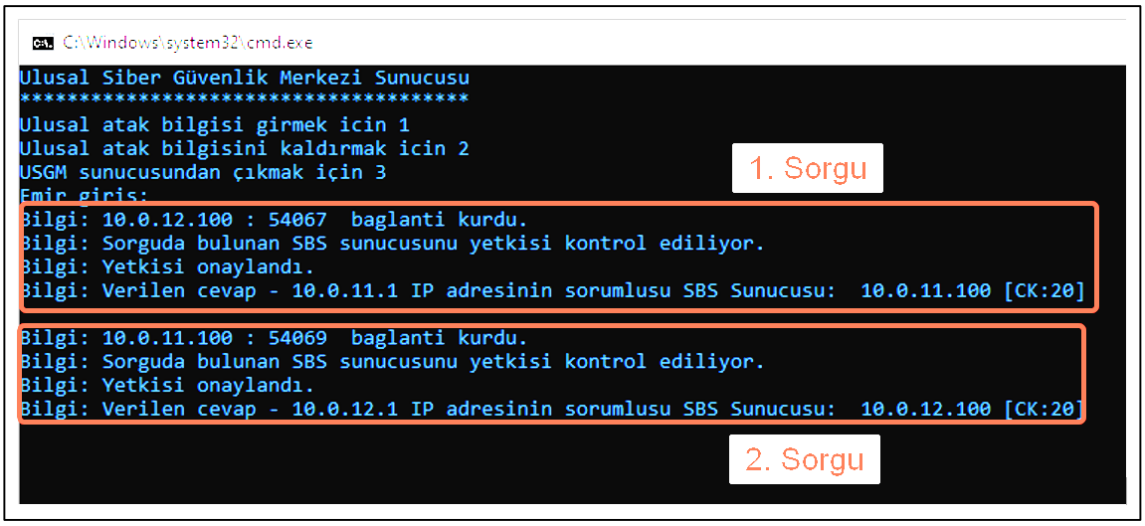


Şekil 5.15 İstemci 11_1'in ping trafiğini başlatması.



Şekil 5.16 SBS2'den İstemci 11_1'in ping trafiğinin bloklanma talebi.

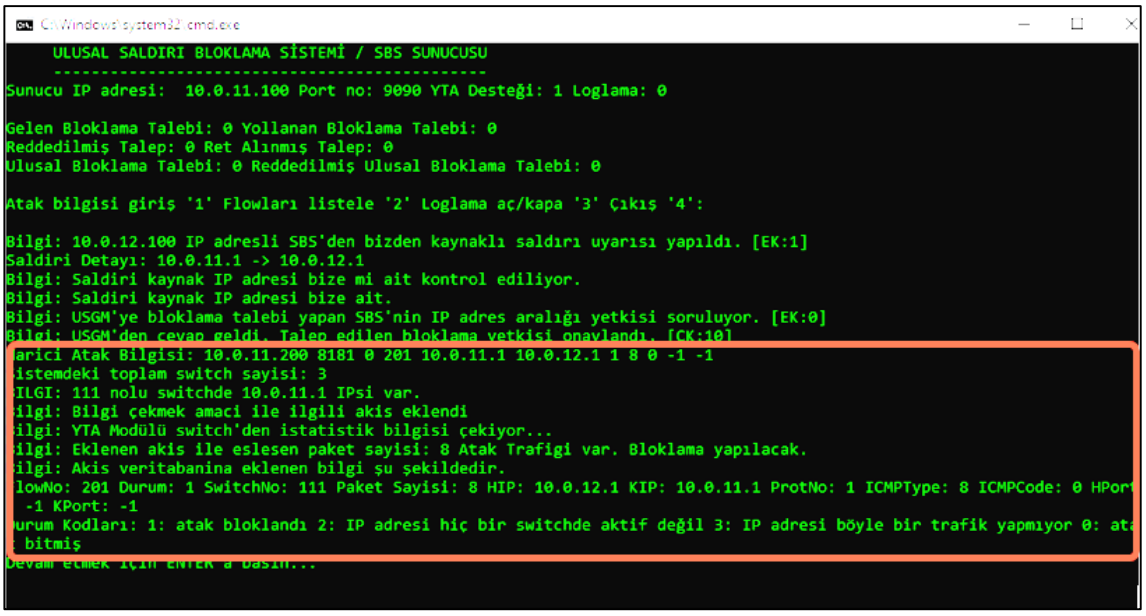
İlk önce SBS2, USGM sunucusuna istemci 11_1'in yani 10.0.11.1 IP adresinin kontrolünü yapan SBS sunucusunun IP adresini emir kodu 1 ile sordu. USGM sunucusu SBS2'ye ilgili SBS Sunucusunun IP adresini (10.0.11.100) cevap kodu 20 ile geri döndü (Şekil 5.17, 1.Sorgu).



Şekil 5.17 USGM sunucusunun sorgulara cevap vermesi.

SBS2 sunucusu, SBS1'in sahip olduğu IP adresine ping atağını kesmesi için gereken uyarı mesajını yolladı. SBS1, öncelikle talepte bulunan SBS2'nin durdurulmasını istediği atağın hedef IP adresinin yönetim hakkına sahip olup olmadığını kontrol etmek amacı ile USGM sunucusu sorgu yaptı. Yani atak hedef IP adresinin yetkili SBS sunucu IP adresini sorguladı. USGM sunucusu atak kaynak IP adresinin sorumlusu SBS IP adresini (10.0.12.100) cevap kodu 20 ile geri döndü (Şekil 5.17, 2.Sorgu). SBS1 sunucusu atak bloklama talebinde bulunan sunucunun yetkisi olduğunu belirleyerek atak bloklama sürecine girdi. SBS2 sunucusuna da bloklama talebinin kabul edildiğini bildiren cevap kodu 10 ile geri dönüş yaptı. Bu aşamada SBS1, YTA desteği olduğu için atak kaynak IP adresinin hangi anahtar cihazda olduğunu tespit etti (Şekil 5.18). Atak kaynak IP adresinin 111 nolu anahtar cihazda olduğunu belirledi. İkinci adım olarak da atağın halen daha devam edip etmediği tespit etmek amacı ile istatistik amaçlı atak trafik akışını ilgili anahtara ekledi. Bu akış sayesinde eşleşen paket sayısının arttığını tespit etti. Yani halen

daha atağın devam ettiğini anladı. Bu aşamada atağı bloklamak amaç ile ilgili anahtarda bloklama yapılacak şekilde bir akış ekledi ve atağı durdurdu. Ayrıca daha sonra bu atağın devam edip etmediğinin kontrol edilebilmesi için bu akış ile ilgili bilgileri akış veri tabına eklendi.



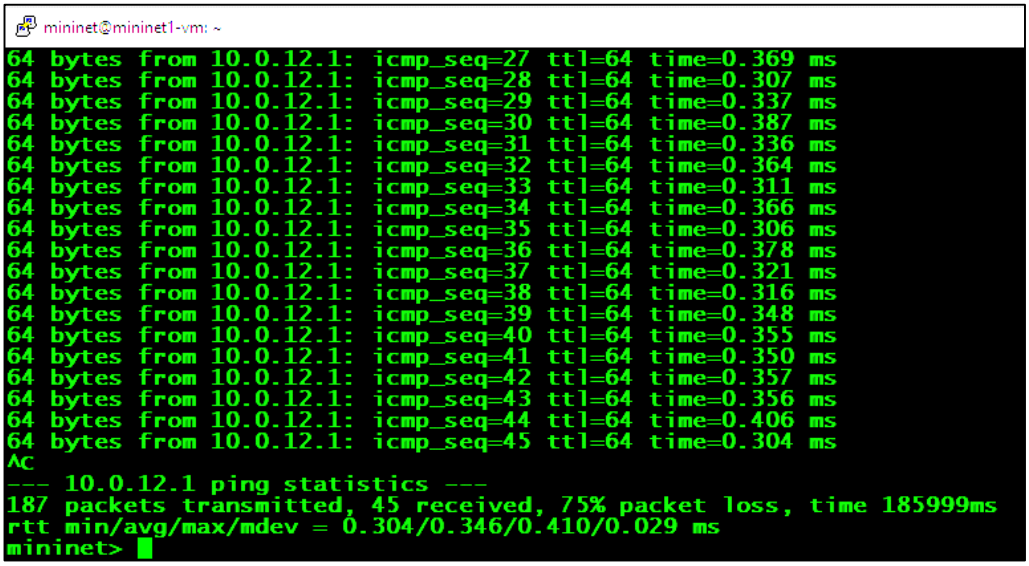
Şekil 5.18 SBS1'de gelen atak uyarısı üzerine yapılan bloklama işlemleri.

Çalışmanın hedeflerini test etmek amacı ile istemci 12_1'e diğer istemcilerden gelecek trafiğin devam edip etmediği test edildi. İstemci 11_2 ve istemci 13_1'in istemci 12_1'e erişebildiği görüldü. Hatta atak kaynağı istemci 11_1, istemci 12_1'ye sadece ping ile erişemediği, diğer portlardan istemci 12_1'e eriştiği de tespit edildi.

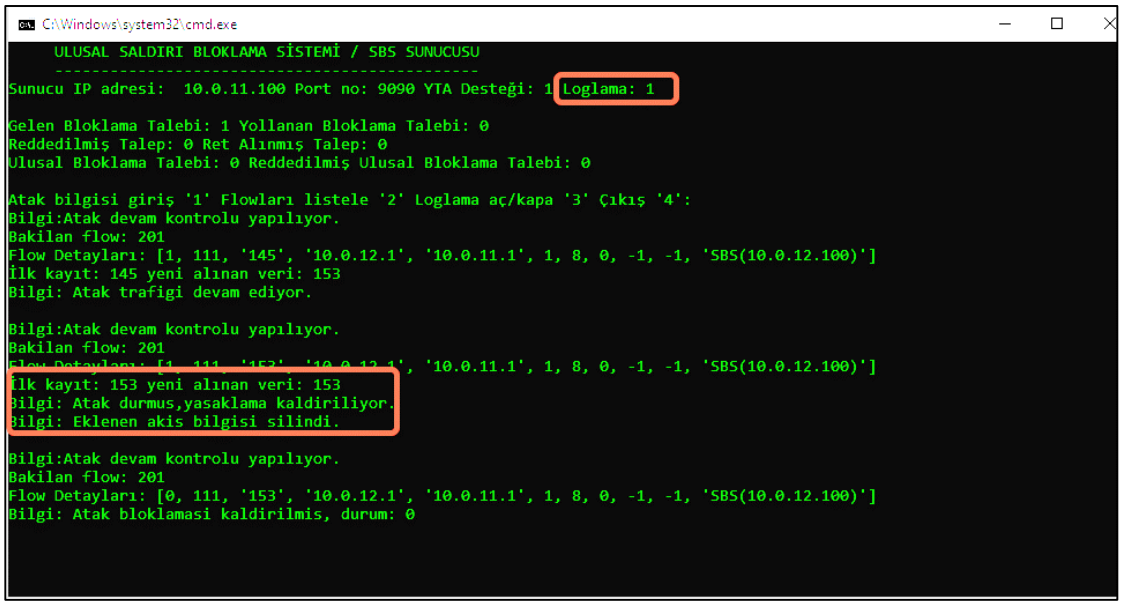
5.3.2 Atak Durduğunda Atak Bloklanmasının Otomatik Kaldırılması (Test 2)

Bu test de, ilk testte istemci 11_1'in gerçekleştirdiği ping atağı durdurulup, SBS1'in atağın durduğunu anlayıp eklenmiş bloklama akış bilgisini kaldırıp kaldırmayacağı kontrol edildi. Sürecin gözlenebilmesi için ilk menüden loglama özelliği açıldı. Şube 1'den devam ettirilen ping atak trafiği durduruldu (Şekil 5.19). Geliştirilen sistemde SBS sunucuları belirlenmiş periyotlarda atak trafiğin devam edip etmediğini

kontrol etmektedirler. Loglama özelliği açılmış SBS1 sunucusu Şekil 5.20'de görüldüğü gibi atak trafiğinin durduğunu gözlemledi ve bloklamayı kaldırdı. İstemci 11_1'in tekrar ping trafiğine devam edebildiği görüldü.



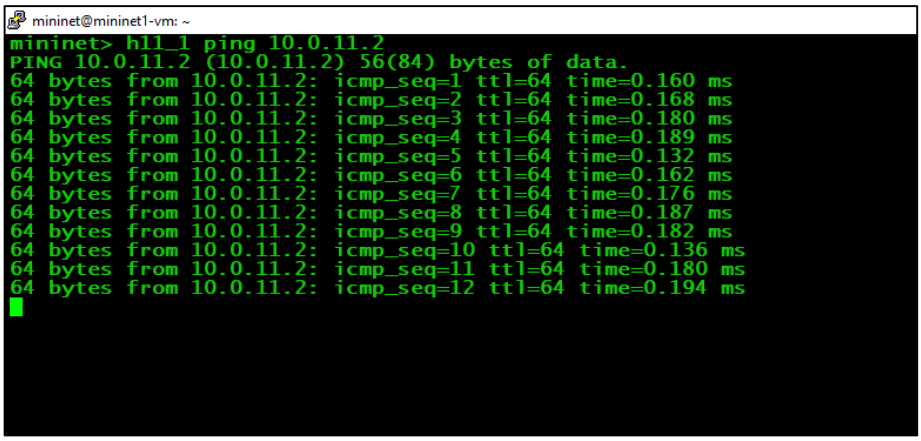
Şekil 5.19 İstemci 11_1'de ping trafiğinin durdurulması.



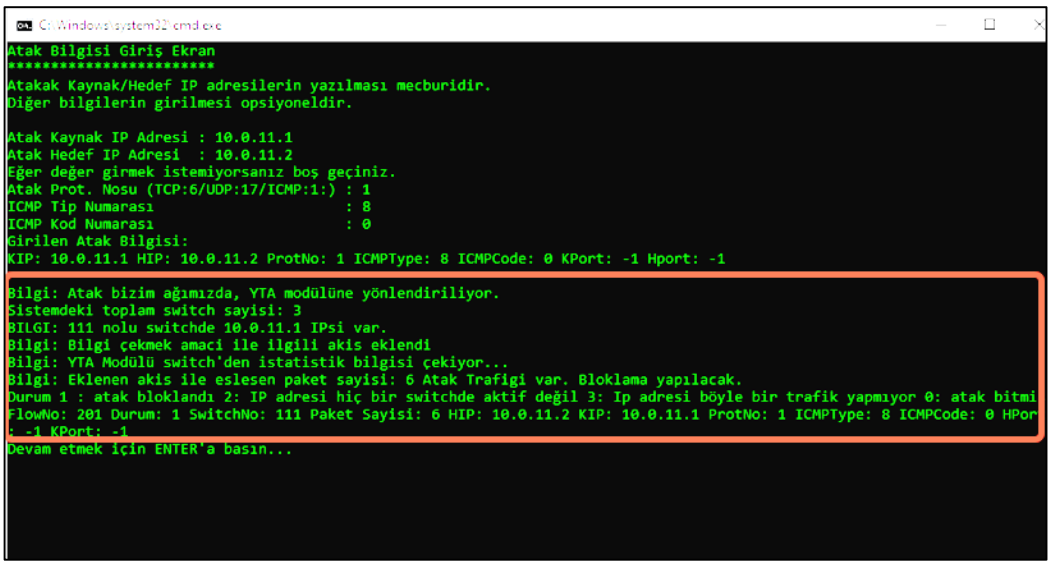
Şekil 5.20 SBS1 sunucusunun daha önce eklediği bloklamayı otomatik kaldırması.

5.3.3 Atağın Yerel Bir IP Adresinden Gelmesi (Test 3)

Bu aşamada, atak trafiğinin aynı ağ içerisinde gerçekleşme durumu test edilmiştir. Bu amaç ile istemci 11_1'den istemci 11_2'ye yani 10.0.11.2'ye doğru ping trafiği başlatılmıştır (Şekil 5.21).

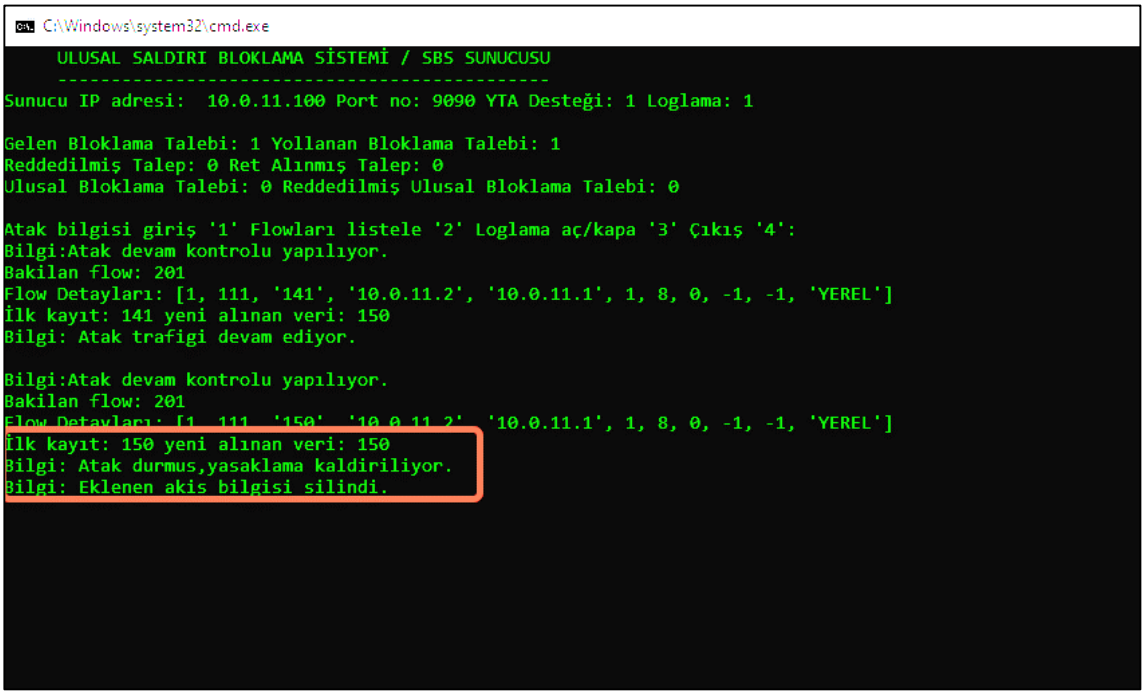


Şekil 5.21 İstemci 11_1'den aynı ağdaki istemci 11_2'ye doğru ping trafiği.



Şekil 5.22 SBS1 sunucusunun atağın yerel olduğunu tespiti ve atağı bloklanması.

İlgili atak trafiğinin bloklama talebi SBS1'de girilince, sunucu atağın kendi ağında olduğunu tespit etti (Şekil 5.22) ve YTA modülünü başlattı. Trafik kaynağı olan anahtar cihazı tespit etti ve atağın halen devam edip etmediğini kontrol etti. Devam ettiğini gözlemleyince atağı durdurdu. Daha sonra istemci 11_1'den gerçekleştirilen ping atağı durduruldu. Bunun sonrasında SBS1, atak trafiğinin durduğunu tespit ettiği ve atak bloklamasını kaldırdı (Şekil 5.23).

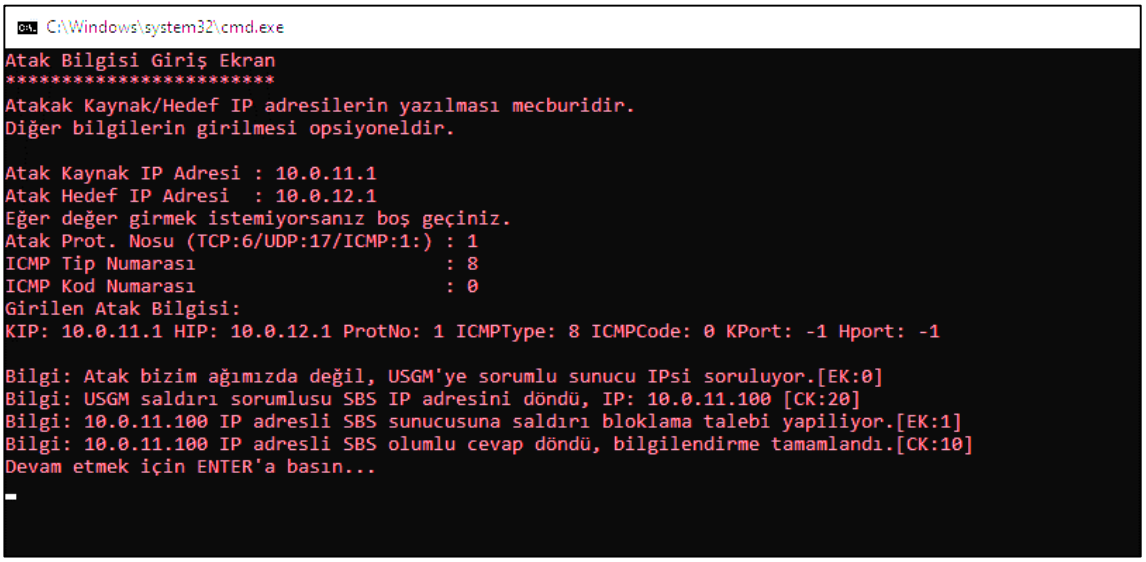


Şekil 5.23 SBS1 sunucusunun daha önce eklediği bloklamayı otomatik kaldırması.

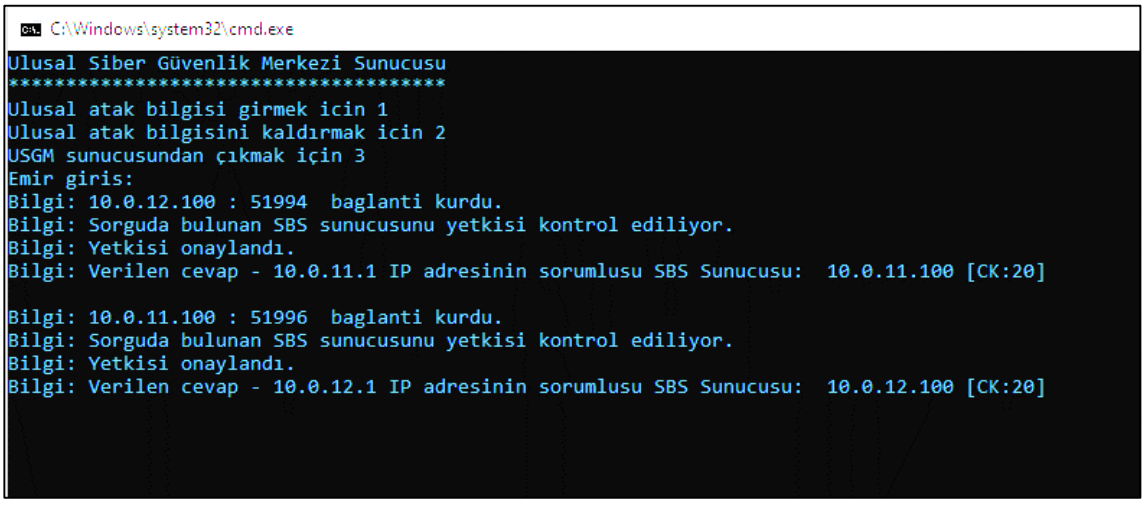
5.3.4 Belirtilen Atak Trafiğin Olmaması (Test 4)

Bu testte atak trafiği olmamasına rağmen atak bloklama talebi gelmesi durumu incelenmiştir. Bu amaçla SBS2 sunucusuna istemci 11_1'den istemci 12_1'e bir atak yapıldığı kaydı girildi (Şekil 5.24). Önce SBS2, USGM sunucusundan atak kaynağı SBS1'in IP adresini öğrendi (Şekil 5.25). SBS1 sunucusu talep edilen bloklama kaynağının kendisine ait bir IP adresi için bloklama talebi yapıp yapmadığını doğrulandı. SBS1 sunucusu SBS2 sunucusuna cevap kodu 10 ile olumlu cevap döndü. Sonrasında YTA kontrolcüsü yardımı ile atak trafiğinin halen daha devam edip etmediğini sorguladı.

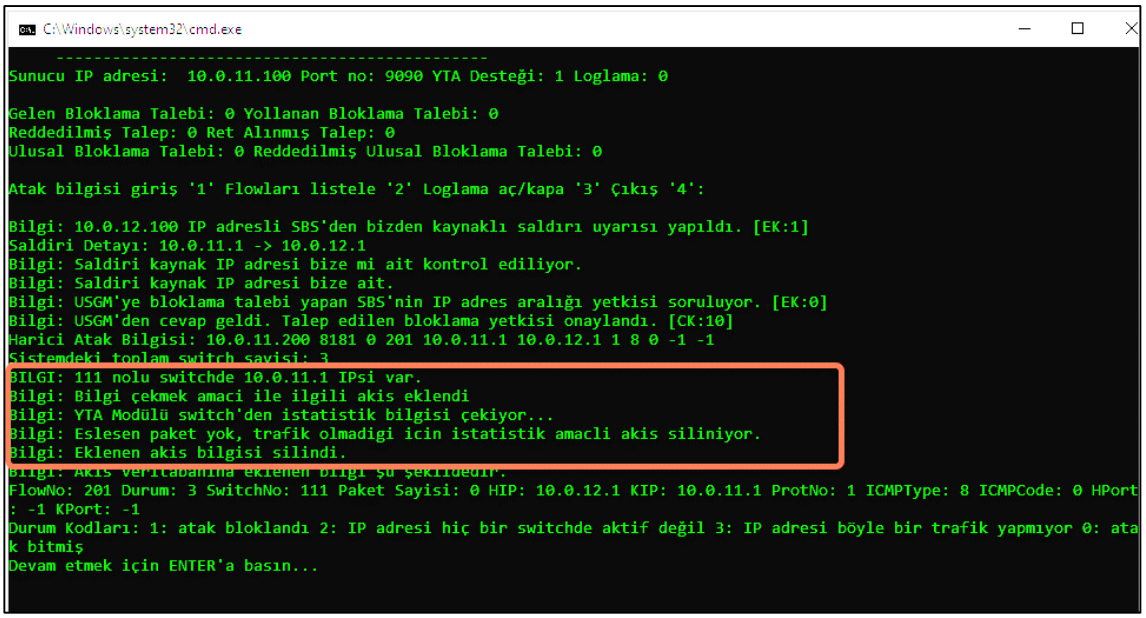
Bunun için SBS1 sunucusu öncelikle atak kaynak IP adresinin hangi anahtar cihaz üzerinde olduğunu belirleme sürecine girdi. Sonuç olarak IP adresinin 111 nolu anahtarda aktif olduğunu belirledi. Bir sonraki adım olarak atak kaynak IP adresinin belirtilen şekilde bir trafik oluşturup oluşturmadığının tespit sürecine girildi. Atak trafiğinin mevcut durumda olmadığını belirledi ve bloklama işlemini gerçekleştirmedi (Şekil 5.26). Talep akış veri tabanına durum 3 olarak yani atak kaynak IP adresinin böyle bir atak gerçekleştirmediği şeklinde bir satır olarak ekledi.



Şekil 5.24 SBS2 sunucusuna olmayan bir atak bilgisi girilmesi.



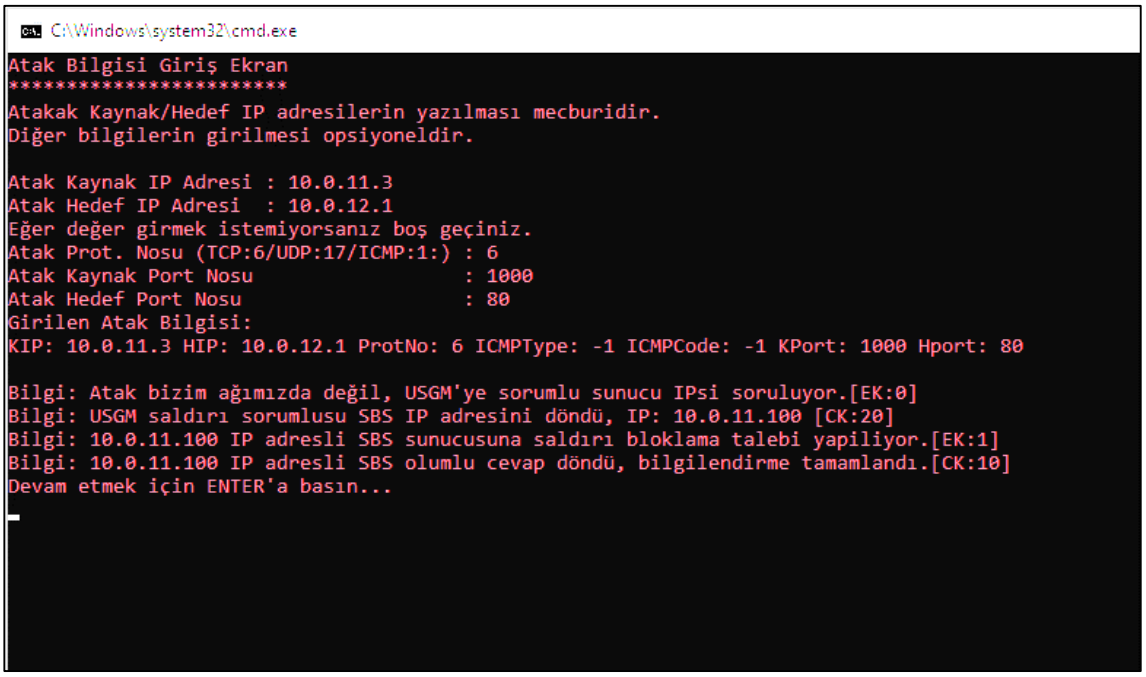
Şekil 5.25 Atak kaynak SBS sunucuları için gereken sorguların USGM'ye yapılması.



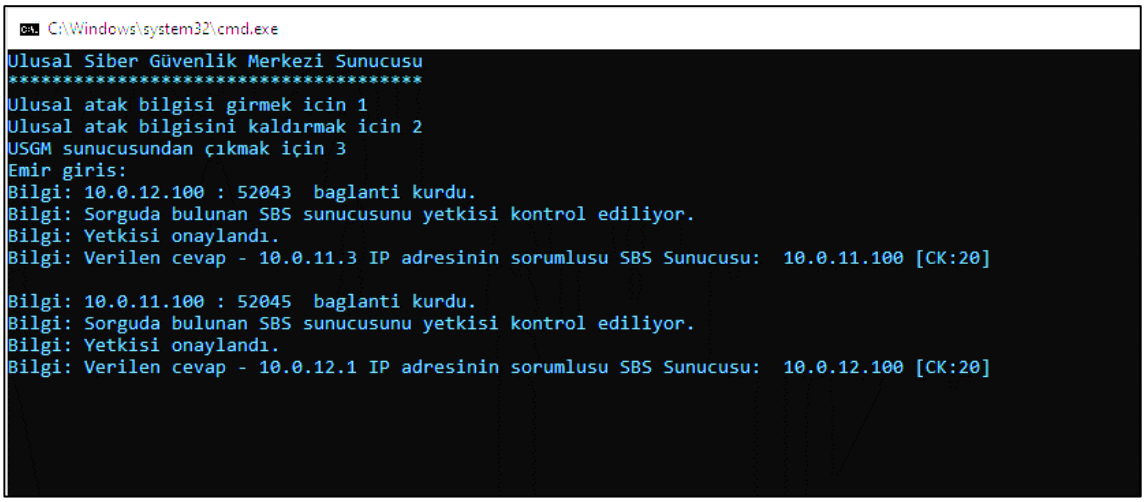
Şekil 5.26 SBS1 sunucunun belirtilen atağın gerçekleşmediğinin tespiti.

5.3.5 Atak Kaynak IP Adresinin Kullanılmıyor Olması (Test 5)

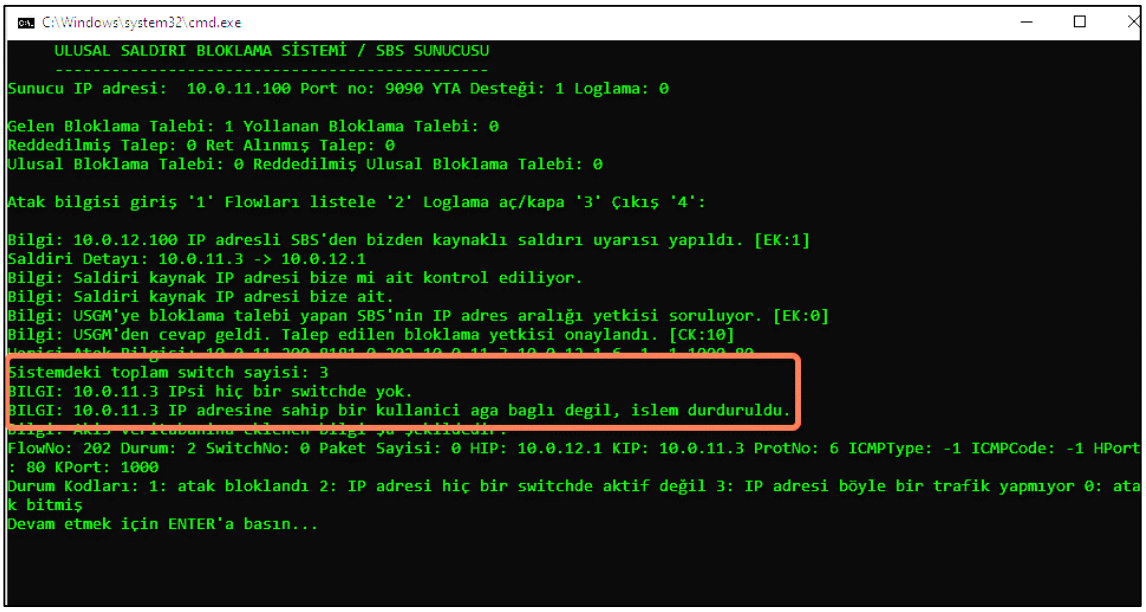
Bu testte belirtilen SBS2 sunucusun SBS1'e ait olan ama hali hazırda kullanılmayan bir IP adresine atak trafiği ile ilgili bilgilendirme yapması durumu incelenmiştir. Atak bilgisi SBS2 sunucusuna girildikten sonra, SBS1 USGM sunucusundan atak kaynağı bilgisini sorgulandı ve IP adresinin SBS1'in kontrolü altına olduğunu belirledi (Şekil 5.27, Şekil 5.28). SBS1 sunucusu YTA kontrolcüsü üzerinden atağı durdurma işlemlerini başlattı. Bu amaçla ilk olarak SBS1 sunucusu atak kaynak IP adresinin hangi anahtar cihazında olduğunu belirleme sürecine girdi. Yaptığı analiz sonrasında ilgili IP adresinin şu anda ağda hiçbir anahtar cihaz üzerinde aktif olmadığını belirledi ve hiçbir bloklama işlemi gerçekleştirmedi (Şekil 5.29). Gelen talebi akış veri tabanına durum kodu 2 olarak yani IP adresinin hiçbir anahtar cihazında aktif olmadığı şeklinde işledi.



Şekil 5.27 SBS2 sunucusunda aktif olmayan bir IP adresi için atak bilgisi girilmesi.



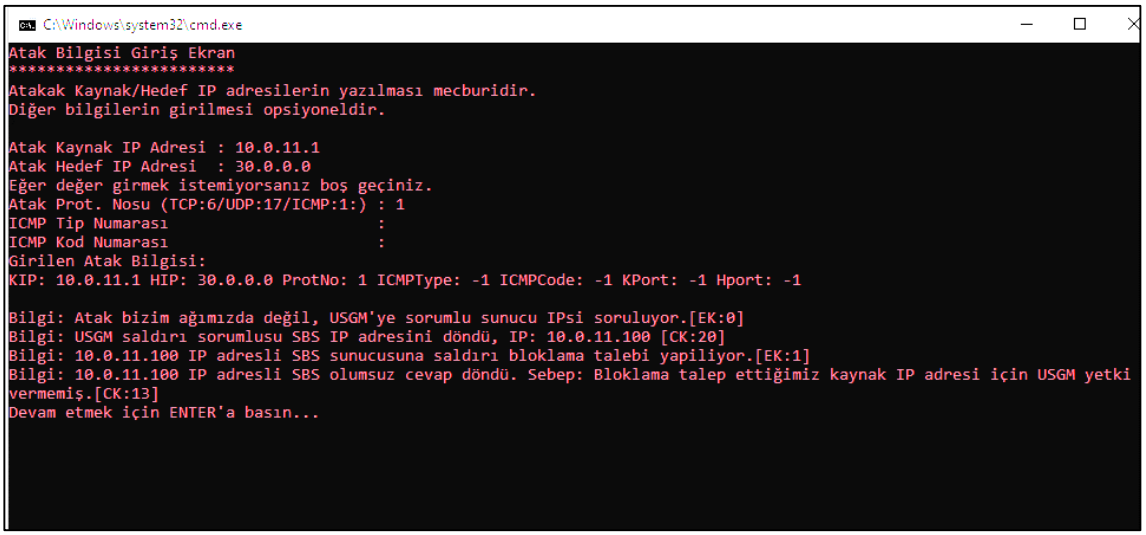
Şekil 5.28 Atak kaynak SBS sunucuları için gereken sorguların USGM'ye yapılması.



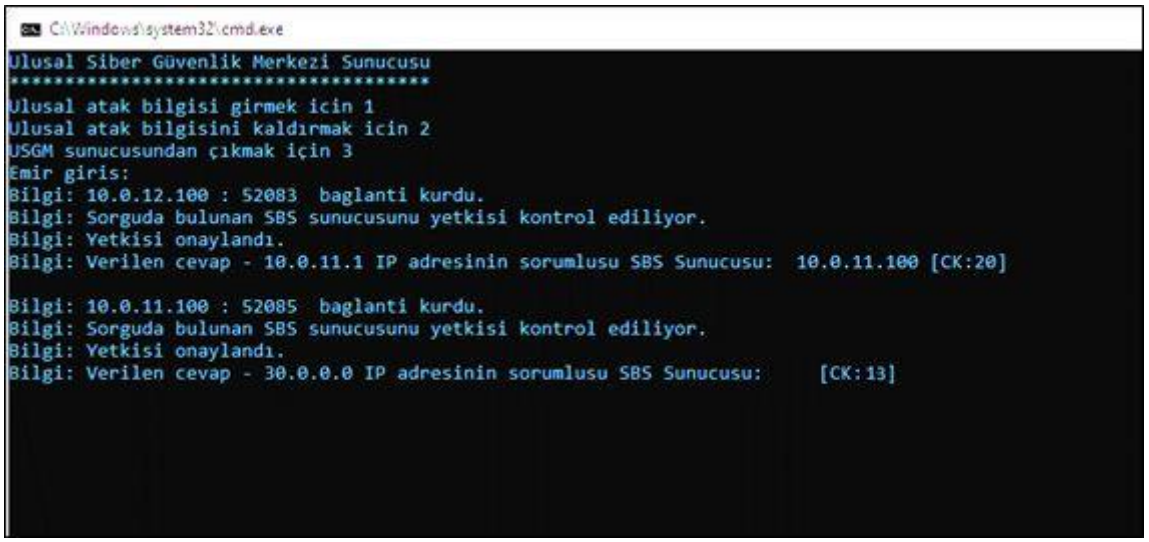
Şekil 5.29 SBS1'de atak kaynak IP adresinin kullanılmadığı tespiti.

5.3.6 Kendisine Ait Olmayan Bir IP Adresi için Atak Bilgilendirmesi (Test 6)

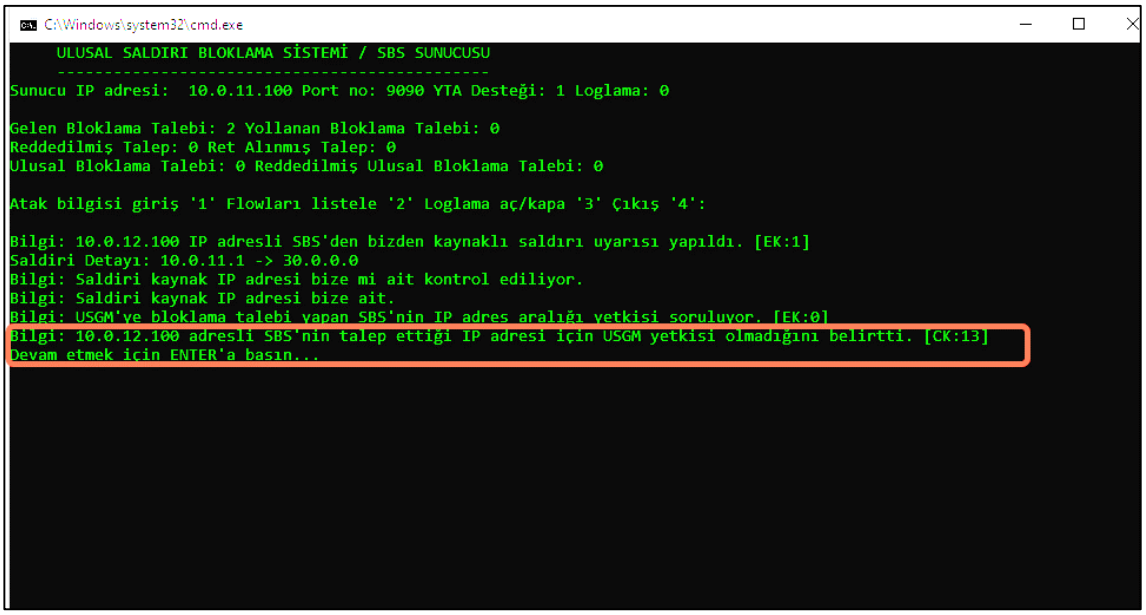
Bu testte SBS2 sunucusunun, kendisine ait olmayan bir IP adresi için saldırı bloklama talebinde bulunması durumu incelenmiştir. Buna göre SBS2, kaynak IP adresi SBS1'e ait olan ancak hedef IP adresi SBS2'nin sorumluluğunda olmayan bir IP adresi ile (30.0.0.1) atak bloklama talebinde bulundu (Şekil 5.30). USGM sunucusu atak kaynak IP adresinin SBS1'de olduğunu belirtti ve SBS2 talebini SBS1'e ulaştırdı (Şekil 5.31). Atak bloklama işlemi yapmadan önce SBS1 sunucusu atak hedef IP adresinin SBS2'ye ait olup olmadığını kontrol amacı ile USGM sunucusuna sorguda bulundu. USGM sunucusunda olumsuz cevap geldiği için uyarıda bulunan SBS2 sunucusuna cevap kodu 13 ile yani bloklama talep edilen hedef IP adresi için USGM sunucusu yetki vermediği gerekçesi ile talebin reddedildiğini geri döndü (Şekil 5.32).



Şekil 5.30 SBS2 sunucusunda kendisine ait olmayan bir IP adresi için atak bilgisi girilmesi.



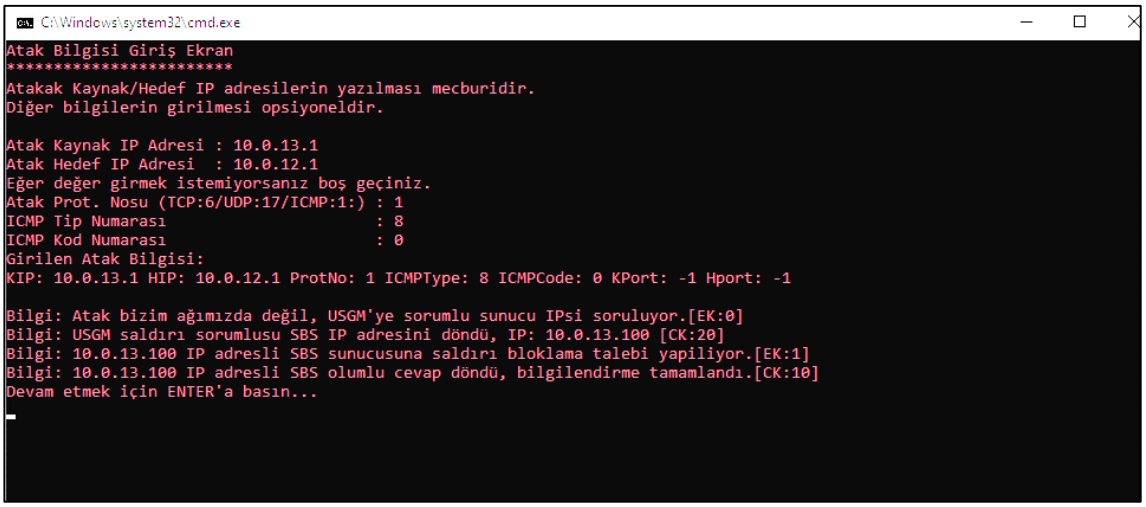
Şekil 5.31 Atak kaynak SBS sunucuları için gereken sorguların USGM sunucusuna yapılması.



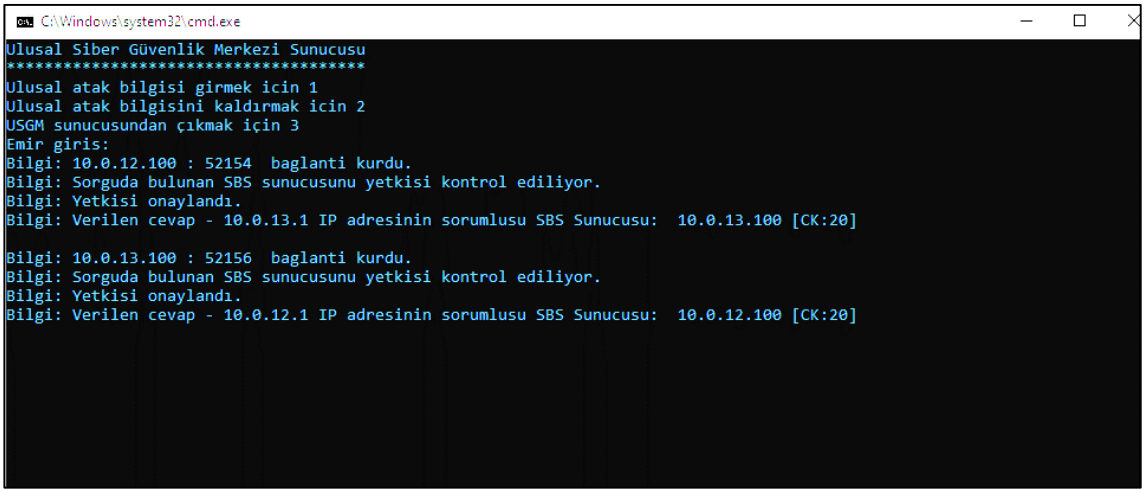
Şekil 5.32 SBS1'de atak hedef IP adresinin SBS2'ye ait olmadığı tespiti.

5.3.7 YTA Desteği Olmayan Şubeye Atak Bilgilendirmesi Gelmesi (Test 7)

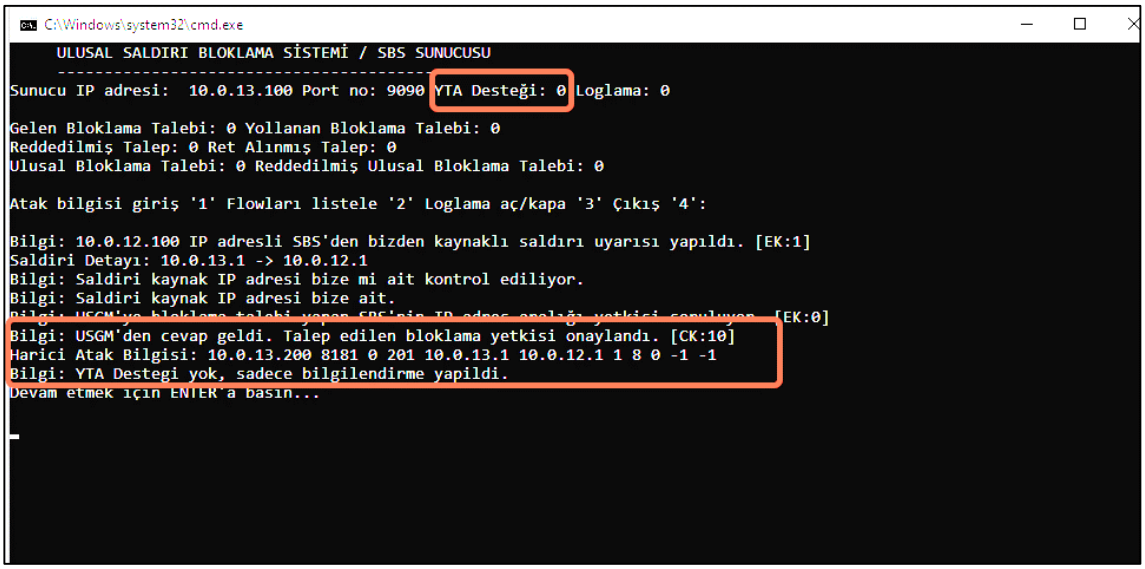
Bu kez amaçlarda belirtildiği gibi YTA desteği olmayan bir ağ altyapısında da bu sistemin kullanılabilmesi test edildi. SBS2 sunucusu SBS3 (YTA desteği olmayan ağ) sunucusunun yetki alanında bulunan 10.0.13.1 IP adresi için bloklama talebinde bulundu (Şekil 5.33). USGM sunucusunda gerekli onaylar alındıktan sonra SBS3 sunucusu bloklama talebini aldı ancak YTA desteği olmadığı için saldırı ile ilgili sadece bilgilendirme yaptı (Şekil 5.34, Şekil 5.35).



Şekil 5.33 SBS2'den YTA desteği olmayan bir şubeye atak uyarısı yapması.



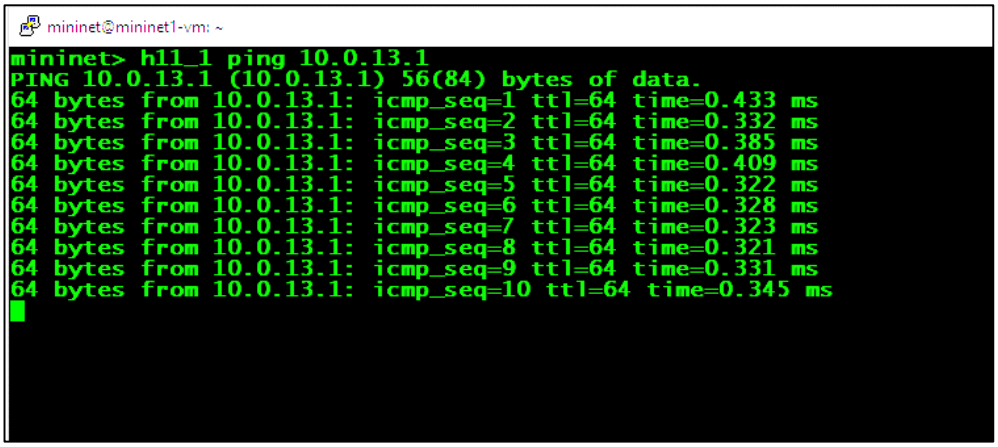
Şekil 5.34 Atak kaynak SBS sunucuları için gereken sorguların USGM sunucusuna yapılması.



Şekil 5.35 YTA desteği olmayan SBS3 sunucunun atak bilgilendirmesini yapması.

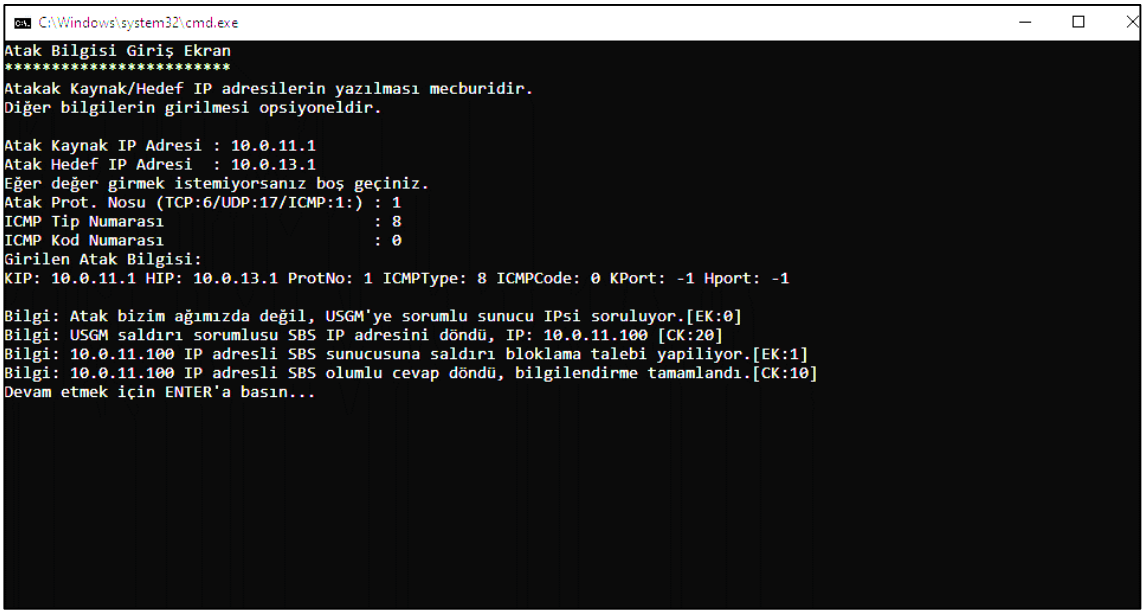
5.3.8 YTA Desteği Olmayan Şubeden Atak Bilgilendirmesi Yapılması (Test 8)

Bu test ile YTA desteği olmayan bir ağın atak bilgilendirmesi yapması ve sonrasında YTA desteği olan bir ağın ilgili atağı bloklaması test edildi. Bu amaçla önce YTA desteği olan bir ağdaki istemci 11_1'den YTA desteği olmana bir ağdaki istemci 13_1'e doğru ping trafiği başlatıldı (Şekil 5.36).

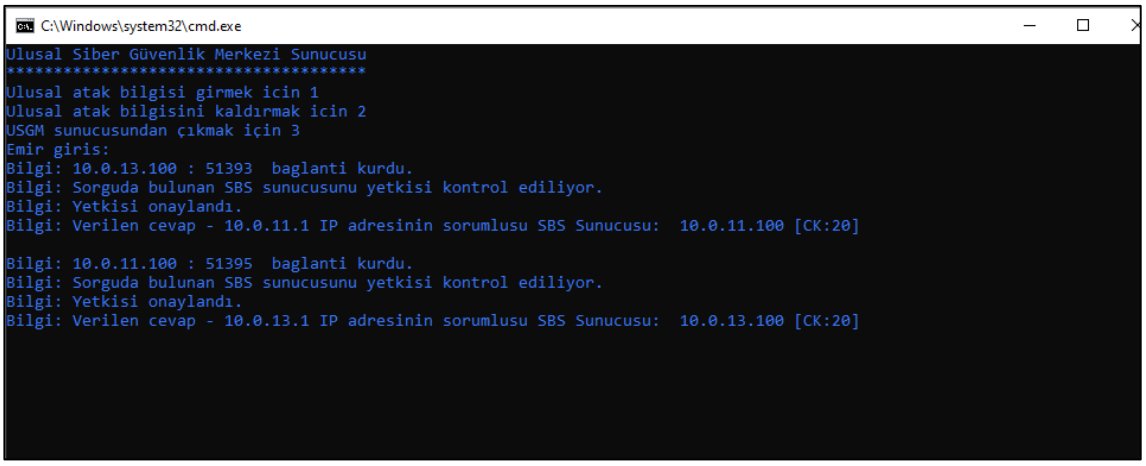


Şekil 5.36 İstemci 11_1'den YTA desteği olmayan bir ağda olan istemci 13_1'e başlatılan ping trafiği.

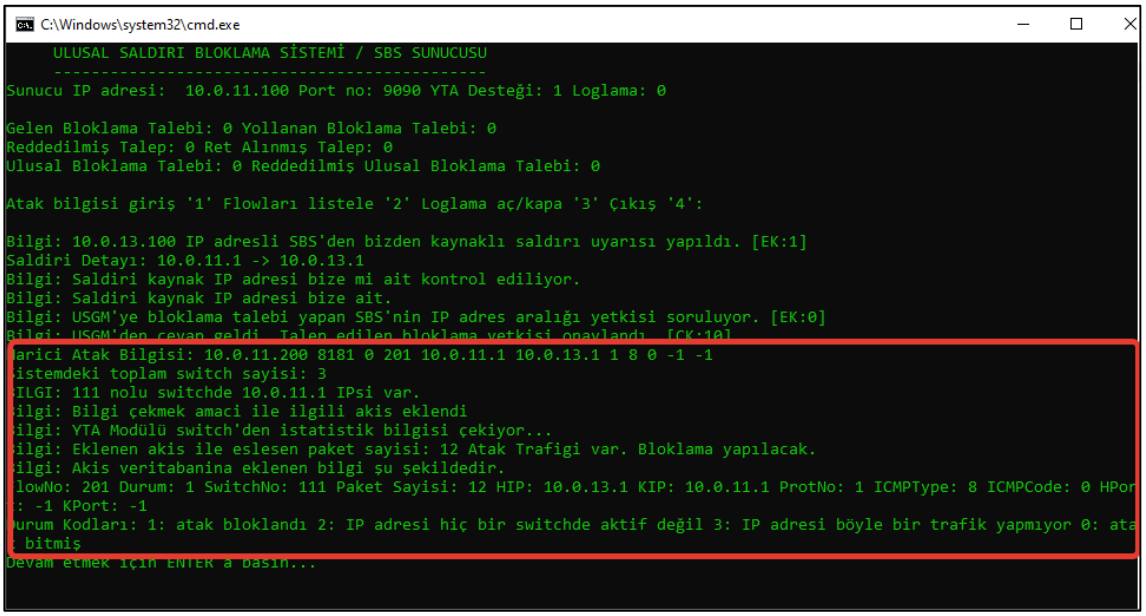
Daha sonra YTA desteği olmayan SBS3'de atak bilgileri girildi (Şekil 5.37). SBS3 atak kaynağının kendi ağında olmadığı tespit etti ve atak kaynağının sorumlusu SBS'yi tespit etmek amacı ile USGM sunucusuna sorguda bulundu (Şekil 5.38). Atak kaynak SBS sunucusunun IP adresini (10.0.11.100) yani SBS1'in IP adresini öğrendi. SBS1 atak hedef IP adresinin talepte bulunan SBS'ye ait olup olmadığını tespit etmek amacı ile USGM sunucusuna sorguda bulundu. Sorgu sonrasında talebi yapan SBS sunucusunun yetkiye sahip olduğunu belirledi ve YTA modülü ile saldırı bloklama sürecine geçti. Öncelikle kaynak IP adresinin 111 nolu anahtarda olduğunu belirledi (Şekil 5.39). Daha sonra atağın halen daha devam edip etmediğini belirlemek amacı ile 111 nolu anahtardan ilgili akış hakkında trafik istatistiği temin etti. Atağın devam ettiğini gözlemleyince atağın bloklanması için gereken akış bilgisini ilgili anahtar cihaza yolladı. Kontrol amacı ile de bu akışın detaylarını, akış veri tabanına kayıt etti.



Şekil 5.37 YTA desteği olmayan SBS3'den atak bilgilendirmesi yapılması.



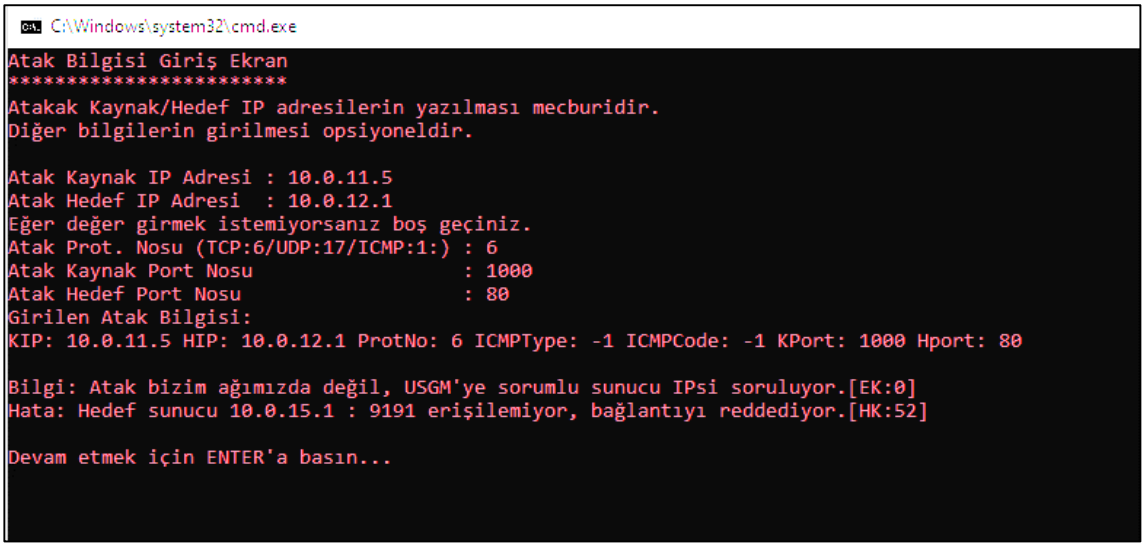
Şekil 5.38 Atak kaynak SBS sunucuları için gereken sorguların USGM sunucusuna yapılması.



Şekil 5.39 SBS1'de gelen atak uyarısı üzerine yapılan bloklama işlemleri.

5.3.9 USGM Sunucusuna Erişilememe Durumu (Test 9)

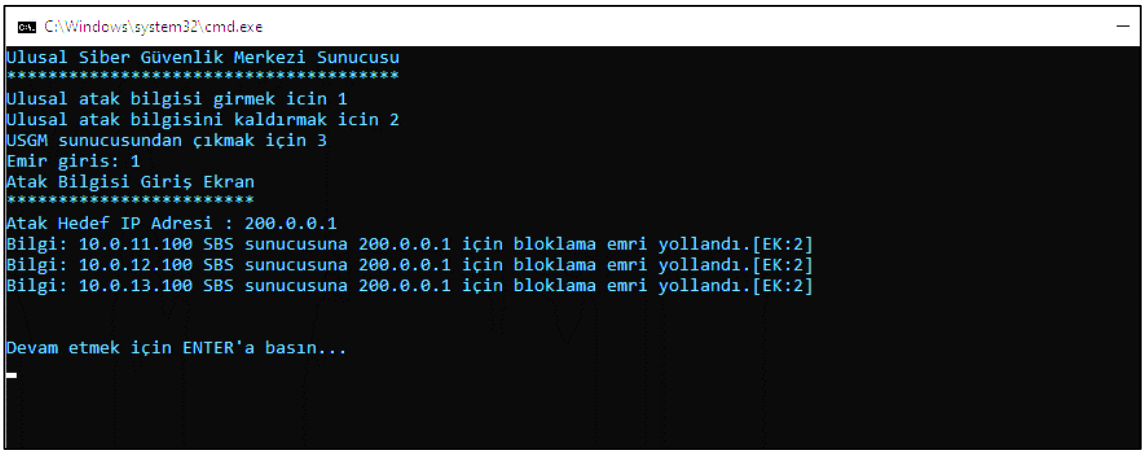
Bu testte USGM sunucusuna bir şekilde erişilememe durumu incelendi. SBS2 sunucusu 10.0.11.5 IP adresinin saldırıda bulunduğuna dair bilgilendirme yaptı (Şekil 5.40). USGM sunucusu devreye alınmağı için bu sunucuya erişilemedi. SBS2 sunucusu hata kodu 52 ile yani USGM sunucusuna erişilemediğini rapor etti.



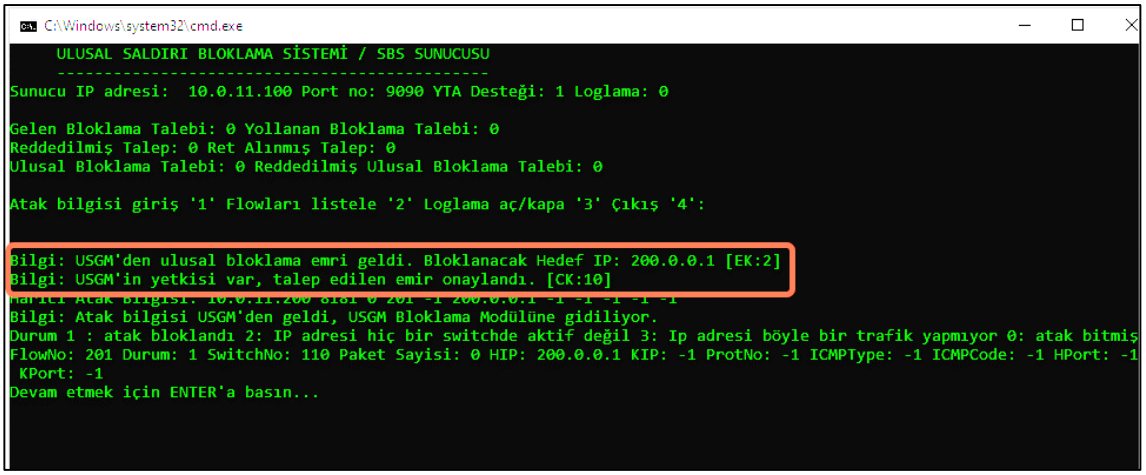
Şekil 5.40 SBS2'in USGM sunucusuna erişememe durumu.

5.3.10 USGM Sunucusundan Ulusal Atak Bloklama Emrinin Gelmesi (Test 10)

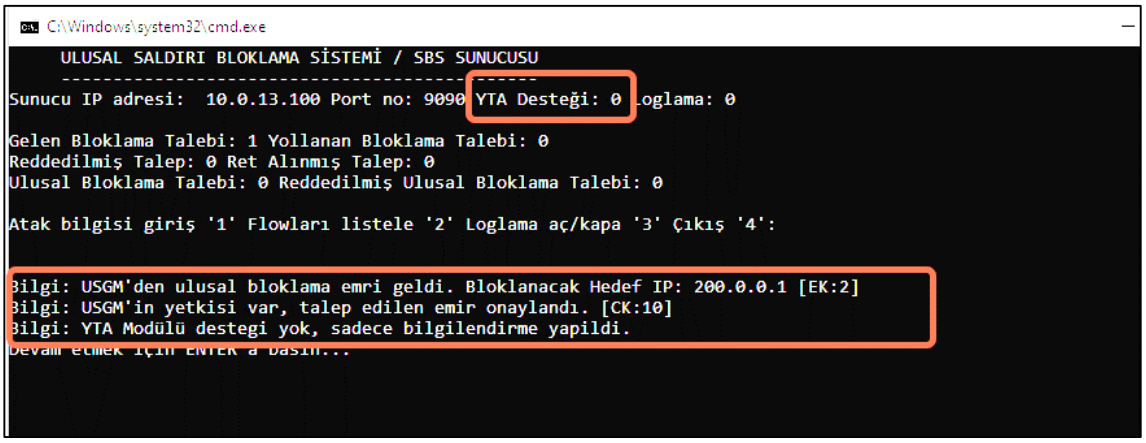
Çalışmanın amaçlardan başka biri olan ulusal seviyede saldırı bloklama işlemi test edildi. Bu amaç ile öncelikle USGM sunucusunda hedef IP adresi 200.0.0.1 olan tüm trafiğin SBS sunucularında bloklanması talep edildi (Şekil 5.41). Talep SBS1, SBS2 ve SBS3 sunucusuna ulaştırıldı. YTA desteği olan SBS1 ve SBS2 talep USGM'den geldiği için hiçbir kontrol yapmadan bloklama işlemini gerçekleştirdi. Sonrasında akış veri tabanına atak talebin USGM'den geldiği ve trafiğin bloklandığı şeklinde bir kayıt düşüldü (Şekil 5.42). SBS3, YTA desteğine sahip olmadığı için talep ile ilgili sadece bilgilendirme yaptı (Şekil 5.43).



Şekil 5.41 USGM sunucusundan ulusal atak bloklama talebi yapılması.



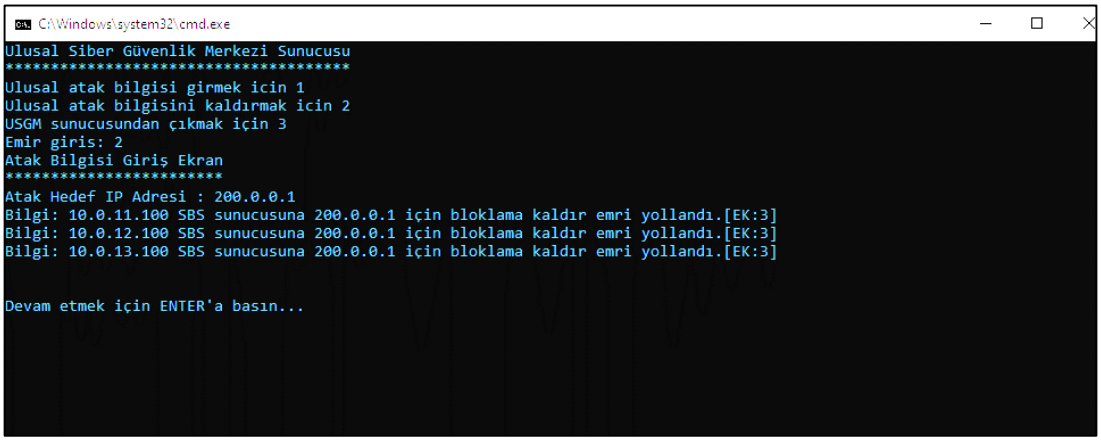
Şekil 5.42 SBS1'de USGM sunucusundan gelen bloklama talebinin uygulanması.



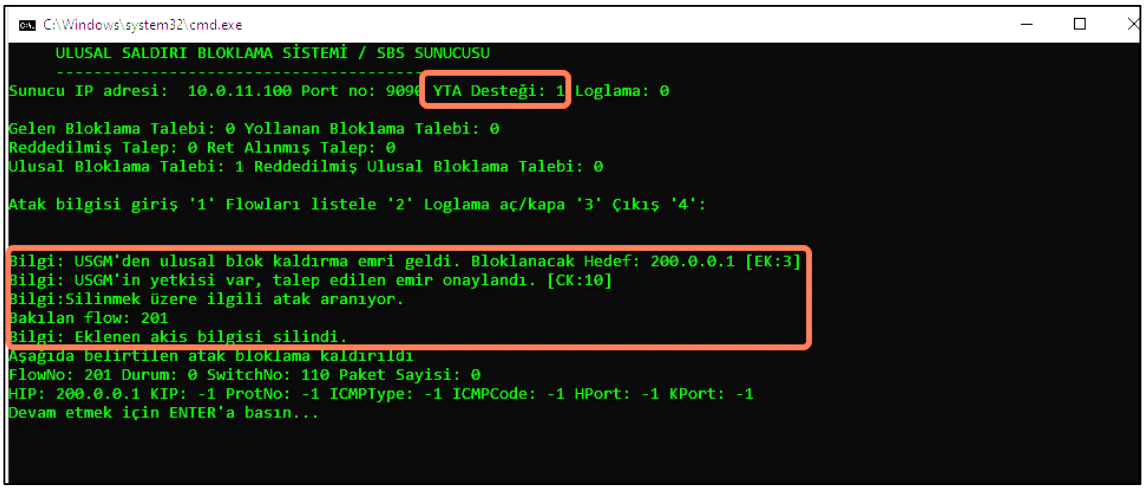
Şekil 5.43 YTA desteği olmayan SBS3'de USGM sunucusundan gelen bloklama talebi ile ilgili bilgilendirme.

5.3.11 USGM Sunucusunda Atak Blok Kaldırma Emrinin Gelmesi (Test 11)

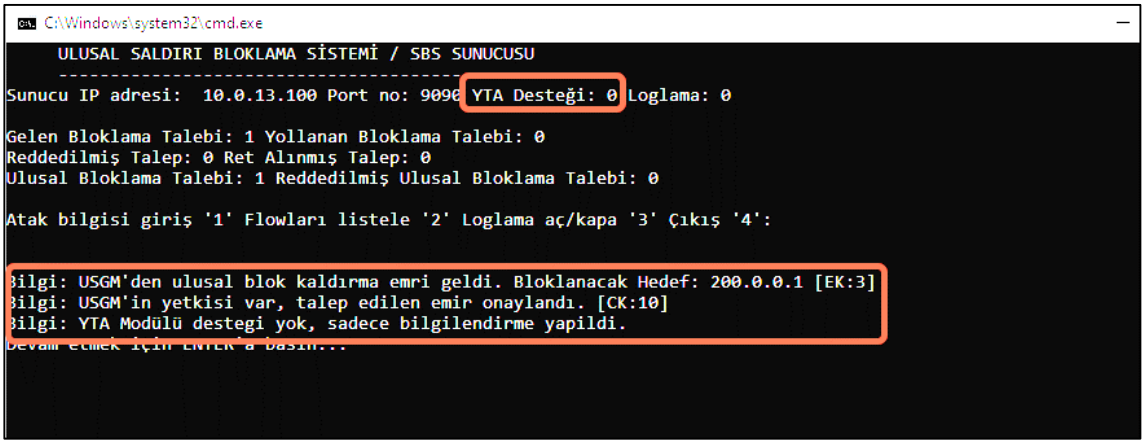
Bu kez ulusal seviyede daha önce yapılmış olan atak bloklama talebinin kaldırılması incelendi. USGM sunucusundan, 200.0.0.1 IP adresi için ulusal blok kaldırma talebinde bulunuldu (Şekil 5.44). SBS1 ve SBS2, YTA desteği sayesinde daha önce bloklamak amaçlı ekledikleri akışı otomatik kaldırdı (Şekil 5.45). Son durumda da akış veri tabanı, bloklama kaldırıldı şeklide revize edildi. SBS3 sunucusu ise YTA desteği olmadığı için durum ile ilgili bilgilendirme yaptı (Şekil 5.46).



Şekil 5.44 USGM sunucusundan ulusal atak blok kaldırma talebi yapılması.



Şekil 5.45 SBS1'de USGM sunucusundan gelen blok kaldırma talebinin uygulanması.



Şekil 5.46 YTA desteği olmayan SBS3'de USGM sunucusundan gelen blok kaldırma talebi ile ilgili bilgilendirme.

5.4 Ulaşılan Hedefler

Testler sonrasında çalışma dahilinde belirlenmiş amaçlardan ulaşılanlar şu şekildedir.

1- Çalışmada ki hedeflerden biri BGP veya benzeri bir yönlendirme protokolü desteği olmayan yerlerde de kullanılabilmesi idi. Oluşturulan test ortamında hiçbir yönlendirme protokolü kullanılmamıştır. Sistem geliştirilen protokol ile hedefleri sağlamıştır.

2- Test ortamında Internet Servis Sağlayıcı unsuru konmamıştır. Hedeflendiği gibi doğrudan atak kaynağı olan ağda atakların kesildiği gözlemlenmiştir.

3- Bloklama sonrasında atak hedefi olan istemcinin haberleşmesinin devam ettiği gözlemlenmiştir.

4- YTA destekli sistemlerde, atak bloklanmadan önce atağın o anda gerçekleşip gerçekleşmediği kontrol edilip eğer atak mevcut ise atağın bloklanma sürecine geçildiği gözlemlenmiştir. Sistem şayet atak trafiği yoksa ya da atak kanyak IP adresi ağa dahil değilse hiçbir bloklanma sürecine geçmemiştir. Bu da hedeflerden biri olan atak kontrol işleminin yapıldığını göstermektedir.

5- YTA destekli sistemlerde atak hedeflendiği gibi kaynak ağda hatta kaynak anahtar cihazda bloklanmıştır.

6- Atak trafiği bittiğinde, YTA sistemi desteği ile oluşturulan sistemin atak bloklama işlemini otomatik kaldırdığı gözlemlenmiştir.

7- Kendi ağında olan bir atağın tespiti durumda sistem bunu belirlemiş ve atak kendi ağında olmasına rağmen atağı bloklanmıştır. Hatta atak trafiği durdurulduğunda, otomatik olarak algılayıp atak bloklama işlemini kaldırmıştır.

8- Hedeflerden biri, YTA desteği olmayan yerlerde de sistemin çalışmasıdır. Test ortamına YTA desteği olmayan bir ağda eklenmiştir. Sistemin bu tip ağlarda da kullanılabildiği de gözlemlenmiştir. Atak bilgileri yazılım arayüzü üzerinden kullanıcıya iletilmiştir.

9- Sistem sadece büyük hacimli DDoS ataklarında değil tekil ataklarda kullanılabilmesi hedeflenmiştir. Testlerde sistemin basit bir ping trafiğinde de çalıştığı görülmüştür.

10- USGM sayesinde atak kaynak IP adresinin hangi SBS alt yapısında bulunduğu bilgisinin merkezi bir sistemden temini sağlanmıştır. Bu da her kurumda tüm sistemin IP aralığı kaydının oluşturulması gereğini ortadan kaldırmıştır.

11- Yine USGM'nin sağladığı bilgilendirme servisi sayesinde, yetkisi olmayan bir IP adresi için bloklanma talebi yapılması durumunda, tespit edildiği ve talebin reddedildiği gözlemlenmiştir.

12- Sistemin gerekmesi durumda ulusal çapta bloklama taleplerini uygulayabilmesi hedeflenmiştir. Testler de USGM sunucusu tarafından yapılan ulusal bloklama talebi ve blok kaldırma talebinin sorunsuz bir şekilde uygulandığı görülmüştür.

BÖLÜM 6

SONUÇLAR VE TARTIŞMA

USBS ile ulusal çapta hem ISP'ler, hem de internet hizmeti alan kurumlar siber saldırıların engellenmesi sürecine doğrudan katılabilirler. Bu sistem ile atak kaynağı uyarılabilmekte, YTA desteği ile atak doğrulanabilmekte ve atak bloklanabilmektedir. Merkezi bir USGM sunucu yapısına da sahip olduğu için atak bilgisi kontrolü yapılabilmekte ve sahte bilgilendirme yapılması engellenebilmektedir.

Saldırı kaynağı IP adresi farklı bir ülkeye ait olabilir. Buna çözüm olarak, ilgili IP adresinin ülkeye giriş yaptığı yerel ISP firmalarında trafiğin bloklanması sağlanabilir. Ayrıca diğer ülkelerin kuracakları Ulusal Siber Güvenlik Merkezi (USGM) sunucuları arasında bilgi paylaşımı yapılarak, sistem global çalışabilecek hale getirilebilir. Bunun için de ilerleyen çalışmalarda USGM sunucuları arasında gerçekleşecek bu paylaşım süreci modellenebilir.

Bu sistem gerek tekil atakları, gerekse DDoS gibi organize atakları uyarı ve bloklama amaçlıdır. Bu tip ataklara çoğu zaman zararlı, yazılım bulaşmış bilgisayarlar sebep olmaktadır. Atağın hedef ağda bloklanması durumunda, atak sebebi olan kaynak bilgisayarın bulunduğu ağa bir uyarı yapılmamaktadır. Sistemin atak kaynağı kurumu, yaptığı atak ile ilgili uyarması sayesinde, zararlı yazılım bulaşmış istemciler tespit edilip temizlenebilir. Bu da bu tip saldırılara sebep olan cihaz sayısının azalmasına yardımcı olacaktır.

NAT arasındaki bir istemcinin saldırı yapması durumunda paylaşılan atak bilgisinde, kurumun ortak kullandığı gerçek IP adresi yer alacaktır. NAT arkasında ki bir cihazın tespiti için haberleşmede kullanılan kaynak ve hedef IP adresinin yanı sıra, port

numaraları da gerekmektedir. Bu sistem bilgilendirme sırasında port numarası bilgilerini de paylaşabilmektedir. Ancak sistem NAT tercüme tablolarına bakıp, atak kaynak bilgisayarın özel IP adresi detaylarını belirleyip ağ yöneticisine raporlamaz. Ağ yöneticisi, güvenlik duvarındaki NAT tercüme tablosuna bakıp, manuel olarak kaynak cihazı tespit edip, bloklanmasını sağlayabilir. Gelecekte NAT tercüme tabloları ile sistem arasında bütünleşme sağlayacak bir modülle, bu bilgi otomatik tespit edilebilir. IPv6'ya tamamı ile geçilmesi durumunda ise tüm istemciler gerçek IP adresi kullanacağı için böyle bir çözüme ihtiyaç kalmayacaktır.

Sistem, tüm kurulan kurumlarda atak bilgilendirme özelliğine sahiptir. Ancak atak onaylama ve bloklama özellikleri için ağ altyapısının, YTA destekli olması gerekmektedir. Çalışma dahilde ki üçüncü konuda, YTA destekli olmayan ağlara tek bir YTA destekli anahtar eklenerek bu özelliklerin de eklenebileceği gösterilmektedir. Buna alternatif olarak ağ cihazı üreticilerinin kendilerine özel geliştirdikleri (DNA Center vb.) YTA çözümlerin için REST API ile haberleşen otomatik bloklama modülleri de geliştirilebilir. Ayrıca BGP kullanan kurumlar için bu sisteme S/RTBH, D/RTBH veya Flowspec gibi teknikleri kullanan modüller eklenip, otomatik bloklama işleminin BGP protokolü tarafından yapılması sağlanabilir.

İlerleyen çalışmalarda, Ulusal Siber Güvenlik Merkezi (USGM) sunucusunda yapılan bilgilendirme sorguları kayıt altına alınabilir. Elde edilen bu veri ile ülke çapında yapılan atakların veri tabanı oluşturulabilir. Atak verisinin doğruluğu da sistem tarafından onaylanabildiği için elde edilen veri gerçekçi olacaktır. Sisteme eklenecek ek bir modül ile atak bittiği zaman da USGM sunucusuna bilgilendirme yapılabilir. Bu da ülke çapında ki aktif siber güvenlik ataklarının gözlemlenmesini ve atakların toplamda gerçekleşme sürelerinin tespitini sağlayacaktır. Yapılacak bir çalışma ile oluşturulacak yapay zeka kullanan bir modül ile ulusal atak veri tabanından çıkarımlar yaptırılıp, ulusal bir tehdit durumunda alınması gereken önlemler raporlanabilir. Bu rapor da otomatik olarak kurumların siber olaylara müdahale ekiplerine e-posta aracılığı ile ulaştırılabilir. Ayrıca bu çalışma ile oluşturulmuş ulusal bloklama özelliği de kullanılarak, atağın etkileri ülke çapında azaltılabilir.

IP sahtekarlığı (IP Spoofing) atağı olması durumunda, uyarılan atak kaynağı sistem, bu atağın kendisinden kaynaklanmadığını raporlayacaktır. Yapılacak bir çalışma ile ulusal çaptaki bu tip ataklar da tespit edebilecektir.

Son olarak bu sistemin kullanılması durumunda etikliği tartışılacak bazı durumlar ortaya çıkabilir. Örneğin bir kurum, kendisine ait bir IP adresine atak olmamasına rağmen bloklama talebi yapabilir. Bu durumda bloklayacak taraf, atak kontrolü yapıp bloklamayı reddedebilir ya da talebi uygulayabilir. Etik olarak hangisinin yapılacağı kararı kurumun ağ yöneticisine kalmaktadır. Diğer bir konuda bir atak olamaması rağmen ulusal siber güvenlik merkezinin yasaklama amaçlı bloklama yapabilmesi ihtimalidir.

KAYNAKLAR

Abbas, S. G., Zahid, S., Hussain, F., Shah, G. A., & Husnain, M. (2020). A Threat Modelling Approach to Analyze and Mitigate Botnet Attacks in Smart Home Use Case. *Proceedings - 2020 IEEE 14th International Conference on Big Data Science and Engineering, BigDataSE 2020*, 122–129. https://doi.org/10.1109/BigDataSE50710.2020.00024

Abou El Houda, Z., Khoukhi, L., & Senhaji Hafid, A. (2020). Bringing Intelligence to Software Defined Networks: Mitigating DDoS Attacks. *IEEE Transactions on Network and Service Management*, *17*(4), 2523–2535. https://doi.org/10.1109/TNSM.2020.3014870

Agrawal, N., & Tapaswi, S. (2018). Low rate cloud DDoS attack defense method based on power spectral density analysis. *Information Processing Letters*, *138*, 44–50. https://doi.org/10.1016/j.ipl.2018.06.001

Akin, G., Bük, O., & Uçar, E. (2020). An inter-domain attack mitigating solution. *Turkish Journal of Electrical Engineering and Computer Sciences*, *28*(2), 757–772. https://doi.org/10.3906/elk-1904-179

Arbor Networks. (2012). *Layered Intelligent DDoS Mitigation Systems*.

Ashi, Z., Aburashed, L., Al-Fawa'reh, M., & Qasaimeh, M. (2020). Fast and Reliable DDoS Detection using Dimensionality Reduction and Machine Learning. *15th International Conference for Internet Technology and Secured Transactions, ICITST 2020*. https://doi.org/10.23919/ICITST51030.2020.9351347

Azodolmolky, S. (2013). Software Defined Networking w OpenFlow. In *Packt Publishing Ltd.*

Bhushan, K., & Gupta, B. B. (2018). Hypothesis Test for Low-rate DDoS Attack Detection in Cloud Computing Environment. *Procedia Computer Science*, *132*,

947–955. https://doi.org/10.1016/j.procs.2018.05.110

Carvalho, L. F., Abrão, T., Mendes, L. de S., & Proença, M. L. (2018). An ecosystem for anomaly detection and mitigation in software-defined networking. *Expert Systems with Applications*, *104*, 121–133. https://doi.org/10.1016/j.eswa.2018.03.027

Casado, M., Koponen, T., Moon, D., & Shenker, S. (2008). Rethinking packet forwarding hardware. *Proc. Seventh ACM SIGCOMM HotNets Workshop*, 1–6. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.143.2052&rep=rep1&type=pdf

Case, J., Rose, M., McCloghrie, K., Waldbusser, S., & Presuhn, R. (2002). *RFC 3416 Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)*.

Chaudhary, V., & Harshan, J. (2020). Fast-forward relaying scheme to mitigate jamming attacks by full-duplex radios. *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, *2020-Augus*, 0–6. https://doi.org/10.1109/PIMRC48278.2020.9217180

Chen, C. Y., Chen, L. A., Cai, Y. Z., & Tsai, M. H. (2020). RNN-based DDoS Detection in IoT Scenario. *International Computer Symposium, ICS 2020*, 448–453. https://doi.org/10.1109/ICS51289.2020.00094

Chen, Yini, Hou, J., Li, Q., & Long, H. (2020). DDoS Attack Detection Based on Random Forest. *IEEE International Conference on Progress in Informatics and Computing (PIC)*. https://doi.org/10.1109/PIC50277.2020.9350788

Chen, Yu, Hwang, K., & Ku, W. S. (2007). Collaborative detection of DDoS attacks over multiple network domains. *IEEE Transactions on Parallel and Distributed Systems*, *18*(12), 1649–1662. https://doi.org/10.1109/TPDS.2007.1111

Costa, W. L., Silveira, M. M., De Araujo, T., & Gomes, R. L. (2020). Improving DDoS Detection in IoT Networks through Analysis of Network Traffic Characteristics. *Proceedings - 2020 IEEE Latin-American Conference on Communications, LATINCOM 2020*. https://doi.org/10.1109/LATINCOM50620.2020.9282265

Cui, J., Zhang, J., He, J., Zhong, H., & Lu, Y. (2020). DDoS detection and defense

mechanism for SDN controllers with K-Means. *Proceedings - 2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing, UCC 2020*, 394–401. https://doi.org/10.1109/UCC48980.2020.00062

Cui, Y., Yan, L., Li, S., Xing, H., Pan, W., Zhu, J., & Zheng, X. (2016). SD-Anti-DDoS: Fast and efficient DDoS defense in software-defined networks. *Journal of Network and Computer Applications*, *68*, 65–79. https://doi.org/10.1016/j.jnca.2016.04.005

Ding, A. Y., Crowcroft, J., Tarkoma, S., & Flinck, H. (2014). Software defined networking for security enhancement in wireless mobile networks. *Computer Networks*, *66*, 94–101. https://doi.org/10.1016/j.comnet.2014.03.009

Doherty, J. (2016). *SDN and NFV Simplified: A Visual Guide to Understanding Software Defined Networks and Network Function Virtualization* (Vol. 91, p. 400). Pearson Education. https://extranet.cranfield.ac.uk/book/operating-systems-and-server-administration/virtualization/,DanaInfo=proquestcombo.safaribooksonline.com+9780134307398

Douligeris, C., & Mitrokotsa, A. (2004). DDoS attacks and defense mechanisms: Classification and state-of-the-art. *Computer Networks*, *44*(5), 643–666. https://doi.org/10.1016/j.comnet.2003.10.003

Erhan, D., & Anarim, E. (2020). Istatistiksel Yöntemler ile DDoS Saldiri Tespiti DDoS Detection Using Statistical Methods. *2020 28th Signal Processing and Communications Applications Conference, SIU 2020 - Proceedings*, *117*, 4–7. https://doi.org/10.1109/SIU49456.2020.9302487

Farhady, H., Lee, H., & Nakao, A. (2015). Software-Defined Networking: A Survey. *Computer Networks*, *81*, 79–95. https://doi.org/10.1016/j.comnet.2015.02.014

Faruq, M. O., & Sattar, A. H. M. S. (2020). An Approach to Mitigate Composition Attacks on Privacy in Non-coordinated Environments. *2020 2nd International Conference on Advanced Information and Communication Technology, ICAICT 2020*, *November*, 123–128. https://doi.org/10.1109/ICAICT51780.2020.9333538

Feamster, N., Rexford, J., & Zegura, E. (2013). The Road to SDN. *Queue*, *11*(12), 20–40. https://doi.org/10.1145/2559899.2560327

Firdaus, D., Munadi, R., & Purwanto, Y. (2020). DDoS Attack Detection in Software Defined Network using Ensemble K-means++ and Random Forest. *3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*. https://doi.org/10.1109/ISRITI51436.2020.9315521

Giotis, K., Argyropoulos, C., Androulidakis, G., Kalogeras, D., & Maglaris, V. (2014). Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments. *Computer Networks*, *62*, 122–136. https://doi.org/10.1016/j.bjp.2013.10.014

Gunawan, D. (2020). A Data Anonymization Method to Mitigate Identity Attack in Transactional Database Publishing. *2020 8th International Conference on Information and Communication Technology, ICoICT 2020*, 1–6. https://doi.org/10.1109/ICoICT49345.2020.9166262

Hakiri, A., Gokhale, A., Berthou, P., Schmidt, D. C., & Gayraud, T. (2014). Software-Defined Networking : Challenges and research opportunities for Future Internet. *Computer Networks*, *75*, 453–471. https://doi.org/10.1016/j.comnet.2014.10.015

Heart, F. E., Kahn, R. E., Ornstein, S. M., Crowther, W. R., & Walden, D. C. (1970). *The interface message processor for the ARPA computer network*. *1*, 551. https://doi.org/10.1145/1476936.1477021

Hu, F. (2014). Network Innovation through OpenFlow and SDN. In *CRC Press, Taylor & Francis Group, LLC*. https://doi.org/10.1201/b16521

IPerf. (2021). *iPerf - The ultimate speed test tool for TCP, UDP and SCTP*. https://iperf.fr/

Irenee, M., Hei, X., Wang, Y., Ji, W., & Jiang, X. (2020). Network Flow Analytics: Multi-Class Classification of DDoS Attacks Based on OKNN. *International Conference on Networking and Network Applications, NaNA 2020*, 271–276. https://doi.org/10.1109/NaNA51271.2020.00053

Jafarian, J. H., Al-Shaer, E., & Duan, Q. (2012). Openflow Random Host Mutation: Transparent Moving Target Defense Using Software Defined Networking. *1st Workshop on Hot Topics in Software Defined Networks (HotSDN 2012)*, 127–132. https://doi.org/10.1145/2342441.2342467

Jantila, S., & Chaipah, K. (2016). A Security Analysis of a Hybrid Mechanism to Defend DDoS Attacks in SDN. *Procedia Computer Science*, *86*(March), 437–440. https://doi.org/10.1016/j.procs.2016.05.072

Javed, M. A., Khan, M. Z., Zafar, U., Siddiqui, M. F., Badar, R., Lee, B. M., & Ahmad, F. (2020). ODPV: An Efficient Protocol to Mitigate Data Integrity Attacks in Intelligent Transport Systems. *IEEE Access*, *8*, 114733–114740. https://doi.org/10.1109/ACCESS.2020.3004444

Joldzic, O., Djuric, Z., & Vuletic, P. (2016). A transparent and scalable anomaly-based DoS detection method. *Computer Networks*, *104*, 27–42. https://doi.org/10.1016/j.comnet.2016.05.004

Koponen, T., Casado, M., Gude, N., Stribling, J., Poutievski, L., Zhu, M., Ramanathan, R., Iwata, Y., Inoue, H., Hama, T., Others, & Shenker, S. (2010). Onix A Distributed Control Platform for Large Procuction Networks. *USENIX Conference on Operating Systems Design and Implement*, *10*, 1–14. http://dl.acm.org/citation.cfm?id=1924943.1924968

Kreutz, D., Ramos, F. M. V., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, *103*(1), 14–76. https://doi.org/10.1109/JPROC.2014.2371999

Kumar, W., & McPherson, D. (2009). Remote Triggered Black Hole Filtering with Unicast Reverse Path Forwarding (uRPF) Remote - RFC5635. *IETF Trust*.

Kurt, B., Yıldız, Ç., Ceritli, T. Y., Sankur, B., & Cemgil, A. T. (2018). A Bayesian change point model for detecting SIP-based DDoS attacks. *Digital Signal Processing: A Review Journal*, *77*, 48–62. https://doi.org/10.1016/j.dsp.2017.10.009

Lai, Y.-K., Lee, C.-C., Huang, B.-H., Wellem, T., Wang, N.-C., Chou, T.-Y., & Nugroho, H. T. (2014). Real-time detection of changes in network with OpenFlow based on NetFPGA implementation. *Microprocessors and Microsystems*, *38*(5), 431–442. https://doi.org/10.1016/j.micpro.2014.04.005

Lai, Y. K., Huang, P. Y., Lee, H. P., Tsai, C. L., Chang, C. S., Nguyen, M. H., Lin, Y. J., Liu, T. L., & Chen, J. H. (2020). Real-Time DDoS Attack Detection using Sketch-

based Entropy Estimation on the NetFPGA SUME Platform. *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2020 - Proceedings*, 1566–1570.

Li, J., Liu, X., Jin, J., & Yu, S. (2021). Too Expensive to Attack: A Joint Defense Framework to Mitigate Distributed Attacks for the Internet of Things Grid. *CoRR*, *April*. http://arxiv.org/abs/2104.00236

Li, L., Sun, H., & Zhang, Z. (2011). The research and design of honeypot system applied in the LAN security. *ICSESS 2011 - Proceedings: 2011 IEEE 2nd International Conference on Software Engineering and Service Science*, 360–363. https://doi.org/10.1109/ICSESS.2011.5982237

Longo, F., Distefano, S., Bruneo, D., & Scarpa, M. (2015). Dependability modeling of Software Defined Networking. *Computer Networks*, *83*, 280–296. https://doi.org/10.1016/j.comnet.2015.03.018

Lucky, G., Jjunju, F., & Marshall, A. (2020). A Lightweight Decision-Tree Algorithm for detecting DDoS flooding attacks. *Proceedings - Companion of the 2020 IEEE 20th International Conference on Software Quality, Reliability, and Security, QRS-C 2020*, 382–389. https://doi.org/10.1109/QRS-C51114.2020.00072

Luong, T. K., Tran, T. D., & Le, G. T. (2020). DDoS attack detection and defense in SDN based on machine learning. *Proceedings - 2020 7th NAFOSTED Conference on Information and Computer Science, NICS 2020*, 31–35. https://doi.org/10.1109/NICS51282.2020.9335867

Mamolar, A. S., Pervez, Z., Alcaraz, C. J. M., & Khattak, A. M. (2018). Towards the transversal detection of DDoS network attacks in 5G multi-tenant overlay networks. *Computers and Security*, *79*, 132–147. https://doi.org/10.1016/j.cose.2018.07.017

Mantur, B., Desai, A., & Nagegowda, K. S. (2015). Centralized Control Signature-Based Firewall and Statistical-Based Network Intrusion Detection System (NIDS) in Software Defined Networks (SDN). *Emerging Research in Computing, Information, Communication and Applications*, 265–271. https://doi.org/10.1007/978-81-322-2550-8_48

Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J., & McPherson, D. (2009).

Dissemination of Flow Specification Rules - RFC 5575. *IETF*.

McCauley, J., Harchol, Y., Panda, A., Raghavan, B., & Shenker, S. (2019). Enabling a permanent revolution in internet architecture. *SIGCOMM 2019 - Proceedings of the 2019 Conference of the ACM Special Interest Group on Data Communication*, 1–14. https://doi.org/10.1145/3341302.3342075

Mckeown, N., Anderson, T., Peterson, L., Rexford, J., Shenker, S., & Louis, S. (2008a). OpenFlow : Enabling Innovation in Campus Networks. *Internetworking Research And Experience*, *38*(2), 69–74. https://doi.org/https://doi.org/10.1145/1355734.1355746 McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., … Turner, J. (2008). OpenFlow: Enabling Innovation in Campus Networks. ACM SIGCOMM Computer Communication Review, 38(2), 69. https://doi.org/10.1145/1355734.1355746

Mckeown, N., Anderson, T., Peterson, L., Rexford, J., Shenker, S., & Louis, S. (2008b). OpenFlow : Enabling Innovation in Campus Networks. *Internetworking Research And Experience*, *38*(2), 69–74. 10.1145/1355734.1355746

Miao, R., Yu, M., & Jain, N. (2014). NIMBUS: Cloud-Scale Attack Detection and Mitigation. *ACM SIGCOMM Computer Communication Review*, *44*, 121–122. https://doi.org/10.1145/2740070.2631446

Mininet. (2021). *Mininet: an instant virtual network on your computer*. http://mininet.org/

Mirkovic, J., & Reiher, P. (2004). A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM SIGCOMM Computer Communication Review*, *34*(2), 39. https://doi.org/10.1145/997150.997156

Monsanto, C., Reich, J., Foster, N., Rexford, J., & Walker, D. (2013). Composing Software-Defined Networks. *10th USENIX Symposium on Networked Systems Design and Implementation (NSDI '13)*.

Morreale, P. A., & Anderson, J. M. (2015). Software Defined Networking Design and deployment. In *CRC Press, Taylor & Francis Group, LLC*. http://vesta.informatik.rwth-aachen.de/opus/volltexte/2013/3789/pdf/dagrep_v002_i009_p095_s12363.pdf

Nadeau, T. D., & Gray, K. (2013). SDN: Software defined networks. In *O'REILLY*.

Naous, J., Erickson, D., Covington, A., & McKeown, N. (2008). Implementing and Deploying a NetFPGA OpenFlow Switch. *Symposium on Architectures for Networking and Communications Systems*, 1–9.

Nascimento, M. R., Rothenberg, C. E., Salvador, M. R., Magalhães, M. F., Salvador, M. R., & Magalhães, M. F. (2010). QuagFlow : Partnering Quagga with OpenFlow QuagFlow : Partnering Quagga with OpenFlow. *Innovation*.

Nugraha, B., & Murthy, R. N. (2020). Deep Learning-based Slow DDoS Attack Detection in SDN-based Networks. *2020 IEEE Conference on Network Function Virtualization and Software Defined Networks, NFV-SDN 2020 - Proceedings*, 51–56. https://doi.org/10.1109/NFV-SDN50289.2020.9289894

Nyasore, O. N., Zavarsky, P., Swar, B., Naiyeju, R., & Dabra, S. (2020). Deep Packet Inspection in Industrial Automation Control System to Mitigate Attacks Exploiting Modbus/TCP Vulnerabilities. *Proceedings - 2020 IEEE 6th Intl Conference on Big Data Security on Cloud, BigDataSecurity 2020, 2020 IEEE Intl Conference on High Performance and Smart Computing, HPSC 2020 and 2020 IEEE Intl Conference on Intelligent Data and Security, IDS 2020*, 241–245. https://doi.org/10.1109/BigDataSecurity-HPSC-IDS49724.2020.00051

Omar, H., & Khan, O. (2020). IRONHIDE: A secure multicore that efficiently mitigates microarchitecture state attacks for interactive applications. *Proceedings - 2020 IEEE International Symposium on High Performance Computer Architecture, HPCA 2020*, 111–122. https://doi.org/10.1109/HPCA47549.2020.00019

Open Networking Foundation. (2009). *OpenFlow Switch Specification 1.0.0 (Wire Protocol 0x01). 0*, 1–36. https://www.opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.0.0.pdf

Open Networking Foundation. (2011). OpenFlow Switch Specification 1.1.0 (Wire Protocol 0x02). *Current*.

OpenDaylight. (2021). *OpenDaylight*. https://www.opendaylight.org/

OpenvSwitch. (2021). *Open Virtual Switch (OVS)*. http://openvswitch.org/

Porras, P., Shin, S., Yegneswaran, V., Fong, M., Tyson, M., & Gu, G. (2012). A security enforcement kernel for OpenFlow networks. *Proceedings of the First Workshop on Hot Topics in Software Defined Networks - HotSDN '12*, 121. https://doi.org/10.1145/2342441.2342466

Postman. (2021). *Postman | The Collaboration Platform for API Development*. https://www.postman.com/

Priyadarshini, R., & Barik, R. K. (2019). A deep learning based intelligent framework to mitigate DDoS attack in fog environment. *Journal of King Saud University - Computer and Information Sciences*, *xxxx*. https://doi.org/10.1016/j.jksuci.2019.04.010

Protocol, V. W. (2009). OpenFlow Switch Specification. In *Wire* (Vol. 0, pp. 1–42).

Qi, H., & Li, K. (2016). *Software Defined Networking Applications in Distributed Datacenters*. Springer. https://doi.org/10.1007/978-3-319-33135-5

Reshikeshan, S. S. M., & Illindala, M. S. (2020). Systematically Encoded Polynomial Codes to Detect and Mitigate High-Status-Number Attacks in Inter-Substation GOOSE Communications. *2020 IEEE Industry Applications Society Annual Meeting, IAS 2020*. https://doi.org/10.1109/IAS44978.2020.9334776

Sahay, R., Blanc, G., Zhang, Z., & Debar, H. (2017). ArOMA: An SDN based autonomic DDoS mitigation framework. *Computers and Security*, *70*, 482–499. https://doi.org/10.1016/j.cose.2017.07.008

Sahoo, K. S., Puthal, D., Tiwary, M., Rodrigues, J. J. P. C., Sahoo, B., & Dash, R. (2018). An early detection of low rate DDoS attack to SDN based data center networks using information distance metrics. *Future Generation Computer Systems*, *89*, 685–697. https://doi.org/10.1016/j.future.2018.07.017

Sanjeetha, R., Ajay Shastry, K. N., Chetan, H. R., & Kanavalli, A. (2020). Mitigating HTTP GET FLOOD DDoS attack using an SDN controller. *Proceedings - 5th IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology, RTEICT 2020*, 6–10. https://doi.org/10.1109/RTEICT49044.2020.9315608

Saravan, R., Shanmuganathan, S., & Palanichamy, Y. (2016). Behavior-based detection of application layer distributed denial of service attacks during flash events. *Turkish Journal of Electrical Engineering & Computer Sciences*, *24*, 510–523. https://doi.org/10.3906/elk-1308-188

Sattar, D., & Matrawy, A. (2019). Towards Secure Slicing: Using Slice Isolation to Mitigate DDoS Attacks on 5G Core Network Slices. *2019 IEEE Conference on Communications and Network Security, CNS 2019*, 82–90. https://doi.org/10.1109/CNS.2019.8802852

Serrano Mamolar, A., Salvá-García, P., Chirivella-Perez, E., Pervez, Z., Alcaraz Calero, J. M., & Wang, Q. (2019). Autonomic protection of multi-tenant 5G mobile networks against UDP flooding DDoS attacks. *Journal of Network and Computer Applications*, *145*(August), 102416. https://doi.org/10.1016/j.jnca.2019.102416

Sezer, S., Scott-Hayward, S., Chouhan, P., Fraser, B., Lake, D., Finnegan, J., Viljoen, N., Miller, M., & Rao, N. (2013). Are we ready for SDN? Implementation challenges for software-defined networks. *IEEE Communications Magazine*, *51*(7), 36–43. https://doi.org/10.1109/MCOM.2013.6553676

Singh, K., Singh, P., & Kumar, K. (2017). Application layer HTTP-GET flood DDoS attacks: Research landscape and challenges. *Computers and Security*, *65*, 344–372. https://doi.org/10.1016/j.cose.2016.10.005

Skowyra, R., Bahargam, S., & Bestavros, A. (2013). Software-Defined IDS for securing embedded mobile devices. *2013 IEEE High Performance Extreme Computing Conference (HPEC)*, 1–7. https://doi.org/10.1109/HPEC.2013.6670325

Smiesko, J., & Uramova, J. (2020). One-parameter Methods for Recognizing DDoS Attacks. *ICETA 2020 - 18th IEEE International Conference on Emerging ELearning Technologies and Applications, Proceedings*, 628–633. https://doi.org/10.1109/ICETA51985.2020.9379155

Sotelo Monge, M. A., Herranz González, A., Lorenzo Fernández, B., Maestre Vidal, D., Rius García, G., & Maestre Vidal, J. (2019). Traffic-flow analysis for source-side DDoS recognition on 5G environments. *Journal of Network and Computer Applications*, *136*(February), 114–131. https://doi.org/10.1016/j.jnca.2019.02.030

Sudugala, A. U., Chanuka, W. H., Eshan, A. M. N., Bandara, U. C. S., & Abeywardena, K. Y. (2020). WANHEDA: A Machine Learning Based DDoS Detection System. *ICAC 2020 - 2nd International Conference on Advancements in Computing, Proceedings*, 380–385. https://doi.org/10.1109/ICAC51239.2020.9357130

Sumadi, F. D. S., & Aditya, C. S. K. (2020). Comparative Analysis of DDoS Detection Techniques Based on Machine Learning in OpenFlow Network. *3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, *18*(2), 221–226. https://doi.org/10.1109/ISRITI51436.2020.9315510

Thorpe, C., Tobin, J., & Murphy, L. (2020). An ISO/IEC 7816-4 Application Layer Approach to Mitigate Relay Attacks on near Field Communication. *IEEE Access*, *8*, 190108–190117. https://doi.org/10.1109/ACCESS.2020.3031979

Tiwari, V. (2013). SDN and OpenFlow for Beginners with hands on labs. In *M.M.D.D. Multimedia LLC*.

Toapanta, S. M., Escalante Quimis, O. A., Mafla Gallegos, L. E., & Maciel Arellano, M. R. (2020). Analysis for the evaluation and security management of a database in a public organization to mitigate cyber attacks. *IEEE Access*, *8*, 169367–169384. https://doi.org/10.1109/ACCESS.2020.3022746

Toapanta Toapanta, S. M., Mera Caicedo, H. A., Naranjo Sanchez, B. A., & Mafla Gallegos, L. E. (2020). Analysis of security mechanisms to mitigate hacker attacks to improve e-commerce management in Ecuador. *Proceedings - 3rd International Conference on Information and Computer Technologies, ICICT 2020*, 242–250. https://doi.org/10.1109/ICICT50521.2020.00044

Turk, D. (2004). Configuring BGP to Block Denial-of-Service Attacks Status - RFC 3882. *The Internet Society*, *2004*, 352. http://onlinelibrary.wiley.com/doi/10.1002/cbdv.200490137/abstract

VirtualBox. (2021). *Virtual Box*. https://www.virtualbox.org/

Vissicchio, S., Vanbever, L., & Bonaventure, O. (2014). Opportunities and research challenges of hybrid software defined networks. *ACM SIGCOMM Computer Communication Review*, *44*(2), 70–75. https://doi.org/10.1145/2602204.2602216

Wahida, F., Sarowar Sattar, A. H. M., & Hossain, M. M. (2020). Coordination Approach to Mitigate Composition Attack. *ICCIT 2020 - 23rd International Conference on Computer and Information Technology, Proceedings*, 19–21. https://doi.org/10.1109/ICCIT51783.2020.9392661

Wahida, F., Sattar, A. H. M. S., & Hossain, M. M. (2021). A New Coordination Approach to Mitigate Composition Attack. *2021 International Conference on Information and Communication Technology for Sustainable Development, ICICT4SD 2021 - Proceedings*, 274–279. https://doi.org/10.1109/ICICT4SD50815.2021.9396903

Wang, B., Zheng, Y., Lou, W., & Hou, Y. T. (2015). DDoS attack protection in the era of cloud computing and Software-Defined Networking. *Computer Networks*, *81*, 308–319. https://doi.org/10.1016/j.comnet.2015.02.026

Wang, K., Yuan, F., Hou, R., Ji, Z., & Meng, D. (2020). Capturing and Obscuring Ping-Pong Patterns to Mitigate Continuous Attacks. *Proceedings of the 2020 Design, Automation and Test in Europe Conference and Exhibition, DATE 2020*, 1408–1413. https://doi.org/10.23919/DATE48585.2020.9116509

Wang, L., Li, Q., Jiang, Y., Jia, X., & Wu, J. (2018). Woodpecker: Detecting and mitigating link-flooding attacks via SDN. *Computer Networks*, *147*, 1–13. https://doi.org/10.1016/j.comnet.2018.09.021

Wang, Z. (2019). An elastic and resiliency defense against DDoS attacks on the critical DNS authoritative infrastructure. *Journal of Computer and System Sciences*, *99*, 1–26. https://doi.org/10.1016/j.jcss.2017.05.012

World Economic Forum. (2021). The Global Risks Report 2021: 16th Edition. In *Weforum.Org*. http://www3.weforum.org/docs/WEF_The_Global_Risks_Report_2021.pdf

Xing, T., Huang, D., Xu, L., Chung, C.-J., & Khatkar, P. (2013). SnortFlow: A OpenFlow-Based Intrusion Prevention System in Cloud Environment. *2013 Second GENI Research and Educational Experiment Workshop*, 89–92. https://doi.org/10.1109/GREE.2013.25

Xming. (2021). *Xming*. http://www.straightrunning.com/XmingNotes/

Yang, L., Dantu, R., Anderson, T., & Gopal, R. (2004). Forwarding and Control Element Separation (ForCES) Framework" RFC 3746 (Internet Standart). *The Internet Society*. https://datatracker.ietf.org/doc/html/rfc3746

Yoon, C., Park, T., Lee, S., Kang, H., Shin, S., & Zhang, Z. (2015). Enabling security functions with SDN: A feasibility study. *Computer Networks*, *85*(2015), 19–35. https://doi.org/10.1016/j.comnet.2015.05.005

Yuan, B., Zou, D., Jin, H., Yu, S., & Yang, L. T. (2017). HostWatcher: Protecting hosts in cloud data centers through software-defined networking. *Future Generation Computer Systems*. https://doi.org/10.1016/j.future.2017.04.023

Yusop, Z. M., & Abawajy, J. H. (2014). Analysis of Insiders Attack Mitigation Strategies. *Procedia - Social and Behavioral Sciences*, *129*, 611–618. https://doi.org/10.1016/j.sbspro.2014.06.002

Zargar, S. T., Joshi, J., & Tipper, D. (2013). A survey of defense mechanisms against distributed denial of service (DDOS) flooding attacks. *IEEE Communications Surveys and Tutorials*, *15*(4), 2046–2069. https://doi.org/10.1109/SURV.2013.031413.00127

Zhang, L., Shou, G., Hu, Y., & Guo, Z. (2013). Deployment of Intrusion Prevention System based on Software Defined Networking. *International Conference on Communication Technology Proceedings, ICCT*, 26–31. https://doi.org/978-1-4799-0077-0/13

Zhou, L., Guo, H., & Deng, G. (2019). A fog computing based approach to DDoS mitigation in IIoT systems. *Computers and Security*, *85*, 51–62. https://doi.org/10.1016/j.cose.2019.04.017

Zhu, L., Tang, X., Shen, M., Du, X., & Guizani, M. (2018). Privacy-Preserving DDoS Attack Detection Using Cross-Domain Traffic in Software Defined Networks. *IEEE Journal on Selected Areas in Communications*, *36*(3), 1. https://doi.org/10.1109/JSAC.2018.2815442

Zinin, A. (2002). Cisco IP routing. In *Addison-Wesley* (Vol. 16, Issue 2).

ÖZGEÇMİŞ

Yirmi yılı aşkın bir süredir ağ teknolojileri ve güvenlik üzerine çalışan Gökhan Akın, kariyerine İTÜ Bilgi İşlem Daire Başkanlığı'nda Ağ uzmanı olarak başlamıştır. İlerleyen süreçte ağ grubu takım lideri olarak görev almıştır. Bir dönem Tübitak/Ulakbim altında ki Ulusal Akademik Ağ Siber Güvenlik Birimi'nin (ULAK/CSIRT) beş üyesinden biri olarak görev yapmıştır. Kariyeri boyunca birçok kurum, ISP ve üniversiteye danışman olarak hizmet vermiştir. Ayrıca raylı ulaşım ve karayolu sistemlerine ağ projeleri hazırlayan Gökhan Akın, bunların yanında ağ uzmanlığı konusunda en prestijli sertifikalardan biri sayılan CCIE sertifikasına sahiptir. Ağ yönetimi ve siber güvenlik üzerine onlarca konferansa konuşmacı olarak katılmasının yanında 2001 yılından beri Cisco Networking Academy'de eğitmen eğitmeni olarak ağ teknolojileri ve güvenlik üzerine dersler vermekte ve yeni eğitmenler yetiştirmektedir. Akademik kariyerine Trakya Üniversitesi'nde Bilgisayar Mühendisliği dalında doktora yaparak devam eden Gökhan Akın, halen özel bir üniversitede öğretim görevlisi olarak eğitim vermektedir. Kendisi şu an Ağ Yöneticileri Derneği'nin kurucu başkanlığını yapmaktadır.

TEZ ÖĞRENCİSİNE AİT TEZ İLE İLGİLİ BİLİMSEL FAALİYETLER

Akın G, Buk O, Uçar E, An Inter-Domain Attack Mitigating Solution, Turk J Elec Eng & Comp Sci, (2020),

Akın G, Karaarslan E, Buk O, Uçar E, SDN Architecture Fundamentals and DOS Prevention Basics: A Case Study with Openflow, International Scientific Conference (UNITECH), (2015), Gabrovo