The Use of Attack Trees in Assessing Vulnerabilities in SCADA Systems

Eric J. Byres
Group for Advanced Information Technology, British Columbia Institute of Technology
eric_byres@bcit.ca

Matthew Franz and Darrin Miller
Critical Infrastructure Assurance Group, Cisco Systems Inc.
mdfranz@gmail.com darrimil@cisco.com

Abstract

Protocol standards, particularly those for critical control systems in the petroleum and power industry, have traditionally been designed to address a specific application with little regard for security. At best, there has been only passing concern for security issues that may arise in deployment; at worst, protocol designers assume a closed (and therefore secure) environment, which, in many cases, no longer exists. Where security has been a consideration, there has been no clear methodology to assess the security risks in the protocol specification. This paper describes the application of the attack tree methodology to the common SCADA protocol MODBUS/TCP. The authors identify eleven possible attacker goals and identify security vulnerabilities inherent in both the specification and in typical deployments of SCADA systems. These are then used to suggest possible best practices for SCADA operators and improvement to the MODBUS protocol.

1. SCADA protocols and security

Supervisory Controls and Data Acquisition (SCADA) protocols are communications protocols designed for the exchange of control messages on industrial networks. Over the past three decades, several hundred of these protocols have been developed for both serial, LAN and WAN based communications in a wide variety of industries including petrochemical, automotive, transportation and electrical generation/distribution. Approximately 10 protocols currently dominate the industrial marketplace and include protocols such as MODBUS, DNP3, EtherNET/IP, PROFIBUS and Foundation Fieldbus. The choice of protocol is typically a function of the operating requirements, industry preference, vendor and the design history of the system. For example, in an oil refinery an operator workstation might use the MODBUS/TCP protocol to communicate with a control device such as a Programmable Logic Controller (PLC). Alternatively, in power utility's SCADA system, a master located in a central facility could use the DNP3 protocol to query and control slave Remote Terminal Units (RTU) distributed in remote sub-stations.

Most SCADA protocols were designed long before network security perceived to be a problem. The traditional SCADA system was a closed serial network that contained only trusted devices with little or no connection to the outside world. As control networks evolved, the use of TCP/IP and Ethernet became common place and interfacing to business systems became the norm. The result was that the closed trust model no longer applied and vulnerabilities in theses systems began to appear [1]. In particular, network security problems from the business network and the world at large could be passed onto process and SCADA networks, putting industrial production, environment integrity and human safety at risk.

One of the primary weaknesses exploited in attacks against the Internet and business information systems are vulnerabilities in the communications protocols and their implementations. SCADA systems are no exception to this rule, but little is known about the specific vulnerabilities in SCADA protocols. To address this, the Group for Advanced Information Technology (GAIT) at BCIT and the Cisco Systems' Critical Infrastructure Assurance Group (CIAG) chose to investigate possible vulnerabilities in the SCADA protocol MODBUS/TCP. This protocol was selected as a starting point since it is both one of the simplest and most widely used of all SCADA protocols in critical infrastructures.

2. The MODBUS protocol

The MODBUS communications system was created in the late 1970's by the Modicon Corporation (now Schneider Electric) to allow communications to its line of industrial PLCs. The protocol's simplicity and efficiency, combined with the publishing of its specifications by Modicon [2], caused it to become widely adopted throughout the industrial controls and SCADA world as a common industrial standard.

The original MODBUS system was a simple two-layer communication stack running on top of a serial EIA-232 link. As different physical layer options became available, it was subsequently marketed as a number of different of network products, the best known of which are MODBUS, MODBUS+ and MODBUS/TCP. The common element in all of these MODBUS networks is a client-server command structure commonly known as the MODBUS Application Protocol (MBAP), a layer-7 protocol in the Open Systems Interconnection Reference Model (OSI/RM) as illustrated in Figure 1.

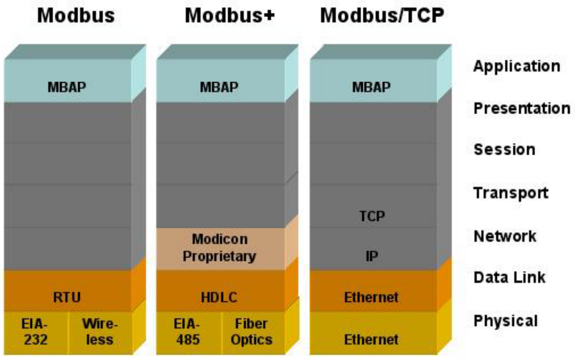

Figure 1 – The MODBUS protocol family OSI stack representation

A simple request-reply scheme is used for all MBAP transactions. The client (also known as master) device initiates a request and the server (also known as slave) replies. For example, when a Human Machine Interface (HMI) workstation requires a value from a PLC it sends a request message to start the data transfer process. The PLC then sends a response providing the requested information. In this situation, the device running the HMI is acting as the client/master and the PLC is acting as the server/slave.

Each message contains a function code is that is set by the client/master and indicates to the server/slave what kind of action to perform. Function codes are the same for requests and responses since the server simply reflects the function code back to the client. There are 127 possible function codes that fall into three general categories: Public function codes, User Defined function codes and Reserved function codes. Sub-codes are added to some function codes to define multiple actions or to allow future enhancements.

3. Using attack trees to model system threats and vulnerabilities

Over the past few years the information technology world has seen exponential growth in the number of security vulnerabilities being reported for common networked systems. For example, the Carnegie-Mellon CERT "*Vulnerabilities Reported 1995-2002*" reports cyber vulnerabilities have grown from less than 300 per year in 1998 to over 4000 only four years later [3]. The overwhelming growth of vulnerabilities has become one of the key challenges facing operational security personnel who must not only consider an increasing number of attacks, but how these attacks can be combined in complex ways. Clearly a methodology is needed to organize attack possibilities, understand their inter-relationships and rank them according to risk.

The approach selected in this paper is the "Attack Tree" technique as initially described by Bruce Schneier [5]. This technique provides a structured yet flexible means of conducting security analyses of protocols, applications, and networks. Although "fault trees" have long been an accepted system analysis technique, this methodology was first applied to the domain information security a *Dr Dobb's Journal* article in 1999. Subsequently, CERTC/CC developed a more formal application of the technique, introduced standardized notation and provided more complex examples. The first published application of attack trees to a network protocol was "An Attack Tree for the Border Gateway Protocol" [4], which is currently under consideration by the IETF Routing Protocol Security working group.

Building on these approaches, the project team relied heavily on attack trees to support our vulnerability analysis and testing of MODBUS/TCP with the goal of identifying flaws that could result in the greatest damage to SCADA systems. One of the primary benefits of using attack trees is that they focus analysis on measurable goals that can ultimately be translated into specific tests against real-world devices, networks, and protocol implementations. This helps avoid the trap of overly-academic security research that often fails to consider the difficulty of conducting the attacks and cannot measure the impact on targeted systems.

Attack trees also encourage a structured elaboration of events (i.e. specific attack goals) that must occur for a successful intrusion to take place. This promotes the consideration of all reasonable avenues of approach for an attack and also facilitates the identification and optimal deployment of countermeasures. Furthermore, since each node (a discrete attacker goal) may be decomposed into subordinate nodes (sub-goals, or a means of achieving the parent goal), attack trees allow security analysis to be conducted at multiple layers of abstraction, allowing researchers to focus on areas of interest while acknowledging other intrusion paths. Lastly, using attack trees allows common attacks to be referenced as reusable modules that apply to multiple network scenarios.

The clearest way to demonstrate this is by example, as illustrated by Schneier in his original article on the attack tree methodology [5]. For instance, consider an individual trying to gain unauthorized physical access to a building. An attack tree for such an act might look like this:

```
Goal: Gain unauthorized physical access
to building
Attack:
   OR
   1. Unlock door with key
   2. Pick lock
   3. Break window
   4. Follow authorized individual into
      building
```

This simple tree should be read as follows: to gain unauthorized physical access to a building, the adversary must unlock the door with a key, pick the lock, break a window, or follow an authorized individual into the building. The "OR" operator defines that only one is required. In the same tree, replacing the "OR" with "AND" would require that all subordinate goals be achieved to realize the parent goal. Attack trees at this level of detail are of limited use. Their true value comes in understanding how an adversary can execute one of the listed subordinate goals. This requires the following, more detailed, attack tree:

```
Goal: Gain unauthorized physical access
to building
OR
1. Unlock door with key
   OR
   1.1. Steal Key
   1.2. Social Engineering
      OR
      1.2.1. Borrow key
      1.2.2. Convince locksmith to
             unlock door
2. Pick lock
3. Break window
4. Follow authorized individual into
building
   AND
   4.1 Wear appropriate clothing for the
      location
   OR
      4.2.1. Act like you belong and
      follow someone else
      4.2.2. Befriend someone
      authorized outside a building
      4.2.3. Appear in need of
      assistance (e.g. carry large box)
```

Now the various sub nodes of the tree are better defined. In order to “unlock door with key” you need to either steal the key or perform some type of social engineering. Sub goal 4 (Follow authorized individual into building) illustrates the use of “OR” and “AND” at the same level of the tree. This should be read as follows: In order to follow an individual into the building the adversary needs to wear appropriate clothing for the location and do one of the next 3 listed items.

The use of attack trees also allows comparison between technical and non-technical (and cyber and physical, in the case of SCADA systems) means of attack, supporting a more holistic analysis of threats and vulnerabilities and integrating physical, personal, and information security disciplines. Even without extensive elaboration, we learn in this tree that following someone into a building is probably the easiest way of gaining entrance with the lowest amount of cost or risk to the adversary.

Published vulnerability analysis of specific protocols is still a relatively new endeavour. To date the routing protocol Border Gateway Protocol (BGP) has received the most attention with ITEF draft Request for Comment (RFC) documents being submitted by Murphy[6] and by Convery et al as noted earlier. The latter draft RFC uses attack trees to describe the possible vulnerabilities of BGP, but does not presently assign any risk or difficulty values to the leaves of the tree. Prior to this report, there does not appear to have been any published attempts to systematically analyze the vulnerabilities of any industrial SCADA protocol.

3.1. Assessing the Risk

When studying the possible security vulnerabilities, it is easy to get caught in a trap of trying to address issues that are technically interesting, but are ultimately of low risk to the system. Thus some method of assessing and rating the risk of any vulnerability is needed. The risk in this case is an expression of the likelihood that a defined threat will exploit a specific set of vulnerability of a particular attractive target to cause a given set of consequences. Risk is influenced using a number of interrelated indicators including:

- Technical Difficulty
- Probability of Apprehension
- Cost of Attack
- Likelihood of Attack Success
- Site Conditions
- Installed Countermeasures

Since this analysis covers a number of different industries and possible deployment scenarios, it is impossible to know the countermeasures and conditions on each site. We developed a representative MODBUS deployment model and an assumed security environment to assess the risk. Both the deployment model and an assumed security environment are based on our experience in industrial facilities and an understanding of current industry practices. Facilities with unusual system designs or superior (or inferior) security practices will find it easy to adjust the analysis to fit their conditions.

For cyber attacks, we believed that the technical difficulty of an attack is the most critical indicator of possible attack success. Compared to physical threats, most cyber attackers appear to have little cost or apprehension concerns. Thus the team rated each edge leaf on an attack tree on a scale of one to four:

1. **Trivial**: Little technical skill required
2. **Moderate**: Average cyber hacking skills required
3. **Difficult**: Demands a high degree of technical expertise
4. **Unlikely**: Beyond the known capability of today's best hackers

Within the attack trees a node's value is derived from the values of its children. In a leaf node (which has no children), the values are entered directly by the team. Non-leaf node's indicator values are computed by indicator functions. Two mathematical functions are defined for each indicator, one for the AND condition (the maximum of the children nodes values) and one for the OR condition (the minimum of the children nodes values.)

The ultimate goal of the analysis is to determine the indicator values associated with the root (topmost) node and understand the path that influenced this value. Since the root of the tree represents the ultimate goal of the attacker, the indicator values associated with the root node reflect the resources required to compromise the system. They also indicate the most likely method of attack and where security resources are required.

3.1. Elaborating Attacker Goals

After identifying base risk and environmental assumptions, we brainstormed possible attacker objectives. The intent was to determine all the attacker goals that an intruder might attempt to achieve against a MODBUS-based SCADA system. The team defined eleven such goals:

1. Gain SCADA System Access
2. Identify MODBUS Device
3. Disrupt Master-Slave Communications
4. Disable Slave
5. Read Data from Slave
6. Write Data to Slave
7. Program Slave
8. Compromise Slave
9. Disable Master
10. Write Data to Master
11. Compromise Master

These attacker goals were then categorized into general classes and relationships in the form of a meta-tree. Each goal was ranked roughly in terms of the potential severity of impact (e.g. reading data from a slave device is likely less serious as compared to writing data to the slave). Figure 2. shows these basic relationships and ranking. As the specific attack trees will illustrate, the relationships between goals in the real world are likely to be far more complex.

In addition, the study team defined four Supporting Goals that would likely not be an end goal on their own, but would be often required by an attacker to achieve his or her objectives. Each is used in more than one attacker goal. These include:

12. Denial of Service Against Networked Device
13. Intercept or Modify Data Through Man-in-the-Middle (MITM) Attack
14. TCP Sequence Number Attack
15. Sniff Traffic

Each of these supporting goals is a well known IT network attack and is beyond the scope of this analysis. See the “An Attack Tree for the Border Gateway Protocol” [3] for discussion of these attacks.

4. Sample attack trees

Below is a representative sample of the set of attack trees developed for the study, along with the estimated difficulty for each node. A complete set of will be available in another work.

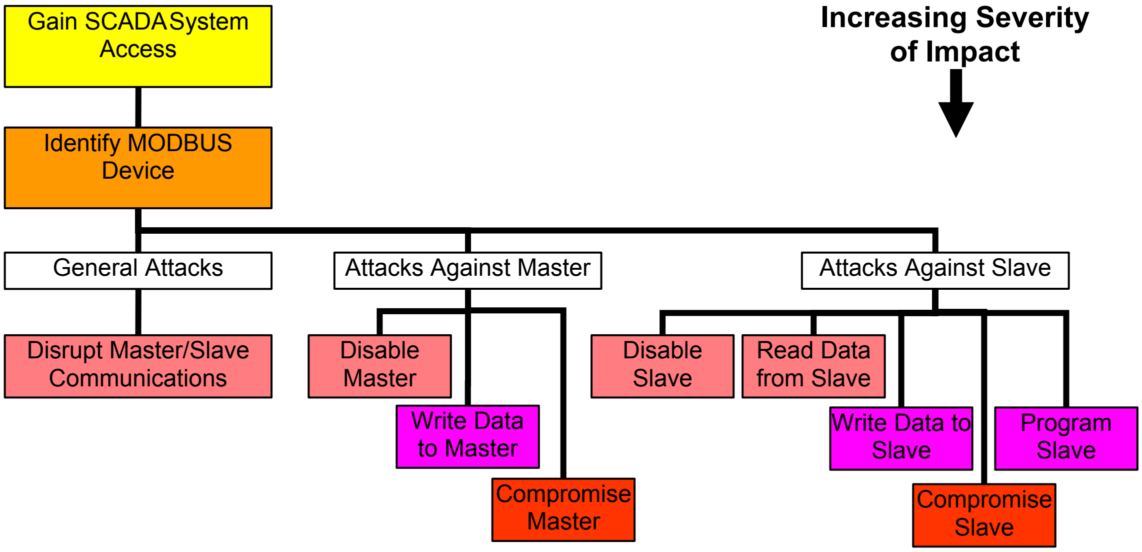

Figure 2: Interrelations and approximate severity of attacker goals

4.1. Attack Goal #1: Gain SCADA System Access

A clear precursor to launching any cyber attack is gaining some sort of network access to the target system. While the obvious (and typically the most restricted) access is via the un-trusted Internet, it is by no means the only point of attack. The following attack tree outlines the methods of gaining access to the SCADA system or Process Control Network (PCN).

```
Attack: Gain SCADA Access (Difficulty=2)
OR
1.  Gain physical access to remote field site
    equipment                                   2
2.  Gain access to SCADA link media             2
    OR
    2.1. Intercept wiring leaving building or
         compound                               2
    2.2. Intercept SCADA link in public carr.   3
    2.3. Intercept SCADA link over radio link   3
3.  Gain local Process Control Network (PCN)
    access                                      2
    OR
    3.1. Gain physical access to device on the
         PCN                                    3
    3.2. Gain dial-in access to device on PCN   2
    3.3. Gain wireless access to the PCN        2
4.  Gain remote access to PCN via IT network    3
    AND
    4.1. Gain Network Access to IT network      3
        OR
        4.1.1.  Gain physical access to IT
                network                         3
        4.1.2.  Gain remote access to IT net    3
    4.2. Compromise or bypass connection
         device between IT and PCN              3
5.  Gain access via semi-trusted 3rd party      2
    AND
    5.1. Gain access to semi-trusted 3rd party
         network                                2
    OR
        5.1.1.  Gain physical access to semi-
                trusted 3rd party               3
        5.1.2.  Gain remote access to semi-
                trusted 3rd party               2
    5.2. Compromise protection between 3rd
         party system and PCN                   2
6.  Gain remote access via un-trusted Internet3
    AND
    6.1. Compromise connection device between
         Internet and IT                        3
    6.2. Compromise or bypass connection
         device between IT and PCN              2
```

By following the path of least resistance, we can see that the most likely successful attack is not via the Internet, but through physical access to an unsecured remote site, or the SCADA communications media. Also accessing the process control network (PCN) directly through a dialup or wireless link is a possibility. Whether an attacker can take advantage of these access methods is dependant on the other attacker goals. This illustrates one of the flexible aspects of attack trees; even the initial set of attacker goals are logically related.

4.2. Attack Goal #2: Identify MODBUS Device

After access to the SCADA systems is achieved, the next requirement for an attacker is to identify devices that may be vulnerable. This assumption is based on well-documented attacker patterns where stealthy (or not so stealthy) reconnaissance activity normally precedes most system compromises. By identifying the vulnerable MODBUS devices, the attacker can then move towards achieving the further goals outlined below in this section. The study team created the following attack tree the attack tree for identifying MODBUS devices on a SCADA system. Also note the relationships between technical (scanning) and non-technical (social engineering) attacks.

```
Attack: Identify MODBUS Device (Difficulty=2)
OR
1.  Social Engineering (e.g. pretend to be PLC
    manufacture’s service engineer)          2
2.  TCP/UDP Port Scan for Port 502           2
    AND
    2.1. Gain local PCN network access (non-
         blind)                              2
    2.2. Deploy TCP/UDP scanning tool        1
3.  MODBUS Message Scan (only against slave) 2
    AND
    3.1. Gain access to remote site or SCADA
         transmission system                 2
    3.2. Deploy MODBUS Message Scanning Tool 2
4.  Management/Application Protocol Scan     2
    AND
    4.1. Gain local PCN access (non-blind)   2
    4.2. Deploy Fingerprinting Tool          2
        OR
        4.2.1.Scan HTTP/SNMP/Telnet port for
              identifying characteristics    2
        4.2.2.Scan other identifying ports   2
5.  Sniff existing MODBUS session            2
    OR
    5.1. Sniff via compromised master        2
        AND
        5.1.1.Compromise Master (goal #11)   2
        5.1.2.Install packet capture util.   2
    5.2. Sniff via intercepted SCADA media   2
        AND
        5.2.1.Gain access to SCADA link media 2
        5.2.2.Install protocol capture tool  1
```

This analysis indicates that once an attacker has achieved access, identifying MODBUS devices would not add a significant level of difficulty. Depending on the type of access achieved, it would be simple to scan for particular MODBUS devices or find MODBUS devices through ancillary data acquisition. However, basic hardening/obfuscation techniques could increase the difficulty for an attacker or cause them to move to more obvious targets. Furthermore, detection of this type of attack is highly unlikely as few SCADA systems deploy any form of intrusion detection system and the direct impact to operations would be minimal.

4.3. Attack Goal #11: Compromise Master

The ability to compromise a master device is probably the most serious of the attacks we identified. It both provides a basis for executing many of the other goals and allows attacks on non-SCADA resources that may have trusted links the master. It also likely gives the attacker the ability to create significant change within the system. Thus this attack goal is arguably the most attractive of all goals, whether the attacker is looking to steal information, disable the SCADA system or attack other corporate assets.

```
Attack: Compromise Master     (Difficulty=2)
OR
1.  Physical Attack on Master                 3
    AND
    1.1. Gain Physical Access to the Master   3
    1.2. Determine Administrator Password     2
2.  Network Attack on Master                  2
    2.1. Gain Non-Blind Network Access        2
        OR
        2.1.1.Compromise Master O/S           2
        2.1.2.Compromise Primary HMI
              Application on Master           3
        2.1.3.Compromise Secondary Application
              on Master                       2
    2.2. Compromise Master via Slave          3
        OR
        2.2.1.Gain Physical Access to Slave   2
            AND
            2.2.1.1.Disable Real Slave Device 1
            2.2.1.2.Deploy Rogue Slave Respond
                    to MODBUS Requests from
                    Master                    2
            2.2.1.3.Corrupt Master with
                    invalid slave response    3
            2.2.1.4.Load Shell App to Master  3
        2.2.2.Gain Access to SCADA Link Media 2
            AND
            2.2.2.1.Disable Real Slave Device 2
            2.2.2.2.Deploy Rogue Slave Respond
                    to MODBUS Requests from
                    Master                    2
            2.2.2.3.Corrupt Master with
                    invalid slave response    3
            2.2.2.4.Load Shell App to Master  3
```

From this analysis, we see that there are two very different paths for an attack to take place. If access is gained to the PCN, it is likely that the master device can be compromised whether through the master device operating system or secondary applications like an embedded HTTP server. If access is made through the SCADA transmission system or a slave device it would be more difficult, but not impossible to compromise a master device. Effectiveness of any of the compromises would be based on the particular underlying technology vulnerabilities of the master device. In the case of either Windows-based or UNIX-based masters these technology vulnerabilities are well known by the hacker community and relatively easy to exploit.

5. Summary and recommendations

The results from analyzing each of the attack trees have been summarized in Table 1, showing the eleven possible attacks goals, their respective technical difficulty, severity of impact and likelihood of detection. It also lists the underlying protocol vulnerabilities that make each attack possible. Analysis of each of the trees indicated that all the avenues for attack are depend on the ability of the attacker to gain network access and identify the existing MODBUS devices. If sufficient security measures are put in place to block all possible intrusion points into the SCADA system, then the chances of a successful attack are greatly reduced. Unfortunately, in our experience the predominant security effort in most SCADA facilities tends focus on attacks via the Internet or through the business network. This leaves open attacks from other intrusion points such as remote field stations, the SCADA transmission infrastructure, trusted 3rd parties or wireless control network connections. The trees also show that once an attacker has access to the SCADA system, any moderately skilled hacker would be able to carry out the majority of the attacks.

5.1. Underlying Security Issues

Five security issues underlie each the path of least resistance in achieving each of the attack goals:

- **Lack of Confidentiality:** All MODBUS messages are transmitted in clear text across the transmission media.
- **Lack of Integrity:** There are no integrity checks built into the MODBUS application protocol, and as a result it depends on lower-

Attacker Goal	Technical Difficultly	Severity of Impact	Prob. of Detection	Underlying Critical Vulnerabilities	Comments
Gain SCADA System Access	1-3	Very Low	Low	• Wireless PCN • 3^{rd} party access • Remote field sites • SCADA transmission media	• Critical precursor for all other attack goals • Difficulty highly dependant on point of access and security measures in place
Identify MODBUS Device	2	Very Low	Low	• Lack of Confidentiality	• Critical precursor for other goals
Disrupt Master-Slave Communications	2	Moderate	High	• Lack of Authentication • Lack of Session Structure • Simplistic Framing Tech.	
Disable Slave	3	Moderate	High	• Lack of Authentication • Lack of Session Structure • Simplistic Framing Tech.	
Read Data from Slave	2	Moderate	Very Low	• Lack of Confidentiality • Lack of Authentication	
Write Data to Slave	2	High	Very Low	• Lack of Authentication • Lack of Session Structure • Lack of Integrity	
Program Slave	2	High	Low	• Possible Lack of Authentication • Lack of Session Structure • Lack of Integrity	
Compromise Slave	3	Very High	Low	• Lack of Integrity • Possible Lack of Authentication	
Disable Master	2	Moderate	High	• Lack of Authentication • Lack of Session Structure	
Write Data to Master	3	High	Low	• Lack of Authentication • Lack of Session Structure	
Compromise Master	2	Extreme	Low	• Lack of Authentication • Lack of Session Structure	• Very useful precursor to other attack goals

layer protocols to preserve integrity.

- **Lack of Authentication:** There is no authentication at any level of the MODBUS protocol, with the possible exception of some undocumented programming commands.
- **Simplistic Framing:** MODBUS/TCP frames are sent over established TCP connections. While such connections are usually reliable, they have a significant drawback for the MODBUS application: TCP does not preserve record boundaries.
- **Lack of Session Structure:** Like many request/response protocols (i.e. SNMP, HTTP, etc.) MODBUS/TCP consists of short-lived transactions where the master initiates an request to the slave that results in a single action. When combined with the lack of authentication and poor TCP initial sequence number (ISN) generation in many embedded devices, it becomes possible for attackers to inject commands with no knowledge of the existing session.

Combined, these shortcomings mean that there is limited security inherent in the SCADA system once its outside defences are breached. All successful attacks are dependent on the ability of the attacker to gain network access.

5.2. Implementing near term best practices

Based on our analysis of SCADA threats and vulnerabilities, we recommend the following steps to reduce risk of intrusion to SCADA systems:

- All external SCADA connections leaving the physical protection of the plant site (including serial links) should be considered as insecure and connections should be encrypted wherever possible.
- All gateway devices that communicate with devices outside the immediate physical protection of the plant site should be considered "bastion-hosts" and susceptible to direct attack. As such they should be hardened and isolated from other SCADA devices on the PCN.
- All connections to trusted 3rd parties should be considered as insecure. Protection through firewalls or VPNs should be deployed.
- Intrusion Detection should be deployed on the SCADA system, either through commercial IDS products, transaction logging or traffic monitoring.

While these measures will not completely remove the risk of intrusion, they will reduce it significantly.

5.3. Towards a secure MODBUS

There are a number of paths worth exploring for the development of a secure MODBUS for critical infrastructures:

- Investigate integrating the MODBUS protocol with existing security protocols that are widely deployed.
- Investigate integrating the security mechanisms into the actual MODBUS protocol itself.
- Investigate on-device and off-device implementations of the security solutions using either of the above two security mechanisms.

6. Conclusions

As noted in the analysis, all the avenues for attack are dependent on the ability of the attacker to gain network access and locate existing MODBUS devices. If sufficient security measures are in place to block every possible intrusion point, then the chance of successful attack is extremely low. Unfortunately, while protection from Internet-based intrusion is in place at most SCADA facilities, it is likely that other less obvious, but equally dangerous intrusion points are available to the attacker. Since there is virtually no security inherent in a MODBUS-based SCADA or industrial control systems, any moderately skilled hacker would be able to carry out a large variety of attacks if system access can be achieved.

The results of our study also indicate that the attack trees can be a very useful tool for modeling threats and vulnerabilities in a wide variety of systems—not just Internet or IT systems. However, the approach is not without its limitations. Lightweight approaches to threat modeling that are useful for protocol designers, vendors, and users in an area that needs more exploration. While our work was the first to apply risk metrics to a communication protocol, more formal approaches that better aggregate subordinate node values and dynamically reflect site-specific parameters (such as known vulnerabilities and deployed countermeasures) are needed. [7]

7. References

[1] Byres, E., Carter, J., Elramly, A., and Hoffman, D.; "Worlds in Collision-Ethernet and the Factory Floor", ISA 2002 Emerging Technologies Conference, Instrumentation, Systems and Automation Society, Chicago, October 2002

[2] MODBUS Application Protocol Specification V1.1, Modbus Organization, June 12, 2002

[3] Carnegie Mellon Software Engineering Institute, CERT/CC Statistics 1988-2003, http://www.cert.org/stats/

[4] Convery, S., Cook, D. and Franz, M.; An Attack Tree for the Border Gateway Protocol, http://www.io.com/~mdfranz/papers/ draft-convery-bgpattack-01.txt

[5] Schneier, B., "Attack Trees." *Dr Dobbs Journal.* December 1999. http://www.schneier.com/paper-attacktrees-ddj-ft.html

[6] Murphy, S; "BGP Security Vulnerabilities Analysis", March 2003, http://www.ietf.org/internet-drafts/draft-murphy-bgp-vuln-01.txt

[7] Franz. M; "Flexible Threat Modeling." www.io.com/~mdfranz/papers/unpub-may04-flexible-threat-modeling.pdf