GU J Sci, Part C, 6(1): 1-16 (2018)

Gazi Üniversitesi

Fen Bilimleri Dergisi

PART C: TASARIM VE TEKNOLOJİ

dergipark.gov.tr/http-gujsc-gazi-edu-tr

Endüstriyel Kontrol Sistemleri ve SCADA Uygulamalarının Siber Güvenliği: Modbus TCP Protokolü Örneği

Erdal IRMAK[1,*], İsmail ERKEK[2]

[1]*Gazi Üniversitesi, Teknoloji Fakültesi, Elektrik Elektronik Mühendisliği Bölümü, 06500, ANKARA*

[2]*Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Bilgi Güvenliği Mühendisliği Lisansüstü Programı, 06500, ANKARA*

Makale Bilgisi

Başvuru: 25/12/2017
Düzeltme: 20/01/2018
Kabul: 25/01/2018

Anahtar Kelimeler

Siber Güvenlik
SCADA
Modbus TCP

Keywords

Cyber Security
SCADA
Modbus TCP

Öz

Elektrik üretim, iletim ve dağıtım sistemleri ulusal güvenlik boyutunda değerlendirilmekte olup kritik altyapılar olarak tanımlanmaktadır. Bu sistemlerin izlenmesi ve denetimi Endüstriyel Kontrol Sistemleri (EKS) veya Danışmalı Kontrol ve Veri Toplama Sistemleri (SCADA) ile sağlanmaktadır. Haberleşme ve internet teknolojisindeki güncel gelişmelere bağlı olarak EKS/SCADA sistemleri de bilişim teknolojileriyle entegre çalışır hale gelmiştir. Bu duruma paralel olarak bilgi ve iletişim teknolojisinde mevcut olan veya ortaya çıkan güvenlik zafiyetleri SCADA sistemlerini de direkt olarak etkileyebilmektedir. Bu nedenle çalışmada, EKS/SCADA sistemlerinin siber güvenliği üzerinde durulmuş ve bu sistemlerde en fazla kullanılan endüstriyel haberleşme protokollerinden birisi olan Modbus TCP protokolünde tespit edilen kimlik doğrulama eksikliğinin istismar edilebildiği ortaya konulmuştur. Bu güvenlik sorununa çözüm olarak saldırıyı engellemeye veya hafifletmeye yönelik Python programlama dili kullanılarak bir program yazılmıştır. Önerilen çözüm, çeşitli testlere tabi tutulmuş ve gerçekleştirilen siber saldırıların engellenebildiği ispatlanmıştır. Sunulan çalışmanın, EKS/SCADA sistemlerinin ve bu sistemlerin haberleşmesinde kullanılan endüstriyel protokollerin güvenliğine katkılar sağlayacağı değerlendirilmektedir.

Cyber Security in Industrial Control Systems and SCADA Applications: Modbus TCP Protocol Example

Abstract

Electrical energy generation, transmission and distribution systems are evaluated in terms of national security dimension and defined as critical infrastructures. Monitoring and controlling of these systems are provided by Industrial Control Systems (ICS) or Supervisory Control and Data Acquisition (SCADA) systems. According to the latest advances in communication and internet technology, ICS/SCADA systems have started to become integrated with these systems. As a result of this situation, current or existing vulnerabilities in information and communication technology affect SCADA systems directly. Therefore, this paper focuses on the cyber security of ICS/SCADA systems. It has been proved that the lack of authentication detected in Modbus TCP protocol, one of the most used in ICS/SCADA systems, can be exploited. In order to solve this security issue, a software is developed using the Python programming language for blocking or mitigating the cyber attacks. The proposed solution is subjected to several tests and results show that the attacks can be prevented successfully. Thus, it is considered that the proposed work will contribute to the security of ICS/SCADA systems and the industrial protocols used for communicating these systems.

1. GİRİŞ (INTRODUCTION)

Siber bağımlılığın giderek arttığı günümüzde internet teknolojisi ülkeler arasındaki sınırları ortadan kaldırmış, küresel bir bilgi ağı haline gelerek küreselleşme kavramının artmasında da önemli bir rol üstlenmiştir. Özellikle son yıllarda oldukça gelişme kaydeden nesnelerin interneti, büyük veri, bulut bilişim gibi teknolojilerin internet teknolojisiyle entegre çalışarak kullanımının artması, toplum hayatını ve yaşam tarzını tahmin edilmesi güç boyutlara götüreceğini düşündürmektedir.

*İletişim yazarı, e-mail: erdal@gazi.edu.tr

DOI: 10.29109/http-gujsc-gazi-edu-tr.364411

Bilgi teknolojileri kullanımı hız, kolaylık, şeffaflık ve maliyet gibi nedenlerle toplum hayatını kolaylaştırmaktadır. Ancak kullanılan bu teknolojinin sunduğu avantajların yanında getirdiği bir takım riskler ve dezavantajlar da mevcuttur. Zafiyet ve riskleri de beraberinde getiren bu sistemlerin güvenliği, toplumların refahı ve kamusal düzenin sağlanması açısından önemlidir.

Günümüzde elektrik, doğalgaz, su, kanalizasyon, ulaşım, ilaç ve kimya sanayisi, kağıt hamuru ve kağıt sanayisi, yiyecek içecek sektörü ve parçalı üretim sanayisinde kontrol ve izleme sistemleri ile sıkça kullanılmaktadır. Öte yandan bu sistemlere yönelik fiziksel ve siber saldırılarla yetkisiz erişim elde edilebilmekte ve farklı müdahalelerle bu sistemlerin işleyişi ve fonksiyonu değiştirilebilmekte ve bozulabilmektedir [1].

Literatürde SCADA sistemleri olarak adlandırılan denetim sistemleri ilk tasarlandığı dönemlerde, internet teknolojisi, kablolu ve kablosuz erişim teknolojileri günümüz teknolojisi ile kıyaslanamayacak düzeydeydi [2]. SCADA sistemleri diğer bilgi teknolojileri altyapılarından izole olarak tasarlanmaktaydı [3]. Ayrıca bu sistemlerin üzerinde çalışan işletim sistemlerinin ve uygulamaların güvenlik açıkları da izole sistemler olma düşüncesiyle dikkate alınmamaktaydı. Fakat günümüzde internet teknolojisinin gelişmesiyle birlikte SCADA sistemlerini yöneten operatörler, bu sistemleri internet teknolojisini kullanarak izlemekte ve kontrol etmektedir. Ayrıca SCADA sistemi bileşenleri kendi aralarında haberleşmek için yine internet teknolojisini kullanmaktadır. Dolayısıyla, internet teknolojisinde ortaya çıkan güvenlik zafiyetleri bu teknoloji altyapısını kullanan SCADA sistemlerini de direk veya dolaylı olarak etkilemekte ve risk altında olmasına sebep olabilmektedir.

Yapılan literatür araştırmasında kritik altyapıların fiziksel ve siber güvenliğine yönelik geliştirilen test ortamlarında bu sistemlere gerçekleştirilen siber saldırılar ve farklı güvenlik çözüm önerileri yer almaktadır. SCADA haberleşme protokollerinden en yaygın olarak kullanılan Modbus protokolünün test edilmesi ve simüle edilmesi amacıyla uygun bir yazılım platformu olan Modbus Poll, SCADA sistemlerinin siber güvenlik çalışmalarında kullanılmıştır. Yanfei ve arkadaşları Modbus tabanlı ZigBee kablosuz sensör teknolojisi geliştirirken yazılım test aracı olarak Modbus Poll kullanmışlardır [4,5].

Sağıroğlu ve arkadaşlarının yaptığı çalışmada [6], enerji otomasyon sistemlerinin haberleşmesinde kullanılan Modbus protokolünün güvenlik açıklıkları incelenmiş, tespit edilen güvenlik açığını gidermek ve sistemi güvenli hale getirmek için açık kaynak kodlu güvenlik duvarı kullanılmıştır.

Xiong ve arkadaşları Modbus TCP protokolünün zafiyet analizi için geleneksel fuzzing (bulandırma) yöntemleri dışında Modbus TCP protokolüne özel bir fuzzing metodolojisi geliştirmişlerdir [7]. Uyguladıkları simülasyon ortamındaki sonuçlara göre, geliştirdikleri fuzzing teknolojisinin geleneksel fuzzing işlemlerine göre daha etkili olduğunu vurgulamışlardır.

Bhatia ve arkadaşları Modbus protokolünün sel (flooding) saldırılarına karşı zafiyet barındırdığını ve bu saldırıların kontrol sisteminin fonksiyonlarını bozmaya yönelik komut enjeksiyonu da içerdiğinden bahsetmişlerdir [8]. Çalışmada, bu saldırıları tespit etmeye yönelik anomali tabanlı tespit algoritması ve imza tabanlı Snort eşik modülü karşılaştırılmıştır. Sonuç olarak, imza tabanlı tespit sisteminin eşik değerlerinin çok dikkatli belirlenmesi gerektiği ve anomali tabanlı tespit sisteminin saldırı belirlemede çok az bir zaman kayması sonucu başarılı olduğunu belirlemişlerdir.

Shang ve arkadaşlarının yaptıkları çalışmada [9] endüstriyel kontrol sistemlerinde karşılaşılan zararlı yazılımlara yönelik STS'leri incelenmiş ve Modbus TCP protokolü uygulama katmanından gelen tehditlere yönelik paket boyutunda derinlemesine analiz edilmiştir. Modbus TCP haberleşmesi için savunma modeli olarak STS tabanlı bir "White list" kuralı önerilmiştir. Bo Chen ve arkadaşlarının yaptıkları çalışmada [10] akıllı şebekelerin siber güvenliği için gerçek-zamanlı güç sistemi simülatörü ve haberleşme sistemi simülatörünü entegre çalıştırarak bir deney düzeneği ortamı hazırlamışlardır. Güç sistemi simülasyonu için RTDS (Real-Time Digital Simulator) güç şebekesi simülatörü kullanılmış olup, haberleşme sistemi simülasyonu için Opnet'in SITL (System-in-the-Loop) simülatörü ve açık kaynak kodlu Linux araçları ve sunucuları kullanılmıştır. Bu deney düzeneğinde Modbus TCP protokolüne yönelik iki çeşit siber saldırı düzenlenmiş ve bu saldırıların tespitine ve engellenmesine yönelik önerilerde bulunmuşlardır.

Avrupa CRUTIAL projesi kapsamında farklı siber saldırı senaryolarının etkilerini gözlemlemek amacıyla 2 adet deney düzeneği geliştirilmiştir. İlk deney düzeneğinde öncelikli olarak denetim merkezi ve simüle

edilmiş alt sistemler arasındaki haberleşme altyapısına dikkat edilmiştir. Bu sistemlerin haberleşme altyapısına yönelik belirli DoS saldırıları analiz edilmiştir. Diğer test düzeneğinde emüle edilmiş Akıllı Elektronik Cihazlar (IED – Intelligent Electronic Device) tarafından kontrol edilen fiziksel mikro şebeke altyapısı temel alınmıştır. Buradaki IED'ler yerel ağ üzerinden Matlab/Simulink kullanılarak haberleşmektedir. Bu ortam Dağıtık Enerji Kaynağı (DER – Distributed Energy Resource) uygulamalarında potansiyel güvenlik açıklarının tespitinde kullanılmıştır [11,12].

Anomali tabanlı STS araştırmaları için SCADA Denetim Sistemlerinin Güvenlik Analizi için Deney Düzeneği (TASSCS - The Testbed for Analyzing Security of SCADA Control Systems) Arizona Üniversitesi'nde geliştirilmiştir. Sandia'daki VCSE projesine benzer OPNET döngü sistemi emülasyonu ve simüle elektrik şebekesi sağlamak amacıyla PowerWorld yazılımını kullanmıştır. Simülasyon tabanlı kontrol çözümleri PowerWorld simülatörüyle haberleşen ModbusRSim yazılımı kullanılarak sunulmuştur [13].

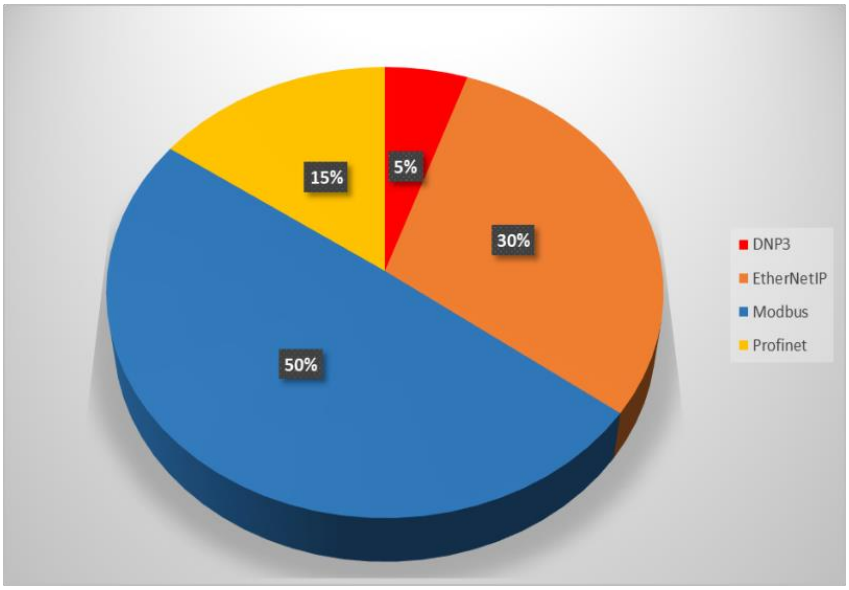
Yukarıda verilen literatür örneklerinden de görüldüğü gibi SCADA sistemlerinin ve bu sistemlerde sıkça kullanılan Modbus TCP protokolünün siber güvenliği oldukça önemlidir ve konuyla ilgili güncel çalışmalar yürütülmeye devam etmektedir. Bu nedenle sunulan bu çalışmada da Modbus TCP protokolüne dışarıdan ve içeriden müdahale edilerek SCADA bileşenleri arasında akan Modbus TCP paketlerinin manipüle edilmesi konusu üzerinde durulmuş, oluşturulan simülasyon test ortamında Modbus TCP paketleri manipüle edilmiş ve tasarlanan güvenlik kontrol mekanizmasıyla saldırılar engellenmiştir.

2. MODBUS TCP PROTOKOLÜ (MODBUS TCP PROTOCOL)

Modbus protokolü SCADA'ya özel geliştirilmiş ve endüstriyel standart haline gelmeye başlamış bir protokoldür. Farklı tip ağlara bağlı cihazlar arasındaki istemci/sunucu haberleşmesi için uygulama katmanında mesajlaşma görevi görmektedir. Günümüz uygulamalarında şu şekilde kullanılmaktadır:

- Ethernet üzerinden TCP/IP tabanlı haberleşme,
- Farklı ortamlar üzerinden (kablo: EIA/TIA-232-E, EIA-422, EIA/TIA-485-A, fiber optik, radio, vb) asenkron seri haberleşme,
- Modbus Plus, yüksek hızlı andaç geçirme (token passing) ağı üzerinden haberleşme sağlar.

Birçok üretici bu protokolü kullanmakta, sistemlerini geliştirmekte ve cihaz üretimi yapmaktadır. Şekil 1'de görüleceği üzere, göre endüstriyel denetim amaçlı SCADA sistemleri içerisinde, Modbus TCP protokolü %50 gibi ciddi bir kullanım oranına sahiptir [14].



Şekil 1. *SCADA haberleşme protokollerinin dağılımları.*

Modbus protokolü, belirli bir RTU'ya (remote terminal unit – uzak terminal birimi) istek mesajının gönderilerek ve tekrar yanıt alınan usta/köle (master/slave) prensibine göre çalışmaktadır. Eğer, protokol yayın tipindeyse hiçbir yanıt alınmaz. Veriler, ASCII ve RTU olmak üzere iki transmisyon modunda gönderilip alınabilir. RTU, daha kısa çerçeveye sahip olmasından ve eşlik denetimi, hata denetimi veya CRC olmasından dolayı tercih edilmektedir. ASCII modu daha uzun mesaj çerçevesine sahip olmasından dolayı sistemi yavaşlatmaktadır. Modbus protokolünün, Modbus Serial ve Modbus TCP olmak üzere iki adet değişkeni vardır. IP bağlantılı ağlarda çalışan Modbus TCP, MTU'ya (master terminal unit – ana terminal birimi) çoklu işlem özelliği sağlar ve RTU'nun çoklu sunucudan paralel işlem çalıştırmasına imkan verir [15]. Modbus protokolünün temel fonksiyonları aşağıdaki gibidir:

- Okuma için kontrol bobin komutları ve tekli veya grup bobin ayarlamaları
- Girdi gruplarının giriş durumlarını okumak için girdi kontrol komutları
- Bekleyen yazmaçları okumak ve ayarlamak için yazmaç kontrol komutları
- Hata bulma testi ve fonksiyon raporu
- Program fonksiyonları
- Sorgulama kontrol fonksiyonları
- Sıfırlama

Modbus protokolünün mesaj yapısı Şekil 2'de gösterilmiştir. Buna göre mesaj yapısının ilk alanı adresin depolandığı tek baytlık alandır. İstek çerçevesinde hedef adresin IP adresi, yanıt çerçevesinde MTU'nun IP adresi bulunur. Modbus protokolü en fazla 248 köle cihazına sahip olabilir fakat pratikte tek bir MTU, 2 veya 3 köle cihazla bağlantılıdır. İkinci alan, hedef cihazda uygulanacak fonksiyonları barındırır. İstek çerçevesinde bu bayt uygulanacak hedefin fonksiyonunu tanımlar. İstek hedef istasyonda başarılı bir şekilde tamamlanırsa fonksiyon alanı geri çağrılır, başarısız olunursa en soldaki bit gönderilir ve böylece problemli yanıt döner. Üçüncü alan veri bölümü olup fonksiyon kodundaki değişkendir. Son iki bayt, çerçevedeki hata denetimi için CRC alanıdır. Fonksiyon kodlarının bulunduğu fonksiyon alanı Tablo 1'de gösterilmiştir.

Modbus TCP protokolü hem IP tabanlı hem LAN tabanlı ağlarda çalışır. Şekil 3'te ana cihazın, IP tabanlı ağ üzerinden birçok köle cihaza bağlı olduğu durum gösterilmiştir. Modbus TCP protokolünde, köle cihaz sadece pasif işlemler yaptığı için ana cihaz istemci olarak tasarlandığında köle cihaz sunucu olarak tasarlanır.

Modbus TCP protokolü mesajlarını TCP paketlerinde kapsüllediğinden dolayı TCP PDU (protocol data unit), Modbus uygulama protokolünü (MBAP) barındırır. MBAP başlığı; işlem tanımlayıcı, protokol tanımlayıcı, uzunluk ve birim tanımlayıcı olmak üzere dört alan barındırır. İstek ve yanıtların çiftli eşleştirilmesi işlem tanımlayıcısı tarafından gerçekleştirilirken protokol tanımlayıcısı MBAP başlığı (Modbus için 0) tarafından kapsüllenen uygulama protokolünü gösterir. Birim tanımlayıcısı, işlemle ilgili köle cihazları gösterir ve sadece legal sistemleri için kullanılır. Uzunluk alanı veri paketinin kalan bayt miktarını gösterir [16].

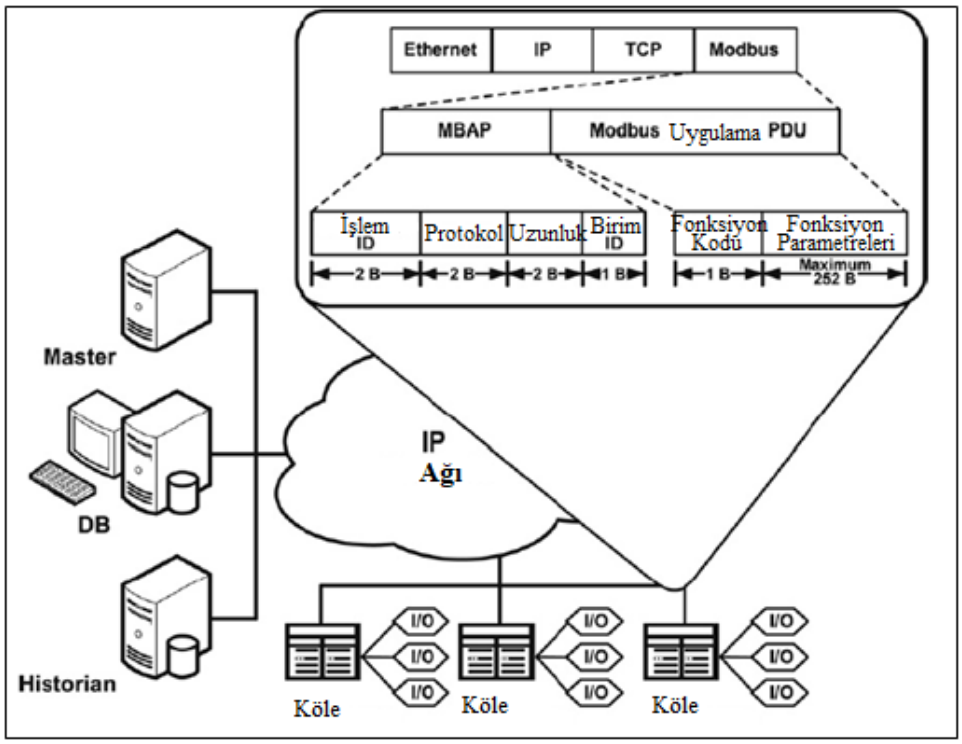
Modbus Mesajı

Adres	Fonksiyon	Veri	CRC Kontrolü
8 bit	8 bit	N x 8 bit	16 bit

Şekil 2. *Modbus mesaj yapısı.*

Tablo 1. *Modbus protokol çerçevesindeki fonksiyon kodları*

Kod	Hex	Fonksiyon	Tipi	
01	01	Sarmalları oku	Tek bit erişimi	Veri erişimi
02	02	Ayrık girişleri oku		
05	05	Tekli sarmala yaz		
15	0F	Çoklu sarmala yaz		
03	03	Bekletme yazmaçlarını oku	16 bit erişimi	
04	04	Giriş yazmaçlarını oku		
06	06	Tekli yazmaca yaz		
16	10	Çoklu yazmaca yaz		
22	16	Yazmaca yazmayı maskele		
23	17	Çoklu yazmaçları oku/yaz		
24	18	FIFO sorgusunu oku		
20	14	Dosya kaydını oku	Dosya Kaydı erişimi	
21	15	Dosya kaydını yaz	Arıza Tespitleri	
07	07	İstisnai durumu oku		
08	08	Arıza tespit		
11	0B	Haberleşme olay sayacını al		
12	0C	Haberleşme olay kaydını al		
17	11	Sunucu ID raporla		



Şekil 3. *Modbus TCP mimarisi.*

2.1. Modbus TCP Protokolüne Saldırı Çeşitleri (Attacks Types against Modbus TCP Protocol)

Modbus sistemlerine ve ağlarına yönelik saldırılar bu protokolün özelliklerine, uygulamalarına ve altyapısına göre istismar edilir. Modbus seri protokolüne yapılan saldırılar ana ve köle cihazlarına ve seri haberleşme ağına yönelik gerçekleşirken, Modbus TCP'ye yapılan saldırılar IP ağına, ana ve köle cihazlarına gerçekleştirilir [17].

Bu saldırılarda mesajın içeriğine erişilebilmesinden dolayı taşınan bilginin gizliliğinin ifşa olmasına sebep olunabilir, hizmet engellemesine sebep olduğu için sistem erişilebilirliğine etki eder ve araya girerek ele geçirilen bilginin yeniden üretilebilmesinden dolayı veri bütünlüğü etkilenir.

Modbus seri protokolünde fonksiyon kodu değiştirilerek son sistemin çökmesine sebep olunabilir [13-18]. Aynı zamanda aşağıda belirtilen fonksiyon kodlarında yapılan değişiklerle sisteme zarar verilebilir.

- Fonksiyon kodu 08 ve alt fonksiyon kodu 0A hedef cihaza gönderildiğinde saymacı sıfırlar ve hata yazmaç değerlerini değiştirir. Bu durum saha cihazının yapılandırmasını değiştirir ve hata işlemlerini etkiler. Bu tehdit kategorisi saha cihazını değiştirme sınıfında değerlendirilir.
- Fonksiyon kodu 08, 01'e değiştirildiğinde son cihaz yeniden başlar ve güç açma testi çalışır. Bu mesaj saha cihazının yapılandırma ayarlarını değiştirmesine sebep olur. Bu tehdit, kesme ve değiştirme kategorisinde değerlendirilir.
- Fonksiyon kodu 17, saha cihazına gönderildiğinde saha cihazının durum bilgisi döner. Bu bilgiyle ağ dinlenebilir ve daha farklı saldırıların temeli oluşturulabilir. Bu durum sistemin gizliliğine etki eder.

Modbus TCP protokolüne yönelik yapılan saldırılar şu şekildedir [19]:

- TCP paketinin çerçeveli yapısına etki etmeye yöneliktir. Çoklu Modbus mesajları tek bir TCP paketinin içinde bulunamaz. Bu yüzden mesajlar MTU tarafından parçalarına ayrılır ve RTU'a gönderilir. Bu saldırıda hatalı mesajlar enjekte edilebilir veya mesajlar değiştirilerek hedef sisteme gönderilebilir.
- Son çerçeve biti legal olmayan bir paket TCP bağlantısını kesebilir. Bu tip bir paket gönderilecek Modbus mesajlarını sonlandırabilir ve haberleşmede kesmelere neden olabilir.
- Yüksek önceliğe sahip saha cihazlarına veya ana cihaza sık istekte bulunmak veya diğer bir deyişle bombardıman yapmak hizmet kesintilerine neden olabilir.

Yukarıda bahsedilen saldırılar sonucunda sistemde gizlilik kaybı, erişim kaybı ve veri bütünlüğünün bozulması gibi etkiler görülebilir. Tüm bu etkileri azaltabilmek amacıyla Modbus TCP protokolünün güvenlik seviyesini artırmak için detayları aşağıda anlatıldığı gibi bir deney düzeneği ortamı hazırlanmış, bu ortamda normal Modbus TCP paketleri ve saldırı paketleri analiz edilmiş ve önerilen çalışmayla bu saldırı başarıyla engellenmiştir.

3. ÖNERİLEN ÇALIŞMA (PROPOSED WORK)

Çalışmanın bu bölümünde daha önce detayları verilen Modbus TCP protokolünün kimlik doğrulama ve şifreleme kullanmama zafiyetleri gösterilmiştir. Bu güvenlik açıklarını kullanarak araya girme saldırısı (MITM) ile simülasyon ortamında akan Modbus paketleri açık metin olacak şekilde görüntülenebilmiş ve Metasploit Framework modülü olan Modbusclient kullanılarak aynı simülasyon ortamında yazmaç değerlerinin okunup değiştirilebildiği gözlenmiştir. Saldırı verileri ve normal veriler gönderilirken Wireshark aracı kullanılarak paketler yakalanmış ve bu veriler karşılaştırılarak analiz edilmiştir. Yapılan bu analiz sonucunda saldırıyı engellemeye veya hafifletmeye yönelik Python programlama dili kullanılarak bir program yazılmıştır. Ayrıca programa, kontrol sistemini yöneten operatör tarafından kasıtlı veya yanlışlıkla yazmaç değerlerine belirli bir değer üzerinde değer girilmesini engelleyecek bir kontrol fonksiyonu eklenmiştir. Böylece Modbus TCP protokolü ile haberleşen bir kontrol sistemine yönelik iç veya dış ağdan yetkisiz bir şekilde veri girilmesi ve siber güvenlik camiasında en fazla risk teşkil ettiği düşünülen tehditlerden biri olan ve literatürde "intruder" olarak isimlendirilen küskün veya kötü niyetli çalışanın belirlenen bir değerden fazla değer girmesinin engellenmesi veya hafifletilmesi amaçlanmıştır.

3.1. Deney Düzeneği Ortamı (Testbed Environment)

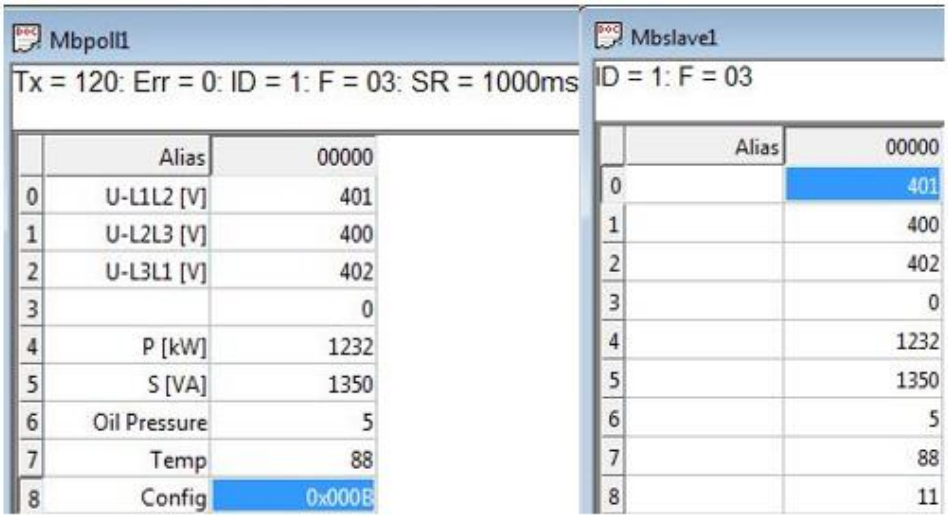
Modbus TCP protokolünün güvenliğini sağlamaya yönelik yapılan çalışmada Modbus TCP paketlerini analiz etmek amacıyla Modbus Poll [20] isimli bir program kullanılarak simülasyon ortamı hazırlanmıştır. Saldırıları gerçekleştirmek amacıyla Kali Linux 2.0 işletim sistemi kullanılmış ve ara katman olarak paketlerin kontrol edileceği ve üzerinde Python programının olduğu Ubuntu işletim sistemi tercih edilmiştir.

Modbus TCP protokolüne yönelik paketlerin analizi için Modbus Poll ve Modbus Slave simülasyon ortamı kurulmuştur. Modbus Poll, Modbus protokolünü simule ve test etmek için Modbus Slave veya diğer cihazları geliştiren kişilere yardımcı olması amacıyla tasarlanmış Modbus Master simülatörüdür. Çoklu ara yüzle birçok Modbus Slave veya veri alanı aynı anda görüntülenebilmektedir. Her bir ekranda Modbus Slave ID, fonksiyon ve adres özel olarak belirlenebilmekte ve yazmaçlara okuma ve yazma işlemleri gerçekleştirilebilmektedir.

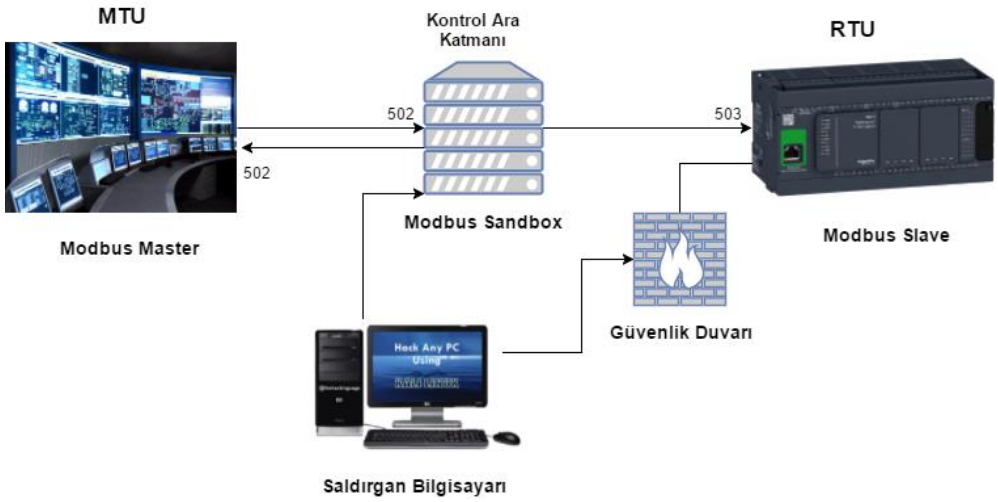
Tasarlanan deney düzeneğinde Modbus Master programı Windows 7 işletim sistemi üzerinde koşmakta olup bu program daha önce detayları verilmiş olan MTU olarak çalışmaktadır. Modbus Slave programı ise Windows XP işletim sistemi üzerinde koşmakta olup RTU olarak çalışmaktadır. Modbus Master ve Modbus Slave programlarının koştuğu bu işletim sistemleri aynı ağda olup birbirleriyle haberleşebilmektedirler. Böylece Modbus Master üzerindeki yazmaç alanlarına girilen değerler Modbus Slave üzerindeki yazmaçlara gönderilip yazılabilmekte ve okunabilmektedir.

Şekil 4'te çalışmada kullanılan Modbus Master ve Modbus Slave simülatörlerine ait ekran görüntüleri verilmiştir. Modbus Master'da yazmaçlara girilen değerler, aralarındaki Modbus TCP protokolü bağlantısı sayesinde Modbus Slave'deki yazmaçlara gönderilerek yazılmaktadır.

Tasarlanan deney düzeneğinin topolojisi Şekil 5'te verilmiştir. Topolojiye göre MTU olarak Windows 7 makinası üzerinde yüklü olan Modbus Master çalışmaktadır. Modbus paketlerini varsayılan 502 portu üzerinden göndermektedir. Varsayım olarak bir üretim tesisinin veya nükleer bir santralin kontrol sisteminin yönetildiği merkezi sunucu olarak düşünülebilir. RTU olarak Windows XP üzerinde yüklü olan Modbus Slave çalışmaktadır. Bu makinada 503 portu dinlemeye alınmış olup Modbus haberleşmesi bu port üzerinden sağlanmaktadır. Varsayım olarak bir üretim tesisinin sıcaklık değerlerini veya nükleer bir santralin santrifüj dönme frekans değerlerini merkezi sunucuya göndermekle görevli bir PLC cihazı olarak düşünülebilir. Kontrol ara katmanı olarak, belirtilen iki cihaz arasındaki haberleşme sırasında gelip giden paketlerin kontrol edildiği, üzerinde Modbus TCP protokol güvenliği sağlanması amacıyla yazılmış olan Python kodunun çalıştığı Ubuntu makinası bulunmaktadır. Geliştirilen algoritmaya göre makina 503. portunu dinlemeye almış olup MTU üzerinden girilen değerler ilk olarak kontrol ara katmanına gitmekte ve burada paketlerin kontrolünün ardından PLC cihazına gönderilip cihaz üzerindeki yazmaçlara yazılmaktadır.



Şekil 4. *Modbus Master ve Modbus Slave.*

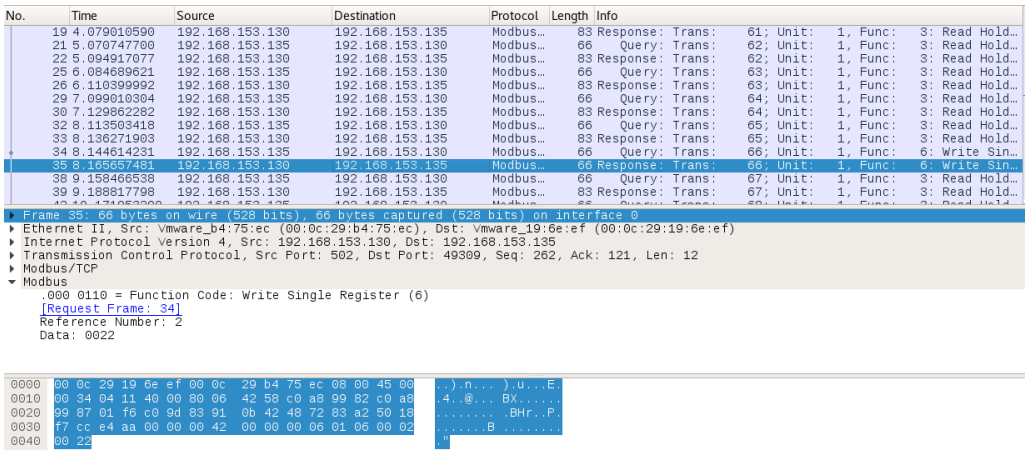


Şekil 5. *Deney düzeneği topolojisi.*

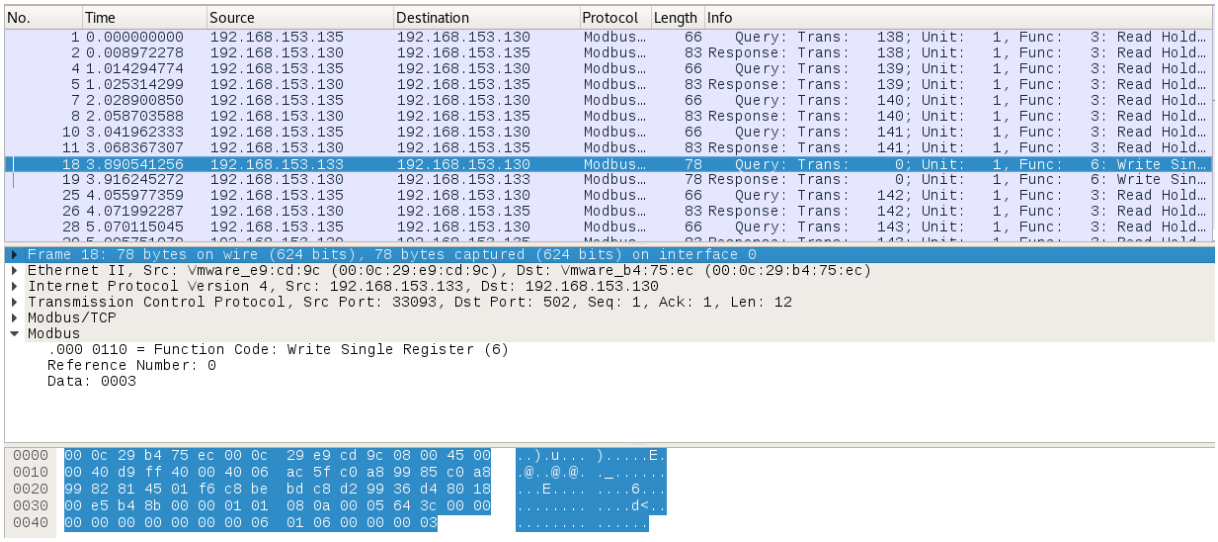
3.2. Paketlerin Analizi (Analysis of the Packets)

Modbus Master, Modbus Slave ve saldırgan makinaları arasında akan Modbus TCP paketleri Wireshark aracı kullanılarak yakalanıp analiz edilmiştir. İlk etapta Modbus Master ve Modbus Slave makinaları arasında akan normal Modbus TCP paketleri yakalanmış ve daha sonrasında saldırgan makinasıyla Metasploit Framework Modbusclient modülü kullanılarak manipüle edilmiş Modbus TCP paketleri yakalanmıştır. Böylece normal Modbus TCP paketleriyle manipüle edilmiş paketler karşılaştırılarak analiz edilmiştir. Şekil 6'da Modbus Master ve Modbus Slave makinaları arasında akan paketlerin "Write_Register" fonksiyon koduyla Wireshark üzerinden yakalanması gösterilmiştir. Şekil 7'de ise saldırgan makinasıyla manipüle edilmiş Modbus TCP paketinin gönderildiği Wireshark çıktısı gösterilmektedir.

Yakalanan paketlerin karşılaştırılması sonucunda 192.168.153.135 IP adresli Master cihazından 192.168.153.130 IP adresli Slave cihazına Modbus TCP paketleri gönderilirken manipüle edilen paketlerin 192.168.153.133 IP adresli saldırgan makinesiyle gönderildiği gözlenmiştir. Paketlerin diğer parametrelerinde hiçbir değişiklik olmadığı, sadece kaynak IP adreslerinin değiştirilerek gönderildiği anlaşılmaktadır. Slave cihazın kendisine gelen paketlerde kaynak IP adresini kontrol etmediği fark edilmiştir. Yapılan bu analiz sonucunda iki cihaz arasında gerçekleşen haberleşme esnasında Modbus Slave cihazına gelen paketlerde, kaynak IP adresinin kontrolüne yönelik içerisinde bir Python kodunun çalıştığı bir ara katman yerleştirilmesi amaçlanmıştır. Böylece Modbus Slave cihazı sadece Master cihazından gelen paketleri kabul edip kendi yazmaçlarına işleyecek, başka bir IP adresinden gelecek Modbus TCP paketlerini ara katmandaki kontrol sonrasında Master cihazına göndermeyecektir.



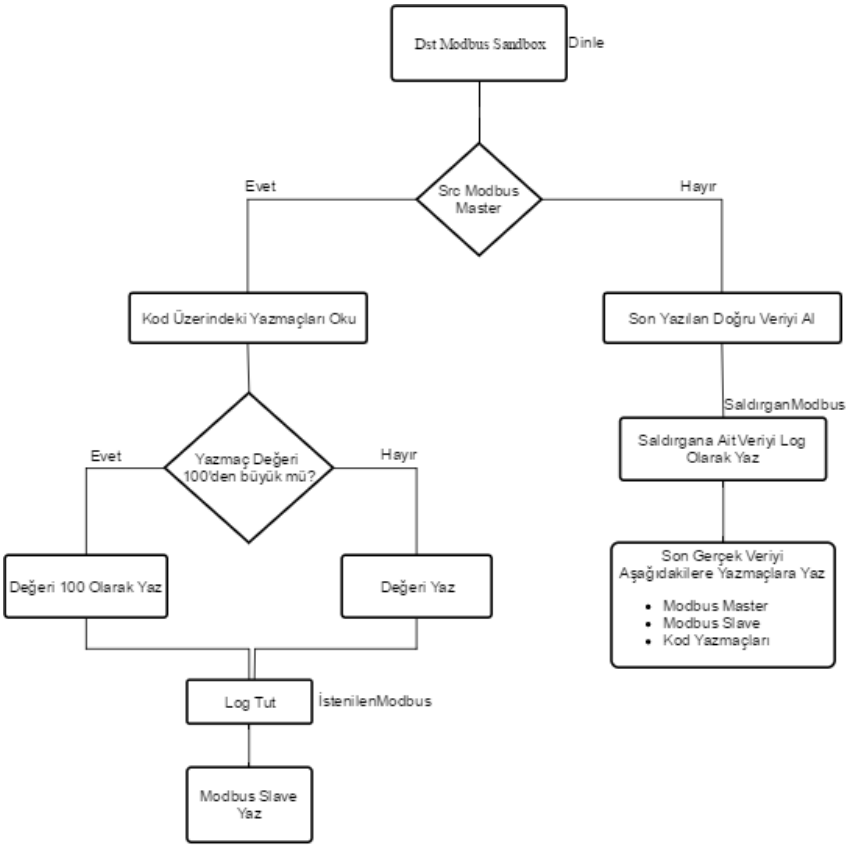
Şekil 6. *Normal Modbus TCP paketi.*



Şekil 7. *Manipüle edilmiş Modbus TCP paketi.*

3.3. Kontrol Ara Katmanı (Control Interlayer)

Modbus Master ve Modbus Slave cihazları arasına yerleştirilen ve iki cihaz arasında akan paketlerin denetimi için bir kontrol ara katmanı kullanılmıştır. Bu kontrol ara katmanı, üzerinde Python bir kod koşan Ubuntu işletim sistemine sahip bir makina olup ağ üzerinde akan Modbus TCP paketlerini kendi üzerine alır ve üzerinde çalışan Python koduyla denetim sağladığı için Sandbox olarak da isimlendirilebilir. Python kodunun akış diyagramı Şekil 8'de verilmiştir.



Şekil 8. *Python kodunun akış diyagramı.*

Şekil 8'de verilen akış diyagramına göre çalışma kapsamında hazırlanan kod ilk başta kendi 502. portunu dinlemeye alır ve kendine gelen Modbus paketlerinin kaynak IP adreslerini alıp Modbus Master IP adresiyle karşılaştırır. IP adresleri örtüşüyorsa kod üzerindeki yazmaç değerlerini okur. Bu değerler 100'den büyükse yazmaç değeri olarak 100 yazar, 100'den küçükse değeri okuduğu gibi yazar. Burada belirlenen 100 değeri, simülasyon ortamında seçilen varsayılan bir değerdir. Bir sistem odasında ortamın sıcaklık değeri veya bir nükleer santralde santrifüj dönme frekans değeri olarak düşünülebilir. Daha sonra "İstenilenModbus" log dosyasına yazmaçlardaki değerleri kaydedip Modbus Slave cihazındaki yazmaçlara değerler yazılır. IP adresleri örtüşmüyorsa kod bunu saldırı olarak algılayacak ve en son yazdığı doğru veriyi alacaktır. Saldırgana ait veriyi "SaldırganModbus" isimli log dosyasına kaydeder. Daha sonrasında son gerçek veriyi Modbus Master, Modbus Slave ve kod üzerindeki yazmaçlara yazar.

Tasarlanan deney düzeneğiyle saldırgan Modbus Slave cihazının yazmaçlarına Metasploit Framework Modbusclient modülü kullanarak yetkisiz bir şekilde veri yazmaya çalıştığında Modbus Slave cihazı önündeki IP ve port tabanlı çalışan güvenlik duvarına takılacaktır. Saldırgan aynı şekilde Modbus Sandbox yazmaçlarına veri girmeye çalıştığında Python kodunun kontrol mekanizmasına takılacak ve yetkisiz bir şekilde veri giremeyecektir. Aynı zamanda saldırganın yaptığı işlemler Sandbox tarafında loglanacaktır. Bununla beraber Modbus Master tarafında legal olarak 100 üzerinde bir değer basılmaya çalışıldığında yazmaçlara en fazla 100 değeri girilebilecektir. Bu sınırlamayla birlikte bir üretim tesisindeki veya bir elektrik dağıtım bölgesindeki kontrol sistemlerinde verilerin belirlenen bir değer üzerinde girilmesi olanaksız hale getirilmiştir.

4. SALDIRI ANALİZİ (ATTACK ANALYSIS)

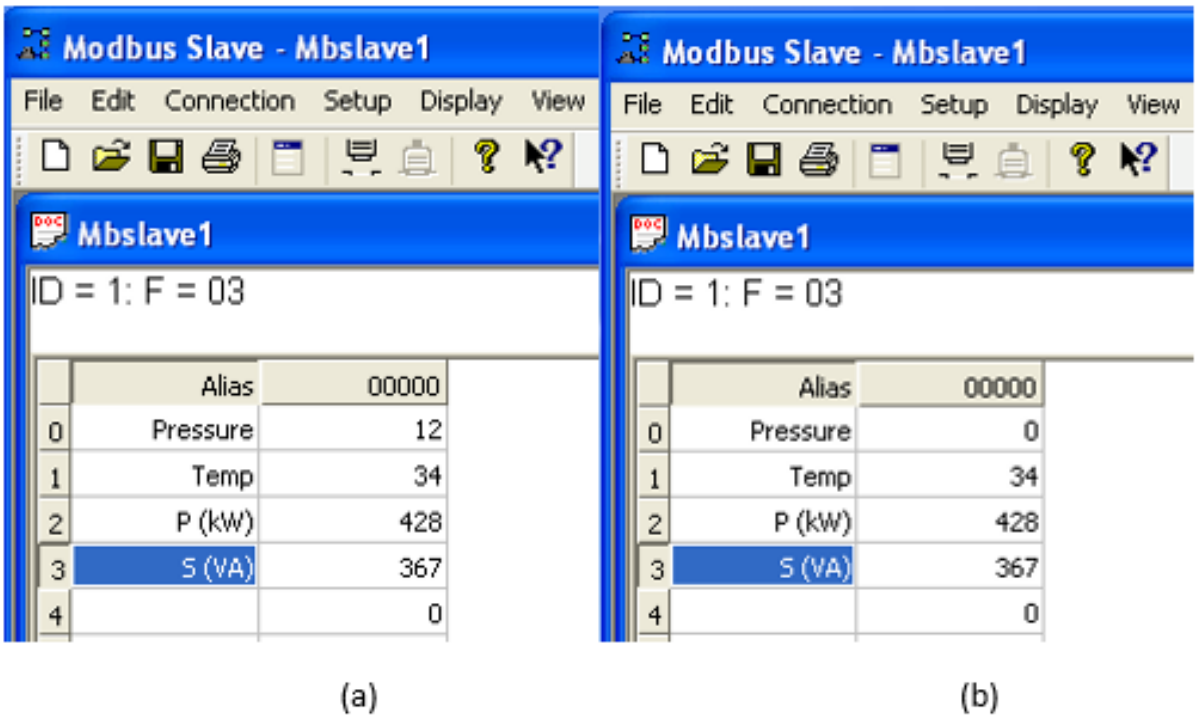
Şekil 4'te verilen deney düzeneğinde Kali Linux işletim sistemine sahip saldırgan bilgisayarında bulunan sızma testi araçları kullanılarak sunucu ve port taraması işlemi ve bu işlemler sonucu elde edilen bilgilerle Modbus paketlerindeki veri bölümleri manipüle edilip değiştirilmeye çalışılmıştır. Bu bölümün ilerleyen kısımlarında ekran görüntüleri ile birlikte verilecek olan bu tarama ve veri manipülasyon işlemleri, ilk etapta çalışmada önerilen savunma mekanizması olmadan gerçekleştirilerek gösterilmiş, daha sonrasında savunma mekanizması yani kontrol ara katmanı eklenerek gerçekleştirilmiş ve sonuçlar analiz edilmiştir. Böylece savunma mekanizması olarak tasarlanan kontrol ara katmanı olan Modbus Sandbox'ın zafiyet barındırıp barındırmadığı incelenmiş olacaktır.

Öncelikle kontrol ara katmanı eklenmeden sadece Modbus Master ve Modbus Slave cihazlarının hiçbir güvenlik önlemi olmadan haberleştiği ortamda saldırı analizi gerçekleştirilmiştir. Bunun için ilk etapta nmap tarama aracı kullanılarak iç ağda bulunan ve Modbus TCP protokolünün varsayılan port olarak çalıştığı 502. port üzerinde modbus servisinin çalıştığı cihazlar tespit edilmiştir. Böylece hedef alınan cihazlar belirlenmiştir. Bu aşamada Modbus Slave cihazının üzerinde 502. portun açık olduğu ve üzerinde modbus servisinin çalıştığı gözlenmiştir. Şekil 9'da ağ taramasına ait sonuçlar gösterilmiştir.

Tarama işlemi sonrasında daha önce detayları verilmiş olan Metasploit Framework modbusclient modülü kullanılarak üzerinde modbus servisinin çalıştığı tespit edilen cihaz üzerindeki yazmaç değerleri manipüle edilmeye çalışılmıştır. Bunun için yazmaç adresi 0 olan adrese, yetkisiz bir şekilde 0 değeri girilmiştir. Manipüle işlemi Şekil 10'da, bu işlem sonucunda Modbus Slave cihazının yazmaçlarındaki değer değişimleri, saldırı öncesindeki değerler ve saldırı sonrasındaki değerler olacak şekilde Şekil 11'de verilmiştir.

Şekil 9. *Savunmasız sistemde Nmap taraması.*

Şekil 10. *Savunmasız sistemde manipüle işlemi.*



Şekil 11. *Modbus Slave yazmaç değerleri (a) Saldırı öncesi (b) Saldırı sonrası.*

Şekil 10'da açıkça görüldüğü gibi, saldırı sonucunda istismar modülünün içinde yazmaç 0 adresine başarılı bir şekilde 0 değeri girilmiştir. Böylece savunmasız sistemde yani Modbus Master ve Modbus Slave cihazları arasındaki haberleşmede herhangi bir kontrol ara katmanının kullanılmadığında gerçekleştirilen siber saldırının başarılı bir şekilde sonuçlandığı gözlemlenmiştir.

İkinci aşamada ise Modbus Master ve Modbus Client cihazları arasında savunma mekanizması olarak Modbus Sandbox'ın kullanıldığı Şekil 4'te gösterilen topolojide siber saldırı gerçekleştirilmiştir. Saldırının bu bölümünde de saldırı aracı olarak tekrar ağ taraması için Nmap aracı ve veri manipülasyonu için modbusclient modülü kullanılmıştır. Fakat ağ topolojisine göre Modbus Slave bu sefer varsayılan Modbus portu yani 502. port üzerinden değil 503. port üzerinden haberleşmektedir. Modbus Sandbox ise bu sefer varsayılan Modbus haberleşme portu yani 502. port üzerinden haberleşmektedir. Yani bir bakıma saldırgan ağ üzerinde direkt olarak Modbus varsayılan portu açık olan sunucuları tararsa Modbus Slave

cihazlarını tespit edemeyecektir. Bu da bir bakıma saldırganın Modbus cihazını varsayılan saldırı vektörleriyle tespit edememesinden dolayı güvenlik önlemi olarak sayılabilir.

Çalışmada Nmap taramasıyla ağ üzerinde hem 502 hem de 503. portu açık olan cihazlar tespit edilmiş ve bu cihazlar üzerinde modbusclient kullanılarak manipülasyon işlemi gerçekleştirilerek Modbus Slave ve Modbus Sandbox makinalarındaki yazmaç değerlerinin değişimleri ve saldırının başarılı sonuçlanıp sonuçlanmadığı gözlenmiştir. Şekil 12'de Nmap tarama sonuçları ve Şekil 13'te manipülasyon işlemine ait ekran görüntüleri mevcuttur. Tarama sonucuna göre, yukarda da bahsedildiği gibi Modbus Slave cihazında 503. portun Modbus Sandox cihazında 502. portun açık olduğu tespit edilmiştir.

Şekil 13'e göre modbusclient modülü kullanılarak Modbus Slave cihazının yazmaç 0 adresine 0 değeri yetkisiz bir şekilde girilmeye çalışılmış, fakat topolojiye göre IP ve Port tabanlı çalışan güvenlik duvarı bu saldırıyı engellemiş ve Modbus Slave cihazındaki yazmaçlara herhangi bir müdahalede bulunulamamıştır.

Nmap tarama sonuçlarına göre 502. portunun açık olduğu tespit edilen Modbus Sandbox cihazına yönelik manipülasyon işleminde cihaz üzerinde çalışan ve detayları yukarıdaki bölümlerde verilen Python kodu bu saldırıyı engelleyerek saldırgana ait logları "SaldırganModbus" isminde bir dosyaya kaydeder. Aynı zamanda kod içerisinde bulunan kontrol fonksiyonuyla Modbus Master cihazındaki yazmaç adreslerine 100 üzerinde bir değer girildiğinde bu işlem engellenecek ve Modbus Slave cihazındaki yazmaçta ilgili adrese en fazla 100 olarak yazacaktır. Şekil 14'te Modbus Sandbox cihazına yönelik manipülasyon işlemine ait ekran görüntüsü verilmiştir.

```
root@kali:~# nmap -sS -p 502,503 192.168.153.0/24 --open

Starting Nmap 7.40 ( https://nmap.org ) at 2017-06-11 18:16 +03
Nmap scan report for 192.168.153.137
Host is up (0.00072s latency).
Not shown: 1 closed port
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT    STATE SERVICE
502/tcp open  mbap
MAC Address: 00:0C:29:94:57:FA (VMware)

Nmap scan report for 192.168.153.138
Host is up (0.00083s latency).
Not shown: 1 closed port
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT    STATE SERVICE
503/tcp open  intrinsa
MAC Address: 00:0C:29:71:8A:DB (VMware)
```

Şekil 12. *Savunmalı sistemde Nmap taraması.*

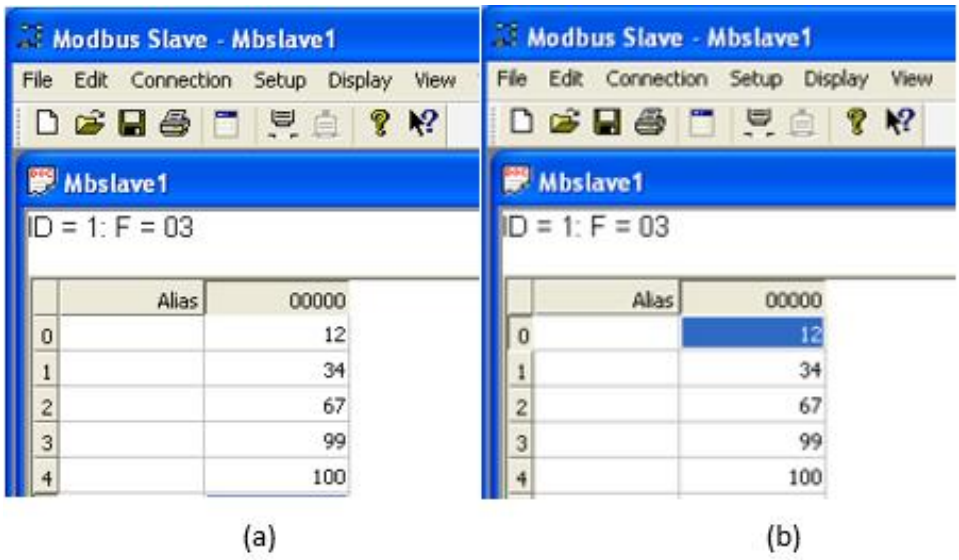
```
msf auxiliary(modbusclient) > show options

Module options (auxiliary/scanner/scada/modbusclient):

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   DATA           0                no        Data to write (WRITE_COIL and WRITE_REGISTER modes only)
   DATA_ADDRESS   0                yes       Modbus data address
   DATA_COILS                      no        Data in binary to write (WRITE_COILS mode only) e.g. 0110
   DATA_REGISTERS                  no        Words to write to each register separated with a comma (WRITE_REGISTERS mode only)
e.g. 1,2,3,4
   NUMBER         1                no        Number of coils/registers to read (READ_COILS ans READ_REGISTERS modes only)
   RHOST          192.168.153.130  yes       The target address
   RPORT          503              yes       The target port (TCP)
   UNIT_NUMBER    1                no        Modbus unit number


Auxiliary action:

   Name           Description
   ----           -----------
   WRITE_REGISTER Write one word to a register


msf auxiliary(modbusclient) > run

[-] 192.168.153.130:503 - Auxiliary failed: Rex::ConnectionTimeout The connection timed out (192.168.153.130:503).
[-] 192.168.153.130:503 - Call stack:
[-] 192.168.153.130:503 -   /usr/share/metasploit-framework/vendor/bundle/ruby/2.3.0/gems/rex-socket-0.1.6/lib/rex/socket/comm/lo
cal.rb:291:in `rescue in create_by_type'
```

Şekil 13. *Savunmalı sistemde manipülasyon işlemi.*

Şekil 14. *Savunmalı sistemde Modbus Sandbox cihazına yönelik manipülasyon işlemi.*

Bu siber saldırı işlemi sonrasında saldırı opsiyonlarında da belirtildiği gibi 0. yazmaç adresine 0 değeri yetkisiz bir şekilde girilmeye çalışılmış ve saldırı kodu çalıştırıldığında kodun başarılı bir şekilde çalışıp ilgili yazmaca değerin girildiği belirtilmektedir. Fakat Modbus Sandbox ve Modbus Slave cihazlarının yazmaçları incelendiğinde saldırının başarısız olduğu ve yazmaç değerlerinin değişmediği gözlemlenmiştir. Daha önce belirtildiği gibi Şekil 14'te savunmalı sistemde saldırının başarılı bir şekilde gerçekleştirildiği ilgili yazmaca saldırganın belirlediği değerin girildiği gösterilmiştir, fakat Şekil 15'te de görüldüğü gibi saldırı sonrasında yazmaçlarda bir değişiklik olmamıştır. Şekil 16'da Modbus Sandbox cihazındaki yazmaç değerlerinin aktığı komut satırı çıktıları gösterilmektedir.

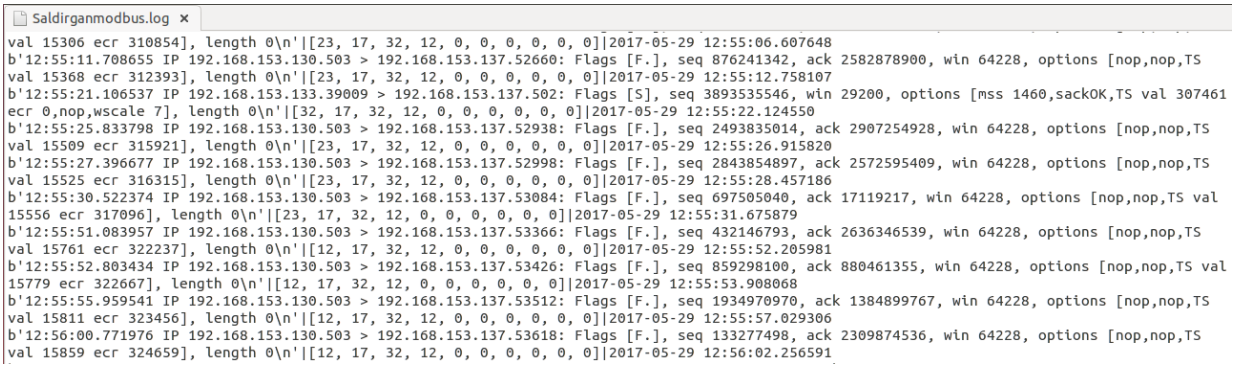
Tüm bunlarla birlikte Modbus Sandbox cihazına yönelik siber saldırı sonucu oluşan tüm logların kaydedildiği "SaldırganMobdus" isimli kayıt dosyasına ait bir ekran görüntüsü de Şekil 17'de gösterilmiştir.



Şekil 15. *Savunmalı sistemde Modbus Slave yazmaçları (a) saldırı öncesi (b) saldırı sonrası.*

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
1 packet captured
4 packets received by filter
0 packets dropped by kernel
b'10:22:44.297291 IP 192.168.153.135.49174 > 192.168.153.137.502: Flags [P.], se
q 3606355813:3606355825, ack 1696433380, win 256, length 12\n'
[12, 34, 67, 99, 100, 0, 0, 0, 0, 0]
yazdi
```

Şekil 16. *Modbus Sandox cihazı yazmaç değerleri.*



Şekil 17. *SaldırganModbus log dosyası.*

5. SONUÇ (CONCLUSION)

Bu çalışmada, elektrik üretim, iletim ve dağıtım tesisleri, iletişim ve haberleşme, ulaşım, gaz üretim gibi kritik altyapı olarak nitelendirilebilecek farklı tesislerde denetim amaçlı kullanılan SCADA sistemleri ve bu sistemlerin güvenliğini artırmaya yönelik bir yazılım geliştirilmiştir. SCADA sistemlerinin haberleşmesinde en sık kullanılan haberleşme protokollerinden Modbus TCP protokolünün kimlik doğrulama zafiyeti ve veri iletimi sırasında haberleşme olmaması zafiyeti incelenmiş, bu zafiyetler istismar edilerek veriler manipüle edilmiştir. Gerçekleştirilen saldırıda bir Linux işletim sistemiyle önemli işletmelerin otomasyon sistemlerini kontrol edebilecek bir PLC cihazının üzerindeki yazmaç adreslerindeki değerlerin değiştirilebileceği ve basit bir sistemle karmaşık sistemlere zarar verilebileceği gösterilmiştir. Bu sebeple endüstriyel otomasyon sistemlerine yönelik güvenlik çözümleri üretilirken basit bir saldırı senaryosundan karmaşık bir saldırı senaryosuna kadar çok aşamalı saldırı tekniklerinin dikkate alınması gerektiği ortaya konulmuştur.

Modbus TCP protokolünde tespit edilen güvenlik zafiyeti sonrasında, bu açığa yönelik saldırıları hafifletmek veya engellemek için ara kontrol katmanı üzerinde bir Python kodu geliştirilmiş ve bir kontrol mekanizması oluşturulmuştur. Geliştirilen bu güvenlik kontrol mekanizması ile EKS'de en sık kullanılan haberleşme protokollerinden birisi olan Modbus TCP protokolüne yönelik dışarıdan yetkisiz bir kullanıcı tarafından müdahalede bulunulması engellenmiş ve aynı zamanda kod içerisindeki kontrol fonksiyonuyla EKS ağı içerisinde kötü niyetli bir kullanıcı tarafından veya yetkili personel tarafından yazmaçlara girilen değerler denetim altına alınmıştır. Sunulan çalışmanın gerçek sistemlerle entegre çalıştırılması hem iç ağdan hem de dış ağdan gelebilecek siber saldırıların etkilerini hafifletecektir.

Genel bir sonuç olarak, fonksiyonel işlemlerini kaybetmesi, zarar görmesi veya veri iletiminde oluşan manipülasyonlar sonucunda toplum düzenini, insan hayatını, ekonomik kayıpları, ulusal veya global düzeyde güvenliği sekteye uğratabilecek kritik altyapı sistemlerinin güvenlik bilinciyle oluşturulması ve buna uygun tasarlanması gerektiği şüphesizdir. Özellikle ülkemizde elektrik üretim, iletim ve dağıtımına ilişkin kontrol sistemlerinin haberleşme protokollerinin yeniden gözden geçirilmesi ve siber güvenlik açısından ele alınması gerekmektedir. Kritik altyapılar içerisinde en önemli sistemlerden birisi olan SCADA sistemlerinin güvenliği hayati derecede öneme sahiptir. Bu nedenle, sunulan çalışmanın kritik bir altyapı olan SCADA sistemlerinin güvenliğine katkı sağlayacağı değerlendirilmektedir.

KAYNAKLAR (REFERENCES)

[1] M. Unver, C. Canbay, Ulusal ve uluslararası boyutlarıyla siber güvenlik. EMO elektrik Mühendisliği Dergisi, 438 (2010) 94-103.

[2] R. Sanz, K. Årzén, Trends in software and control. IEEE Control System Magazine, 23:3 (2003) 12-15.

[3] R. Chandia, J. Gonzalez, T. Kilpatrick, M. Papa, Security strategies for scada networks. Critical Infrastructure Protection, 253 (2007) 117–131.

[4] L. Yanfei, W. Cheng, Y. Chengbo, Q. Xiaojun, Research on zigbee wireless sensors network based on modbus protocol. Proceedings of 2009 international forum on information technology and applications, 1 (2009) 487–490.

[5] L. Yanfei and W. Cheng, An improved design of Zigbee wireless sensor network. 2nd IEEE international conference on computer science and information technology, (2009) 515–518.

[6] R. Bayindir, Ş. Sağıroğlu, A. Özbilen, İ. Çolak, Investigating industrial risks based on information security for observable electrical energy distribution system and suggestions. Gazi University Journal of Faculty of Engineering and Architecture, 24: 4 (2009) 715–723.

[7] Q. Xiong et al., A vulnerability detecting method for Modbus-tcp based on smart fuzzing mechanism. IEEE international conference on electro information technology, (2015) 404–409.

[8] S. Bhatia, N. Kush, C. Djamaludin, J. Akande, and E. Foo, Practical Modbus flooding attack and detection. Conferences in research and practice in information technology series, (2014) 57–65.

[9] W. L. Shang, L. Li, M. Wan, P. Zeng, Security defense model of Modbus tcp communication based on zone/border rules : misuse. International conference on network security and communication engineering (2014).

[10] B. Chen, N. Pattanaik, A. Goulart, K. L. Butler-Purry, D. Kundur, Implementing attacks for Modbus/tcp protocol in a real-time cyber physical system testbed. IEEE international workshop technical committee on communications quality and reliability, (2015).

[11] G. Dondossola, G. Garrone, J. Szanto, G. Deconinck, T. Loix, H. Beitollahi, ICT resilience of power control systems: experimental results from the crutial testbeds, Proceedings of the international conference on dependable systems and networks, (2009) 554–559.

[12] G. Dondossola, G. Deconinck, F. Garrone, H. Beitollahi, Testbeds for assessing critical scenarios in power control systems. 5508 (2009) 223–234.

[13] M. Mallouhi, Y. Al-Nashif, D. Cox, T. Chadaga, S. Hariri. A testbed for analyzing security of scada control systems. IEEE PES innovative smart grid technologies conference, (2011) 1–7.

[14] Modbus IDA, MODBUS application protocol, (2006) 1–51.

[15] B. Dutertre, Formal modeling and analysis of the Modbus protocol. Critical Infrastructure Protection, (2007) 189–204.

[16] A. Swales, Open Modbus/tcp specification, Schneider electric, (1999) 1–26.

[17] P. Huitsing, R. Chandia, M. Papa, S. Shenoi, Attack taxonomies for the Modbus protocols. International Journal of Critical Infrastructure Protection, (2008) 37–44.

[18] Modbus over serial line–specification and implementation guide. (2002).

[19] T. H. Morris, "On cyber attacks and signature based intrusion detection for Modbus based industrial control. Journal of Digital Forensics, Security and Law, 9: 1 (2009) 37–56.

[20] Internet: http://www.modbustools.com/download.html, Modbus poll simulator. Erişim Tarihi: 25-May-2017.