T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ETHERCAT TABANLI BİR SCADA SİSTEMİNDE KURAL VE MAKİNE ÖĞRENMESİNE DAYALI SALDIRI VE ANOMALİ TESPİTİ

DOKTORA TEZİ

Kevser OVAZ AKPINAR

Enstitü Anabilim Dalı	**:**	**BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ**
Tez Danışmanı	**:**	**Doç. Dr. İbrahim ÖZÇELİK**

Mayıs 2019

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ETHERCAT TABANLI BİR SCADA SİSTEMİNDE KURAL VE MAKİNE ÖĞRENMESİNE DAYALI SALDIRI VE ANOMALİ TESPİTİ

DOKTORA TEZİ

Kevser OVAZ AKPINAR

Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ

Bu tez 29 / 05 /2019 tarihinde aşağıdaki jüri tarafından oybirliği/oyçokluğu ile kabul edilmiştir.

Prof. Dr.
Celal ÇEKEN
Jüri Başkanı

Prof. Dr.
Cemil ÖZ
Üye

Doç. Dr.
Ahmet SAYAR
Üye

Prof. Dr.
Resul KARA
Üye

Doç. Dr.
İbrahim ÖZÇELİK
Üye

BEYAN

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

Kevser OVAZ AKPINAR

29.05.2019

TEŞEKKÜR

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

Anlayış ve yardımlarını esirgemeyen sevgili annem, babam ve kardeşim Samet OVAZ'a, kızlarım Azra ve Sare AKPINAR'a, ayrıca sonsuz desteğinden ötürü çok değerli eşim Mustafa AKPINAR'a teşekkür ederim.

Ayrıca bu çalışmanın maddi açıdan desteklenmesine olanak sağlayan Türkiye Bilimsel ve Teknolojik Araştırma Kurumuna (TÜBİTAK) (Proje No: 118E263) ve Sakarya Üniversitesi Bilimsel Araştırma Projeleri (BAP) Komisyon Başkanlığına (Proje No: 2015-50-02-025) teşekkür ederim.

İÇİNDEKİLER

SİMGELER VE KISALTMALAR LİSTESİ

AAKR	: Otomatik ilişkili çekirdek regresyonu
ACF	: Otokorelasyon fonksiyonu
ANN	: Yapay sinir ağları
BT	: Bilişim teknolojileri
CIA	: Gizlilik, bütünlük, erişilebilirlik
DCS	: Dağıtılmış kontrol sistemleri
DDOS	: Dağıtık servis durdurma
DL	: Veri bağlantısı
DNS	: Alan adı sistemi
DOS	: Servis durdurma
DT	: Karar ağaçları
ECAT	: EtherCAT protokolü
EKS	: Endüstriyel kontrol sistemleri
ELK	: Elasticsearch, Logstash, Kibana yığını
ENI	: EtherCAT ağ bilgisi
ESI	: EtherCAT köle bilgisi
ESL	: EtherCAT anahtar bağlantısı
EtherCAT	: Ethernet for Control Automation Technology
FCS	: Çerçeve kontrol dizisi
FN	: Yanlış negatif
FNR	: Yanlış negatiflik oranı
FP	: Yanlış pozitif
FPGA	: Alanda programlanabilir kapı dizileri
FSM	: Sonlu durum makinesi
HMI	: İnsan makine arabirimi
IDS	: Saldırı engelleme sistemi

IPS	: Saldırı önleme sistemi
IRQ	: Kesme isteği
k-NN	: k en yakın komşu
L-Critical/LL	: Alt kritik değer
LLDP	: Bağlantı katmanı keşif protokolü
MDNS	: Çok yöne yayınlı alan adı sistemi
MITM	: Ortadaki adam
PLC	: Programlanabilir lojik kontrolör
RTU	: Uzak terminal birimi
SCADA	: Merkezi denetleme kontrol ve veri toplama
SPRT	: İstatistiksel olasılık oran testi
SVM GA	: Genetik algoritma kullanılmış karar destek makinesi
TN	: Gerçek negatif
TP	: Gerçek pozitif
U-Critical/UL	: Üst kritik değer
VPN	: Özel sanal ağ
WKC	: Çalışma sayacı
YSA	: Yapay sinir ağları

ŞEKİLLER LİSTESİ

TABLOLAR LİSTESİ

ÖZET

Anahtar kelimeler: Endüstriyel kontrol sistemleri, SCADA, EtherCAT protokolü, Snort, IDS/IPS, önişlemci, sıfırıncı-gün saldırıları, anomali tespiti

Endüstriyel kontrol sistemleri (EKS) bulundukları konum ve bileşenleri bakımından kritik altyapıya sahip sistemler olup, bilişim teknolojilerinden (BT) bağımsız olarak uygulama alanına göre kendilerine ait kabul ve işleyişleri bulunmaktadır. Bu sistemler, günümüzde otomasyon hiyerarşisinde yer alan seviyeler arası yatay ve dikey entegrasyonun tek bir protokolle sağlanması fikrinden yola çıkılarak Ethernet ile de adapte edilmiş durumdadır. Dolayısıyla EKS'ler hem doğalarından hem de Ethernet üzerinden bilişim teknolojilerinin sunduğu hizmetlerin içerisine dahil edildiklerinden dolayı siber saldırılara karşı tehdit altındadır. Bu durum, çoğunlukla iletişim altyapısı üzerinden gelen saldırıların tespiti için özelinde EKS çözümlerini gerektirir.

Bu çalışmada, otomasyon uygulamalarında yaygın bir kullanıma sahip olan, Ethernet tabanlı gerçek zamanlı EtherCAT protokolü için Snort saldırı tespit sistemi üzerinde bilinen ve bilinmeyen saldırıları tespit eden bütüncül bir yapı ve makine öğrenmesi teknikleriyle anomali tespiti olmak üzere ikisi kural biri anomali tespitine dayanan 3 farklı yaklaşım sunulmaktadır. Sistem, geliştirilen önişlemci yardımıyla, bilinen saldırılar için güvenli düğüm yaklaşımı, bilinmeyen saldırılar için ise saha veri yolu tekrar periyodunu tespit ederek istatistiksel tekniklerle ve özgün çözümlerle kural tabanlı olarak saldırı tespitini kapsamaktadır. Tespitler bir günlükleme ve izleme yapısı olan ELK yığını üzerinde kullanıcıya sunulmaktadır. Ayrıca, yine bilinmeyen saldırılar için oluşturulan su seviye kontrol otomasyonu test ortamı üzerinde olaylar gerçeklenerek bir veri seti hazırlanması ve çeşitli öğrenme tekniklerinin veri seti üzerinde anomali tespitini kapsamaktadır.

Bilinmeyen saldırıların tespiti kapsamında uygulanan periyot tespitinin %95-%99 doğrulukla yapılabildiği görülmüştür. Önerilen sistem üzerinde ise MAC aldatma, veri enjeksiyonu, DoS, köle saldırıları gibi ataklar gerçeklenmiş, alarm ve günlüklemeler incelendiğinde saldırıların başarıyla tespit edildiği görülmüştür. Ayrıca, k-NN ve SVM GA tekniklerinin olay tespitinde başarılı sonuç verdikleri belirlenmiştir.

RULE AND MACHINE LEARNING BASED INTRUSION AND ANOMALY DETECTION IN AN ETHERCAT BASED SCADA SYSTEM

SUMMARY

Keywords: Industrial control systems, SCADA, EtherCAT protocol, Snort, IDS/IPS, preprocessor, zero-day attacks, anomaly detection

Industrial control systems (ICS) are critical infrastructures in terms of their location and components. These systems have their own features and operation related to the application field independent from the information technologies (IT). They are also adapted with the Ethernet technologies based on the idea of providing horizontal and vertical integration between the levels in the automation hierarchy with a single protocol. Therefore, ICSs are threatened by cyber attacks, due to both their nature and support of IT services through Ethernet. This risk requires ICS specific solutions to detect and prevent attacks which use communication infrastructure.

In this study, two rule based which detect known and unknown attacks on the Snort system and one anomaly based which uses machine learning techniques, in total of three different approaches were presented as a holistic structure for Ethernet based real-time EtherCAT protocol, which is widely used in automation applications. In the case of rule based intrusion detection, the EtherCAT preprocessor was proposed, which applies the trust node approach for known attacks, and identifies the field bus repetition period for unknown attacks, with statistical techniques and novel solutions. The findings were presented to the user on the ELK stack, which is a logging and monitoring structure. For anomaly based intrusion detection, the water level control automation testbed was developed, a dataset was prepared by generating events and various machine learning techniques were applied on the dataset.

According to the findings obtained in this research, it was concluded that the period determination which was applied within the scope of unknown attack detection can be made with 95% - 99% accuracy. When the logs and alerts of the realized MAC spoofing, data injection, DoS, slave attacks were investigated, it was seen that the attacks were able to be detected successfully. For anomaly detection part of the study, k-NN and SVM GA techniques were found to be successful in detecting events.

BÖLÜM 1. GİRİŞ

1.1. Endüstriyel Kontrol Sistemleri ve Güvenlik

Sürdürülebilirlik ve güvenliği sağlamak için kritik altyapı ağlarının sürekli işletilmesi ve izlenmesi gerekmektedir. Bu yapıyı sağlayan kritik altyapı varlıklarının tümüne endüstriyel kontrol sistemleri (EKS) denilmekte olup, sistemlerin kontrolü denetleyici kontrol ve veri toplama (SCADA) sistemleri tarafından sağlanmaktadır. SCADA sistemlerinin kullanılması 1960'larda başlamıştır. 1990'lı yıllarda ise sunduğu mimariler, merkezi hesaplama birimleri, farklı topoloji desteği, kolay kurulan ağ alt yapısı ve gerçek zamanlı desteği ile SCADA sistemler, kritik altyapı gerektiren ağlar üzerinde ağırlığını iyice hissettirmiş ve yaygınlığını arttırmıştır [1]. Günümüzde ise modern SCADA sistemleri olarak bilinen; açık dağıtık sistem mimarisine sahip, yedekli yapıda çalışan, endüstri standartları, LAN, iş istasyonları ve ağdaki diğer düğümler üzerinde çalışabilen dağıtık fonksiyonları destekleyen, TCP/IP iletişimi kullanan (RISC-UNIX-TCP/IP modelleri) yapılar kullanılmaktadır. Bu sistemler, doğalgaz/petrol boru hatları, enerji sektörü, bina endüstrisi, rüzgâr enerji sistemleri, solar sistemler, su toplama, dağıtım ve arıtım sistemleri, kimya, otomotiv, çimento vb. gibi prosesin ve üretimin olduğu birçok endüstri dalında kullanılmaktadır.

Bir SCADA sistemi, proses, programlanabilir kontrolör (PLC), döngü kontrolörleri, dağıtılmış kontrol sistemleri (DCS), giriş/çıkış sistemleri, kontrol ünitesi üzerinde bulunan eyleyici ve algılayıcı gibi saha cihazlarından aldığı gerçek verileri periyodik/aperiyodik ve gerçek zamanlı olarak telemetri veya kablolu iletişimle toplar, altyapı ve topoloji hiyerarşisine göre bu bilgileri işler ve gerekli sonuçları saha veya ofis iletişimi yoluyla uygular. SCADA sistemlerde köle-köle, efendi-köle ve topolojiye göre efendi-efendi arası iletişim yapılmaktadır.

EKS uygulamalarındaki donanımların otomasyon içinde kullanıldıkları konuma göre farklı gereksinimleri olduğundan, belirli bir hiyerarşik model takip edilerek projelendirilmesi gerekmektedir. Önceleri otomasyon hiyerarşisi veya bilgisayarla tümleşik üretim (CIM) modeli kullanılırken bu yapı sonradan Purdue modeli olarak bilinen, CIM'de tanımlanan seviyelerin bölgelere ayrıldığı ve güvenlik parametrelerinin eklendiği bir yapıya dönüşmüştür [2], [3]. Bu model, EKS ihtiyaçları, bileşenler arası bağlantılar ve bağımlılıklar göz önüne alınarak, operasyonel teknolojiler (OT) altyapılarının birbirinden farklı bölgelerden oluştuğunu ve bölgeler arası ayrım ve iletişimin nasıl olması gerektiğini tanımlar.

(a)

(b)

Şekil 1.1. (a) Purdue referans modeli (b) Güvenli ağ mimarisi

Şekil 1.1. (a)'da yer alan kurumsal bölge içerisinde, BT sistem ve uygulamaları ile VPN ve internet erişimlerinin yer aldığı seviye 5 ve raporlama, planlama, lojistik, operasyonel işler ile varlık yönetimi ve bakım çalışmalarının yürütüldüğü seviye 4 yer almaktadır.

Üretim bölgesinde son ürünün elde edilmesindeki üretim kontrol yönetimi yer almaktadır. Bu bölgede raporlama ve planlama sistemleri, mühendislik istasyonları, ağ dosya sunucuları, DNS, DHCP, Active Directory hizmetleri, üretim yedeklemesi gibi uygulama ve servisler sunulmaktadır [2]. Hücre/saha bölgesinde ise algılayıcı (sensör) ve eyleyicilerin (aktuatör) bulunduğu seviye 0, bu elemanların sürekli, yığın ve ayrık olarak kontrol edildiği PLC, RTU, DCS gibi aygıtların bulunduğu seviye 1 ve HMI, alarm/uyarı ve kontrol odasında yer alan iş istasyonların bulunduğu seviye 2 yer almaktadır.

Güvenlik bölgesindeki sistemler anomali tespiti için izleme ve belirli bir eşik değer üzerine çıkan durumlarda otomatik uyarı sistemlerini barındırmaktadır. Bu sistemler genellikle diğer kontrol sistem elemanlarından ayrı bir ortamda bulunmaktadır.

Purdue modelde kurum içindeki servis ve hizmetlere yapılacak uzaktan erişim ve internet erişimleri seviye 5 üzerinden yapılması önerilse de EKS ortamında bu şekildeki direkt bir iletişim risk faktörü oluşturmaktadır. Ayrıca, üretim ve kurumsal bölgelerin birbirleri ile doğrudan iletişimi de güvenlik nedeniyle önerilmemektedir. Bu nedenle, modele Arındırılmış Bölge (DMZ) olarak tanımlanan bir tampon katman eklenerek erişimler bu katman üzerinden yapılması önerilmiştir (Şekil 1.1.(b)). DMZ bölgesi hem seviye 1 içindeki aygıtların üretim ve kurumsal bölge ile iletişimini hem de kurumsal bölgedeki servislere iletişim ihtiyacını karşılamaktadır.

Yine günümüzde EKS'de sıkça kullanılan derinlemesine güvenlik kavramı her katman için ilgili katmanın gerekliliklerine, kabullerine ve sahip olduğu varlıklara göre önlem alınıp riskleri aza indirgemenin gerekliliğini tanımlamaktadır. Bu nedenle, değiştirilmiş Purdue modelinde, bölgeler arası iletişim ve bölgeye dışarıdan erişimler için her seviyeye bölge erişim noktası olarak güvenlik duvarı yerleştirilmesi de NIST

tarafından önerilmiştir [4]. Seviye 4, 5'e yapılacak erişimlerin kontrolünde, içinde kullanıcı veritabanının yer aldığı ve güvenlik duvarına doğrudan bağlı bir sunucu olan veritabanı bölgesi de önerilmiştir. Bunun yanı sıra, seviye 4 ve 3 içine güvenlik duvarı ile ayrılmış, içinde SIEM yazılımları ve günlükleme (log) toplayıcıların yer aldığı izleme bölgesi eklenmesi de öneriler içindedir.

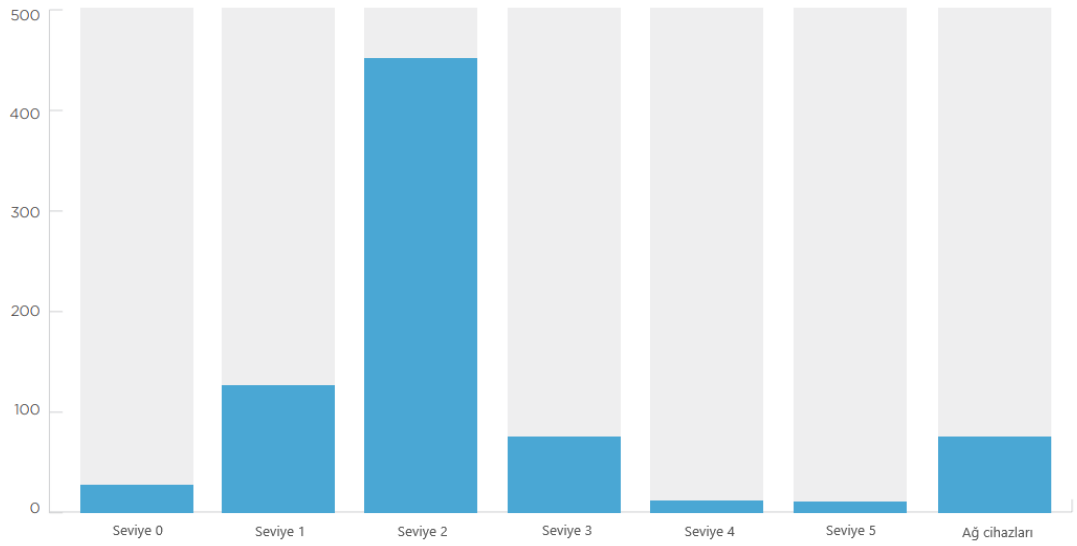
Bunlara ek olarak, son kullanıcı tarafında anti-virüs ve anti-zararlı yazılım araçlarının yerleştirilmesi ile sıkılaştırma, ağın konfigürasyonlarının düzenlenmesi ile sıkılaştırma, fiziksel güvenlik, politikaların iyileştirilmesi gibi pratiklerin uygulanması da gerekmektedir.

EKS üzerinde erişim kontrolü, ağ güvenliği, günlükleme yönetimi ve uzaktan erişimler için önerilen bölgelere ayırma, bölgelerin IDS/IPS ve güvenlik duvarı ile entegrasyonları, izleme sistemlerinin yerleştirilmesi ile tümüyle güvenli bir ağ mimarisi oluşmasını sağlamaktadır. Böylece yetkilendirme, kimlik doğrulama ve izleme (AAA) olarak bilinen güvenlik gereksinimleri de karşılanmış olmaktadır. BT ortamlarında gizlilik, bütünlük ve erişilebilirlik (CIA) gibi 3 güvenlik parametresi olması gerekirken, operasyonel teknolojilerde (OT) bunlara güvenlik parametresi de eklenmelidir [5]. Bu iyileştirmelerle, eskiden soyutlanmış bir yapı halinde olan EKS'lerin, günümüzde OT ile BT'nin uyumlaştırılması sürecinde karşılaşacakları olası siber tehditlere karşı daha gürbüz bir yapıya dönüştürülmesi hedeflenmektedir.

Bu kapsamda piyasada ticari olarak sunulmuş farklı güvenlik çözümleri mevcuttur. Mevcut güvenlik duvarı çözümlerinden bazıları; 3eTI CyberFence Family, WurldTech OpShield, Tofino Xenon, Phoenix Contact mGuard Series, EWon, Moxa EDR Series, Fortinet, Belden- Hirschmann, data diyot çözümleri; Waterfall Security, Fox-IT, Nexor, Vado, IDS/IPS çözümleri; SecurityMatters, Claroty, Darktrace, Dragos, Indegy, Cyberbit olarak kullanılmaktadır.

2009 yılında SCADA sistemleri içerisinde bulunan donanım ve yazılım elemanları maliyeti 40.9 milyar dolardır. Bu elemanlar eyleyiciler, algılayıcılar yoğunlukta olmak üzere iletişim elemanları, kontrol elemanları ve programlama ünitelerinden meydana

gelmektedir [6]. Dolayısıyla otomasyon sistemleri, eleman sayısının oldukça fazla olması ve sistemlerin doğası gereği karmaşıktır. Diğer bir taraftan ise tüm bu elemanların kullanıldıkları sistem içerisinde güvenli bir şekilde iletişim ve veri aktarımının sağlanması da gereklidir. Önerilen güvenli ağ mimarileri ve IDS/IPS/güvenlik duvarı/data diyot çözümleri her ne kadar EKS güvenliğine katkı sağlasa da saldırıların da paralel olarak gelişmesi ve etkileşimli OT ağlarının artışı kritik sistemler üzerindeki risklerin artmasına neden olmuştur. 2016 yılındaki bir çalışmaya göre en fazla zafiyet bulunduran bölgeler sırasıyla hücre/saha bölgesindeki seviye 2, seviye 1 ve DMZ bölgesindeki bileşenler olarak belirtilmiştir [7]. Bu tür saldırıların ortak amacı çalışan prosesi izlemek/kontrol altına almak, PLC'leri kötüye kullanmak, yazılım ve ağ elemanlarının bütünlüğünü bozmaktır. Bu amaca yönelik gerçekleşen saldırıların birçoğu iletişim altyapısı üzerinden, protokollerin sömürülmesiyle gerçekleştirilmekte olup, nihai sonuçlar ise algılayıcı ve eyleyici üzerinden olmaktadır.



Şekil 1.2. EKS tabanlı zafiyetlerin seviye etki dağılımı [7]

Şekil 1.2.'de, EKS'de bulunan zafiyetlerin Purdue modelde yer alan seviyeler üzerindeki etkisi görülmektedir. Buna göre zafiyetler en fazla gözetimli kontrolün sağlandığı cihazlar üzerinde, sonrasında sırasıyla PLC, RTU, DCS gibi cihazların yer aldığı seviye 1 ve üretim bölgesi üzerinde yer almaktadır. Ayrıca ağ cihazları üzerinde yer alan zafiyetlerin de oldukça fazla olduğu

görülebilmektedir. Bu da iletişim altyapısı üzerinden gelebilecek olası saldırılara açık bir kapı bırakmaktadır. Tablo 1.1.'de ise bu zafiyetleri kullanarak kritik altyapılı sistemler üzerine yapılmış bazı saldırılar ve sonuçları listelenmektedir.

Tablo 1.1. Geçmişten günümüze siber saldırılar [9]–[14]

Saldırı	Yıl	Yer	Sonuç
Trojan sızdırılması (bilinen ilk siber saldırı)	1982	Gaz Boru Hattı - Sibirya	TNT patlaması
Şirket bilgisayarlarına sızma ve alarm	1992	Chevron Acil Alarm Sistemi - California	Alarm sistemi kapatılması
Petrol boru hattı kontrolörü hatası	1992	Doğal Gaz Şirketi - Texas, Amerika	Maddi zarar ve ölümler
Dialup modem ile Salt River Projesi kapsamındaki fatura sistemine trojan bulaşması	1994	Salt River Projesi - Phoenix, Amerika	Admin yetkileri, loglar, şifreler ve çeşitli verilerin değiştirilmesi
Telefon sistemine sızılması	1997	Telefon Şirketi - MA, Amerika	Önemli hatların 6 saat servis dışı bırakılması
Trojan bulaştırılması	1999	Gaz Şirketi-Rusya	Gaz kontrolünün değiştirilmesi
SCADA sistem hatası	1999	Benzin Depolama Birimi - İngiltere	Patlama sonucu çevre kirliliği ve ölümler
Kritik sistem altyapısında konfigürasyon hatası	1999	Benzin Boru Hattı - Washington, Amerika	Boru hattı patlaması ve ölümler
Atık kontrol sistemine kablosuz radyo üzerinden girilmesi	2000	Atık Yönetim Santrali - Avustralya	Ham haldeki atıkların çevreye taşması
SQL sistemine sızılması	2003	CSX Şirketi - Florida, Amerika	Ulaşım sistemi sinyal ve alarmların kapatılması
Bilgisayar solucanının sisteme sızması	2003	Nükleer Santral - Ohio, Amerika	2 izleme sistemini servis dışı bırakması
SCADA tabanlı tren koruma sistemine sızılması	2009	Metro Tren Yolu - DC, Amerika	Tren çarpışması ve ölümler
Stuxnet virüsü	2010	Nükleer Santrali - İran	Santrifüj seviyelerinde yükselme
Backdoor yerleştirilmesi	2012	Petrol Santralleri - Orta Doğu Ve Kuzey Afrika	Bilgi sızdırılması, fonksiyonların değiştirilmesi
Dragonfly	2013	Abd, İspanya, Fransa, İtalya, Almanya, Türkiye, Polonya	E-posta üzerinden kimlik avı, Watering Hole ile web siteler üzerinden zararlı bulaştırılması, SCADA yazılımlarının trojan gibi kullanılması
Havex/Energetic Bear RAT	2014	Avrupa/ABD	Hidroelektrik barajların kapatılması, güç üretim santrallerinin aşırı yüklenmesi
Blackenergy 2/3	2014-2015	Ukrayna TV Ve Enerji Sektörü	HMI yazılımlarının servis dışı bırakılması, şebekelerin ana istasyonla bağlantısının kesilmesi
Crashoverride	2016	Ukrayna	Şebeke enerjisinin kesilmesi
Ukrayna siber saldırısı (Petya zararlı yazılımı) (Wannacry benzeri)	2017	Ukrayna	Havaalanındaki panoların, metro bilet otomatlarının devre dışı bırakılması, enerji santrali SCADA bağlantısının çalışmaması, bazı banka hizmetlerinin verilememesi
Triton/Trisis/Hatman	2017	Orta Doğu	Acil kapatma/güvenlik enstrümanlı sistemlerin devre dışı bırakılmasının hedeflenmesi

Saldırıların birçoğu programlama üniteleri üzerinden kodların değiştirilmesi, kontrol elemanlarının işlevsiz hale getirilmesi, ağ verilerinin dinlenmesi, sızdırılması, manipüle edilerek işleyişin değiştirilmesi veya servis durdurucu saldırılar olarak

karşımıza çıkmaktadır. Bunlardan bazıları BT' de bilinen ve başarı elde edilen, OT ve BT uyumlaştırılması neticesinde ise EKS ortamlarına taşınan saldırılardır. Örneğin BT sistemlerde sıkça yapılan servis durdurma (DoS), dağıtık servis durdurma (DDoS), yeniden oynatma, şifre kırılmasına yönelik kaba kuvvet saldırıları ve fragmantasyon atakları kritik sistemlerde de oldukça etkilidir. Bunlara ek olarak EKS'ye özgün kriptografik ataklar, PLC RAM belleği üzerinde çalışan programın bulunduğu bellek ile program durumunu veya değişkenlerin tutulduğu kaydedici belleği hedef alan saldırılar da literatürde mevcuttur [6], [8].

Kritik altyapılı sistemlere yönelik yapılan saldırıların başlıca sebebi zayıflıklardır. Örneğin, sisteme giriş izni sağlanması için SCADA sistemlerde şifreler oldukça güvensiz verildikleri, bazı yapılarda ise varsayılan şifrelerin dahi değiştirilmemesi nedeniyle yönetici yetkilerine erişmek çok güç olmamaktadır. Ek olarak, sistem zafiyetlerinin kapatılmasına yönelik yamaların düzenli yapılmaması, güvenlik duvarı desteği alınmaması, anti-virüs gibi 3. parti yazılımların kullanılmaması, fiziksel güvenliğe önem verilmemesi, sıfırıncı-gün (zero-day exploit) zafiyetleri ve günlükleme/izleme tekniklerinin zayıf olması gibi faktörler bu sistemler üzerine yapılan saldırıları daha da cazip hale getirmiştir. Saldırıların artışı, bir diğer alan olan savunma sistemleri üzerindeki çalışmaları hem akademik hem de ticari boyutta tetiklemiştir. Bu kapsamda, Marquez ve ark. günlükleme/izleme sistemlerinin öneminden yola çıkarak rüzgâr tribünleri takip sistemlerini araştırmışlardır [9]. Yine 2014'te Sclechtingen ve Santos rüzgâr tribünleri üzerine durum izleme sistemi geliştirmiştir. 18 ayrı tribününden gelen sinyaller 35 ay süresince toplanıp veri madenciliği/bulanık mantık teknikleri ile geliştirilen bir metotla test edilmiştir [10]. Bu çalışmalar enerji sektörüne özelleşmiş çözümler olup protokol tabanlı değillerdir. McQueen ve ark. küçük ölçekli SCADA sistemlerde siber risk azaltma metodu geliştirmişlerdir. Çalışmada graf metodu ve yaklaşım algoritmaları ile yönlendirilmiş graf tasarlanmıştır. Bu yaklaşım, düğümlerin saldırı aşamasını, kenarların ise saldırıyla karşılaşma zamanını gösterildiği bir metottur [11]. Çalışmada saldırı tespiti graf üzerinden tanımlanmakta olup karmaşık kritik altyapılarda tespit zorlaşmaktadır. Yakkali ve Subramanian kritik sistemlerin tasarımı da zafiyet barındırmada önem arz ettiğinden minimum karar ağaçları kullanarak verimli sistem tasarımını gerçekleştiren

bir altyapı geliştirmişlerdir [12]. Bu yapı, sistemin amacına göre ağırlıklandırılmış yapı içerisinde en uygun olanları seçerek işlem yapan bir tasarımdır. Chopade ve Bikdash, yine graf modelleri kullanarak elektrik iletim ve dağıtım grid sistemleri üzerinde çalışmış, topoloji özelliklerini tespit etmiş, saldırı ve hata tolere yüzdelerini hesaplamışlardır [13]. Bu çalışma ise elektrik iletim/dağıtım sektörüne özel bir çalışma olup protokol bilgisi barındırmamaktadır.

SCADA sistemler üzerine yapılan saldırıları önlemek de önem arz etmektedir. Bu amaçla IDS, IPS, çeşitli kimlik doğrulama yöntemleri, zararlı yazılımdan korunma amacıyla 3. Parti yazılımlar, TLS veya IPSec kullanımı, VPN kullanımı gibi muhtemel atakları asgariye indirmeye yönelik çözümler kullanılmaktadır [14]. Çözümler kapsamında Wei ve ark. kritik altyapılı sistemleri siber saldırılardan korumak amacıyla elektrik güç sistemlerine entegre katmanlı güvenlik altyapısı önermişlerdir [15]. Bu yapıda gereksinim, verimlilik, birliktelik, performans, modülerlik, ölçeklenebilirlik ve yönetilebilirlik gibi parametreler bulunmakta olup, bu öneri ancak yeni bir sistemin kurulumunda uygulanabilmektedir. Benzer doğrultuda Yang ve ark. 2013'te DNP3 protokolü tabanlı çalışan saldırı tespit sistemi geliştirmişlerdir [16]. Sistem kural tabanlı olup, bilinen ataklar için imza bilinmeyen ataklar için ise model tabanlı saldırı tespit etmektedir. Önerilen yapı ise Snort üzerinde uygulanarak gösterilmiştir. Çalışma DNP3 protokolü kabulleriyle geliştirilmiş olup Snort üzerinde hâlihazırda yer alan DNP3 önişlemcisi kullanılmıştır. Benzer şekilde Sayegh ve ark. SCADA sistemler üzerinde çalışan genel amaçlı bir saldırı tespit sistemi önermişlerdir [17]. Bu önerinin diğer çalışmalardan farkı öğrenme algoritmaları ile ağ trafiği davranışlarını analiz etmesi sıklıkla oluşan örüntüler dışına çıkılması halinde alarm üretmesidir. Yine saldırı tespit metodu öneren bir diğer araştırmacılar Buckner, Beaver ve Borges-Hink'tir. 2013 yılında Naive Bayes, Rastsal Forests, J48, NNge, OneR, SVM gibi makine öğrenmesi yöntemlerini kullanarak Uzak Terminal Birimleri (RTU) üzerine yapılan komut işleme ve veri enjeksiyonu gibi saldırıları tespit etmeye çalışmışlardır [18]. Linda ve ark. ise yapay sinir ağlarından faydalanarak 2009 yılında hata geri yayılım algoritması ve Levenberg-Marquardt algoritmaları kullanılarak belirlenmiş bir pencere içindeki verileri inceleyerek, rastsal şekilde üretilmiş saldırıları tespit etmeyi hedeflemişlerdir

[19]. Bu iki çalışma ise loglar üzerinden yapılmış olup, sadece bilinen saldırıları tespit edebilmektedir.

Saldırıları tespit ve önleme yolunun yanı sıra, önceden saldırı ihtimallerini gözeterek sistemin bunlara olan tepkisini ölçmek ve sistemde var olan fakat bilinmeyen zafiyetleri tespit etmek de dikkat edilmesi gereken diğer bir husustur. Bu sayede kritik altyapılı sistem içerisinde bulunan topoloji, protokol, donanım veya kullanılan yazılımlardan kaynaklı açıklıkları kapatmak mümkün olabilir. Kritik sistemlerde haberleşmenin sağlandığı paket alışverişi, protokol standartlarında tanımlandığı şekilde yapıldığı için, bir protokolün düz metin iletişim gerçekleştirmesi, yetkilendirme, kimlik doğrulama, güvenli bağlantı oluşturma gibi herhangi bir güvenlik mekanizmasının bulunmamasından kaynaklanan zafiyetlerin, protokolü kullanan altyapılarda var olduğu kabul edilebilir. Diğer bir değişle SCADA sistemlerinde kullanılan MODBUS TCP, PROFINET, EtherCAT, DNP3 gibi protokollerde yer alan açıklıklar bu standartları kullanan yapılarda da mevcut olduğu için sistemleri tehdit etmektedir. Bu kapsamda, Zhu ve ark. 2011 yılında yaptıkları çalışmada protokollerin çalışma prensipleri ve protokoller üzerine, ağ, transfer, uygulama katmanı bazlı açıklıkları incelemişlerdir [20]. Benzer şekilde Xiaosheng ve ark. 2012'de EtherCAT protokolünü, paket yapısını incelemiş, akıllı istasyonların, efendi ve köle yapılarının özelliklerini dikkate alarak, endüstriyel EtherCAT tabanlı bir sistemin en efektif şekilde nasıl dizayn edilmesi gerektiğini incelemiştir [21]. Yine 2012'de Li ve ark. EtherCAT tabanlı bir otomasyon sisteminde çalışacak veri toplama sistemi önermiştir [22]. Önerme, köle istasyonların direk bellek modunda çalışırken, RAM kontrolörlerden beklemeden okuma yapması üzerine kurulmuştur. Bu sayede, gerçek zamanlılığın büyük veriler üzerinde daha verimli sürdürülmesine olanak sağlanmıştır. Bu çalışmalar bir güvenlik çözüm önerisi olmayıp mevcut sistemleri iyileştirmeye yönelik yaklaşımlardır.

Diğer bir yandan, EKS ağlarında saha ve bazı fabrika seviyesindeki veriler efendi ve köleler arasında genellikle otomatik olarak sürekli değişmektedir. Veriler periyodik yoklama programları veya önceden tanımlı görevler yoluyla çekilmekte olup, algılayıcı/eyleyici verisi, PLC durum bilgisi, yazma komutları veya önceden tanımlı

keskin zaman aralıklarında işleyen dağıtık saat verisi olabilmektedir. Bu süreç tekrarı veya döngüsel iletişimden ötürü saha iletişimi güçlü periyodik kalıplar barındırmaktadır. Güvenlik açısından bakıldığında, periyodikliği öncelikle eyleyici/algılayıcı, saha ve hücre düzeyinde, veri aktarımlarının gerçekleştirildiği cihazlar üzerinden yapılan iletişimlerde belirlemek, anomalileri tespit etmek açısından önemlidir. Bu şekilde çevrimli bir davranışa sahip sistemde anomali tespiti için senkronizasyon periyodu ve trafik örüntüsünün bilinmesi gerekir.

EKS dünyasında iletişimde yaygın olarak kullanılan EtherCAT protokolü CIM mimarisindeki yönetim, hücre, saha ve algılayıcı/eyleyici seviyelerinin tümünü desteklerken, Purdue referans modelinde yer alan seviye 0-5 arasındaki tüm iletişim ihtiyaçlarını tek başına karşılayabilmektedir. Gerek ürün yelpazesinin geniş olması gerekse hızlı iletişim gerçekleştirmesi nedeniyle özellikle Avrupa'da otomasyon sektörünün enerji, makine, bina gibi birçok alanında kullanılmaktadır. Protokolün Ethernet tabanlı olması EKS'lerin dış dünyaya açılmalarına, TCP/IP ile entegrasyonuna ve bilgi teknolojilerinde sunulan web, ftp, mail gibi birçok servisten faydalanmalarına olanak sağlamıştır. Fakat bu entegrasyon, sistemleri Ethernet üzerinden yapılabilen saldırılara da açık hale getirmiştir. Ayrıca bu protokolde diğer birçok EKS protokolü gibi kimlik doğrulama, şifreleme ve yetkilendirme özellikleri yer almamakta, veriler düz metin şeklinde iletilmekte ve saha seviyesinde herhangi bir güvenlik adımından geçirilmemektedir. Bu nedenle EtherCAT protokolü, hem TCP/IP hem EKS tabanlı hem de doğasından kaynaklı saldırılara maruz kalmakta, dolayısıyla olası siber tehditlere karşı bir çözüme ihtiyaç duymaktadır. Diğer taraftan EtherCAT protokolü, diğer EKS protokolleri gibi saha iletişiminde güçlü periyodik tekrarlar içermektedir ve yapılan herhangi bir saldırı bu yapının bozulmasına neden olmaktadır. Ayrıca, DeviceNet protokolünde olan EDS, PROFIBUS'ta olan GSD konfigürasyon dosyaları, EtherCAT ortamında da ENI adıyla yer almakta olup, efendi, köle ve tüm iletişim kabulleri bu yapıda tutulmaktadır. Bu konfigürasyon dosyasında yer almayan herhangi bir iletişim saldırı olarak tanımlanabilir.

EtherCAT protokolünün doğasından kaynaklı güvenlik parametrelerinin bulunmaması, Ethernet tabanlı olmasının risk faktörü oluşturması, otomasyon

uygulamalarında yaygın bir kullanıma sahip olması ve kullanım hızının yüksek bir ivme ile artmasıyla dış tehditlere daha çok açık olmasından dolayı, tez kapsamında EtherCAT tabanlı çözüm önerileri sunulmuş ve protokol güvenliğine katkı sağlanması hedeflenmiştir. Bu hedef doğrultusunda, bulanıklaştırıcı ve saldırı vektörü geliştirilmesi yoluyla zafiyet analizinin ardından açıklıkların önlenmesi amacıyla önişlemci yapısının geliştirilmesi, sıfırıncı-gün saldırılarını önlemek amacıyla periyot tespiti ve makine öğrenmesi yöntemleriyle anomali tespiti yapan tümleşik bir yapının geliştirilmesi amaçlanmıştır. Bu kapsamda tez 4 aşamalı olarak yürütülmüştür. İlk aşamada, protokol içerisinde yer alan zafiyet ve zayıflıkların tespiti ile alakalı çalışmalar yapılmıştır. Burada bulanıklaştırma ve saldırı vektörü geliştirilmesi yöntemleri kullanılmıştır. Sistemlerde tehdit unsuru taşıyan bu zafiyetlerin ortaya çıkarılmasından sonra EtherCAT tabanlı sistemlerin güvenliğini arttırmak ve oluşabilecek saldırıların önüne geçebilmek adına ilgili açıklıklara çözümler geliştirilmiştir. Bu kapsamda, EtherCAT protokolünün kullanıldığı Merkezi Denetleme Kontrol ve Veri Toplama (SCADA) sistemlerindeki siber tehditlerin/istismarların engellenmesi için, saldırı önleme amaçlı Snort açık kaynak kodlu saldırı tespit ve önleme sistemi (IDS/IPS) üzerine EtherCAT önişlemcisi (preprocessor) geliştirilmiştir. Bu çözüm içerisinde ENI dosyaları kullanılarak geliştirilmiş güvenli düğüm yaklaşımı önerisi de yer almaktadır. Bu hedefi takiben bilinmeyen saldırıların tespiti ve önlenmesi amacıyla periyot tabanlı anomali tespiti çalışması yapılmıştır. Son aşamada ise oluşturulan bir test ortamından elde edilen veriseti üzerinden makine öğrenmesi yöntemleri kullanılarak anomali tespiti yapılmış ve performans değerleri ölçülmüştür. Tez neticesinde, EtherCAT altyapılı sistemlere uygulanabilen hem bilinen/tespiti yapılan açıklıklar için hem de sıfırıncı-gün saldırılarının önüne geçmek için EtherCAT protokolü her açıdan değerlendirilip, çözüm önerileri geliştirilmiştir.

Bir sonraki bölümde, endüstriyel iletişim sistemleri genel bir bakış açısı ile aktarılmakta ve literatür çalışması sunulmaktadır. Literatür çalışması tezin aşamalarına uygun olarak gruplandırılmış olup önişlemciler ve anomali tespit çalışmalarından oluşan 2 alt bölümde aşağıdaki gibi aktarılmaktadır.

1.2. Çalışmanın Amacı ve Önerilen Çözüm Yöntemi

Tez kapsamında, kritik altyapılı sistemlerde kullanılan EtherCAT protokolü tabanlı bir SCADA sisteminde bilinen ve bilinmeyen saldırıların tespit edilmesi ve muhtemel açıklıkların istismar edilmesini önlemek amacıyla ikisi kural biri anomali tabanlı olan özgün çözümlerin uygulanması için,

a. Her aşamada gerçek donanımların kullanıldığı ilgili test ortamlarının oluşturulması
b. Otomasyon sektöründe yaygın olarak kullanılan EtherCAT protokolünün saha iletişiminde kullanılan protokollerin analizi ve modellenmesi,
c. İlgili protokol ve alt protokollerin hem standartları çerçevesinde ve hem de standartları dışında paketler üretilmesi sonucu oluşabilecek servis engelleme, yavaşlatma, köle istasyonlara saldırı, veri enjeksiyonu gibi zafiyetlerin araştırılması ve ilgili saldırı vektörlerinin bulanıklaştırma veya geliştirilen program üzerinden gerçeklenmesi,
d. Elde edilen veriler ve analizler ışığında, kritik etkili saldırıların önlenmesi amacıyla bir önişlemci ve önişlemci üzerine entegre edilecek farklı yaklaşımlarla gerçek zamanlı istismarların önüne geçilmesi ile tümleşik bir sistem oluşturulması,
e. Makine öğrenmesi tekniklerinin saldırıların tespiti üzerindeki etkisinin oluşturulan bir veriseti üzerinden araştırılması ile bilgi güvenliğinin artırılmasına katkı sağlanması,
f. Protokol tabanlı açıklıkların engellemesiyle SCADA sistemler üzerinde zafiyetlerin ne derece önlendiğinin tespit edilmesine katkı sağlanması ve Purdue referans modelinde tanımlanan hücre/saha bölgesindeki seviye 0 - 1 ve 2 üzerinde risk azaltılmasının gerçeklenmesi amaçlanmaktadır. Tez neticesinde ülkemizde yaygın kullanım potansiyeli olan bir altyapıya güvenlik çözümü geliştirilmesi, benzer çözümlerde bulunmayan saldırı önleme, derin paket analizi, sıfırıncı-gün tehditleri gibi özellikleri de desteklemesiyle ülkemizdeki SCADA altyapılı sistemlerin güvenliğinin artırılmasına katkı sağlanması sonucu elde edilecektir.

1.3. Tez Organizasyonu

Yapılan çalışmanın sunulduğu tez, aşağıdaki biçimde sunulmaktadır.

Bölüm 1: Giriş- Bu bölümde problemin tanımı, çalışmanın amacı ve tez organizasyonu hakkında bilgi sunulmaktadır.

Bölüm 2: Endüstriyel İletişim Sistemleri- Bu bölümde endüstriyel iletişim sistemlerinin yapısı, güncel durumu ve EtherCAT protokolü genel bir bakış açısı ile sunulmaktadır.

Bölüm 3: EtherCAT protokolünde zafiyet tespiti- Bu bölümde, bu ve sonraki iki bölüm kapsamında yapılan çalışmalarda kullanılan test ortamı, bulanıklaştırıcı yaklaşımı, gri-kutu bulanıklaştırıcısı geliştirilmesi ve saldırı vektörleri hakkında bilgiler sunulmaktadır.

Bölüm 4: EtherCAT Önişlemcisi Geliştirilmesi ve Güvenli Düğüm Yaklaşımı- Bu bölümde Snort yapısı, geliştirilen çözümleyici ve önişlemci yapıları hakkında bilgiler sunulmaktadır. Bölüm sonunda sonuçlar incelenmektedir.

Bölüm 5: EtherCAT Tabanlı Sistemlerde Saha Seviyesindeki Anomalilerin Periyot Tespiti ile Bulunması- Çalışmanın otokorelasyon fonksiyonunun senkronlu iletişimde kullanılabilirliği ve periyot analizi aşamaları hakkında bilgiler verilmektedir. Ayrıca bölüm sonunda anomali tespiti sonuçları da sunulmaktadır.

Bölüm 6: ECAT Tabanlı Anomali Tespitinde Makine Öğrenmesi Yöntemleri- Bu bölümde, kullanılan test ortamı, geliştirilen PLC programı aşamaları hakkında bilgi verilmektedir. Ayrıca veriseti oluşturmada kullanılan saldırılar ve makine öğrenmesi yöntemlerinin bu verisetinde uygulanması ile çıkan sonuçlar irdelenerek aktarılmaktadır.

Bölüm 7: Sonuçlar: Bu bölümde, yapılan çalışmalardan elde edilen sonuçlar sıralanmış ve bunlar üzerine değerlendirmelere yer verilerek yapılan çalışmanın gerek bilime gerekse endüstriye getireceği katkılar tartışılmıştır.

BÖLÜM 2. ENDÜSTRİYEL İLETİŞİM SİSTEMLERİ ve İLİŞKİLİ ÇALIŞMALAR

Endüstriyel kontrol sistemleri (EKS), fiziksel cihazları, süreçleri veya olayları doğrudan izleyerek, yöneterek bir değişikliği saptayan veya gerçekleştiren donanım ve yazılım bütününden oluşmaktadır. EKS'ler doğalgaz/petrol boru hatları, enerji sektörü, bina endüstrisi, rüzgâr enerji sistemleri, solar sistemler, su toplama, dağıtım ve arıtım sistemleri, kimya ve otomotiv sektörü gibi süreç ve sürekliliğin olduğu birçok kritik altyapıda kullanılmaktadır. Bu otomasyon sistemleri ise planlama, yürütme, hücre, saha ve algılayıcı-eyleyici seviyelerine sahip bir hiyerarşi içerisinde projelendirilirken, sistemlerin kontrolü, sürekliliğinin sağlanması ve izlenmesi SCADA sistemleri tarafından yürütülür.

EKS'de saha seviyesindeki iletişim önceleri PROFIBUS, Interbus, DeviceNet, ControlNet ve diğer saha veriyolu protokolleri ile sağlanırken, günümüzde seviyeler arası yatay ve dikey entegrasyonun tek bir protokolle kolay bir şekilde sağlanabilmesi fikrinden yola çıkarak Modbus/TCP, EtherNet/IP, PROFINET, Ethernet Powerlink, Sercos III ve EtherCAT gibi Ethernet tabanlı endüstriyel iletişim protokolleri üzerinden sağlanmaktadır. Bu protokoller farklı teknoloji grupları veya firmalar tarafından yönetilmektedir. En sık kullanılan protokoller, Tablo 2.1.'de gösterilmiştir. Petrol ve gaz dağıtım sistemleri, enerji sektörü, hidroelektrik santralleri, otomotiv sektörü, vb. tüm kritik altyapılı sistemlerin kontrolü ve izlenmesi için kullanılan SCADA yazılımları, prosese dâhil olan tüm donanımlar arasındaki iletişimi Ethernet tabanlı bu protokoller üzerinden yapmaktadır.

Tablo 2.1. Ethernet tabanlı endüstriyel ağ protokolleri

Standart / Profil	**Marka**	**Mimari Sınıfı**	**Ethernet "Tip" değeri**
IEEE 802.3	Ethernet (10/100/ 1000/10000 Mbps)	Gerçek Zamanlı Olmayan (NRT) Ethernet	0800h IP
IEC61784 CPF-2	Ethernet/IP	TCP/IP	0800h IP
IEC61784 CPF-3/3	PROFINET-CBA	Ethernet	8892h
IEC61784 CPF-3/4	PROFINET IO	Ethernet	8892h
IEC61784 CPF-4	P-Net	TCP/IP	0800h IP
IEC61784 CPF-10	Vnet/IP	TCP/IP	0800h IP
IEC61784 CPF-11	TCnet	Ethernet	888Bh
IEC61784 CPF-12	EtherCAT	Değiştirilmiş (Modified) Ethernet	88A4h
IEC61784 CPF-13	Ethernet Powerlink	Ethernet	88ABh
IEC61784 CPF-14	EPA	Ethernet	88BCh
IEC61784 CPF-15	MODBUS/TCP	TCP/IP	0800h IP
IEC61784 CPF-16	SERCOS III	Değiştirilmiş (Modified) Ethernet	88CDh

Ethernet ve TCP/IP tabanlı protokollerin yaygın olarak kullanılması, bunlara yönelik yapılan saldırıların çeşitliliğinin artışına ve saldırıların pozitif yöndeki sonuçlarının literatürde sıkça ele alınmasına neden olmuştur. Ethernet tabanlı bir protokolün TCP/IP yapısı ile kolay entegre olabilmesinden ve buna bağlı olarak da bilgi teknolojileri (BT) sektöründe sunulan hizmetlerin (web, ftp, mail, vpn, vb.) SCADA içerisine dâhil edilmek istenmesinden dolayı ise olası bir siber saldırı sonucunda yüksek hasar potansiyeline sahip olan SCADA sistemleri ve konumlandırıldıkları endüstrilerin, ciddi bir siber tehdit altında kalmasına neden olmuştur [11], [12], [16]. Endüstriyel otomasyon uygulamalarına çözüm sunan Ethernet tabanlı protokoller de

belirtilen kapsam doğrultusunda aynı istismarlara maruz kalabilmekte ve dolayısıyla bu altyapıyı kullanan sistemlerdeki siber tehditlerin azaltılması da ayrı bir önem arz etmektedir [23]. 2015 yılındaki bir çalışma SCADA saldırılarının %65'ten fazlasının iletişim altyapısı üzerinden gerçekleştiğini belirtmiştir [24]. Bununla birlikte, son on yıl içinde iletişim altyapısı üzerinden Iran nükleer santrali, ABD metro çarpışma önleme sistemi, Orta Doğu ve Kuzey Afrika petrol santrali ve Ukrayna güç dağıtım şebekeleri üzerine yapılan ve küresel çapta ses getiren saldırılar, iletişim altyapılarının önemini göstermektedir [25]–[31]. Saldırılarda yaygın olarak kullanılan yöntemler ortadaki adam (MITM), DoS, DDoS, yeniden oynatma, tampon taşması veya kriptografik yöntemlerdir. Saldırıların olumlu neticelenmesinin ve çeşitliğinin fazla olmasının arkasında yatan en önemli etken EKS protokollerinde kimlik doğrulama, şifreleme ve yetkilendirme gibi temel güvenlik kritiklerinin bulunmamasıdır. Oluşan bu siber riskleri aza indirgemek için protokol ve standartların doğru uygulanması, uygulamalardan kaynaklı zafiyetlerin tanımlanması ve minimize edilmesi gereklidir. EKS dünyasında diğer risk faktörleri ise güvenlik ekiplerinin ve yöneticilerin sistem varlıkları hakkında yeterli bilgi sahibi olmamaları, yamaların ve güncellemelerin zamanında yapılmaması, güncel açıklıkların takibi ve gerekli iyileştirmelerin önemsenmemesi gibi nedenlerdir. 2016 yılındaki bir zafiyet raporuna göre 2014 yılından 2015 yılına kadar açıklanan zafiyetlerin hızlı bir ivmeyle artış gösterdiği ve bu zafiyetlerden 516 tanesinin henüz sömürülmediği için yamasının bulunmadığı belirtilmiştir [7]. Bu da incelenen 1552 zafiyetten %33'ünün sıfırıncı-gün saldırısı olduğunu göstermektedir. EKS'nin güncel durumu ve güvenlik yönetimleri hakkındaki daha detaylı bilgiler Cheminod ve ark. araştırmasında yer almaktadır [32].

Otomasyon sektörüne hizmet eden firmalar EKS üzerindeki tehditleri endüstriyel güvenlik duvarları ve saldırı önleyici sistemler kullanarak engellemeye çalışmaktadırlar [16], [33], [34]. Otomasyon endüstrisi ilgili protokollere ait saldırı önleyici sistemlerin geliştirilmesi için bulanıklaştırıcı (fuzzer) yaklaşımı kullanmaktadır ve bu yaklaşımla Modbus/TCP, DNP3, EtherNET/IP ve Profinet protokolleri için bulanıklaştırıcı uygulamaları ve ürünleri geliştirilmiştir [35], [36]. Fakat hem uluslararası hem de ulusal boyutta yüksek bir kullanım ivmesine sahip olan, ülkemizde de hareket kontrol sistemleri, makine, gıda, kimya, tekstil, savunma, ulaşım

sektörlerinde kullanılan EtherCAT protokolü için sadece Beckhoff (EDR-G903 Firewall/NAT/VPN), Moxa (EDR-810) gibi firmalarının sunduğu birkaç adet güvenlik duvarı çözümü bulunmaktadır [37]–[40]. Bu çözümler bir saldırı önleyici sistem olmadığı için EtherCAT kabullerine göre özelleşmiş bir çözüm ihtiyacını karşılamamaktadırlar.

2.1. EtherCAT Protokolü

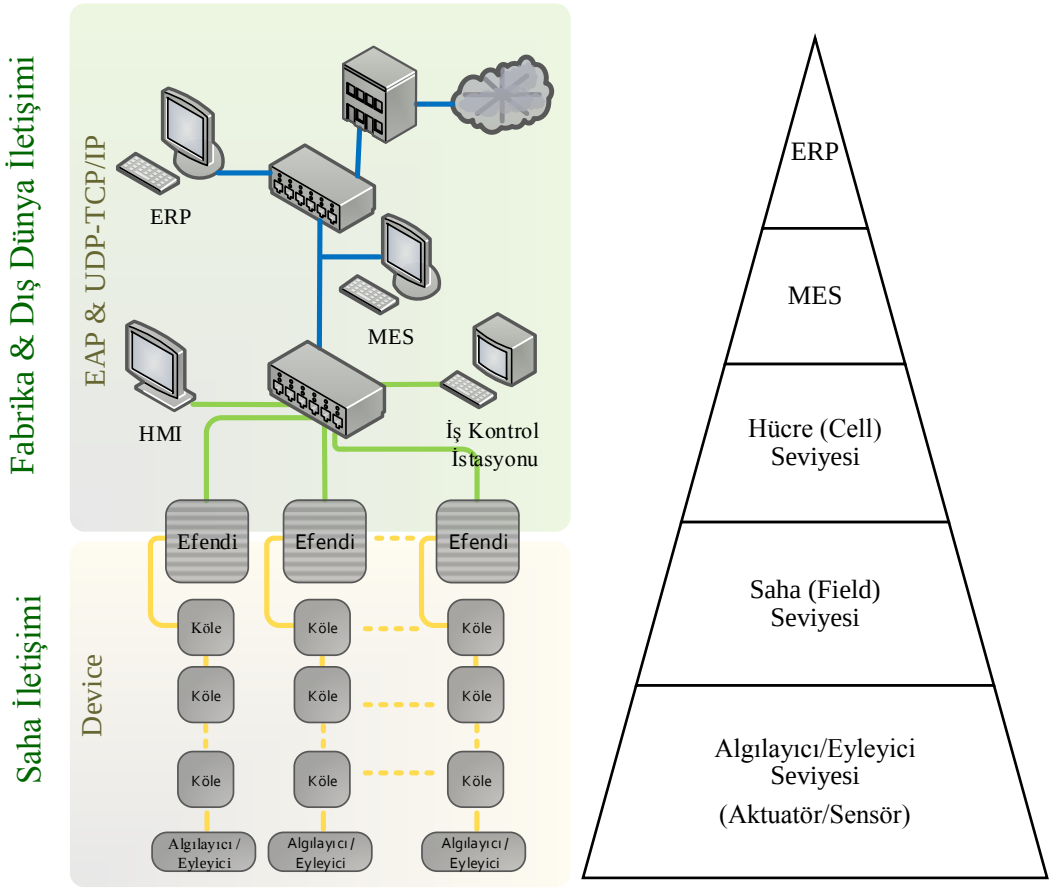
Otomasyon hiyerarşisinde saha veriyolu iletişiminde kullanılan protokoller gerçek-zamanlılık kriterlerini çevrim zamanı ve çevrim zamanı gecikmesine koydukları keskin değerlerle sağlarlar. Bu durum Ethernet tabanlı protokoller de de geçerlidir. Bu sebeple Ethernet tabanlı protokoller gerçek-zamanlılık kriterlerini sağlamak için protokol yapısında ve kullandıkları donanımlarda ciddi değişiklikler yapmaktadır. Gerçek zamanlı Ethernet tabanlı protokoller tolereli ve keskin gerçek zamanlı (soft ve hard real time) olmak üzere genel olarak 2 kategoride incelenir. Modbus/TCP tolereli gerçek zamanlı protokol iken CC-Link, PROFINET, Sercos III protokolleri keskin gerçek zamanlıdır. EtherCAT protokolü, EKS de yaygın olarak kullanılan ve gecikmelerin tolere edilmediği keskin gerçek zamanlı olarak kabul gören bir protokoldür. Kısa çevrim süresi, hızı, topoloji esnekliği, ölçeklenebilirliği, ürün çeşitliliği ve fiyat avantajı Modbus/TCP, EtherNet/IP, PROFINET RT veya Sercos III protokollerine kıyasla daha çok tercih edilmesine neden olmaktadır [41]. PROFINET gerçek zamanlı olan ve olmayan iletişimi PROFINET NRT, IRT ve RT olarak 3 yapıyla sağlarken, EtherCAT'in sadece tek yapı kullanması, kısa çevrim zamanı ile PROFINET IRT'den bile hızlı olması da (0.1ms cevap süresi ve 100Mbit/sn veri hızında 0.1ms altında dalga bozunumu (jitter)) protokolün diğer tercih sebeplerindendir [42], [43]. EtherCAT 0.1 msn cevap süresi ile Ethernet/IP, Ethernet Powerlink, PROFINET IRT ve Sercos III'den de hızlıdır [43]. Protokol kısa çevrim sürelerini havada işlem teknolojisi (on-the-fly) ile sağlar [44]. Bu teknoloji çerçevenin tamamının gelmesini beklemeden bayt bayt işlediği için benzer yöntem kullanan fakat giriş ve çıkış verilerini ayrı ayrı işleyen Sercos III'den bile hızlı olmasına sebep olmaktadır.

Protokolün geçmişi ise 2003 yılına dayanmakta olup, bu süreden beri EtherCAT organizasyonu tarafından yönetilmektedir. Protokolün kurucusu ve en büyük destekçisi Beckhoff firmasıdır. Firma kuruluşundan bu yana bilgisayar tabanlı otomasyon uygulamalarına yöneldiğinden diğer otomasyon şirketlerinden farklı kulvarda yer almaktadır. Beckhoff programlanabilir lojik kontrolör (PLC) aygıtları Windows tabanlı işletim sistemlerine sahiptir. EtherCAT'in Ethernet tabanlı oluşu, havada işlem teknolojisi, UDP/IP ile entegrasyonu ve PLC aygıtlarının Windows üzerinden yönetilmesi otomasyon endüstrisinde ciddi avantajlar elde etmesine neden olmaktadır.

2.1.1. Tüm seviyelerde EtherCAT

EtherCAT tabanlı sistemlerde PLC programlama ve konfigürasyon TwinCAT programı ile yapılmaktadır. Bu program diğer ortamlardan farklı bir özelliğe sahiptir; sistemde PLC cihazı bulunmuyorsa veya test amaçlı bir PLC gerekliyse, TwinCAT yüklenmiş mühendislik istasyonu PLC olarak iş görebilir.

EtherCAT geleneksel efendi-efendi, köle-efendi ve köle-köle arası iletişimi desteklerken, bu iletişim saha seviyesi (aygıt protokolü), fabrika seviyesi (EtherCAT otomasyon protokolü-EAP) ve IP entegrasyonu ile dış dünya seviyesinde gerçekleşebilir. Diğer SCADA sistemleriyle veya aynı yapı içinde yer alan farklı SCADA yapılarıyla iletişimde IP yönlendirmesi gerektiğinde, IEEE 802.3'te belirtilen Ethernet çerçevesine ek olarak UDP/IP üzerinden IP adresi eklenerek kullanılabilir [45]. Bu özellik sadece yönlendirme amaçlı kullanılabilmektedir. Diğer bir değişle UDP/IP desteği bulunan tüm cihazlarda kullanılamaz. Protokol bu yönüyle Şekil 2.1.'deki gibi endüstriyel sistemlerde yer alan hiyerarşik modelin tüm katmanlarına uygulanabilen bir protokol yapısına sahiptir.

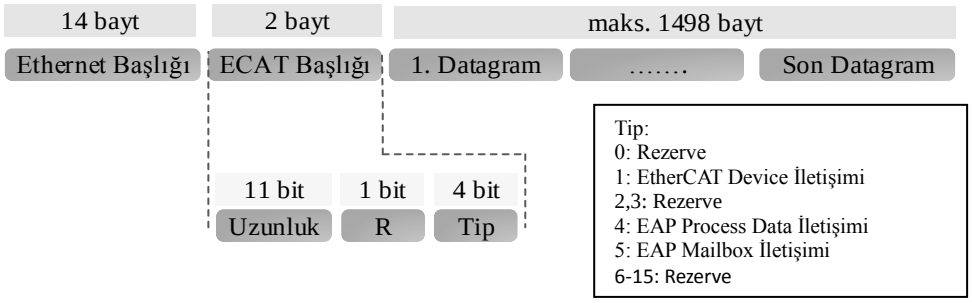


Şekil 2.1. Tüm seviyelerde EtherCAT

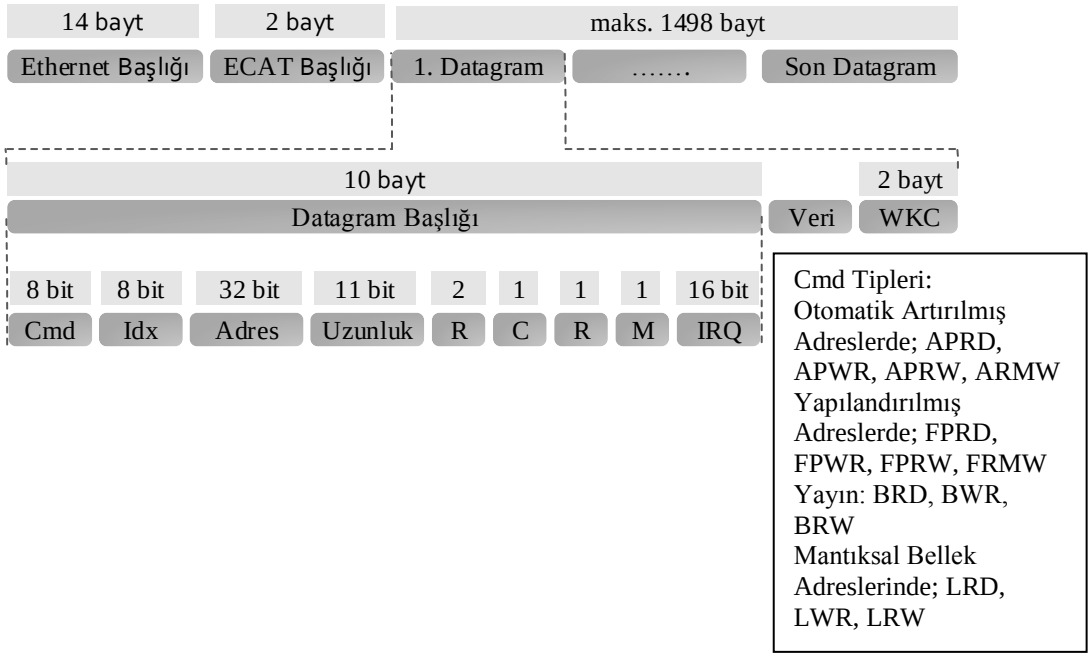
EtherCAT protokolü Ethernet tabanlı bir standart olduğu için dışta Ethernet çerçeve yapısını barındırmaktadır. EtherCAT paketleri kullanılan alt protokolden bağımsız başlık hariç toplamda maksimum 1498 bayt olmak üzere tek bir Ethernet yapısında bir veya birden fazla veri paketi taşıyabilmektedir (Şekil 2.2.). Başlık yapısı içinde bulunan tip alanı, takip eden verinin çeşidini tanımlamakta olup Şekil 2.2.'deki gibi değerler alabilmektedir.

Aygıt Protokolü, saha veya algılayıcı/eyleyici seviyesi iletişimlerini gerçekleştirmede kullanılır. Efendi köle arası iletişimde kullanılıp, kendine ait bir çerçeve yapısına sahiptir.

Şekil 2.3.'de yer alan veri paketi yapısı, EtherCAT çerçevesindeki tip alanı "1" olduğunda yani aygıt iletişiminde kullanılan çerçeveyi göstermektedir.



Şekil 2.2. EtherCAT başlığı



Şekil 2.3 Saha iletişimi paket yapısı

Görüldüğü üzere, EtherCAT ana paketinin başlık tanımından sonra gelen her EtherCAT veri paketi yine kendi içinde bir başlık yapısı barındırmaktadır. Bu sayede bir trenin vagonları gibi köle istasyonlardan geçerken her köle kendi paketini tanımaktadır. Burada en fazla 1498 bayt olabilen EtherCAT paketinin sondaki 2 baytlık çalışma sayacı (working counter -WKC) alanı, varılan her istasyonla etkileşime

göre yazma/okuma erişimi için 1, okuma ve yazma için 3 olarak artırılan değerden oluşur [46].

Veri paketini takip eden daha fazla paket olup olmadığı ise başlık içindeki "M" bitine 1 değeri atanmasıyla anlaşılır. Yine başlık içinde her istasyonun paketini tanıyabilmesi için belirli bir örüntüyle adlandırılan 32 bitlik adres yapısı mevcuttur. Bu alandaki köle adresleri belirli bir örüntüyle 3 farklı şekilde adreslenerek kullanılabilir:

a. İstasyonların adres atamaları pozisyonlarına göre gerçekleşecekse (başlık içindeki adres alanı), otomatik bir adres tanımlaması yapılır (otomatik olarak 1 artarak devam eder) ve 16 bit pozisyon ve 16 bit ofset alanından oluşan bir adres kullanılır. Böyle bir durumda "cmd" komut alanı "APxx" şeklindeki komutları içerir.
b. Kullanıcı tarafından atanan adresler kullanılacaksa, 16 bit adres ve 16 bit ofset alanları kullanılır ve komut alanı "FPxx" şeklindeki komutlardan oluşur.
c. Mantıksal bir adresleme istenirse, 32 bitin tamamı mantıksal olarak adreslenir ve komut alanı adreslemeyi tanımlamak için her komutun başına "Lxx" ifadesini ekler.

Komut tipleri ise farklılık göstermektedir:

a. Otomatik artırılan adresler ve atanmış adresler R (okuma), W (Yazma), RW (okuma ve yazma) ve RMW (okuma ve birden çok yazma) yetkisine sahiptir.
b. Mantıksal adresleme için sadece R(okuma), W(yazma) ve RW (okuma ve yazma) kullanılabilir.
c. NOP komutu herhangi bir işlem yapılmadan boş geçmek için kullanılır.
d. Broadcast komutu (Bxx) ise okuma, yazma ve okuma/yazma yetkisine sahiptir. Sayısız alıcıya giden bu paketlerde her istasyon çerçevenin belirli bir kısmından sorumludur. Başlangıç durumlarında veya kölelerin durumlarını kontrol etmede kullanılır.

EAP protokol ailesi ise altında bulunan Mailbox ve Process Data protokolleri ile iletişim kurmaktadır. EAP efendiler arası iletişimi sağlamaktadır. Burada EtherCAT başlık yapısı içindeki tip alanı 4 veya 5 olarak tanımlanmalıdır. EAP protokolü otomasyon hiyerarşisindeki hücre, MES ve ERP (ofis) iletişimini gerçekleştirmektedir. EAP içerisinde yer alan Process Data protokolü efendi istasyonlar arasında çevrimli veriyi yayınlayarak proses verilerinin değişimini sağlarken, Mailbox protokolü transferi asenkron yapılacak verilerin taşınması için geliştirilmiştir (asenkron aygıt erişimleri). Verilerin türüne göre farklı alt protokoller (Can over EtherCAT - CoE, File over EtherCAT - FoE, Automation Device Protocol Over EtherCAT - AoE, Servo Drive – SoE ve Ethernet over EtherCAT - EoE) vasıtasıyla iletişim kurabilmektedir. Örneğin CoE, Can protokol verisinin EtherCAT protokolü üzerinden taşınması için geliştirilmiştir. Her birinin çerçeve yapıları farklı olup, Mailbox protokolündeki başlık yapısını takip eden protokol veri alanı içinde tanımlanırlar. Mailbox protokolünde taşınan veri, köle istasyonların durum bilgileri, dosya transferi, aygıtların belirli parametrelerinin iletimi şeklinde olabilir.

EtherCAT iletişiminin 3. iletişim şekli ise dış dünya ile haberleşmesidir. Burada IP üzerinden haberleşme gerçekleşeceğinden, EtherCAT başlığına UPD ve IP başlıkları tanımlanır. Bunun için toplam 28 bayt uzunluğunda fazladan bir alana ihtiyaç vardır. Böylece UDP protokolü üzerinden haberleşme de desteklenmiş olur.

2.1.2. EtherCAT köle bilgisi/ağ bilgisi dosyaları (ENI-ESI)

EtherCAT tabanlı sistemlerde köle (ESI) ve ağ bilgilerini (ENI) içeren konfigürasyon dosyaları köle ve efendi istasyonlar arası güvenli bir ilişkiyi göstermektedir. Bu dosyalar XML formatında olup iletişimin başlangıç konfigürasyonlarını içermektedir. Her köle, üreticisi tarafından sağlanan ve fabrika ayarlarının belirtildiği bir ESI dosyasına sahiptir. ESI'ler üretici bilgilerini, modül, grup veya sıra numarası gibi aygıt bilgilerini ve iletişimde varsayılan olarak atanan parametreleri içerir. Bu dosyalar üretim sırasında belirlenir ve TwinCAT yüklü dizinde saklanır. Köle EEPROM'larından gönderilen tüm ESI dosyaları ve çevrimiçi bilgileri ENI adı verilen tek bir dosya halinde birleştirilir. Bu ENI dosyası efendi ve köleler arasındaki iletişim

kabullerini içermektedir. Fabrika tarafından atanmış MAC adresi, PLC konfigürasyon bilgileri, senkron veri değişim bilgilerini, Mailbox protokolü ve alt protokollerin bilgilerini, köle bilgilerini ve Process Data protokol bilgilerini içerir. ENI dosyaları gerektiğinde 3. parti yazılımlar veya mühendislik istasyonu tarafından dışa aktarılabilir. Aşağıdaki Şekil 2.4.'de ESI ve ENI konfigürasyon örnekleri gösterilmektedir.

Şekil 2.4. Örnek ENI.xml ve ESI.xml dosyaları (EL1xxx modülü)

2.2. EKS'ne Yapılan Saldırıların Tespit ve Önlenmesine Yönelik Çalışmalar

Bu bölümde önişlemcilerle ilgili bilgiler verilecek, literatürde yer alan ilgili çalışmalar EtherCAT harici protokoller tabanlı ve EtherCAT tabanlı olmak üzere 2 kategoride incelenecektir.

2.2.1. EtherCAT harici protokol tabanlı çalışmalar

Ethernet tabanlı endüstriyel otomasyon protokollerinin birçoğu yetkilendirme ve kimlik doğrulama olmadan düz metin olarak iletişim kurduğu bilinmektedir [47], [48]. Dolayısıyla protokol çerçevesi bilinirse gizlilik, erişilebilirlik ve bütünlük olarak bilinen temel güvenlik parametrelerinin bozulabileceği saldırılar geliştirilebilir [49]. Buradan hareketle literatürde SIEMENS S7 iletişiminde bütünlüğü bozmaya yönelik birçok saldırı mevcuttur. Bunlardan bazıları, PLC'deki veri bloklarının ve giriş/çıkış birimlerinin bellek adreslerini basit sorgularla tespit etmeye çalışma, modül marka bilgisini, model numarasını ve PLC özelliklerini MITM ile elde etme, mühendislik istasyonu tarafından yazılan PLC programını PLC'ye yüklenirken değiştirme veya

kod/program enjeksiyonu gibi saldırılardır [50], [51]. Buna benzer diğer saldırılar, PLC RAM belleğin görüntülenmesi, PLC donanımını durdurma/çalıştırma olarak sayılabilir. Bir başka açıklık ise PROFINET protokolünde oturum kimliği ve bilgisinin şifresiz olarak iletilmesidir. Kimlik bilgisi sunucu tarafında aynı anda kullanılmadığı sürece birden fazla kez kullanılabilmektedir. Bu zafiyetin sömürülmesi ile oturum çalma ve yetki yükseltme saldırıları yapılabilmektedir [52]. Aynı şekilde PROFINET protokolünü bulanıklaştırmaya veya servisi durdurmaya yönelik DoS saldırıları gerçekleştirilebilmektedir. Bu türden saldırılar çok karmaşık değildir. Örneğin literatürde sadece PROFINET paketlerini değiştirerek geçersiz/anlamsız bilgiler üretip DCP servislerine yönelik saldırılar gerçekleştirilmiştir [46]. Bu zafiyet 2014'te açıklanmış ve zafiyet veritabanına "CVE-2014-2252" koduyla eklenmiştir [53]. Tüm bu açıklıkların sömürülmesi ve test amacıyla kullanılmaları için geliştirilen yazılım parçacıkları Metasploit altyapısında bulunmaktadır [54]–[56].

Saldırı tespit ve izleme sistemleri, hem protokol hem de sistem üzerindeki olası atakların önceden tespiti anlamında önem taşımaktadır [57], [58]. Bu kapsamda, Ntalampiras ve ark., birbirinden bağımsız kritik altyapıları için HMI tabanlı hata izleme sistemi geliştirmişlerdir. Kritik altyapılardaki elemanlardan gelen verileri eğitim verilerine olan uzaklıklarına göre kategorilendirip, servis durdurma ve yeniden oynatma saldırıları olarak 2 çeşitte sonuç üretmişlerdir [59].

Genel amaçlı çalışmaların dışında protokol tabanlı araştırmalar da mevcuttur. Örneğin Goldenberg ve ark. Modbus/TCP protokolünü saldırı tespiti için modellemişlerdir. Burada her bir HMI-PLC arası iletişim kanalı, kararlı sonlu otomatlar (DFA) kullanılarak tanımlanmıştır. Anomali tespitinde yüksek doğruluk ve HMI üzerindeki yanlış konfigürasyonların kolay tespiti gibi sonuçlar elde etmişlerdir [60].

2014'te Siemens S7 SCADA protokolü, saldırı tespit ve izleme sistemi üzerinde modellenmiştir. Burada sadece periyodik olan iletişim temel alınmakta olup, istemci-sunucu bağlantıları gerçeklenmiştir. Eşler arası (peer communication) iletişim ve aperiyodik iletişim ele alınmamıştır [61]. Saldırı tespit sistemi geliştirilmesinin dışında özgün çözümler de yer almaktadır. Örneğin Cook ve ark., siber atakların kendilerine

has davranışları olmasından yola çıkarak ataklara kimlik belirlemeye çalışmışlardır. Ardından atak kimlik belirlemede kullanılan, dijital izleme, ağ izleme, zararlı yazılım analizi, balkovanı (honeypot), izleri takip etme gibi yöntemleri değerlendirmişlerdir [62]. Atak tespitinde farklı bir yöntem de 2015'te ajan düğüm eklenmesi olarak çalışılmıştır. Güç sistemleri üzerine kapsama ağacı yöntemiyle belirlenen en kritik düğümler arasına, ajan düğüm olarak adlandırılan ve sürekli aynı farazi verileri üreterek saldırganı yanıltmayı planlayan sanal düğümler yerleştirilmektedir. Bu ortam bir efendi istasyon tarafından izlenmekte olup, ajan düğüm verilerinin değiştirilmeye çalışılması durumunda saldırıların tespit edilebilmesine dayanmaktadır [63]. 2015'te, Modbus kullanan SCADA sistem saldırılarının zorluk dereceleri, hangi açıklıklardan avantaj sağladıkları ve amaçları, atak ağaçları kullanılarak değerlendirilmiş ve protokol tanımlamalarında nelere dikkat edilmesi gerektiği ve güvenlik riskleri ele alınmıştır [64]. Ramachandruni ve ark. Modbus ve S7 200 PLC'nin benzetimini gerçekledikleri bir balkovanı (honeypot) sistemi geliştirmişlerdir. Bu sistemi dış ağa bağlayıp 30 gün boyunca veri toplamışlar ve kritik sistemlerdeki bu protokoller üzerine oluşabilecek saldırı vektörlerini tanımlamışlardır [65].

Yukarıda geliştirilen çözüm ve sistemlerin gerçek zamanlı sistemler üzerinde test edilmesi kritik altyapıların doğası gereği neredeyse imkânsızdır. Bu kapsamda genel amaçlı yazılımsal ve donanımsal test ortamı ve simülasyon önerileri de mevcuttur. FPGA kartları üzerinde farklı mimarilerin gerçeklenebilmesinden yola çıkarak Ethernet Powerlink, Sercos ve EtherCAT protokollerinde geliştirilecek çözümlerin FPGA üzerine adapte edilmesi ile ilgili bir bilgilendirme çalışması da bulunmaktadır [66]. Benzer şekilde laboratuvar ortamında bir test ortamı oluşturan Morris ve ark., endüstriyel kontrol sistemlerindeki farkındalık ve eğitimlerin çalışan sistemler üzerinde yapılmasının zorluğundan bahsetmişlerdir. Geliştirdikleri test ortamı tamamen sanal olmakla birlikte HMI, PLC bellek haritalama, I/O bağlantıları gibi özellikleri barındırmaktadır [67].

Belirtildiği üzere literatürde tehditlerin tespit edilmesine dayalı çalışmaların birçoğu izleme sistemi geliştirilmesi, saldırı/protokollerin anormal davranışlarının matematiksel modellenmesi veya benzetimlerinin yapılmasına dayanmaktadır.

Çalışmalar hem kritik sistemleri izleyen hem de saldırılardan koruyan bir yapının gerekliliğini göstermektedir. Olası saldırıların sistem üzerinde ekstra trafik üretmeden ve yük getirmeden tespit edilmesi pasif izleme olarak adlandırılmaktadır. Bu yaklaşım gerçek zamanlılık kriterlerini bozmadan deaktif olarak anomali durumlarında uyarı vermektedir.

Gerek bilgi teknolojileri gerekse EKS ağlarında, IDS'ler ağ içerisindeki kötücül aktivitelerin tespit edilebilmesi için kullanılmaktadır. Kullanım alanlarına göre kullanıcı tabanlı, ağ tabanlı ve zafiyet değerlendirmeli IDS olmak üzere 3 farklı kategoride uygulanabilir [68]. Saldırı tespit yöntemine göre ise anomali tabanlı, kural tabanlı ve spesifikasyon tabanlı olmak üzere 3 türde uygulanabilmektedir. Bunlar içinde anomali tabanlı IDS'ler güvenlik, gizlilik ve esneklik gibi temel kriterlerde avantaja sahiptirler [69]. Yaygın olarak kullanılan bir IDS/IPS olan Snort, ağdan gelen paketleri analiz ederken içinde yer alan önişlemcilerden yardım almaktadır. Snort geliştiricileri, SCADA protokol paketlerini işlemesi için sadece Modbus/TCP ve DNP3 protokollerine ait önişlemcilerini sunmuşlardır. EtherCAT protokolü tabanlı bir önişlemci henüz desteklenmemektedir. Ayrıca, EtherCAT tabanlı bir önişlemcinin geliştirilmesi yapısının diğer 2 protokolden farklı olmasından dolayı daha güçtür. Desteklenen protokollerin ortak noktası ağdaki paketleri TCP veya UDP üzerinden transfer etmekte olmasıdır. Snort yapısı sadece 3. katmandaki önişlemci geliştirmelerini desteklediğinden bu protokollerin önişlemcileri sunulmaktadır. EtherCAT ise TCP/UDP başlığına sahip değildir ve saha/fabrika seviyesindeki iletişimde (IP tünelleme hariç iletişim için) paketlerin Snort üzerinde 2. katmandan yönlendirilmesi için ekstra bir çalışmaya ihtiyaç vardır. Bu yönlendirme ile EtherCAT paketleri sadece Ethernet çerçevesi üzerinden değerlendirilmeyecek, aynı zamanda protokole ait çerçeve yapıları da tüm alanlarıyla analiz edilebilecektir.

2.2.2. EtherCAT tabanlı çalışmalar

EtherCAT protokolünün donanım ve yazılım desteği çeşitliliği, topolojiden bağımsız çalışabilmesi ve performans gibi birçok avantajı vardır. Verilerin bayt bayt işlendiği

havada işlem teknolojisi ile kısa çevrim süresine de sahiptir [70]. Buna rağmen literatürde protokolün güvenlik yönüyle incelendiği çalışma çok azdır.

EtherCAT protokolünün DoS/DDoS gibi ataklara açık olduğu, hem MAC tabanlı hem de efendi/köle arası iletişimde, tip alanı da dâhil istenmeyen EtherCAT akışını önleyen bir sistem veya ürün ihtiyacı [71] kitabında belirtilmektedir. Buradan hareketle, EtherCAT verilerinin güvenli bir şekilde yüksek hızla toplanmasını amaçlayan gerçek zamanlı bir veri toplama sistemi geliştirilmiştir [72]. Benzer şekildeki bir öneri de dağıtık EtherCAT tabanlı sistemlerde veri toplamaya yöneliktir. Yüksek doğruluk ve hızda veri toplamayı amaçlayan bu sistemde FPGA üzerinde gerçekleme yapılmıştır [73]. Bir diğer çalışma da oluşturulan bu veri toplama sistemlerinin performans ve tasarımsal olarak analizine dayanmaktadır [74]. FPGA kullanımının gerçek zamanlılıkta avantaj sağlamasından yol çıkarak, EtherCAT efendi istasyonu veya endüstriyel Ethernet, çoklu-eksen akıllı sürücüler veya tamamen EtherCAT tabanlı köle istasyonu gerçeklenmesi gibi çalışmalar, FPGA ortamında geliştirilmiştir [75]–[78]. Efendi köle bileşenlerinin yanı sıra, EtherCAT tabanlı dağıtık kontrol sistemler de bu altyapılarda donanımsal olarak gerçeklenebilmektedir [79]. Bu veri toplama yaklaşımlarının ortak özelliği, önerilerin geleneksel veri toplama kartlarının muadili olmaları ve sistem güvenliği açısından potansiyel tehdit tespiti veya önleme gibi bir özelliklerinin olmamasıdır.

Donanımsal olarak gerçekleme dışında, yazılımsal olarak gerçekleme çözümleri de mevcuttur. Örneğin, EtherCAT çerçevesinde bazı eklemeler yapılarak öncelikli paketlerin daha hızlı bir şekilde iletiminin sağlanması veya farklı geliştirme ortamları (Matlab) ile uyumlaştırılması gibi çalışmalar da protokolün daha efektif kullanımı amacıyla gerçekleştirilmiştir [80], [81].

Protokol ve saldırı önleme sistemlerine yönelik çalışmaların yanında, protokolün iyileştirilmesine yönelik performans analizi araştırmaları da bulunmaktadır. Örneğin çerçeve boyutunu içindeki verilerin örüntüsüne göre iyileştiren bir algoritma geliştirilmesi bu çalışmalardan biridir [82]. Ayrıca, protokol etkinliğini Matlab üzerinde simülasyon ile değerlendiren veya donanımsal olarak EtherCAT anahtar

cihazlar üzerindeki gecikmeleri ölçen araştırmalar da yakın zamanda gerçekleştirilmiştir [83], [84].

Bir diğer çalışma BeStorm tarafından yapılmış olup dinamik olarak çalışan bir EtherCAT test aracı geliştirilmiştir [38]. Araç ticari olup sadece kara kutu bulanıklaştırması yapmaktadır. Kara kutu bulanıklaştırması en temel seviye bulanıklaştırma tekniği olup hedef hakkında hiçbir bilgi bilinmediğinde yapılmaktadır. Doktora kapsamında yapılan bu çalışmaya en yakın öneri ise Granat ve ark. tarafından sunulmuştur. Fakat çalışmada yenilikçi bir saldırı tespit mekanizması önerilmemiş, geliştirilen önişlemci yapısı açıklanmamış, sadece saldırı tespitinde kullanılan Snort kuralları belirtilmiştir [85].

Belirtildiği üzere EKS veya özelinde EtherCAT protokolü üzerine, gerek saldırıları tespit etme/önleme, iyileştirme yapma, gerekse bu sistemlere yönelik önerilen çözümlerin test edilmesi için çeşitli araştırmalar yapılmıştır. Fakat literatürde protokollerin güvenlik problemlerine çözüm sunacak öneriler yetersizdir. Böyle olmasına karşın sistemlerin güvenliklerini tehdit eden birçok faktör de EKS'nin yapısından dolayı bulunmaktadır. Örneğin kritik yapılı sistemlerin bulundukları stratejik konumdan dolayı küçük çaptaki saldırılar bile ölümcül sonuçlar yaratabilir. Öte yandan, diğer sistemlere uygulanan sızma testleri gibi zorunlu kontroller bu sistemlere uygulanamaz veya uygulanması önerilmez. Dolayısıyla zafiyetlerin ortaya çıkarılması güçtür. Bu sistemlerde şifreleme, kimlik doğrulama ve yetkilendirme gibi güvenlik değişkenleri de bulunmamaktadır. Çerçeve yapısı ve iletişim kabulleri bilindiğinde saldırılardan kaçınmak neredeyse imkânsızdır. Tüm bu nedenlerden dolayı EKS'ler diğer yapılara kıyasla olası saldırılara daha fazla açıktır. Tezin bu aşamasında, EKS güvenliğini artırmak amacıyla saldırı önleme ve tespit çalışmalarından yararlanılarak özelinde EtherCAT protokol açıklıklarını tespit etme ve önleme üzerine yoğunlaşılmıştır.

2.3. Periyodiklik Tabanlı ve Makine Öğrenmesi Tabanlı Anomali Tespitine Yönelik Çalışmalar

Kritik altyapılı sistemlerde gerçek-zamanlılık ve keskin zaman aralıklarındaki işlerin tamamlanması ilkeleri esastır. Bu anlamda anomali tespiti yapılacak kritik altyapılı bir sistem üzerinde öncelikli olarak sistem üzerindeki döngülerin tespiti gerekmektedir. Daha sonra, bu döngülere bağlı olarak örüntülerin tespiti ve örüntü dışındaki saldırıların anlamlandırılması yani anomali tespiti gerçekleşmelidir. Bu anlamda literatür çalışması 2 kategori halinde ele alınacaktır.

2.3.1. Periyot tespiti çalışmaları

Bir sistem üzerindeki periyodikliği belirlemek için pratik ve en sık kullanılan yöntemlerden biri spektral analizdir (ışınlar, dalga uzunlukları, güç, varış zamanları incelenmesi) [86]–[88]. Otomasyon sistemlerindeki iletişimlerin spektral analizlerinde ise uzaysal-zamansal ilişki ve benzerlik üzerine yoğunlaşılmaktadır. Bu kapsamda, Argon ve ark. yüksek ölçekli internet akışlarındaki tek bir dominant periyodun bulunması için güç spektral yoğunluğunu, çoklu periyotların tespiti için ise zamana bağlı otokorelasyon fonksiyonunu kullanmıştır [86]. İkinci yöntem gürültüden daha çok etkilenen yapıya sahip bir çözümdür. Bunun yanında zamansal ortalama alınarak uygulanan ikinci yöntemin, çoklu periyotları tespit ederek altındaki örüntülerin daha kolay anlaşılırlığını sağladığı belirtilmiştir. Fakat çalışma TCP/IP tabanlı protokollerde gerçekleştirilmiş olup, EKS tabanlı değildir. Ayrıca, güvenlik unsurları da ele alınmamıştır. Splunder ve ark., paketler arası varış zamanlarından hareketle ağdaki trafik örüntülerini tespit etmişlerdir [88]. Çalışma, paket içerikleri ve derin analizleri kapsamamaktadır. Benzer şekilde ayrık Fourier dönüşümü ve otokorelasyon araçları kullanılarak endüstriyel trafik örüntüleri tespit edilmeye çalışılmıştır [89]. Su dağıtım sistemi günlüklemeleri üzerinde gerçekleştirilen bu çalışmadaki sınırlama, anlamsal ilişkilerin göz ardı edilip sadece paket sayısı veya paket boyutu gibi geleneksel gözlemlerin elde edilmesidir. Bu türden niteliklerle örüntülerin yakalanması kolaydır ve anomali tespiti hakkında dar kapsamlı bilgiler vermektedir [90]. Bunun yanı sıra çalışmada düşük frekanslar gözardı edilmiştir, fakat komut enjeksiyonu gibi saldırılar

bu türden ataklardandır. Ayrıca çalışma sadece TCP/IP iletişim protokollerini kapsamaktadır. Bu çalışmalarda yer almayan anlamsal ilişkileri kapsayan 2013 yılındaki çalışmada, Modbus/TCP protokol iletişim kanalı deterministik sonlu otomata (DFA) kullanılarak ve 100 adet mesaj yakalanarak Python'da modellenmiş olup, modele uymayan geçişler veya tanımlanmamış durumlar tespit edilmiştir [60]. DFA oluşturulması sırasındaki eğitim ve test aşamaları 2 veriseti tarafından (PLC-HMI arası iletişim) Pcapy ve Impacket kullanılarak elde edilmiştir. Çalışmadaki sınırlama yeniden oynatma gibi veya kabul edilen paketlerden oluşan DoS saldırılarına karşı çözümü kapsamamaktadır. Ayrıca gönderilen Modbus/TCP istek mesajlarındaki küçük farklılıklar tespit edilememektedir. Bir başka dezavantajı ise geçişler ve durumlarda zamansal ilişki göz ardı edilmiştir. 2016'da Barbosa ve ark., endüstriyel kontrol ağındaki mesaj tekrarlarından ve zamanlarından hareketle, trafik içindeki periyodik örüntüyü bulan bir teknik önermişlerdir [91]. Veriler, iki adet su üretim ve bir adet elektrik-gaz toplama ağından toplanmış olup periyodiklik 3 adet modül geliştirilerek bulunmuştur. 4. modülde ise periyot dışındaki durumların tespiti yapılmıştır. Bu çalışmadaki çözümler protokol tabanlı olmayıp genel olarak endüstriyel otomasyon sistemleri için önerilmiştir. Geliştirilen çözümlerde protokol bazlı modifikasyonlar gerekmektedir. Anlamsal ilişkilere Modbus/TCP üzerinden bakılmıştır. Bu protokol, TCP/IP altyapısı üzerinden iletişim kurmakta olup ve karmaşık olmayan bir standarttır. Ayrıca, çalışmada çevrimsiz olan iletişimler ve istek mesajlarının sırası göz ardı edilmiş olup, saldırı tespiti/analizi gibi tespitler bulunmamaktadır. 2015'te ayrık-zamanlı Markov zincirleri kullanılarak normal iletişim modellenip, belirli bir sırada gerçekleşen saldırılara tespit önerisi sunulmuştur [92]. Karmaşık periyodik örüntüler büyük model oluşturulmasına sebep olacağından saldırı tespiti zorlaşacaktır. Broido ve ark. ise bir DNS sunucu üzerindeki 1, 7 ve 26 günlük DNS güncelleme günlüklemeleri üzerindeki adreslerin frekans spektrumundaki IP-makine adı eşleşme güncellemelerini analiz etmişlerdir [93]. Güncelleme periyotları varış zamanlarına bağlı olarak otokorelasyon yöntemi ile tespit edilmiş ve genellikle 60-75 dakikada tekrarlandığı belirtilmiştir. Daha sonra periyot içine düşen güncellemeler analiz edilmiştir [94].

2.3.2. Anomali tespiti çalışmaları

Ağ üzerindeki anomalileri tespit etmeden önce, normal olarak nitelendirilen davranışların tanımlanması gerekmektedir [95]. Buna göre ağın normal durumu, tüm sistem dinamiklerini içeren temel değişkenlerin birbirleri ile olan ilişkilerinin bir iletişim modeli tarafından temsil edilmesi olarak tanımlanır. Bu formal modelden değişim derecesi belirlenmiş oranda yüksek olan bir olay veya obje ise anormaldir. Anomaliler Ahmed ve ark. tarafından noktasal, anlamsal ve toplamsal olarak 3 grupta incelenmiştir. Buna göre, normal bir şekilde seyreden örüntünün ani değişimiyle oluşan olay noktasal, belirli bir içerik içerisinde olayın anormal olması anlamsal, tek başına anormal olmayan birden fazla olayın, birleştiklerinde anomali teşkil etmesi ise toplamsal anomali olarak tanımlanmaktadır [96]. Bunun dışında, Barford ve Plonka anomalileri üç grupta incelemişlerdir. Bunlar, ağdaki donanım, konfigürasyon değişimlerinden veya kesintilerden kaynaklı olaylar, yığılmalardan ötürü aniden yükselen, yazılım tabanlı bir sisteme veya web siteye erişimlerin fazlalaşması gibi zamanla düzene giren anomaliler ve ağın kötüye kullanıldığı saldırılar olarak tanımlanmıştır [97]. Yine Sestito ve ark. anomalileri dört grupta açıklamıştır [98]. Bunlar; ağdaki işlemlerden kaynaklı, ani değişimlerden kaynaklı, ölçüm hatalarından kaynaklı ve saldırılardan kaynaklı anomalilerdir.

Endüstriyel kontrol sistemlerinde saldırılar protokol tabanlı ve sistem tabanlı olarak iki ana grupta incelenebilir. Protokol tabanlı saldırılar genellikle protokolün kendine has açıklıklarının sömürülmesi yoluyla gerçekleşir. Örneğin segmentasyon hataları, bellek taşması, yeniden oynatma saldırıları, yığın taşması, protokol çerçevesinde paket üretip sisteme yollayarak sistemin tepkisini ölçmeye yönelik MITM saldırıları, fragmantasyon saldırıları, protokolün kullandığı portlara yapılan DoS saldırıları protokol tabanlı saldırı türlerindendir [8]. Veritabanına yönelik yapılan enjeksiyon denemeleri [20], PLC RAM belleği üzerindeki çalışan programın bulunduğu alan ve program durum, değişkenlerin tutulduğu kaydedici üzerine saldırı [6], ağ cihazlarına yönelik kimlik denetimi saldırıları (kriptografik), kullanılan yazılımlardaki açıklıkların sömürülmesine yönelik bellek taşması, yetki yükseltme gibi saldırılar ise sistem tabanlı saldırı türlerindendir. Ağ üzerindeki anomalileri önlemek için

sınıflandırma tabanlı anomali tekniklerinden Destek Vektör Makineları, Bayesian Network, Neural Network, Kural Tabanlı yöntemler gibi yöntemler kullanılmaktadır. İstatistiksel olarak da saldırılar önlenebilmektedir [96]. Temel Bileşen Analizi, Sinyal İşleme Teknikleri, korelasyon analizi gibi yöntemler bu grupta yer almaktadır. Bunun dışında sistem modeli oluşturma ve kümeleme gibi yöntemler de literatürde sıklıkla kullanılmaktadır. Geleneksel kural tabanlı sistemler, çok fazla kural yazılması gerektiğinden, kural veritabanı kullanılsa bile imzası bulunmayan bir saldırı önlenemediğinden sıfırıncı-gün saldırılarında başarısızdır. Model oluşturma yöntemlerinde karmaşık model oluşturulması zor olduğundan tespiti yetersizdir. Spesifikasyon tabanlı çözümler ise karmaşık ve dinamikleri farklı olan EKS üzerinde doküman ve manueller yetersiz veya eksik olduğundan zordur [99].
Anomali tespiti için literatürde çok fazla sınıflandırma tabanlı yöntem kullanılmıştır. EKS anomali sınıflandırmasında önemli olan hızlı, ölçeklenebilir ve gürbüz bir çözüm üretmektir. Kritik altyapılarda çalışan sistem üzerinden veriseti elde etmek stratejik açıdan güç olduğundan veya mevcut veri setlerindeki saldırılar iyi tanımlanmadığından, laboratuvar ortamında test verisi oluşturmak gerekmektedir. Junejo ve Goh su arıtma sistemi test ortamı oluşturup 10 adet proses tabanlı saldırı oluşturmuşlardır [99]. Bunun yanında Grosso ve Sparks saldırı tespiti yapmak için test girdisi üretilmesi ile ilgili öneri sunmuşlardır [100], [101]. Maglaras ve Yoo ise hali hazırda saldırı vektörü olmadığından, sadece saldırı tespit modeli geliştirmiş, FNR (yanlış negatiflik oranı), doğruluk gibi performans metriklerini incelememişlerdir [102], [103].

Literatürdeki EKS saldırı tespit çalışmalarının birçoğu paket günlüklemeleri üzerinde çalışmakta olup, gerçek zamanlı bir yaklaşıma sahip değildir [99], [102]–[105]. Ancak, yakın zamanda yapılmış [106] çalışması, ağın eğitim aşamasında günlükleme, test aşamasında gerçek zamanlı sistem üzerinden gelen paketler kullanmıştır.

Anomali tespitinde gelişmiş analiz teknikleri ile izleme önem arz etmektedir [107]. Bu analitik süreçler 3 farklı kategoride yürütülebilmektedir: Açıklamalı, tahminsel veya kurala dayalı [92]. Açıklamaya dayalı yöntemde geçmiş ve şimdiki zaman aktiviteleri izlenerek çıkarım yapılır. Tahmine dayalı yöntemde gelecekteki durumlar üzerinde

durulurken, kurala dayalı yöntemde olan veya olması muhtemel olaylar için çözümler ele alınmaktadır. Tahmine dayalı analizlerde örüntülerin analizi; trend oluşturma, olasılık hesaplama ve belirsizlik azaltma gibi amaçlarla gerçekleştirilmektedir. Bu işlemlerde zaman-serileri analizleri, ekonometrik analizler, karar ağaçları, vektör destek makineları, sinir ağları, Bayes sınıflandırması vb. yöntemler yer almaktadır. Zaman serilerine bağlı örüntü tespiti birçok alanda uygulanmaktadır. Örneğin, veri madenciliğinde bir dizideki sembollerin zaman serileri yöntemi kullanılarak tekrarlamalarının tespiti yapılmaktadır [108]. Otokorelasyon fonksiyonu aracı (ACF), genellikle müzik, insan konuşma sinyallerindeki periyodiklik gibi döngüsel bir çevrimi çıkarmak için kullanılmaktadır [86], [109]. Bunun dışında makine öğrenmesi de tahminsel analizlerde sıklıkla tercih edilmektedir. Ancak, ağ içerisinde IDS tabanlı çalışmalarda, belirsizlik tespiti problemi, yüksek hata maliyeti, içeriklerin göz ardı edilmesi, trafik yoğunluğu, değerlendirme zorlukları sebebiyle makine öğrenmesini efektif olarak kullanmak güçtür [110]. Makine öğrenmesi, belirsizlik bulmada değil sınıflandırmada başarılı bir yöntemdir. Bu kapsamda makine öğrenmesi bilinmeyen saldırıların bulunmasından ziyade, saldırı çeşidinin tespitinde kullanılması tercih edilmeli ve aynı zamanda küçük ve orta ölçekli ağlarda uygulanmalıdır. İbrahim, dağıtık zaman-gecikmeli sinir ağları kullanarak kılavuzlu yapay sinir ağı (YSA) modeli ile saldırı tespiti gerçekleştirmiştir [111]. Çalışma, diğer ANN tabanlı çalışmaların tespit oranları ile karşılaştırılmıştır. Gu ve ark. ise imza ve komut kontrol sunucusu bilgileri olmadan, yerel bir ağda olasılık tabanlı korelasyon kullanarak anomali tespiti yapan bir metot geliştirmişlerdir [87]. Anomali olarak ağ trafiğindeki Botnet komut ve kontrol kanalları tespit edilmektedir. Tespit, Snort üzerinde önişlemci şeklinde geliştirilmiş izleme modülün aktivite günlüklemelerini toplayıp, ilişkisel analizi yapan modüle aktarılması için dağıtım birimine iletmesi yoluyla yapılır.

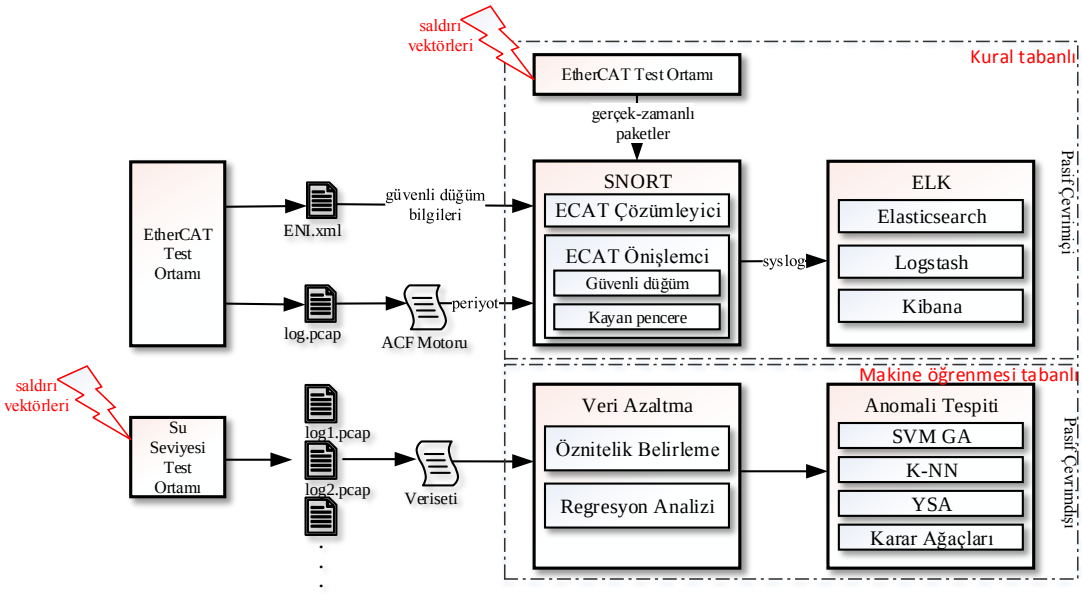
TCP/IP tabanlı protokollerin EKS'ye entegre edilmesiyle, geleneksel bilgi teknoloji ağlarında yer alan açıklıkların bu sistemlere de taşındığı, bu yüzden de bu sistemler üzerindeki yüksek sayıdaki sıfırıncı-gün saldırılarının yapılmasının araştırmacıları anomali tabanlı çalışmalara yoğunlaştırdığı bilinmektedir [112], [113]. Bilgi teknolojileri ağlarında yapılan anomali tespiti çalışmaları ağdaki fazla değişken trafik yüzünden yüksek yanlış-pozitif (FP) oranına sahiptir [110]. Buna karşın, endüstriyel

ağlarda trafik fazla değişkenlik göstermemekle birlikte, geçerli komutlar ve komut gönderme sıklıkları bilinirse kod enjeksiyonu, DoS gibi saldırıların önüne geçilebilir [91]. Gao ve ark. laboratuvar ortamında geliştirilen bir su seviyesi kontrol sistemine DoS, MITM, yeniden oynatma saldırıları gerçekleştirmişlerdir [114]. Ardından geri yayılım algoritması kullanarak YSA tabanlı anomali tespiti gerçekleştirmişlerdir. Geri yayılım algoritmasında yüzdesel olarak su seviyesi, cevap sıklığı ve su tankı pompasının açık/kapalı olması giriş olarak alınmıştır. Benzer şekilde Linda ve ark. laboratuvar ortamında geliştirdikleri kontrol sistemi üzerinde Nmap, Nessus, Metasploit araçları yardımıyla yapay olarak gerçekleştirilen saldırıları yapay zekâ kullanarak tespit etmişlerdir [19]. Bu tespitler, yakalanan belirli bir pencere boyutundaki paketler içerisindeki IP adresi, varışlar arası zaman, protokol sayısı, bayrak kodları, veri uzunluğu gibi öznitelikler üzerinden yapılmıştır. Fakat yapılan çalışma genel kapsamlı olup, semantik olarak yetersizdir. Yang ve ark. ise bir test ortamında Autoassosiative Kernel Regression (AAKR) ve istatistiksel olasılık oran testi (SPRT) kullanarak anomali tespiti yapan bir çalışma gerçekleştirmişlerdir [115]. 2013'te Kim ve ark., m-bağlantılı SCADA ağları için kural dizisi kullanarak anomali tespiti yapan IDS taslağı önermişlerdir [116]. Önerileri kritik altyapıların hiyerarşik yapısını içermemekte olup, atanmış bir hat üzerinden bilgi toplanması önerilmemiş ve herhangi bir test/simülasyon ortamında gerçeklenmemiştir. Hadeli ve ark. sistem konfigürasyon dosyalarında yer alan sistem açıklamalarından yola çıkarak, anomali tespiti yapan ve güvenlik mekanizmaları (güvenlik duvarları, Snort vb.) için konfigürasyon önerileri sunan bir yapı önermişlerdir [117]. Bu önermenin uygulanacağı her bir protokol yapısı için özel bir ayrıştırıcı gerekmektedir.

Protokole özel çalışmalar da mevcuttur. 2013`te Morris ve ark. 50 adet taslak formatta Modbus protokolüne ait kural tanımlamışlardır [118]. Burada unutulmaması gereken nokta, IDS kurallarının sadece Modbus tabanlı değil, diğer birçok protokolde gerekli olduğudur. Benzer şekilde 2015`te Erez ve Wool, Modbus/TCP protokolünde kontrol kaydedici değerlerindeki değişiklikleri tespit eden bir yapı önermişlerdir [119]. Sonlu Durum Makinesı (FSM) kullanarak kaydedicileri sınıflandırdıktan sonra, istatistiklere bakarak etkinlik alanı tabanlı anomalileri tespit etmişlerdir. Burada TCPdump ağ dinleme programı ile bir sistemi dinleyip, her paketin ilk 96 baytlık kısmını kaydedip,

2 adet veriseti üzerinden çalışmışlardır. Pencere içerisindeki algılayıcı kaydedicilerin en düşük ve en yüksek eşik değerlerinin dışına çıkan paketlerde anomali belirlemişlerdir. Aynı sistemden verileri toplayan Wool ve Goldenberg ise Modbus trafik örüntülerinin komut ve bellek erişimlerini dikkate alarak modellemişlerdir [60]. Barcelona şehrindeki su dağıtım sisteminden toplanan algılayıcı verilerinde kayıp, yanlış veri olup olmadığını kontrol eden Quevedo ve ark. da çalışmalarını yine Modbus protokolü üzerinde yapmışlardır [120]. Burada günlük ve 10 dk' lık model üzerinde çalışılmıştır. Günlükte ARIMA modeli, 10 dk' lık talep örüntüsünde korelasyon analizi ve kılavuzsuz bulanık mantık sınıflandırması (Lamda) kullanılmıştır. Veri doğrulaması genel olarak 2 farklı yolla yapılabilir: sinyal tabanlı (düşük - seviye) ve model tabanlı (yüksek - seviye). Sinyal tabanlı doğrulamada sinyal verileri ve değişimleri baz alınırken, model tabanlıda farklı zaman ve uzaydaki ilişkiyi bozan çok değişkenli prosedürlere veya farklı miktarlar arasındaki analitik ilişkiye bakılmaktadır.

Literatür çalışmasında görüldüğü üzere EtherCAT protokol güvenliğine katkı sağlayan detaylı bir çalışma henüz yapılmamıştır. Protokolün Ethernet tabanlı olması, şifreleme, kimlik doğrulama ve yetkilendirme mekanizmalarına sahip olmaması nedeniyle zafiyetleri de mevcuttur. Bu anlamda hem kural tabanlı hem de anomali tespitine dayanan çözüm önerilerine ihtiyaç duyulmaktadır. Çözümün kural ve anomali tabanlı olması, hem bilinen hem de bilinmeyen saldırıların tespiti anlamında önem arz etmektedir. Tez kapsamında, güvenli düğüm yaklaşımı tabanlı ve periyot tabanlı olarak kurala dayalı 2 adet ve makine öğrenmesi tabanlı anomali tespitine dayanan, hem bilinen hem de bilinmeyen saldırılara çözüm öneriler Şekil 2.5.' teki gibi çalışılmıştır. Bu şekil, tez çalışmasının nihayi sonuçlarını göstermekte olup, sonraki bölümlerin daha iyi yorumlanabilmesi adına sunulmaktadır. Bu nedenle, sonuçlar ve tartışma bölümünde ayrıca tartışılacaktır.



Şekil 2.5. Tez çalışmasının özeti

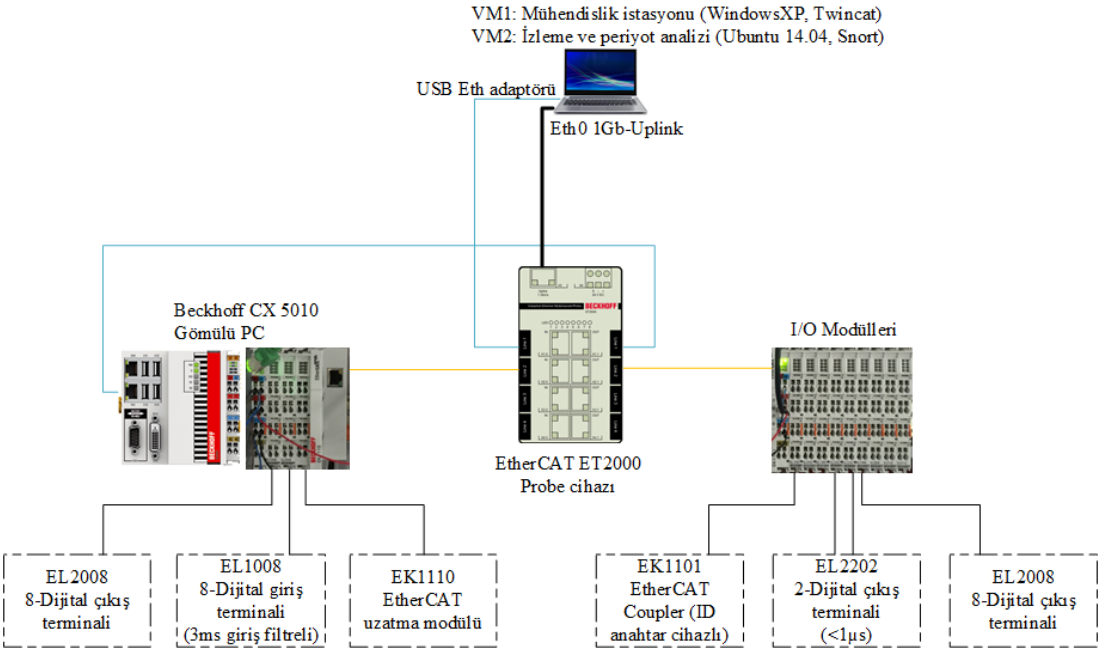
BÖLÜM 3. ETHERCAT PROTOKOLÜNDE ZAFİYET TESPİTİ

Bu bölümünde kullanılan test ortamı tanıtılmış, EtherCAT saha seviyesi iletişimi daha önce literatürde yer almayan güvenlik bakış açısı ile ele alınmış ve protokol zafiyetleri incelenmiştir. Protokol kimlik doğrulama, şifreleme ve yetkilendirme gibi temel güvenlik parametrelerini içermediği için ortam erişim denetimi (MAC) sızdırma, uzaktan veri yürütme ve diğer ileri seviye bilgi gerektiren gelişmiş saldırılara açık olduğu görülmüştür.

3.1. Test Ortamı

Endüstriyel ağlarda gerçek zamanlılık, çevrimler ve donanımsal karakteristikler gibi belirli kavramların iletilen verilerin doğru ulaştırılmasında önemi büyük olduğundan, geliştirmenin sonrasında test ortamı oluşturulmalıdır. Literatürde saldırıları hem simülasyon [39], [121], [122], hem de saldırı önleme ve sistem geliştirme tabanlı [123]–[126] çeşitli altyapılar yer almaktadır. Bu çalışmalardan bazıları ise veri toplamaya yönelik veya iletişim altyapılarına alternatif topolojiler olarak sunulmuştur [127]–[129].

Tez çalışmasının 3, 4 ve 5. bölümünde yapılan çalışmalar, Sakarya Üniversitesi - Siber Güvenlik Laboratuvarı bünyesinde yer alan donanımlardan analog/dijital giriş çıkış birimleri, Windows CE tabanlı Beckhoff PLC, mühendislik istasyonu, ağ Probe cihazı ile oluşturulan test ortamında uygulanmıştır (Şekil 3.1.).



Şekil 3.1. Test ortamı

Test ortamında yer alan ET2000 ağ Probe cihazı; EtherCAT protokolü için gerçek-zamanlı sistemlere uygun olarak tasarlanmış, 4 adet giriş ve 4 adet çıkış kanalını desteklemektedir. Toplam 4 adet kanaldan gelen verileri zaman damgası basarak 1000Mbps hızındaki Ethernet Uplink kanaldan birleştirerek göndermektedir. Bu cihazın gerçek-zamanlı iletişime ve ağın bant genişliğine herhangi bir etkisi bulunmamaktadır. Pasif modda ağı dinlemekte ve orijinallerini çıkış kanalından hedeflerine yollarken kopyalanmış olanları da Uplink kanalından alıcıya teslim etmektedir. Şekil 3.1.'deki topolojide; mühendislik istasyonu (VM1) üzerinden PLC programlama ve PLC-IO birimleri arasındaki iletişim için toplam 2 kanal verisi, Uplink üzerinden izleme ve saldırı tespiti için toplanmaktadır. Endüstriyel çok kanallı Probe cihaz yardımıyla kopyalanan ağ verileri, gerçek zamanlı zaman-damgaları basılarak VM2'ye gönderilmektedir. EtherCAT anahtarlama linki (ESL) olarak da adlandırılan zaman damgası, 16 bayt uzunluğunda olup, Şekil 3.2.'deki yapıdadır ve gelen her paketin sonuna ET2000 tarafından eklenmektedir.

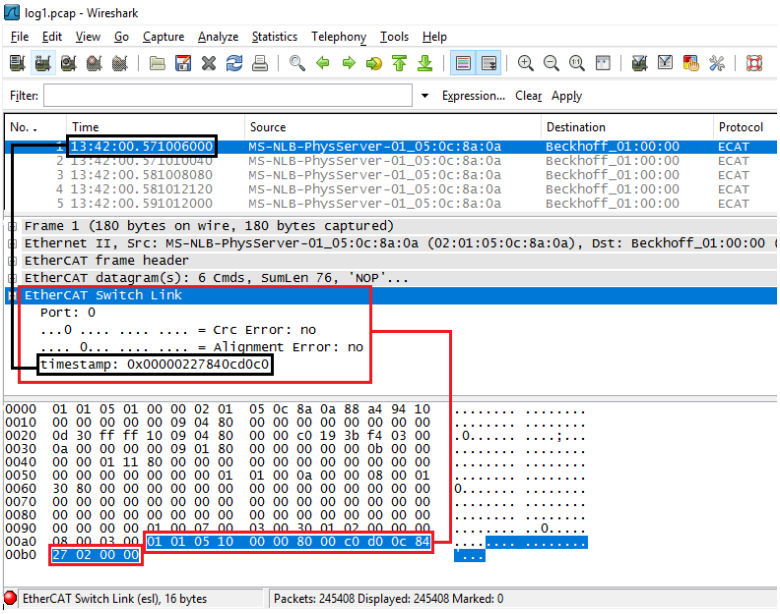
6 bayt	1 bayt	1 bayt		8 bayt
MAC Adresi	Port	Rezerve	CRC Hatası	Zaman Damgası

Şekil 3.2. ESL yapısı

Paket eğer 1500 bayttan (maksimum Ethernet çerçeve uzunluğu) büyükse bu işlem için maksimum paket uzunluğu ayarının değiştirilmesi gerekmektedir. Zaman-damgasının ilk 6 baytı sembolik MAC adresini, 1 baytı paketin geldiği kanalı (port), 1 baytı CRC hatası/sıralama hatası/rezerve ve son 8 baytı zaman-damgasını temsil etmektedir. Wireshark, ESL alanlarını tanıyabilmekte ancak sadece Beckhoff tarafından güncellenen ECAT ayrıştırıcılarına sahip sürümleri ayrışmış zaman-damgalarını liste panelinde görüntüleyebilmektedir. ESL eklenen bu paketler, çözümlenen zaman-damgaları sayesinde Şekil 3.3.'de görüldüğü üzere gerçek zamanlı olarak izlenebilmektedir. Bunun için mutlak zaman tip alanı ve nanosaniye birimlerinin aktive edilmiş olması gerekmektedir.

Test ortamında PLC efendi istasyonu için EtherCAT destekli Beckhoff CX 5010 gömülü bilgisayar kullanılmıştır. Bu bilgisayar üzerinde TwinCAT ve Microsoft Windows Embedded CE 6 yüklü Intel Atom işlemci bulunmaktadır. PLC üzerinde ayrıca 2 adet 8 bitlik dijital giriş/çıkış birimi ve 1 adet uzatma modülü yer almaktadır.

Bu bölümde yapılacak yürütülen çalışmalar için sistem üzerinde devamlı çalışan bir iş oluşturulmuştur. Geliştirilen program, çıkış birimi üzerindeki modülde yer alan bir ışığı 5 sn. süreyle açıp, sonrasında 5 sn. süreyle söndürmekte ve bunu sürekli gerçekleştirmekte olup, ilgili program CX 5010 üzerine koşulmuştur.



Şekil 3.3. ESL içindeki zaman-damgası çözümlemesi

3.2. EtherCAT Protokolünde Zafiyet Tespiti

Bu bölümde EtherCAT üzerinde gerçekleştirilen zafiyet tespiti çalışmaları anlatılacaktır. Zafiyet tespitinde bulanıklaştırma (fuzzing) ve saldırı vektörü olmak üzere 2 farklı yaklaşım uygulanmıştır.

3.2.1. Bulanıklaştırma

Donanım, yazılım veya sistemlerdeki hataları bulma, servis kalitesi, anomali tespiti veya tepki ölçme, gibi amaçlarla uygulanan bir yöntemdir. 1989 yılında çevirmeli modemlerin kullanıldığı dönemde Prof. Barton Miller tarafından, telefon hattında gürültüye neden olan yağmur nedeniyle, komutların iletilememesini tespit ederek (hata doğrulayan modemlerden önce), lisansüstü eğitim alan bir öğrencisine işletim sistemleri ödevi olarak vermesi ile ortaya çıkmıştır [130]. Gelişen teknolojiyle yeni donanım, yazılım geliştirilmesi veya mevcutlara özellik eklenmesi kod karmaşıklığı ve sistemlerin güvenliği gibi konuların yıllar içinde önemini artırmasına neden olmuş ve araştırmacıların bu alana yönelmesinde etkili olmuştur.

Bulanıklaştırma yöntemi kendisi küçük fakat etkisi yüksek birçok zafiyetin tespitine yardımcı olabilir. Bunlardan bazıları programcıların sıklıkla yaptıkları basit programlama hatalarından kaynaklanmaktadır. Örneğin,

a. Kapsamı algoritmada uygularken yapılan hatalar
   Sayma hatası: "30 metre uzunluğundaki bir çit 5 metre aralıklarla bölünmek istenirse kaç demir gerekir?" gibi basit bir problemin hesaplama hatası algoritmanın yanlış yazılması ile sonuçlanabilir.
b. Programlama dili kullanım hatası: 32-bit bir sistemde işaretsiz bir tamsayı 0-+4,294,967,295 arası değer alabilir. Bu değerden büyük bir rakam depolanacaksa 33bitlik bir alan gerekmektedir. Çünkü "4,294,967,295" sayısı "1111 1111 1111 1111 1111 1111 1111 1111" sayısına eşittir ve bu rakama 1 eklenirse 0 elde edilir ve taşma meydana gelir. Programcının yapılacak işleme göre bu durumları kontrol ederek uygulaması gerekmektedir. Aksi takdirde zafiyet mevcut olur.
c. Yığındaki tampon taşması: Uygulamanın yığın içindeki tamponda yeterli alan olup olmadığını kontrol etmeden güvensiz verileri tampona kopyalaması sonucunda, tamponda yeteri kadar boş alan yok ise, tampon dışındaki bellek alanlarına taşma olabilir. Taşan veri o tamponu kullanan program tarafından yararlı olmayabilir fakat taşma olan alanı kullanan diğer bir programın kullandığı bellek alanı olduğundan farklı bir program tarafından zaman içinde kullanılabilir. (örneğin, program fonksiyonunun "return" durumunda hangi adresten devam edeceğinin yazıldığı yığındaki bellek alanı vb.) Kötü amaçlı yazılan değerler ise bu programın çökmesine veya yanlış işlemesine neden olabilir. Bu saldırı DoS atağı olarak değerlendirilebilir.
d. Yığın taşması: Bir programın sonsuz olarak belirli bir fonksiyonu çağırması sonucu yığında depolama alanı kalmaması durumudur. Burada saldırgan yığın üzerine sürekli istediği veriyi de yazabilir Böyle bir durumda kodlar artık bellekte yer aldığından sonraki bir saldırı senaryosunda kullanılabilir.

Yukarıda örneklendiği gibi, yapılan hatalar saldırganlar tarafından sömürülebileceğinden, bulanıklaştırma yöntemiyle bu zafiyetlerin önceden tespit

edilmesi önem arz etmektedir. Bulanıklaştırma çalışmaları hedef sistem ve uygulama tekniği bakımından birkaç şekilde gerçeklenebilmekte olup, bu çeşitlilik Tablo 3.1.'de özetlenmiştir. Paket üretiminde izlenecek 2 çeşit yöntem bulunmaktadır. Bunlardan ilki mutasyon tabanlı paket üretimidir. Burada ağdaki paketlerin örnekleri alınarak benzer paketler oluşturulup bir trafik üretilir. Diğer bir değişle eldeki veriler manipüle edilerek zafiyet aranır. Diğer bir yöntem ise üretim tabanlı veri oluşturulmasıdır. Burada hedef sisteme gönderilecek veriler belirli algoritmayla modellenerek ya da bilinçli bir üretim yapılarak oluşturulur. Bunun haricinde bulanıklaştırıcılar hedefin bilinmesi, hedef hakkında hiçbir bilgi sahibi olunmaması veya kısmi bilgi sahibi olunmasına göre de gruplandırılırlar. Örneğin, hedef hakkında bilgi sahibi olup, hedeften gelen paketleri değiştirerek bulanıklaştırma yapmak, beyaz kutu mutasyon tabanlı bulanıklaştırma olarak tanımlanır. Bunun yanında paket üretiminde ister mutasyon isterse üretim tabanlı olsun, ya da beyaz/kara/gri kutu bulanıklaştırma yöntemi olsun, bir bulanıklaştırıcı aracı aptal ya da belirli bir algoritmayla veriseti oluşturan bir yapıya sahip yani akıllı olabilir. Yeni nesil bulanıklaştırıcılar akıllı bulanıklaştırıcı kavramı ile geliştirilmesi zor ve sadece belirli bir yapı için oluşturulduklarından az da olsa günümüzde kullanılmaktadır.

Tablo 3.1. Bulanıklaştırıcı terimleri

Bulanıklaştırıcı	**Hedef**	**Uygulama Verisi Oluşturma**
Kara Kutu Bulanıklaştırıcı	Bilinmiyor	Mutasyon Tabanlı/Üretim Tabanlı
Beyaz Kutu Bulanıklaştırıcı	Biliniyor	Mutasyon Tabanlı/Üretim Tabanlı
Gri Kutu Bulanıklaştırıcı	Kısmen biliniyor	Mutasyon Tabanlı/Üretim Tabanlı

Literatürde bulanıklaştırma yapılmasını sağlayan bazı altyapılar mevcut olup, başlıcaları Tablo 3.2.' de sıralanmıştır. Bunlardan en çok kullanılan Spike, Sulley ve Peach'tir [131]. Açık kaynak kodlu, geliştirilmesi devam eden ve protokol modellemeye olanak sağlayan en uygun altyapı Peach'tir. Peach altyapısıyla, paketlerin farklı yöntemlerle üretilmesi ve ilgili paketlerin yayımlayıcılar sayesinde ağa gönderilmesi sağlanabilmektedir. Bunların haricinde Scapy Kütüphanesi de popüler bir güvenlik testi altyapısıdır. İçerisinde bulunan fonksiyonlarla ağ tarama, paket üretme veya bulanıklaştırıcı geliştirilebilir. Fakat günlükleme, ajan çalıştırma gibi fonksiyonları bulunmamaktadır.

Tez kapsamında bu altyapılardan en uygun olan Peach ile mutasyon tabanlı bir bulanıklaştırıcı geliştirilmesi test edilmiş olup, mutasyon stratejilerinin yeterli olmaması, yazılan değiştiricilerin ise bit türünde tüm esnekliği sağlamaması nedeniyle bu altyapı kullanılmamış, daha esnek ve hızlı testleri gerçekleştirmek için altyapıdan bağımsız bir bulanıklaştırıcı gerçeklenmiştir.

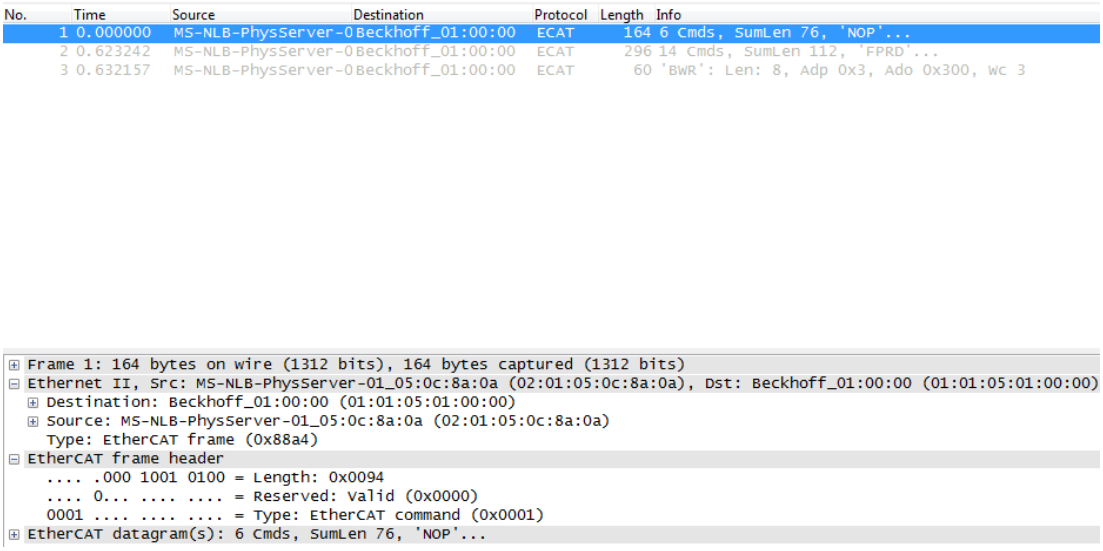
Tablo 3.2. Bulanıklaştırma altyapıları

Araç	**Özellik**
Spike	Script yazılarak her olay gerçekleniyor, açık-kaynak, çökme olduğunda manüel başlatmak gerekli, eskiden popülerdi
Peach	Geliştiriliyor, protokol modelleme var, açık-kaynak, çökme olursa otomatik devam edebiliyor, uzun süreli testlere uygun
Sulley	Geliştirilmiyor, açık-kaynak, Spike'den sonra geliştirildi, modül sayısı fazla
Scapy	Python kütüphanesi, paket oluşturma, ağ trafiği tarama, açık-kaynak
BeStrorm	Ücretli, geliştirme yapmaya çok uygun değil
Mu Test Suite	IEC61850, MODBUS ve DNP3 için geliştirilmiş, gelişimi devam etmekte

3.2.1.1. Gri-kutu EtherCAT bulanıklaştırıcı geliştirilmesi

Çalışma "C" programlama dilinde ve paket tabanlı işlemlerin yapılabildiği "pcap" kütüphanesi eklenerek geliştirilmiştir. Derleme ise "GNU C Derleyicisi" üzerinde, yazılan "makefile" ile tamamlanmıştır.

Paket içerisinde manipüle edilecek alanlar belirlenerek kısmen gri kutu bulanıklaştırıcısı geliştirilmiştir. Çalışmada örnek PCAP dosyaları kullanıldığından mutasyon tabanlı paket üreten bir bulanıklaştırıcıdır. Gerçek zamanlı çalışan bir sistemden elde edilen PCAP örnekleri incelendiğinde 3 farklı çerçeve yapısına ait çok sayıda paket bulunduğu gözlemlenmiştir. Bu paket yapıları Şekil 3.4.'de gösterilmiş olup Ethernet başlık yapısı içindeki kaynak ve hedef MAC adresleri örnekteki gibi tespit edilmiştir. Tip alanı ise EtherCAT protokolüne ait 0x88a4 olarak atanmıştır. EtherCAT başlık yapısı içindeki uzunluk alanı her 3 çerçevede de değişirken, "reserve" alanının içi boştur. Komut olarak ise tip 1 türü yani EtherCAT aygıt iletişimi sırasında kullanılan çerçeve yapısının kullanıldığı anlaşılmaktadır. Bir başka değişle saha ve algılayıcı eyleyici seviyelerindeki iletişim örneğini içermektedir.



Şekil 3.4. EtherCAT PCAP örneği

Geliştirilen program öncelikle PCAP dosyasını alıp paketin başında yer alan zaman damgası, uzunluk gibi alanlar ayrıştırılmıştır. Ardından PCAP içindeki ilk paket işaretlenmektedir. Paket içerisinde bulanıklaştırılacak olan alanlar belirli bir yapı ile belirlenmektedir. İlk olarak Ethernet başlığı içerisindeki kaynak, hedef MAC adresleri ve tip alanı kontrol edilmekte ve yeni bir yapıya aktarılmaktadır. Sonrasında gelen EtherCAT başlık yapısı da yeni oluşturulan bir yapı içerisine yerleştirilmektedir. Bu iki başlık yapısı bulanıklaştırılacak olan paketin başına değiştirilmeden eklenmektedir. Mutasyon edilecek olan paket içerisindeki 14 baytlık Ethernet ve 2 baytlık EtherCAT çerçeveleri üretilecek olan yeni paketin başında 16 bayt yer kaplayacak şekilde yerleştirilmektedir.

Çerçeveler dışında kalan alanlar 10 bayt uzunluğundaki datagram başlık yapısı, en fazla 1500 bayt olabilen veri alanı ve çalışma sayacı alanı olan 2 baytlık bir alandır. Bu alanların gerekli bitleri bulanıklaştırılmaktadır. Ethernet çerçeve yapısı dışında yer alan çerçeve kontrol bitleri (FCS) ise fiziksel ortama verilirken donanımsal olarak gerçeklendiğinden bu alan üzerinde herhangi bir işlem yapılmamaktadır. Datagram başlık yapısı ve çalışma sayacı tüm durumlarda bulanıklaştırılmaktadır.

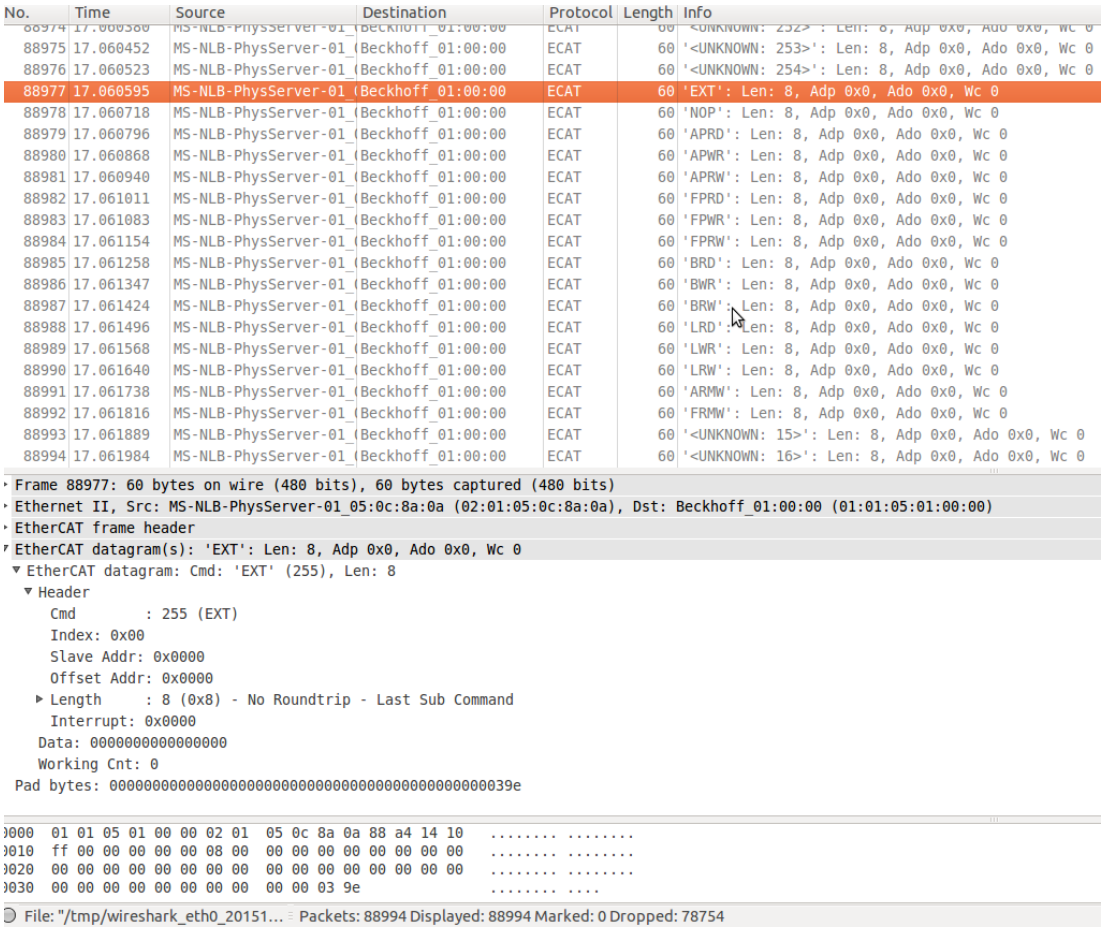
Paket içerisinde bulanıklaştırılacak olan alanlar belirli bir yapı ile belirlenmektedir. Veri alanı değişkenlik göstermektedir. Bir Ethernet paketinin minimum boyutu 60

baytdır. Bunun ilk 14 baytlık kısmı Ethernet başlık yapısıdır. Veri alanı 46 bayttır. EtherCAT protokolü Ethernet üzerinde taşındığından, en az 46 bayt kadar alan üzerinde taşınmalıdır. Paket içerisinde tek bir datagram olsa dahi, bunun 12 baytı başlık bölümü (EtherCAT ve datagram başlıkları toplamı), 2 baytlık bölümü ise çalışma sayacı alanıdır. Bunların dışında kalan 32 bayt uzunluğundaki alan veri taşınması içindir. Gönderilen veri bu değerden küçükse minimum paket uzunluğunu tamamlamak adına kalan kısımlara 0x00 değeri yazılarak bulanıklaştırma gerçekleştirilmektedir. 60 bayttan büyük veriler için bu işlem uygulanmamaktadır. Veri alanı içerisinde bulanıklaştırma yapılacak olan bitler yakalanan paket içerisindeki datagram başlığı içinde yer alan uzunluk alanına göre tespit edilmektedir. Datagram içerisinde ne kadar veri bulunuyorsa bulanıklaştırma sürecine tabi tutulmakta, veri alanı içinde kalan bitler ise paketin 60 bayt değerinden küçük veya büyük olma durumuna göre 0x00 değeri ile doldurulmaktadır.

Bulanıklaştırma yapılacak olan değerler tespit edildikten sonra paketin manipülasyonu en sondan başlayarak yapılmaktadır. Burada amaç, sonuç alma ihtimali yüksek bitlerin ilk değiştirilmesidir. İterasyon sayısı bulanıklaştırılacak bayt sayısının bite dönüştürülmesi ve üssel olarak hesaplanması şeklinde bulunmaktadır (1 baytlık alan için 2^8 adet iterasyon yapılacaktır). Herhangi bir yaklaşım uygulanmazsa, çalıştırma sonunda iterasyon sayısı kadar paket yollanmış ve tüm ihtimaller test edilmiş olmaktadır (Şekil 3.5.).

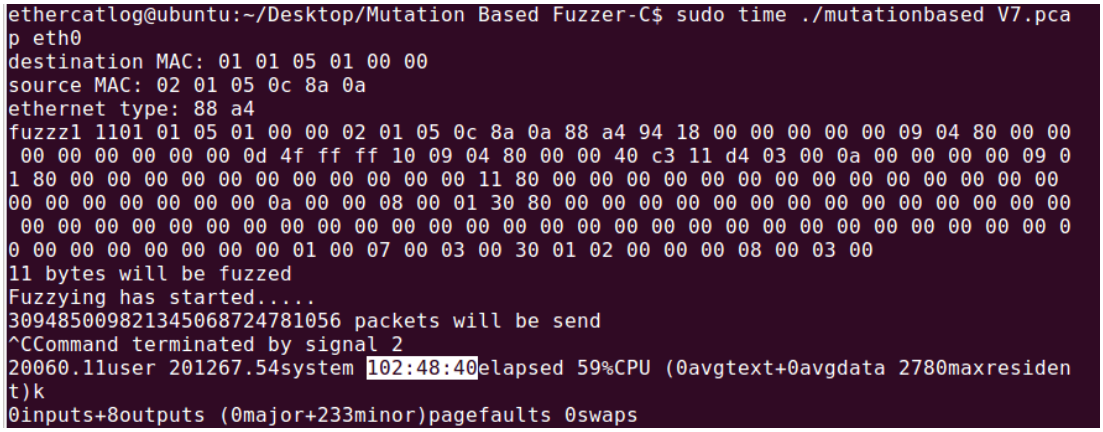
Gönderilecek paketlerin ağ içinde belirgin olması için bulanıklaştırılan paketlerin “reserve” alanı içine paket numarası eklenmiştir. Böylece bulanıklaştırma yapılan paketler ayrıştırılabilir hale gelmiştir. Ayrıca gelen paketleri analiz edecek bir ajan program yazılmıştır. Ajan program gelen paketleri incelemek ve belirli alanların değişmesi durumunda sonuçları veritabanına yazmakla görevlidir. Bu anlamda gelen paketlerden çıkarım yapılabilecek birkaç alana yoğunlaşılmış olup bunlardan bazıları çalışma sayacı (WKC) ve AL kaydedici durum alanlarıdır. WKC alanı giden paketler için 0 olduğundan, cevap paketlerinde değerin değişmesi halinde, paketin köleler tarafından kabul edilmesi ve işlenip işlenmediğinin belirlenmesinde kullanılmaktadır. İkinci alan ise AL durumunu tutan kaydedicinin değerini paketin sonuna ekleyen bir

yapıdır. Örneğin AL durumu “0x0008”, sistemin OP durumunda olduğunu, “0x001c” ise sistemin hata durumunda olduğunu göstermektedir.



Şekil 3.5. Bulanıklaştırıcı ile üretilen paketlerin yakalandığı Wireshark ekran görüntüsü

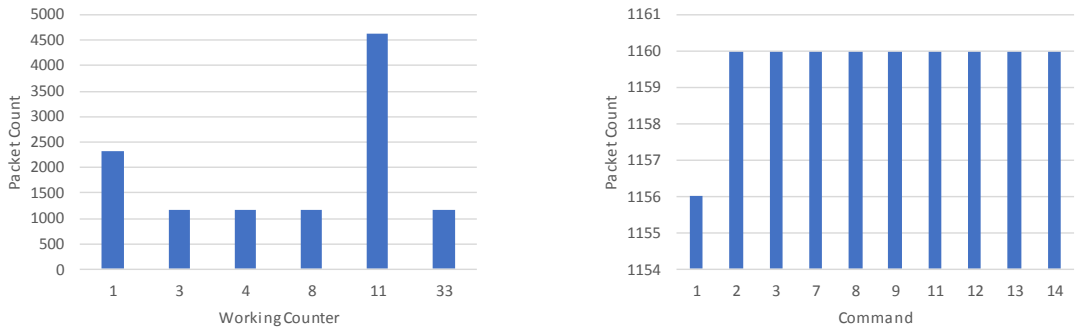
1. Durum: Bulanıklaştırıcı test ortamında ilk olarak 102 saat 48 dakika ve 40 saniye süresince çalıştırılmış olup toplamda 335 milyon paket üretmesi sağlanmıştır (Şekil 3.6.). Bu paketler veri tabanına kaydedilip incelenmiştir. Yapılan incelemede üretilen bazı paketlerin kabul edildiği ve gelen paketlerin WKC değerlerinin 1, 3, 11, 33 olarak değiştiği gözlemlenmiştir. Ayrıca gelen paketlerde IRQ alanı 4 olarak değişmiştir. Bu değer köle istasyon içindeki Data Link (DL) durumunun değiştiğini belirtmektedir.



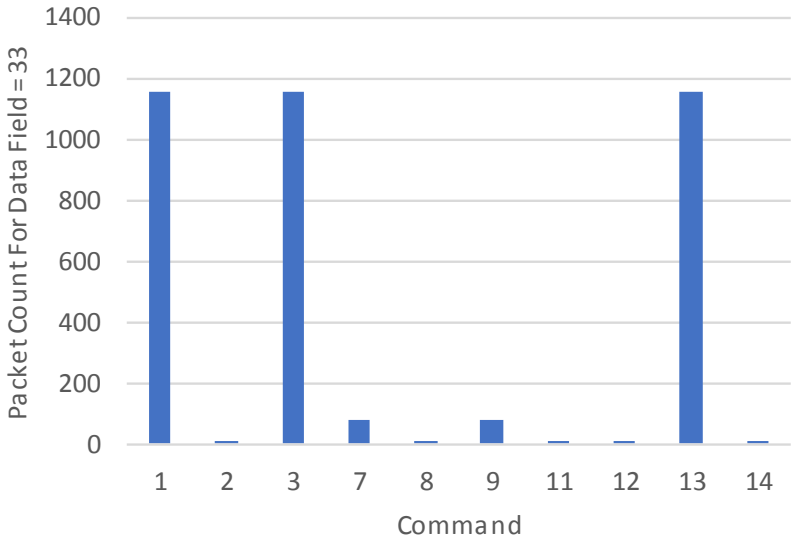
Şekil 3.6. Bulanıklaştırıcı testi

2. Durum: Bu testte paketin sadece adres, veri ve komut gibi alanlarına kabul olasılığı yüksek değerler yerleştirilerek, toplamda 209.945 paket gönderilmiştir. Gelen cevap paketlerinden WKC, IRQ, CMD, Data ve AL durum kaydedicisi durumları analiz edilmiştir. Gönderilen paketlerden 11.596 adet paketin kabul edildiği gözlemlenmiştir. Bu paketlerden 11.592 paket normal iletişim, 4 paket ise kesme (interrupt request by slave-IRQ) olarak kabul görmüştür. Bunun yanında AL durum kaydedicisi incelenmiştir. PLC-I/O bağlantısı devam ediyorsa kaydediciden 0x0008 (OP), etmiyorsa 0x1c00 yani hata durumu döndüğü görülmüştür. Şekil 3.7.' de gelen paketlerde görülen farklı WKC değerleri ve bunlardan kaç adet geldiği görülmektedir. Benzer şekilde komut ve paket sayısı grafiğinde görüldüğü üzere NOP, FPRD, FPWR, FPRW ve LRD komutlu paketler kabul edilmemiştir. Şekil 3.8.' de ise kabul gören paketlerin veri alanı 33 olanlarının hangi komutları işledikleri görülmektedir. Buna göre, köleler en çok APRD, APRW ve ARMW komutlarını kabul etmiştir. Şekil 3.9. kabul edilen paketlerdeki veri farklılıklarına göre WKC sayılarındaki değişimi göstermektedir. Veri alanı 33 olduğunda 3661 adetle en fazla sayıda paket kabul edilmiştir.

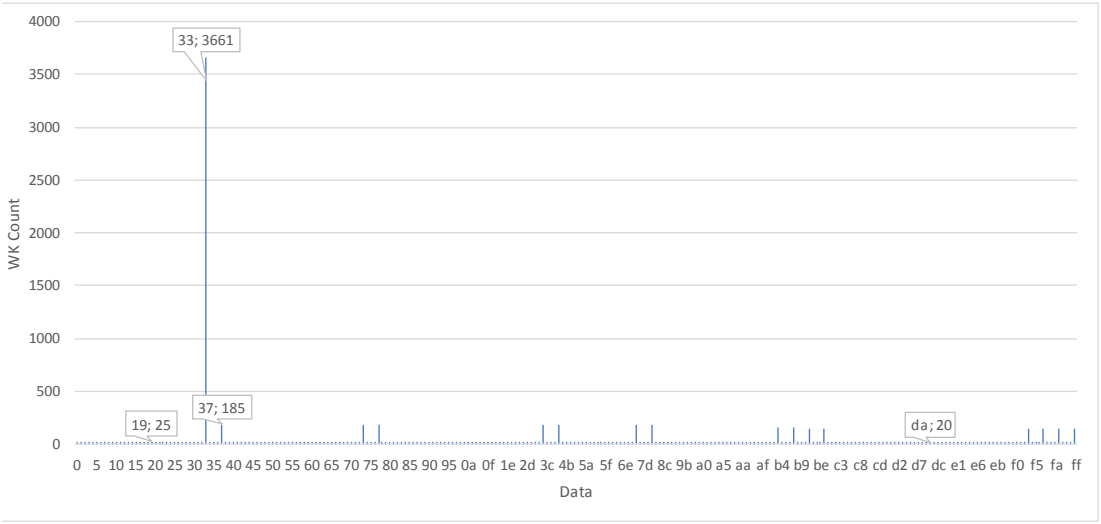
Şekil 3.10. komut paket sayısı ve veri değişimlerini 3 boyutta göstermektedir. Buna göre komut değişimlerinde ve veri alanlarında benzer bir örüntü oluşmaktadır. Bu durum, kölelerin adreslenmesinin kabul ettikleri komut türlerine olan etkisini göstermektedir. Veri alanı içinde köle mantıksal adresleri yer aldığından, var olmayan köleler için istek paketleri kabul edilmemiştir.



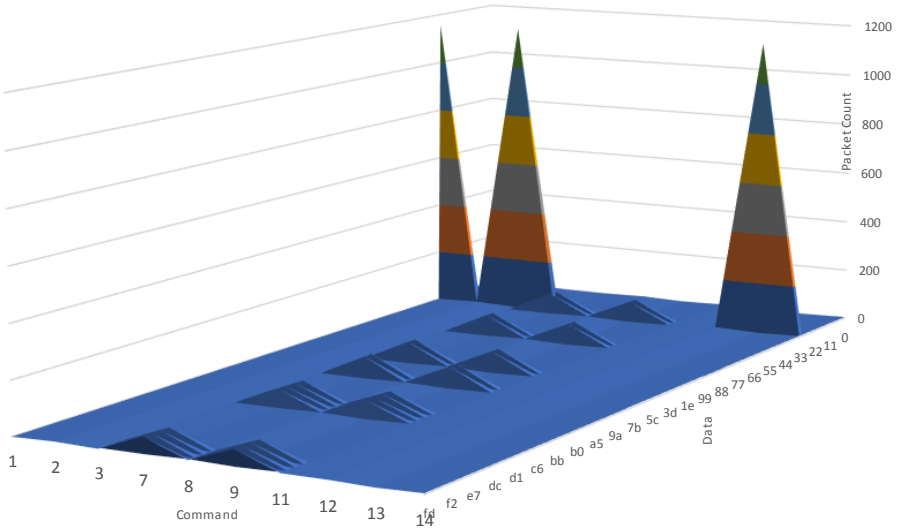
Şekil 3.7. Paket sayısı - WKC grafiği ve Paket sayısı - CMD grafiği



Şekil 3.8. Komut - veri alanı:33 grafiği



Şekil 3.9. Veri - WKC sayısı grafiği



Şekil 3.10. Komut-veri-paket sayısı grafiği

Karşılaşılan problemler:

a. EtherCAT spesifikasyonu içerisinde belirtildiği üzere bazı alanlar 11 bit gibi 1 baytın katlarından farklı alanlardır. Burada bitwise (bit seviyesinde operasyonlar) işlemler gerekmektedir.
b. Ağdaki bitlerin işlemci tarafından saklanma durumların önemlidir. Burada küçük sonlu veya büyük sonlu olmasına dikkat edilmesi gerekmektedir.
c. Bulanıklaştırılacak alanlar tespit edildikten sonra bu alanların tüm ihtimallerinin denenmesi istenebilir. Bu durumda tüm ihtimallerin hesaplanması ve her iterasyonda azaltılması/artırılması gerekmektedir. 4 bayt uzunluğundaki bir alan için toplam ihtimal $2^{(4*8)}$= 4,294,967,296 olduğu düşünülürse "math.h" kütüphanesi gibi standart yığınların bulanıklaştırma hesaplamaları için yeterli olmadığı tespit edilmiştir. Bu problemi aşmak için uygulamaya ait sayısal aritmetik fonksiyonları geliştirilmiştir.

Bulanıklaştırma çalışmasında, çözümün efektif olmaması görülerek ikinci bir yöntem kullanılmış, saldırı vektörleri geliştirilerek zafiyet analizi yapılmıştır. Sonraki bölümde bu yaklaşım detaylandırılmaktadır.

3.2.2. Saldırı Vektörü Oluşturulması

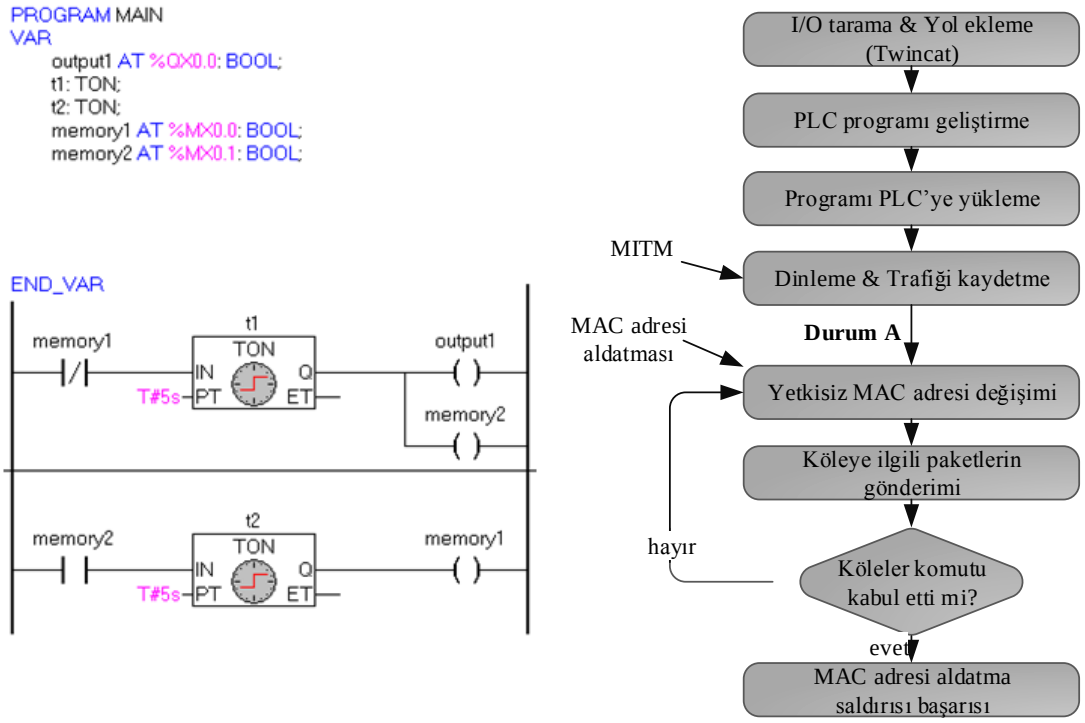
Literatürde zafiyet, tehdite karşı korunmayı veya yeniden kararlı durumuna geçilmesini önleyen/azaltan hata veya zayıflıklar olarak tanımlanmaktadır [132]. Benzer şekilde EtherCAT zafiyetleri de çerçeve yapısı, seviye içi/arası iletişim veya diğer EKS protokolleri gibi şifreleme, kimlik doğrulama veya yetkilendirme mekanizmalarının olmayışından kaynaklanan protokol tabanlı zayıflıklardır. EtherCAT protokolü iletişimi esnasında gözlemlenen başlıca zafiyetler, protokolün yetkilendirme, şifreleme ve kimlik doğrulaması olmadan saha, fabrika ve IP üzerinden dış dünya ile iletişimi gerçekleştirmesinden kaynaklanmaktadır. Bu nitelikler EtherCAT dışında daha güvenli olarak tanımlanabilecek birçok protokolde dahi tam olarak sağlanamamaktadır. Eksiklikler neticesinde güvenli iletişimin temel bileşenleri olarak kabul edilen CIA; gizlilik (confidenciality), bütünlük (integrity), erişilebilirlik (availability) kavramlarını zayıflatıcı, çeşitli saldırılar gerçekleştirilebilmektedir [49]. Sakarya Üniversitesi-Siber Güvenlik Laboratuvarı bünyesinde yer alan donanımlar kullanılarak oluşturulan test ortamı üzerinde literatürde yer alan zafiyetlerin bazıları atak vektörü halinde uygulanmış olup, açıklıkların bir kısmı tespit edilmiştir. Başarı elde edilen saldırı vektörleri aşağıda açıklanmaktadır.

MAC aldatma saldırısı: Saldırı, paketlerdeki güvenli efendi adresinin yetkisiz bir efendi MAC adresi ile değiştirilmesiyle yapılmış olup 4 aşamada uygulanmıştır.

a. PLC programı geliştirilmesi ve iletişim paketlerinin yakalanması
b. Paketlerin analiz edilmesi
c. MAC adreslerinin manipüle edilmesi
d. Saldırının gerçekleştirilmesi

İlk aşamada test ortamında bir PLC programı geliştirilmiştir. Program TwinCAT yüklü bilgisayarda yazılmış olup, çıkış birimindeki LED'i 5 saniye yakıp, 5 saniye söndürmektedir (Şekil 3.11.(a)). Program çıkışları EL2008 terminalindeki EK1101 modülünde yer alan ilk çıkışa bağlanmıştır. PLC ve giriş/çıkış (I/O) birimleri arasındaki trafik MITM tekniği kullanılarak bir ağ Tap cihazı yardımıyla alınmıştır.

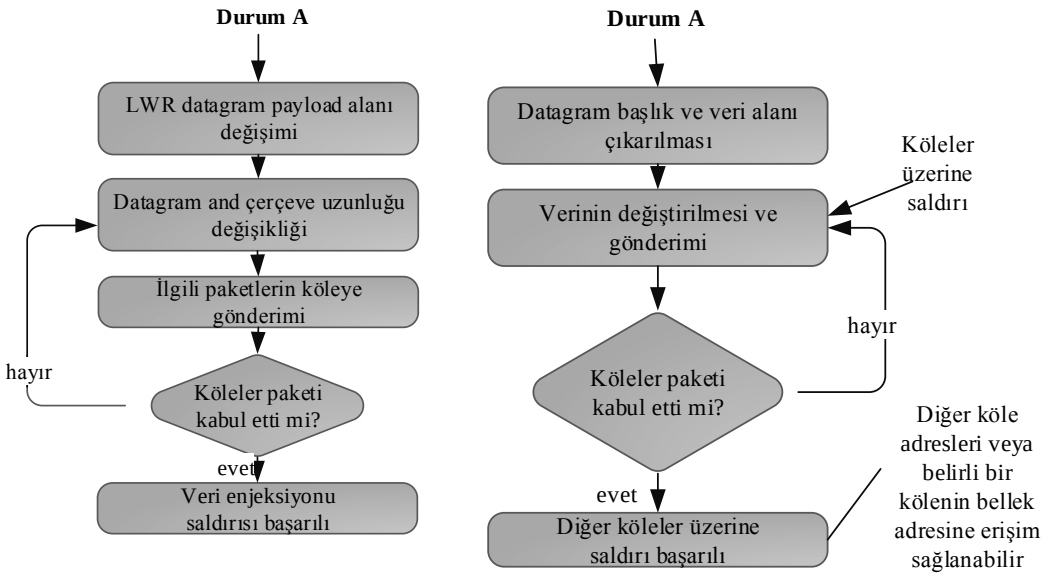
Ardından, trafik analiz edilmiştir. NOP, mantıksal yazma (LWR), mantıksal okuma (LRD), yayın komutu (Bxx) ve otomatik artırılan fiziksel okuma çoklu yazma (ARMW) komutlarının senkron olarak gönderildiği gözlenmiştir. Çıkışı ayarlayan komut LED sönük olsa bile her çevrimde senkron olarak iletilmektedir. Sönme durumu için "0", yanma durumu için "1" yollanmaktadır. Bunun yanında az miktarda Link Layer Discovery Protokol (LLDP) ve Multicast Domain Name System (MDNS) paketlerinin de gönderildiği tespit edilmiştir. MAC aldatma saldırısı için paketler öncelikle Wireshark paket koklama programından K12 dosya formatında çıkarılmıştır. Sonrasında her paketin MAC adres alanı yetkisiz bir adres ile değiştirilmiştir (Şekil 3.11.(b)). PLC'nin gücü kesilmediği takdirde kölelerin gelen paketleri kabul ettikleri görülmüştür. Bu durum gelen paketlerde yer alan WKC alanlarının artması ve



Şekil 3.11. (a) Geliştirilen PLC programı, (b) MAC aldatma saldırı akışı

LED'lerin yanmasıyla anlaşılmıştır. Giden paketlerdeki WKC değerleri 1 iken gelen paketlerde bu değer 5 olmuştur. Ayrıca, efendi PLC herhangi bir hata bildirimi yapmadığı için sistem yöneticisinin saldırıyı anlamasının zor olduğu görülmektedir. Saldırının başarısının sebebi efendi ve köleler arasında sadece temel seviye bir kimlik

doğrulamanın olması, bunun ise konfigürasyon ve cihaz tarama aşamalarında gerçekleşmesidir.



Şekil 3.12. (a) Veri enjeksiyonu saldırı akışı, (b) Köle istasyonlar üzerine saldırı akışı

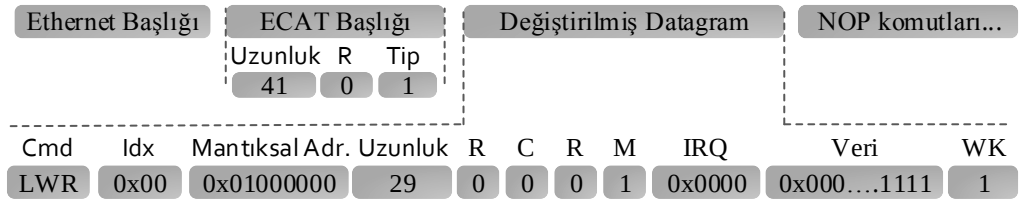
PLC gücü kesintiye uğradığında, I/O birim konfigüsrasyonları silineceğinden saldırı başarısız olmaktadır. Bu yüzden, TwinCAT tarafından öncelikle bir ağ taraması yapılmış, PLC ve I/O arasında bir ağ yolu atanmış olmalıdır. Ayrıca, sistem konfigürasyonları yapılmış ve PLC çalışma durumunda olmalıdır.

Veri enjeksiyonu saldırısı: Saldırı öncelikle sistemden paketlerin MITM yöntemiyle alınmasıyla oluşturulmuştur (Şekil 3.12.(a)). LWR komutları çıkışlara yazmaktan sorumlu olduğundan, LWR komutu olan datagramların veri alanları değiştirilmiştir. Veri alanları 16 bayt olan bu datagramlar 20 bayt olarak uzatılmıştır. Sonrasında, datagram uzunluk alanı 20, çerçeve uzunluk alanı ise 150 olarak güncellenmiştir. Kimlik doğrulama veya bütünlük kontrolü olmadığından kölelerin paketleri kabul ettikleri gözlemlenmiştir. Bu tarz saldırılar ağ durumuna göre gerçek zamanlılık kapasitesini bozabilmekte ve hatta fiziksel hasarlara sebep olabilmektedir.

DoS saldırısı: Burada yakalanan trafik içerisinden seçilen bir datagramın her çevrimde sürekli olarak belirli bir adrese gönderilmesi sağlanmıştır. Aynı zamanda TwinCAT

programından PLC ile bağlantı kurulmuş ve PLC'nin içinde yer aldığı ağı taraması, etrafındaki aygıtlardan bilgi alması sağlanmıştır. Aygıt tarama aşamasının gerçekleşemeden programın cevap vermediği, mühendislik istasyonu olan bilgisayarın mavi ekran hatası verdiği gözlemlenmiştir. Saldırgan tarafından I/O birimlerine gönderilen paketlerin köleler tarafından değerlendirilmesi sonucunda PLC'ye geri gelmesi ve PLC'nin bunlara yanıt vermeye çalışması, TwinCAT tarafından istenen tarama işlemini yerine getirememesi ile sonuçlanmıştır.

Köle istasyonlar üzerine saldırı: İletişim esnasında elde edilen paket örüntüsü incelendiğinde, Mantıksal Yazma (LRW) komutunun yer aldığı bir datagramın, köle I/O birimi (EL 2008) üzerindeki bir adresteki LED ışığını yakmak için kullanıldığı görülmüştür. Buradan hareketle ilgili datagram paket içerisinden ayrıştırılarak adresleme yapısı tespit edilmeye çalışılmıştır (Şekil 3.12.(b)). Aynı modül (EK 1101) üzerindeki farklı I/O birimlerinin aynı mantıksal adresi paylaştığı gözlemlenmiştir. Buradan hareketle LWR komutu içeren ve orijinal datagram ile aynı boyutta (29 bayt) paketler C programlama dili kullanılarak geliştirilen bir yazılımla üretilmiştir. Minimum çerçeve uzunluğunu (60 bayt) sağlamak için sona NOP komutu içeren datagramlar eklenmiştir (Şekil 3.13.). LWR komut alanının son 2 baylık kısmı 255 olarak güncellenerek ağa yollanmıştır ve I/O birimlerin son 16 çıkışının enerjilendiği görülmüştür.



Şekil 3.13. LWR komutlu değiştirilmiş datagram örneği

Bu çalışmalar neticesinde EtherCAT protokolü içinde:

a. Efendi istasyonlarla köleler ile arasında herhangi bir kimlik doğrulama mekanizmasının bulunmadığı, sadece konfigürasyon ve aygıt tarama esnasında temel düzeyde tanıma gerçekleştirdiği,

b. Köle istasyonların aptal (dummy) cihazlar olarak, üzerlerindeki adreslere erişim izni verdikleri ve efendileri ile her akış için efendi doğrulama yapmadıkları, bellekleri üzerindeki adres haritası doğru tespit edildiği takdirde ilgili adrese gelen paketlere cevap verdikleri,
c. Hem PLC, hem de köle tarafında DoS atağı gibi saldırılara karşı herhangi bir güvenlik mekanizmasının bulunmayışı nedeniyle doğru oluşturulmuş paketlere cevap verilmeye çalışılması, gibi eksiklikler tespit edilmiştir.

BÖLÜM 4. ETHERCAT ÖNİŞLEMCİSİ GELİŞTİRİLMESİ VE GÜVENLİ DÜĞÜM YAKLAŞIMI

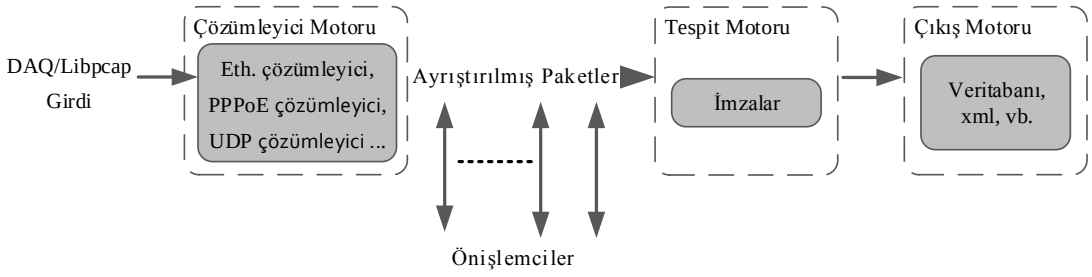
Bu bölümünde olası saldırıları tespit etmek, önlemek ve azaltmak adına açık kaynak kodlu Snort saldırı tespit ve önleme sistemi (IDS/IPS) üzerine protokolü desteklemesi için geliştirme yapılmış, bulunan açıklıklardan hareketle EtherCAT tabanlı önişlemci geliştirilmiştir. Bu önişlemciye güvenli düğüm yaklaşımı olarak adlandırılan yeni bir yöntem eklenmiş ve önişlemci üzerine 3 adet kural tanımlanmıştır. Önişlemcinin, geleneksel kural tanımlarına ek olarak katmanlı mimaride 2. katmanı desteklemesi ve güvenli düğüm yaklaşımı yoluyla derin paket analizine dayalı çözüm sunması, literatürde daha önce tanımlanan Modbus/TCP veya DNP3 önişlemcilerinden farkını göstermektedir. Güvenli düğüm yaklaşımı ise öncelikle mühendislik istasyonu tarafından tanınan düğümleri EtherCAT ağ bilgisi (ENI) konfigürasyon dosyasından almakta, ardından ağdan gelen paketleri protokol kabullerine uygun olarak ayrıştırmaktadır. Bu bölümde sunulan çözümler pasif izleme yoluyla iletişimi kesmeden ve sisteme ekstra yük getirmeden saldırıları tespit etmektedir. Geliştirilen yöntem ve uygulamalar gerçek donanımlarla oluşturulan bir test ortamında test edilmiş, çalışmanın ilgili saldırıları tespit ettiği ve EtherCAT tabanlı sistemlerde temel düzeyde bir güvenlik sağladığı görülmüştür.

4.1. SNORT Üzerine EtherCAT Önişlemcisi Geliştirilmesi

Yapılan zafiyet tespiti çalışmasında görüldüğü üzere EtherCAT tabanlı sistemlerde protokolden kaynaklı zayıflıklar mevcuttur ve risk barındıran bu sistemlerin sağlamlaştırılması gereklidir. Bu kapsamda, Snort açık kaynak kodlu IDS/IPS altyapısında önişlemci geliştirilmiş ve EtherCAT protokolü üzerinden gelecek saldırıların Snort yardımıyla tespit edilmesi amaçlanmıştır. Bir sonraki bölümde Snort yapısı ve önişlemci önerisi detaylandırılmaktadır.

4.1.1. Snort yapısı

Snort programı açık kaynak kodlu bir yazılım olup, ağ içerisindeki anomalileri tespit etmek ve önlemek için kullanılan saldırı önleme/tespit (IDS/IPS) sistemidir. İmza tabanlı olarak çalışmaktadır. İstenirse bir makineya, istenirse ise PfSense gibi bir güvenlik duvarı üzerine derin paket incelemesi için kurulabilir. Gelen trafiğin Snort üzerinden geçmesi sağlanarak var olan ağa dâhil edilir. İstenen kurallar (imzalar) kural veritabanına Oinkcode aracılığıyla eklenir ve bu kurallarla eşleşen trafik/akış/paketler günlükleme, engelleme, paket düşürme gibi aksiyonlarla işleme alınır. Snort kuralları ilgili veritabanındaki hali hazırda bulunan kurallarla çalıştırılabileceği gibi, manuel olarak da tanımlanabilmektedir. Libpcap kütüphanesi yardımıyla paketleri tanıyabilen Snort yazılımı, gelen paketleri öncelikle çözümleyici modülüne sokarak 2, 3 ve 4. katmanlardaki başlık yapılarını tanıyabilmektedir. Sonrasında, "Snort.conf" dosyasında varsayılan olarak gelen önişlemcilerden hangileri aktif edildiyse paket ilgili olana yönlendirilir. Burada da kural tabanlı eşleştirme yapılabilir. Son olarak ilgili önişlemci bulunuyorsa önişlemciye, yoksa 4. katman protokolleri seviyesinde kurallar ve diğer seçenekleri tespit motoru tarafından işleme alınır. Son olarak veritabanına, Syslog, XML veya farklı formatlarda, alarm, günlükleme, düşürme, engelleme gibi aksiyonlar işlenerek kaydedilir.



Şekil 4.1. Snort içindeki paket akışı

Şekil 4.1. Snort yapısına yönlendirilen bir Ethernet paketinin işlem akışını göstermektedir. Snort programının ilk sürümleri sadece OSI 3-4 katmanlarına kadar ayrıştırabilirken, yeni versiyonu uygulama katmanı protokollerini de tanıyabilmektedir. Snort'a gelen paketler ilk olarak Ethernet çerçevesinden

ayrıştırılmaktadır. Bunun sonunda 2. katmanda yer alan üst katman protokol tip alanına bakılarak bir sonraki protokol hakkında ön ayrıştırma işlemi de yapılır. Kalan paket TCP/UDP, ARP, VLAN, PPP gibi bir sonraki protokol ayrıştırması için ilgili modüle teslim edilir. Son olarak taşıma katmanında yer alan protokoller yardımıyla paket içindeki verilere erişilir. Buradaki önemli nokta, sadece belirtilen seviyelerde yer alan protokoller üzerinden taşınan veriler, varsayılan olarak işlenebilmektedir. Örneğin taşıma katmanından sonra yer alan veya uygulama katmanı protokolleri üzerinde bir analiz yapılacaksa, bu protokollerin ayrıştırıldığı ek önişlemciler gerekmektedir. Bu önişlemciler üzerinde kural aksiyonları alarm ver, geçir, düşür, günlüklemeden düşür, reddet olarak tanımlanabilmektedir. Önişlemcilerin bir kısmı ilk versiyonlarla varsayılan olarak gelmekte olup, bir kısmı ise Snort 2.6 versiyonunda dinamik olarak çalışma zamanında yüklenebilmektedir. Bunlar dinamik önişlemciler olarak adlandırılmaktadır. DNS, ARPSpoof, FTP/Telnet, SSH gibi önişlemciler olduğu gibi endüstriyel otomasyon sistemlerinde kullanılan DNP3, Modbus protokolleri için de önişlemciler mevcuttur. Bu önişlemcilerin sınırlı sayıda kuralları bulunmaktadır. Örneğin Modbus önişlemcisi sadece 3 kural barındırmakta olup bunlar fonksiyon kodu veya protokol ID alanı kontrolü gibi basit paket analizine dayalıdır. Ayrıca, bu önişlemcilerin ortak özelliği TCP/IP üzerinden gelen paketlerde tanımlı olmalarıdır. Bunun sebebi Snort sisteminin gelen paketleri iletim katmanına kadar çözümleyebilmesidir. Sonrasında önişlemci kullanılacaksa paket önişlemciye devredilir. Yani önişlemciler iletim katmanından önceki paket alanlarına erişim sağlayabilir. Fakat EtherCAT saha ve EAP iletişiminde kendi çerçeve yapısına sahip olup veri bağı katmanı üzerinde iletim katmanına sahip değildir. Bu da EtherCAT önişlemcisi (ECAT önişlemcisi) geliştirilmesini güçleştirmektedir. Bir sonraki bölümde bu probleme çözüm olarak sunulan Snort çözümleyici modülü ve geliştirilen önişlemci anlatılmaktadır.

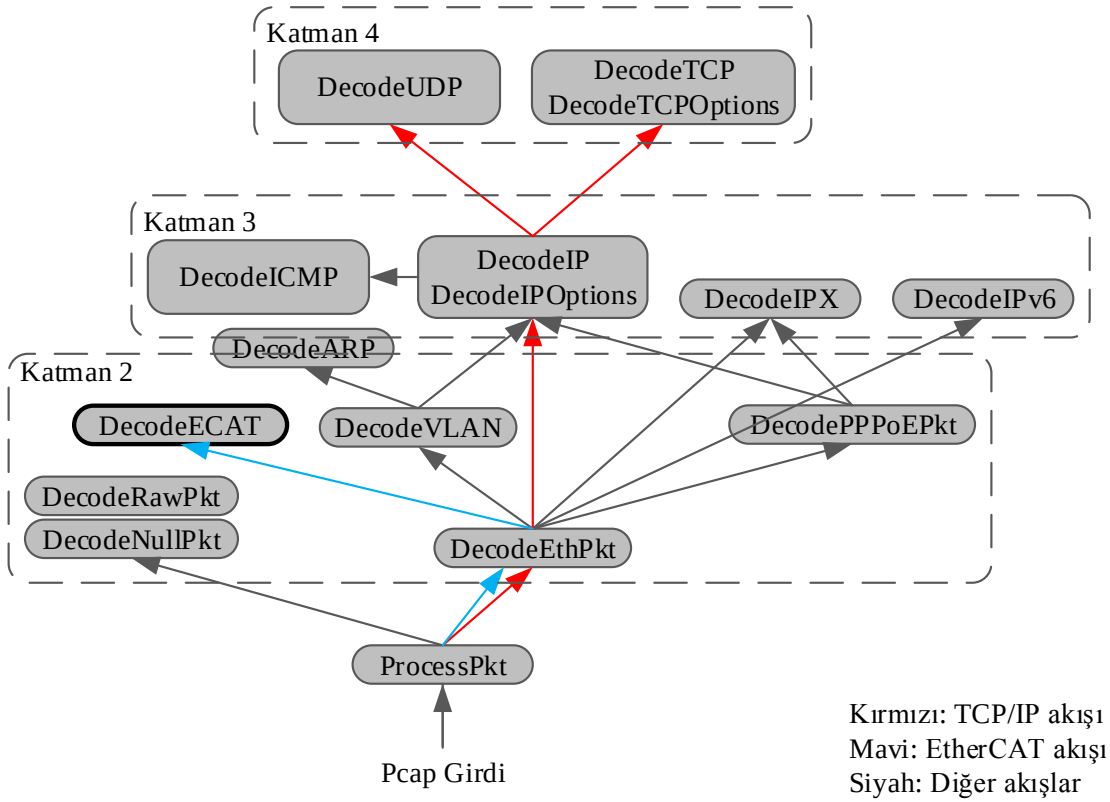
4.1.2. EtherCAT önişlemcisi geliştirilmesi

Bu bölümde, Snort sistemine gelen EtherCAT paketlerinin ayrıştırılması için gerekli olan çözümleyici geliştirilmesi, paketleri protokol çerçeve yapısına uygun olarak ayrıştıran ve istatistiklerini çıkaran önişlemci geliştirilmesi, başlangıçta onaylanan

konfigüsrasyonda yer almayan düğüm ve bilgileri tespit eden güvenli düğüm yaklaşımının geliştirilmesi ve tüm bu çözümlerin kurulan bir izleme sistemi üzerinden kullanıcıyla paylaşılması aktarılmaktadır.

4.1.2.1. EtherCAT çözümleyici

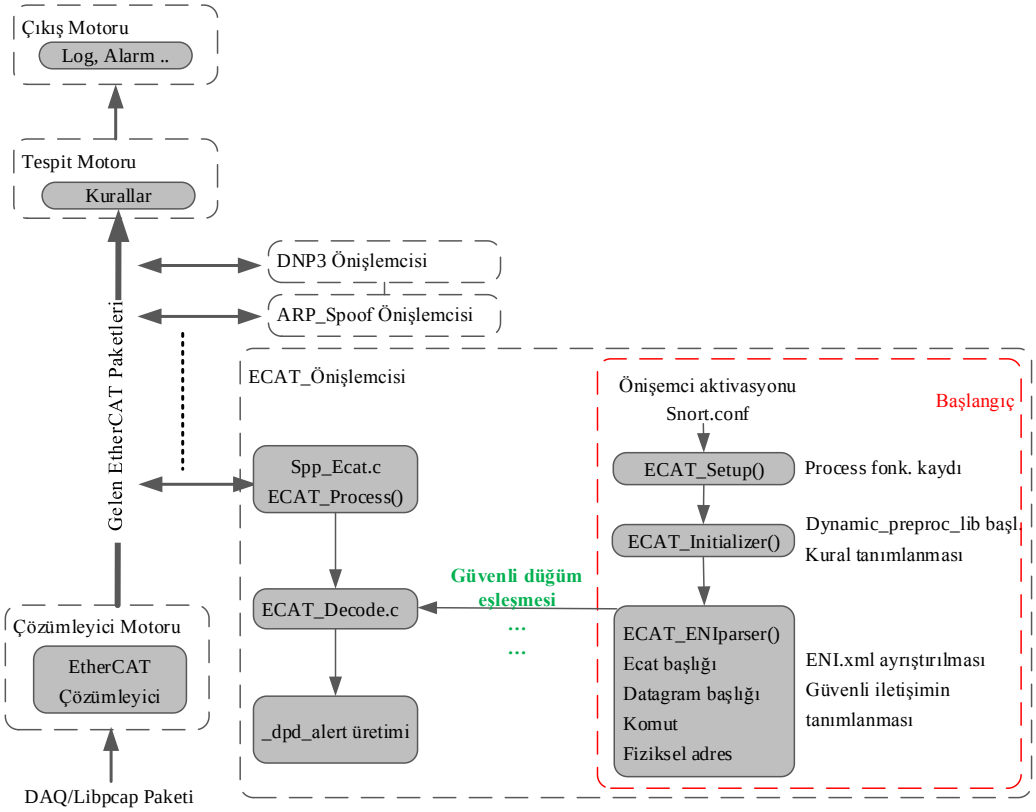
Görüldüğü üzere EtherCAT protokolünü ayrıştıran bir önişlemci Snort tarafından önerilmemiştir. Dolayısıyla EtherCAT paketleri için herhangi bir imza tanımlanamamaktadır. Bu kapsamda EtherCAT paketleri üzerinde aksiyon yapmaya ve kural tanımlamaya imkân veren bir önişlemci geliştirilmiştir. Bilindiği üzere EtherCAT iletişimi fabrika ve saha seviyelerinde IP adresi barındırmamaktadır. Bu yüzden TCP/IP kullanılmadan Ethernet üzerinden iletişim sağlayan endüstriyel otomasyon sistem protokolleri, önişlemci olarak tanımlansalar bile Şekil 4.2. ’de belirtilen ayrıştırma yapısında bulunmadıklarından, geliştirilememektedirler. Daha önce geliştirilen DNP3 ve Modbus/TCP önişlemcileri, paketleri Şekil 4.2.’de yer alan kırmızı okları takip ederek ayrıştırılmaktadır. Yani, gelen paketler 4. katman ayrıştırması bittikten sonra önişlemciye teslim edilir. Benzer protokollerin önişlemcileri başlangıçta TCP ya da UDP protokol modüllerine atanarak geliştirilirler. EtherCAT bu şekilde çalışmadığından, öncelikle EtherCAT çözümleyicisi geliştirilmiş, ardından kalan paket ECAT önişlemcisine teslim edilmiştir. Snort sistemine gelen EtherCAT paketleri mavi ok işaretli yolu takip etmektedir. EtherCAT çözümleyicisi, Ethernet çözümleyicisinden gelen paketi almakta, ileride istatistikler, önişlemci veya kurallar tarafından kullanılacak EtherCAT başlık yapısı, datagram sayısı, veri alanı başlangıcı gibi değerleri tespit etmektedir. Daha sonra veri alanının ilk bitine konumlanılarak, paket ECAT önişlemcisine teslim edilmektedir. Snort sistemine böyle bir önişlemcinin kaydedilmesi için iletim katmanı değeri “none” olarak tanımlanmalıdır. Geliştirilen bu çözümleyici 2. katman üzerinden iletilen protokoller için sunulmuş bir çözümdür.



Şekil 4.2. Snort çözücüsü içindeki paket akışı

4.1.2.2. EtherCAT önişlemci

"Snort.conf" dosyasında aktif hale getirilmiş önişlemci, öncelikle "initilizer" fonksiyonu ile tanımlanır ve "setup" başlangıç fonksiyonu ile Snort yapısına kayıt ettirilmektedir. Bu 2 fonksiyon, Snort programının başlangıç konfigürasyonları yüklenirken çağırılmaktadır. Kayıt esnasında her paket gelişinde çağrılması istenen "process" fonksiyonu da belirtilmektedir. Bu fonksiyon, dinamik önişlemcilerin kurallarının yazıldığı kural dosyasını okuyup, önişlemcinin kimlik numarası ile örtüşen kuralları önişlemciye vermektedir. Burada kural dosyasında desteklenen ilgili önişlemci kurallarının tümü için birer adet fonksiyon tanımlanması gerekmektedir. İlgili fonksiyonlar çalıştıktan sonra işlenen paketlerde saldırı veya anomali tespit edilenler "_dpd" yapısı çağrılarak çeşitli günlükleme formatlarında kaydedilmektedir. Kurallar kontrol edildikten sonra ilgili günlükleme ve alarmlar için tespit modülü çağrılmaktadır. Tespit modülü dinamik önişlemci kural dosyalarından sorumludur. Önişlemcilerin kuralları yazıldıktan sonra kontrol sırasında eşleşme olması durumunda alarmlar ve kural sonuçları çıkış modülüne aktarılmaktadır (Şekil 4.3.).



Şekil 4.3. ECAT önişlemcisi

4.1.2.3. Güvenli düğüm yaklaşımı

Snort önişlemcileri IDS/IPS özelliklerini önceden tanımlanmış kuralları kontrol ederek sağlarlar. Her önişlemci eşsiz üretici ID (gid) ve her kural Snort ID (sid) numarasına sahiptir. Kural "sid" numaraları belirli bir önişlemcideki ilgili kuralı tanımlamaktadır. Önişlemci kuralları genel kural sözdizininden farklı olup şu şekildedir: kural_tipi_veya_aksiyon (yazdırılacak_mesaj; Snort_ID; üretici_ID; revizyon_numarası; üst_bilgi; sınıf_tipi;).

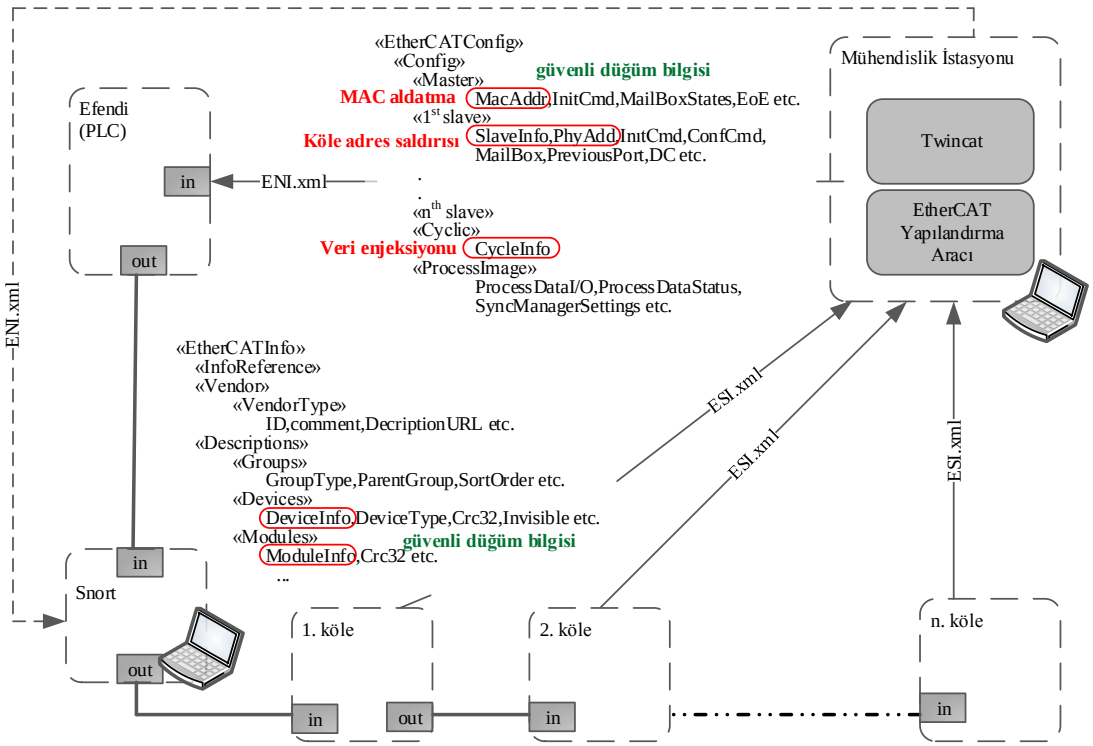
Hali hazırda var olan endüstriyel otomasyon sistemi protokol önişlemcileri, protokol ID, paket uzunluğu, tipi alanı kontrolü gibi basit kurallara sahiptir. Geliştirilen ECAT önişlemcisi ise derin paket analizine dayanmaktadır. Bunun için EtherCAT tabanlı sistemde yer alan düğümler tespit edilir ve bu düğümler güvenli düğüm olarak adlandırılır. Bu çözüm yapılan zafiyet analizinin bir çıktısı olarak geliştirilmiştir.

EtherCAT zafiyet analizinde görüldüğü üzere efendi ve köle istasyonlar arasında herhangi bir kimlik doğrulama bulunmamaktadır. Bu eksiklikten faydalanarak, farklı saldırılar geliştirilebilmektedir. Bu saldırılardan biri de sistem üzerindeki belirli sayıda paket yakalanıp bir köle tespit edildikten sonra, farklı kölelerin veya düğümlerin yer alıp almadığını kontrol etmek için farklı ve benzer adreslere paket yollama yöntemidir. Bu yöntemle ağ haritası, adresler ve düğümlerdeki konfigürasyon bilgileri tahmin edilebilir.

Bu şekildeki saldırıları önlemek için EtherCAT sistemine dâhil olan aygıtlar tespit edilmiş ve önişlemcinin başlangıç konfigürasyonu sırasında Snort yazılımına yüklenmiştir. Bu düğümler güvenli düğüm olarak adlandırılmış, bunlar haricindeki bir kaynak veya hedef adres içeren iletişimlerde alarm ve günlükleme kaydeden kural, 146 EtherCAT önişlemci ID numarası ile kayıt edilmiştir. EtherCAT tip alanı ile gelen her yeni paket, Ethernet ve EtherCAT başlık yapıları ayrıştırıldıktan sonra, önişlemciye verilmiş, burada paket içindeki her bir datagramın ait olduğu adres, komut, komut sırası, veri boyutu ve efendi cihazların MAC adreslerinin güvenli iletişim olarak tanımlanan adreslerle eşleşip eşleşmediği karşılaştırılmıştır. İmza tabanlı olan bu sistemde 3 kural geliştirilmiştir: güvensiz köle, güvensiz efendi ve veri enjeksiyonu. Tanımlanan kurallara göre güvenli iletişimde yer almayan adresler için alarm üretilmiş ve günlükleme dosyasına kaydedilmiştir. Bu kurallardan bir örnek aşağıda verilmiştir.

Alert (msg:“ECAT UNTRUSTED SLAVE”; sid:1; gid:146; rev: 1; metadata: rule-type preproc; classtype: protocol-commanddecode;).

Güvenli düğümlerin tespiti yaklaşımı EtherCAT saha seviyesindeki iletişim için sunulan bir çözümdür. Akışı şu şekildedir: TwinCAT üzerinde ağ taraması ve başlangıç konfigürasyonu yapılır, köleler tarafından gönderilen ESI dosyaları efendi istasyona bağlı olan EtherCAT konfigürasyon aracına yüklenir, oluşan ve XML tabanlı olan ENI dosyası Snort’un çalışma zamanında otomatik olarak ayrıştırılır. ESI dosyaları belirli bir köle için bilgileri tutmaktadır. ESI dosyaları ve/veya çevrimiçi köle bilgileri efendi tarafında birleştirilir ve ENI dosyası oluşturulur (Şekil 4.4.). Bu dosya,



Şekil 4.4. ESI-ENI dosyalarının işlenmesi

başlangıç konfigüsrasyonları, efendi, köle bilgileri ile çevrim ve işlem imaj bilgilerini içerir. EtherCAT (ECAT) önişlemcisi ENI dosyasını geliştirilen ayrıştırıcı yardımıyla çalışma zamanında işler. Bu işlem Snort açılıştaki ön yükleme aşamasında tamamlanarak efendi ve köle düğümleri ile veri boyutu, komutlar, hatta komut sıraları belirlenir. Bu bilgiler EtherCAT iletişimi esnasında mühendislik istasyonu tarafından onaylanmış bilgiler olup, bu özniteliklere uyan akışlar güvenli kabul edilmektedir. Neticede, çalışma zamanında ayrıştırıcı modülünden çıkan her paket önişlemciye gider ve dinamik olarak güvenli olup olmadığı kontrol edilir ve böylece temel düzey bir güvenlik mekanizması oluşturulmuş olur.

4.1.2.4. Önişlemcinin test edilmesi

ECAT önişlemcinin testi için bölüm içinde sunulan test ortamı kullanılmıştır. Efendi ve köleler arasına Snort yüklenmiş bir sanal makine konumlandırılmıştır. ECAT önişlemci kaydedilmesi ve başlangıç durumuna getirilmesi aşağıdaki Şekil 4.5.(a), (b), (c)'de gösterilmektedir. Önişlemci, TwinCAT yüklenmiş mühendislik istasyonundan

dışarı aktarılmış ENI.xml dosyasını okur ve erişilebilir güvenli düğümleri tespit eder. Sonra aktif edilmiş tüm önişlemcileri, versiyon ve revizyon sayıları ile yükler ve sisteme gelecek paketleri bekler. Bir EtherCAT paketi alındığında önişlemci EtherCAT ve datagram başlıkları gibi tüm datagramları ve başlıkları ayrıştırır. Ayrıştırılan paketlerden rastgele alınan bir örneklem Şekil 4.5.(d)' de gösterilmektedir.

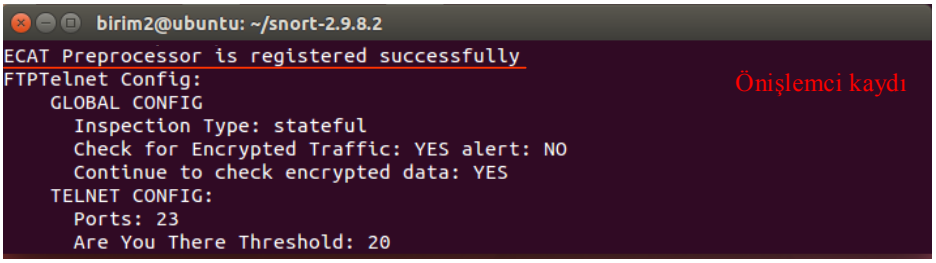
Köle istasyonlar üzerine yapılan saldırı için LWR komutu içeren datagamlardan oluşturulan EtherCAT paketi güvensiz bir düğüm adresi ile üretilmiş (adres: 257) ve ağa gönderilmiştir. Aşağıdaki Şekil 4.5.(e)'de bu pakete karşılık üretilen köle erişim alarmının tetiklenmesi gösterilmektedir.

MAC aldatma saldırısı için yetkisiz bir MAC adresi ile güvenli efendi adresi değiştirilmiş ve ağa gönderilmiştir. Oluşturulan alarm Şekil 4.5.(f)'de gösterilmektedir. ECAT önişlemcisinin tanımlanan saldırıları tespit edebildiği ve düzgün bir şekilde alarm dosyasına kaydettiği görülebilmektedir.

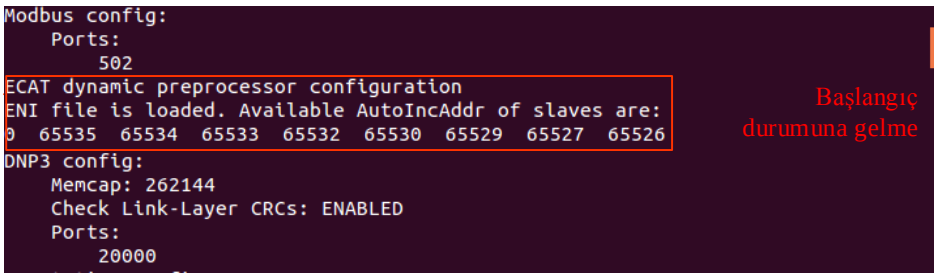
Veri enjeksiyonu saldırısı için ENI dosyasında yer alan komut isimleri, komut sıraları ve beklenen veri uzunluğu gibi bazı alanlar çekilmiştir. Bu bilgiler ENI içinde çevrimli veriler kategorisi altında yer almaktadır. Bilgilerden herhangi biri (örneğin komut sırası) eşleşmezse alarm üretilmektedir. Örneğin, test ortamından çekilen ENI dosyası analiz edildiğinde, ARMW komutunun paket içindeki 4. datagramda olması gerektiği ve 4 baytlık bir veri içermesi gerektiği görülmüştür. Test etmek amacıyla sıra ve veri uzunluğu değiştirilmiş datagramlar üretilmiştir. Komut sırasında, veri boyutunda veya veri alanında yapılan herhangi bir değişikliğin tespit edilmesi halinde alarm üretildiği gözlemlenmiştir (Şekil 4.5.(f)). Böylece daha önce tanımlanan saldırıların önişlemci tarafından yakalandığı görülmektedir.

Güvenli düğüm yaklaşımı, MAC aldatma, veri enjeksiyonu ve köle adresleri üzerine saldırılarını, onaylanmamış bileşenlere erişmek istedikleri takdirde kurallar yardımıyla tespit edebilmektedir. MAC aldatma saldırıları ENI dosyasındaki efendi donanımsal adresinin kontrolü ile yapılabilirken, veri enjeksiyonu beklenen her komut için veri

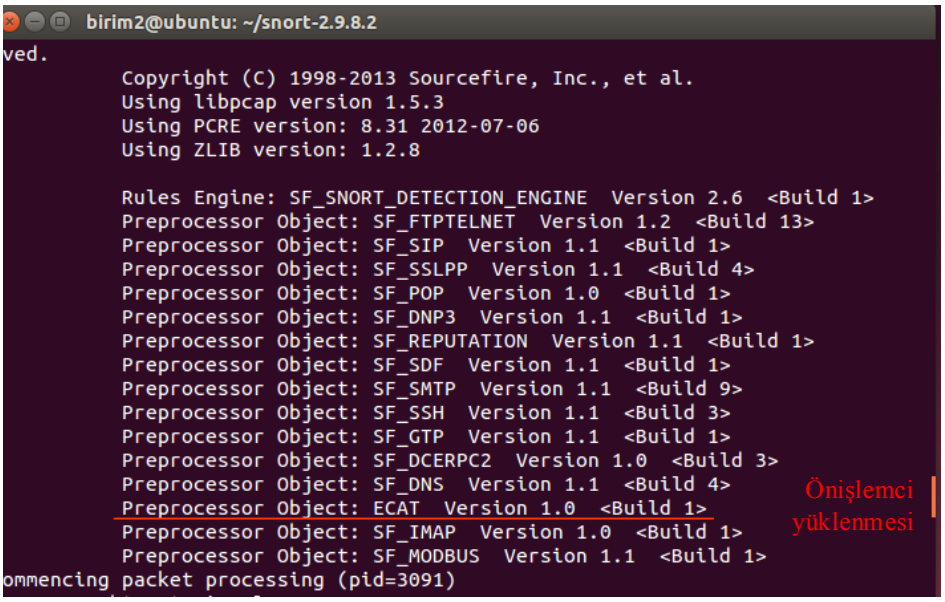
alanlarının kontrolü ile yapılabilmektedir. Köleler üzerine yapılacak olası saldırılar ise mantıksal, otomatik arttırılan veya fiziksel köle adreslerinin kontrolü ile yapılabilir.



(a)

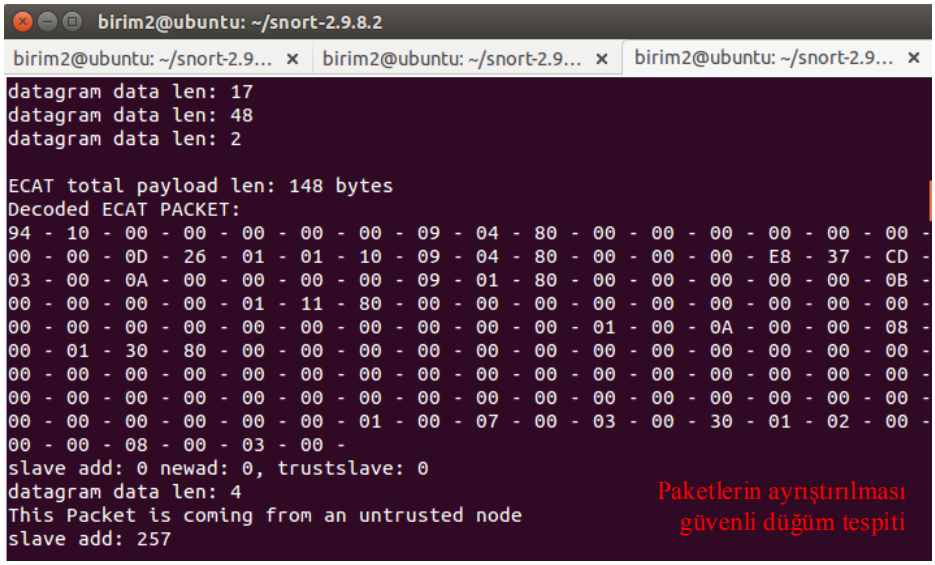


(b)

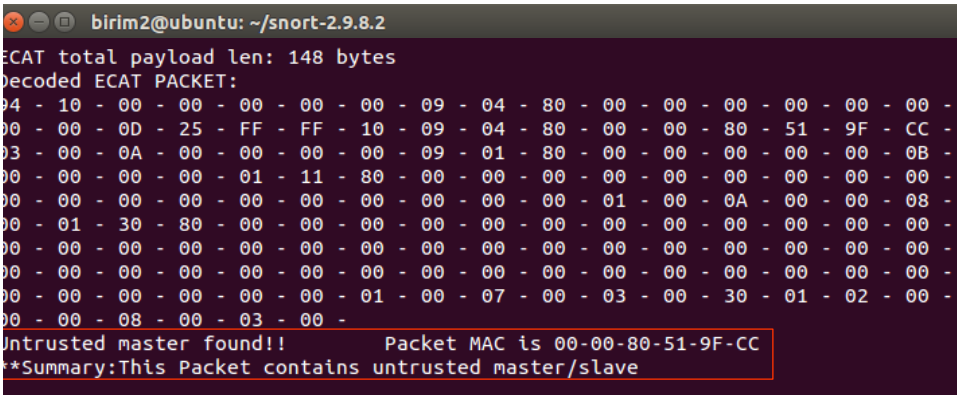


(c)

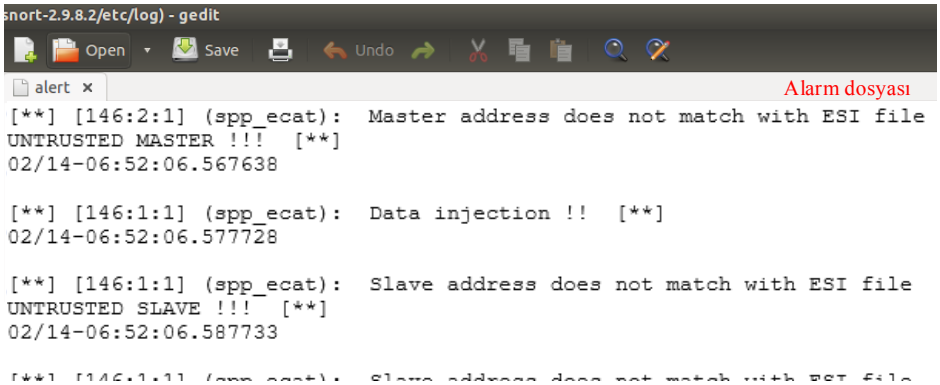
Şekil 4.5. ECAT önişlemci çalışması, sonuçlar ve günlüklemeler



(d)



(e)

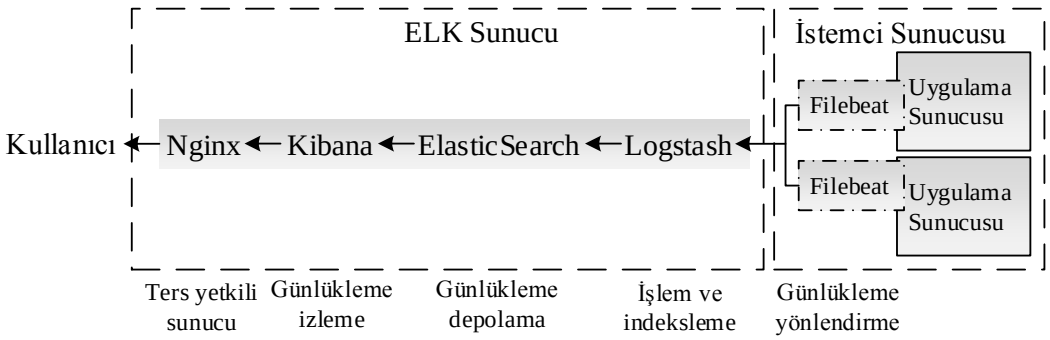


(f)

Şekil 4.5. ECAT önişlemci çalışması, sonuçlar ve günlüklemeler (Devamı)

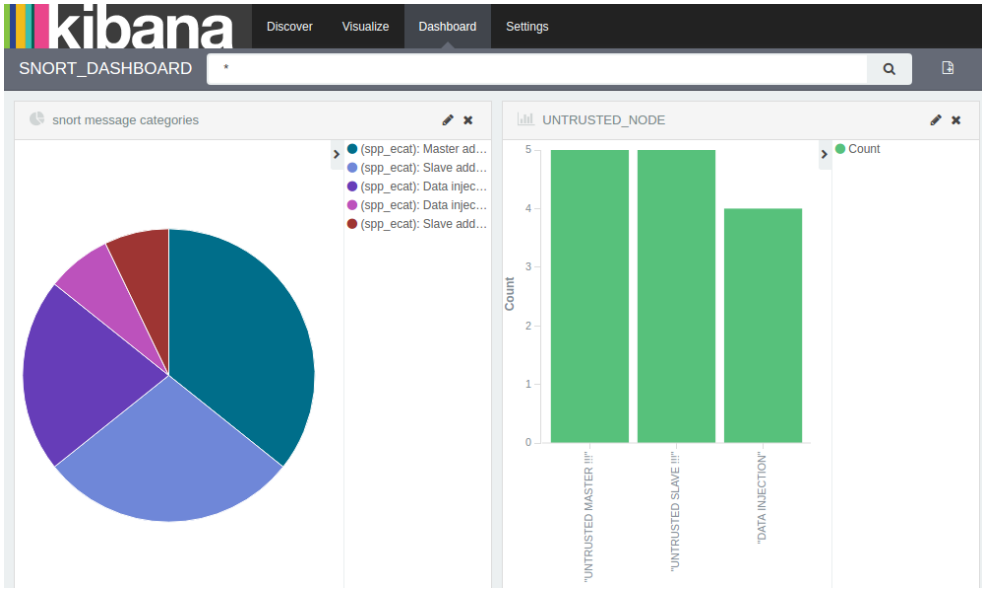
Burada sunulan yöntemler ve kurallar ileride farklı çalışmaların yapılabilmesine ve geliştirilmeye uygun olarak tasarlanmıştır. Örneğin yeniden oynatma gibi diğer saldırılar için zaman tabanlı yeni kurallar eklenebilir, geliştirmeler yapılabilir.

Önişlemci günlüklemelerinin ve güvenli düğüm yaklaşımı alarmlarının görüntülenmesi için bir sunucu istemci sistemi kurulmuştur. Bu işlem için Linux sistemlerde günlükleme toplama ve izlemeye yönelik sıkça tercih edilen bir yapı olan ELK kullanılmıştır. Bu izleme sistemi, ECAT önişlemcisinden çıkan Snort günlüklemelerini Filebeat aracı yardımıyla alır. Formatları Syslog olan ve JSON tabanlı veritabanı sorguları ile çekilen veriler, kullanıcıya oluşturulan bir panelde sunulmaktadır. Sistemin çalışması esnasında oluşan saldırıların hepsinin aynı zamanda bu panelde görüntülendiği izlenmiştir.



Şekil 4.6. ELK istemci-sunucu entegrasyonu

Oluşturulan yapı ise Şekil 4.6.'da özetlenmektedir. Buna göre istemci olan Snort IPS/IDS sistemi, konfigürasyon dosyası olan "Snort.conf" içindeki günlükleme bölümü, Linux auth.log dosyasına gönderecek şekilde ayarlanmıştır. Filebeat ise bu günlüklemeleri sunucu tarafa yönlendirmektedir. Günlüklemeler Syslog formatındadır. Sunucu tarafta ise Kibana arayüzü yardımıyla kullanıcı tarafından bu günlüklemeler görüntülenebilmekte olup ilgili konfigürasyon bilgileri Ek 1'de verilmiştir (Şekil 4.7.).



Şekil 4.7. ELK gösterge paneli

4.2. Sonuçlar

Bu bölümde EtherCAT saha seviyesi zafiyet analizi gerçekleştirilmiştir. Analiz neticesinde EtherCAT tabanlı bir önişlemci güvenli düğüm yaklaşımı eklenerek tasarlanmıştır. Literatürde, EtherCAT çalışma prensipleri göz önünde tutularak zafiyetlerin araştırıldığı bir çalışma daha önce yapılmamıştır. Bu nedenle önişlemci geliştirilmesinin öncesinde fabrika, saha ve dış dünya haberleşmelerini kapsayan bir araştırma gerçekleştirilmiştir. Sonrasında saha seviyesindeki zafiyetleri tespit etmek için saha iletişimini etkileyen MAC aldatma, DoS, veri enjeksiyonu ve köle adresleri saldırıları gibi çeşitli saldırı vektörleri geliştirilmiştir.

Sonuçlar EtherCAT protokolünün köleler ve efendi istasyonları arasında bir güvenlik mekanizmasının bulunmadığını göstermektedir. EtherCAT trafiği kolaylıkla saldırganlar tarafından dinlenebilir ve Ethernet üzerinden değiştirilen paketler yollanabilir. Bu da Ethernet üzerinden taşınan diğer bilinen TCP/IP saldırılarının da uygulanabilirliğini kanıtlamaktadır. Analiz esnasında, köle ve efendi arasında başlangıç durumunda bir yol oluşturulduğu ve PLC enerjisinin kesilmesi halinde bu yolun hemen sonlandırıldığı tespit edilmiştir. Bu başlangıç durumu dışında efendi ve köleler arasında akışa dayalı olan veya olmayan ek bir güvenlik işlemi

bulunmamaktadır. Bu zafiyet MAC aldatma saldırısına sebep olmaktadır. Ayrıca, kaba kuvvet yöntemiyle ağdaki kölelerin taranması saldırılarına yol açabilecek, efendi tarafından köleleri tanıyan bir mekanizma da bulunmamaktadır. Kölenin adres haritası veya EtherCAT ağındaki adres hiyerarşisi bilinirse, kölenin diğer bellek adreslerine veya diğer köle adreslerine erişilebilir. Ek olarak, fiziksel hasara dahi yol açabilecek veri boyutu değişimleri için de herhangi bir kontrol bulunmamaktadır.

Analiz sonuçlarına göre EtherCAT protokolünün Ethernet altyapısı üzerinden gelebilecek saldırılara karşı bir güvenlik mekanizmasına ihtiyacı olduğu anlaşılmaktadır. Snort sisteminde ise daha önce EtherCAT paketlerini işleyen bir önişlemci uygulaması bulunmadığından bu bölüm kapsamında yapılan çalışmalar EtherCAT tabanlı kritik sistemlerde temel düzeyde bir güvenlik sağlamaktadır. Çalışma pasif modda çalışmakta olduğundan gerçek zamanlılık kriterlerini bozmamakta ve ağa herhangi bir paket gönderimi yapmamaktadır. ECAT önişlemcisi Snort sistemini başlangıç aşamasında yüklenmekte olup gelen her EtherCAT paketini işlemektedir. Snort üzerinde daha önce kritik altyapılı sistemler için yazılmış DNP3, Modbus gibi önişlemci yapıları da mevcuttur. Diğer bir değişle Snort EKS protokollerinin gerçek-zamanlılık kriterlerini destekleyebilmektedir. Bu yüzden EtherCAT protokolü üzerinde de paket işleme sırasında herhangi bir probleme rastlanmamıştır.

Snort sisteminin paketleri öncelikle DecodeECAT çözümleyici modülü ile içeri alıp, kendine ait katmansal yapıdaki 3. seviyede, ayrıştırma motoru tarafından ayrıştırması sağlanmıştır. Böylece yeni kural ekleme veya yeni saldırı önleme metotlarının kolayca adapte edilebileceği fonksiyonel bir yapı geliştirilmiştir. Yeni kurallar tanımlanmıştır. Bunlardan bir tanesi güvenli düğüm yaklaşımı kapsamında önerilmiştir. Bu yöntemle konfigürasyon dosyasından efendi donanım adresleri, köle adresleri, mantıksal otomatik artırılan veya fiziksel adresler ile komutlar, komut sıraları ve veri boyutları çekilmiştir. Kontrollerle MAC aldatma, veri enjeksiyonu ve köle adresleri üzerine yapılabilecek olası saldırılara çözüm önerilmiştir. Çözümün bir diğer avantajı ise PROFINET veya Sercos gibi diğer otomasyon sistem protokollerinde de XML tabanlı benzer konfigürasyon dosyalarının bulunmasıdır. Bu nedenle, yapılan yaklaşımlar

diğer protokollere, önişlemci yapılarına uygulanabilir veya genelleştirilerek kritik altyapılara özgü yeni öneriler getirilebilir.

Bazı gerçek-zamanlı Ethernet tabanlı protokolleri 2. katman üzerinde iş yapmaktadır. Ancak Snort 3, 4 ve uygulama katmanı üzerindeki protokolleri desteklemektedir. Yapılan çalışma, 2. katmanda çalışan protokollerin Snort tarafından nasıl desteklenebileceğine bir çözüm getirmiş olup, test ortamında yapılan saldırılarında tespitini mümkün hale getirmiştir.

BÖLÜM 5. ETHERCAT TABANLI SİSTEMLERDE SAHA SEVİYESİNDEKİ ANOMALİLERİN PERİYOT TESPİTİ İLE BULUNMASI

EtherCAT tabanlı kritik altyapıların iletişim altyapısından gelebilecek olası saldırılara açık olduğu önceki bölümlerde aktarılmıştır. Bu sistemler bilinen saldırıların yanı sıra bilinmeyen tehditlere karşı da risk altındadır. Dolayısıyla EtherCAT üzerindeki anomalileri önleyici bir çözüm ihtiyacı bulunmaktadır. Diğer taraftan protokol, saha seviyesindeki iletişimini çevrimli olarak gerçekleştirmektedir. Bu bilgiler ışığında, tezin bu bölümünde, özellikle EtherCAT ağları için protokole özgü işlemleri ve saha seviyesindeki iletişimde yer alan protokol alanlarını kullanan yeni bir periyodiklik tespit sistemi geliştirilmiştir. Çözüm, saha iletişimindeki istatistikleri kullanan anomali tespitine dayanmaktadır. Periyodiklik farklı hassasiyet, anlamlılık düzeyi ve gecikme boyutlarında tespit edilebilmektedir. Yaklaşım 2 algoritma barındırmaktadır: Wireshark paket koklama programı üzerinde çalışan eklenti ve otokorelasyon fonksiyonu (ACF) kullanılarak periyot analizi. Çalışmada sunulan yaklaşımların test edilmesi için 4 PLC programı geliştirilmiş olup, yüksek doğrulukta tespit gerçekleştirilmiştir. Sistem üzerindeki herhangi bir kötücül aktivite periyodik örüntüyü değiştireceğinden, DoS ve kod enjeksiyonu saldırılarının yapay trafik izleri test edilmiştir. Önceden tespit edilen periyot kullanılarak trafik örüntüleri Snort IDS/IPS üzerinde geliştirilen bir anomali tespit modülü ile belirlenmektedir. Denemeler neticesinde iletişim örüntülerini bozan saldırıların tespit edilebildiği görülmüştür.

5.1. Çevrimli İletişimde Periyot Tespiti

EtherCAT protokolü iletişimi esnasında gözlemlenen başlıca zafiyetler, protokolün yetkilendirme, şifreleme olmadan ve kimlik doğrulaması olmadan saha, fabrika ve IP üzerinden dış dünya ile iletişimi gerçekleştirmesinden kaynaklanmaktadır. Bu

nitelikler EtherCAT dışında daha güvenli olarak tanımlanabilecek birçok protokolde dahi tam olarak sağlanamamaktadır. Bu kapsamda EtherCAT protokolü ile çalışan sistemlere, bilinmeyen ve sıfırıncı gün saldırıları olarak tanımlanan saldırılar gelebilmektedir. Bu türden saldırılar, protokolün yapısından veya sistemden kaynaklı zafiyetlerin sömürülmesi yoluyla gerçekleşmektedir. Sıfırıncı-gün saldırıları daha önce karşılaşılmamış ataklar olduğundan bunların imza tabanlı bir sistemle tespiti mümkün değildir. İmza veya kural tabanlı çözümler ancak bilinen saldırıları engellerler. EtherCAT tabanlı EKS'lerde, protokol zafiyetlerinden faydalanarak oluşturulan bu türden saldırılar için anomali tespiti yapan bir yapı ihtiyacı bulunmaktadır. Bu kapsamda, EtherCAT protokol iletişiminin işleyişlerini ve içeriklerini göz önünde bulundurarak izleme yapan ve normal iletişim davranışlarının dışına çıkıldığında tespit eden bir yapı gereklidir.

EtherCAT protokolü, saha seviyesindeki iletişim ve fabrikada seviyesindeki Process Data iletişim ihtiyacını çevrimli (senkron) şekilde; IP üzerinden dış dünya iletişimini ve fabrika seviyesindeki Mailbox iletişimini ise çevrimsiz (asenkron) şekilde yaparak karşılamaktadır. Tez kapsamında saha seviyesindeki iletişim üzerine yoğunlaşıldığı için ve saha iletişimi çevrimli olarak gerçekleştiğinden, çevrimsiz iletişimin yapıldığı diğer çerçeve yapıları burada anlatılmamaktadır.

Saha seviyesinde anomali tespiti yapılması için ise 2 parametrenin tespiti gerekmektedir: senkronizasyon periyodu ve örüntü tespiti. Periyot tespiti, örüntü tespit edilirken kullanılacak olup, zamana bağlı ve her konfigürasyonda farklılık göstermektedir. Çalışma boyunca periyotla kastedilen, PLC programının çevrimi ve ağda transfer edilen diğer değerlerin hat üzerinde tekrar ettiği zamanın bulunmasıdır.

5.1.1. Otokorelasyon fonksiyonu

Korelasyon, iki rastsal değişken arasındaki ilişkinin gücünü ve yönünü belirlemekte kullanılan istatistiksel bir yöntemdir. İki değişken arasında birlikte hareket etmenin veya nedensel olmayan ilişkinin ölçüsüdür. Otokorelasyon ise seri değerlerinin herhangi bir zaman dilimi ile olan ilişkisini geciktirerek açıklayan bir gösterim

biçimidir. Kavramsal olarak bir serinin herhangi bir dönemdeki değeri ile bir önceki veya bir sonraki dönem değeri arasında birlikte hareket etme ilişkisini ima eder [133]. Eğer bir seride otokorelasyon varsa serinin gözlemleri arasında bir korelasyonun var olduğu söylenebilir. Genelde otokorelasyonlar, serinin momentumundan ortaya çıkar. Bunun dışında otokorelasyon, tahmin ve gerçekleşmeler arasındaki farklardan oluşan hataların (kalıntıların), seri boyunca birbirleri arasındaki ilişkilerin bulunmasında da kullanılmaktadır. Hatalar arasında ilişkilerin bulunması durumunda, otokorelasyon limitler dışına taşacak ve zaman içinde belirli bir çevrime sahip olacaktır [133]. Bu durum, tahminlerin daha doğru yapılabileceği ve farklı modellerin kullanılmasının gerekliliğini göstermektedir. Bu yöntem, EtherCAT protokolünde pasif ağ izleme yoluyla elde edilen akışların içlerindeki ilişkiyi tespit etmede kullanılabilir. İlişkinin yüksek olduğu değerlerde periyot barınmaktadır. Bu yöntemde [-1, 1] arası değerlerde ilişki tespit edilmektedir. Otokorelasyonun -1 veya -1'e yakın olması, ilişkinin güçlü olduğu anlamı taşır. -1 negatif yönde yani ters bir ilişki olduğu, 1 ise pozitif yönde güçlü bir ilişkiyi belirtmektedir.

$$r_k = \sum_{t=k+1}^{n} (y_t - \hat{y})(y_{t-k} - \hat{y}) \Big/ \sum_{t=1}^{n} (y_t - \hat{y})^2 \qquad (5.1)$$

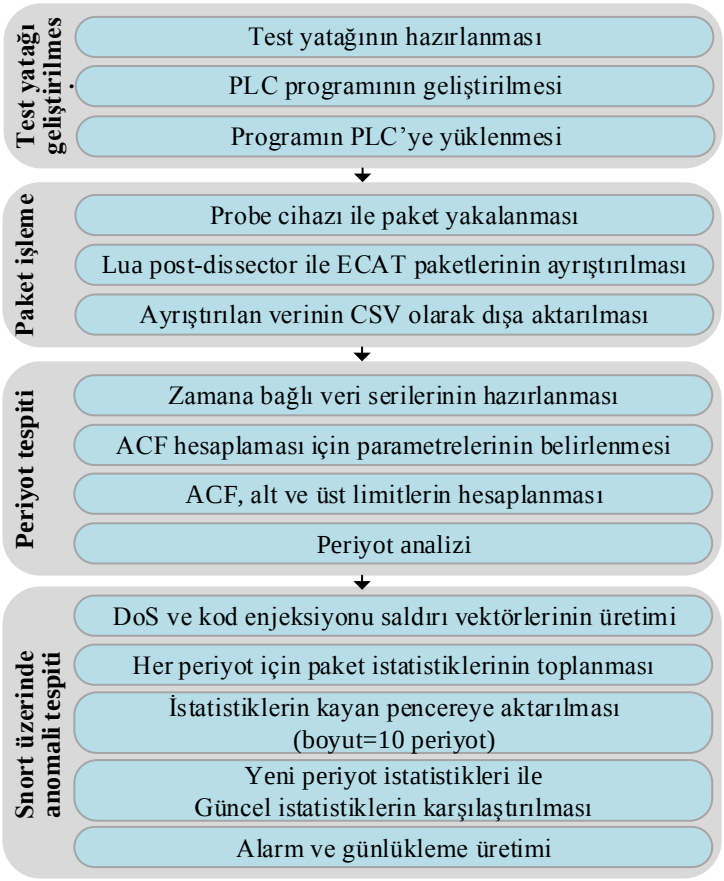
Denklem (5.1)' de, n; eleman sayısını, y_t; t anındaki serinin değerini, $\bar{y}$; serinin ortalamasını, k ise otokorelasyon adımını (lag - gecikme) temsil etmektedir. Bu adımda, kullanıcı tarafından belirtilen anlamlılık seviyesi ve istenen gecikme sayısı kullanılmakta olup, istenen gecikme miktarı kadar otokorelasyonlar hesaplanmaktadır. ACF korelogramı ACF sonuçlarının (r_k) gecikmelere bağlı olarak grafikte gösterim şekline denir. Korelogramda en önemli soru serinin durağan olup olmamasıdır. Serinin durağan olması beklenir. Seri durağansa geçmiş değerlerle ($r_k = 0$, $k = 1,\ldots,n$) ilişkili olmadığı anlamı taşır. Ancak r_k çoğu zaman sıfırdan farklıdır. Bu nedenle korelogramdaki r_k değerlerinin ilişkisi istatistiksel bir test olan hipotez testi ile sınanır. Hipotez testi alt-kritik (L-critical) ve üst-kritik (U-critical) değerleri ile yapılır. Belirli bir gecikme değeri sonunda çevrim oluşuyorsa kritik değerler aşılır. Kritik değerler gecikmeler kullanılarak aşağıdaki denklemle hesaplanır [134], [135].

$$Critical\ Values = \pm t_p \sqrt{\left(1 + 2\sum_{k=1}^{lag}(r_k^2)\right) \Big/ (n - lag)} \quad (5.2)$$

Denklem (5.2)' de, n gözlem sayısı, t_p normal olasılık dağılımında belirli bir değer için t test değerini göstermektedir. Bu değerler çevrim sınırlarının aşılıp aşılmadığının tespitinde kullanılır.

5.1.2. Periyot analizi

Periyot analizi çalışması 4 aşamadan oluşmaktadır. Bunlar: test ortamının oluşturulması, paketlerin işlenmesi, periyot tespiti ve anomali tespitidir (Şekil 5.1.).



Şekil 5.1. Çalışmanın akışı

İlk 3 aşama bu alt bölümde yer almakta olup son aşama sonuçlar bölümünde sunulmaktadır. İlk aşamada test ortamı oluşturulmuş ve PLC programı yazılmıştır. Veri

toplama ve verilerin hazırlanması ikinci aşamada tamamlanmıştır. ACF hesaplamaları ve ACF kullanılarak periyodun bulunması 3. aşamada gerçekleşirken, son aşamada ise bulunan periyot kullanılarak ağ trafiği izlenmiş ve Snort sistemine gelen anomaliler tespit edilmiştir.

5.1.2.1. Test ortamı

Bu bölümde yürütülen çalışmalar Bölüm 3.1.'de sunulan test ortamında gerçeklenmiştir. Çalışmalar kapsamında, test ortamında kullanılmak üzere 4 adet PLC programı geliştirilmiş olup program detayları Tablo 5.1'de verilmiştir.

Tablo 5.1. Geliştirilen PLC programları

Program #	ICS seviyesi	Detaylar
1	Saha	I/O üzerindeki LED ışığın 5 sn. yanıp, 5 sn. sönmesi
2	Saha	I/O üzerindeki LED ışığın 2 sn. yanıp, 2 sn. sönmesi
3	Saha	Konveyör bant üzerindeki matlabın 2 sn. delmesi, 2 sn. beklemesi ve sonrasında bantın 25 adım ilerlemesi
4	Saha	Kavşaktaki 2 trafik ışığının kontrolü

Geliştirilen ilk 3 program detayları tabloda sunulmuş olup, 4. programda 2 trafik ışığının bir kavşak üzerindeki kontrolü simüle edilmiştir. Her çevrimde kırmızıdan kırmızı-sarıya geçiş 5 sn., kırmızı-sarıdan sarıya geçiş 2 sn., yeşilden sarıya geçiş 5 sn. ve sarıdan kırmızıya geçiş 2 sn. olarak tanımlanmıştır. Toplam 7 çevrim sonunda akşam olduğu ve ışıkların 10 sn. söndüğü varsayılmıştır. Bu süre sonunda sabah olduğu varsayılarak çevrim yeniden başlamaktadır. Bu şekildeki bir çevrim 108 sn. sürmektedir.

5.1.2.2. Paketlerin işlenmesi

Ağ trafiği daha önce belirtildiği üzere ET2000 cihazı yardımıyla 1Gbps hızında alınmaktadır. Paketler alındıktan sonra ise işlenmektedir. Bu aşamada Lua Post-Dissector ve periyot tespit algoritmaları olmak üzere 2 algoritma geliştirilmiş olup

Post-Dissector paketlerin ayrıştırılmasında kullanılırken diğer algoritma periyodun analizinde kullanılmaktadır.

Wireshark açık kaynak kodlu bir ağ koklama programı olup, farklı protokol paketlerini tanıyabilmektedir. EKS protokollerinden Modbus, Profinet ve DNP3 protokollerini, geliştirilen ayrıştırıcılar (dissector) yoluyla destekleyen Wireshark, EtherCAT içinde bir ayrıştırıcıya sahiptir. Paket içerikleri, Wiresharkın paket detaylarının yer aldığı alt panelde, depolandıkları ağaç yapısında sunulmaktadır. Üstte yer alan paket görüntü liste panelinde ise istenen alanları filtrelenen paketler, her satırda bir paket olacak şekilde listelenmektedir. Buradaki veriler ayrıştırıcılardan sorgu yapılarak çekildiğinden, protokol ağaç yapısının ancak tanımlanan noktalarına erişilebilmektedir. Buradan izlenen/toplanan günlüklemelerinin Matlab, Excel, SAS vb. ortamlara aktarılması için dışa aktarılmaları gerekmekte olup, bu işlem farklı formatlarda gerçekleşebilmektedir. Burada dikkat edilmesi gereken, csv veya json gibi günlükleme formatlarındaki dışa aktarımlarda, paket liste görüntü panelinde yer alan içerikler aktarılmaktadır. Bu şekildeki bir aktarımda bazı işlemlerin sonucu veya farklı detayların aktarılması istenirse, ayrıştırıcılar üzerinde iş yapan Wireshark eklentisi geliştirmek gerekmektedir.

Lua, Rio De Janerio Üniversitesi, Brezilyadaki bilim adamları tarafından geliştirilmiş bir programlama dili olup, Wireshark kurulumunda otomatik olarak gelen prototip geliştirme ve komut dizisi (script) dili olarak kabul görmüştür [136]. Eskiden Lua eklenti olarak yüklenebilirken, yeni sürümlerde kurulum aşamasında otomatik olarak yüklenmektedir. Lua`nın Wiresharkta yüklü olduğu Tools>Lua sekmesine gelerek görülebilir. Wireshark temel olarak C dilinde yazılmıştır. Dolayısıyla kullanılan tüm ayrıştırıcı, Post-Dissector, Tapping işlemleri de C de yapılmaktadır. Lua dili yazılan C dosyalarına kolay ulaşmak, Wireshark'ın yapısını kolay algılama ve ayrıştırıcıların kolay kullanımını sağlar. Nedeni ise Wireshark temelde paket analizi aşamasını Dissector (ayrıştırıcı), Post-Dissector ve Tapping olarak üçe ayırmasından kaynaklanır. Eğer bir ayrıştırıcı yazılmak istenirse, C dili Lua'dan daha hızlı çalıştığı için C'de yazmak daha avantajlıdır. Ancak daha önce yazılmış ayrıştırıcıları kullanarak işlem tamamlanabiliyorsa, Lua`nın kullanımı programcıya zaman

kazandıracaktır. Ayrıştırıcılar gelen paketleri ayrıştırıcı ağaç yapısına göre kategorize eder. Post-Dissectorlar ise ayrıştırıcılar işlemini bitirdikten sonra ağaca yeni nesneler eklemekten sorumludur. Tapping yapıları ise ayrıştırılmış olan ağaç yapısındaki paketlerden bilgi çekmek için kullanılır.

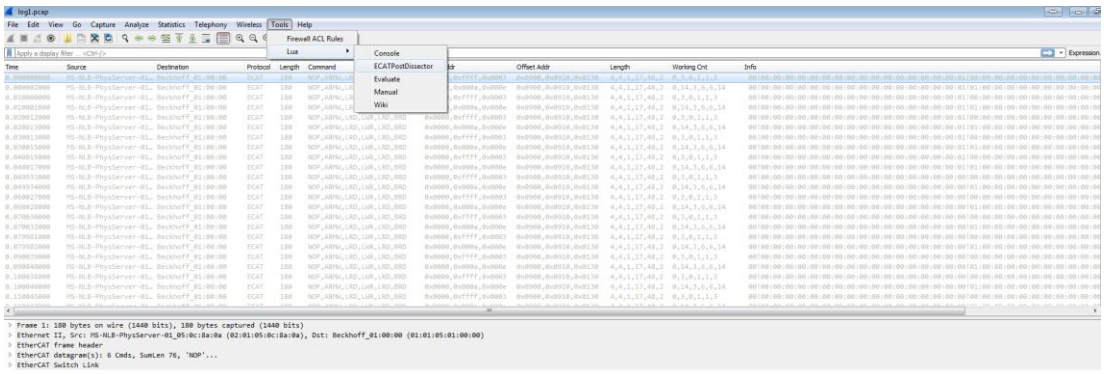
Paket liste görüntü paneli üzerinde yer almayan fakat analiz aşamasında gerekli olan bazı değerlerin panele getirilmesi Post-Dissector yazılması ile gerçekleştirilmiştir. Çalışma kapsamında geliştirilmiş olan ayrıştırıcı Wireshark üzerinde Tools>Lua>ECATPostDissector sekmesinden Şekil 5.2.' deki gibi erişilebilmekte olup, program yüklendiğinde otomatik olarak eklentiler dizininden yüklenmekte ve aşağıdaki gibi bir akışta çalışmaktadır (Algoritma 5.1.).

Algoritma 5.1. ECAT Post-Dissector sözde kodu

1. Her ecat paketini ecat.sub.datagram içine ayrıştır.
2. Her ecat.sub.datagram için
   (a) i = 0
   (b) if (ecat.sub.datagram[i].cmd = xWR (APWR, LWR, FPWR))
      I. Yeni alan i'yi olıuştur, i = ecat.sub.datagram[i].data
      II. i = i+1 (i < n, burada n değeri xWR datagram sayısıdır)
3. post-dissector kaydı için temp-proto protokolü oluştur.
4. temp-proto (pinfo, tree) tabanlı post-dissector fonksiyon geribildirimi oluştur.
   (a) Çözümleyici alan obje tablosunu alanlara ayır.
   (b) Yeni alanlarla ilgili alanları eşleştir.
   (c) Birleştirilmiş yeni alanlara pinfo.cols.info değeri ata.
5. Sonraki çözümleme için temp-protoyu kaydet.
6. ECAT post-dissector yapısını menü araçları içine bileşen olarak yerleştir.

Geliştirilen temel ECAT Post-Dissector yardımıyla yakalanan her EtherCAT paketinin uzunluğu, paket içindeki her datagramın barındırdığı komut isimleri, köle adresleri, ofset adresleri, her datagramın veri alanı boyutu, gelen ve giden WKC değerleri ve veri alanları içeren datagramların içindeki değerlerin onaltılık karışıkları birleştirilmiş olarak görüntü panelinde sunulmaktadır. Bu işlem hem dinamik paketler hem de

günlüklemeler üzerinden yapılabilmektedir. Bu sayede, yakalanan paketler periyot analizi için uygun hale getirilmektedir. Daha sonra bu paketler csv formatında dışa aktarılarak kullanılabilmektedir.



Şekil 5.2. ECAT Post-Dissector Wireshark görünümü

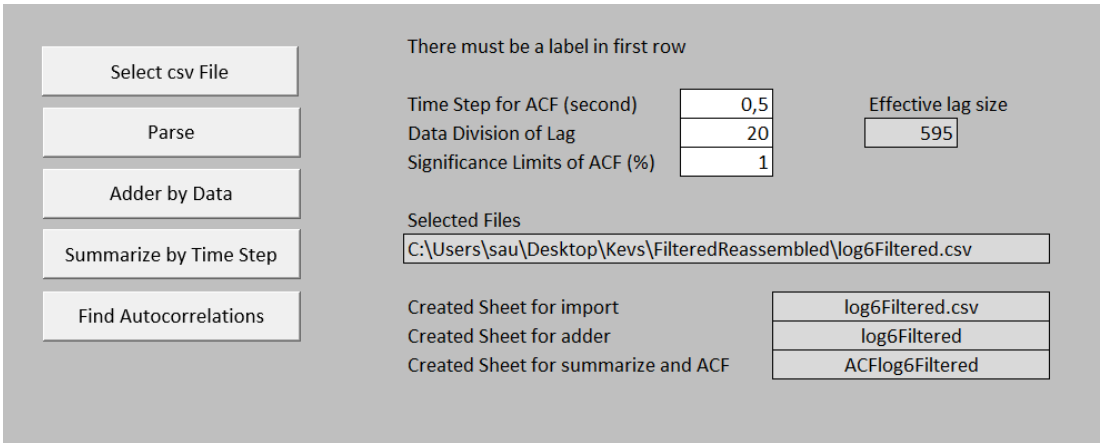
5.1.2.3. Periyot analizi

Çevrimli iletişim için geliştirilen periyot analizi, izlenen paketler üzerinden periyot çıkarımı yapmakta kullanılmaktadır. Bu kapsamda izlenen paketler periyot analizi modulüne girdi olarak verilmekte ve analiz aşaması gerçekleşmektedir.
Çevrimli EtherCAT paketleri belirli periyotlarla konfigürasyon, program, giriş/çıkış birimi sayısı ve çevrim zamanı gibi kavramlara bağlı olarak aynı hat üzerinde tekrarlanmaktadır. Literatür çalışmasında belirtildiği üzere periyot analizinde çeşitli yöntemler kullanılmaktadır. Ancak TCP/IP protokol yığınını kullanmayan EtherCAT çevrimli iletişimi ile ilgili şimdiye kadar yapılmış bir analiz çalışması bulunmamaktadır. EtherCAT paketleri, saha iletişimi ve Process Data iletişiminde belirli zaman aralıklarında birbiri ile benzerlik veya ilişki barındırmakta olduğundan otokorelasyon fonksiyonu ile akışlar içerisindeki periyot tespit edilebilir.

Otokorelasyon tabanlı (ACF) periyot analizi arayüzü Şekil 5.3.'de verilmiştir. Geliştirilen bu uygulamaya Wireshark üzerinden dışa aktarılan *csv* uzantılı dosya girdi olarak verilmekte ve kullanıcının belirttiği parametreler kullanılarak otomatik olarak periyot tespit edilmektedir. Uygulamanın hem Windows hem de Linux ortamında çalışan versiyonları makrolar üzerinden geliştirilmiştir. Bu arayüz yardımıyla

yapılacak olan analiz işleminden önce, ACF'nin uygulanacağı zaman adımı (time step), gecikme bölümü (division of lag) ve anlamlılık sınırının (significance limit) belirtilmesi gerekmektedir. Bu değerler;

Zaman adımı: Bulunacak periyodun hassasiyeti (paketlerin ne kadar süreyle toplanacağı),
Gecikme bölümü: Uygulanacak gecikme sayısını bulmak için veriyi kaç parçaya bölmek gerektiği (böylece ACF grafiğindeki gecikme sayısı bulunmuş olur),
Anlamlılık sınırı: Verinin seriyi yüzdesel olarak ne kadar temsil etmesi istendiği, şeklindedir. Verinin boyutu bilindiği için en iyi sonucu vermesi için gerekli olan minimum gecikme sayısı (effective lag size) otomatik olarak hesaplanarak kullanıcıya sunulmaktadır.



Şekil 5.3. ECAT periyot analizi arayüzü

Periyot tespiti algoritması 4 işlemde periyodu tespit eder. Bunlar: *csv* dosyasını almak ve ayrıştırmak, datagram verilerini birleştirmek, zaman adımına göre toplamak ve otokorelasyonların hesaplanmasıdır.

Periyot tespiti için belirli bir zaman aralığında ağın pasif modda izlenmesi gerekmektedir. İzlenen ağdaki paketler λ ile gösterilmiştir. *i* değeri, kaçıncı paket olduğunu tanımlamaktadır. İzlenerek elde edilen PCAP kayıtları içerisinde zaman, EtherCAT başlık bilgileri, datagram başlık bilgileri, datagram verisi gibi birçok bilgi yer almaktadır. Ψ , her bir paketin içerisindeki datagramların veri alanlarını elde eden

fonksiyondur (Denklem 5.3). Her paket içinden ayrıştırılan datagramların verisi, ω_{ij} ile ifade edilmektedir. Burada *j* indisi datagramı göstermektedir.

$$\omega_{ij} = \Psi(\lambda_i)_j \tag{5.3}$$

Ω ise *i*. paket ve *j*. datagram içindeki datagram verisini birer bayt şeklinde ayırmaktadır ve sonucu τ değişkeninde *k* indisi ile tutmaktadır (Denklem 5.4).

$$\tau_{ijk} = \Omega(\omega_{ij})_k \tag{5.4}$$

Γ fonksiyonu ise onaltılık sayı sistemindeki bu baytları ondalık sisteme çevirmekte ve her datagram verisi ve paket için dönüştürülen bu değerleri toplamaktadır (Denklem 5.5). Burada *n* her datagramın veri alanındaki bayt sayısı, *m* ise her paketteki datagram sayısını ifade etmektedir. Bu işlem sonrasında hazır olan her paket için toplam değeri olan δ bulunmuş olur.

$$\delta_i = \sum_{j=1}^{m}\sum_{k=1}^{n}\Gamma(\tau_i) \tag{5.5}$$

Otokorelasyonları bulmak için serilerin zamana bağlı aralıklarda ifade edilmesi gerekmektedir. δ ile bulunan her toplam değer, Φ fonksiyonu ile aralık bazında ifade edilir (Denklem 5.6). Aralık bazında ifade edilen değerler toplanarak φ yeni serisine aktarılır.

$$\varphi_t = \Phi\left(\sum(\delta_i)\right) \tag{5.6}$$

i kadar olan paket boyutu aralık bazı olan *t* ile gösterildiğinden $i>t$ olacaktır. Böylece ACF fonksiyonun kullanacağı aralık bazında yeni veriseti elde edilmiş olmaktadır.

Yukarıda teorik olarak ifade edilen işlemler, periyot bulma esnasında aşağıdaki gibi uygulanmaktadır. Öncelikle, eklenti yardımıyla istenilen formatta oluşturulan izleme verileri girdi olarak alınmaktadır. Alınan bu verideki çevrimi bulmak için Denklem

5.7'deki formül kullanılarak yeterli veri gecikme boyutu bulunabilir. Diğer bir ifade ile çevrim olup olmadığının tespiti için eleman sayısının *N'* değerine eşit olması gerekmektedir [137]. Burada, *N* serideki veri sayısını, r_1 ise birinci otokorelasyon değerini temsil etmektedir. Bu işlemle efektif gecikme sayısı kullanıcıya öneri olarak sunulmaktadır.

$$N' = N\frac{(1-r_1)}{(1+r_1)} \tag{5.7}$$

Periyot analizi için girdi olarak alınan veriler, bir sonraki aşamada ayrıştırılmaktadır. Burada her bir EtherCAT datagramı içerisinde yer alan veri alanı, 1'er bayt şeklinde onaltılık sayı sistemine göre "ayrıştırıcı" (parse) modülü yardımıyla bölünmektedir. Daha sonra bu değerler ayrı ayrı ondalık sayılara dönüştürülmektedir. "Datagram verileri birleştirme" (adder by data) modülü ise bu veri alanlarını toplayarak önce datagram, daha sonra paket bazında bir toplam değeri elde etmektedir. Kullanıcı tarafından belirtilen zaman aralıklarını "ACF için zaman adımı" (time step for ACF) kısmında tanımlanarak "zaman adımına göre toplama" (summarize by time step) adımında kullanılmaktadır. Bu adımda, periyot için istenen hassasiyet göz önünde tutularak, "datagram verileri birleştirme" (adder by data) modülünden elde edilen veri alanları toplanmaktadır. Otokorelasyon (find autocorrelations) adımında ise otokorelasyon fonksiyonu uygulanarak zaman aralıkları arasındaki ilişkiler tespit edilmektedir [135].

Ayrıştırıcı modülü veri alanını 1 baytlık verilere böldüğünden, veri alanındaki küçük bit değişimleri bile ondalık dönüşümün ardından kolaylıkla tespit edilebilmektedir. Burada her periyottaki veri boyutundaki değişimler belirli bir pencere boyutuyla kaydedilmektedir. Üst ve alt limitlerin belirli miktarda aşılması durumunda saldırı olarak tanımlanmaktadır. En karmaşık saldırılarda dahi paket verilerinde değişimler olacağından "Datagram verileri birleştirme" modülü ile bulunan değerler farklılaşacak ve saldırı tespiti mümkün olacaktır.

5.2. Sonuçlar

EtherCAT paketleri içindeki datagramlarda yer alan veri alanlarının dönüşümleri sonucunda elde edilen, PLC üzerinde 4 farklı program çalıştırılarak oluşturulan φ_t serilerinin otokorelasyon fonksiyon sonuçları bir sonraki bölümde sunulmaktadır.

5.2.1. Periyot tespiti sonuçları

Periyot tespiti yapılan 4 programın önceden tanımlanmış parametreleri Tablo 5.2.'de verilmektedir. Gecikmelerin bölümleme sayısı ve anlamlılık sınırları daha doğru bir karşılaştırma yapılması açısından benzer alınmıştır. Alt ve üst kritik değerler ise kullanıcı tarafından arayüzden girilen anlamlılık sınırlarından hesaplanmıştır. Anlamlılık seviyesi %5 ise, güven aralığı %95'tir.

Tablo 5.2. Geliştirilen PLC programlarında alınan parametreler

Günlükleme	Program #	Zaman adımı	Gecikme bölümü	Anlamlılık sınırı (p)
1	1	0.5 sn.	20	0.01
2	1	0.2 sn.	20	0.01
3	2	0.1 sn.	20	0.01
4	3	0.5 sn.	20	0.01
5	4	1 sn.	10,000	0.01

U-critical üst limit, L-critical alt limit anlamı taşımaktadır. Dolayısıyla, Şekil 5.4., Şekil 5.5., Şekil 5.6., Şekil 5.7. ve Şekil 5.8.'de Y ekseninin pozitif kısımda yer alan kırmızı eğri üst limit değerlerini, negatif kısımda kalan değerler ise alt limitleri belirtmektedir. Program 1'de (günlükleme 1) ACF hesaplaması için zaman adımı 0,5 sn., gecikme bulunması için bölüm sayısı 20, anlamlılık seviyesi *p* ise 0,01 alınmıştır. Yapılan çalışma sonunda, bu programın periyodu lag1 - lag21 arasında olduğu görülmüştür (Şekil 5.4.). 0,5 sn. hassasiyet ile yapılan çalışma sonucunda 20 gecikme değeri sonrasında en yüksek anlamlı otokorelasyon sonucu 0,99 olarak bulunmuştur. Her gecikme değeri arasındaki fark 0,5 sn. olduğundan serinin çevrim boyutu 10 sn. (20x0,5) olarak elde edilmiştir. Bu seri için efektif gecikme sayısı 183 olarak hesaplanmıştır.

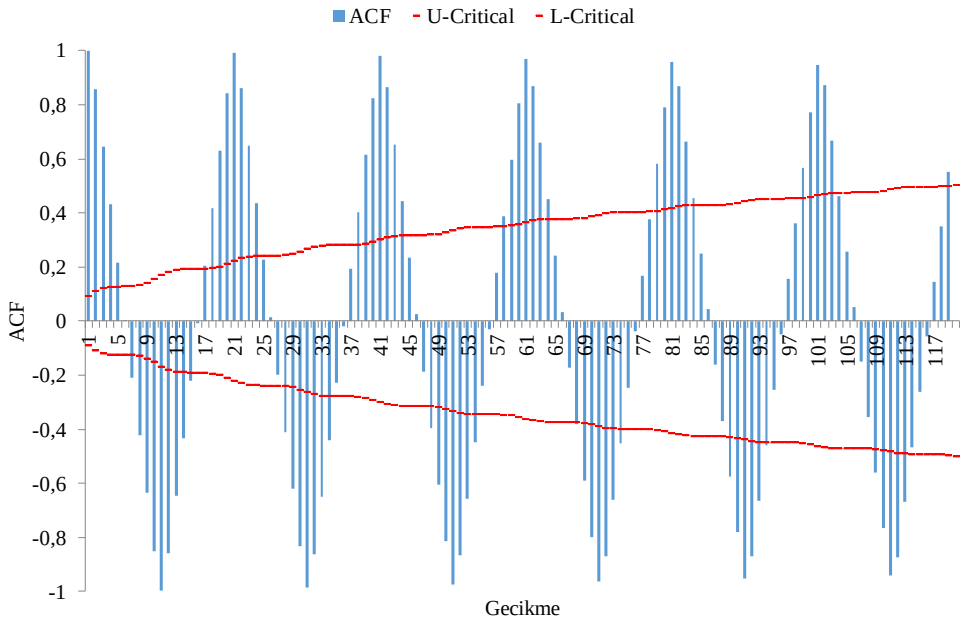
Aynı programda ACF hesabı için zaman adımı 0,2 sn. alındığında periyot lag1 - lag51 arasında değişmiştir (Şekil 5.5). Hassasiyet 0,2 sn. ve gecikme farkı 50 olduğundan serinin çevrim boyutu yine 10 sn. olarak elde edilmiştir. Görüldüğü üzere aynı programda çevrim değişse dahi periyot tespit edilebilmektedir. Hassasiyetin arttığı seride efektif gecikme sayısı 169 olmuştur.

Program 2'de ACF hesaplaması için zaman adımı 0,1 sn., gecikme bulunması için bölüm sayısı 20, anlamlılık seviyesi *p* ise 0,01 alınmıştır. Yapılan çalışma sonunda, bu programın (günlükleme 3) periyodu lag1 - lag41 arasında olduğu görülmüştür (Şekil 5.6.). 0,1 sn. hassasiyet ile yapılan çalışma sonucunda 40 gecikme değeri sonrasında en yüksek anlamlı otokorelasyon sonucu 0,95 olarak bulunmuştur. Her gecikme değeri arasındaki fark 0,1 sn. olduğundan serinin çevrim boyutu 4 sn. olarak elde edilmiştir. Bu seri için efektif gecikme sayısı 32 olarak hesaplanmıştır.

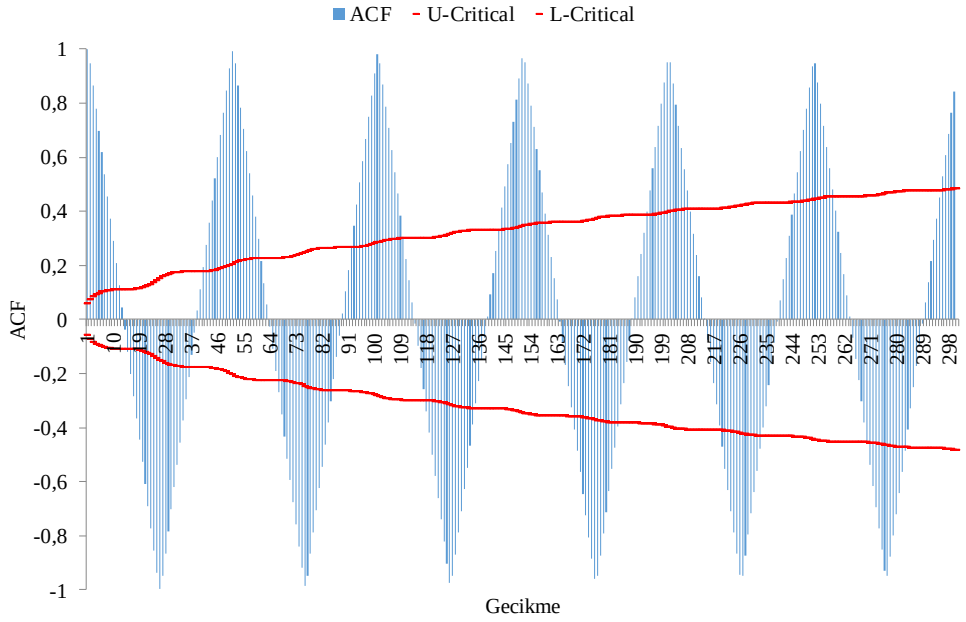
Program 3'de ACF hesaplaması için zaman adımı 0,5 sn., gecikme bulunması için bölüm sayısı 20, anlamlılık seviyesi *p* ise 0,01 alınmıştır. Yapılan çalışma sonunda, bu programın periyodu lag1 - lag12 arasında olduğu görülmüştür (Şekil 5.7.). 0,5 sn. hassasiyet ile yapılan çalışma sonucunda 11 gecikme değeri sonrasında en yüksek anlamlı otokorelasyon sonucu 0,99 olarak bulunmuştur. Her gecikme değeri arasındaki fark 0,5 sn. olduğundan serinin çevrim boyutu 5,5 sn. olarak elde edilmiştir. Bu seri için efektif gecikme sayısı 595 olarak hesaplanmıştır.

Program 4 için ACF zaman adımı 1sn., gecikme bölümlemesi 10,000, anlamlılık aralığı *p* 0,01 olarak alınmıştır (günlükleme 5). Periyot ise lag1 - lag109 arasında 108 sn. olarak bulunmuştur (Şekil 5.8.).

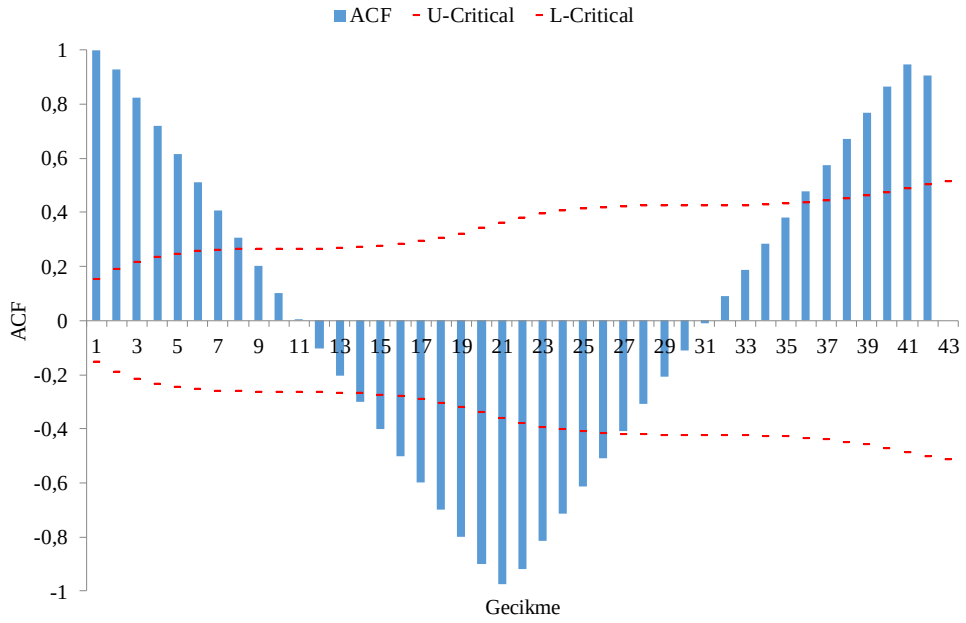
Geliştirilen 4 programdan alınan 5 günlükleme için ACF korelogramları aşağıdaki şekillerde verilmektedir. Görüldüğü üzere Tablo 5.1.'de verilen periyotlar ile tespit edilenler aynıdır. 1 numaralı programın 0,5 sn. ve 0,2 sn. hassasiyetleri için periyotlar 10 sn. olarak bulunmuştur. Bu sonuç, farklı zaman hassasiyetleri belirlense bile periyodun doğru bir şekilde tespit edilebildiğini göstermektedir.



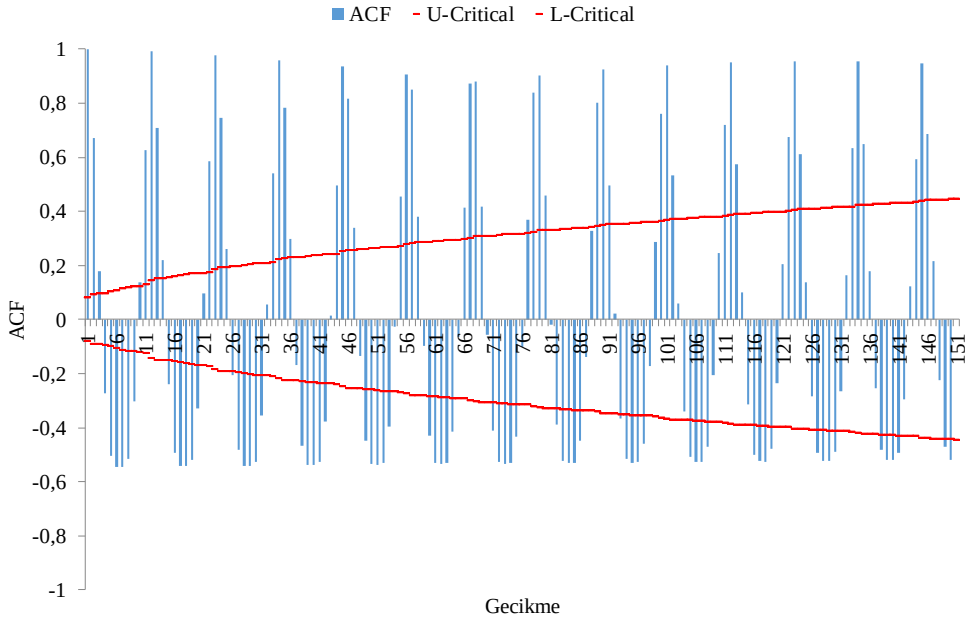
Şekil 5.4. Günlükleme 1 EtherCAT üzerinden 5 sn. I/O yakıp söndürme senaryosu – 0,5 sn. hassasiyet



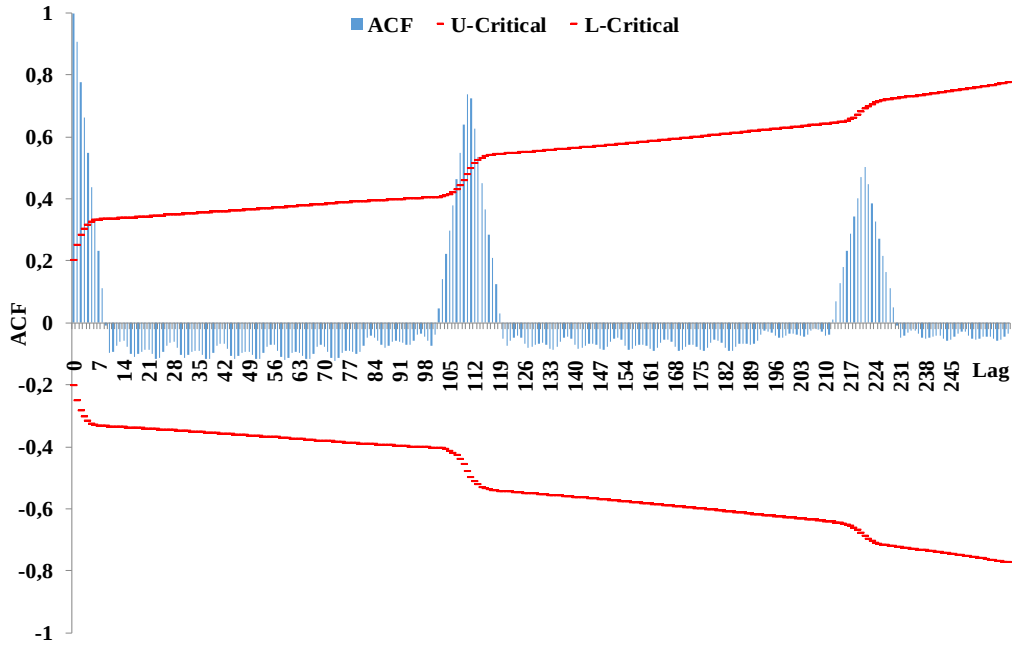
Şekil 5.5. Günlükleme 2 EtherCAT üzerinden 5 sn. I/O yakıp söndürme senaryosu – 0,2 sn. hassasiyet



Şekil 5.6. Günlükleme 3 EtherCAT üzerinden 2 sn. I/O yakıp söndürme senaryosu – 0,1 sn. hassasiyet



Şekil 5.7. Günlükleme 4 EtherCAT üzerinden matkap senaryosu – 0,5 sn. hassasiyet



Şekil 5.8. Günlükleme 5 Kavşakta trafik ışıkları senaryosu – 1 sn. hassasiyet

5.2.2. Anomali tespiti sonuçları

Bu bölümde tespit edilen periyotun anomali tespitinde ne şekilde kullanılabileceği anlatılmaktadır. Bu kapsamda 2 farklı saldırı geliştirilmiştir: DoS, kod enjeksiyonu. Bu saldırılar EKS'de sıkça gerçekleşen MITM veya yeniden oynatma atakları şeklinde oluşturulmuştur.

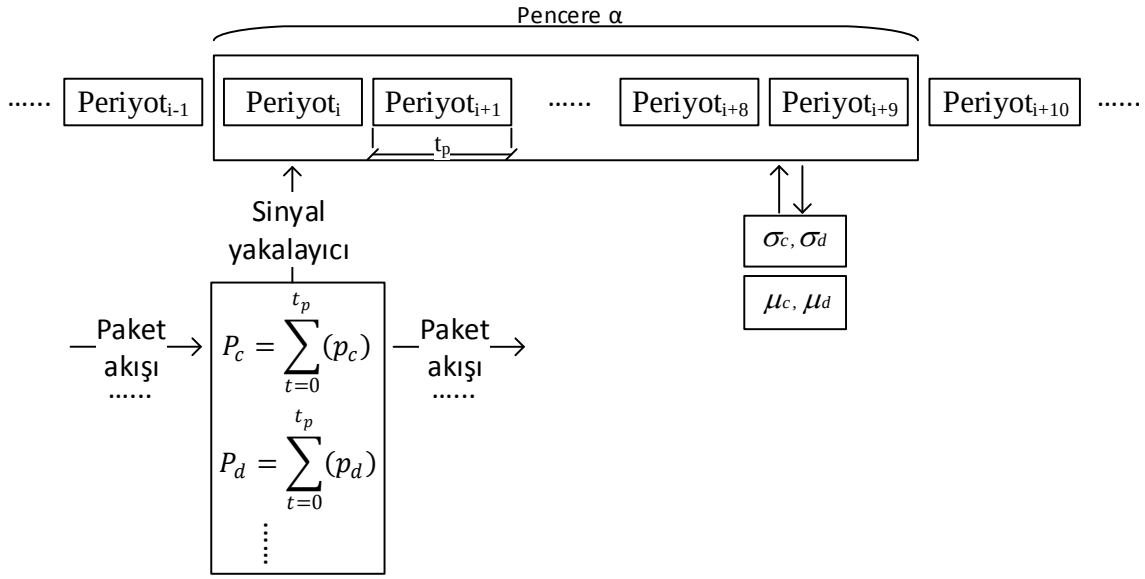
Kod enjeksiyonu saldırısında EtherCAT paketlerinin boyut ve içerik gibi veri alanları değiştirilmiştir. Köleler üzerine yazma istekleri de bu türden saldırılardandır. Tespiti yapılan periyot içinde yazma isteklerinin veri alanları toplamları kullanılmaktadır.

DoS saldırıları BT ve EKS'de en yaygın gerçekleşen saldırı türlerindendir. Bu saldırıyı tespit etmek için periyot içindeki toplam paket sayısı hesaplanmaktadır. Belirli bir periyot içinde saldırı varsa, alınan paket sayısı ani bir şekilde yükselmektedir. Saldırı sadece 1 periyot süresi içinde gerçekleşse dahi hat üzerindeki periyot bilinirse tespit edilir.

Periyot tespit edildikten sonra protokolün kabulleri ve genel istatistiklerin analiz edilmesi gerekmektedir. Anomali tespiti Snort üzerinde yapıldığından, tespiti yapılan periyot bir önceki bölümde geliştirilen önişlemciye girdi olarak verilmektedir. Bu işlem Snort sisteminin konfigürasyonlarının yer aldığı "Snort.conf" dosyasına yazılmasıyla gerçekleşir. Ardından, çalışma sırasında istenen istatistikler alınır ve paketler üzerinde ayrıştırma işlemi yapılır. Anomaliler ise kayan pencere yaklaşımı kullanan anomali tespit algoritması yardımıyla (Şekil 5.9.) tespit edilir [138]. Buna göre alınan her paket gerekli alanların ayrıştırılma işlemi için sinyal yakalayıcı yardımıyla algoritmaya verilmektedir. DoS ve kod enjeksiyonu saldırı tespitleri için, paket sayısı (p_c) ve paket içindeki tüm datagramların sadece veri alanlarının toplamı (p_d) gibi istatistikler alınmıştır. Burada yapılan hesaplamalar periyot bazlı uygulanmaktadır (t_p). Periyot zamanlayıcısı dolduğunda, gönderilen bir sinyal yakalanarak, toplanan istatistikler $Period_{i,\ i+1,\ \ldots}$ olarak kaydedilmektedir. Böylece her periyodun nitelikleri Denklem 5.8'deki gibi oluşmaktadır:

$$Period_i = \{ P_c, P_d, \ldots \} \tag{5.8}$$

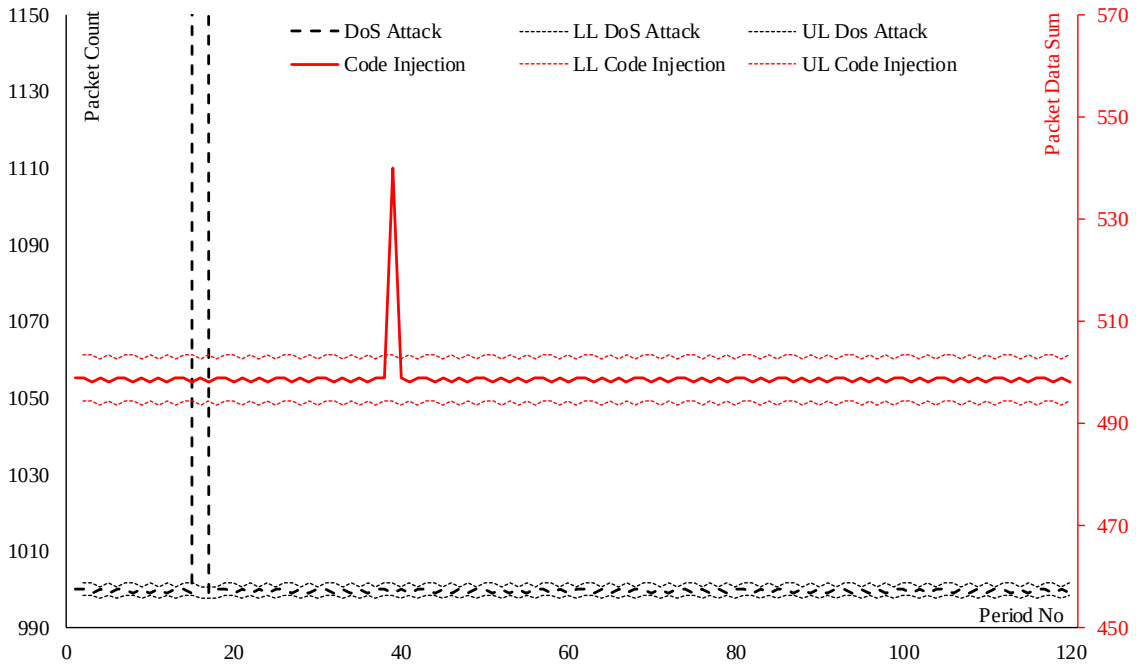
Anomali tespit algoritması α sayıdaki periyodun istatistik verisini tutmaktadır. Çalışma kapsamında bu rakam 10 olarak belirlenmiştir. Bunun yanında, güncel pencere içindeki periyotların ortalamaları (μ) ve standart sapmaları (σ) da kaydedilmektedir. Çevrimli iletişim için paket boyutları ve sayılarının normal dağılım gösterdiği tespit edilmiştir. Bu kapsamda hipotezimiz, otokorelasyondaki güvenlilik sınırının 0,9974 olarak test edilmesidir. Bu nedenle, bir periyotluk istatistik verisi toplandığında, değerlerin $\mu \pm 3\sigma$ olması beklenmektedir. Değerler bu kriteri sağlıyorsa, en sondaki periyot silinir ve yeni değerler eklenir. Aksi durumda alarm verilir ve pencerede herhangi bir değişiklik yapılmaz.



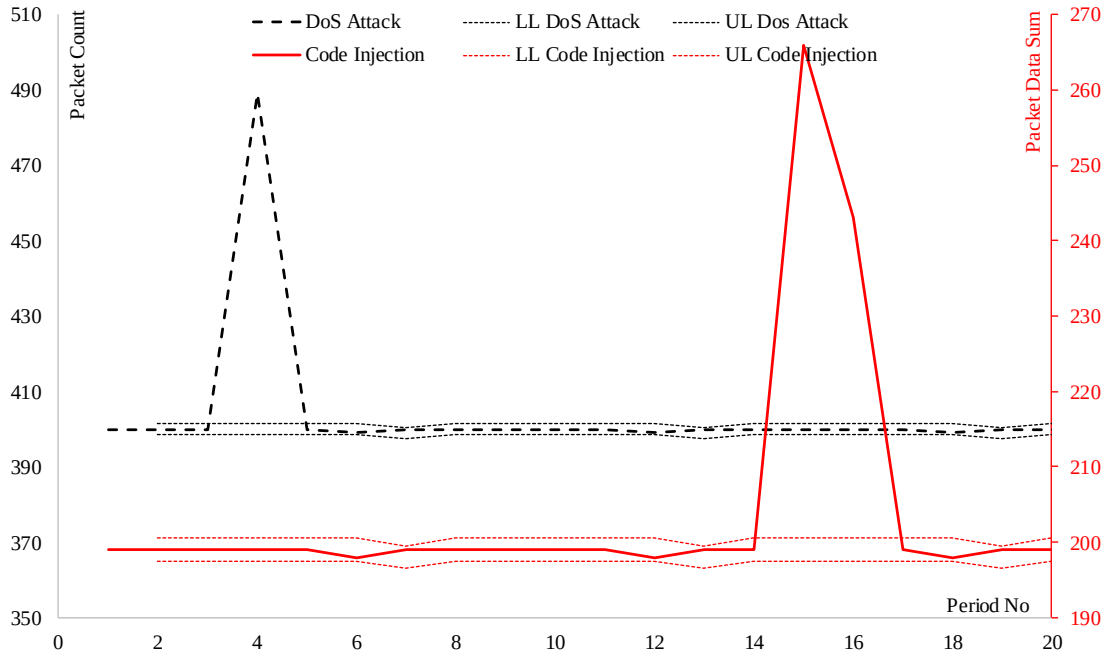
Şekil 5.9. Kayan pencere tabanlı anomali tespit algoritması

Şekil 5.10., Şekil 5.11., Şekil 5.12. ve Şekil 5.13.'de DoS ve kod-enjeksiyonu saldırılarının tespiti, periyot tabanlı kayan pencere yardımıyla, 4 farklı senaryo üzerinden gösterilmektedir. LL ve UL kısaltmaları sırasıyla, alt limit (μ-3σ) ve üst limit (μ+3σ) anlamı taşımaktadır. Her 4 senaryoda DoS saldırı tespiti daha önceki periyotlardaki toplam elde edilen paket sayısı ile yapılmıştır.

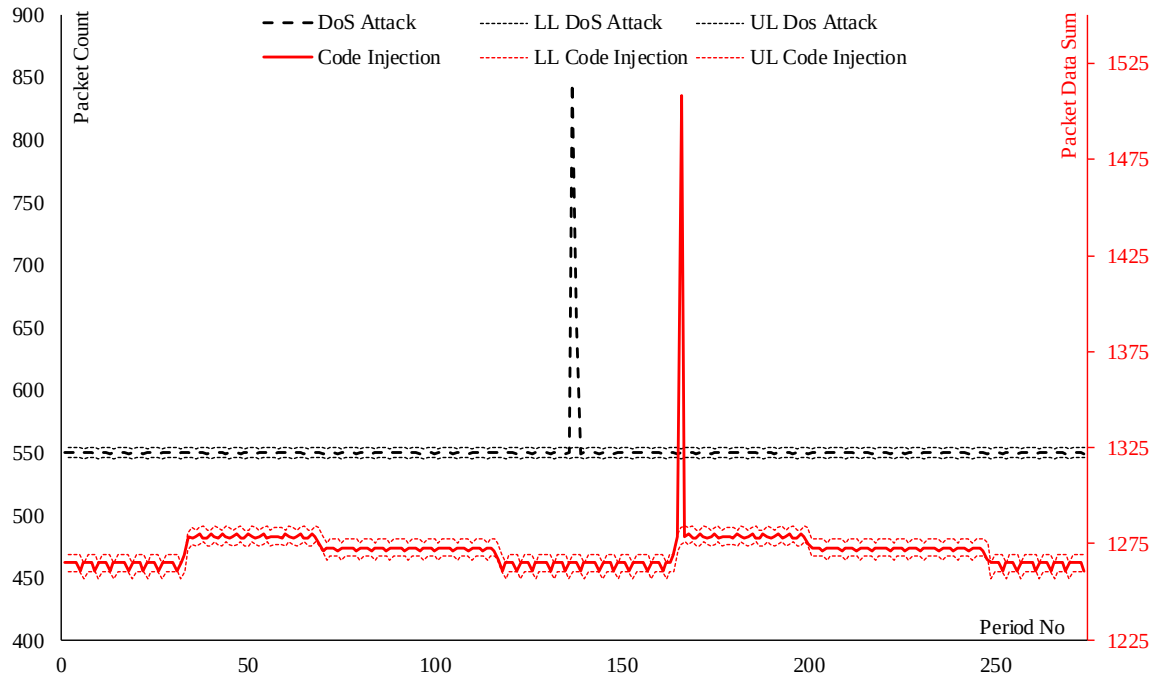
DoS saldırısı, her senaryo için sırasıyla 16, 4, 137 - 138 ve 27 - 29 numaralı periyotlarda rastgele olarak geliştirilmiştir. Kod enjeksiyonu ise her periyottaki yazma komutlarının veri alanları incelenerek belirlenmiştir. Saldırı, her 4 senaryo için sırasıyla 39, 15 - 16, 166 ve 66 - 68 numaralı periyotlarda rastgele olarak geliştirilmiştir. Grafiklerde saldırıların olduğu bölgeler alt ve üst limitlerin aşıldığı aralıklardır. İletim sırasında farklı örüntüler dahi olsa saldırılar ilgili istatistikleri değiştirdiğinden tespit yapılmıştır. Saldırıların tespit hassasiyeti LL ve UL değerlerinin değiştirilmesiyle artırılabilir. Hassasiyet çok yüksek belirlenirse yanlış alarmlar artarken, çok düşük belirlenmesi ise saldırıların tespit edilememesine yol açar.



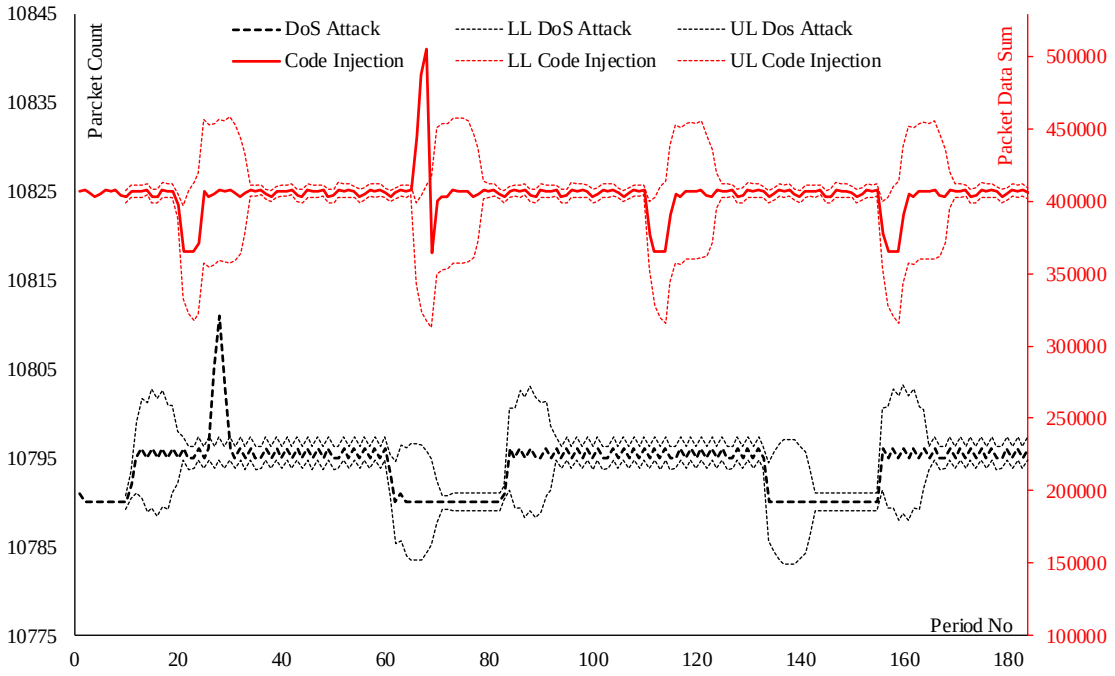
Şekil 5.10. Program 1 - Anomali tespiti



Şekil 5.11. Program 2 - Anomali tespiti



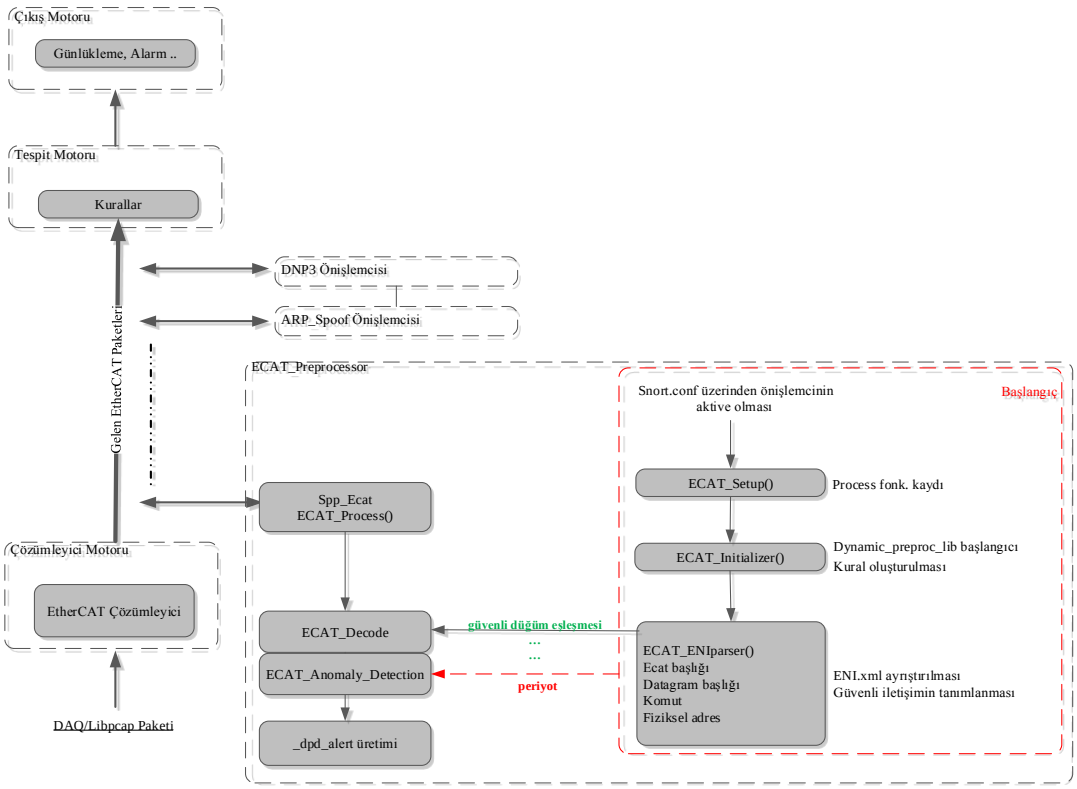
Şekil 5.12. Program 3 - Anomali tespiti



Şekil 5.13. Program 4 - Anomali tespiti

Önişlemci üzerinden yapılan periyot tespiti sonuçları incelendiğinde, 0,95 ve 0,99 doğruluğa sahip otokorelasyon ile tespit gerçekleştirildiği görülmüştür. Çalışmanın

Snort IDS/IPS'e adapte edilmesi ile periyot bilindiğinde, EKS ortamında sorunsuz şekilde çalıştığı ve çoğunlukla çevrimsellik içeren EtherCAT saha veriyolu trafiğinde yer alan ilgili saldırıların tespit edilebildiği anlaşılmıştır. Böylece Snort önişlemcisi Şekil 5.14.'deki halini almıştır.



Şekil 5.14. ECAT önişlemcisinde periyodun kullanılması

Sonuç olarak, EtherCAT saha seviyesindeki iletişimin çevrimsellik içermesinden dolayı hat üzerindeki tekrarlar yakalanabilmekte olup, saldırı gerçekleşmesi durumunda çevrimsel iletişim değişeceğinden, periyot değerleri kullanılarak anomali tespitinin %95 - %99 başarı ile yapılabilindiği görülmüştür. Bu çözüm ise kural tabanlı olan Snort sistemine yeni bir kural eklenmesiyle gerçekleştirilmiştir. Ağ trafiğinde tek bir istek paketinin artışı veya azalışı bile istatistikleri değiştirdiğinden tespiti mümkündür. Çalışma sıfırıncı-gün saldırılarına da çözüm sunmaktadır. Sıfırıncı-gün saldırıları öncelikle bir uç düğüm tespit eder, sonrasında ağı tarar ve içindeki kötücül yazılımı yayma yoluna gider. Nihayetinde sistemi durdurur veya bozar. Son aşama saha seviyesinde çevrim zamanı içinde veri transferi ile mümkün olduğundan periyot

içindeki değerler değişecektir. Bu da yapılan çalışma ile tespit edilebilir. Diğer yandan sunulan pasif izleme yöntemi, geliştirilen algoritmalar, sıfırıncı-gün saldırılarına getirilen çözüm çalışmaları, şifreleme, kimlik doğrulama ve yetkilendirme içermeyen diğer EKS protokollerine de uygulanabilmektedir. Senkron iletişim yaklaşımı barındıran çalışan sistemler üzerine pasif izleme yoluyla adapte edilerek sistem üzerine yük getirmeden gerçek sistemlerde uygulanabilirler.

BÖLÜM 6. ETHERCAT TABANLI ANOMALİ TESPİTİNDE MAKİNA ÖĞRENMESİ YÖNTEMLERİ

Bu bölümde yapılan çalışmalar EtherCAT iletişimdeki normal ve anormal davranışların tespitine dayanmaktadır. Literatürde normal ve anormal kavramları daha önce belirtildiği gibi farklı tanımlamalara sahiptir. Çalışma kapsamında, çevrimsel bir EtherCAT iletişimindeki 1 çevrim boyunca gözlemlenen durum, değişken ve ağ dinamiklerinin olası değişimleri normal davranış olarak tanımlanmaktadır. Belirli bir oranın üzerindeki değişimler, ani değişimler, kesintiler, normal olduğu halde bu çevrim içerisinde gözlemlenmemesi gereken durumlar ile 2 veya 2'den fazla çevrim toplamında gözlemlenebilen değişimler ise anormal davranış olarak değerlendirilecektir. Wool ve ark. Modbus üzerindeki anomali tespitinde DFA kullanarak sistemi modellemiş ve modeldeki beklenmeyen geçişleri anomali olarak kabul etmişlerdir. Modbus başlığındaki; "transaction identifier", "unit identifier", "fonksiyon kodu" ve "verilere" bakılmış, kabul paketleri gelen "echo" paketlerinden, hatalar ise "echo" içindeki 2 baytlık hata değeri ile tespit edilmiştir [60]. Başka bir çalışmalarında ise kaydedici adresleri içeren paketleri inceleyip kaydedicileri geliştirdikleri bir algoritmayla sınıflandırmışlardır [119]. Benzer yapılar EtherCAT protokol kabullerinde de var olduğundan, bu çalışmada öncelikle EtherCAT protokolü tabanlı, laboratuvar ölçekli hazırlanmış bir test ortamında gerçek bir kritik altyapılı sistem oluşturulmuştur. Test ortamı üzerinde PLC programı geliştirilerek sistem üzerinde sürekli çalışan bir proses oluşturulmuş, anomali olarak çeşitli olaylar gerçeklenmiş, olaylardan veriseti elde edilerek öznitelikler belirlenmiş ve azaltılmış, ve son olarak makine öğrenmesi yöntemleriyle sistemin normal ve anormal davranışları türleriyle tespit edilip, yapılan çalışmalar sonraki bölümlerde sırasıyla detaylandırılmıştır.

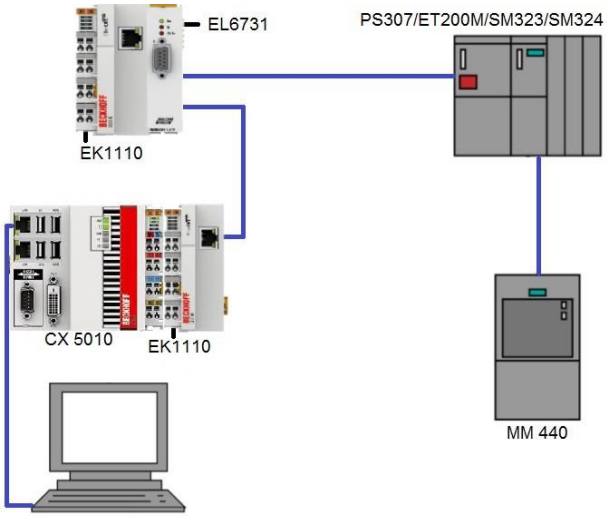
6.1. Test Ortamı

Test ortamı, daha önce oluşturulan bir otomasyon sisteminin EtherCAT bileşenleri ile adapte edilmesiyle gerçeklenmiştir (Şekil 6.1.). Test ortamındaki bileşenler SIEMENS firmasına ait 1 – 6 numaralı donanımlardan oluşmaktadır (Tablo 6.1.). Tüm bu donanımlar kendi aralarında PROFIBUS protokolü üzerinden haberleşme sağlamaktadır. EtherCAT üzerinde çalışacak sisteme adapte edilmeleri için 7 – 10 numaralı bileşenler eklenerek mühendislik istasyonu ile haberleşmeleri sağlanmıştır. EL6731 modülü EtherCAT protokolü üzerinden gelen paketleri PROFIBUS protokolüne dönüştürmek için kullanılmaktadır. ET2000 cihazı ise gelen/giden iletişim paketlerinin tek bir hat üzerinden izlenmesi için kullanılmaktadır.

Tablo 6.1. Test ortamı bileşenleri

No	Bileşen	Adet	Model	Açıklama
1	PS 307 2A	1	307-1BA00-0AA0	Güç kaynağı
2	ET200M	1	153-1AA03-0XB0	Uzak I/O modülü
3	SM 323	2	323-1BH01-0AA0	DI/DO modülü
4	SM 334	1	334-0KE00-0AB0	AI/AO modülü
5	Micromaster 440	1	6SE6440-2UC17-5A A1	Sürücü
6	PC	1		Operatör kontrolü
7	Beckhoff PLC	1	CX5010	PLC
8	ELxx	2	EK1110	EtherCAT uzatma modülü
9	Beckhoff ETxx	1	ET2000	Tab cihazı
10	ELxx	1	EL6731	ProfiBus efendi istasyonu

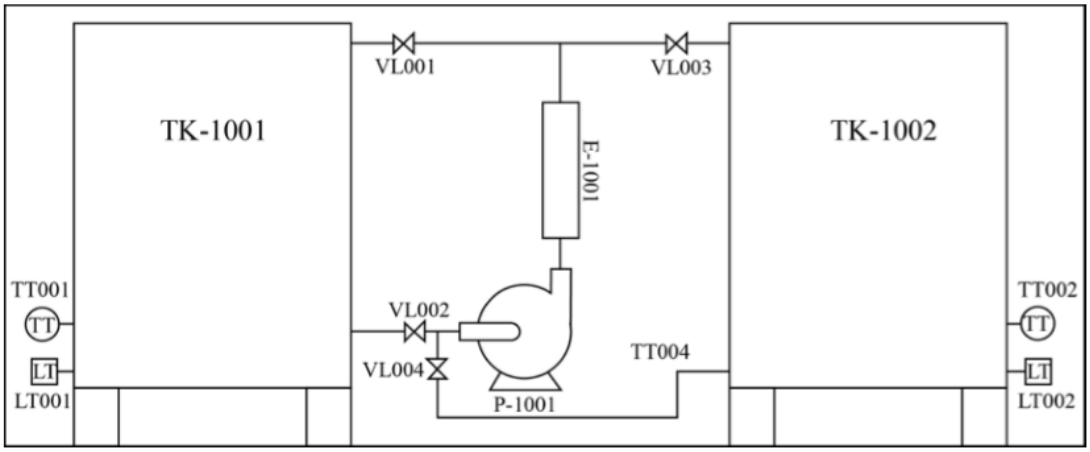
Oluşturulan test ortamı Şekil 6.1. ve Şekil 6.2. üzerinde gösterilmiştir. PLC programlama ve tüm yapılandırmalar mühendislik istasyonu tarafından yapılmakta ve PLC donanımı EL6731 modülü yardımıyla PROFIBUS sistemi üzerindeki I/O birimlerine komutlar göndermektedir. MM 440 sürücüsüne pompa motoru bağlıdır. Diğer I/O birimlerine ise algılayıcılar bağlıdır.



Şekil 6.1 Test ortamı – 1

Şekil 6.2. Test ortamı – 2

MM 440 sürücüsünün ve diğer I/O birimlerin kontrol ettikleri aktüator/algılayıcı seviyesindeki bileşenler Şekil 6.3. üzerinde gösterilmiştir. Sistem üzerinde kontrolü sağlanan 2 adet tank (TK-1001/TK-1002), 2 adet seviye algılayıcı (LT001/LT002), 2 adet sıcaklık algılayıcı (TT001/TT002), 2 adet elektrikli (VL002/VL004) ve 2 adet selenoit (VL001/VL003) olmak üzere 4 adet vana, 1 su ısıtıcı (E-1001) ve 1 pompa (P-1001) bulunmaktadır.



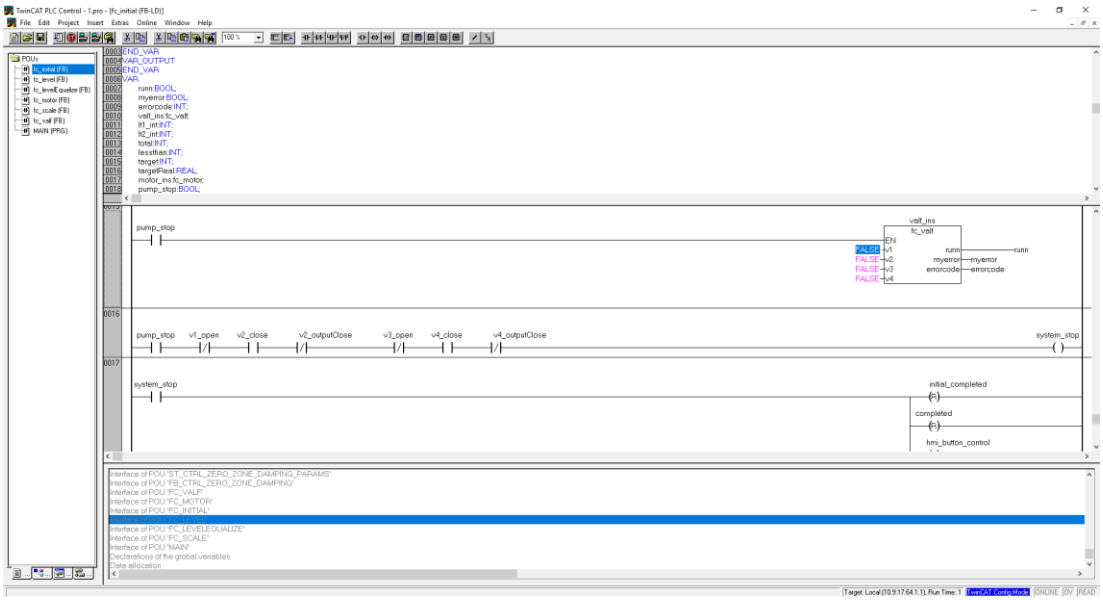
Şekil 6.3. Su seviye kontrol otomasyon sistemi bileşenleri

6.2. Su Seviyesi Kontrol Otomasyonu PLC Programı

Şekil 6.3.'deki sisteme, su otomasyonu PLC programı geliştirilip yüklenmiştir. Buna göre sistem ilk enerjilendiğinde tankların içindeki su seviyelerini ölçmekte, su miktarı çok olan tanktan diğerine su akışı yapacak şekilde başlangıç konumuna gelmektedir. Önce uygun vanalar açılmakta (selenoid ve elektrikli valfler), sonrasında motor hazır komutundan çalışma komutuna (başlatma=0x00207F0C) geçmektedir. 2000 devirde çalıştırılan motor, su seviyesi çok olan tankın suyu %5 kalana kadar çalışmaktadır. Ardından motor tekrar durarak hazır duruma geçmekte (P1 durdurma:0x00007E04) ve vanalar (V2-V3 veya V1-V4) kapatılmaktadır. Duran sistem 30 saniye kadar beklemekte, ardından işlemin tersi yönde harekete başlamaktadır. Böylelikle bir çevrim tamamlanarak, sürekliliği olan bir proses oluşturulmuştur (Şekil 6.4.). Algılayıcılardan alınan su seviye bilgisi için ölçeklendirme yapılmıştır. Buna göre

aşağıdaki formül kullanılarak gelen veriler 0-100 aralığına (LO_LIM, HI_LIM) dönüştürülmüştür.

OUT = [((FLOAT (IN) - K1)/(K2 - K1)) * (HI_LIM - LO_LIM)] + LO_LIM



Şekil 6.4. PLC programı ekran görüntüsü

Kullanılan algılayıcılar giriş değerlerini UNIPOLAR olarak göndermektedir. Buna göre girdi değerleri 0-27648 arasında; K1 = 0.0 ve K2 = +27648.0 olarak elde edildiği kabul edilmiştir. Ayrıca sürücü verileri hariç PROFIBUS ortamından elde edilen veriler küçük sonlu olarak TwinCAT PLC tarafından kullanılan formata dönüştürülmüştür.

6.3. Olay Oluşturulması

Sistemden veriseti elde ederek saldırı tespiti yapılması için öncelikle, ağ trafiği normal olarak kaydedilmiş, ardından kesintisiz bir şekilde anomali olarak kabul gören 15 farklı olay literatürdeki çalışmalar göz önünde tutularak uygulanmıştır. Bu olaylar Tablo 6.2.' de detaylandırılmıştır.

Tablo 6.2. Saldırı türleri

Saldırı	Grup	Kategori	Açıklama	Hedef
A1	G4	Giriş akışı	VL001 kapatılması	TK-1001 ye akışın durdurulması
A2			VL003 kapatılması	TK-1002 ye akışın durdurulması
A3			VL001 açılması	TK-1001 ve TK-1002 arasındaki akışın başlatılması
A4			VL003 açılması	TK-1002 ve TK-1002 arasındaki akışın başlatılması
A5		Çıkış akışı	P-1001 kapatılması	Pompalamanın durdurulması
A6			P-1001 açılması	Pompalamanın başlatılması
A7			VL002 kapatılması	TK-1001 den P-1001 ye pompalamanın durdurulması
A8			VL004 kapatılması	TK-1002 den P-1001 ye pompalamanın durdurulması
A9	G3	Tank seviyesi	LT001 5 olarak değiştirilmesi	İşin durdurulması(TK-1001 to TK-1002)
A10			LT002 5 olarak değiştirilmesi	İşin durdurulması(TK-1002 to TK-1001)
A11	G1	Sistem	Baağlantı hatası	Hataya zorlama
A12			Yeniden bağlantı kurulması	Bağlantı hatasından sonra sistemin çalıştırılması
A13	G4		Durum makinesının değiştirilmesi	Sistem durumunun değiştirilmesi
A14	G2		PLC programının yüklenmesi ve çalıştırılması	Çalışan işin değiştirilmesi
A15			Yeniden oynatma	Ani artış anomalisi

Buna göre saldırılar 4 farklı grupta oluşturulmuştur (G1,G2,G3,G4). Bunlar,

G1: Ağdaki donanım, konfigürasyon değişimlerinden veya kesintilerden kaynaklı olaylar

G2: Yığılmalardan ötürü aniden yükselen, yazılım tabanlı bir sisteme veya web siteye erişimlerin fazlalaşması gibi zamanla düzene giren anomaliler

G3: Ölçüm hatalarından kaynaklı anomaliler

G4: Ağın kötüye kullanıldığı saldırılar

Anomalilerin ise sistem, tank seviyesinin manipüle edilmesi, gelen ve giden akış olmak üzere 4 farklı yapıya etkisi oluşmaktadır.

Normal ağ trafiği için sistem 1 saat çalıştırılmıştır ve bu veri grubu A0 olarak gösterilmiştir. Durum makinesı için OP durumundan sırasıyla INIT, PREOP, SAFEOP, OP durumlarına geçiş yapılmış, link hatası için I/O birimleriyle olan bağlantı kesilmiş ardından tekrar bağlantı aktif hale getirilmiştir. Ayrıca sürücüye bağlı motoru çalıştıran bir PLC programı geliştirilip sisteme yüklenmesi esnasındaki durumlar da alınmıştır. Sistemin izlenmesi için EL6731 modülü ve PLC arasına ET2000 Tab cihazı yerleştirilmiştir.

6.4. Öznitelik Belirlenmesi, Tanımlayıcı İstatistikleri ve Özniteliklerin Azaltılması (Regresyon Analizi)

Öznitelik belirlenmesi ve azaltılması için 3 aşamalı bir işlem uygulanmıştır.

a. Elde edilen olaylar Wireshark üzerinde açılmış ve uygun formata getirilmiştir. Paketlerin uygun şekilde manipüle edilmesi için Wireshark üzerinde daha önceki bölümde açıklanan Post-Dissector eklentisi geliştirilmiştir. Bu eklenti aktif edildiğinde PCAP kayıtlarındaki komutlar, "padding" verileri, veri uzunlukları, kullanılan kaydediciler ayrıştırılmakta ve üst panelde sunmaktadır. Böylece istatistiki değerler paket bazlı belirlenebilmiştir.
b. Wireshark üzerinden "csv" formatında dışa aktarılan paketlerin öznitelikleri geliştirilen bir ayrıştırıcı programına girdi olarak alınmış ve program yardımıyla belirlenmiştir. Belirlenen öznitelikler Tablo 6.3.'deki gibidir. Öznitelikler, tabloda görüldüğü gibi belirli bir "pencere" içindeki toplam ve ortalama gibi değerleri tespit edilerek değerlendirilmiştir. Pencere boyutu 1 sn. olarak alınmıştır.
c. Özniteliklerin 15 saldırı ve normal davranış durumu üzerindeki etkisini tespit etmek için her bir olayda etkili olan öznitelikleri belirleme ve gereksiz öznitelikleri azaltma işlemine tabi tutulmuştur. Öznitelik seçiminde regresyon analizi kullanılarak öznitelik boyutu indirgenmiştir.

Tablo 6.3. Analizde kullanılan öznitelikler

Öznitelik	Açıklama
1	Pencere içindeki toplam paket sayısı
2	Pencere içindeki paket sayılarının ortalaması
3	Pencere içindeki paketlerin padding veri boyutlarının toplamı
4	Pencere içindeki paketlerin padding veri boyutlarının ortalaması
5	Pencere içindeki throughput (total) (bayt/sn)
6	Pencere içindeki throughput ortalaması (average)
7	Pencere içindeki farklı efendi kaynak adres sayısı
8	Pencere içindeki farklı efendi hedef adres sayısı
9	Pencere içindeki Nope komutu veri boyutu ortalaması
10	Pencere içindeki Nope komutu veri boyutu toplamı
11	Pencere içindeki APRD komutu offset (register) değerleri ortalaması
12	Pencere içindeki APRD komutu offset (register) değerleri toplamı
13	Pencere içindeki APWR komutu offset (register) değerleri ortalaması
14	Pencere içindeki APWR komutu offset (register) değerleri toplamı
15	Pencere içindeki APRW komutu veri boyutu ortalaması
16	Pencere içindeki APRW komutu veri boyutu toplamı
17	Pencere içindeki FPRD komutu offset (register) değerleri ortalaması
18	Pencere içindeki FPRD komutu offset (register) değerleri toplamı
19	Pencere içindeki FPWR komutu offset (register) değerleri ortalaması
20	Pencere içindeki FPWR komutu offset (register) değerleri toplamı
21	Pencere içindeki FPRW komutu veri boyutu ortalaması
22	Pencere içindeki FPRW komutu veri boyutu toplamı
23	Pencere içindeki BRD komutu offset (register) değerleri ortalaması
24	Pencere içindeki BRD komutu offset (register) değerleri toplamı
25	Pencere içindeki BWR komutu offset (register) değerleri ortalaması
26	Pencere içindeki BWR komutu offset (register) değerleri toplamı
27	Pencere içindeki BRW komutu veri boyutu ortalaması
28	Pencere içindeki BRW komutu veri boyutu toplamı
29	Pencere içindeki LRD komutu veri boyutu ortalaması
30	Pencere içindeki LRD komutu veri boyutu toplamı
31	Pencere içindeki LWR komutu veri boyutu ortalaması
32	Pencere içindeki LWR komutu veri boyutu toplamı
33	Pencere içindeki LRW komutu veri boyutu ortalaması
34	Pencere içindeki LRW komutu veri boyutu toplamı
35	Pencere içindeki ARMW komutu offset (register) değerleri ortalaması
36	Pencere içindeki ARMW komutu offset (register) değerleri toplamı
37	Pencere içindeki FRMW komutu veri boyutu ortalaması
38	Pencere içindeki FRMW komutu veri boyutu toplamı
39	Pencere içindeki NOP komutu data alanının sayısal değerinin ortalaması
40	Pencere içindeki NOP komutu data alanının sayısal değerinin toplamı
41	Pencere içindeki LRD komutu data alanının sayısal değerinin ortalaması
42	Pencere içindeki LRD komutu data alanının sayısal değerinin toplamı
43	Pencere içindeki LWR komutu data alanının sayısal değerinin ortalaması
44	Pencere içindeki LWR komutu data alanının sayısal değerinin toplamı
45	Pencere içindeki APRD komutu data alanının sayısal değerinin ortalaması
46	Pencere içindeki APRD komutu data alanının sayısal değerinin toplamı
47	Pencere içindeki LRW komutu data alanının sayısal değerinin ortalaması
48	Pencere içindeki LRW komutu data alanının sayısal değerinin toplamı
49	Pencere içindeki APRW komutu data alanının sayısal değerinin toplamı
50	Pencere içindeki APRW komutu data alanının sayısal değerinin toplamı
51	Pencere içindeki FPRW komutu data alanının sayısal değerinin ortalaması
52	Pencere içindeki FPRW komutu data alanının sayısal değerinin toplamı
53	Pencere içindeki BRW komutu data alanının sayısal değerinin ortalaması
54	Pencere içindeki BRW komutu data alanının sayısal değerinin toplamı

Tüm verisetinde 1sn pencere boyutu ile oluşturulmuş tanımlayıcı istatistikler Tablo 6.4. ’te verilmiştir. Tanımlayıcı istatistiklerde yer alan APRW komutu (öznitelik 15, 16), FPRW komutu (öznitelik 21, 22), BRW komutu (öznitelik 27, 28), FRMW komutu (öznitelik 37, 38), LWR veri (öznitelik 43, 44), APRD veri (öznitelik 45, 46), APRW veri (öznitelik 49, 50), FPRW veri (öznitelik 51, 52) ve BRW veri (öznitelik 53, 54) alanlarının veri içermediği görülmüştür ve verisetinden çıkartılarak ilk indirgeme işlemi yapılmıştır. Böylece 50 öznitelikten 32 özniteliğe indirgenmiştir. Bu aşamadan sonra regresyon denklemi yardımı ile 2 farklı yoldan öznitelikler seçilmiştir. Bir öznitelik seçiminde değişkenin anlamlı olup olmadığı *p* değeri ile belirlenmektedir [139]–[141].

a. Her saldırı durumu için alt verisetleri hazırlanmıştır. Bu alt verisetlerinde tek bir saldırı ve saldırı olmama durumu bulunmaktadır. Buna uygun olarak her saldırı için uygun değişkenler tek tek belirlenmeye çalışılmıştır (Tablo 6.5.). Buna göre saldırı bazlı en anlamlı değişkenler sırasıyla LRW veri toplamı, ARMW komut toplamı, APRD komut toplamı, APWR komut toplamı, LRW veri ortalaması, FPRD komut toplamı, SumPacketLength öznitelikleri olduğu görülmektedir.

b. Tüm saldırı ve normal davranışları barındıran bir örneklem veriseti rastsal olarak ana verisetinden çekilip, regresyon denklemi ile anlamlı değişkenler belirlenmeye çalışılmıştır. Bu alt verisetinde 180 adet veri, en anlamlı 32 öznitelik ile çalışılmıştır. Bu 32 öznitelik anlamsız öznitelikler çıkartılarak 11 adete düşürülmüştür. Bu 11 adet özniteliğe 1 nolu adımda anlamlı olan ve bu denklemde yer almayan öznitelikler eklenerek, regresyon denklemi yeniden oluşturulmuştur. 16 değişkenli bu model üzerinde anlamsız değişkenler olduğu görülmüş ve bu değişkenler modelden çıkartılarak oluşturulan yeni model sonucu Tablo 6.6.’da gösterilmiştir. Regresyon denklemi sonunda saldırı ve normal durum şartlarını belirlemede SumPadBayt, SumPacketLength, FPWR komut toplamı, BRD komut toplamı, LRD komut toplamı, LRD veri toplamı, LRW veri toplamı, NOP veri ortalaması, LRD veri ortalaması, LRW veri ortalaması özniteliklerinin anlamlı olduğu görülmüştür. Oluşturulan denklemin

korelasyon katsayısı ise 0,837 olmuştur. Bu sonuç, ilgili öznitelikler ile saldırıların ilişkisinin güçlü olduğunu göstermektedir. Aynı şekilde denklemin anlamlılık düzeyi $1{,}2 \cdot 10^{-19}$ olarak bulunmuştur. Bu da anlamlılık düzeyinin sıfıra çok yakın olduğunu ve denklemin istatistiksel olarak anlamlı olduğunu göstermektedir.

Tablo 6.4. Pencere verisetinin tanımlayıcı istatistikleri

Değişken	Ortalama	Standart Hata	Orta Değer	Mod	Standart Dağılım	Aralık	En düşük	En yüksek	Toplam	Adet	En Yüksek (100)	En Düşük (100)
SumPadBayt	3.518,21	478,49	1.672	1.672	34.992,31	800.500	864	801.364	18.815.391	5.348	3.344	1.640
AvgPadBayt	16,37	0,02	16,23	16,23	1,75	27,95	16	43,95	87.566,86	5.348	16,24	16,23
SumPacketLength	33.294,23	3.324,17	21.420	21.420	243.096,40	5.622.824	11.306	5.634.130	178.057.550	5.348	42.630	21.210
AvgPacketLength	207,51	0,10	207,96	207,96	7,55	147,95	76,78	224,73	1.109.749,71	5.348	208,67	207,94
Sum NOP (Cmd0)	605,08	58,68	400	400	4.291,04	99.508	-	99.508	3.235.971	5.348	400	396
Avg NOP (Cmd0)	3,87	0	3,88	3,88	0,33	22,08	-	22,08	20.716,64	5.348	3,92	3,88
Sum APRD (Cmd1)	638,79	208,83	-	-	15.271,77	653.296	-	653.296	3.416.247	5.348	-	-
Avg APRD (Cmd1)	1,63	0,55	-	-	40,19	1.169,75	-	1.169,75	8.693,89	5.348	-	-
Sum APWR (Cmd2)	71,26	33,81	-	-	2.472,58	100.247	-	100.247	381.094	5.348	-	-
Avg APWR (Cmd2)	0,07	0,03	-	-	2,45	138,16	-	138,16	391,30	5.348	-	-
Sum APRW (Cmd3)	-	-	-	-	-	-	-	-	-	5.348	-	-
Avg APRW (Cmd3)	-	-	-	-	-	-	-	-	-	5.348	-	-
Sum FPRD (Cmd4)	11.880,73	300	10.752	10.752	21.938,73	609.280	-	609.280	63.538.166	5.348	21.504	10.752
Avg FPRD (Cmd4)	106,04	1,12	104,39	104,39	81,74	2.544,20	-	2.544,20	567.111,32	5.348	105,41	104,39
Sum FPWR (Cmd5)	141,96	65,02	-	-	4.755,08	223.008	-	223.008	759.196	5.348	-	-
Avg FPWR (Cmd5)	0,34	0,19	-	-	13,61	690,45	-	690,45	1.828,32	5.348	-	-
Sum FPRW (Cmd6)	-	-	-	-	-	-	-	-	-	5.348	-	-
Avg FPRW (Cmd6)	-	-	-	-	-	-	-	-	-	5.348	-	-
Sum BRD (Cmd7)	45.817,77	4.458,05	30.400	30.400	326.017,07	7.561.696	912	7.562.608	245.033.424	5.348	30.400	30.096
Avg BRD (Cmd7)	294,09	0,23	295,15	295,15	16,91	303,27	0,73	304	1.572.784,30	5.348	298,04	295,06
Sum BWR (Cmd8)	795,62	16,35	768	768	1.195,39	62.473	-	62.473	4.254.996	5.348	768	768
Avg BWR (Cmd8)	7,35	0,01	7,46	7,46	1,08	36,64	-	36,64	39.308,16	5.348	7,53	7,46
Sum BRW (Cmd9)	-	-	-	-	-	-	-	-	-	5.348	-	-
Avg BRW (Cmd9)	-	-	-	-	-	-	-	-	-	5.348	-	-
Sum LRD (Cmd10)	5.574,60	542,59	3.700	3.700	39.679,79	920.449	-	920.449	29.812.973	5.348	3.700	3.663
Avg LRD (Cmd10)	35,79	0,03	35,92	35,92	2,09	37	-	37	191.408,71	5.348	36,27	35,91
Sum LWR (Cmd11)	1.657,31	161,31	1.100	1.100	11.796,69	273.647	-	273.647	8.863.272	5.348	1.100	1.089
Avg LWR (Cmd11)	10,64	0,01	10,68	10,68	0,62	11	-	11	56.905,21	5.348	10,78	10,68
Sum LRW (Cmd12)	5.423,91	527,93	3.600	3.600	38.607,36	895.572	-	895.572	29.007.072	5.348	3.600	3.564
Avg LRW (Cmd12)	34,82	0,03	34,95	34,95	2,04	36	-	36	186.235,24	5.348	35,29	34,94
Sum ARMW (Cmd13)	359.889,78	34.200,43	232.000	232.000	2.501.078,36	57.714.640	-	57.714.640	1.924.690.560	5.348	459.360	229.680
Avg ARMW (Cmd13)	2.246,99	1,52	2.252,43	2.252,43	111,24	2.320	-	2.320	12.016.877,45	5.348	2.274,51	2.251,76
Sum FRMW (Cmd14)	-	-	-	-	-	-	-	-	-	5.348	-	-
Avg FRMW (Cmd14)	-	-	-	-	-	-	-	-	-	5.348	-	-

Tablo 6.4. Pencere verisetinin tanımlayıcı istatistikleri (Devamı)

Değişken	Ortalama	Standart Hata	Orta Değer	Mod	Standart Dağılım	Aralık	En düşük	En yüksek	Toplam	Adet	En Yüksek (100)	En Düşük (100)
Sum NOP (Data1)	0,36	0,25	-	-	18,57	960	-	960	1.952	5.348	-	-
Sum LRD (Data2)	0	0	-	-	0,12	6	-	6	12	5.348	-	-
Sum LWR (Data3)	-	-	-	-	-	-	-	-	-	5.348	-	-
Sum APRD (Data4)	-	-	-	-	-	-	-	-	-	5.348	-	-
Sum LRW (Data5)	5.455,53	585,87	3.700	3.700	42.844,73	993.280	-	993.280	29.176.151	5.348	5.300	2.500
Sum APRW (Data6)	-	-	-	-	-	-	-	-	-	5.348	-	-
Sum FPRW (Data7)	-	-	-	-	-	-	-	-	-	5.348	-	-
Sum BRW (Data8)	-	-	-	-	-	-	-	-	-	5.348	-	-
Avg NOP (Data1)	0	0	-	-	0,04	2,85	-	2,85	3,37	5.348	-	-
Avg LRD (Data2)	0	0	-	-	0	0	-	0	0,01	5.348	-	-
Avg LWR (Data3)	-	-	-	-	-	-	-	-	-	5.348	-	-
Avg APRD (Data4)	-	-	-	-	-	-	-	-	-	5.348	-	-
Avg LRW (Data5)	33,30	0,08	35,92	35,92	6,19	54,37	-	54,37	178.076,55	5.348	46,60	24,27
Avg APRW (Data6)	-	-	-	-	-	-	-	-	-	5.348	-	-
Avg FPRW (Data7)	-	-	-	-	-	-	-	-	-	5.348	-	-
Avg BRW (Data8)	-	-	-	-	-	-	-	-	-	5.348	-	-
GroupNumber	1,21	0,02	1	1	1,50	15	1	16	6.483	5.348	7	1
Attack	0,02	0	-	-	0,15	1	-	1	130	5.348	1	-

Tablo 6.5. Saldırı bazlı etkili değişkenlerin anlamlılıkları

p-değer tablosu	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
SumPadBayt	×	×	×	×	×	×	×	×	×	×	×	×	$3{,}1\cdot10^{-150}$	×	0,0018
AvgPadBayt	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
SumPacketLength	×	×	×	×	×	×	×	×	×	×	$3{,}8\cdot10^{-101}$	×	$5{,}4\cdot10^{-151}$	$2{,}51\cdot10^{-82}$	0,0018
AvgPacketLength	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
Sum NOP (Cmd0)	×	×	×	×	×	×	×	×	×	×	×	×	$6{,}5\cdot10^{-153}$	×	×
Avg NOP (Cmd0)	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
Sum APRD (Cmd1)	×	×	×	×	×	×	×	×	×	×	$3{,}7\cdot10^{-104}$	×	$1{,}6\cdot10^{-165}$	$2{,}11\cdot10^{-83}$	0,0017
Avg APRD (Cmd1)	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
Sum APWR (Cmd2)	×	×	×	×	×	×	×	×	×	×	×	×	$1{,}9\cdot10^{-149}$	$9{,}8\cdot10^{-158}$	×
Avg APWR (Cmd2)	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
Sum FPRD (Cmd4)	×	×	×	×	×	×	×	×	×	×	$1{,}51\cdot10^{-96}$	×	$1{,}5\cdot10^{-143}$	$3{,}99\cdot10^{-82}$	0,0018
Avg FPRD (Cmd4)	×	×	×	×	×	×	×	×	×	×	×	×	$2{,}37\cdot10^{-8}$	×	$8{,}21\cdot10^{-15}$
Sum FPWR (Cmd5)	×	×	×	×	×	×	×	×	×	×	×	×	$1{,}3\cdot10^{-155}$	$7{,}54\cdot10^{-82}$	×
Avg FPWR (Cmd5)	×	×	×	×	×	×	×	×	×	×	×	×	$1{,}1\cdot10^{-157}$	×	×
Sum BRD (Cmd7)	×	×	×	×	×	×	×	×	×	×	×	×	$5{,}1\cdot10^{-150}$	$1{,}6\cdot10^{-157}$	×
Avg BRD (Cmd7)	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
Sum BWR (Cmd8)	×	×	×	×	×	×	×	×	×	×	×	×	$3{,}4\cdot10^{-135}$	×	0,0019
Avg BWR (Cmd8)	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
Sum LRD (Cmd10)	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
Avg LRD (Cmd10)	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
Sum LWR (Cmd11)	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
Avg LWR (Cmd11)	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
Sum LRW (Cmd12)	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
Avg LRW (Cmd12)	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
Sum ARMW (Cmd13)	×	×	×	×	×	×	×	×	×	×	$4{,}6\cdot10^{-104}$	×	$9\cdot10^{-133}$	$1{,}17\cdot10^{-84}$	0,0018
Avg ARMW (Cmd13)	×	×	×	×	×	×	×	×	×	×	×	×	$2{,}13\cdot10^{-8}$	×	$4{,}11\cdot10^{-06}$
Sum NOP (Data1)	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
Sum LRD (Data2)	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
Sum LRW (Data5)	0,0007	0,0003	$6{,}54\cdot10^{-17}$	$7{,}89\cdot10^{-16}$	0,0002	0,0749	$3{,}52\cdot10^{-14}$	$2{,}36\cdot10^{-8}$	0,0364	$3{,}1\cdot10^{-6}$	×	0,0009	×	0,0020	0,0017
Avg NOP (Data1)	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
Avg LRD (Data2)	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
Avg LRW (Data5)	×	×	×	×	×	×	×	×	×	×	×	×	×	×	0,0017

× : p>0,1

Tablo 6.6. Örneklem veriseti için regresyon denklemi ve özniteliklerin anlamlılıkları

	Katsayılar	Standart Hata	t İstatistiği	P-Değeri
Intercept	23,97471	2,924345	8,198316	6E-14
SumPadBayt	-0,00887	0,002315	-3,83273	0,000179
SumPacketLength	0,005112	0,00133	3,842959	0,000172
Sum FPWR komutu	-0,00017	9,35E-05	-1,83161	0,06878
Sum BRD komutu	-0,00912	0,002433	-3,75025	0,000243
Sum LRD komutu	0,044081	0,01264	3,48731	0,000623
Sum LRD data	64,27536	21,23337	3,027091	0,002858
Sum LRW data	0,006807	0,001494	4,556349	9,97E-06
Avg NOP data	33,16901	9,235732	3,591379	0,000432
Avg LRD data	-107208	35206,69	-3,04509	0,002701
Avg LRW data	-0,66086	0,115392	-5,72704	4,61E-08

6.5. Makine Öğrenmesi ile EtherCAT Anomali Tespiti

Bu bölümde regresyon analizi sonucu çıkan bilgiler kullanılarak öznitelikler ile saldırıların tespiti çalışılmıştır. Toplam 5.348 adet veriden oluşan (5.348 sn.) veriseti eğitim ve test olarak %70 ve %30 oranlarında ikiye bölünmüştür. Eğitim verisetinde 3.744 adet verinin 94 tanesi saldırı, 3.653 tanesi normal ağ trafik verisidir. Test tarafında 43 tane veri saldırı, 1.565 tane veri ise normal ağ verisi olarak gösterilmiştir. Eğitim ve test verisinin ayrıştırılmasında rastsal örneklem yöntemi seçilmiştir fakat her saldırıdan hem eğitim hem de test ortamında bulunması gerektiğinden dolayı saldırı verileri ikiye bölünerek dağıtılmıştır. Örnek olarak bir saldırının 15 adet verisi bulunuyor ise bu verinin 11 tanesi eğitim verisetinde, 4 tanesi de test verisetinde olacak şekilde bölünmüştür. Bazı saldırıların veri sayısı çok düşük olduğundan dolayı aynı saldırı her iki verisetinde de bulunmaktadır. Veriseti özeti Tablo 6.7.'de gösterilmiştir.

Tablo 6.7. Veriseti dağılımı

	Saldırı Tipi	Saldırı Verisi	Normal Veri	Toplam Veri
Eğitim	15	94	3.653	3.744
Test	15	43	1.565	1.604

Modeller için RapidMiner, Weka, Matlab ve Excel programlarından faydalanılmıştır. 18 farklı teknik kullanılmasına rağmen burada iyi performans göstermiş 4 teknik

üzerinde sonuçlar gösterilmektedir. Kullanılan teknikler arasında en kötü sonucu Polinomiel Regresyon ile sınıflandırma (doğruluk 0,11%, duyarlılık %0) gerçekleştirmiştir. Bu aşamada YSA, karar ağaçları, k-en yakın komşu (k-NN) ve genetik algoritma ile optimize edilmiş karar destek makinesı (SVM GA) modellerinin yüksek doğrulukta sonuçlar verdiği görülmüştür. YSA'da geri yayılım algoritması kullanılmıştır ve eğitim çevrimi 500 iterasyondur. Öğrenme katsayısı 0,3, moment değeri ise 0,2 alınmıştır. Hedeflenen hata değeri 0,00001 olarak belirlenmiştir ve tüm veriseti [-1 1] aralığında normalize edilmiştir. YSA'da tek katmanlı olarak 80 neron ve sigmoid türünde aktivasyon fonksiyonu kullanılmıştır. Karar ağaçlarında ise ağacı dallandırmanın kararı kazanç oranına göre belirlenmesi seçilmiştir. Ağacın maksimum derinliği sınırsız belirlendiği takdirde ezberleme yapabileceğinden dolayı, ezberleyemeyecek kadar düşük fakat saldırıları tespit edebilecek kadar büyük olmasını sağlayacak biçimde tasarlanmış ve 20 olarak belirlenmiştir. Ağaçta genişlemeleri engelleme için budama işlemi de önem kazanmaktadır. Budama sınırı 0,25 olarak belirlenmiştir ve iyimser model kurgulanması sağlanmıştır. En düşük kazanç değeri 0,1 olarak alınmıştır. Bu değer düğüm bölünmeden önce hesaplanmaktadır ve kazanç bu değerden düşükse düğüm bölünerek dallanır. Parçalama sonrasında her yapraktaki en az veri boyutu 4 olarak alınmıştır. Bunun sebebi saldırıların 1-32 veri arasında değişmesidir. Bu değerin yüksek olması saldırıların belirlenememesine, çok düşük olması (örneğin 1) ağacın fazladan dallanmasına sebep olur. Kullanılan bir diğer teknik ise SVM GA'dır. SVM için radyal çekirdek seçilmiştir. Radyal çekirdek $e^{(-g\ ||x-y||\hat{}2)}$ şeklinde tanımlanmaktadır. Burada g gamma değerini göstermektedir. Düzeltilmiş gamma parametresi çekirdek performansında en fazla öneme sahip parametredir ve probleme özgü olarak dikkatle belirlenmesi gerekmektedir. Burada GA tarafında popülasyon başlangıcı rastsal atanmıştır ve en fazla 10.000 adet jenerasyon üretilmektedir. 30'dan fazla jenerasyonda gelişme olmazsa ilerleme kaydedilemediği için durması belirlenmiştir. GA tarafında seçim turnuva yöntemi ile yapılmaktadır ve kesir değeri 0,75'dir. GA mutasyon çeşidi Gausian'dır. Çaprazlama olasılığı ise 1 olarak alınmıştır. k-NN tekniğinde ise gruplandırma öznitelikler kullanılarak Öklid mesafesine göre hesaplanmaktadır. İki öznitelikte düzlem, üç öznitelikte hacim yapısında olurken, 4 ve üstü öznitelikte geometrik gösterimleri yoktur. Öznitelik sayısı

arttıkça daha doğru sonuç verirler. Yaklaşım olarak diğer üç yönteme göre daha basittir ve en temel yöntemler arasında yer alır.

6.6. Sonuçlar

Saldırının olup olmadığının tahmini binomial bir davranıştır. Bu yüzden ikili sınıflandırma olarak tahminler gösterilebilmektedir. İkili tahminde sınıflandırma doğruluk (accuracy), kesinlik (precision) ve duyarlılık (recall) şeklinde gruplandırılır. Bu yaklaşımlar ikili sınıflandırıcı için örnek karışıklık matrisi kullanılarak belirlenir. Bu matriste dört durum bulunmaktadır. Bunlar:

a. Gerçek Pozitifler (TP): Gerçekte saldırı ve saldırı olarak tahmin
b. Gerçek Negatifler (TN): Gerçekte saldırı değil ve normal ağ trafiği olarak tahmin
c. Yanlış Pozitifler (FP): Saldırı olarak tahmin fakat aslında normal ağ trafiği ("Tip I hatası" olarak da bilinir)
d. Yanlış Negatifler (FN): Normal ağ trafiği olarak tahmin fakat aslında saldırı ("Tip II hata" olarak da bilinir.)

Buna göre:

Doğruluk = Her iki sınıfta da doğru tahmin edilenler / Tüm veri (TP+TN)/Tüm veri
Kesinlik = (TP)/(TP+FP)
Duyarlılık = TP/(TP+FN) olarak hesaplanmaktadır.

Buna göre dört modelin sonuçları Tablo 6.8.'de gösterilmiştir. Bu sonuçlara göre karar ağaçlarının dört teknik arasında en düşük performansı gösterdiği söylenebilir. Sonrasında YSA ve SVM GA sırasıyla gelmektedir. En iyi tahmin sonuçlarını k-NN modeli vermiştir ve saldırıların hepsini doğru şekilde belirleyebilmiştir. Bunun sebebi öznitelik sayısı kadar boyuta ulaşabilmesidir.

Tablo 6.8. Model tahmin sonuçları

Model	Doğruluk	Duyarlılık	Kesinlik
Yapay Sinir Ağları (NN)	98,28%	100,00%	98,27%
Karar Ağaçları (DT)	98,17%	100,00%	98,16%
SVM GA	99,81%	100,00%	99,81%
k En Yakın Komşu	100,00%	100,00%	100,00%

Tahmin sonuçları saldırıların belirlenebileceğini dört model içinde yüksek doğrulukta göstermiştir. Bu tahminlerde hangi saldırının olabileceğinin tahmini de önemli olmaktadır. Çalışmanın bir diğer çıktısı olarak saldırı tür tahminine ilişkin sonuçlar Tablo 6.9.'da ve Şekil 6.5.'de gösterilmiştir. Bu aşamada önemli olan tüm saldırılar kesikli değildir. Diğer ifadeyle bir saldırı belirli bir zaman diliminde gerçekleşmiş, tekrar başka bir yerde gerçekleşmemiştir.

Dört modelin hepsi normal ağ trafiğini doğru şekilde tahmin etmiştir. A1 saldırı türünü sadece k-NN modeli doğru olarak tespit edebilmiştir. Burada önemli olan nokta A1 saldırısı sadece 1 satırlık veride bulunmaktadır. Bu durum da modellerin bu saldırıyı uygun tipte bulmasını güçleştirmektedir. A2 saldırısında da tek satırlık veri söz konusu olmasına rağmen, k-NN dışında SVM GA modelinin de doğru gruplandırma tahmini yapabildiği görülmektedir. Bu saldırı tipinde SumData5 özniteliğinin önemli olduğu belirlenmiştir. SVM GA, A3 saldırısının yarısını A3 olarak bulurken, kalan saldırıları normal ağ trafiği olarak belirlemiştir. Yine A3 saldırısında k-NN tüm saldırıları belirleyebilmiştir. A4, A5, A6, A7, A8, A9, A10 ve A12 saldırılarını yapay sinir ağları ve karar ağaçları doğru biçimde belirleyememiştir. A4 saldırısında ise SVM GA modeli yarısını doğru olarak tahmin ederken, bu saldırılardan kalanların %80'ini A3, %20'sini A1 saldırısı olarak belirlemiştir. A5 saldırısında ise SVM GA yarısını doğru olarak belirlerken, aynı saldırıyı A1, A4 ve A6 saldırısı olarak da tahmin ettiği durumlar olmuştur. A6 saldırısında SVM GA saldırı olduğunu – A1 ve A2 saldırıları – belirlemiş fakat doğru olarak sınıflandıramamıştır. A7 'de de benzer durum söz konusudur ve %22,22'lik veri doğru biçimde tahmin edilirken, kalan %77,78'lik veri normal ağ trafiği, A2, A5 ve A6 saldırıları olarak tahmin edilmiştir. A8 'de ise %57,14'lik doğru tahmin oranı dışındaki tahminlerde A6 ve A7 saldırısı olarak belirlenmiştir. A9 saldırılarında %34,38'lik doğru tahmin, %34,38'lik A8, %17'lik A7

ve %8,5'luk A6 ve kalan tahminlerde A4 olmuştur. Aslında saldırı olduğunu %100 doğru tespit edilmiş fakat türü tam olarak belirlenememiştir. SVM GA modelinde A10 saldırılarında %14,29'luk saldırının A8 saldırısı, geri kalanın doğru belirlenmesi gerçekleşmiştir. A11 saldırısında ise en kötü performansı SVM GA gerçekleştirmiştir. YSA, karar ağaçları ve k-NN modeli saldırıların tamamını bulurken, SVM GA bu tip saldırıların %14,29'luk kısmını A10 saldırısı olarak tahmin etmiştir. A12 saldırısında SVM GA saldırı tipini A8 olarak bulmuştur ve saldırı tespitini gerçekleştirmesine rağmen türünü doğru belirleyememiştir. A13 türündeki saldırıda YSA ve k-NN tüm saldırıları doğru biçimde belirlemiştir. Karar ağacı ise %7,14'lik verisetini normal ağ trafiği olarak belirlemişken, SVM GA ise %57,14'lük bölümünü A15, %7,14'lük bölümünü A7 saldırı türü olarak tahmin etmiştir. A14 saldırısında ise karar ağacı saldırıyı normal ağ trafiği olarak belirlemişken, YSA saldırının %14,28'ini A12, %28,56'sını normal ağ trafiği olarak belirlemiştir. SVM GA modeli ise A14 saldırısının %14,28'ini A11, %28,56'sını A12 olarak tahmin etmiştir. A15 saldırısında ise SVM GA %8,33'lük A11 ve %8,33'lük A13 saldırısı belirlemiştir.

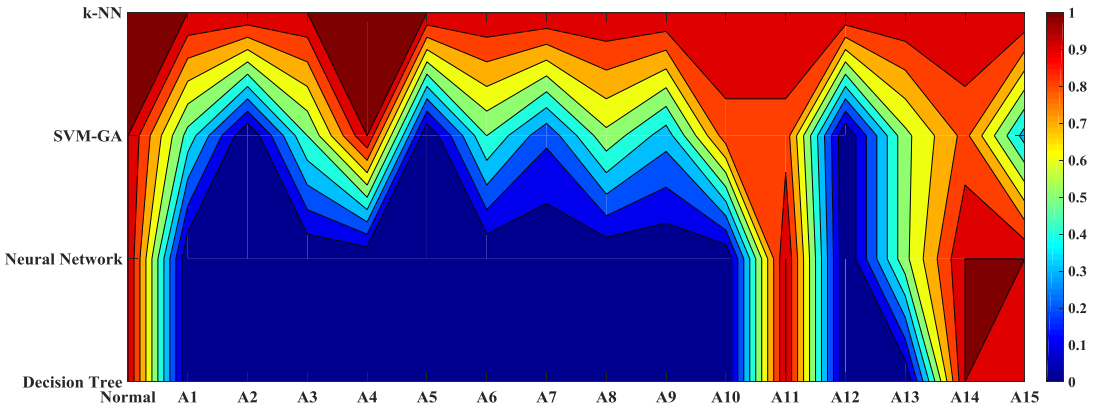
Tablo 6.9. Model saldırı türü tahmin sonuçları

	A0	**A1**	**A2**	**A3**	**A4**	**A5**
Karar Ağaçları (DT)	100,00%	0,00%	0,00%	0,00%	0,00%	0,00%
k En Yakın Komşu	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
Yapay Sinir Ağları (NN)	100,00%	0,00%	0,00%	0,00%	0,00%	0,00%
SVM GA	100,00%	0,00%	100,00%	46,15%	50,00%	50,00%

	A6	**A7**	**A8**	**A9**	**A10**	**A11**
Karar Ağaçları (DT)	0,00%	0,00%	0,00%	0,00%	0,00%	100,00%
k En Yakın Komşu	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
Yapay Sinir Ağları (NN)	0,00%	0,00%	0,00%	0,00%	0,00%	100,00%
SVM GA	0,00%	22,22%	57,14%	34,38%	85,71%	85,71%

	A12	**A13**	**A14**	**A15**
Karar Ağaçları (DT)	0,00%	92,86%	0,00%	100,00%
k En Yakın Komşu	100,00%	100,00%	100,00%	100,00%
Yapay Sinir Ağları (NN)	0,00%	100,00%	57,14%	100,00%
SVM GA	0,00%	35,71%	57,14%	83,33%

Genel olarak saldırı tespit modelleri sırasıyla normal ağ trafiği, A11, A15 ve A13 durumları en doğru tahmin edilen türler olmuştur. Sonrasında performans olarak sırasıyla A2, A10, A8, A4, A5, A3, A9 ve A7 gelmektedir. En kötü performansı A1, A6 ve A12 saldırılarında göstermişlerdir. A1, A6 ve A12 türlerinde tahmin edilecek veri sayısı çok düşük olduğundan performansları da kötü olmuştur.



Şekil 6.5. Saldırı tipine göre tahmin sonuçları yüzeyi

Tekniklere göre saldırı tipinin doğru belirlenmesinin oranlarını gösteren ısıl alan grafiği Şekil 6.5.'de görülmektedir. Bu şekilde kırmızı renk gruplandırma tahminlerinin doğruluğunu, koyu mavi renk yanlışlığını göstermektedir. Buna göre grafik üzerinde k-NN ve SVM GA modellerinin YSA ve karar ağaçları ile karşılaştırıldığında daha iyi performans sergilediği rahatlıkla görülebilmektedir. YSA ve karar ağaçlarının özellikle A1 – A10 aralığı ve A12 saldırı biçimi için saldırı grup tespitinde yanlış belirlemeleri söz konusudur.

BÖLÜM 7. TARTIŞMA VE SONUÇ

Otomasyon endüstrisinde bulunan kritik üretim aşamaları, gerçek zamanlılık, ürün sürekliliğini sağlayan operasyonel teknolojilerin son yıllarda bilgi teknolojileri ile entegre edilmesi, sistemlerde diğer bir gereklilik olan güvenlik unsurunu öne çıkarmıştır. Otomasyon hiyerarşisinde bulunan seviyeler arası/içi iletişimde, verilerin toplanması, saklanması, iletimi, donanımların kontrolü gibi her aşamada uygulanması gereken bu unsur günümüzde çeşitli ticari veya açık kaynak ürünler üzerinden sağlanmaktadır. Son yıllarda SCADA sistemler üzerine yapılan stuxnet, dragonfly, havex, blackenergy, triton gibi kritik siber saldırılar göz önüne alındığında bu faktörünün önemi daha da artmıştır. Bu anlamda, kritik altyapıların güvenliliğinin standartlaştırılması amacıyla, hiyerarşik model üzerinde kar amacı gütmeyen kurumlar veya ticari firmalar tarafından sunulan çeşitli referans modeller de sıklıkla uygulanmaktadır. Fakat Ethernet üzerinden iletişim sağlayan EKS protokollerinin çoğu açık metin, kimlik doğrulama ve yetkilendirme olmadan iletişim sağladıklarından, BT ile uyumlaştırma süresi neticesinde veya EKS'nin sağlamlaştırılmasına yeteri kadar önem verilmediği için iletişim altyapısı hala sıfırıncı gün ve hatta bilinen zafiyetleri barındırmakta ve olası tehditlere karşı açık durumdadır.

Güvenlik zafiyetleri, donanımdan ve ağ topolojisinden bağımsız olarak sistem üzerine yapılabilecek olası saldırıların temelini oluşturması açısından oldukça önemlidir. Sistemler üzerine yapılan saldırıların yol açtığı sonuçların kritik olması nedeniyle, ilgili protokol zafiyetlerinin önceden tespiti gerekmektedir. Belirli testler uygulanarak protokol tabanlı tahlillerin yapılmasının ardından, bulunan zafiyetler kullanılarak olası saldırıların tahmini ve analizi, çalışılması gereken bir diğer konudur. Bu sayede, zafiyetin istismarı sonucu oluşabilecek saldırının etkisi önceden tahmin edilebilmekte ve tedbir alınabilmektedir.

Tez kapsamında, EKS güvenliğine katkı sağlamak amacıyla, mevcut genel çözümlere kıyasla özel bir çözüm olan EtherCAT tabanlı kritik yapılarda kullanılabilecek, EtherCAT tabanlı sistemler adına kapsamlı bir çözüm önerisi sunulmuştur.

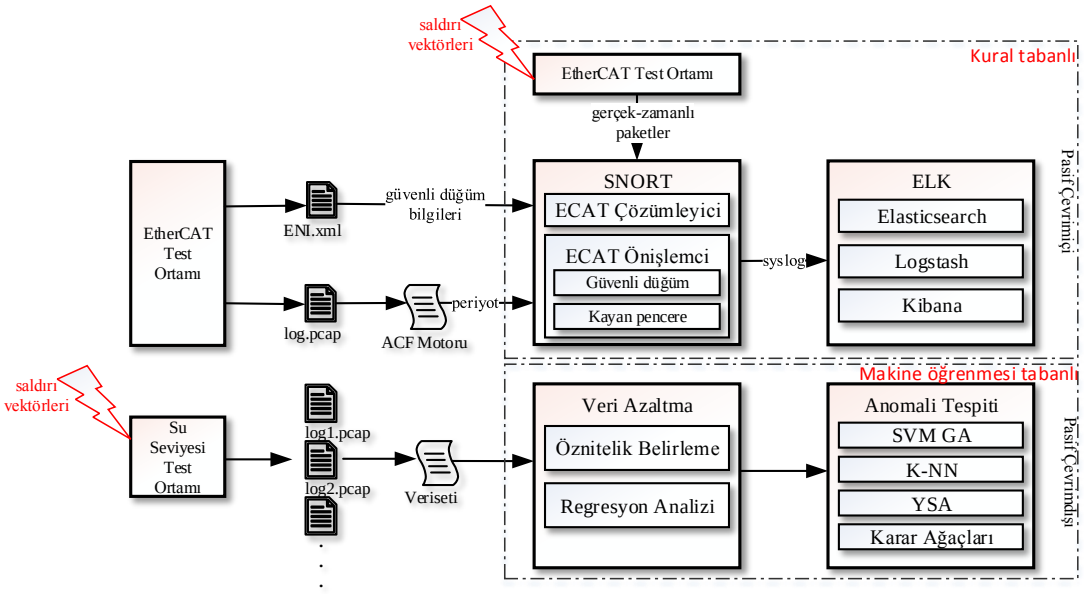
7.1. Sonuçlar

Bu tezde protokol zafiyetlerinin tespit edilmesi, zafiyetlerden elde edilen bilgilerle saldırı vektörlerinin oluşturulması, hem bilinen hem de bilinmeyen olası saldırıları tespit eden yaklaşımlar geliştirilmesi, öneriye eklenen özgün çözümlerle siber tehditlere karşı daha gürbüz bir sonuç veren yapının oluşturulması amaçlanmıştır. Bu kapsamında, geçmiş çalışmalar incelendikten sonra,

a. EtherCAT protokol yapısının ve çalışma prensiplerinin analizi,
b. Her aşama için test ortamlarının oluşturulması, PLC programlarının geliştirilmesi ve ilgili konfigürasyonların yapılması,
c. Bulanıklaştırıcı geliştirme ve saldırı vektörü oluşturma yoluyla zaafiyet testlerinin yapılması. Ayrıca veriseti oluşturulması,
d. Önişlemci geliştirilmesi ve test edilmesi,
e. Bilinen saldırılar: Güvenli düğüm yaklaşımı geliştirilmesi, veri enjeksiyonu, köleler üzerine saldırı, MAC aldatma, DoS gibi bilinen saldırıların kurallarının oluşturulması ve test edilmesi,
f. Bilinmeyen (sıfırıncı-gün) saldırılar: Periyot analizine dayanan anomali tespit modülü geliştirilmesi ve test edilmesi,
g. 15 farklı saldırı vektörü üzerinden oluşturulan veriseti ile kılavuzlu makine öğrenmesi yöntemleriyle anomali tespiti yapılması,
h. Saldırıların çevrimiçi izlenmesi ve takibi için ELK izleme sisteminin oluşturulması,

çalışmaları başarı ile tamamlanmıştır. Böylece, EtherCAT tabanlı sistemler üzerine yapılacak saldırıları tespit eden, Şekil 7.1'de sunulan bütüncül bir yaklaşım önerilmiştir. Çözümler pasif izleme yöntemleriyle yürütülmüş olup, çalışan sistem

üzerinde herhangi bir yük veya gecikme oluşmamaktadır. Kural tabanlı olan yöntemler çevrimiçi, anomali tabanlı çözüm ise çevrimdışı olarak çalışmaktadır.



Şekil 7.1. Tez çalışmasının özeti

Zafiyet analizinde 2 farklı yöntem kullanılmış olup bunlar bulanıklaştırma ve saldırı vektörü oluşturulmasıdır. Bulanıklaştırma yaklaşımında 2 farklı test uygulanmıştır. İlk test kapsamında üretilen 335 milyon paketten köle tarafından kabul edilen paketlerin WKC değerlerinin 1, 3, 11, 33 olarak arttığı gözlemlenmiştir. Çerçeve içindeki IRQ alanı ise köleler tarafından 4 olarak güncellenmiştir. Bu değer bağlantının hala aktif olduğunu göstermektedir. Köleler kendilerine ait paketi kabul edip bağlantı durumunu aktif olarak belirtmiştir. Ayrıca kabul edilen paketlerin çoğunlukla yazma komutu içerdiği anlaşılmaktadır. Bu da kölelerin genellikle yazma komutu içeren paketleri icra ettikleri anlaşılmaktadır. İlk test kapsamında ise 209.945 adet paketten 11.596 adet paketin kabul edildiği görülmüştür. Bu paketlerin çoğunluğu normal trafik, 4 adet paket ise kesme paketidir. Köleler genellikle otomatik arttırılmış adresleme ile gönderilen okuma, yazma ve çoklu yazma komutlarını icra etmişlerdir. Bu komutlar içinde ise köleler tarafından WKC değeri en çok 11 olarak artırılmıştır. Kabul gören paket sayısını adresleme türünün etkilediği gözlemlenmiştir.

Zafiyet analizindeki bir diğer yaklaşım saldırı vektörüdür. EtherCAT üzerinde efendi ve köleler arasında sadece sistemin başlangıç durumunda bir yol atanmaktadır. Bu yol sistemin enerjisi kesilmedikçe korunmakta ve iletişim esnasında herhangi bir doğrulama yapılmamaktadır. Bu da EtherCAT trafiğinin kolaylıkla dinlenebileceğini hatta trafiğe müdahale edilebileceğini göstermektedir. Buradan hareketle MAC aldatma saldırısı, DoS saldırısı, kaba kuvvet saldırısı ile köle adreslerine yönelik ataklar ve veri enjeksiyonu gibi saldırılar geliştirilmiştir. Ayrıca, efendilerin köleleri veya kölelerin efendilerini tanıyacakları bir mekanizma da bulunmadığından MITM saldırıları da uygulanmıştır. Sistemden MITM ile elde edilen veriler analiz edilerek saldırı oluşturulduğunda, diğer kölelerde yer alan bilgilere ulaşıldığı, sistemin durdurulduğu, kölelerin farklı efendilerden gelen komutların işlenmesine izin verdikleri ve bu türden saldırıların başarı gösterdiği görülmüştür. Buna ek olarak sistemin topolojisini tespit etmeye veya kölelerin diğer bellek adreslerinde tutulan kaydedici değerlerine erişmeye yönelik türden saldırılara da açık oldukları görülmüştür.

Zafiyet analizi sonunda Ethernet altyapısı üzerinden gelen saldırılara açık oldukları gözlemlenen EtherCAT tabanlı sistemler için öncelikle Snort önişlemcisi geliştirilmiştir. Önişlemci, geliştirilen bir çözümleyici tarafından yönlendirilen trafik akışı üzerinden saldırı tespiti yapmaktadır. Snort sistemi test ortamına pasif modda bağlanmış olup gelen trafik saldırı tespiti için gerçek zamanlı olarak basılan zaman damgaları ile trafik üzerinde herhangi bir yük oluşturmadan sadece kopyalanarak dağıtılmaktadır. Alarm ve günlükleme olarak kaydedilen şüpheli durumlar ise ELK yığını kurulu olan bir sisteme sistem günlüklemelerinden otomatik olarak aktarılmaktadır. Snort üzerinde gerçek zamanlılığı destekleyen Modbus, DNP3 gibi EKS protokollerine ait önişlemciler daha önce bulunduğundan, EtherCAT protokolü üzerinde de saldırı tespiti esnasında herhangi bir probleme rastlanmamıştır. Çalışma, EtherCAT çözümleyici sayesinde diğer EtherCAT tabanlı sistemlere uygulanabilmekte olup, Snort sistemini protokol tabanlı yeni kural veya yaklaşımlar eklemeye uygun hale getirmiştir. Bu kapsamda, güvenli efendi, güvenli köle ve veri enjeksiyonu olmak üzere 3 adet kural tanımlanmıştır. Eklenen kurallardan ikisi önerilen güvenli düğüm yaklaşımı ile tanımlanmış olup, EtherCAT üzerinde yer alan konfigürasyon ve kimlik

bilgilerinin yer aldığı ENI dosyasından aldığı bilgileri, akış içindeki değerler ile karşılaştıran bir kuraldır. Çalışmanın, MAC aldatma, veri enjeksiyonu ve köle adresleri üzerine yapılan saldırıları, tanımlanan kurallar yardımıyla tespit edebildiği gözlemlenmiştir. Ayrıca, 3, 4 ve uygulama katman protokollerini destekleyen Snort yapısının 2. katmanda çalışan EtherCAT protokol paketlerini başarıyla ayrıştırdığı da görülmüştür.

EKS'ler atanmış sistemlerdir. Konfigüsrasyonlar mühendislik istasyonları tarafından başlangıçta tanımlanır. Buna göre her donanım kendine özgü bir kimliğe sahiptir. Sistemi kontrol eden ve hesaplamaları yapan efendi cihazlar sistemin çalışması esnasında bu kimlik verilerine göre prosesleri yürütmektedir. Kimlik verileri GSD, ESI, ENI veya protokole bağlı olarak diğer dosya formatlarında kaydedilmektedir. Geliştirilen güvenli düğüm yaklaşımı, kimlik verilerine dayalı olduğundan dosya formatları ve sistemlerin çalışma prensipleri bilinirse diğer protokollere de uygulanabilmektedir.

EKS yapılarında çalışma esnasında paketlerin işlenmesi veya sistem üzerine aşırı yük bindirilmesi sistemdeki gerçek zamanlılığı bozabileceğinden çalışma pasif izleme yöntemi kullanılarak sunulmuştur. Hali hazırda çalışan EKS üzerine sızma testi yapılması önerilmemektedir. Bu da zafiyetlerin bulunmasını güçleştirmektedir. Pasif izleme yöntemiyle kurallar tanımlanarak EKS üzerindeki olası tehdit ve risklerin aza indirgenmesine katkı sağlanmıştır.

Bilinen saldırıların tespitine ek olarak, sıfırıncı gün yani bilinmeyen saldırıların da tespiti önem arz etmektedir. Bu kapsamda geliştirilen periyot tespitine dayalı anomali tespit çalışması, önişlemci üzerinden 0,95 ve 0,99 doğrulukla hat üzerindeki periyotu tespit edebilmektedir. Bu işlemlerde kullanılan otokorelasyon fonksiyonunun EKS çevrimli iletişimin periyodiklik tespitinde kullanılabileceğini göstermektedir. Test amacıyla geliştirilen 4 adet programın periyot değerlerinin sırasıyla 10, 4, 5,5 ve 108 saniye olarak bulunduğu, saldırıların ise konfigürasyon dosyasında tanımlanan periyot kullanılarak, başarıyla tespit edilebildiği sistem günlüklemelerinden doğrulanmıştır.

Bu da saha veriyolu trafiğindeki periyot değerinin önemini ve periyot bilindiğinde anomali tespitinin mümkün olduğunu kanıtlamaktadır.

Periyot başına düşen değerlerin protokol alanlarına bağlı olarak toplandığı ve belirli bir pencere içerisinde ardışık olarak tutularak karşılaştırıldığı bu yapıda, trafik akışındaki az sayıdaki paket sapmaları bile istatistikleri etkilemekte, dolayısıyla belirli bir standart sapmanın dışına çıkabilmektedir. Sıfırıncı gün saldırıları genellikle ağ topolojisini tarama işleminden sonra köle belleklerine yazma okuma yapma veya çevrimsel iletilen veriler üzerinde değişiklik yapma üzerine yoğunlaşmaktadır. Dolayısıyla bu çözüm sıfırıncı gün saldırılarının tespitini de mümkün kılmaktadır.

Çalışmanın önişlemci geliştirilmesi ve bilinmeyen saldırıların tespitine çözümleri test ortamında gerçeklenmiş, saldırılar düzenlenmiş ve sonuçları alarm ve günlükleme olarak kaydedilmiştir. Kaydedilen sistem syslog dosyaları ise ELK yığınının kurulu olduğu bir sanal makineye yönlendirilmiştir. Kullanıcı tarafında tanımlanan bir görüntüleme ekranı üzerinden ise istatistiki bilgilerin başarılı bir şekilde kullanıcıya sunulduğu görülmüştür.

Çalışmanın son aşamasında ise 15 farklı saldırı vektörü ve 1 normal trafik akışını içeren bir veriseti, geliştirilen su seviye kontrol otomasyonu test ortamından alınan verilerle oluşturulmuştur. Verisetinde korelasyon kullanılarak saldırı tespitinde etkisiz olan veriler çıkarılmıştır. Ayrıca saldırı tabanlı ve tüm verisetinde anlamlı olan değişkenler tespit edilmiştir. 18 teknik kullanılarak anomali tespiti yapılan veri setinde k-NN, SVM GA, YSA ve karar ağaçları yöntemleri en yüksek doğrulukta saldırı tespiti yapmışlardır. Buna göre k-NN ve SVM GA modelleri diğer 2 yönteme göre saldırıların birçoğunda daha iyi sonuç verdiği görülmüştür. A1 – A10 ve A12 saldırılarında, YSA ve karar ağaçlarının özellikle grup tespitinde yanlış tespitleri bulunmaktadır. k-NN yöntemi SVM GA'ya kıyasla daha başarılı olup tüm olayların gruplarını doğru tahmin etmiştir. SVM GA ise birinci tanka giden su akışını durdurma, duran pompanın çalıştırılması ve bağlantının yeniden kurulması gibi olayların tespitinde başarısız olduğu görülmüştür. Bu da k-NN algoritmasının saha veri yolu iletişiminde, ağdaki donanım, konfigürasyon değişimlerinden veya kesintilerden kaynaklı olaylarda,

yığılmalardan ötürü aniden yükselen, yazılım tabanlı bir sisteme veya web siteye erişimlerin fazlalaşması gibi zamanla düzene giren anomalilerde, ölçüm hatalarından kaynaklı anomalilerde ve ağın kötüye kullanıldığı saldırılarda, yüksek doğrulukla birçok saldırının tespitinde kullanılabileceğini göstermektedir. SVM GA yönteminin ise kesintilerden kaynaklı olaylarda ve ağın kötüye kullanıldığı saldırıların bazılarında başarısız tespitleri mevcuttur. Bu tespitler su seviye kontrolü test ortamı ve saldırı çeşitlerinde geçerli olup farklı EKS uygulamalarında değişkenlik gösterebilmektedir.

7.2. Çalışmanın Bilime Katkısı

a. Bilime/endüstriye yenilik getirme;

- EKS protokollerinin birçoğunda konfigürasyon ve kimlik bilgileri bir dosyada saklanmaktadır. Ayrıca, bu protokoller çevrim-zamanı yapısını da desteklemektedir. Bu anlamda, otokorelasyon kullanılarak periyot tespiti ve güvenli düğümlerin tespit edilmesine dayalı sunulan çözümler diğer Ethernet tabanlı gerçek zamanlı protokollere de uygulanabilmektedir.

- Şimdiye kadar kritik altyapılı sistemlere yönelik istismarların tespiti amacıyla yapılan genel amaçlı veya DNP3, Modbus/TCP, Profinet gibi bazı protokoller için zafiyet analizleri yapılmasına rağmen, sektörde popüler olan ve yaygın olarak kullanılan gerçek zamanlı EtherCAT protokolü üzerine yapılmış bir zafiyet araştırması bulunmamaktadır. Bu anlamda çalışma EtherCAT tabanlı sistemler için yol gösterici olmaktadır.

- EtherCAT protokolü üzerinden gelen siber tehditleri tespit için sadece uluslararası pazarda ve genellikle güvenlik duvarı seviyesinde ürünler bulunmaktadır. Fakat bu ürünler, IDS/IPS sistemleri gibi derin paket analizi özelliğini desteklememektedirler. Ayrıca bu ürünler, EtherCAT protokolü açıklık tespiti üzerine yapılacak bir çalışmayla ortaya

çıkabilecek zafiyetlere veya sıfırıncı gün saldırılarına da çözüm niteliği taşımamaktadır. Bu çalışmayla, EtherCAT tabanlı sistemlerde kural tabanlı ve anomali tabanlı olarak Şekil 7.1.' deki gibi bütüncül bir yaklaşımla saldırı tespiti yapılarak, EKS üzerindeki siber risklerin azaltılmasına katkı sağlanmaktadır.

b. Bilinen bir yöntemi yeni bir alana uygulama;

- Mevcut güvenlik duvarları (Beckhoff EDR-G903 Firewall/NAT/VPN, Moxa EDR-810) veya yazılımsal diğer IDS/IPS ürünleri yerine alternatif bir çözümün ortaya çıkartılmasına katkı sağlanmaktadır.

- Sıfırıncı gün saldırılarına çözüm olan ACF tabanlı yaklaşım periyot tespitinde kullanılmıştır. Bu yöntem literatürde farklı alanlardaki periyodikliğin tespitinde kullanılmış olup EKS güvenliğinde daha önce uygulanmamıştır.

c. Diğer katkılar;

- Literatürde ve endüstride gerçek zamanlı sistemlere entegre edilebilecek EtherCAT protokolü için yazılmış bir önişlemci uygulaması bulunmamaktadır. Bu çalışmada Snort yapısına eklenti olarak entegre edilebilecek EtherCAT protokolü için bir önişlemci yazılımı geliştirilmiştir.

- İmza tabanlı çözümde güvenli düğümlerin tespiti yaklaşımı çözümü: EtherCAT protokolünde kimlik doğrulama olmamasından kaynaklı, sistemdeki mevcut düğümleri tespit etmek veya farklı kölelerin/düğümlerin yer alıp almadığını kontrol etmek için yapılan, farklı veya benzer adreslere paket yollama, ağ haritası veya ağdaki düğümlerin adres ve konfigürasyon bilgilerinin çıkarılması saldırılarının önüne geçebilmektedir.

- Sıfırıncı-gün çözümü: Çalışma kapsamında kullanılan zaman serileri yöntemleri, benzerlik oluşturan örüntüler üzerinde çalıştığından birebir saldırı olmasa dahi benzer türden olan sıfırıncı gün saldırılarına da çözüm niteliği taşımaktadır.

- Hem EtherCAT tabanlı açıklıkların muhtemel siber saldırılara açık halde sistemde barındırılmasının önüne geçilecek hem de bilinmeyen siber saldırılara karşı EtherCAT tabanlı sistemler daha iyi korunmuş olacaktır. Bu anlamda EtherCAT protokolünü kullanan EKS üzerindeki siber risklerin azaltılmasına katkı sağlanmıştır.

7.3. İleriki Çalışmalar

Bu çalışma kapsamında EtherCAT saha veri yolu iletişimi ele alınmış olup geliştirilen yaklaşımlar saha veri yolu kabulleri göz önünde tutularak önerilmiştir. Sonraki çalışmalar için, fabrika iletişimi gibi asenkron iletimin gözlendiği diğer seviyelerde görülebilecek saldırı tespiti çalışmaları yapılabilir.

Açık kaynak kodlu olan Snort altyapısı üzerinde yapılan uygulamalar, ileriki çalışmalarda kullanılmaya uygun olarak geliştirilmiştir. Önerilen önişlemciye farklı kurallar eklenerek EtherCAT üzerinde çıkabilecek diğer zafiyetler önlenebilir veya farklı çözüm önerileri eklenebilir.

EtherCAT protokolünde, şifreleme, yetkilendirme ve kimlik doğrulama gibi temel güvenlik parametrelerinin bulunmamasından dolayı zafiyetlerin olabileceği varsayımından yola çıkarak zafiyet analizi yapılmıştır. Tespit edilen protokol zafiyetleri de göz önüne alınarak, farklı kriptografik yaklaşımlar veya güvenlik model önerileri gibi protokol üzerinde iyileştirmeler yapılabilir.

Gerçek zamanlı protokollerin birçoğunda şifreleme, yetkilendirme ve kimlik doğrulama özellikleri bulunmamakta, onlarda da çevrimli veriler iletilmekte veya konfigürasyon dosyası yer almaktadır. Bu faktörlerden dolayı bu çalışmada uygulanan yaklaşımlar diğer protokollere de uygulanarak diğer protokollerin güvenliği üzerine de çalışılabilir.

KAYNAKLAR

[1] Ebata, Y., Hayashi, H., Hasegawa, Y., Komatsu, S., Suzuki, K., Development of the Intranet-based SCADA (supervisory control and data acquisition system) for power system. 2000 IEEE Power Engineering Society Winter Meeting. Conference Proceedings (Cat. No.00CH37077), 3, 1656–1661.

[2] Obregon, L., Secure Architecture for Industrial Control Systems. , 2015.

[3] Williams, T. J., A Reference Model for Computer Integrated Manufacturing from the Viewpoint of Industrial Automation. IFAC Proceedings Volumes, 23 (8), 281–291, 1990.

[4] Stouffer, K., Pillitteri, V., Lightman, S., Abrams, M., Hahn, A., Guide to Industrial Control Systems (ICS) Security. Gaithersburg, MD, , 2015.

[5] Internet: Brocklehurst, K., IT-OT Convergence and Conflict: Who Owns ICS Security? .

[6] Johnson, R. E., Survey of SCADA Security Challenges and Potential Attack Vectors. International Conference for Internet Technology and Secured Transactions, , 5, 2010.

[7] FireEye iSight Intelligence, Overload Critical Lessons From 15 years of ICS Vulnerabilities. , 2016.

[8] Sayegh, N., Chehab, A., Elhajj, I. H., Kayssi, A., Internal security attacks on SCADA systems. 2013 3rd International Conference on Communications and Information Technology, ICCIT 2013, , 22–27, 2013.

[9] García Márquez, F. P., Tobias, A. M., Pinar Pérez, J. M., Papaelias, M., Condition monitoring of wind turbines: Techniques and methods. Renewable Energy, 46, 169–178, 2012.

[10] Schlechtingen, M., Santos, I. F., Wind turbine condition monitoring based on SCADA data using normal behavior models. Part 2: Application examples. Applied Soft Computing, 14 (1), 447–460, 2014.

[11] McQueen, M. A., Boyer, W. F., Flynn, M. A., Beitel, G. A., Quantitative Cyber Risk Reduction Estimation Methodology for a Small SCADA Control System. Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06), , 226–226, 2006.

[12] Yakkali, H., Subramanian, N., Efficient design of SCADA systems using minimum spanning trees and the NFR Framework. 2010 42nd Southeastern Symposium on System Theory (SSST 2010), , 346–351, 2010.

[13] Chopade, P., Bikdash, M., Critical infrastructure interdependency modeling: Using graph models to assess the vulnerability of smart power grid and SCADA networks. 2011 8th International Conference & Expo on Emerging Technologies for a Smarter World, , 1–6, 2011.

[14] Aloul, F., Al-ali, A. R., Al-dalky, R., Al-mardini, M., Smart Grid Security : Threats , Vulnerabilities and Solutions. Smart Grid and Clean Energy Smart, (971), 1–6, 2012.

[15] Dong Wei, Yan Lu, Jafari, M., Skare, P., Rohde, K., An integrated security system of protecting Smart Grid against cyber attacks. 2010 Innovative Smart Grid Technologies (ISGT), , 1–7, 2010.

[16] Yang, Y., McLaughlin, K., Littler, T., Sezer, S., Pranggono, B., Wang, H. F., Intrusion Detection System for IEC 60870-5-104 based SCADA networks. 2013 IEEE Power & Energy Society General Meeting, , 1–5, 2013.

[17] Sayegh, N., Elhajj, I. H., Kayssi, A., Chehab, A., SCADA Intrusion Detection System based on temporal behavior of frequent patterns. MELECON 2014 - 2014 17th IEEE Mediterranean Electrotechnical Conference, (April), 432–438, 2014.

[18] Beaver, J. M., Borges-Hink, R. C., Buckner, M. a., An Evaluation of Machine Learning Methods to Detect Malicious SCADA Communications. 2013 12th International Conference on Machine Learning and Applications, 2, 54–59, 2013.

[19] Linda, O., Vollmer, T., Manic, M., Neural network based intrusion detection system for critical infrastructures. 2009 International Joint Conference on Neural Networks, , 1827–1834, 2009.

[20] Zhu, B., Joseph, A., Sastry, S., A Taxonomy of Cyber Attacks on SCADA Systems. 2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing, , 380–388, 2011.

[21] Liu, X., Ren, H., Zhao, Z., Zhang, P., EtherCAT technology for the network of smart substation. Conference Proceedings - 2012 IEEE 7th International Power Electronics and Motion Control Conference - ECCE Asia, IPEMC 2012, 3, 2300–2304, 2012.

[22] Li, M., Liu, H., Wang, L., Zhang, Q., Wang, J., EtherCAT Data Acquisition System Based on DMA Mode. 2nd International Conference on Industrial Technology and Management, , 5, 2012.

[23] Schieferdecker, I. K., Großmann, J., Schneider, M., Model-Based Security Testing. Electronic Proceedings in Theoretical Computer Science, 80 (Proc. MBT 2012), 1–12, 2012.

[24] Security, D., Dell security annual threat report 1. , 2015.

[25] Internet: Kevin, P., Slammer Worm Crashed Ohio Nuke Plant. http://www.securityfocus.com/news/6767 , 2016.

[26] Internet: Tony, S., Hacker Jailed for Revenge Sewage Attacks. http://www.theregister.co.uk/2001/10/31/hacker_jailed_for_revenge_sewage/ , 2016.

[27] Abrams, M., Weiss, J., Control system cyber security case study. Bellingham, Washington, , 2007.

[28] National Transportation Safety Board, Safety study: supervisory control and data acquisition (SCADA) in liquid pipelines. , 2005.

[29] Weiss, J., A review of selected actual control system cyber incidents. ICSJWG 2009 Fall Conference, , 2009.

[30] Miller, B., Rowe, D., A survey SCADA of and critical infrastructure incidents. Annual Conference on Research in Information Technology, , 51–56, 2012.

[31] Lee, R. M., Assante, M. J., Conway, T., Analysis of the cyber attack on the Ukrainian power grid. Washington, DC, USA, , 2016.

[32] Cheminod, M., Durante, L., Valenzano, A., Review of security issues in industrial networks. IEEE Transactions on Industrial Informatics, 9 (1), 277–293, 2013.

[33] Beaver, J. M., Borges-Hink, R. C., Buckner, M. A., An Evaluation of Machine Learning Methods to Detect Malicious SCADA Communications. 2013 12th International Conference on Machine Learning and Applications, , 54–59, 2013.

[34] Adamiak, M., Baigent, D., Mackiewicz, R., IEC 61850 Communication Networks and Systems in Substations: An Overview for Users. SISCO Systems, , 2004.

[35] Han, X., Wen, Q., Zhang, Z., A Mutation-Based Fuzz Testing Approach for Network Protocol Vulnerability Detection. Proceedings of 2012 2nd International Conference on Computer Science and Network Technology, , 1018–1022, 2012.

[36] Baud, M., Felser, M., Profinet IO-Device Emulator based on the Man-in-the-middle Attack. 2006 IEEE Conference on Emerging Technologies and Factory Automation, , 437–440, 2006.

[37] Dell Security, Annual Threat Report. , 2015.

[38] Internet: BeStorm, Dynamic Testing (Fuzzing) on the Ethercat Protocol by BeSTORM. http://www.beyondsecurity.com/dynamic_fuzzing_testing_ethercat_protocol , 2016.

[39] EtherCAT, EDR-G903 Firewall/NAT/VPN Secure Router. , 2015.

[40] Moxa, EDR-810 Series. , 2015.

[41] EtherCAT Technology Group, Industrial Ethernet Technologies: Overview. , 2014.

[42] Prytz, G., A performance analysis of EtherCAT and PROFINET IRT. 2008 IEEE International Conference on Emerging Technologies and Factory Automation, , 408–415, 2008.

[43] KingStar, White paper: 5 Real-Time, Ethernet-Based Fieldbuses Compared Which Standard Stands Apart? , 2016.

[44] EtherCAT, EtherCAT-the Ethernet fieldbus. , 2012.

[45] Beckhoff, Moving up to Industrial Ethernet: The EtherCAT protocol. , 2008.

[46] Timorin, A., SCADA Deep Inside: Protocols and Security Mechanisms. Balkan Computer Congress, , 2014.

[47] Shaw, W. T., Cyber Security for SCADA Systems. Tulsa, Okla. : PennWell Corp., C2006., 562 , 2006.

[48] Kleinmann, A., Wool, A., A Statechart-Based Anomaly Detection Model for Multi-Threaded SCADA Systems. , 132–144, 2016.

[49] Internet: Basaran, A., Siber Savas Cephesinden Notlar. http://securitist.blogspot.com.tr/2012/04/guvenlik-101.html , 2016.

[50] Klick, J., Lau, S., Marzin, D., Malchow, J., Roth, V., Internet-facing PLCs - a new back Orifice. Black Hat, , 2015.

[51] Wang, J., Hui, L. C. K., Yiu, S. M., Zhou, G., Zhang, R., F-DDIA: A Framework for Detecting Data Injection Attacks in Nonlinear Cyber-Physical Systems. Security and Communication Networks, 2017, 1–12, 2017.

[52] Sans Institute-Andrew Hildick-Smit, Security for Critical Infrastructure SCADA Systems. , 2005.

[53] NIST-National Vulnerability Database, CVE-2014-2252 Detail. , 2014.

[54] Beresford, D., Exploiting Siemens Simatic S7 PLCs. Black Hat USA, , 1–26, 2011.

[55] Siemens CERT, Security Vulnerabilities in Siemens SIMATIC S7-1200 CPU. , 2011.

[56] Siemens CERT, Web Vulnerability in SIMATIC S7-1200 CPU. , 2015.

[57] Krishnan Sadhasivan, D., Balasubramanian, K., A Fusion of Multiagent Functionalities for Effective Intrusion Detection System. Security and Communication Networks, 2017, 1–15, 2017.

[58] Ismail, I., Mohd Nor, S., Marsono, M. N., Stateless Malware Packet Detection by Incorporating Naive Bayes with Known Malware Signatures. Applied Computational Intelligence and Soft Computing, 2014, 1–8, 2014.

[59] Ntalampiras, S., Soupionis, Y., Giannopoulos, G., A fault diagnosis system for interdependent critical infrastructures based on HMMs. Reliability Engineering & System Safety, 138, 73–81, 2015.

[60] Goldenberg, N., Wool, A., Accurate modeling of Modbus/TCP for intrusion detection in SCADA systems. International Journal of Critical Infrastructure Protection, 6 (2), 63–75, 2013.

[61] Kleinmann, A., Wool, A., Accurate Modeling of the Siemens S7 SCADA Protocol for Intrusion Detection and Digital Forensics. Journal of Digital Forensics, Security and Law, 9 (2), 37–50, 2014.

[62] Cook, A., Nicholson, A., Janicke, H., Maglaras, L., Smith, R., Attribution of Cyber Attacks on Industrial Control Systems. EAI Endorsed Transactions on Industrial Networks and Intelligent Systems, 3 (7), 151158, 2016.

[63] Alotaibi, K. F., Hamidi, M. M., Talebi, M., Xu, J., Homaifar, A., Using spy node to identify cyber-attack in power systems as a novel approach. 2015 IEEE International Conference on Electro/Information Technology (EIT), , 581–586, 2015.

[64] Byres, E. J., Franz, M., Miller, D., The Use of Attack Trees in Assessing Vulnerabilities in SCADA Systems. International Infrastructure Survivability Workshop (IISW'04), , 2004.

[65] Ramachandruni, R. S., Poornachandran, P., Detecting the network attack vectors on SCADA systems. 2015 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2015, , 707–712, 2015.

[66] Samuelian, M., Seitz, B., A Universal Approach for implementing Real-Time Industrial Ethernet. , 2008.

[67] Vaughn, R. B., Morris, T., Addressing Critical Industrial Control System Cyber Security Concerns via High Fidelity Simulation. Proceedings of the 11th Annual Cyber and Information Security Research Conference on - CISRC '16, , 1–4, 2016.

[68] Planquart, J.-P., Application of neural networks to intrusion detection. , 2001.

[69] Alcaraz, C., Cazorla, L., Fernandez, G., Context-Awareness Using Anomaly-Based Detectors for Smart Grid Domains. CRiSIS 2014, , 17–34, 2014.

[70] Sheng, J., Chung, S., Hansel, L., McLane, D., Morrah, J., Baeg, S.-H., Park, S., JAUS to EtherCAT Bridge: Toward Real-Time and Deterministic Joint Architecture for Unmanned Systems. Journal of Control Science and Engineering, 2014, 1–20, 2014.

[71] Knapp, E., Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems. Waltham, MA: Syngress., , 2011.

[72] Lei, W., Junyan, Q., The real-time networked data gathering systems based on EtherCAT. Proceedings - 2009 International Conference on Environmental Science and Information Application Technology, ESIAT 2009, 3, 513–515, 2009.

[73] Feng, T., Li, Q., Ren, G., Wang, H., Wang, F., Zhang, W., Yang, B., Wu, J., The implementation of distributed high-speed high-accuracy data acquisition system based on EtherCAT. Proceedings of the 2013 IEEE 8th Conference on Industrial Electronics and Applications, ICIEA 2013, , 1649–1653, 2013.

[74] Qi, J., Wang, L., Jia, H., Yang, B., Design and performance evaluation of networked data acquisition systems based on EtherCAT. 2010 2nd IEEE International Conference on Information Management and Engineering, , 467–469, 2010.

[75] Song, I.-S., Jeon, Y.-H., Kim, J.-H., Seo, S.-H., Kwon, K.-H., Chun, J.-H., Jeon, J.-W., Implementation and analysis of the embedded master for EtherCAT,. International Conference on Control, Automation and Systems, , 2010.

[76] Jian Wang, Hong Wang, Zhi-jia Yang, An FPGA based slave communication controller for Industrial Ethernet. 2008 9th International Conference on Solid-State and Integrated-Circuit Technology, , 2062–2065, 2008.

[77] Gong, F. M., EtherCAT Technology with Design of EtherCAT Slave Station. Applied Mechanics and Materials, 707, 368–371, 2014.

[78] Park, J. H., Lee, S., Lee, K. C., Lee, Y. J., Implementation of IEC61800 based EtherCAT slave module for real-time multi-axis smart driver system. Control Automation and Systems (ICCAS), , 2010.

[79] Orfanus, D., Indergaard, R., Prytz, G., Wien, T., EtherCAT-based platform for distributed control in high-performance industrial applications. 2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA), , 1–8, 2013.

[80] Cena, G., Bertolotti, I. C., Valenzano, A., Zunino, C., A high-performance CAN-like arbitration scheme for EtherCAT. 2009 IEEE Conference on Emerging Technologies & Factory Automation, , 1–8, 2009.

[81] Farkas, L., Janicek, L., Murgas, J., Hnat, J., Interconnecting Matlab with TwinCAT. International Conference on Manufacturing Engineering, Quality and Production Systems; Recent advances in manufacturing engineering, , 2011.

[82] Knezic, M., Dokic, B., Ivanovic, Z., Increasing EtherCAT performance using frame size optimization algorithm. ETFA2011, , 1–4, 2011.

[83] Knezic, M., Ivanovic, Z., Evaluation of Ethernet over EtherCAT Protocol Efficiency. Infoteh-Jahorina, , 2013.

[84] Kalman, G., Orfanus, D., Measuring latencies over industrial Ethernet switches. 2013 21st Telecommunications Forum Telfor (TELFOR), , 365–368, 2013.

[85] Granat, A., Höfken, H., Schuba, M., Intrusion Detection of the ICS Protocol EtherCAT. 2nd International Conference on Computer, Network Security and Communication Engineering, , 113–117, 2017.

[86] Argon, O., Shavitt, Y., Weinsberg, U., Inferring the periodicity in large-scale internet measurements. 2013 Proceedings IEEE INFOCOM, , 1672–1680, 2013.

[87] Gu, G., Zhang, J., Lee, W., BotSniffer: detecting botnet command and control channels in network traffic. Proceedings of the Fifteenth Annual Network and Distributed System Security Symposium, , 2008.

[88] Splunder, J. Van, Instituut, M., Leiden, U., Periodicity detection in network traffic. Institute of Mathematics, University of Leiden, , 2015.

[89] Barbosa, R. R. R., Sadre, R., Pras, A., Towards periodicity based anomaly detection in SCADA networks. Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012), , 1–4, 2012.

[90] J. Han, G. Dong, Y. Yin, Efficient mining of partial periodic patterns in time series database. Proceedings 15th International Conference on Data Engineering (Cat. No.99CB36337), , 106–115, 1999.

[91] Barbosa, R. R. R., Sadre, R., Pras, A., Exploiting traffic periodicity in industrial control networks. International Journal of Critical Infrastructure Protection, 13, 52–62, 2016.

[92] Mongeau, S. A., Continuous fraud monitoring and detection via advanced analytics. 24th Annual ACFE Global Fraud Conference, , 2014.

[93] Broido, A., Nemeth, E., kc claffy, Spectroscopy of private DNS update sources. Proceedings the Third IEEE Workshop on Internet Applications. WIAPP 2003, , 19–29, 2003.

[94] Broido, A., Nemeth, E., Claffy, K. C., Spectroscopy of DNS update traffic. Proceedings of the 2003 ACM SIGMETRICS international conference on Measurement and modeling of computer systems - SIGMETRICS ’03, , 320, 2003.

[95] Bhuyan, M. H., Bhattacharyya, D. K., Kalita, J. K., Network Anomaly Detection: Methods, Systems and Tools. IEEE Communications Surveys & Tutorials, 16 (1), 303–336, 2014.

[96] Ahmed, M., Naser Mahmood, A., Hu, J., A survey of network anomaly detection techniques. Journal of Network and Computer Applications, 60, 19–31, 2016.

[97] Barford, P., Plonka, D., Characteristics of network traffic flow anomalies. Proceedings of the First ACM SIGCOMM Workshop on Internet Measurement - IMW ’01, , 69, 2001.

[98] Sestito, G. S., Turcato, A. C., Dias, A. L., Rocha, M. S., da Silva, M. M., Ferrari, P., Brandao, D., A Method for Anomalies Detection in Real-Time Ethernet Data Traffic Applied to PROFINET. IEEE Transactions on Industrial Informatics, 14 (5), 2171–2180, 2018.

[99] Junejo, K. N., Goh, J., Behaviour-Based Attack Detection and Classification in Cyber Physical Systems Using Machine Learning. Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security - CPSS ’16, , 34–43, 2016.

[100] Del Grosso, C., Antoniol, G., Merlo, E., Galinier, P., Detecting buffer overflow via automatic test input data generation. Computers & Operations Research, 35 (10), 3125–3143, 2008.

[101] Sparks, S., Embleton, S., Cunningham, R., Zou, C., Automated Vulnerability Analysis: Leveraging Control Flow for Evolutionary Input Crafting. Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007), , 477–486, 2007.

[102] Maglaras, L. A., Jiang, J., Intrusion detection in SCADA systems using machine learning techniques. 2014 Science and Information Conference, , 626–631, 2014.

[103] Lee, S., Yoo, H., Seo, J., Shon, T., Packet Diversity-Based Anomaly Detection System with OCSVM and Representative Model. 2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), , 498–503, 2016.

[104] Lahrache, A., Cocconcelli, M., Rubini, R., Anomaly detection in a cutting tool by K-Means clustering and Support Vector Machines. Diagnostyka, 18 (3), 21–29, 2017.

[105] Ahmad, I., Hussain, M., Alghamdi, A., Alelaiwi, A., Enhancing SVM performance in intrusion detection using optimal feature subset selection based on genetic principal components. Neural Computing and Applications, , 2014.

[106] Yoo, H., Shon, T., Novel Approach for Detecting Network Anomalies for Substation Automation based on IEC 61850. Multimedia Tools and Applications, , 2014.

[107] Ran, C., Yu, Z., Ruwei, H., Ting, L., Hongchun, Y., Combined control strategy of wind farm and battery storage based on integrated monitoring system. 2015 Sixth International Conference on Intelligent Systems Design and Engineering Applications (ISDEA), , 218–221, 2015.

[108] Laxman, S., Sastry, P. S., A survey of temporal data mining. Sadhana, 31 (2), 173–198, 2006.

[109] de Cheveigné, A., Kawahara, H., YIN, a fundamental frequency estimator for speech and music. The Journal of the Acoustical Society of America, 111 (4), 1917–30, 2002.

[110] Sommer, R., Paxson, V., Outside the closed world: on using machine learning for network intrusion detection. 2010 IEEE Symposium on Security and Privacy, , 305–316, 2010.

[111] Ibrahim, L. M., Anomaly network intrusion detection system based on distributed time-delay neural network. Journal of Engineering Science and Technology, 5 (4), 457–471, 2010.

[112] Cheminod, M., Durante, L., Valenzano, A., Review of Security Issues in Industrial Networks. IEEE Transactions on Industrial Informatics, 9 (1), 277–293, 2013.

[113] Ericsson, G. N., Cyber security and power system communication-essential parts of a smart grid infrastructure. IEEE Transactions on Power Delivery, 25 (3), 1501–1507, 2010.

[114] Gao, W., Morris, T., Reaves, B., Richey, D., On SCADA control system command and response injection and intrusion detection. 2010 eCrime Researchers Summit, , 1–9, 2010.

[115] Yang, D., Usynin, A., Hines, W. J., Anomaly-based intrusion detection for SCADA systems. 5th intl. topical meeting on nuclear plant instrumentation, control and human machine interface technologies (npic&hmit 05), , 2006.

[116] Kim, S.-J., Kim, B.-H., Yeo, S.-S., Cho, D.-E., Network anomaly detection for m-connected SCADA networks. 2013 Eighth International Conference on Broadband and Wireless Computing, Communication and Applications, , 351–354, 2013.

[117] Hadeli, H., Schierholz, R., Braendle, M., Tuduce, C., Leveraging determinism in industrial control systems for advanced anomaly detection and reliable security configuration. 2009 IEEE Conference on Emerging Technologies & Factory Automation, , 1–8, 2009.

[118] Morris, T. H., Jones, B. A., Vaughn, R. B., Dandass, Y. S., Deterministic intrusion detection rules for MODBUS protocols. 2013 46th Hawaii International Conference on System Sciences, , 1773–1781, 2013.

[119] Erez, N., Wool, A., Control variable classification, modeling and anomaly detection in Modbus/TCP SCADA systems. International Journal of Critical Infrastructure Protection, 10, 59–70, 2015.

[120] Quevedo, J., Puig, V., Cembrano, G., Blanch, J., Aguilar, J., Saporta, D., Benito, G., Hedo, M., Molina, A., Validation and reconstruction of flow meter data in the Barcelona water distribution network. Control Engineering Practice, 18 (6), 640–651, 2010.

[121] Chabukswar, R., Sinópoli, B., Karsai, G., Giani, A., Neema, H., Davis, A., Simulation of Network Attacks on SCADA Systems. Workshop on Secure Control Systems, , 2010.

[122] Morris, T., Srivastava, A., Reaves, B., Gao, W., Pavurapu, K., Reddi, R., A control system testbed to validate critical infrastructure protection concepts. International Journal of Critical Infrastructure Protection, 4 (2), 88–103, 2011.

[123] Wang, C., Fang, L., Dai, Y., A simulation environment for SCADA security analysis and assessment. 2010 International Conference on Measuring Technology and Mechatronics Automation, ICMTMA 2010, , 2010.

[124] Mathioudakis, K., Frangiadakis, N., Merentitis, A., Gazis, V., Towards generic SCADA simulators: A survey of existing multi-purpose co-simulation platforms, best practices and use-cases. Conf-Scoop.Org, .

[125] Adnan, R., Kregel, B., Fitzpatrick, J., Govindarasu, M., Sridhar, S., Higdon, M., Hahn, A., Development of the PowerCyber SCADA security testbed. , 2010.

[126] Gedwillo, T., Parrott, J., Ryan, D., Cyber Security of SCADA Systems Test Bed – Design Document. , 2011.

[127] Neema, S., Bapty, T., Koutsoukos, X., Neema, H., Sztipanovits, J., Karsai, G., Model Based Integration and Experimentation of Information Fusion and C2 Systems. Fusion: 2009 12Th International Conference on Information Fusion, Vols 1-4, , 2009.

[128] Liu, Y., EtherCAT-based Functional Safety-Integrated Communication. , 1005–1008.

[129] EtherCAT, EtherCAT Communication Principles. , 2018.

[130] Rathaus, N., Evron, G., Open Source Fuzzing Tools. 1 edition. Ed., Syngress, 210 , 2007.

[131] Eddington, M., Advanced Fuzzing with Peach 2. , 2016.

[132] Aven, T., A unified framework for risk and vulnerability analysis covering both safety and security. Reliability Engineering and System Safety, 92 (6), 745–754, 2007.

[133] Sevuktekin, M., Nargelecekenler, M., Ekonometrik Zaman-Serileri Analizi – Eviews Uygulamali. 3rd ed. Ed., Nobel Akademik Yayincilik, 494 , 2005.

[134] DeLurgio, S. A., Forecasting Principles and Applications. Irwin/McGraw-Hill, Boston MA, , 1998.

[135] Hipel, K. W., McLeod, A. I., Time Series Modelling of Water Resources and Environmental Systems. 1st ed. Ed., Elsevier Science, 73 , 1994.

[136] Ierusalimschy, R., de Figueiredo, L. H., Celes, W., The evolution of Lua. Proceedings of the third ACM SIGPLAN conference on History of programming languages - HOPL III, , 2-1-2–26, 2007.

[137] Dawdy, D. R., Matalas, N. C., Statistical and probability analysis of hydrologic data, part III: analysis of variance, covariance and time series. Handbook of Applied Hydrology, a Compendium of Water-Resources Technology, Ven Te Cho. Ed., New York, McGraw-Hill Book Company, , 68–90, 1964.

[138] Akpinar, M., Yumusak, N., Naive forecasting of household natural gas consumption with sliding window approach. Turkish Journal of Electrical Engineering & Computer Sciences, 25, 30–45, 2017.

[139] Stuart, M., Ross, S. M., Introduction to Probability and Statistics for Engineers and Scientists. Journal of the Royal Statistical Society. Series A (Statistics in Society), , 2006.

[140] Abraham, V. M., Walpole, R. E., Myers, R. H., Probability and Statistics for Engineers and Scientists. The Mathematical Gazette, , 2007.

[141] Gupta, B. C., Guttman, I., Statistics and Probability with Applications for Engineers and Scientists. 1st editio. Ed., Wiley, 896 , 2013.

EKLER

EK 1: Snort tarafından tespit edilen ve günlükleme olarak tutulan saldırıların kullanıcı tarafında izlenmesi için kurulan ELK (Elasticsearch-Logstash-Kibana) ortamı açıklamaktadır.

1.1. Sunucu Tarafında Yapılan İşlemler

1.1.1. Java kurulumu

ELK yığını üzerinde işlem yapılması için Java gereklidir. Kurulum 64 bitlik sistemler için olan aşağıdaki versiyonu ile yapılmıştır.

Java kurulumu:
sudo add-apt-repository -y ppa:webupd8team/java
sudo apt-get update
sudo apt-get -y install oracle-java8-installer

1.1.2. Elasticsearch kurulum ve yapılandırılması

Elasticsearch bir nosql veritabanıdır. Tüm metin araması, analiz gibi özelliklere sahiptir. Aşağıda kurulum detayları verilmiştir.

Elasticsearch açık GPG anahtarının eklenmesi:
wget -qO - https://packages.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add –

Elasticsearch kaynak listesinin eklenmesi:
echo "deb http://packages.elastic.co/elasticsearch/2.x/debian stable main" | sudo tee -a /etc/apt/sources.list.d/elasticsearch-2.x.list

Apt paket veritabanın güncellenmesi:
sudo apt-get update

Elasticsearch kurulması:
sudo apt-get -y install elasticsearch

Elasticsearch konfigürasyonu:
sudo vi /etc/elasticsearch/elasticsearch.yml
yml dosyasındaki ilgili satırın yorumdan kaldırılması:
network.host: localhost

Servisin başlatılması:
sudo service elasticsearch restart
sudo update-rc.d elasticsearch defaults 95 10

1.1.3. Kibana kurulum ve yapılandırılması

Elastic'teki günlüklemelerin anlık olarak takip edilmesini ve istatistik alarak grafik şeklinde görüntülenmesini sağlayan bir web uygulamasıdır. Aynı zamanda eski tarihli günlüklemeleri de analiz etmeyi desteklemektedir.

Kibana kaynak listesinin oluşturulması:
echo "deb http://packages.elastic.co/kibana/4.5/debian stable main" | sudo tee -a /etc/apt/sources.list.d/kibana-4.5.x.list

Apt veritabanının güncellenmesi:
sudo apt-get update

Kibana yüklenmesi:
sudo apt-get -y install kibana

Kibana yapılandırılması:
sudo vi /opt/kibana/config/kibana.yml

İlgili satırın güncellenmesi:
server.host: "localhost"

Servisin başlatılması:
sudo update-rc.d kibana defaults 96 9
sudo service kibana start

1.1.4. Nginx (Kibana erişimi için Reverse Proxy) kurulum ve yapılandırılması

Kibana Localhost'u dinlediği için dışarıdan erişimler için Reverse Proxy gereklidir. Bunu için aşağıdaki işlemler yapılmalıdır.

Nginx kurulması:
sudo apt-get install nginx apache2-utils

kibanaadmin kullanıcısının oluşturulması:
sudo htpasswd -c /etc/nginx/htpasswd.users kibanaadmin

Varsayılan dosyanın açılması ve içeriğinin silinip aşağıdaki gibi oluşturulması (sunucu ismi kullanıcınınkiyle aynı olmalı):
sudo vi /etc/nginx/sites-available/default

```
server {
   listen 80;
   server_name example.com;
   auth_basic "Restricted Access";
   auth_basic_user_file /etc/nginx/htpasswd.users;
   location / {
      proxy_pass http://localhost:5601;
      proxy_http_version 1.1;
      proxy_set_header Upgrade $http_upgrade;
      proxy_set_header Connection 'upgrade';
      proxy_set_header Host $host;
      proxy_cache_bypass $http_upgrade;
```

```
    }
}
```

Servisin başlatılması:
sudo service nginx restart

1.1.5. Logstash kurulum ve yapılandırılması

Sunucudaki günlüklemeleri Filebeat aracı ile toplayıp, filtreledikten sonra Elasticsearch programına gördermekten sorumludur. Günlüklemeler ile Elasticsearch arasında köprü vazifesi görmektedir.

Kaynak liste oluşturulması:
echo 'deb http://packages.elastic.co/logstash/2.2/debian stable main' | sudo tee /etc/apt/sources.list.d/logstash-2.2.x.list

Paket veritabanının güncellenmesi:
sudo apt-get update

Logstash yüklenmesi:
sudo apt-get install logstash

Konfigürasyon dosyası oluşturulması ve içeriğinin doldurulması (beats-input.conf konfigürasyonu):
sudo vi /etc/logstash/conf.d/02-beats-input.conf

```
input {
  beats {
    port => 5044
    ssl => true
    ssl_certificate => "/etc/pki/tls/certs/logstash-forwarder.crt"
    ssl_key => "/etc/pki/tls/private/logstash-forwarder.key"
  }
}
```

Syslog mesajlarını filtreleyen konfigürasyon dosyası oluşturulması ve içeriğinin doldurulması (syslog-filter.conf konfigürasyonu):
sudo vi /etc/logstash/conf.d/10-syslog-filter.conf

```
filter {
  if [type] == "syslog" {
    grok {
      match => { "message" => "%{SYSLOGTIMESTAMP:syslog_timestamp}
%{SYSLOGHOST:syslog_hostname}
%{DATA:syslog_program}(?:\[%{POSINT:syslog_pid}\])?:
%{GREEDYDATA:syslog_message}" }
      add_field => [ "received_at", "%{@timestamp}" ]
      add_field => [ "received_from", "%{host}" ]
    }
    syslog_pri { }
    date {
      match => [ "syslog_timestamp", "MMM  d HH:mm:ss", "MMM dd HH:mm:ss" ]
    }
  }
}
```

Çıkış konfigürasyon dosyası oluşturulması ve içeriğin doldurulması (elasticsearch-output.conf konfigürasyonu):
sudo vi /etc/logstash/conf.d/30-elasticsearch-output.conf

```
output {
  elasticsearch {
    hosts => ["localhost:9200"]
    sniffing => true
    manage_template => false
    index => "%{[@metadata][beat]}-%{+YYYY.MM.dd}"
    document_type => "%{[@metadata][type]}"
  }
```

}

Logstash konfigürasyonunun test edilmesi:
sudo service logstash configtest

Servisin başlatılması:
sudo service logstash restart
sudo update-rc.d logstash defaults 96 9

1.1.6. SSL sertifikalarının üretilmesi

Filebeat üzerinden ELK yığınına günlükleme yollanması için SSL sertifikalarının oluşturulması ve anahtar eşleşmesinin yapılması gereklidir.

Klasör oluşturulması:
sudo mkdir -p /etc/pki/tls/certs
sudo mkdir /etc/pki/tls/private

IP adres üzerinden SSL sertifika üretilmesi:
sudo vi /etc/ssl/openssl.cnf
subjectAltName = IP: ELK_server_private_IP ([v3_ca] bölümünde Elk sunucu IP adresi ile değiştirilmelidir)
cd /etc/pki/tls
sudo openssl req -config /etc/ssl/openssl.cnf -x509 -days 3650 -batch -nodes -newkey rsa:2048 -keyout private/logstash-forwarder.key -out certs/logstash-forwarder.crt

1.1.7. "Kibana Dashboard" belirlenmesi

cd ~
curl -L -O https://download.elastic.co/beats/dashboards/beats-dashboards-1.1.0.zip
sudo apt-get -y install unzip
unzip beats-dashboards-*.zip
cd beats-dashboards-*
./load.sh

1.1.8. Filebeat indeks taslağının Elasticsearch üzerine yüklenmesi

cd ~
curl –O https://gist.githubusercontent.com/thisismitch/3429023e8438cc25b86c/raw/d8c479e2a1adcea8b1fe86570e42abab0f10f364/filebeat-index-template.json
curl -XPUT 'http://localhost:9200/_template/filebeat?pretty' -d@filebeat-index-template.json

1.2. İstemci Tarafında Yapılan İşlemler

1.2.1. SSL sertifikalarının istemci tarafa aktarımı

ELK sunucuda:
scp /etc/pki/tls/certs/logstash-forwarder.crt user@client_server_private_address:/tmp

İstemci tarafında:
sudo mkdir -p /etc/pki/tls/certs
sudo cp /tmp/logstash-forwarder.crt /etc/pki/tls/certs/

1.2.2. Filebeat kurulum ve yapılandırılması

Filebeat kaynak listesi oluşturulması ve GPG anahtarının yüklenmesi:
echo "deb https://packages.elastic.co/beats/apt stable main" | sudo tee -a /etc/apt/sources.list.d/beats.list
wget -qO - https://packages.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add –

Filebeat kurulumu:
sudo apt-get update
sudo apt-get install filebeat

Filebeat yapılandırılması:
sudo vi /etc/filebeat/filebeat.yml

Konfigürasyon dosyasında ilgili satırın yorum satırına çekilmesi:

```
    paths:
      - /var/log/auth.log
      - /var/log/syslog
#       - /var/log/*.log
```

“document_type:” satırının yorumdan kaldırılması ve güncellenmesi:

```
document_type: syslog
```

Ayrıca aynı dosyada “output” bölümünün “elasticsearch:” satırının silinmesi, “#logstash:” satırının yorumdan kaldırılması, “hosts: ["localhost:5044"]” satırının yorumdan kaldırılması, “localhost” yerine ELK sunucunun IP adresinin eklenmesi gerekmektedir.

Dosyadaki “hosts” girdinin altına ilgili satırın eklenmesi:

```
bulk_max_size: 1024
```

“tls” bölümünün yorumdan kaldırılması ve ilgili satırın yorumdan kaldırılması ve güncellenmesi:

```
tls:
    # List of root certificates for HTTPS server verifications
    certificate_authorities: ["/etc/pki/tls/certs/logstash-forwarder.crt"]
```

Filebeat servisinin yeniden başlatılması:

```
sudo service filebeat restart
sudo update-rc.d filebeat defaults 95 10
```

Elk sunucuda Filebeat testi (ekranda ID, log, index parametreleri var ise doğru çalışmaktadır):

```
curl -XGET 'http://localhost:9200/filebeat-*/_search?pretty'
```

Kibana ortamına erişim:

Tarayıcı üzerinden IP adresi ile erişim sağlanabilmektedir.

ÖZGEÇMİŞ

Kevser Ovaz Akpınar, 03.06.1986'da Kastamonu'da doğdu. İlk ve orta eğitimini Kuzey Kıbrıs Türk Cumhuriyeti'nde, lise eğitimini Aydın'da tamamladı. 2004 yılında Söke Hilmi Fırat Anadolu Lisesi'nden mezun oldu. 2004 yılında başladığı Pamukkale Üniversitesi Bilgisayar Mühendisliği bölümünü 2008 yılında bitirdi. Yine 2006 yılında başladığı Anadolu Üniversitesi Halkla İlişkiler ve Tanıtım bölümünü 2017'de tamamladı. 2008-2012 yılları arasında Amerika Birleşik Devletlerinde University of Texas at San Antonio'da bilgisayar bilimlerinde yüksek lisans eğitimini tamamladı. 2013 yılında Sakarya Üniversitesinde doktora eğitimine başladı ve araştırma görevlisi olarak çalışmaya başladı. Halen Sakarya Üniversitesi Bilgisayar Mühendisliği bölümünde Araştırma Görevlisi olarak görev yapmakta ve doktora eğitimini sürdürmektedir.