GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Implementing and Auditing CIS Controls (Security 566)"
at http://www.giac.org/registration/gccc

Man-In-The-Middle Attack Against Modbus TCP Illustrated with Wireshark

GIAC (GCCC) Gold Certification

Author: Gabriel Sanchez, gmgsanchez@gmail.com
Advisor: Stephen Northcutt
Accepted: October 2017

Abstract

Though attacks on the industrial control system (ICS) and their protocols are not a new occurrence, recent years have highlighted a growing trend in such attacks. To make matters worse, cyber defenders have also dealt with a slow migration to more secure ICS protocols due to costs associated with equipment downtime. With the increase in attacks and the slow migration to more secure ICS protocols, it is crucial for cyber defenders to be able to quickly set up labs to mimic and observe how potential attacks on the ICS network function so that necessary defenses and detection mechanisms can be put in place. This paper lays out how to setup a lab with multiple virtual machines and ICS software that can observe a Master workstation controlling a PLC. First, Wireshark will be used to illustrate and compare normal Modbus TCP communications between the Master and PLC workstations. Wireshark will then be used to demonstrate and compare a MITM attack with an Ettercap filter that manipulates the Modbus TCP communications against both workstations.

© 2017 The SANS Institute, Author Retains Full Rights

© 2017 The SANS Institute Author retains full rights.

1. Introduction

Though attacks on the industrial control system (ICS) and their protocols are not a new occurrence, the technology industry has experienced a significant increase in the frequency of such attacks towards ICS networks. In fact, the IBM Managed Security Services has reported that the number of attacks aimed at ICS has increased by 110% between 2015 and 2016 (Kovacs, 2016). For the cyber defenders that are tasked with both defending and detecting such attacks, the ability to quickly set up a lab and understand the inner workings of an attack will greatly assist with setting up the proper defenses. This research will specifically show how a lab running a common ICS protocol of Modbus TCP can be set up to observe normal and malicious communications between a Master and PLC workstation. Wireshark will be utilized to demonstrate normal Modbus TCP communications which will then be compared with a man-in-the-middle (MITM) attack, combined with the proper Ettercap filter, to modify the Modbus TCP commands between both workstations. The Ettercap filter modifies any Modbus TCP command with a hex value of ff00 (which turns a PLC coil on) to a hex value 0000 (which turns a PLC coil off). By changing the coil to off the attacker maliciously modifies the Master workstation's attempt to turn on the PLC workstation's coil and, instead, turns the coil off without any errors occurring. Since ICS networks are typically considered critical systems, the speed at which updates and infrastructure changes can be implemented is very slow. Cyber defenders also must deal with losses due to equipment downtime associated with updating the software which can be significant even in comparison with the risks posed by vulnerabilities (Kaspersky Lab, 2016). With the increase of attacks and the slow migration to more secure ICS protocols, it is crucial for cyber defenders to be able to quickly set up labs to mimic and observe how potential attacks, such as a MITM with an Ettercap filter, on the ICS network function so that necessary defenses and detection mechanisms can be implemented.

2. Modbus

Modbus is a serial communications protocol developed by Modicon in 1979 for use with its programmable logic controllers (PLCs) which are employed as part of the

Gabriel Sanchez, gmgsanchez@gmail.com

© 2017 The SANS Institute Author retains full rights.

company's factory production, traffic lights, elevators, conveyor belts, substation automation, etc. (Schneider Electric, 2017). Despite its age, Modbus is still one of the most commonly used protocols for field device communications (UNSERVER, 2016). Therefore, it is no surprise that attackers would target this protocol when attempting to manipulate an ICS environment. Versions of the Modbus protocol exist for serial lines (Modbus RTU and Modbus ASCII) and for Ethernet (Modbus TCP) (Schneider Electric, 2017). Ethernet has become the de facto standard of both corporate enterprise systems and factory networking (Modbus, 2017). The Modbus TCP protocol includes the Ethernet header, IP header, TCP header, Modbus TCP, and data. These headers are shown in Figure 1.

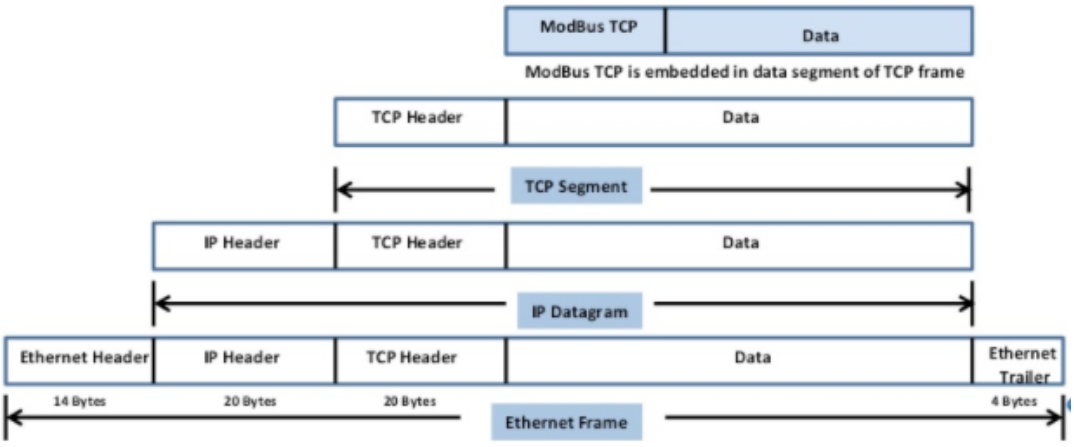


Figure 1 (Electromate, 2014) – Modbus TCP

Ethernet allows the TCP/IP wrapper to introduce the Modbus TCP protocol, historically done via serial lines, to unwrap the layers until it reaches the Modbus TCP header. The Modbus TCP header and its data include the Modbus Application Header (MBAP) and the Protocol Data Unit (PDU) as seen in Figure 2.

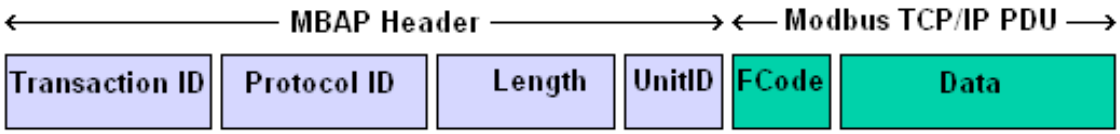


Figure 2 (Simply Modbus, 2017) – MBAP Header and PDU

Gabriel Sanchez, gmgsanchez@gmail.com

© 2017 The SANS Institute Author retains full rights.

Each byte has a specific function and is translated appropriately by the receiving device. The MBAP Header and the PDU have the following functions:

- **Transaction ID:** Two bytes set by the Client to uniquely identify each request.
- **Protocol ID:** Two bytes set by the Client.
- **Length:** Two bytes identifying the number of bytes in the message to follow.
- **Unit Identifier:** One byte set by the Client and echoed by the Server for identification of a remote slave connected.
- **FCode:** The Function Code (Read or Write Registers/Coils)
- **Data:** The Data Address of the coil/register and the status to write/read (Simply Modbus, 2017).

For additional information, a technical reference guide for Modbus TCP has been written by Acromag Incorporated (Acromag Incorporated, 2005).

3. Modbus TCP Lab Setup

The lab consists of a Master workstation, PLC workstation, and a Simulated Attacker workstation which are all loaded with VMware Fusion software (VMware Workstation software or separate physical workstations are suitable alternatives). The Master workstation can read and write to the receiving PLC workstation. The PLC workstation was simulated with a programmable logic control (PLC) that waited to receive commands from the Master workstation. The simulated attacker was a Kali Linux workstation installed with Wireshark and Ettercap software. The software and configuration below was installed on each workstation:

1) Master Workstation:
    - Windows 7 Professional Operating System
    - IP address on same subnet (192.168.238.134)
    - Modbus Master 0.4.8 software setup.
        i. Unit ID 8
        ii. Function Code “Write Single Coil (0x05)”
        iii. Start Address of 0

© 2017 The SANS Institute Author retains full rights.

iv. Decimal value of 1

v. Modbus TCP Settings: PLC=192.168.238.129 TCP_Port 502

This setup is shown in Figure 3.

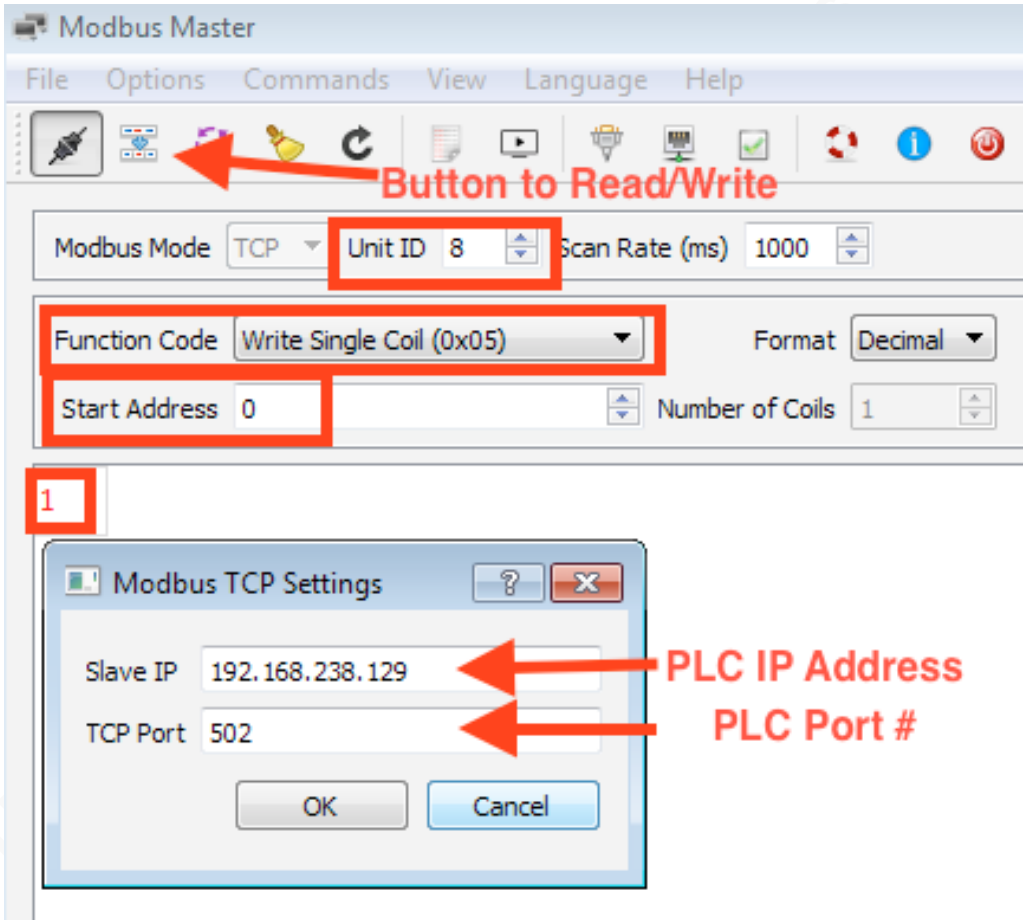


Figure 3 – Modbus Master Configuration

2) PLC Workstation:

- Linux Debian 3.6 Operating System
- IP address on same subnet (192.168.238.129)
- ModbusPal 1.6 software setup.
  - i. TCP Port 502
  - ii. Slave ID 8
  - iii. Coils Addresses 1-10.

This setup is shown in Figure 4.

© 2017 The SANS Institute Author retains full rights.

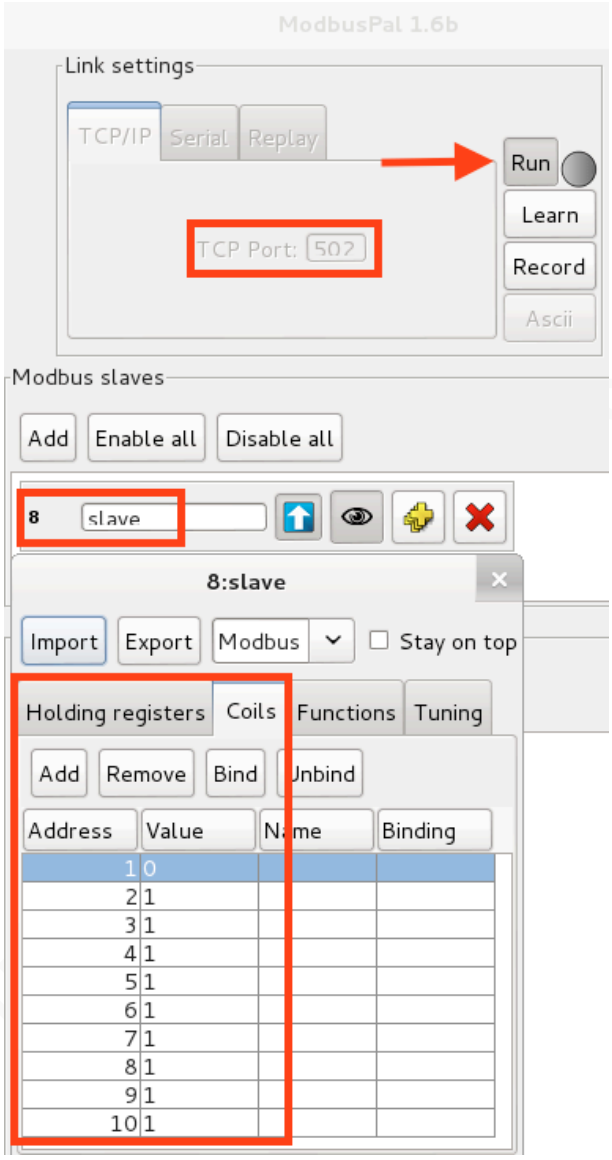


Figure 4 – PLC Configuration

3) Simulated Attacker Workstation:
    - Kali Linux 64-bit Operating System
    - IP address on same subnet (192.168.238.133)
    - Wireshark 2.2.0 Packet Analyzer
    - Ettercap 0.8.2

The entire lab network is shown in Figure 5.

© 2017 The SANS Institute Author retains full rights.

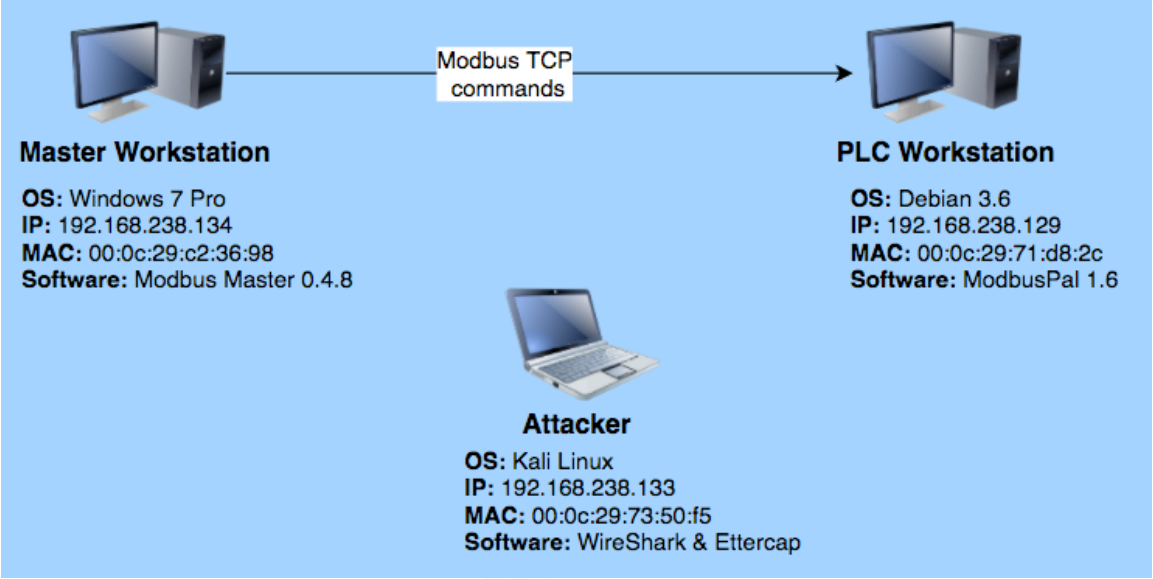


Figure 5 – Lab Network

To compare normal and MITM Modbus TCP communications, Wireshark, using the "Start Capturing Packets" feature, was utilized to capture packets prior to each exercise. This is shown in Figure 6.

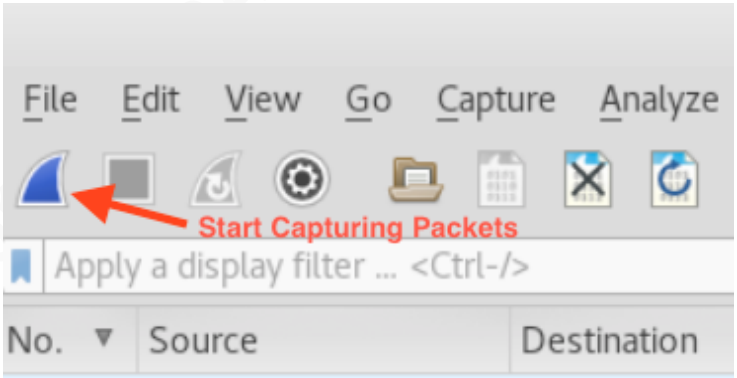


Figure 6 – Wireshark Capture Packets

3.1. Normal Modbus TCP Communications

In regard to the Modbus TCP communications between the Master and PLC workstations, the PLC workstation had ModbusPal 1.6 software with 10 coils. Each coil was designated a number between one and ten which represented an address for each single coil. The value of each coil will have either a binary value of zero (coil is off) or

© 2017 The SANS Institute Author retains full rights.

one (coil is on). The initial configuration shown in Figure 7 shows that the first PLC coil had a value of zero and that the remaining nine coils had a value of one.

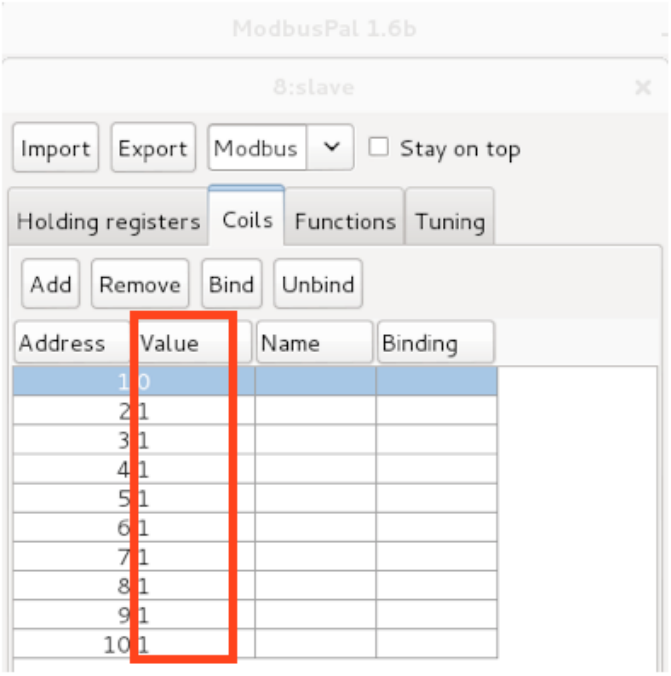


Figure 7 - PLC Coils

The Master workstation with the Modbus Master software sent its first Read/Write to the PLC workstation with a Modbus TCP command of one with a decimal format. This action is shown in Figure 8.

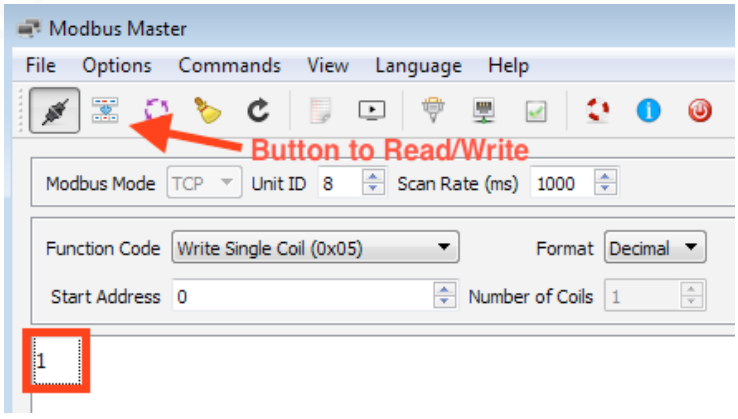


Figure 8 – Modbus Master First Read/Write

© 2017 The SANS Institute Author retains full rights.

The PLC workstation then received the Modbus TCP function code command of 0x05 from the Master workstation which changed the value of the first coil value to a one as can be seen in Figure 9.

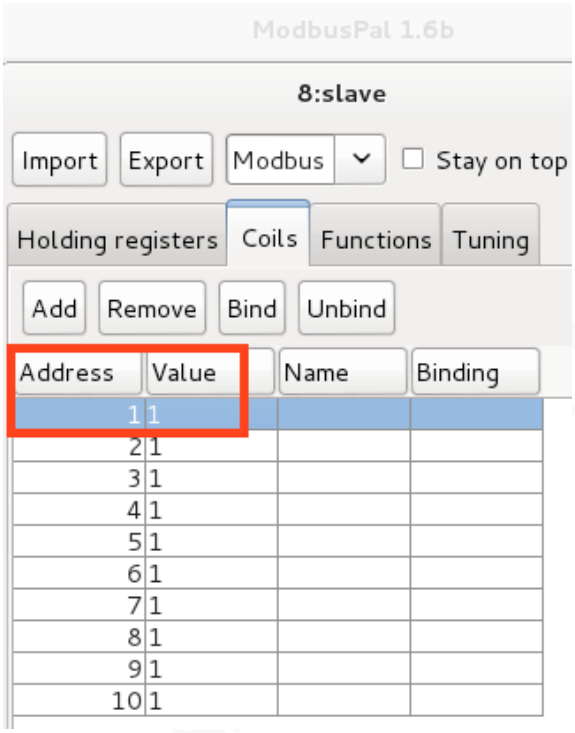


Figure 9 – PLC Coils

As a result, coils one through ten showed a value of one which meant that all of the coils were on.

3.2. MITM Modbus TCP Communications

Next, a MITM attack was performed with an Ettercap tool that maliciously modified the Modbus TCP commands between the Master and PLC workstations. Weaknesses of the Modbus TCP that were targeted by the Ettercap tool involved sending commands in clear text and lack of authentication within the Modbus TCP protocol. The Kali Linux workstation first used the Ettercap tool to put the network interface into unified sniffing mode. This is shown in Figure 10.

© 2017 The SANS Institute Author retains full rights.

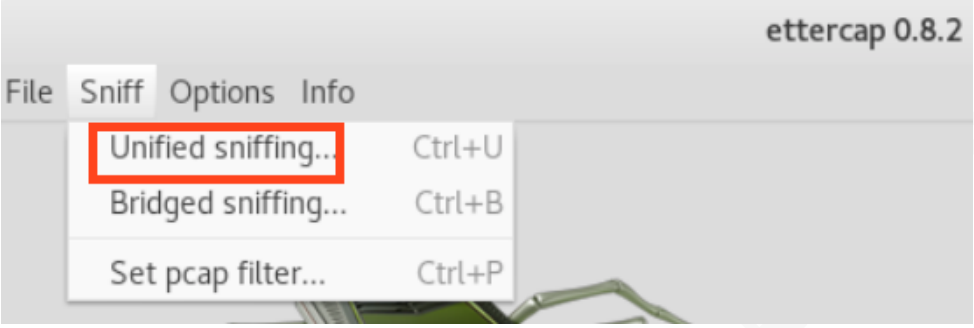


Figure 10 – Unified Sniffing

The Ettercap tool then sniffed packets on the network. This is shown in Figure 11.

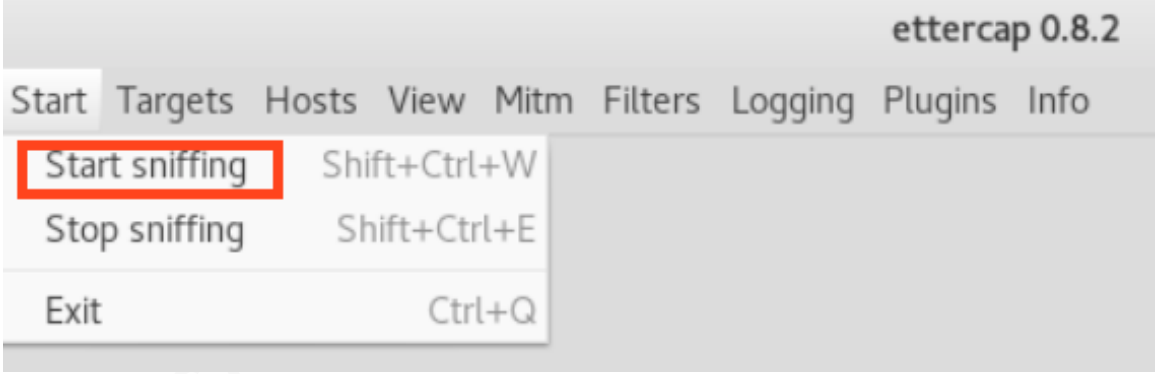


Figure 11 – Ettercap Sniffing

The Attacker workstation then scanned for hosts to find the Master and PLC workstations to attack. This is shown in Figure 12.



Figure 12 – Ettercap Scan Hosts

© 2017 The SANS Institute Author retains full rights.

Results of the Ettercap scan included both the Master and PLC workstation MAC addresses. This is shown in Figure 13.

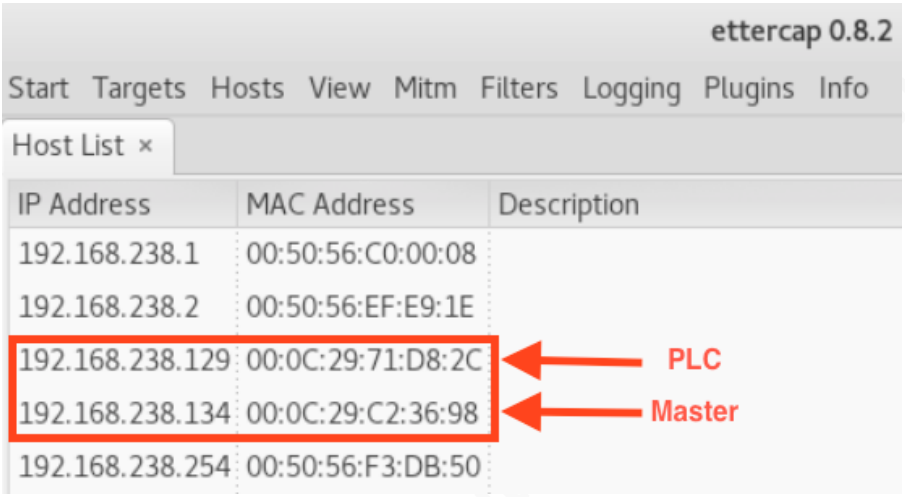


Figure 13 – Ettercap "Scan for hosts" Results

The Attacker workstation then used the MAC addresses provided by the Ettercap scan for a MITM with ARP poisoning (ARP spoofing) to send falsified ARP messages over a LAN (Glyn, 2017). This is depicted in Figure 14.

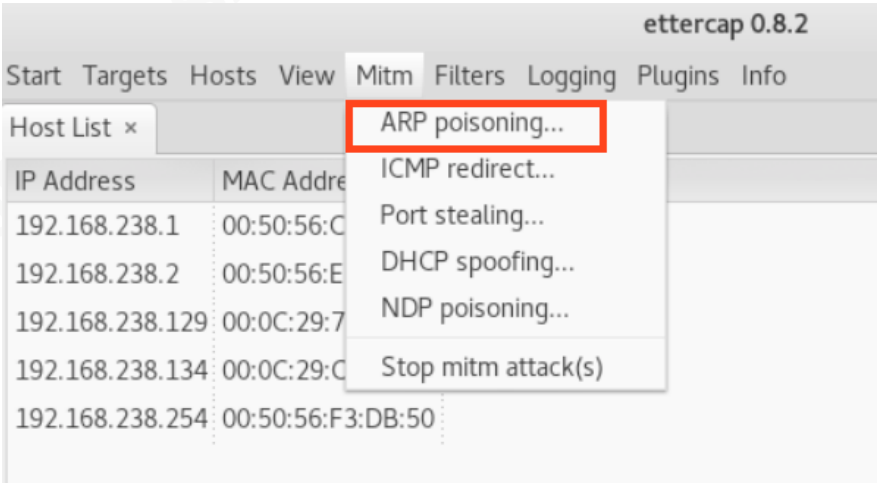


Figure 14 – ARP Poisoning

The ARP spoofing resulted in the linking of the attacker's MAC address with the IP address of a legitimate computer or server on the network (Glyn, 2017). This concept is depicted in Figure 15.

© 2017 The SANS Institute Author retains full rights.

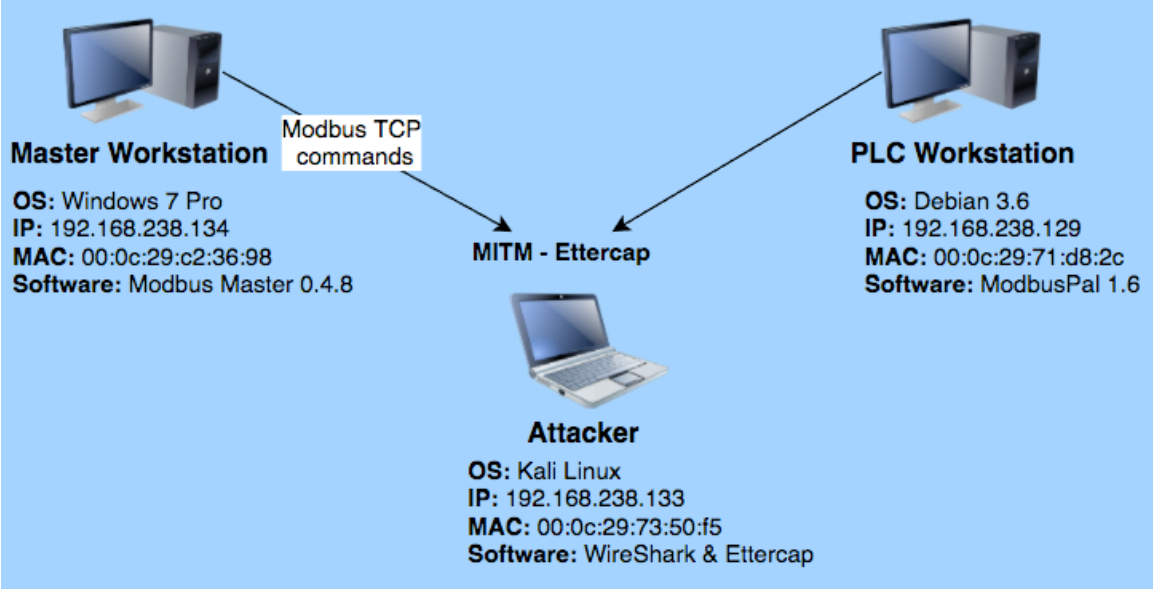


Figure 15 – MITM Attack

An Ettercap filter within the Ettercap tool was then created to modify Modbus TCP communications coming from the Master workstation with a destination to the PLC workstation. This resulted in the normal Modbus TCP communications from the Master workstation sending a read/write to the PLC workstation with a Modbus TCP command of one for the PLC's first address. The Ettercap MITM attack, combined with the proper filter, modified any Modbus TCP command with a hex value of ff00 (which resulted in the coil being on) to a hex value 0000 (which meant that the coil was off). The filter called Modbus.filter was used to compile text into a binary filter that could be interpreted by the Ettercap tool. To find the proper syntax with the etterfilter utility, the requirements for the lab were first mapped with syntax available from the etterfilter utility.

© 2017 The SANS Institute Author retains full rights.

Actions for Lab	Etterfilter Syntax
Isolate TCP protocol with destination port 502. This port is associated with Modbus TCP.	Commands: ip.proto and tcp.dst
Search payload for Modbus TCP command of hex value ff00.	**search(*where*, *what*)** This function searches the string 'what' in the buffer 'where'. The buffer can be either DATA.data or DECODED.data. The former is the payload at layer DATA (on top TCP or UDP) as it is transmitted on the wire; the latter is the payload decoded/decrypted by dissectors. (Linux.die.net, 2017)
Modify payload for Modbus TCP command ff00 to hex value 0000.	**replace(*what*, *with*)** This function replaces the string 'what' with the string 'with'. They can be a binary string and must be escaped. The replacement is always performed in DATA.data since is the only payload which gets forwarded. The 'DECODED.data' buffer is only used internally and never reaches the wire. (Linux.die.net, 2017)

Table 1 – Ettercap Filter Syntax

The final syntax, based on Table 1, for the Modbus.filter, is shown in Figure 16.

```
if (ip.proto == TCP && tcp.dst == 502) {
   if (search(DATA.data, "\xff\x00")) {
         msg( "Found Modbus On Switch of ff 00" );
         replace("\xff\x00", "\x00\x00");
   }
}
```

Figure 16 – Ettercap Filter

The Ettercap Modbus.filter file for modifying the Modbus TCP in the MITM attack was then compiled with the command “etterfilter Modbus.filter –o Modbus.ef” and loaded into the Ettercap tool. Adding the filter into Ettercap tool is shown in Figure 17.

© 2017 The SANS Institute Author retains full rights.

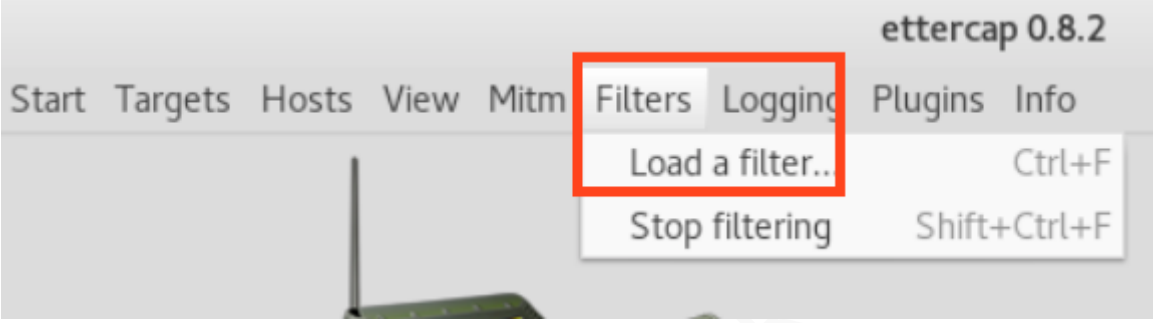


Figure 17 – Load Ettercap Filter

In the second attempt to Read/Write normal Modbus TCP communications, the Master workstation again sent a single write coil command of one with the hex value of ff00 to the first PLC workstation address. Similarly, to Figure 7, it was expected that the coil would be changed to a value of one (coil was turned on). However, with the MITM attack and the Ettercap filter, the coil was changed to zero (coil was turned off). By changing the coil to zero the attacker maliciously modified the Master workstation's attempt to turn on the PLC workstation's coil and, instead, turned the coil off without any errors occurring. The change in coil values can be observed from the ModbusPal software loaded on the PLC workstation shown in Figure 18.

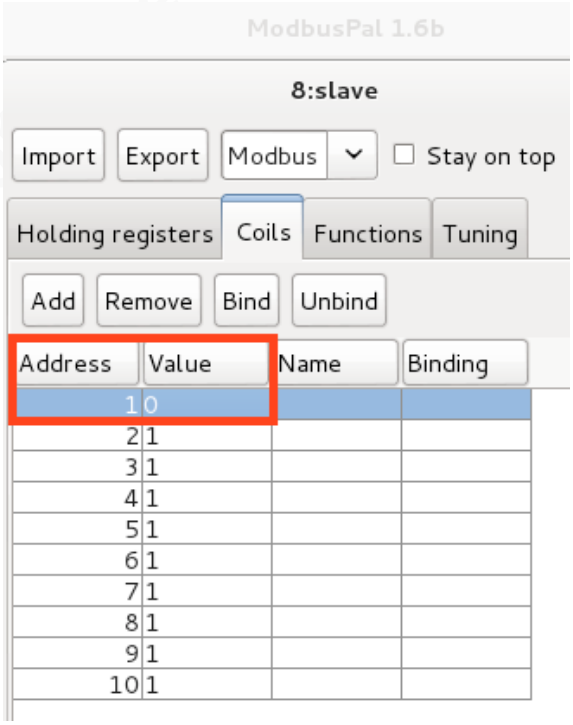


Figure 18 – Result of Ettercap Filter on PLC workstation

Gabriel Sanchez, gmgsanchez@gmail.com

© 2017 The SANS Institute Author retains full rights.

With Modbus TCP's inability to combat against clear text messages and lack of Modbus TCP with built in authentication the Ettercap filter successfully modified the packet from the Master to the PLC workstation to change the coil to a hex value of 0000.

4. Wireshark

Given the many vulnerabilities of the Modbus TCP protocol, protocol analyzers can help illustrate attacks, such as a MITM, with the Ettercap filter on the Modbus TCP protocol. One method used to observe or analyze network data was by using full packet capture which is utilized for obtaining a copy of the entire packet, including the payload and header, as it traverses a network (Sanchez, 2016). The Wireshark tool was utilized in this lab to capture packets for both normal Modbus TCP communications and a MITM attack on the Modbus TCP protocol.

4.1. Normal Modbus TCP Communications

The normal Modbus TCP communications showed that the Master, as the source workstation, sent a Modbus TCP packet to the destination PLC workstation to turn on a single coil. Both MAC addresses correlate to the correct workstation addresses of the Master and PLC as depicted in Figure 3. Table 2 shows a summary of the IP's, ports, and MAC addresses of Frame 5.

Source	**IP/Port**	**MAC**		**Dest.**	**IP/Port**	**MAC**
Master	192.168.238.134 Port: 46851	00:0c:29:c2:36:98		PLC	192.168.238.129 Port: 502	00:0c:29:71:d8:2c

Table 2 – Frame 5

A packet analysis with Wireshark of the Modbus TCP packet confirmed that Frame 5 had the unit ID eight, the function code to write a single coil, and the hex value of ff00 (ModbusTools, 2017). This is illustrated with Wireshark in Figure 19.

© 2017 The SANS Institute Author retains full rights.

```
Frame 5: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: Vmware_c2:36:98 (00:0c:29:c2:36:98), Dst: Vmware_71:d8:2c (00:0c:29:71:d8:2c)
Internet Protocol Version 4, Src: 192.168.238.134, Dst: 192.168.238.129
Transmission Control Protocol, Src Port: 46851, Dst Port: 502, Seq: 1252658470, Ack: 1290630787, Len: 12
Modbus/TCP
  Transaction Identifier: 27
  Protocol Identifier: 0
  Length: 6
  Unit Identifier: 8
Modbus
  .000 0101 = Function Code: Write Single Coil (5)
  Reference Number: 0
  Data: ff00
  Padding: 0x00
```

Figure 19 – Master to PLC Packet Analysis

The PLC workstation then sent a response back in Frame 6 to the Master workstation to confirm the hex value of ff00 within its Modbus TCP packet. This is illustrated with Wireshark in Figure 20.

```
Frame 6: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: Vmware_71:d8:2c (00:0c:29:71:d8:2c), Dst: Vmware_c2:36:98 (00:0c:29:c2:36:98)
Internet Protocol Version 4, Src: 192.168.238.129, Dst: 192.168.238.134
Transmission Control Protocol, Src Port: 502, Dst Port: 46851, Seq: 1290630787, Ack: 1252658482, Len: 12
Modbus/TCP
  Transaction Identifier: 27
  Protocol Identifier: 0
  Length: 6
  Unit Identifier: 8
Modbus
  .000 0101 = Function Code: Write Single Coil (5)
  [Request Frame: 5]
  Reference Number: 0
  Data: ff00
  Padding: 0x00
```

Figure 20 – PLC to Master Packet Analysis

Finally, in Frame 7, an acknowledgment from the Master workstation to the PLC workstation was sent to convey that the Modbus TCP communication was complete. The entire communication of the query, response, and acknowledgment between the Master workstation and the PLC workstation is illustrated with Wireshark in Figure 21.

No.	Source	Destination	Info
5	192.168.238.134	192.168.238.129	Query: Trans: 27; Unit: 8, Func: 5: Write Single Coi
6	192.168.238.129	192.168.238.134	Response: Trans: 27; Unit: 8, Func: 5: Write Single Coi
7	192.168.238.134	192.168.238.129	46851→502 [ACK] Seq=1252658482 Ack=1290630799 Win=255 Len=0

Figure 21 – Master/PLC show Query/Response/Acknowledgment

© 2017 The SANS Institute Author retains full rights.

4.2. MITM Attack on Modbus TCP Communications

Wireshark was utilized to analyze the normal behavior, abnormalities, and malicious behavior of the Modbus TCP communications. The MITM attack showed that the Master workstation, as the source, sent a Modbus TCP packet to the destination PLC workstation to turn on a single coil. However, one notable observation between Table 2 and Table 3 are the MAC addresses for the PLC workstation which do not match up. This proved to be significant since the MAC address of 00:0c:29:73:50:f5, as shown in bold, in Table 3 was associated with the Kali Linux workstation MAC address and not the PLC workstation. This is the first indication that a MITM attack could potentially be occurring. Table 3 also shows a summary of the IP's, ports, and MAC addresses of Frame 82.

Source	IP/Port	MAC		Dest.	IP/Port	MAC
Master	192.168.238.134 Port: 46851	00:0c:29:c2:36:98		PLC	192.168.238.129 Port: 502	**00:0c:29:73:50:f5**

Table 3 – Frame 82

A packet analysis that utilized Wireshark for the Modbus TCP packet confirmed that Frame 82 had the Kali Linux workstation MAC address, unit ID of eight, function code to write a single coil, and the hex value of ff00 to turn the coil on. This packet is shown in Figure 22.

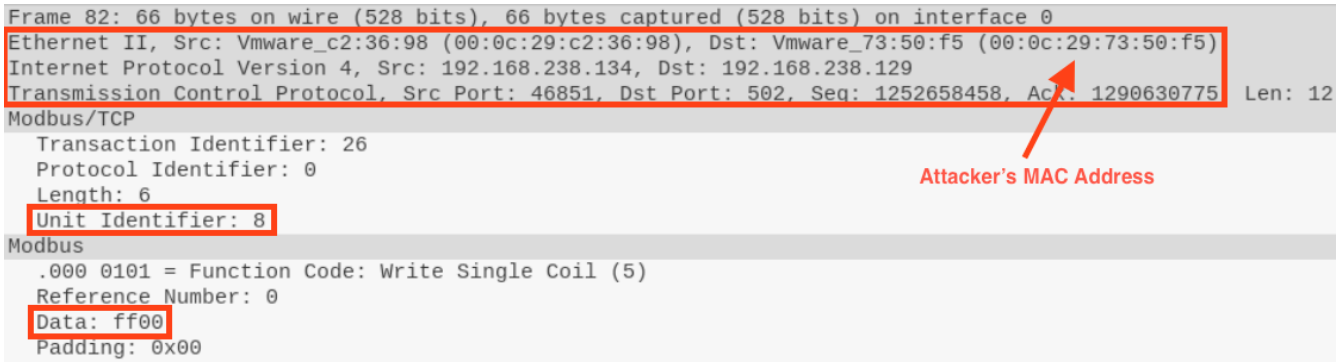


Figure 22 – Frame 82

© 2017 The SANS Institute Author retains full rights.

The next packet, Frame 83, was the Master workstation that sent a TCP retransmission packet to the destination PLC workstation. The TCP retransmission error occurred since the Kali Linux workstation MAC address of 00:00c:29:73:50:f5 was poisoning the Master's workstation MAC address as the source. The spoofed MAC address associated with the Kali Linux workstation is in bold in Table 3 along with a summary of the IP's, ports, and MAC addresses of Frame 83.

Source	IP/Port	MAC		Dest.	IP/Port	MAC
Master	192.168.238.134 Port: 46851	**00:0c:29:73:50:f5**		PLC	192.168.238.129 Port: 502	00:0c:29:71:d8:2c

Table 3 – Frame 83

Frame 83, with the use of Wireshark, confirmed that the MAC address of the Kali Linux workstation associated with the Master workstation had been spoofed. The spoofed MAC address can be seen in the packet shown in Figure 23.

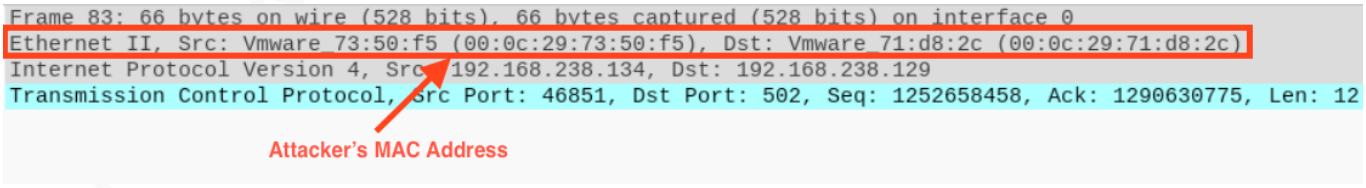


Figure 23 – Frame 83

The following packet, Frame 84, was from the PLC workstation which sent a Modbus TCP packet to the Master workstation. Table 4 shows that the Master is still associated with the Kali Linux workstation MAC address, in bold, and includes summary of the IP's, ports, and MAC addresses of Frame 84.

Source	IP/Port	MAC		Dest.	IP/Port	MAC
PLC	192.168.238.129 Port: 502	00:0c:29:71:d8:2c		Master	192.168.238.134 Port: 46851	**00:0c:29:73:50:f5**

Table 4 – Frame 84

© 2017 The SANS Institute Author retains full rights.

A packet analysis of Frame 84, which utilized Wireshark for the Modbus TCP packet, showed that the response from request Frame 82 had a function code to write a single coil and the hex value of 0000. The modified data of 0000 was successful since the MAC address of 00:0c:29:73:50:f5 was associated with the Kali Linux workstation MAC address and not the Master workstation. Additionally, Frame 84 showed a response from the Modbus TCP data request to be 0000 as opposed to the request from the Master in Frame 82 of ff00. This confirmed that the Modbus TCP data was successfully modified by the MITM Ettercap filter and that the PLC workstation accepted the Modbus TCP command. The modified data by the attacker's MAC address is illustrated with Wireshark Figure 24.

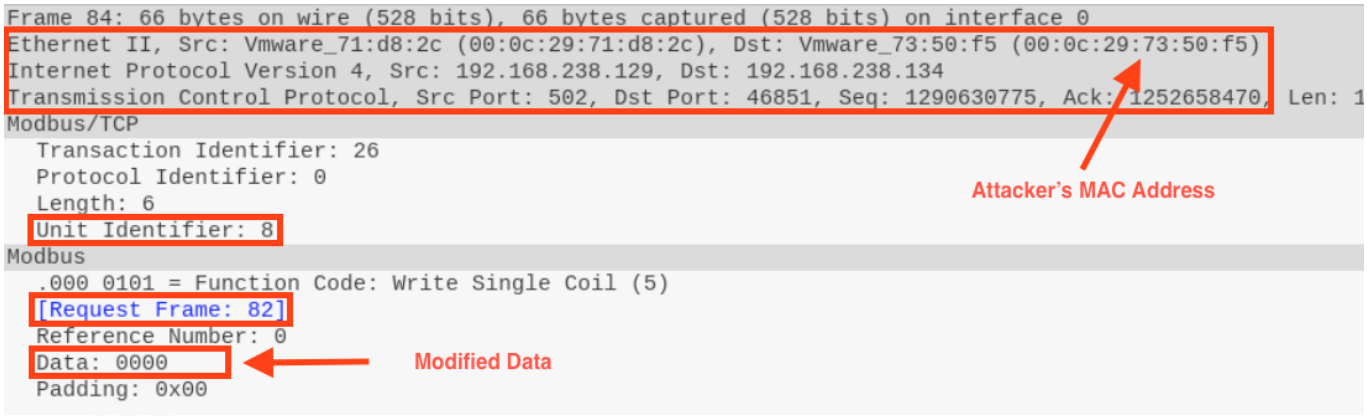


Figure 24 – Frame 84

The next packet, Frame 85, which had the PLC workstation as the source, sent a TCP retransmission packet to the Master workstation which was very similar to Frame 83 that originated from the Kali Linux workstation MAC address highlighted in bold in Table 5. The PLC workstation with IP address 192.168.238.129 was now the source of the packet instead of 192.168.238.134. Table 5 shows a summary of the IP's, ports, and MAC addresses of Frame 85.

Source	**IP/Port**	**MAC**		**Dest.**	**IP/Port**	**MAC**
PLC	192.168.238.129 Port: 502	**00:0c:29:73:50:f5**		Master	192.168.238.134 Port: 46851	00:0c:29:c2:36:98

Table 5 – Frame 85

© 2017 The SANS Institute Author retains full rights.

A packet analysis of Frame 85, which employed Wireshark for the Modbus TCP packet, showed the spoofed MAC address for the PLC workstation as is shown in Figure 25.

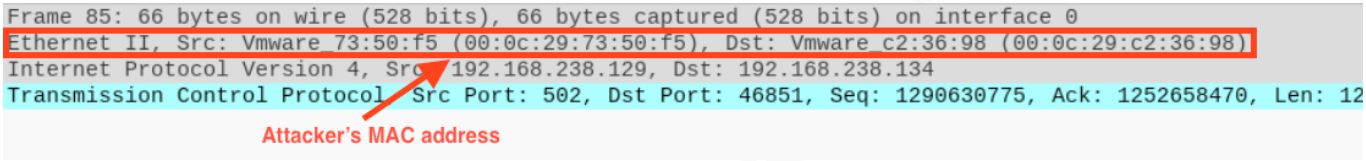


Figure 25 – Frame 85

With the Master workstation as the source, Frame 86 sent a TCP acknowledgment packet to the Kali Linux workstation MAC address spoofing the PLC workstation which originated from Frame 84. Table 6 shows the spoofed MAC address in bold and a summary of the IP's, ports, and MAC addresses of Frame 86.

Source	IP/Port	MAC		Dest.	IP/Port	MAC
Master	192.168.238.134 Port: 46851	00:0c:29:c2:36:98		PLC	192.168.238.129 Port: 502	**00:0c:29:73:50:f5**

Table 6 – Frame 86

A packet analysis of Frame 86 employed with Wireshark showed that the Kali Linux MAC address of 00:0c:29:73:50:f5 was used for the PLC workstation instead of 00:0c:29:71:d8:2c. The attacker's MAC address shown in place of the PLC MAC address is illustrated by Wireshark in Figure 26.

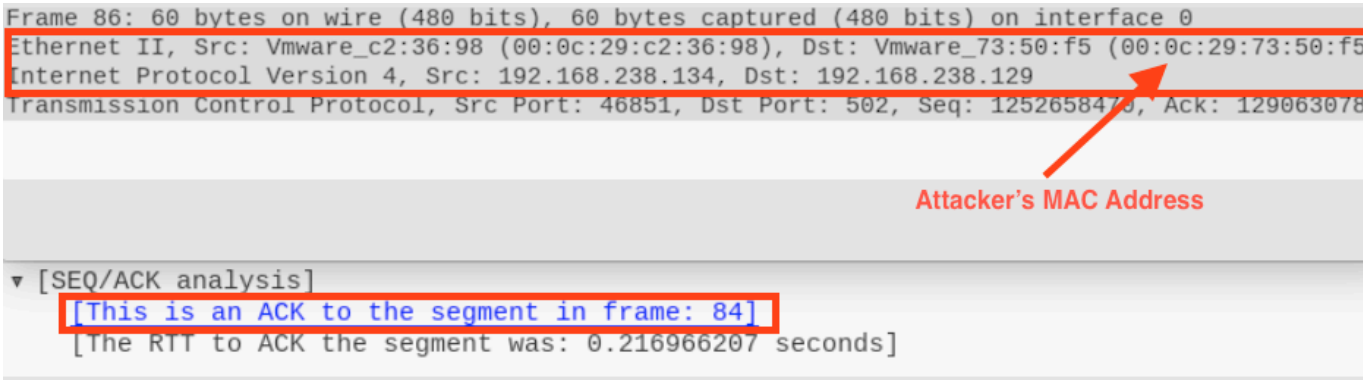


Figure 26 – Frame 86

© 2017 The SANS Institute Author retains full rights.

For Frame 87 the Master workstation sent a TCP duplicate acknowledgment packet to the PLC workstation. The duplicate acknowledgment was a result of Frame 86 already having sent an acknowledgment to the PLC workstation. In bold, an acknowledgment that the Kali Linux workstation MAC address was spoofing the PLC workstation is shown in Table 7. Table 7 also shows a summary of the IP's, ports, and MAC addresses of Frame 87.

Source	**IP/Port**	**MAC**		**Dest.**	**IP/Port**	**MAC**
Master	192.168.238.134 Port: 46851	**00:0c:29:73:50:f5**		PLC	192.168.238.129 Port: 502	00:0c:29:71:d8:2c

Table 7 – Frame 87

With the use of Wireshark the packet analysis of Frame 87 showed that the second acknowledgment and the spoofed Kali Linux MAC address of 00:0c:29:73:50:f5 were being utilized instead of the authentic MAC address for the Master workstation 00:0c:29:c2:36:98. This is illustrated with Wireshark in Figure 27.

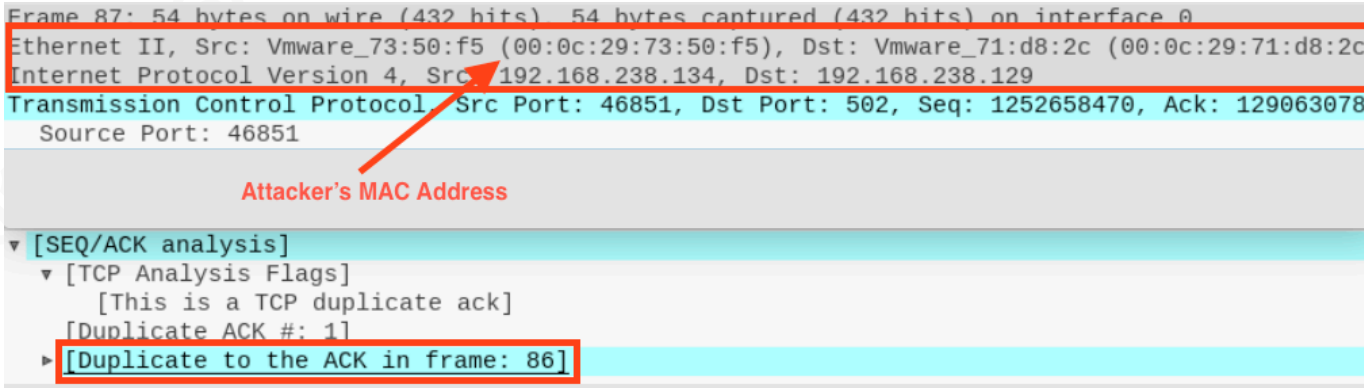
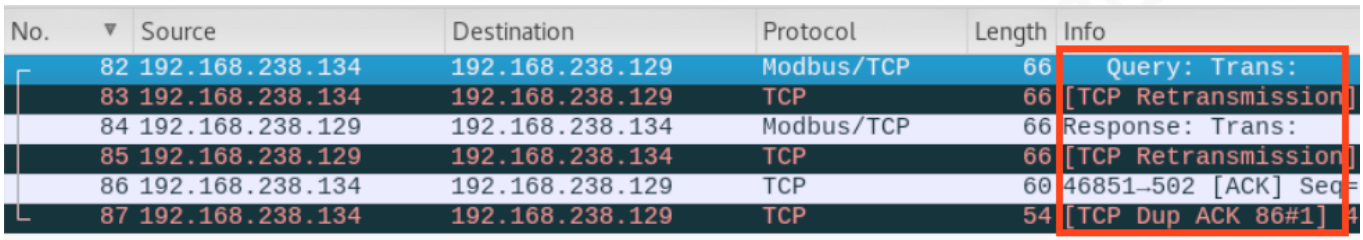


Figure 27 – Frame 87

Frames 82-87, as shown in Figures 22-27, completed the query, response, and acknowledgment communications. This is illustrated with Wireshark in Figure 28.

© 2017 The SANS Institute Author retains full rights.



No.	Source	Destination	Protocol	Length	Info
82	192.168.238.134	192.168.238.129	Modbus/TCP	66	Query: Trans:
83	192.168.238.134	192.168.238.129	TCP	66	[TCP Retransmission]
84	192.168.238.129	192.168.238.134	Modbus/TCP	66	Response: Trans:
85	192.168.238.129	192.168.238.134	TCP	66	[TCP Retransmission]
86	192.168.238.134	192.168.238.129	TCP	60	46851→502 [ACK] Seq=
87	192.168.238.134	192.168.238.129	TCP	54	[TCP Dup ACK 86#1] 4

Figure 28 – Query/Response/Acknowledgment

Overall, Wireshark illustrated critical differences between normal Modbus TCP communications and the MITM attack on the Modbus TCP communications. Normal communications contained MAC addresses that correlated to the correct workstations and did not contain TCP errors. The MITM attack, on the other hand, showed that communications were not typical to Modbus TCP communications since the Kali Linux workstation's MAC address was posing as both the Master and PLC workstations. The MITM attack also demonstrated TCP retransmission errors and duplicate acknowledgments. Furthermore, Wireshark illustrated that the MITM attack with the Ettercap filter successfully changed a request from the Master to the PLC for a coil to be turned off instead of being turned on. No indication was returned to the Master that its original Modbus TCP command had been modified.

5. Conclusion

A lab was set up with three machines configured as a Master, PLC, and attacker workstations. Both the Master and PLC workstations ran Modbus TCP software for managing coils and the attacker's workstation was loaded with Ettercap for a MITM attack. Wireshark was employed to illustrate communications between the Master and a PLC workstation. In the first scenario, the Master workstation sent a Modbus TCP command to turn on a single coil in the PLC workstation. The second scenario completed the same task except a MITM attack was introduced into the scenario with an Ettercap filter to modify the write coil command from the Master workstation to the PLC workstation from on to off. In scenario one, Wireshark showed the expected Modbus TCP communications between the Master and PLC workstations. In scenario two,

© 2017 The SANS Institute Author retains full rights.

Wireshark revealed abnormalities in Modbus TCP communications such as unknown MAC addresses, retransmission errors, and duplicate acknowledgments. As long as the Modbus TCP protocol exists within an ICS network, attackers will continue to take the path of least resistance and attack the flaws with this protocol. Utilizing the research provided in this paper defenders will be able to quickly setup a lab to mimic and observe attacks on ICS networks. This will assist the cyber defenders in testing their defenses and improve detection of potential attacks on one's ICS networks.

© 2017 The SANS Institute, Author Retains Full Rights

Gabriel Sanchez, gmgsanchez@gmail.com

© 2017 The SANS Institute Author retains full rights.

6. References

Acromag Incorporated. (2005). Introduction to Modbus TCP/IP. Retrieved from https://www.acromag.com/sites/default/files/Acromag_Intro_ModbusTCP_765A.pdf

Electromate. (2014, May 14). Galil Presentation The Evolution Of Ethernet In Motion And I/O Control Webinar. Retrieved from https://www.slideshare.net/Electromate/galil-presentation-the-evolution-of-ethernet-in-motion-and-io-control-webinar

Glynn, F. (2017). ARP Spoofing | Veracode. Retrieved from https://www.veracode.com/security/arp-spoofing

Linux.die.net. (2017). etterfilter(8) - Linux man page. Retrieved from https://linux.die.net/man/8/etterfilter

Kaspersky Lab. (2016). Threat Landscape for Industrial Automation Systems in the second half of 2016 | Kaspersky Lab ICS CERT. Retrieved from https://ics-cert.kaspersky.com/reports/2017/03/28/threat-landscape-for-industrial-automation-systems-in-the-second-half-of-2016/

Kovacs, E. (2016, December 27). IBM Reports Significant Increase in ICS Attacks | SecurityWeek.Com. Retrieved from http://www.securityweek.com/ibm-reports-significant-increase-ics-attacks

Modbus. (2017). Modbus FAQ. Retrieved from http://www.modbus.org/faq.php

ModbusTools. (2017). Modbus Protocol. Retrieved from http://www.modbustools.com/modbus.html#function05

© 2017 The SANS Institu Author retains full rights.

Sanchez, G. (2016, January 21). Don't Always Judge a Packet by Its Cover. Retrieved from https://www.sans.org/reading-room/whitepapers/access/dont-judge-packet-cover-36745

Schneider Electric. (2017). What is Modbus and How does it work? Retrieved from http://www.schneider-electric.com/en/faqs/FA168406/

Simply Modbus. (2017). Simply Modbus - About Modbus TCP. Retrieved from http://www.simplymodbus.ca/TCP.htm

UNSERVER. (2016). What is Modbus? Retrieved from https://unserver.xyz/blog/what-is-modbus/

Gabriel Sanchez, gmgsanchez@gmail.com

© 2017 The SANS Institute Author retains full rights.