Uludağ Üniversitesi Mühendislik Fakültesi Dergisi, Cilt 26, Sayı 2, 2021 ARAŞTIRMA

DOI: 10.17482/uumfd.769939

YAZILIM TANIMLI AĞLARDA BAĞLANTI KATMANI KEŞİF PROTOKOLÜNÜN İSTİSMARINA DAYALI TOPOLOJİ ZEHİRLEME SALDIRILARININ İNCELENMESİ

Mevlüt Serkan TOK [*]
Mehmet DEMİRCİ [**]

Alınma: 15.05.2020; düzeltme: 30.03.2021; kabul: 24.05.2021

Öz: Yazılım tanımlı ağlar, geleneksel ağ mimarilerindeki ağ cihazları üzerinde bulunan veri düzlemi ve kontrol düzleminin ayrıldığı ve ağın tamamına hâkim merkezi kontrolcüler tarafından paket iletim kararlarının alındığı bir ağ teknolojisidir. Kontrolcünün ağın bütün işleyişini yönetme zorunluluğu kontrol kolaylığı sağladığı gibi güvenlik açıklıkları da yaratabilmektedir. Bu çalışmada, ele geçirilen uç cihazlar üzerinden kontrolcünün yazılım tanımlı ağ topolojisini keşfetmede kullandığı OpenFlow Keşif Protokolünün (OFDP – OpenFlow Discovery Protocol) temelini oluşturan bağlantı katmanı keşif protokolünün (LLDP- Link Layer Discovery Protocol) kötüye kullanımı vasıtasıyla LLDP Enjeksiyonu ve LLDP yeniden gönderim saldırıları gerçekleştirilmiş, saldırı sonrasında kontrolcü üzerinde tutulan topoloji ve ağdaki uç cihazların bağlantı durumları tespit edilmiştir. Yapılan saldırı testleri sonucunda Floodlight kontrolcünün LLDP istismarını temel alan saldırılara karşı savunmasız olduğu ve kontrolcü üzerinde sahte bağlantı kayıtları oluşturulmasının mümkün olduğu görülmüştür. Ayrıca, kontrolcünün saldırı neticesinde kayıtlanmış sahte linkleri dikkate alarak geçersiz rotalar hesapladığı, bu rotaları kullanması ön görülen uç cihazların erişiminin kesildiği ve ağın genel başarımında azalma olduğu gözlemlenmiştir.

Anahtar Kelimeler: Yazılım Tanımlı Ağ, Bağlantı Katmanı Keşif Protokolü, LLDP, OpenFlow, Floodlight, Siber Güvenlik.

An Investigation of Topology Poisoning Attacks in Software Defined Networks Through Exploiting Link Layer Discovery Protocol

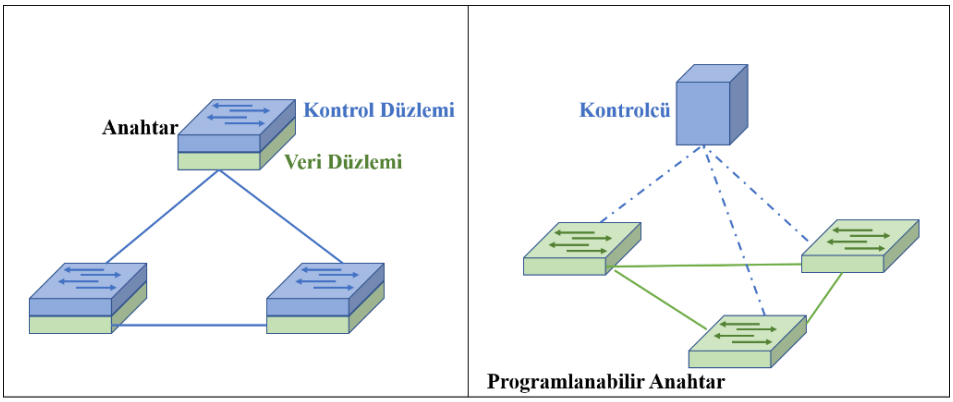
Abstract: In software defined networks (SDNs), data plane and control plane on the conventional network devices are separated and packet forwarding decisions are taken by the controller in charge of the entire network. The necessity of managing all the network operations not only facilitates control for the controller but also causes some security vulnerabilities and new attack surfaces in SDNs. In this study, Link Layer Discovery Protocol (LLDP), which forms the basis of OpenFlow Discovery Protocol (OFDP) for topology discovery in SDNs, is exploited to poison the topology by adding fabricated links to the controller. Fake LLDP Injection and LLDP replay techniques are used to create fake links on the controller and network access status of end devices are evaluated during attacks. Our findings demonstrate that the Floodlight controller is vulnerable to attacks based on LLDP exploitation and it is possible to create fake links on the controller. Due to the fake links, it is determined that invalid routes are created on controller and despite having other active links, end devices subject to invalid routes lost their access to network. Thus, attacks on the data link layer affect the performance of the network negatively.
Keywords: Software Defined Network, Link Layer Discovery Protocol, LLDP, OpenFlow, Floodlight, Cyber Security.

* Gazi Üniversitesi Fen Bilimleri Enstitüsü, 06560, Yenimahalle/ ANKARA
** Gazi Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü, 06570 Maltepe / Ankara
İletişim Yazarı: Mevlüt Serkan TOK (mevlutserkan.tok@gazi.edu.tr)

1. GİRİŞ

Dünyada internet kullanıcısı ve internete bağlı cihaz sayısı her geçen gün artmakta, artan internet kullanımı veri trafiğini artırmaktadır. 5G teknolojisinin hayata geçmesi bu veri trafiğini daha da arttıracağı, geleneksel ağ mimarisinin ve internet altyapısının bu artan trafiği yönetmede yetersiz kalacağı değerlendirilmektedir (Hu ve diğ., 2014; Hakiri ve diğ., 2014). Geleneksel ağ mimarisinin karmaşık bir yapıya sahip olduğu ve yönetimsel zorluklar barındırdığı uzun süredir bilinmektedir. Ağ anahtarlarının ve yönlendiricilerin yanı sıra güvenlik duvarları, yük dengeleyiciler, saldırı tespit sistemleri gibi çok sayıda farklı türdeki cihazların karşılıklı çalışabilirliğinin sağlanması ve orkestrasyonu geleneksel ağların yönetimindeki başlıca sorunlardır.

Son yıllarda yaygınlaşan yazılım tanımlı ağ teknolojisinde bilgi sistem ağlarının programlanabilirliğinin sağlanması amaçlanmıştır (Feamster ve diğ., 2014). Şekil 1'de gösterildiği üzere yazılım tanımlı ağ teknolojisinin temelinde geleneksel mimarideki ağ cihazları üzerinde bulunan veri düzlemi ve kontrol düzleminin birbirinden ayrılması ve paket iletim kararlarının ağın tamamını bilen merkezi kontrolcüler tarafından alınması yer alır. Yazılım tanımlı ağlarla birlikte sağlanan programlanabilirliğin ve merkezi kontrolün geleneksel ağ mimarilerindeki yönetimsel sorunların çözümünü kolaylaştıracağı öngörülmektedir (Demirci ve diğ., 2019). Yazılım tanımlı ağların başarımı birtakım temel şartların sağlanmasıyla mümkündür. Bunlardan en önemlisi kontrolcünün paket iletim kararlarını alabilmesi için kontrol ettiği ağın tamamını bilmesi, ağ trafiğinin en az bir defa kontrolcünün üzerinden geçirilerek trafiğin ne şekilde ve hangi istikamete iletilmesi gerektiğinin tespit edilmesidir (Pakzad ve diğ., 2016). Kontrolcünün topolojiyi bilme zorunluluğu ağdaki sistemlerin öğrenilmesinde ilave bir keşif mekanizmasına ihtiyaç duyulmasına sebep olmuştur. Dolayısıyla topoloji keşfi, yazılım tanımlı ağların işleyişinin sağlanması amacıyla çalıştırılan önemli bir servistir (Baidya ve Hewett, 2020). Paket anahtarlama ve yönlendirme gibi kritik işlemler topoloji keşfinin doğru çalışmasına dayanmaktadır.



Şekil 1:
Geleneksel ağ ve yazılım tanımlı ağ mimarisi

Merkezi bir kontrolcünün bulunmadığı geleneksel ağlarda ağda hizmet veren cihazların cinsleri, işletim sistemi versiyonları, IP adresleri, etki alanı adları, faal arayüzleri vb. bilgileri komşu cihazlarla paylaşmak maksadıyla kullanılan ilk keşif protokol örneklerinden biri CDP (Cisco Discovery Protocol) olup 1994 yılında tescillenmiş ve kullanılmaya başlanmıştır (Cisco Systems, 2006). Müteakip yıllarda Nortel gibi firmalar kendi keşif protokollerini üretmiş ve 2000 yılında yayınlanan RFC 2922 ile IEEE 802 teknolojisini esas alan Link Layer Discovery Protocol (LLDP) ile herhangi bir üreticiye ait protokole bağımlı kalınmadan açık kaynaklı olarak ağlarda keşif ve komşu tanıma işlemleri gerçekleştirilmeye başlanmıştır (Bierman ve

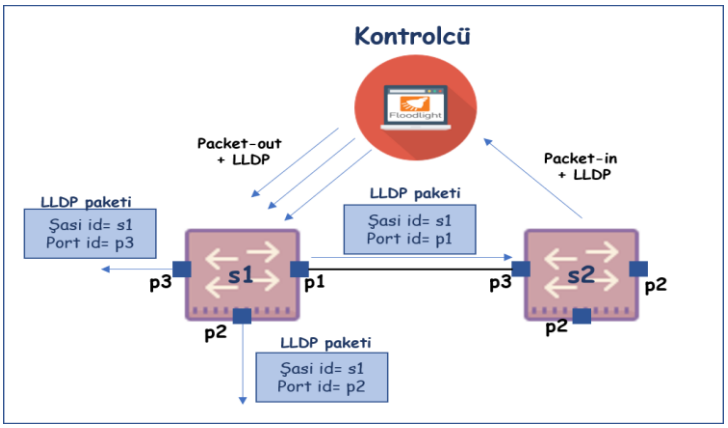
Jones, 2000). Tüm bu keşif protokollerinde ağ cihazları belirli mesaj formatları içerisine kendi özniteliklerine ait bilgileri eklemekte ve komşu cihazlarla paylaşarak ağ cihazlarının birbirlerinden haberdar olmasını sağlamaktadır.

Yazılım tanımlı ağlarda henüz standart bir topoloji keşif protokolü bulunmayıp fiilen standart olarak kabul gören OFDP (OpenFlow Discovery Protocol) vasıtasıyla topoloji keşfi gerçekleştirilmektedir. OFDP geleneksel ağlarda da kullanılan LLDP çerçevelerini sarmallayan ve bu çerçevelerin ağ anahtarları ile kontrolcü arasında transferini sağlayan bir protokoldür (Alharbi ve diğ., 2015). LLDP IEEE 802.1ab standardı ile tanımlanan, veri bağlantı katmanında çalışan, geleneksel ağlarda ağ keşfi amacıyla uzun süredir kullanılan ve yapısında kimlik doğrulama ve bütünlük kontrol mekanizması bulunmayan bir protokoldür (Congdon, 2002). Örnek bir OFDP çerçeve yapısı Şekil 2'de gösterilmiştir. Şase numarası, port numarası, yaşam süresi, seçenek bayrakları ve LLDP veri birim sonu başlıkları LLDP çerçeve formatını oluşturan başlıklardır.

Başlama Eki	Hedef MAC	Kaynak MAC	Ether-tipi 0x88CC	Şase No	Port No	Yaşam Süresi	Seçenek Bayrakları	LLDP Veri Birim Sonu	Çerçeve Kontrol Dizimi

Şekil 2:
OFDP çerçeve başlıkları (Ochoa-Aday ve diğ., 2015)

Yazılım tanımlı ağlarda bağlantı keşifleri kontrolcü tarafından başlatılmakta ve kontrolcü her bir ağ anahtarı için bağlantı keşfini aynı esaslarla yapmaktadır. Şekil 3'te gösterilen S1 anahtarı için bağlantı keşfi yapılacaksa, kontrolcü S1 anahtarının her bir portu için ayrı ayrı LLDP çerçeveleri hazırlar. Şasi kimliği olarak S1 anahtarının kimliği yazılırken port kimliği olarak her bir portun numarası yazılır. Kontrolcü ürettiği bu LLDP çerçevelerini OpenFlow packet-out mesajlarının içine koyarak S1 anahtarına gönderir. S1 aldığı bu mesajları portlarından iletir. Örneğin P1 portundan çıkan mesaj, S2 anahtarının P3 portundan alınır. S2 anahtarı S1'den gelen LLDP çerçevesine istinaden bir packet-in mesajı hazırlar, kendi şasi kimliğini ve S1 anahtarından gelen mesajı hangi portundan aldığını (bu durumda P3) ürettiği LLDP çerçevesine ekler, OFDP mesajının içerisine koyar ve kontrolcüye gönderir. Kontrolcü kendisine gelen bu mesajı açtığında S1 anahtarının P1 portu ile S2 anahtarının P3 portu arasında bir bağlantı olduğunu tespit eder. Bu işlem, kontrolcü tarafından yönetilen tüm ağ anahtarları için belirli zaman aralıklarıyla tekrarlanır.



Şekil 3:
Yazılım tanımlı ağlarda topoloji keşfi

Yazılım tanımlı ağların özellikle veri merkezleri ve 5G alanındaki gelişmeler ile önem kazanması bu ağ teknolojisinde bulunan güvenlik açıklıkları üzerine yapılan çalışmaların

sayısını da arttırmıştır. Ağ operasyonlarını yürütmek üzere geleneksel ağlarda kullanılan birçok protokolden faydalanan yazılım tanımlı ağlar tıpkı geleneksel ağlarda olduğu gibi bu protokollerden kaynaklanan açıklıkları da bünyesinde barındırmaktadır (Combe ve diğ., 2018). Yazılım tanımlı ağlar bu açıklıkların bir kısmından geleneksel ağlarla aynı seviyede etkilenirken bir kısım açıklıklar geleneksel ağlara kıyasla yazılım tanımlı ağları daha fazla etkilemektedir (Tok ve Demirci, 2021).

Geleneksel ağlarda, ağa bir şekilde erişim sağlayan saldırganların yakaladıkları CDP ve LLDP mesajlarını analiz ederek ağda hizmet veren ağ cihazlarının parmak izlerini tespit edebilmesi keşif protokolleri konusundaki en önemli siber risk olarak kabul edilmektedir (Vyncke ve Paggen, 2008). Bu riskin dışında söz konusu protokollerin istismarının geleneksel ağların performansına bir etkisi bulunmamaktadır. Yazılım tanımlı ağlardaysa LLDP protokolünün istismarı daha ciddi sonuçlara yol açabilmektedir. Bu ağlarda kontrolcünün topoloji keşif faaliyetini OFDP ile gerçekleştirilmesi nedeniyle ilave tedbir alınmadığı takdirde, ele geçirilen cihazlar üzerinden ağ anahtarlarına sahte LLDP çerçevesi gönderimi ile kontrolcü kandırılabilmekte ve kontrolcünün tuttuğu bağlantı tablosuna gerçekte var olmayan bağlantılar eklenebilmektedir (Smyth ve diğ., 2017). Kontrolcü paket iletim kararlarını veren birim olduğundan eklenen bu sahte bağlantılar kontrolcünün hatalı paket iletim kararları almasına, ağdaki cihazlarının bağlantılarının sekteye uğramasına ve hatta cihazların ağ erişiminin kesilmesine sebep olmaktadır. Dolayısıyla geleneksel ağlara kıyasla yazılım tanımlı ağların LLDP kaynaklı saldırılardan daha fazla etkilendiğini değerlendirmek mümkündür (Azzouni ve diğ., 2017).

Yazılım tanımlı ağların güvenliği üzerine odaklanan bu çalışmanın amacı;

(1) Floodlight kontrolcüsü tarafından kontrol edilen yazılım tanımlı bir ağda sahte LLDP mesaj enjeksiyonu ve LLDP mesaj yeniden gönderim saldırı senaryolarını uygulayarak yazılım tanımlı ağlarda LLDP saldırılarına yönelik kavram ispatı sağlamak,

(2) Floodlight kontrolcüsünün bu saldırılardan etkilenme düzeyini ve ağda bulunan sistemlerin saldırı sonrası ağ bağlantı durumlarını tespit etmek,

(3) LLDP istismarına dayalı topoloji zehirleme saldırılarının yazılım tanımlı ağlara yönelik hizmet dışı bırakma saldırılarında bir saldırı vektörü olarak kullanılabilirliğini test etmek,

(4) Topoloji yapısındaki farklılıkların bu saldırıların ağ üzerindeki etkilerini arttırıp arttırmadığını gözlemlemek,

(5) LLDP istismarına yönelik geçmişte yapılan çalışmaları inceleyerek koruyucu tedbir önerileri sunmaktır.

Çalışmanın ikinci bölümünde çalışmanın yöntemi açıklanmış, üçüncü bölümde elde edilen bulgular paylaşılmış, dördüncü bölümde araştırma alanıyla ilgili geçmişte yapılan çalışmalar sunulmuş, beşinci bölümde elde edilen bulgular tartışılmış ve altıncı bölümde sonuçlar açıklanarak gelecek çalışmalara ışık tutacak önerilerde bulunulmuştur.

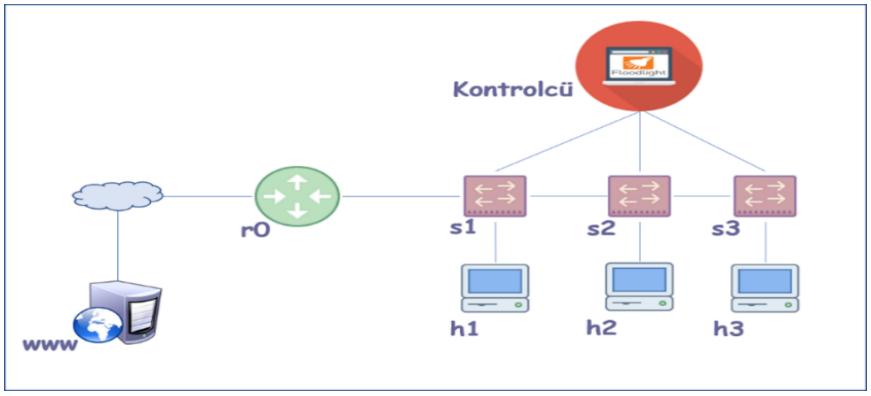
2. YÖNTEM

Çalışmada, seçilen saldırı tekniklerinin Windows 10 işletim sistemli, 16 GB RAM ve i5-5200 2.20 Ghz Intel işlemcili bir bilgisayarda kurulu sanallaştırma platformu üzerinde Mininet (“Mininet”, 2018) vasıtasıyla oluşturulmuş bir yazılım tanımlı yerel alan ağı topolojisinde uygulanarak sonuçlarının gözlemlenmesi amaçlanmıştır. İki aşamalı icra edilen saldırı testlerinin ilk aşamasında basit bir topolojide saldırıların gerçekleştirilerek kavram ispati sağlanması amaçlanmıştır. Bu kapsamda ilk aşama testler için oluşturulan topoloji Şekil 4’te gösterilmiştir. Kontrolcüye ve birbirlerine bağlı üç ağ anahtarı ve her ağ anahtarına bağlı birer tane uç cihaz bulunmaktadır. Yerel ağın internete erişiminde ağ adres çevrim (Network Addres Translation – NAT) protokolünden faydalanılmış ve internete R0 aracılığıyla erişilmiştir.

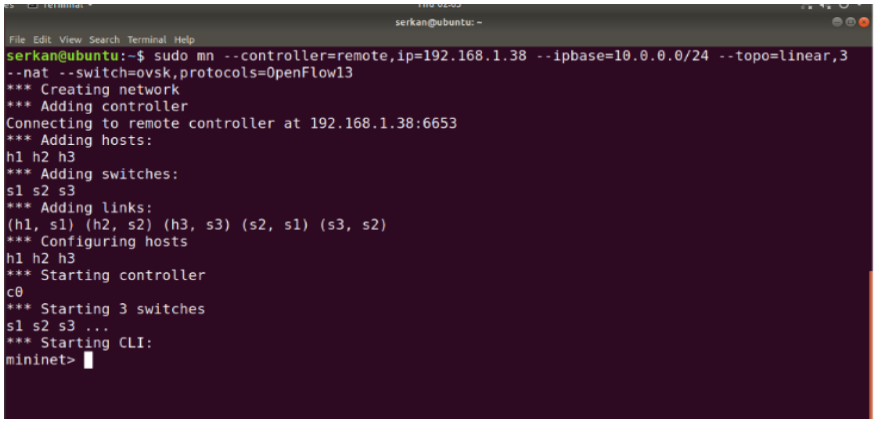
Farklı topolojilerin LLDP saldırılarından nasıl etkilendiğini tespit etmek amacıyla Mininet üzerinde kurulan beş katlı, basit ağaç topolojili, 31 anahtar ve 32 uç cihazdan oluşan test ağı ve test senaryosu 2.3.4. bölümde sunulmuştur.

2.1. Mininet Ağının Oluşturulması

Ubuntu 18.04.1 sürümlü sanal makine üzerine Mininet yazılım tanımlı ağ emülatörü yüklenmiş, kurulacak ağın uzak kontrolcü üzerinden kontrol edilmesi amacıyla "--controller=remote" parametresi kullanılmış, Mininet topolojisinin dışarıda bulunan bir web sunucusuna bağlanabilmesi için "--nat" parametresi tanımlanmış, "--ipbase" parametresiyle /24 prefixli bir ağ bloğunun sanal ağa tahsisi sağlanmıştır. Oluşturulan sanal test ağının bileşenleri ve sanal ağın kontrolünün gerçekleştirildiği Mininet komut satırı ara yüzü Şekil 5'te gösterilmiştir.



Şekil 4:
Saldırıların gerçekleştirildiği topoloji



Şekil 5:
Mininet üzerinde yazılım tanımlı ağ emülasyonu ve Mininet komut satırı arayüzü

2.2. Floodlight Kontrolcü

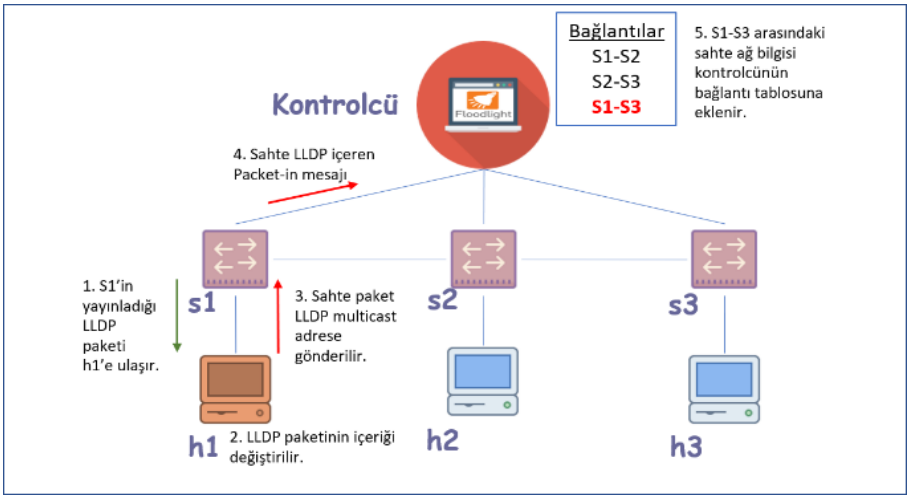
Floodlight kontrolcü web arayüzüne sahip, Java temelli bir kontrolcüdür. Yazılım tanımlı ağ yönetiminde en popüler kontrolcülerden biridir, açık kaynak kodludur, basit API'lere sahiptir ve eğitim amaçlı kullanımı yaygındır (Göransson ve diğ., 2017). Bu sebeple çalışmada Floodlight kontrolcü kullanımına karar verilmiş ve Ubuntu 14.04 işletim sistemi dağıtımlı sanal makine üzerine kurulan Floodlight v1.2 kontrolcü vasıtasıyla deney ağının kontrolü sağlanmıştır.

2.3. Bağlantı Katmanı Keşif Protokolünün Sömürülmesi

Topoloji zehirleme saldırısındaki amaç, kontrolcünün yöneteceği ağ ile ilgili yanlış bağlantı bilgilerine sahip olmasının sağlanmasıdır. Kontrolcülerin, yönettikleri ağda üretilen LLDP mesajlarının bütünlüklerini kontrol edememesi sahte LLDP mesajları vasıtasıyla sahte link bilgilerinin kontrolcüye kaydedilmesini mümkün kılmaktadır (Wang ve diğ., 2016). LLDP'ye dayalı topoloji zehirleme saldırılarında iki ayrı yöntem kullanılır. Birinci yöntemde sahte bir LLDP mesajı oluşturulup ele geçirilmiş bilgisayar üzerinden gönderilirken ikinci yöntemde gerçek LLDP mesajlarının değiş-tokuş edilerek farklı noktalardan yeniden gönderimi söz konusudur (Kaur ve diğ., 2017). Çalışma kapsamında LLDP saldırıları Scapy kütüphanesi ile sahte paket oluşturularak gönderimini sağlayan Python betikleri vasıtasıyla gerçekleştirilmiştir.

2.3.1. Sahte LLDP Mesaj Enjeksiyonuyla Sahte Link Oluşturma

Sahte LLDP mesaj enjeksiyonuyla sahte link üretme saldırısının gerçekleştirilebilmesi için saldırganın ağ yapısı hakkında bilgi sahibi olması ve ağdaki anahtarların şasi id numalarını bilmesi gerekmektedir. Bu saldırı Şekil 6'da gösterildiği üzere toplam beş safhada gerçekleştirilmekte olup her bir safhada yapılan işlemler aşağıda sunulmuştur.



Şekil 6:
Sahte LLDP mesaj enjeksiyonu ile sahte bağlantı ekleme senaryosu

(1) Saldırıyı gerçekleştirmek üzere bir uç sistem ele geçirilmiş, anahtardan gelen gerçek LLDP paketi Wireshark vb. bir trafik yakalama uygulaması ile kaydedilir.

(2) Scapy (python tabanlı paket manipülasyon kütüphanesi) yardımıyla LLDP mesajının içeriği değiştirilerek sahte bir LLDP mesajı hazırlanır. Değişiklik kapsamında LLDP mesaj başlıklarından şasi id başlığına S3 anahtarının id numarası, port id başlığına S3 anahtarının H3'e bağlı olan portunun numarası (1 nu.lı port) yazılır. Mesajın kaynak MAC adresine S3 anahtarının MAC adresi yazılır. Bu sayede LLDP mesajının S3 anahtarının 1 numaralı portundan gönderilmiş gibi düzenlenmiş olur.

(3) Scapy yardımıyla oluşturulan sahte LLDP mesajının hedef MAC adresi bölümüne LLDP çoklu yayın (multicast) MAC adresi (01:80:C2:00:00:0E) yazılarak S1 anahtarına gönderilir. Bu çoklu yayın MAC adresi üzerinde LLDP çalıştıran tüm ağ cihazları tarafından dinlenen bir adrestir.

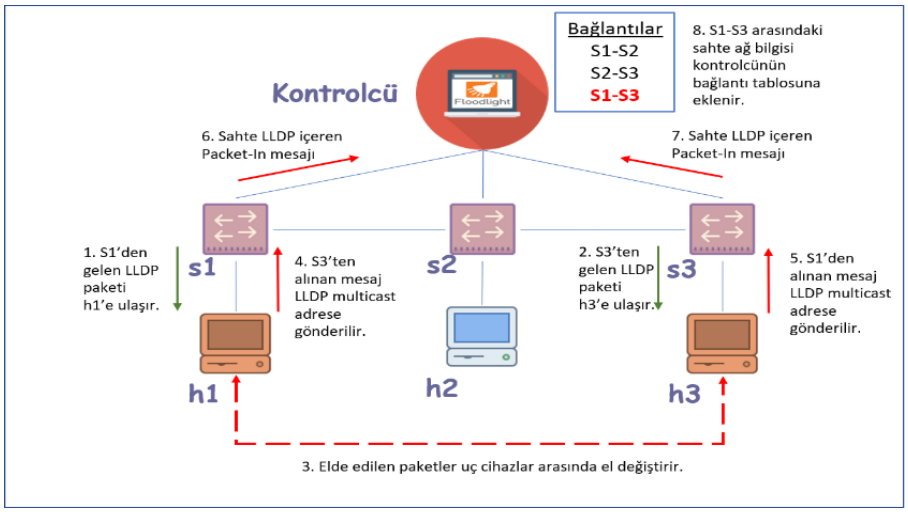
(4) S1 anahtarı kendisine gelen LLDP mesajının S3 anahtarının 1 numaralı portundan geldiğini kontrolcüye bildirmek üzere LLDP mesajını OpenFlow Keşif Protokolü packet-in mesajının içerisine koyar ve kontrolcüye gönderir.

(5) Kontrolcü kendisine ulaşan packet-in mesajını işleyerek S1-S3 arasında tek yönlü (unidirectional) bir bağlantı bulunduğu bilgisini bağlantı durum tablosuna ekler.

2.3.2. Gerçek LLDP Mesajlarının Yeniden Gönderimiyle Sahte Link Kaydı Oluşturma

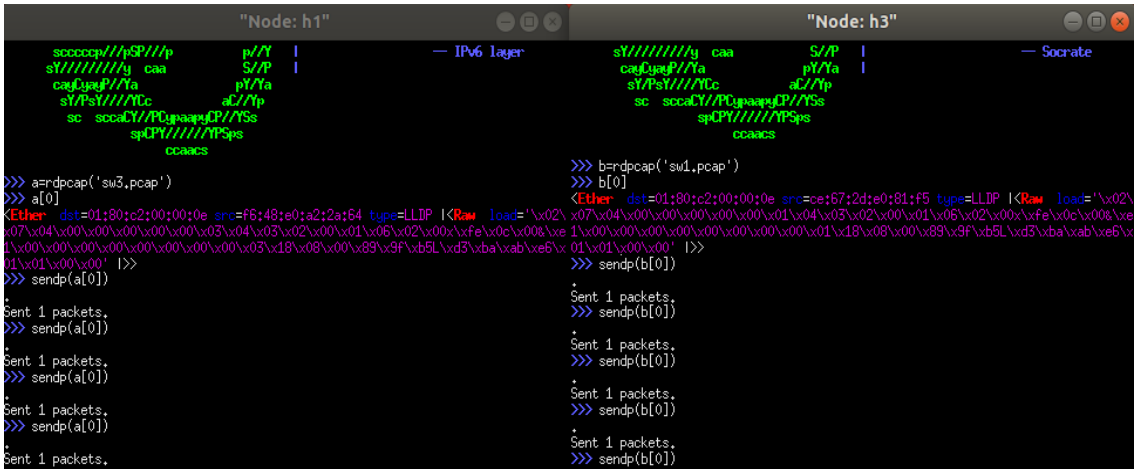
Gerçek LLDP mesajlarının yeniden gönderimi senaryosunda ağa bağlı iki cihazın ele geçirilerek birbirileri arasında iletişim kurması zorunluluğu bulunmaktadır. Bu saldırı yöntemi Şekil 7'de gösterildiği üzere toplam 8 safhada gerçekleştirilmekte olup her bir safhada gerçekleştirilen faaliyetler aşağıda sunulmuştur.

(1) H1 uç sistemi bağlı bulunduğu S1 anahtarından gelen gerçek LLDP mesajını bir ağ trafiği yakalama uygulamasıyla kaydeder.
(2) H3 uç sistemi bağlı bulunduğu S3 anahtarından gelen gerçek LLDP mesajını bir ağ trafiği yakalama uygulamasıyla kaydeder.
(3) H1 ve H3 kaydettikleri LLDP mesajlarını bant dışı bir bağlantı üzerinden (kablosuz vb.) birbiri ile değişir.
(4) H1, Scapy yardımıyla H3'ten aldığı LLDP mesajını S1 anahtarına gönderir.
(5) H3, Scapy yardımıyla H1'den aldığı LLDP mesajını S3 anahtarına gönderir.
(6) S1 kendisine gelen LLDP mesajı OpenFlow Keşif Protokolü packet-in mesajının içerisine koyar ve kontrolcüye gönderir.
(7) S3 kendisine gelen LLDP mesajı OpenFlow Keşif Protokolü packet-in mesajının içerisine koyar ve kontrolcüye gönderir.
(8) Kontrolcü S1 ve S3'ten gelen packet-in mesajlarını inceleyerek S1 ve S3 arasında iki yönlü (bidirectional) bir bağlantı olduğunu tespit ederek sahte link bilgisini bağlantı tablosuna ekler.



Şekil 7:
Gerçek LLDP paketlerinin anahtarlara yeniden gönderimi ile sahte bağlantı ekleme senaryosu

Şekil 8'de görüldüğü üzere LLDP mesajlarının yeniden gönderimi saldırısında Scapy ile karşılıklı değiş tokuş edilen LLDP mesajları direk gönderilir. Bu saldırı yöntemde sahte LLDP mesaj enjeksiyonu saldırısına kıyasla LLDP paketlerinin bütünlüğüne zarar verilmez.

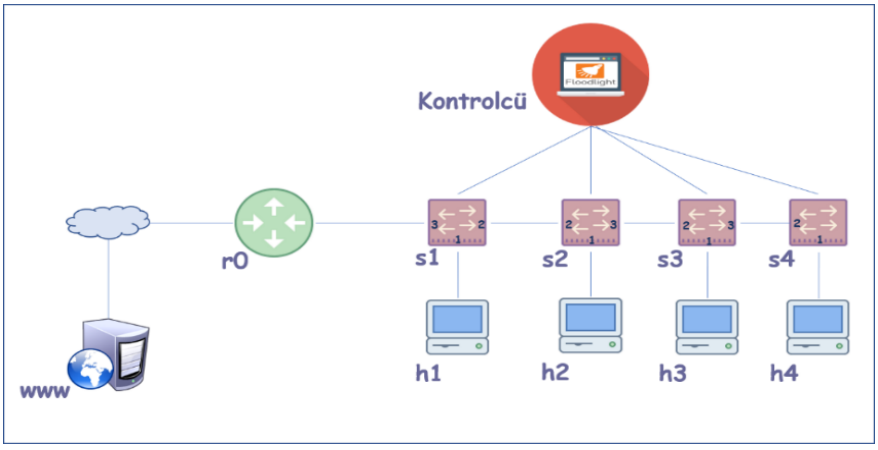


Şekil 8:

H1 ve H3 bilgisayarlarından LLDP mesajlarının Scapy uygulaması ile anahtarlara yeniden gönderimi

2.3.3. Saldırıların Kontrolcüde Tutulan Rotalar Üzerinde Etkisinin Tespiti

Kontrolcü üzerinde sahte link kayıtları oluşturmanın ağın başarımını ne kadar etkilediğini ve yönlendirme rotalarında değişiklik olup olmadığını tespit etmek amacıyla deney topolojisine bir ağ anahtarı ve uç cihaz daha eklenmiştir. H1 ve H3 üzerinden LLDP mesajı yeniden gönderme saldırısı yapılarak S1-S3 arasına sahte link eklenmiş, saldırıdan doğrudan etkilenmeyen H4'ün bağlı olduğu S4-1 portu ile ağ geçidinin bağlı olduğu S1-3 portu (ağ geçidine bağlı port) arasındaki rotanın değişip değişmediği ve H4'ün ağ erişim durumu kontrol edilmiştir. Test topolojisi ve port bağlantıları Şekil 9'da sunulmuştur.



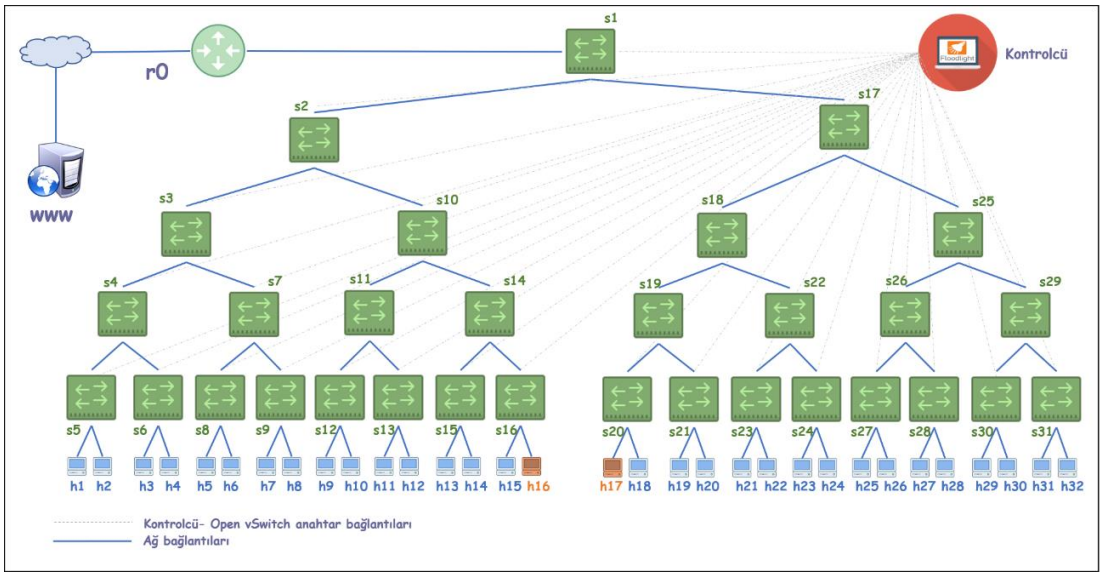
Şekil 9:

Rota tespit ve ağ erişim kontrol testinde kullanılan lineer topoloji

2.3.4. Farklı Yazılım Tanımlı Ağ Topolojilerinde LLDP Saldırılarının Etkileri

LLDP istismarına dayanan saldırıların daha karmaşık topolojilerde ne tür etkiler yaratacağının ve saldırı sonucu eklenecek bağlantıların ağa bağlı cihazların ağ erişimine bir etkisinin olup olmadığının tespit edilmesi amacıyla basit ağaç topolojisindeki ağda LLDP mesajı yeniden gönderim saldırısı tekrarlanmıştır. Şekil 10'daki topoloji üzerinde öncelikle

bütün sistemlerin birbiri ile haberleşebildiği kontrol edilmiştir. Sonrasında H16 ve H17 uç cihazlarının saldırganlarca ele geçirildiği varsayılmış ve bu cihazlara S16 ve S20'den gelen LLDP mesajları kaydedilerek cihazlar arasında kurulan ayrı bir bağlantıdan mesajlar değiş tokuş edilmiş ve LLDP mesajları cihazların bağlı bulundukları anahtarlara gönderilmiştir. Bu sayede S16-S20 arasında sahte bir çift yönlü bağlantı bilgisinin kontrolcü üzerinde oluşturulması sağlanmıştır. Saldırı sonrası değişen rotaları ve erişilemeyen bağlantıları belirlemek amacıyla Mininet komut satırı ara yüzünden pingall komutu ile tüm uç cihazlar arasındaki bağlantı durumu tespit edilmiştir.



Şekil 10:
Saldırı öncesi sistemlerin ağ bağlantı durumları

3. BULGULAR

Gerçekleştirilen saldırılarda kontrolcünün üzerinde tuttuğu bağlantı tablolarında sahte bağlantı kayıtları başarıyla oluşturulmuştur. Elde edilen bulgular aşağıda alt başlıklar halinde sunulmuştur.

3.1. Sahte LLDP Mesaj Enjeksiyonu Saldırısı

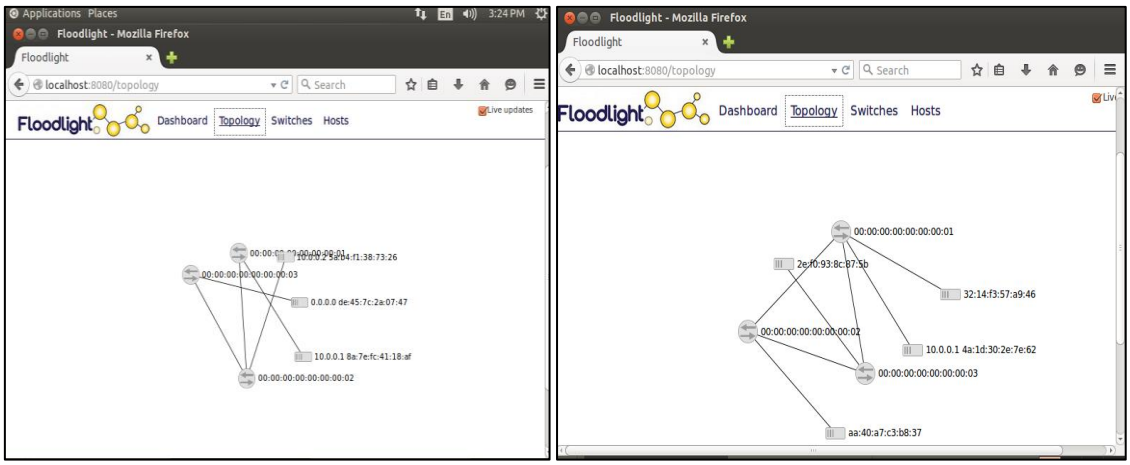
Üretilen sahte mesajın S1 anahtarına iletimi neticesinde kontrolcüde tutulan topoloji değişmiş ve S1-S3 arasında gerçekte var olmayan bir linkin kaydının kontrolcüye eklenmesi sağlanmıştır. Saldırı öncesi ve sonrası kontrolcü üzerinde tutulan topoloji görüntüsü Şekil 11'de gösterilmiştir.

Floodlight kontrolcüye REST API üzerinden yapılan bağlantı durum sorgusunda S1 ve S3 ağ anahtarlarının arasında tek yönlü bir linkin eklendiği tespit edilmiştir. Kontrolcüde tutulan topolojide gerçekleştirilen bu değişikliğin uç sistemlerin bağlantı durumunu nasıl etkilediği gözlemlenmiş, bunun sonucunda H1 ve H3 sistemlerinin ağ erişiminin kesildiği görülmüştür.

3.2. LLDP Mesajlarının Yeniden Gönderimi Saldırısı

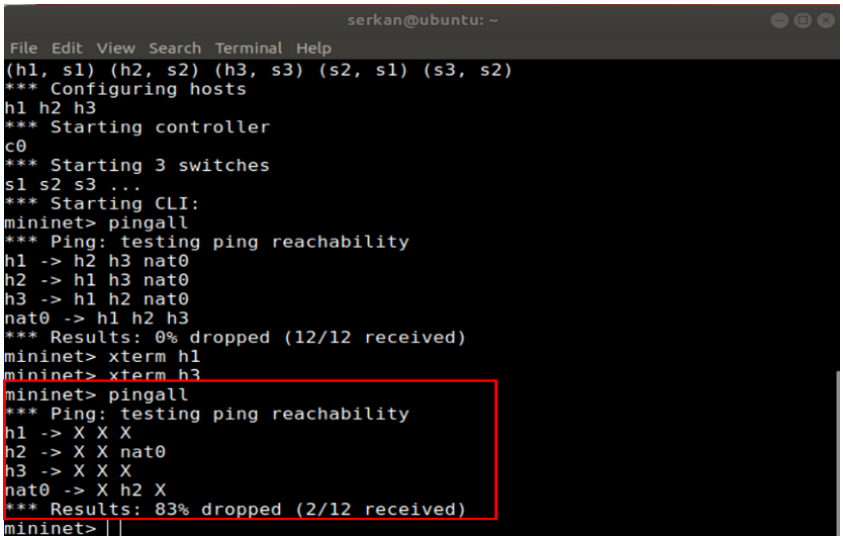
Saldırı sonucunda kontrolcü üzerinde tutulan topolojinin değiştiği tespit edilmiş ve REST API çağrısı ile yapılan sorgulamada kontrolcü bağlantı durum tablosuna S1-S3 anahtarları arasında gerçekte olmayan bir çift yönlü bağlantı eklendiği görülmüştür. Saldırı süresince sistemlerin ağ bağlantı durumları Mininet komut satırından pingall komutu ile kontrol edilmiş,

Şekil 12'de gösterildiği üzere H1 ve H3 sistemlerinin saldırı süresince ağ erişimlerinin kesildiği, sadece H2 ve NAT0 (r0) arasındaki bağlantının faal olduğu tespit edilmiştir.



a. *b.*

Şekil 11:
Kontrolcü üzerinde tutulan topoloji;
a. *Saldırı öncesi* ***b.*** *Saldırı sonrası*



Şekil 12:
Saldırı sonrası sistemlerin ağ bağlantı durumları

3.3. Saldırıların Kontrolcüde Tutulan Rotalar Üzerinde Etkisinin Tespiti

Saldırı öncesi ve saldırı sonrası H4'ün ağ geçidine ulaşmak için kullanacağı S4-P1 ve S1-P3 portları arasındaki iletişim için tutulan rota bilgisi REST API vasıtasıyla sorgulanmış, H4 cihazının uzak bir sistem ile veri iletim performansı saldırı öncesi ve saldırı sonrası icra edilen 60 sn. iperf testleri ile ölçümlenmiştir. Yapılan kontroller sonucunda, saldırı sonrasında söz konusu rotanın Floodlight kontrolcüsünün yaptığı en kısa yol hesaplaması neticesinde değiştiği, H4 sisteminin saldırı sonrasında fiilen bağlı olduğu S4:2 portundan S4:2>S3:3>S3:2>S2:3>S2:2>S1:2>S1:3 rotasını kullanarak ağa erişmesi mümkün olduğu halde

sahte link dikkate alınarak oluşturulmuş rota üzerinden ağa erişmeye çalıştığı ve ağa erişemediği görülmüştür. Elde edilen bulgular Tablo 1'de sunulmuştur.

Tablo 1. Saldırı sonucu değişen rota ve H4-uzak sunucu arası bağlantı testi sonuçları

Durum	S4-P1->S1-P3 iletişimi için kontrolcü tarafından hesaplanan rota	H4-uzak sunucu arası veri transfer miktarı	H4-uzak sunucu arası bant genişliği
Saldırı Öncesi	S4:1>S3:3>S3:2>S2:3>S2:2>S1:2>S1:3	448 Mb	62.6 Mbit/sn.
Saldırı Sonrası	S4:1>S4:2>S3:3>S3:1>S1:1>S1:3	0	0

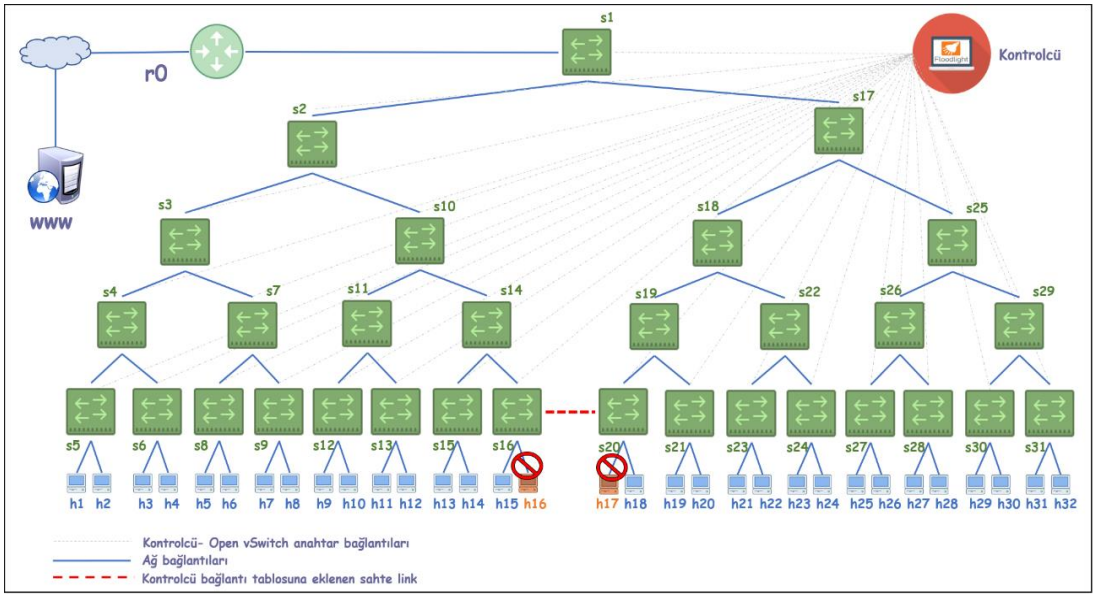
3.4. Çok Katmanlı Ağ Topolojisinde LLDP Saldırılarının Etkileri

H16 ve H17 üzerinden gerçekleştirilen saldırılar neticesinde kontrolcü üzerinde oluşturulan S16 ve S20 anahtarları arasındaki sahte iki yönlü bağlantının mantıksal gösterimi Şekil13'te sunulmuştur. Kontrolcü bu bilgiye dayanarak uç cihazlar arasındaki rotaları yeniden hesaplamıştır. Bu durum faal olan bağlantılar yerine sahte bağlantının da yer aldığı rotaların hesaplanmasına ve ağ topolojisinin bütünlüğünün bozulmasına neden olmuştur. Örneğin H15 cihazı H18 cihazına erişmek istediğinde kök düğümdeki S1 anahtarına gitmeden S16-S17 arasındaki yolu kullanmaya çalışmış, ancak gerçekte böyle bir bağlantı bulunmadığından H18'e erişim sağlayamamıştır. Saldırıları gerçekleştiren H16 ve H17 cihazları da saldırı sonucunda ağ erişimlerini kaybetmiştir. Bu kapsamda; S16-S17 sahte bağlantısının güzergâh hesaplamasına dahil edildiği tüm cihaz-cihaz rotaları aynı şekilde sekteye uğramış, topolojideki mevcut cihaz-cihaz bağlantılarının %21'inde erişim sağlanamamıştır. Saldırı sonucu yapılan pingall bağlantı testine göre ağda bulunan uç cihazlar arasında erişim durumunu gösteren bağlantı durum matrisi Şekil 14'te sunulmuştur. Topolojide ve ağ bağlantılarında oluşan kesintinin yanı sıra H16-H17 arasındaki sahte bağlantıyı kullanmaya çalışan sistemlerden gelen ağ paketleri H16 ve H17 tarafından Wireshark uygulaması yardımıyla yakalanmıştır.

5. İLGİLİ ÇALIŞMALAR

Alharbi ve diğ. (2015) yapmış oldukları çalışmada POX kontrolcüye LLDP'ye dayalı saldırılar sonucu eklenen tek yönlü bağlantıların kontrolcü tarafından yönlendirme ve rota tespitinde dikkate alınmadığı ancak çift yönlü bağlantıların dikkate alındığı görülmüştür. Rota hesaplamasına dahil edilen bu bağlantılar nedeniyle en kısa yol algoritmasının doğru çalışmadığı ve tüm uç cihazlarda çeşitli oranlarda bağlantı kaybı yaşandığı tespit edilmiştir. LLDP kaynaklı saldırıların önlenmesi amacıyla her bir LLDP paketine bir mesaj doğrulama kodu (MAC – message authentication code) eklenmesi yöntemine dayanan bir karşı tedbir geliştirmişlerdir. Eklenen mesaj doğrulama kodunun oluşturulmasında HMAC anahtarlanmış-özet tabanlı mesaj doğrulama kodu (Krawcyzk ve diğ., 1997) kullanılan yöntemle sahte LLDP paketlerinin bağlantı keşfine dahil edilmesi önlenmiştir. Yapılan testlerde kullanılan kriptografik yöntemin işlemci üzerinde %8 oranında ilave yük oluşturduğu gözlemlenmiştir.

Hong ve diğ. (2015) yazılım tanımlı ağ yönetiminde kullanılan Floodlight, OpenDaylight, Beacon ve POX kontrolcülerini test ederek tamamının LLDP kullanımı kaynaklı zafiyetleri barındırdığını tespit etmiş, TopoGuard adını verdikleri gerçek zamanlı ve otomatize edilmiş bir



Şekil 13:

Saldırı sonucu değişen ağ topolojisi

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17	H18	H19	H20	H21	H22	H23	H24	H25	H26	H27	H28	H29	H30	H31	H32	R0
H1		+	+	+	+	+	+	+	+	+	+	+	+	+	+	X	X	X	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
H2	+		+	+	+	+	+	+	+	+	+	+	+	+	+	X	X	X	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
H3	+	+		+	+	+	+	+	+	+	+	+	+	+	+	X	X	X	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
H4	+	+	+		+	+	+	+	+	+	+	+	+	+	+	X	X	X	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
H5	+	+	+	+		+	+	+	+	+	+	+	+	+	+	X	X	X	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
H6	+	+	+	+	+		+	+	+	+	+	+	+	+	+	X	X	X	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
H7	+	+	+	+	+	+		+	+	+	+	+	+	+	+	X	X	X	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
H8	+	+	+	+	+	+	+		+	+	+	+	+	+	+	X	X	X	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
H9	+	+	+	+	+	+	+	+		+	+	+	+	+	+	X	X	X	X	X	+	+	+	+	+	+	+	+	+	+	+	+	+
H10	+	+	+	+	+	+	+	+	+		+	+	+	+	+	X	X	X	X	X	+	+	+	+	+	+	+	+	+	+	+	+	+
H11	+	+	+	+	+	+	+	+	+	+		+	+	+	+	X	X	X	X	X	+	+	+	+	+	+	+	+	+	+	+	+	+
H12	+	+	+	+	+	+	+	+	+	+	+		+	+	+	X	X	X	X	X	+	+	+	+	+	+	+	+	+	+	+	+	+
H13	+	+	+	+	+	+	+	+	+	+	+	+		+	+	X	X	X	X	X	X	X	X	X	+	+	+	+	+	+	+	+	+
H14	+	+	+	+	+	+	+	+	+	+	+	+	+		+	X	X	X	X	X	X	X	X	X	+	+	+	+	+	+	+	+	+
H15	+	+	+	+	+	+	+	+	+	+	+	+	+	+		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	+
H16	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
H17	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
H18	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
H19	+	+	+	+	+	+	+	+	X	X	X	X	X	X	X	X	X	+		+	+	+	+	+	+	+	+	+	+	+	+	+	+
H20	+	+	+	+	+	+	+	+	X	X	X	X	X	X	X	X	X	+	+		+	+	+	+	+	+	+	+	+	+	+	+	+
H21	+	+	+	+	+	+	+	+	+	+	+	X	X	X	X	X	X	+	+	+		+	+	+	+	+	+	+	+	+	+	+	+
H22	+	+	+	+	+	+	+	+	+	+	+	X	X	X	X	X	X	+	+	+	+		+	+	+	+	+	+	+	+	+	+	+
H23	+	+	+	+	+	+	+	+	+	+	+	X	X	X	X	X	X	+	+	+	+	+		+	+	+	+	+	+	+	+	+	+
H24	+	+	+	+	+	+	+	+	+	+	+	X	X	X	X	X	X	+	+	+	+	+	+		+	+	+	+	+	+	+	+	+
H25	+	+	+	+	+	+	+	+	+	+	+	+	+	+	X	X	X	+	+	+	+	+	+	+		+	+	+	+	+	+	+	+
H26	+	+	+	+	+	+	+	+	+	+	+	+	+	+	X	X	X	+	+	+	+	+	+	+	+		+	+	+	+	+	+	+
H27	+	+	+	+	+	+	+	+	+	+	+	+	+	+	X	X	X	+	+	+	+	+	+	+	+	+		+	+	+	+	+	+
H28	+	+	+	+	+	+	+	+	+	+	+	+	+	+	X	X	X	+	+	+	+	+	+	+	+	+	+		+	+	+	+	+
H29	+	+	+	+	+	+	+	+	+	+	+	+	+	+	X	X	X	+	+	+	+	+	+	+	+	+	+	+		+	+	+	+
H30	+	+	+	+	+	+	+	+	+	+	+	+	+	+	X	X	X	+	+	+	+	+	+	+	+	+	+	+	+		+	+	+
H31	+	+	+	+	+	+	+	+	+	+	+	+	+	+	X	X	X	+	+	+	+	+	+	+	+	+	+	+	+	+		+	+
H32	+	+	+	+	+	+	+	+	+	+	+	+	+	+	X	X	X	+	+	+	+	+	+	+	+	+	+	+	+	+	+		+
R0	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	X	X	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	

Şekil 14:

Eklenen sahte bağlantının rota hesaplamasına dahil edilmesi sonucu mevcut cihaz-cihaz rotalarının erişim durumu matrisi.

araç geliştirerek Floodlight kontrolcüye uygulamıştır. TopoGuard üç temel bileşenden meydana gelmektedir. Bu bileşenlerin tamamı ağı yöneten kontrolcüde çalışan port yöneticisi, uç cihaz sorgulayıcı (prober) ve topoloji güncelleme doğrulayıcısıdır. Port kontrolcüsünün görevi her bir anahtarın portuna bağlı cihazların türünün kaydını (uç cihaz – ağ anahtarı) tutmaktadır. Bir uç cihazın bağlı olduğu porttan LLDP mesajı geldiği tespit edilirse alarm üretilerek saldırı tespiti

gerçekleştirilmektedir. Ağdaki bağlantıların keşfinde kontrolcü keşif işlemini başlatırken LLDP mesajının içerisine bir özet değeri (HMAC) koymaktadır. OFDP paketi içerisinde LLDP mesajı tekrar kontrolcüye döndüğünde kontrolcü önce özet değeri kontrol etmekte, ardından LLDP mesajının kimden geldiğini port kontrolcüsüyle iletişime geçerek tespit etmektedir. Eğer mesajı gönderen bir uç cihaz ise bunu bir saldırı olarak değerlendirmekte ve ağ bilgisini dikkate almamaktadır. Yapılan testlerde TopoGuard'ın etkili bir şekilde topoloji zehirleme saldırılarına karşı koruma sağlarken ağın normal işleyişi üzerinde göz ardı edilebilir etkileri olduğu, LLDP mesajlarına bir defaya mahsus HMAC eklenmesi işleminde LLDP mesaj üretim süresinin %80 artmasına rağmen dikkate değer bir performans kaybı yaşanmadığı gözlemlenmiştir.

Smyth ve diğ. (2017) yapmış oldukları çalışmada LLDP istismarına dayanarak yazılım tanımlı ağ üzerinde sahte ağlar oluşturulmasını tespit etmek üzere bağlantı gecikmelerini gözlemlemeyi önermişlerdir. Çalışmanın temelinde yeni öğrenilen ağların doğrudan işleme alınmayıp ağ üzerindeki gecikmelerin yeniden hesaplanarak elde edilen bulguların istatistiksel yöntemlerle analiz edilmesi ve anomalilerin tespit edilmesi konsepti bulunmaktadır. Bu konseptin en büyük faydası LLDP paketlerine ilave bir başlık veya veri eklemeden, ağ anahtarları üzerinde herhangi bir ilave uygulama kurmadan doğrudan kontrolcünün ağdaki bağlantıların gecikme değerlerini gözlemleme ve gecikme değişimlerini tespit etme yeteneğinden faydalanılmasıdır. Gerçekleştirilen testlerde ağdan toplanan örnek veri ne kadar fazla olursa yanlış – negatif hata oranının o kadar azaldığı, saldırı tespit hassasiyetinin arttığı ve ağ topolojisinin şekli ve büyüklüğü değiştikçe saldırı tespitinde kullanılacak eşik değerinin yeniden hesaplanması gerektiği tespit edilmiştir. Çalışma içerisinde geliştirilen yöntemin kontrolcü üzerinde oluşturacağı ilave işlem yüküne yönelik herhangi bir veri paylaşılmamıştır.

Azzouni ve diğ. (2017) OpenFlow Keşif Protokolünün (OFDP) ciddi güvenlik açıklıkları barındırdığını, verimsiz ve işlevsel yönden kısıtlı bir protokol olduğunu değerlendirerek sOFTD (Secure OpenFlow Topology Discovery) ismini verdikleri güvenli bir topoloji keşif protokolü önermişlerdir. Bu protokol keşif sürecinin kontrolcüler yerine minimal değişiklikle keşif yeteneği kazandırılmış ağ anahtarları tarafından gerçekleştirilmesini temel almaktadır. Geleneksel ağlarda LLDP mesaj selini önlemek için sıkça kullanılan uç cihazların erişim portlarından gelen LLDP mesajlarının filtrelenmesi yöntemi de kullanılmaktadır. Ağ keşfi için kontrolcülerin varsayılan süre aralıklarıyla LLDP mesajı göndermesi pratiğinin kullanılmadığı sOFTD protokolünde LLDP paketlerinin sadece kontrolcü tarafından anahtar-port eşleşmelerine yönelik bir değişiklik tespit edildiğinde şifrelenerek ve daha kısa timeout (örn. 1) değerleriyle gönderilmesini ön görmüşlerdir. Çalışmada test sonuçları, anahtar üzerinde sOFTD ile oluşan ilave işlem yükü vb. veriler paylaşılmadığından sOFTD protokolünün ne derece etkili olduğu tespit edilememiştir.

Alimohammadifar ve diğ. (2018) kontrolcü üzerinde oluşturulan sahte bağlantıları, bu bağlantıları oluşturma yönteminden bağımsız olarak tespit etmek amacıyla tetkik (probing) tabanlı bir bağlantı doğrulama yöntemi önermişlerdir. Gizli tetkikle doğrulama (SPV – stealthy probing verification) adını verdikleri bu yöntemde normal trafikten ayırt edilemeyecek şekilde tasarlanmış gizli tetkik paketleri gönderilerek gerçek bağlantılar aşamalı olarak doğrulanmakta ve sahte bağlantılar tespit edilmektedir. Yöntemin işlevselliğinin tespiti amacıyla OpenDaylight kontrolcü tarafından yönetilen Mininet üzerinde kurulmuş bir benzetilmiş bir yazılım tanımlı ağda gerçekleştirilen testlerde yöntemin başarıyla çalıştığı görülmüştür. Bulut tabanlı gerçek bir SDN topolojisinde yöntem daha ayrıntılı olarak test edilmiş ve SPV'nin hem gerçek hem de benzetilmiş deney ortamında neredeyse gerçek zamanlı (120 ms'den daha az) olarak sahte bağlantıları tespit ettiği, bu yönüyle önerilen çözümün büyük ağ topolojileri için de ölçeklenebilir bir yöntem olduğu değerlendirilmiştir.

Marin ve diğ. (2019) bilinen LLDP saldırılarının dışında Floodlight kontrolcünün ağ topolojisi keşfini yürüten modüllerinde bulunan yazılımsal açıklıkları istismar edecek iki farklı tip saldırı yöntemi daha tanımlamışladır. Tersine döngü (reverse loop) adı verilen yöntemde ağdaki anahtarlar arasında bulunan bağlantıları keşfeden ve kaydını tutan bağlantı keşif hizmeti

(LDS – Link Discovery Service) tek yönlü bir bağlantı tespit etmesi halinde bağlantının çift yönlü olup olmadığının kontrolü için tekrar LLDP paketi göndererek bağlantı keşfi yapması işleminde saldırgan tarafından LINK-TYPE alanı değiştirilmiş bir LLDP paketinin kontrolcüye gönderilmesi durumunda kontrolcünün ağ topolojisini sonsuz bir döngü içerisinde sürekli hesaplamaya başlaması ve bir süre sonra kontrolcünün sistem kaynaklarının tükenmesi sağlanmaktadır. Topoloji dondurma adı verilen yöntemdeyse topolojinin her yeni bağlantı tespiti sonucu yeniden hesaplanarak güncellenmesi işlemine müdahale edilmesi söz konusudur. Bir S1 anahtarının portunun iki farklı anahtarın portuna bağlantısı olduğunu içeren sahte LLDP mesajları oluşturularak kontrolcüye gönderilmesi halinde LDS bu bağlantıları kabul etmekte ve S1 anahtarının portunu genel yayın etki alanı portu olarak sınıflandırmaktadır. Bu durumda LDS S1 anahtarının portunu topolojinin graf yapısından silmekte ancak rota hesaplamasında eski topolojiye göre bu bağlantıları hesaplamaya almaya devam etmekte, bu durum sistematik hataya yol açarak topoloji güncellemesini engellemektedir.

Bui ve diğ. (2019) üzerinde sıkça çalışılan uç cihaz kaynaklı saldırılar yerine ağ anahtarlarının bir şekilde (konsol bağlantısı, parola kırma vb.) saldırganlar tarafından ele geçirilmesi halinde gerçekleştirilecek LLDP saldırılarının ağa vereceği zararın neler olacağı üzerine odaklanan bir çalışma gerçekleştirmiştir. Güvenliği ihlal edilmiş tek bir anahtarın dahi ağdaki trafiği üzerinden geçirerek saldırganın trafiği dinlemesini sağlamak amacıyla topoloji zehirleme saldırısı gerçekleştirebileceğini tespit etmişlerdir. Bununla birlikte rota tespitinin belirsizliği (deterministik olmayışı) ve ağda yapılandırılacak yük dengeleme faaliyetinin bu saldırıların etkisini azaltabileceği ve ağın farklı noktalarındaki trafiğin daha küçük bir anahtar grubuna yönlendirilmesini zorlaştıracağı değerlendirilmiştir. Özellikle rota tespitinde rastgele rota değişimi tekniğinin (Duan ve diğ., 2013) kullanımının bu tip saldırılara yönelik oldukça etkili bir tedbir olduğu tespit edilmiştir.

Nehra (2019) doktora tezi çalışmaları kapsamında topoloji keşif sürecinin güvenliğinin sağlanmasında işlemci maliyet gerektiren kriptografik ve sabit cihaz-port eşleştirme yöntemleri yerine daha az maliyetli ve kriptografik yöntemler içermeyen bir çözüm geliştirmeyi amaçlamıştır. Akademik alanda sıkça üzerinde çalışılan POX, Ryu, OpenDayLight, Floodlight, Beacon, ONOS ve HPE-VAN kontrolcülerinin LLDP istismarına dayanan saldırılara karşı performanslarını gözlemlemiştir. LLDP iletişiminde kimlik doğrulama ve mesaj bütünlüğünün kontrolünün bulunmamasının bu protokolün istismarına neden olan başlıca faktörler olduğunu tespit etmiş, LLDP paketlerinin kimlik ve bütünlük kontrolünü sağlamak maksadıyla kontrolcü tarafından her bir LLDP mesajına jeton (token) verilerek LLDP mesajının geri dönüşünde bu jetonun kontrol edilmesine dayanan bir mekanizma önerisinde bulunmuştur. TILAK adını verdiği bu mekanizmanın uygulandığı yazılım tanımlı ağ anahtarları ile yapılan testlerde TILAK'ın topoloji keşif işlemlerinde anahtarların kendi sistemlerine oranla daha az işlemci kanyağı tükettiği ve daha hızlı keşif işlemlerini gerçekleştirdiği tespit edilmiştir.

Wang ve diğ. (2020) yazılım tanımlı ağ mimarisiyle tasarlanmış taşıtların internetinde (internet of vehicles- IoV) topoloji zehirleme saldırılarının etkilerini incelemek maksadıyla oluşturdukları test topolojisinde dört popüler kontrolcüye (POX, Ryu, Floodlight, OpenDaylight) ayrı ayrı LLDP mesaj yeniden gönderim ve LLDP sahte mesaj enjeksiyonu saldırıları düzenlemiş, saldırıların yazılım tanımlı taşıt ağının dört ana katmanı olan uygulama katmanı, kontrolcü katmanı, yol üstü birimler katmanı (RSU – roadside unit) ve taşıt katmanındaki etkilerini gözlemlemiştir. Saldırı sonucu oluşturulan sahte bağlantıların rota hesaplamalarına dahil edilmesiyle ağ trafiğinin ağın belirli noktalarında yoğunlaştığı ve yüksek miktardaki trafiğin bu noktalarda bulunan ağ cihazlarının kaynaklarını tüketebileceği ve ağ cihazlarının hizmet dışı kalabileceği tespit edilmiştir. Taşıt kenar ağlarında LLDP saldırılarını tespit eden ve söz konusu saldırılar sonrasında ağın kendi kendini iyileştirmesini sağlayan derin pekiştirmeli öğrenme tabanlı bir saldırı tolerans yöntemi geliştirmişlerdir. Yöntemin etkinliğini tespit etmek üzere gerçekleştirilen testlerde yöntemin ağın saldırı tespiti ve ağın iyileştirilmesi

görevlerini başarıyla yerine getirdiği, ağdaki düğümlerin sayısının artmasının yöntemin işlerliğine olumlu katkı yaparak saldırıların etkisini azalttığı gözlemlenmiştir.

Chou ve diğ. (2020) OFDP'nin güvensiz yapısından kaynaklanan riskleri azaltarak LLDP istismarına dayanan saldırıların önlenmesi amacıyla bir korelasyon tabanlı topoloji anomalileri tespit mekanizması önermişlerdir. Ağ trafiğinin analiz edilmesi ve her bir LLDP çerçevesinin gidiş dönüş süresinin ölçülerek ortadaki adam saldırısı vasıtasıyla topoloji keşfi yapılıp yapılmadığı tespit edebilmek amacıyla Spearman sıra korelasyonundan faydalanan bu çalışma kapsamında LLDP yapısında bir kimlik doğrulama mekanizması bulunmaması dikkate alınarak dinamik bir kimlik doğrulama anahtarı ve sayma mekanizması LLDP çerçevelerine eklenmiştir. Saldırganların kendi ürettikleri sahte LLDP mesajları vasıtasıyla kontrolcüye sahte ağ ekletmesi bu kimlik doğrulama anahtarlarının kontrolüyle engellenirken yakalanan gerçek LLDP çerçevelerinin yeniden gönderimi yoluyla sahte ağ ekletmesini engellemek amacıyla çerçevelerin geliş gidiş süreleri, eklenmeye çalışılan bağlantının işlerlik durumu ve rota bilgisi korelasyon işlemine tabi tutulmuştur. LLDP mesaj seli saldırılarıysa saldırının geldiği porttaki aşırı LLDP trafiğinin tespiti ve portun paket iletimine kapatılması vasıtasıyla gerçekleştirilmiştir.

6. TARTIŞMA

Yapılan testler ve elde edilen bulgular açıkça göstermiştir ki LLDP istismarına dayanan saldırılar yazılım tanımlı ağlarda ciddi sonuçlara yol açabilmektedir. Ağın genel başarımını azaltma ve uç cihazların erişimlerinde kesintiye neden olmanın yanı sıra bu saldırılar ortadaki adam saldırılarına da zemin hazırlayabilmektedir.

İcra edilen testlerin ilk aşamasında oluşturulan basit topolojide LLDP paketlerinin yeniden gönderimi ile icra edilen saldırıda uzak sunucuya erişmeye imkân sağlayacak faal bir bağlantısı bulunan H4 cihazı, saldırı nedeniyle değişen rota yüzünden S1 anahtarının 3 numaralı portuna erişememiş ve uzak sunucuya bağlantı sağlayamamıştır. Kontrolcü kandırılması sebebiyle gerçekte bulunan faal bağlantıları dikkate almayarak sahte bağlantı üzerinden en kısa rotayı oluşturmuştur. Bu bilgiler ışığında her ne kadar Floodlight kontrolcüsünde LLDP saldırılarını önlemeye yönelik çeşitli tedbirler alındığı bilinse de (Nehra ve diğ., 2017) kontrolcü varsayılan yapılandırma ayarları ile çalıştırıldığı takdirde bu tedbirlerin yarar sağlamadığı gibi kontrolcünün erişim kesintilerinde alternatif rota oluşturma çabasının da bulunmadığı tespit edilmiştir. Bu kapsamda Floodlight kontrolcüsünün varsayılan olarak güvenli yapılandırılma konusunda yetersiz olduğu değerlendirilmiştir.

Testlerin ikinci aşamasında farklı ağ topolojilerinin LLDP saldırılarından nasıl etkilendiğini tespit etmek amacıyla beş katlı basit ağaç topolojisinde iki host arasında LLDP mesaj yeniden gönderim saldırısı icra edilmiş ve kontrolcüye sahte bağlantı eklenmesi sonucunda ağın başarımının %21 oranında azaldığı görülmüştür. Ağın bu tip saldırılardan etkilenme derecesi oluşturulan sahte bağlantının yeri ve ağın topolojisiyle doğrudan ilişkilidir. Örneğin simetrik bir yapıya sahip olan basit ağaç topolojisinde sahte bağlantının eklendiği yer cihaz-cihaz rotalarında kesintiye neden olsa dahi hiçbir uç cihazın ağın varsayılan ağ geçidine erişimi kesilmemiştir. Bu durum tamamen bu çalışmaya özgü bir durum olup ağın omurga anahtarına veya ağ geçidine yakın portlardan eklenecek sahte bağlantıların daha fazla sayıda cihaz-cihaz ve ağ geçidi-cihaz rotalarının erişilmez hale gelmesine neden olacağı değerlendirilmektedir.

Alharbi ve diğ. (2015) tarafından yapılan çalışmada POX kontrolcüsünün tek yönlü linkleri yönlendirmede dikkate almadığı ve LLDP ataklarında sahte LLDP mesaj enjeksiyonunun ağa bağlı uç sistemlerin ağa erişiminde bir değişiklik yaratmayacağı tespit edilmişse de Floodlight kontrolcü için böyle bir durumun söz konusu olmadığı, hem tek yönlü hem de çift yönlü sahte sahte link eklendiğinde H1 ve H3 sistemlerinin ağ erişiminin kesildiği tespit edilmiştir. Bu açıdan değerlendirildiğinde Floodlight kontrolcüsünün sahte LLDP mesaj enjeksiyonu saldırılarından POX kontrolcüsüne nazaran daha fazla etkilendiği tespit edilmiştir. Floodlight 1.2 sürümü ile birlikte, gerekli yapılandırma ayarları uygulandığı takdirde LLDP paketleri

zaman damgalı ve özet değerleri ile birlikte iletilebilmekte, bu sayede LLDP çerçevelerinin bütünlüğü kontrol edilebilmekte ve paket gecikme süreleri hesaplanabilmektedir. Ancak bu tedbirlerin LLDP kaynaklı riskleri tamamen ortadan kaldırmadığı göz önünde bulundurulmalıdır. Bir saldırganın alınan tedbirlerle sahte LLDP çerçevesi üretmesi engellense dahi gerçek LLDP paketlerini ele geçirmesi ve kullanması mümkündür.

Geleneksel ağlarda daha çok cihaz komşuluk ilişkilerinin kurulması amacıyla kullanılan LLDP protokolünün ağda çalışan cihazların tespiti konusunda sağladığı avantaj, bu protokolün saldırganların keşif gayretlerini destekleyebileceği bilinmektedir. Uç cihazlardan geleneksel bir ağa LLDP saldırısı yapmayı önlemenin endüstriyel anlamda en etkili kabul edilen yöntemi uç cihaz bağlı anahtar portlarına bu uç cihazlardan gelecek LLDP çerçevelerinin dikkate alınmadan ve iletilmeden engellenmesine yönelik bir yapılandırma ayarı girilmesidir. Ağ anahtarı üzerinde LLDP mesaj iletimi global olarak kapatılabildiği gibi her bir arayüz için ayrı ayrı LLDP mesajlarına ne işlem yapılacağını (düşür/geçir) belirlemek mümkündür (Odom, 2020). Bu yöntemin en büyük kusuru ölçeklenebilir olmaması ve ağa ilave anahtar eklendiğinde bu durumdan etkilenecek diğer anahtar portlarında yapılandırmanın elle değiştirilmesi zorunluluğudur.

Yazılım tanımlı ağlarda standart olarak kabul görmüş bir LLDP güvenlik prosedürü bulunmamaktadır. Ancak bir kısım kontrolcülerde geleneksel ağ anahtarlarında belirli portlardan LLDP mesaj alımını operatör müdahalesiyle kısıtlamaya yarayan API'ler bulunmaktadır (Hong ve diğ., 2015). Bu API'ler yardımıyla Open vSwitch anahtarların uç cihazlara bağlanan portlarından LLDP mesajlarının yayınlanmasının engellenmesi mümkündür. Bu yöntem küçük boyuttaki yazılım tanımlı ağlarda sürdürülebilir bir çözüm olarak kabul edilebilir. Ancak büyük, sürekli topolojisi değişen ve ölçeklenebilirlik istenen ağlarda bu tür statik yöntemlerin uygulanabilirliği düşüktür ve sık sık operatör müdahalesi gerektirmektedir. Literatürde mevcut uygulama dışında önerilen diğer yöntemler incelendiğinde her bir çözüm önerisinin LLDP saldırılarını önlemede avantaj sağlarken başka alanlarda dezavantajları beraberinde getirdiği gözlemlenmiştir.

Uç cihazların bağlı olduğu portlardan LLDP mesajlarının geçişinin engellenmesi yöntemini bir adım ileriye taşıyan Hong ve diğ. (2015) portlardan LLDP mesajı yayımının filtrelenmesi işlemini ağın yapısına göre dinamik olarak gerçekleştirecek şekilde otomatize etmiştir. Hem LLDP sahte mesaj saldırısı hem de LLDP mesajı yeniden gönderim saldırısı başarıyla engellenerek ağ topolojisinin bütünlüğü korunmuştur. Ancak çalışma kapsamında icra edilen testler üç anahtar – iki uç cihazdan müteşekkil nispeten basit bir ağda gerçekleştirilmiştir. LLDP mesajlarına bir defaya mahsus HMAC eklenmesi işleminde LLDP mesaj üretim süresinin %80 arttığı gözlemlenmiştir. Bu durumda önerilen yöntemin çok katmanlı büyük ağlarda nasıl bir performans göstereceği bilinmemektedir.

LLDP'de kimlik doğrulama mekanizmasının bulunmamasını dikkate alarak önerilen kriptografik tekniklerde kontrolcü ve ağ anahtarı üzerinde ilave modüller kurulmasına ihtiyaç duyulduğu ve kimlik doğrulama aşamasında gereken şifreleme / şifre çözme işlemlerinin kontrolcü üzerinde yük oluşturacağı aşikardır. Örneğin Alharbi ve diğ. (2015) tarafından önerilen çözümde işlemci üzerinde oluşacak işlem yükünün %8 seviyesinde olduğu dikkate alındığında, bu yükün paket iletme, paket filtreleme vb. diğer ağ operasyonlarının performansında olumsuz etki yaratacağı değerlendirilmektedir.

Derin öğrenme ile ağ trafiğinin öğrenilerek saldırıların tespit edilmesine dayanan yöntemlerde öğrenme ve tespit mekanizmalarının nereye kurulacağı ve nasıl bir işlem yükü yaratacağı göz önünde bulundurulmalıdır. Örneğin Wang ve diğ. (2020) tarafından ağdaki trafiğin dinamik olarak gözlemlenerek derin öğrenme yöntemleri ile saldırı tespiti yapılması önerisinde öğrenme ve saldırı tespit işlemleri kontrolcü üzerine kurulan derin pekiştirmeli öğrenme ajanı tarafından gerçekleştirilmektedir. Çalışmada bu ajanın kontrolcü üzerinde ne kadar ilave kaynağa ihtiyaç duyacağı, ne kadar işlem yükü oluşturduğuna değinilmemiştir ancak kontrolcüye ciddi bir yük getireceği açıktır. Ağdaki tüm paket iletim kararlarını alan

kontrolcünün ayrıca bu tür bir işlem yükünü de üstlenmesinin ağdaki diğer operasyonlara nasıl etki edeceği dikkate alınmalıdır. Ayrıca derin öğrenmeye dayanan yöntemlerin geliştirilmesinde kullanılan veri setlerinin gerçek bir trafiği tamamen temsil etmesi mümkün değildir. Geliştirilen çözümün diğer topolojilerde çalışma performansı da gözlemlenmelidir.

LLDP'nin diğer ağ protokollerine kıyasla daha statik ve ön görülebilir bir protokol olması bu protokolü istismar eden saldırıları önlemede istatistiksel çözümlerin de kullanılabilir olmasının önünü açmaktadır. Chou ve diğ. (2020) LLDP paketlerinin ağda dolaşım süresini, eklenmeye çalışılan bağlantının faal durumda olup olmadığını ve mevcut rota bilgisini Spearman sıra korelasyonuna tabi tutarak saldırı tespit etmesi istatistiksel çözümlere yönelik bir örnektir. Yazarların yöntemin performansı ve kontrolcü üzerinde oluşturduğu ilave yük konusunda bilgi vermemişlerdir, bu nedenle yöntemin kriptografik ve derin öğrenme yöntemlerinden üstün olup olmadığı belirlenememiştir.

Tespit edilen bulgular ışığında yazılım tanımlı ağlarda LLDP kaynaklı topoloji zehirleme saldırılarını engellemeye yönelik geliştirilecek bir çözümde bulunması gereken temel özellikler aşağıda sunulmuştur.

a. Ağın topolojisi doğru bir şekilde kontrolcü tarafından bilinmeli, ağda meydana gelen değişiklikler elle müdahale edilmeden dinamik olarak öğrenilmelidir.

b. Gerçek LLDP mesajları sahtelerinden ayırt edilebilmelidir. LLDP mesajlarının kaynağı takip edilmeli, LLDP mesajı göndermemesi gereken uç cihazlardan gelen mesajlar bağlantı keşfinde dikkate alınmamalı ve bir saldırı olarak nitelendirilmelidir.

c. Bir şekilde kontrolcü üzerinde sahte bağlantı kaydı oluşturulması halinde bu bağlantı tetkik edilmeli ve bağlantının faal olmadığı doğrulanarak kontrolcü üzerinden silinmelidir.

d. Kontrolcü tarafından hesaplanan rotaların sahte bağlantıları hesaplamaya dahil etmemesi sağlanmalıdır.

e. Çözüm anahtarlara ve uç cihazlara ilave bir ajan / uygulama kurulumu gerektirmemeli ve kontrolcüde çalışması halinde kontrolcünün ağdaki diğer görevlerini aksatmayacak şekilde mümkün olan en az kaynakla çalışmalıdır.

7. SONUÇ

Bu çalışmada Floodlight kontrolcüsü tarafından kontrol edilen yazılım tanımlı bir ağda sahte LLDP mesaj enjeksiyonu ve LLDP mesaj yeniden gönderim saldırıları icra edilerek kavram ispatı sağlanmış ve Floodlight kontrolcüsü üzerinde sahte bağlantı kayıtları oluşturulmuştur. Yapılan testler neticesinde, LLDP istismarına dayalı topoloji zehirleme saldırılarının yazılım tanımlı ağlara yönelik hizmet dışı bırakma saldırılarında bir saldırı vektörü olarak kullanılabileceği ispatlanmıştır. Saldırının icra edildiği ağ topolojisinin ve saldırı sonucu oluşturulacak sahte linklerin yerleşiminin saldırının yaratacağı etkiye doğrudan tesir ettiği, büyük ağlarda dağıtım katmanına yakın ağ cihazlarını hedef alan saldırıların ağ topolojisinin kontrolcü tarafından tümüyle yanlış öğrenilmesine neden olarak ağın işlerliğini daha fazla sekteye uğratacağı tespit edilmiştir. Ayrıca sahte bağlantıyı dikkate alarak hesaplanan rotaları kullanan cihazların ağ paketlerinin saldırganların ele geçirdikleri sistemler tarafından dinlenebildiği, bu durumun ortadaki adam saldırılarına zemin hazırladığı görülmüştür.

Yazılım tanımlı ağlarda LLDP güvenliğini sağlamaya yönelik çeşitli akademik çalışmaların yapıldığı, bu çalışmalarda istatistiksel yöntemlerin yanı sıra kriptografik ve derin öğrenmeye dayalı yöntemlerin geliştirildiği, LLDP'nin yerine geçecek ve LLDP'nin mevcut yapısından kaynaklanan zafiyetleri ortadan kaldıracak güvenli keşif protokollerinin önerildiği ancak halen kabul görmüş standart bir LLDP güvenliği prosedürünün bulunmadığı tespit edilmiştir. Buna rağmen mevcut koşullarda alınacak birtakım tedbirlerle LLDP istismarına dayanan saldırılardan kaynaklanan riskleri azaltmak mümkündür. Bu tedbirlerin en başında ağ cihazlarının güvenliğinin sağlanması gelmektedir. Anahtarlar parola, kullanıcı adı, SSH erişimi vb. tedbirlerle güvenlik bakımından sıkılaştırılarak saldırganlar tarafından ele geçirilemeyecek şekilde yapılandırılmalı, ağ anahtarlarına yetkisiz erişim engellenmelidir. Kontrolcüde bulunan

API'ler yardımıyla anahtarlara bağlı uç cihazlardan LLDP mesajı gönderimi ve bu cihazların anahtarlardan gerçek LLDP mesajı alması engellenmelidir. Tespit edilen erişim kesintileri araştırılmalı ve sahte bağlantı kayıtlarının bu kesintilere neden olabileceği göz önünde bulundurulmalıdır.

Gelecekte yazılım tanımlı ağ teknolojisinin internet altyapısında daha fazla kullanılacağı göz önünde bulundurulduğunda LLDP istismarına dayalı saldırıların etkisinin azaltılması büyük önem taşımaktadır. Bir kısım kontrolcülerde bu konuda tedbir alındığı görülse dahi bu tedbirlerin ilave yapılandırma veya modül gerektirdiği tespit edilmiştir. LLDP güvenliğinin "varsayılan olarak güvenli" tasarım prensibinin bir gereği olarak tüm kontrolcülerde maliyet etkin ve başarı oranı yüksek çözümlerle sağlanması gerekmektedir. Bu çözümler geliştirilirken geleneksel ve yazılım tanımlı ağ anahtarlarının bir arada bulunduğu hibrit topolojilerde de faaliyet gösterilmesi zorunluluğu dikkate alınmalıdır.

Bu çalışmada elde edilen bulgular ışığında gelecekte yapılacak çalışmalarda ağın başarımına olumsuz etki etmeyen, ağ cihazlarının mevcut kaynakları ile çalışabilen, ağda asgari trafik ve ağ cihazlarında asgari işlem yükü yaratan bir çözümün geliştirilebilmesinin önemli olduğu düşünülmektedir. Bu çözümün ağdaki uç cihazların türünden bağımsız olarak çalışması, kontrolcünün ağdaki diğer görevlerini aksatmaması, değişen topolojilere duyarlı olması, ölçeklenebilir olması, anahtarlara ve uç cihazlara ilave ajan kurulumu gerektirmemesi, her tip kontrolcüye uyarlanabilmesi ve her tip topolojide çalışabilmesi söz konusu çözümün temel bir güvenlik standardı olarak kabul görmesine katkı sağlayacaktır.

ÇIKAR ÇATIŞMASI

Yazarlar, bilinen herhangi bir çıkar çatışması veya herhangi bir kurum/kuruluş ya da kişi ile ortak çıkar bulunmadığını onaylamaktadırlar.

YAZAR KATKISI

Mevlüt Serkan TOK çalışmanın kavramsal ve tasarım süreçlerinin belirlenmesi ve yönetimi, veri toplama, veri analizi ve yorumlama, makale taslağının oluşturulması kategorilerinde; Mehmet DEMİRCİ kavramsal ve tasarım süreçlerinin yönetimi, veri analizi ve yorumlama, fikirsel içeriğin eleştirel incelenmesi kategorilerinde katkı sağlamıştır.

KAYNAKLAR

1. Abazari, F., Esposito, F., Takabi, H., Hosseinvand, H., Pecorella, T. (2019). Teaching software-defined network security through malicious tenant detection. *Internet Technology Letters*, 2(6). doi: 10.1002/itl2.131.

2. Alharbi, T., Portmann, M., Pakzad, F. (2015). The (in)security of Topology Discovery in Software Defined Networks. *2015 IEEE 40th Conference on Local Computer Networks (LCN).* doi:10.1109/lcn.2015.7366363.

3. Alimohammadifar, A., Majumdar, S., Madi, T., Jarraya, Y., Pourzandi, M., Wang, L., Debbabi, M. (2018). Stealthy Probing-Based Verification (SPV): An Active Approach to Defending Software Defined Networks Against Topology Poisoning Attacks. *Computer Security Lecture Notes in Computer Science*, 463–484. doi: 10.1007/978-3-319-98989-1_23

4. Azzouni, A., Trang, N. T. M., Boutaba, R., Pujolle, G. (2017). Limitations of openflow topology discovery protocol. *2017 16th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*. doi: 10.1109/medhocnet.2017.8001642.

5. Baidya, S. S., Hewett, R. (2020). Link discovery attacks in software-defined networks: Topology poisoning and impact analysis. *Journal of Communications*, 596-606. doi:10.12720/jcm.15.8.596-606

6. Bierman, A., & Jones, K. (2000). Physical topology mib. Retrieved March 20, 2021, from https://tools.ietf.org/html/rfc2922

7. Bui, T., Antikainen, M., & Aura, T. (2019). Analysis of topology poisoning attacks in software-defined networking. *Secure IT Systems*, 87-102. doi:10.1007/978-3-030-35055-0_6

8. Chou, L., Liu, C., Lai, M., Chiu, K., Tu, H., Su, S., . . . Tsai, W. (2020). Behavior anomaly detection in sdn control plane: A case study of topology discovery attacks. *Wireless Communications and Mobile Computing, 2020*, 1-16. doi:10.1155/2020/8898949

9. Cisco Systems. (2006, June 29). LLDP-MED and cisco discovery Protocol [IP telephony/voice over IP (VoIP)]. Retrieved March 20, 2021, from https://www.cisco.com/en/US/technologies/tk652/tk701/technologies_white_paper0900aecd804cd46d.html

10. Combe, T., Mallouli, W., Cholez, T., Mathieu, B., & Oca, E. (2018). An SDN and NFV Use Case: NDN Implementation and Security Monitoring. In Zhu, S. Scott-Hayward, L. Jacquin, R. Hill (Eds.), *Guide to Securıty in SDN and NFV: Challenges, opportunities, and applications*. SPRINGER INTERNATIONAL PU.

11. Congdon, P. (2002). Link layer discovery protocol and MIB. *V1. 0 May*, *20*(2002), 1-20. doi: 10.1.1.564.3010.

12. Demirci, S., Yurekten, O., Demirci, M. (2019). Yazılım Tanımlı Ağlar ve Siber Güvenlik. *In S. Sagiroglu (Ed.), Siber Güvenlik ve Savunma: Standartlar ve Uygulamalar* (pp. 123–144). Bilgi Güvenliği Derneği, Ankara.

13. Duan, Q., Al-Shaer, E., Jafarian, H. (2013). Efficient random route mutation considering flow and network constraints. *2013 IEEE Conference on Communications and Network Security (CNS)*. doi:10.1109/cns.2013.6682715

14. Feamster, N., Rexford, J., Zegura, E. (2014). The road to sdn: An intellectual history of programmable networks. *ACM SIGCOMM Computer Communication Review*, *44*(2), 87-98. doi:10.1145/2602204.2602219

15. Göransson Paul, Black, C., & Culver, T. (2017). *Software defined networks: a comprehensive approach*, Amsterdam, Elsevier.

16. 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.

17. Hong, S., Xu, L., Wang, H., Gu, G. (2015). Poisoning Network Visibility in Software-Defined Networks: New Attacks and Countermeasures. *Proceedings 2015 Network and Distributed System Security Symposium*. doi: 10.14722/ndss.2015.23283.

18. Hu, F., Hao, Q., & Bao, K. (2014). A survey on software-defined network and openFlow: from concept to implementation. *IEEE Communications Surveys & Tutorials*, 16(4), 2181-2206.

19. Kaur, N., Singh, A. K., Kumar, N., Srivastava, S. (2017). Performance impact of topology poisoning attack in SDN and its countermeasure. *Proceedings of the 10th International Conference on Security of Information and Networks - SIN '17*. doi:10.1145/3136825.3136881.

20. Krawczyk, H., Bellare, M., & Canetti, R. (1997). HMAC: Keyed-Hashing for message authentication. doi:10.17487/rfc2104

21. Marin, E., Bucciol, N., & Conti, M. (2019). An in-depth look into sdn topology discovery mechanisms. *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. doi:10.1145/3319535.3354194

22. Mininet, (2018). Mininet. Erişim Adresi: https://mininet.org (Erişim Tarihi: 15.06.2020).

23. Nehra, A., Tripathi, M., Gaur, M. S. (2017). 'Global view' in SDN: Existing implementation, vulnerabilities & threats. *Proceedings of the 10th International Conference on Security of Information and Networks - SIN '17*. doi:10.1145/3136825.3136862.

24. Nehra, A. (2019). *Archıtectural Improvements For Secure Sdn Topology Dıscovery* (Yayınlanmamış doktora tezi). India / Malaviya National Institute of Technology Jaıpur. http://idr.mnit.ac.in/bitstream/handle/123456789/941/2014RCP9553-Ajay%20Nehra.pdf?sequence=1&isAllowed=y (Erişim Tarihi: 23 Mart 2021)

25. Ochoa-Aday, L., Cervelló-Pastor, C.,Fernández-Fernández, Adriana. (2015). Current Trends of Topology Discovery in OpenFlow-based Software Defined Networks. doi: 10.13140/RG.2.2.12222.89929.

26. Odom, W. (2020). *CCNA 200-301 Official Cert Guide, Volume 2*. Hoboken, NJ: Cisco Press.

27. Pakzad, F., Portmann, M., Tan, W. L., & Indulska, J. (2016). Efficient topology discovery in openflow-based software defined networks. *Computer Communications, 77*, 52-61. doi:10.1016/j.comcom.2015.09.013

28. Smyth, D., Mcsweeney, S., Oshea, D., Cionca, V. (2017). Detecting Link Fabrication Attacks in Software-Defined Networks. *2017 26th International Conference on Computer Communication and Networks (ICCCN).* doi: 10.1109/icccn.2017.803843.

29. Tok, M.S., Demirci, M. (2021). Security analysis of SDN controller-based DHCP services and attack mitigation with DHCPguard, *Computers & Security*, 109, 102394, doi:10.1016/j.cose.2021.102394.

30. Vyncke, E., Paggen, C. (2008). *LAN switch security: What hackers know about your switches*. Indianapolis, Cisco Press.

31. Wang, J., Tan, Y., Liu, J., Zhang, Y. (2020). Topology Poisoning Attack in SDN-enabled Vehicular Edge Network. IEEE Internet of Things Journal, 1–1. doi: 10.1109/jiot.2020.2984088.

32. Wang X., Gao N., Zhang L., Liu Z., Wang L. (2016) Novel MITM Attacks on Security Protocols in SDN: A Feasibility Study. *2016 International Conference on Information and Communications Security*. doi: 10.1007/978-3-319-50011-9_35.