A Taxonomy of Cyber Attacks on SCADA Systems

Bonnie Zhu, Anthony Joseph, Shankar Sastry
Department of Electrical Engineering and Computer Sciences
University of California at Berkeley, CA
{*bonniez,adj,sastry*}*@eecs.berkeley.edu*

***Abstract*—Supervisory Control and Data Acquisition (SCADA) systems are deeply ingrained in the fabric of critical infrastructure sectors. These computerized real-time process control systems, over geographically dispersed continuous distribution operations, are increasingly subject to serious damage and disruption by cyber means due to their standardization and connectivity to other networks. However, SCADA systems generally have little protection from the escalating cyber threats. In order to understand the potential danger and to protect SCADA systems, in this paper, we highlight their difference from standard IT systems and present a set of security property goals. Furthermore, we focus on systematically identifying and classifying likely *cyber attacks* including cyber-induced *cyber-physical attacks* on SCADA systems. Determined by the impact on control performance of SCADA systems, the attack categorization criteria highlights commonalities and important features of such attacks that define unique challenges posed to securing SCADA systems versus traditional Information Technology (IT) systems.**

***Keywords*-SCADA; Cyber-Physical Systems; Cyber Attacks;**

I. Introduction

The utilization of *Supervisory Control and Data Acquisition* (SCADA) systems facilities the management with remote access to real-time data and the channel to issue automated or operator-driven supervisory commands to remote station control devices, or *field devices*. They are the underlying control system of most critical national infrastructures including power, energy, water, transportation, telecommunication and are widely involved in the constitutions of vital enterprises such as pipelines, manufacturing plants and building climate control.

Remote locations and proprietary industrial networks used to give SCADA system a considerable degree of protection through isolation [16], [29]. Most industrial plants now employ networked process historian servers for storing process data and other possible business and process interfaces. The adoption of Ethernet and transmission control protocol/Internet protocol TCP/IP for process control networks and wireless technologies such as IEEE 802.x and Bluetooth has further reduced the isolation of SCADA networks. The connectivity and de-isolation of SCADA system is manifested in Figure 1.

This work is supported by the National Science Foundation Award CCF-0424422 for the Team for Research in Ubiquitous Secure Technology (TRUST).

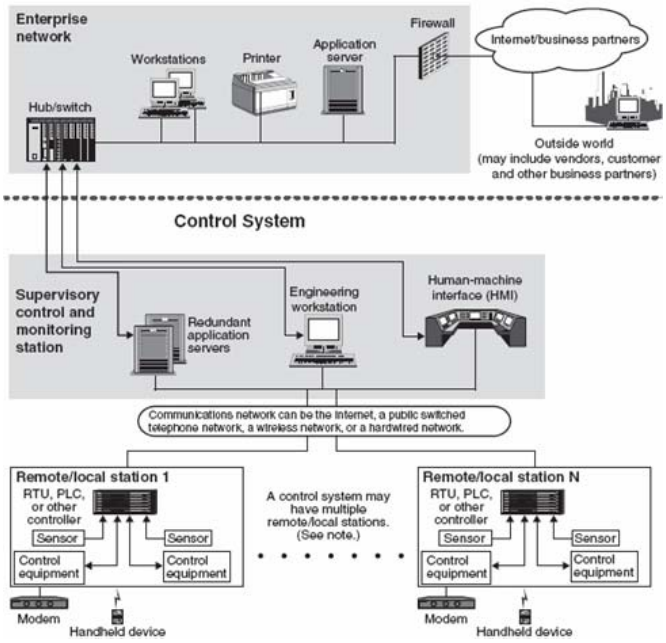

Figure 1. Typical SCADA Components Source: United States Government Accountability Office Report. GAO-04-354 [29]

Furthermore, the recent trend in standardization of software and hardware used in SCADA systems makes it even easier to mount SCADA specific attacks. Thus the security for SCADA systems can no longer rely on obscurity or on being a function of locking down a system.

These attacks can disrupt and damage critical infrastructural operations, cause major economic losses, contaminate ecological environment and even more dangerously, claim human lives.

The British Columbia Institute of Technologys Internet Engineering Lab (BCIT/IEL) maintains an industrial cyber security incident database [4] with more than 120 incidents logged since the initiation. Baker et al at McAfee in their 2011 sequel report [3] surveyed 200 IT security executives in 14 counties from critical electricity infrastructure enterprises, where SCADA systems are widely used, and found out most facilities have been under cyber attacks.

Being one of most sophisticated SCADA malware known

978-0-7695-4580-6/11 $26.00 © 2011 IEEE
DOI 10.1109/iThings/CPSCom.2011.34

to date[1], Stuxnet according to Falliere et. al at Symantec [10], takes advantage of multiple Windows zero-day vulnerabilities and targets the command-and-control software installed in industrial control systems world-wide. It sabotages facilities by reprogramming *Programmable Logic Controllers* (PLCs) to operate as the attackers intend them, most likely out of their specified boundaries while its "misreporting" feature hides the incident from the network operations center. As of April 21st. 2011, There are more than 50 new Stuxnet-like attacks beckon SCADA threats discovered [20].

Most related works have focused on the classification and categorization of attacks on standard IT systems such as [13], [14], [15], communication standards and/or protocols [17], communication devices [26]. There are work done to enumerate possible attacks on small embedded systems [11], [24]. More recently, SCADA-specific security solutions are proposed [21] and SCADA-specific *Intrusion Detection Systems* (IDS) are evaluated [31].

The remainder of this paper is organized as the follows. Section 2 compares SCADA systems with standard IT properties that attribute to their security concerns. Section 3 defines desired security properties, trust model and threat model. Section 4 states vulnerabilities that embedded in SCADA systems. Section 5,6,7 numerate cyber attacks on hardware, software, communication stacks respectively. Section 8 concludes.

II. Difference from IT

In SCADA systems, or control systems in general, the fact that any logic execution within the system has a direct impact in the physical world dictates safety to be paramount. Being on the first frontier to directly face human lives and ecological environment, the field devices in SCADA systems are deemed with no less importance than central hosts [2] [6]. Also certain operating systems and applications running on SCADA systems, which are unconventional to typical IT personnel, may not operate correctly with commercial off-the-shelf IT cyber security solutions.

Furthermore, factors like the continuous availability demand, time-criticality, constrained computation resources on edge devices, large physical base, wide interface between digital and analog signals, social acceptance including cost effectiveness and user reluctance to change, legacy issues and so on make SCADA system a peculiar security engineering task.

SCADA systems are *hard real-time systems* [25] because the completion of an operation after its deadline is considered useless and potentially can cause cascading effect in the physical world. The operational deadlines from event to system response imposes stringent constraints: missing deadline constitutes a complete failure of the system. Latency is very destructive to SCADA system's performance: the system does not react in a certain time frame would cause great loss in safety, such as damaging the surroundings or threatening human lives.

It's not the length of time frame but whether meeting the deadline or not distinguishes hard real-time system from soft real-time system In contrast, *soft real-time systems*, such as live audio-video systems, may tolerate certain latency and respond with decreased service quality, eg. dropping frames while displaying a video. Non-major violation of time constraints in soft real-time systems leads to degraded quality rather than system failure.

Furthermore due to the physical nature, tasks performed by SCADA system and the processes within each task are often needed to be interrupted and restarted. The timing aspect and task interrupts can preclude the use of conventional encryption block algorithms.

As *Real-time operating system* (RTOS), SCADA's vulnerability also rises from the fact that memory allocation is even more critical in an RTOS than in other operating systems. Many field level devices in SCADA system are embedded systems that run years without rebooting but accumulating fragmentation.

Thus, buffer overflow is more problematic in SCADA than in traditional IT.

III. Problem Statement

Before we state the security properties that are desirable for SCADA systems to achieve, we must point out that there are many trade-offs between security and control performance goals. And we will group attacks according to the hierarchy of the SCADA system.

A. Security Property Goal

Control systems have many characteristics that are different from traditional IT systems in terms of risks and operational priorities thus render unique performance and reliability requirements besides the use of operating systems and applications being unconventional to typical IT personnel.

Even where security is well defined, the primary goal in the Internet is to protect the central server and not the edge client. In process control, an edge device, such as PLC or smart drive controller, is not necessarily merited less importance than a central host such as data historian server [6], as they are on the first frontier facing human lives and ecological environment.

These differences between SCADA systems and IT systems demand an adjusted set of security property goals and thus security and operational strategies.

[1]In McAfee's report [3], nearly half of those being surveyed in the electric industry said that they had found Stuxnet on their systems.

[2]Although arguably, a compromised central serverl/controller may cause server harm if the field devices don't have their own individual and local protection.

In the traditional IT community, the set of common desirable security properties are *confidentiality*, *integrity* and *availability*, or *CIA* in short. The paramount, in IT's world is confidentiality and integrity while in control systems is system availability and data integrity as result of human and plant safety being its primary responsibility.

Particularly, most of computer security research focus on confidentiality. To be SCADA system specific, we prioritize security properties of SCADA systems in the order of its importance and desirability in industry, especially in control engineering sector. The modification we make addresses the special needs incurred from the unique characteristics of SCADA systems, namely the time criticality, dispersed distributed-ness and continuous availability.

There are different versions of definition and use of security properties [2] with slight variations. However, in light to differentiate the uniqueness of control systems from standard IT systems, it's necessary for us to stress and explain some more relevant subtleties. Nevertheless, it's not to say that these properties we want to highlight are mutual exclusive, absent of over-lapping.

1) Timeliness: explicitly expresses the time-criticality of control systems, a given resulted from being real-time system, and the concurrencies in SCADA systems due to being widely dispersed distributed systems.

It includes both the *responsiveness* aspect of the system, e.g. a command from controller to actuator should be executed in real-time by the latter, and the timeliness of any related data being delivered in its designated time period, by which, we also mean the *freshness* of data, i.e., the data is only valid in its designated time period. Or in a more general sense, this property describes that any queried, reported, issued and disseminated information shall not be stale but corresponding to the real-time and the system is able and sensitive enough to process request, which may be of normal or of legitimate human intervention in a timely fashion, such as within a sampling period. In reality, if arrives late or repeatedly to the specified node, a message is no longer any good, be it a correct command to an actuator or a perfect measurement from a sensor with intact content. As a matter of fact, any replay of data easily breaches this security goal.

Moreover, this property also implicitly implies the order of updates among peered sensors, especially if they are observing the same process or correlated processes. The order of data arrival at *central monitor room* may play an important factor in the representation of process dynamics and affect the correct decision making of either the controlling algorithms or the supervising human operators.

In a nutshell, all right data should be processed in *right* time, which unfolds an underpinning security goal – *secure time* provision.

2) Availability : means when any component of a SCADA system, may it be a sensory or servomechanical device, communication or networking equipment, or radio channel; computation resource and information such as sensor readings and controller commands etc. that transmits or resides within the system should be ready for use when is needed. Most of SCADA controlled processes are continuous in nature. Unexpected outages of systems that control industrial processes are not acceptable. This desired property for both SCADA systems control performance and security goal requires that the security mechanism employed onto SCADA systems, including but not limited to the overall cryptographic system, shall not degrade the maintainability, operability , and its accessibility at emergency, of the original SCADA system without those security oriented add-ons.

3) Integrity: requires data generated, transmitted, displayed, stored within a SCADA system being genuine and intact without unauthorized intervention, including both its content, which may also include the header for its source, destination and time information besides the payload itself. A very related terminology is *authenticity*, in the content of SCADA system, it implies that the identity of sender and receiver of any information shall be genuine. Using our definition of integrity, then authenticity falls within the same category. One can image how disastrous the consequence can be, if a control command is redirected to an actuator other than its intended receiver or fake or wrong source information of a sensor measurement being reported to the central controller. The *intra-message integrity* means specifically the content of message to be genuine and *inter-message integrity* refers to assure data integrity, the protocol must prevent an adversary from constructing unauthentic messages, modifying messages that are in transit, reordering messages, replaying old messages, or destroying messages without detection.

4) Confidentiality: refers to that unauthorized person should not have any access to information related to the specific SCADA system. At current stage, this need is dwarfed by the desirability of availability in a control performance-centric setting. SCADA systems measure and control physical processes that generally are of a continuous nature with commands and responses are simple and repetitive. Thus the messages in SCADA systems are relatively easy to predict. Hence confidentiality is secondary in importance to data integrity.

However, the confidentiality of critical information such as passwords, encryption keys, detailed system layout map and etc. shall rank high when it comes to security concerns in industry. Applicable reinforcement should be imposed in this aspect. Also, the information regarding physical content flowed within the control algorithm may be subject to leaking critical message to side channel attacks.

The drastic difference in the ordering of desired security properties is mostly due to that SCADA systems are demanded to be real-time operating and continuously functioning.

5) Graceful Degradation: requires the system being capable of keeping the attack impact local and withholding tinted data flow within *tinted* region without further escalating into a full scale, full system cascading event.

Again, all these desired security properties are not mutual exclusive but closely related. For example, by breaching integrity, an adversary can change control signals to cause a device malfunction which might ultimately affect the availability of the network. Overall, a tightly enforced **access control** may render confidentiality, integrity, availability , timeliness and graceful degradation as well.

B. Trust Model

Given that we focus on the cyber attacks on SCADA system, we restrain our attention to attacks mounted through cyber means [3] and assume the basic physical security is provided. Particularly, the *SCADA server* or *Master Terminal Unit* is physically secure, i.e., we assume there are no direct physical tempering on the server where the main control and estimation algorithms reside. Brute force physical sabotage such as cutting wires and cables from communication and power supply or hammering devices or radio jamming are out the scope of this paper.

Furthermore, we assume that the control and estimation algorithms are programmed securely.

C. Threat Model

Typical threats to sensor networks and to conventional IT systems are also threats to SCADA systems if the adversarial have means to exploit the vulnerabilities of SCADA systems[4]. The adversarial sources include but not limited to hostile governments, terrorist groups, foreign intelligence services, industrial spies, criminal groups, disgruntled employees, bot-network operators, phishers, spywaremalware authors, spammers, and attackers [30]. We assume attacks come from one side of SCADA center only and there's no collusion.

IV. Vulnerability

The current common practice of SCADA system leaves window open to various vulnerabilities. To name a few, the entrenched factors are not limited to public information likw a company's network infrastructure, insecure network architecture, operating system vulnerabilities enabled trap doors to unauthorized users and the use of wireless devices. In particular, the lack of real-time monitoring and proper encryption is very detrimental.

[3]As stated in previous sections, these cyber attacks are most likely resulted in physical destruction in SCADA systems.

[4] Note we are making a rather conservative assumption in light of exploring the potentials of cyber security issues in the SCADA system domain. Any further suitable and refined threat model depends on the cost effectiveness of the security measures.

Cyber attacks on SCADA system can take routes through Internet connections, business or enterprise network connections and or connections to other networks, to the layer of control networks then down the level of field devices. More specifically, the common attack vectors are

- Backdoors and holes in network perimeter
- Vulnerabilities in common protocols
- Attacks on field devices through cyber means
- Database attacks
- Communications hijacking and *Man-in-the-middle* attacks
- *Cinderella* attack on time provision and synchronization

From the point view of a control engineer, possible attacks can be grouped into following categories

- bogus input data to the controller introduced by compromised sensors and/or exploited network link between the controller and the sensors
- manipulated and misleading output data to the actuators/reactors from the controller due to tempered actors/ reactors or compromised network link between the controller and the actuators
- controller historian
- Denial of Service – missing the deadlines of needed task actions.

There is still little reported information about actual SCADA attacks nor scenarios designed by red-teams, despite the growing awareness of security issues in industrial networks. However, by leveraging the existing solution and understanding of the conventional IT system, we use the SCADA hierarchy as a reference plane. Then the classification of cyber attacks can fall into the following categories.

V. Cyber Attacks on Hardware

Attacker might gain unauthenticated remote access to devices and change their data set points. This can cause devices to fail at a very low threshold value or an alarm not to go off when it should. Another possibility is that the attacker, after gaining unauthenticated access, could change the operator display values so that when an alarm actually goes off, the human operator is unaware of it. This could delay the human response to an emergency which might adversely affect the safety of people in the vicinity of the plant. Some of the detailed procedure of achieve such attacks are given out in later section when we describe specific SCADA protocols.

The main issue in preventing cyber attacks on hardware is access control. With that in mind, we should mention one of the representative attacks in this category, namely the doorknob-rattling attack. The adversary performs a very few common username and password combinations on serval computers that results in very few failed login attempts. This attack can go undetected unless the data related to login

failures from all the hosts are collected and aggregated to check for doorknob-rattling from any remote destination.

VI. Attacks on Software

As listed in earlier sections, SCADA system employs a variety of software to meet its functionality demands. Also there are large databases reside in data historians besides many relational database applications used in cooperate and plant sessions.

Hosting centralized database , data historians contain vital and potentially confidential process information. These data are not only indispensable for technical reasons, such as that many control algorithms rely on past process data to make correct decisions, but also for business purposes, such as electricity pricing.

Although we've assumed the algorithms of these softwares are trustworthy, there are still vulnerabilities associated with their implementations. The most common implementation flaw is buffer overflow among others such as format string, integer overflow and etc. The fact that most control applications are written in C requires us to take extra precaution with this vulnerability.

A. No Privilege Separation in Embedded Operating System

VxWorks was the most popular embedded operating system in 2005 and claimed 300 million devices in 2006 [23], which is a platform developed by Wind River Systems and has since been acquired by Intel [19]. VxWorks has been used to power everything from the Apple Airport Extreme access points to the Mars rovers and the C-130 Hercules aircraft [18]. VxWorks itself is essentially a monolithic kernel with applications implemented as kernel tasks, This means that all tasks generally run with the highest privileges and there is little memory protection between these tasks.

B. Buffer Overflow

Many attacks boil down to cause buffer overflow as their eventual means to corrupt the intended behavior of the program and cause it to run amok. Some general methods are stack smashing and manipulating function pointer.

The effect of such attacks can take forms such as resetting passwords, modifying content, running malicious code and so on.

The buffer overflow problem in SCADA system takes two fronts. One front is on the workstations and servers which are similar to standard IT systems.

For example, WellinTech KingView 6.53 HistorySvr, an industrial automation software for historian sever widely used in China, has a heap buffer overflow vulnerability that could potentially become the risk of a Stuxnet type mishap if not matched [5] .

The other front manifests itself in field devices and other components that rely on RTOS thereof inherent the susceptible memory challenge. Exploits can take advantage of the fixed memory allocation time requirement in RTOS system to have more successful launchings. Let alone that many field devices run for years without rebooting. Therefore, these SCADA components, especially in legacy networks, are subject to accumulated memory fragmentation, which leads to program stall.

The Hardware/Software Address Protection (HSAP) technique offered by [28] including hardware boundary check method and function pointer XOR method to deal with stack smashing attack and function pointer attack in embedded systems , respectively.

C. SQL Injection

Most small and industrial- strength database applications can be accessed using Structured Query Language (SQL) statements for structural modification and content manipulation. In light of data historians and web accessibility in current SCADA systems, SQL injection, one of the top Web attacks, has a very strong implication on the security of SCADA system.

The typical unit of execution of SQL which comes in many dialects loosely based around SQL-92 ANSI standard is *query*, which is a collection of statements that typically return a single *result set*. SQL injection occurs when an adversary is able to manipulate data input into an Web application, which fails properly sanitize user-supplied input, and to insert a series of unexpected SQL statements into a query. Thus it is possible to manipulate a database in several unanticipated ways. Moreover, if a "command shell" store procedure is enabled, an attacker can move further to prompt level. The process will run with the same permissions as the component that executed the command. The impact of this attack can allow attackers to gain total control of the database or even execute commands on the system.

In the case studied in [22], where the store procedure in SQL server (shown in Fig.2) is enabled by default. Thus an attacker still can get into SCADA system even though two LAN cards are installed.

Intentionally malicious changes to databases can cause catastrophic damage.

VII. Attacks on the Communication Stack

We break down the attacks on the communication stack by using the TCP/IP or the Internet reference model and highlight some of those may have more potentials in harming SCADA systems, in particular on *network layer, transport layer, application layer* and the *implementation of protocols*.

The UDP back door on port $0x4321$ on thousands of devices is known in the public since at least spring 2002.

There are many well-known TCP/IP attacks in literature, readers please refer to [14], [13] for more details.

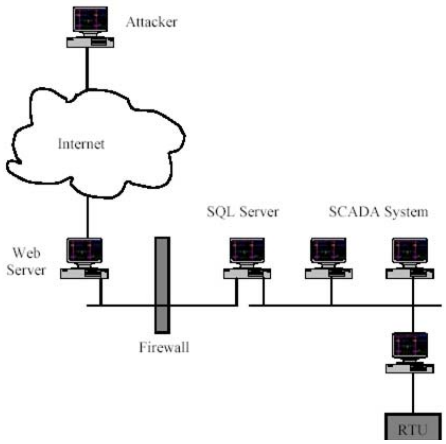

Figure 2. SQL Attack

A. Network Layer

1) Diagnostic Server Attacks through UDP port: Adversaries have access to the same debugging tools that any RTOS developers do. They can read symbol tables, step through the assembly, etc., considering also that many attackers don't even need code-level knowledge. For example Wind River Systems VxWorks weak default hashing algorithm in standard authentication API for VxWorks is susceptible to collisions, an attacker can brute force a password by guessing a string that produces the same hash as a legitimate password [5]. Or through VxWorks debug service runs UDP on port 17185, which is enabled by default, an attacker can execute the following attacks without any authentication required while maintaining a certain level of stealthiness such as remote memory dump, remote memory patch, remote calls to functions, remote task management [6].

The VxWorks Wind DeBug (WDB) is an RPC-based protocol which uses UDP can explored over the Internet by downloading hacking software and adding targets to a host list before running the script.

2) Idle Scan: is to blind port scan by bouncing off a dumb "zombie" host, often a preparation for attack. Both MODBUS and DNP3 have scan functionalities prone to such attacks when they are encapsulated for running over TCP/IP.

3) Smurf : is a type of address spoofing, in general, by sending a continuous stream of modified *Internet Control message Protocol*(ICMP) packets to the target network with the sending address is identical to one of the target computer addresses. In the context of SCADA systems, if an PLC acts on the modified message, it may either crash or dangerously send out wrong commands to actuators.

4) Address Resolution Protocol (ARP) Spoofing/Poisoning: The ARP is primarily used to translate IP addresses to Ethernet Medium Access Control (MAC) addresses and to discover other connected interfaced device on the LAN. The ARP spoofing attack is to modify the cached address pair information.

By sending fake ARP messages which contain false MAC addresses in SCADA systems, an adversary can confuse network devices, such as network switches. When these frames are falsefully sent to another node, packets can be sniffed; or to an unreachable host, DoS is launched; or intentionally to an host connected to different **actuators**, then *physical disasters* of different scales are initiated.

Static MAC address is one of the counter measures. However, certain network switches do not allow static setting for a pair of MAC and IP address. Segmentation of the network may also be a method to alleviate the problem in that such attacks can only take place within same subnet.

5) Chain/Loop Attack: In a chain attack, there is a chain of connection through many nodes as the adversary moves across multiple nodes to hide his origin and identity. In case of a loop attack, the chain of connections is in a loop make it even harder to track down his origin in a wide SCADA system.

B. Transport Layer

SYN flood is to saturate resources by sending TCP connection requests faster than a machine can process.

SCADA protocols, particularly those running over top of transport protocols such as TCP/IP have vulnerabilities that could be exploited by attacker through methodologies as simple as injecting malformed packets to cause the receiving device to respond or communicate in inappropriate ways and result in the operator losing complete view or control of the control device.

C. Application Layer

Currently, there is no strong security control in protocols used in SCADA systems, such as DNP3 without secure authentication, Modbus,*Object Linking and Embedding (OLE) for Process Control* (OPC), *Inter-Control Center Communications Protocol* (ICCP). Practically there is no authentication on source and data such that for those who have access to a device through a SCADA protocol, they can often read and write as well. The write access and diagnostic functions of these protocols are particular vulnerable to cyber and cyber induced physical attacks.

One of possible attacks in both SCADA and conventional IT systems is *DNS forgery*. Such attack is to send a fake DNS reply with a matching source IP, destination port, request ID, but with an attacker manipulated information inside, so that this fake reply may be processed by the client before the real reply is received from the real DNS server. For more details on those attacks studied in conventional IT systems, please refer to [13].

Next, we list potential attacks associated with more SCADA specific protocols.

[5]US-Cert VU #840249.
[6]US-Cert VU #362332

1) MODBUS: Modbus [27] is a *de facto* standard of application layer protocol used in industrial networks. It comes with different flavors from plain Modbus to Modbus+ to Modbus/TCP. A Modbus client (or master) can send a request to a Modbus server (or slave)[7] with a *function code* that specifies the action to be taken and a *data field* that provides the additional information. The general Modbus frame is shown in Figure (3).

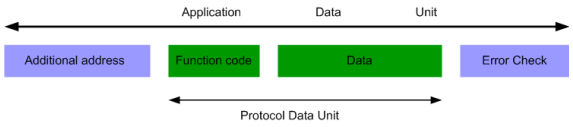

Figure 3. A typical Modbus frame

Among currently little published accounts on attacks against Modbus, Digital Bond [8] has conducted intrusion detection work on studying its potential weakness. Their detection rules include denial of service (e.g., rebooting Modbus servers, configuring them to provide no service-called listen-only mode, and crashing servers with a large size request), reconnaissance (e.g., unauthorized reading of data, and gathering device information), and unauthorized write requests.

Byres and his company have used Achilles Vulnerability Test Platform to perform security tests on Modbus to discover vulnerabilities [6], [7] .

Given that Modbus does not have encryption or any other security measures, there are many ways to directly explore such weakness on the function code level [12]. The function codes 0x05 and 0x0F are used to write a single or multiple outputs (coils) to either ON or OFF in a remote device, respectively. This means that an adversary can turn off and suppress output(s) remotely thus to create a false sense of situation at the HMI end. Unauthorized writes can be accomplished through using function codes 0x06 and 0x10. Accordingly, the forged data may be written to either a single or multiple registers in a remote device. If Modbus is implemented on serial line, function code 0x11 can be used to gather information from a remote device, such as a controller's description. Function code 0x08 is used for diagnostics on serial line. However, combined with subfunction code 0x01, it can initialize and restart the slave (server) port and clear out the communication event counter, which is a ideal attack vector. When combined with subfunction code 0x04, the diagnostics function code can force a remote device into its Listen Only Mode. Similarly, Modbus+ has a function code (08) for log cleaning that can enable an attacker to clear stats of data manipulation and denial of service events.

[7]Initially, Modbus was a master-slave protocol for serial buses. When implementing Modbus over TCP, a Modbus master is a TCP client, and a Modbus slave is a TCP server.

2) DNP3: DNP3 is used between master control stations and remote computers or controllers called *outstations* for the electric utility industry and water companies. DNP3 is implemented by several manufacturers due to its small memory consumption. Its function code 0x0D can reset and reconfigure DNP3 outstations by forcing them to perform complete power cycle. During the re-initialization to default values, many devices clear all queues as well. An attacker can take advantage of this property to cause delay in outstations before they accept requests again. Furthermore, function code 0x13 enable loading new outstation configurations. With unauthorized access, an attacker can manipulate the remote devices with manipulated setting values, suppress output and or create false alarms.

D. Attacks on Implementation of Protocols

Protocol vulnerabilities can reveal themselves as segmentation faults, stack, heap or buffer overflows, etc., all of which can cause the protocol implementation to fail resulting in a potential exploit.

Meanwhile, certain protocol implementations, such as ICCP servers, only allow users to read values, and there are a number of protocols that are in the process of adding security controls to address this deficiency.

Nevertheless, [8] argues that SCADA implementation vulnerabilities are more important than lack of security controls in SCADA protocols.

1) TCP/IP : First of all, in light of the migration to Windows from UNIX in operating system used by many sectors in SCADA systems, there are several attacks specifically exploit the implementation of TCP/IP protocols in Windows. Although there are patches available, restrained to be on-line continuously, it's very likely that these machines do not have up-to-dated patches. Here, we only name a few well known ones.

- WinNuke takes advantage of the absence of status flag URG in handling the TCP protocol.
- TearDrop/NearTear and Ssping utilize implementation error of fragmentation handling in TCP/IP protocol.

A nightmare scenario can be that one company's network is compromised and a polymorphic worm takes down most servers and any unpatched SCADA servers running Windows.

Secondly, these protocol stacks can and do suffer from various vulnerabilities commonly found due to poor software design and coding practices.

2) OPC: OPC servers use Microsoft's OLE technology[8] to provide real-time information exchange between software applications and process hardware.

At the OPC interface level, the item write function takes two parameters: an item handle and a value to write to it. If the server maps handles to memory addresses and

[8]Also known as the Component Object Model, or COM

fails to validate a client-provided handle, the IO interfaces write function allows an attacker to write any value to any memory address, a primitive which can be easily exploited to run arbitrary code on the server (e.g. through stack return addresses). It is an even larger issue that an OPC server can be remotely compromised and used to launch attacks on other systems. Because OPC servers are often exposed in the Demilitarized Zone (DMZ), this could be a communication chain that could allow control system exploitation from the enterprise network or Internet.

[9] gives three possible OPC attack scenarios, of which are all associated with extra open ports:

- Collateral Damage by OPC-Unaware Malware;
- Opportunistic OPC Denial of Service Attack;
- Intelligent, aggressive attack against OPC hosts through a man-in-the-middle (MITM) technique

3) ICCP: The most serious and exposed SCADA protocol stacks are those that are used to exchange information with business partners, such as ICCP, or those used to exchange information between the corporate network and control center network.

According to the LiveData ICCP Server white paper [1], LiveData ICCP server contains a heap-based buffer overflow. The LiveData implementation of ISO Transport Service over TCP (RFC 1006) is vulnerable to a heap-based buffer overflow. By sending a specially crafted packet to a vulnerable LiveData RFC 1006 implementation, a remote attacker may be able to trigger the overflow to execute arbitrary code or crash a LiveData ICCP Server to cause a denial of service.

4) UCA: UCA was expected to be more robust standard than DNP3 when the Electric Power Research Institute (EPRI) decided to use it to serve the SCADA needs of the electric utilities. It's based on the Manufacturing Message Specification from ISO standard 9506.

5) MMS: Tamarack MMS*d* is an implementation of *Manufacturing Message Specification* (MMS) protocol, an international standard (ISO 9506), dealing with messaging system for transferring real time process data and supervisory control information between networked field devices and/or computer applications.

Tamarack MMS*d*[9] components do not properly handle malformed RFC 1006 packets either. This vulnerability may allow a remote, unauthenticated attacker to cause a denial of service condition.

VIII. Conclusion and Future Work

The cyber-physical security of real-time, continuous systems necessitates a comprehensive view and holistic understanding of network security, control theory and the physical system. Ultimately, any viable technical solutions and research directions in securing SCADA systems must lie in the conjunction of computer security, communication network and control engineering. The idea of looking into the problem in the context of control performance holds its solid bearings. However, the very large installed base of such systems means that in many instances we must for a long time to come rely on retrofitted security mechanisms, rather than having the option to design them in from scratch. This leads to a pressing need for robust SCADA-specific intrusion detection systems (IDS) and resilient control.

Our next step is to categorize the attacks in terms of their manifestation and realization in order to shed more light into intrusion prevention and detection.

[9] Vulnerability Note VU#372878

Acknowledgment

The first author gratefully acknowledges Vern Paxson for sharing his insight and expertise with us on intrusion detection and network security besides his review and feedback on the first draft. Special thanks also go to numerous domain experts with whom we have had many stimulating discussions and to our anonymous reviewers for their meaningful comments.

References

[1] *Vulnerability Note VU#190617* LiveData ICCP Server heap buffer overflow vulnerability, http://www.kb.cert.org/vuls/id/190617

[2] Ross Anderson, *Security Engineering – A Guide to Building Dependable Distributed Systems* , 2001, Wiley. ISBN 0-471-38922-6.

[3] Stewart Baker,Natalia Filipiak,Katrina Timlin, *In the Dark Crucial Industries Confront Cyberattacks*, 2011, Mcafee report, http://www.mcafee.com/us/resources/reports/rp-critical-infrastructure-protection.pdf

[4] BCIT Industrial Security Incident Database (ISID) http://www.bcit.ca/appliedresearch/security/services.shtml

[5] Dillon Beresford, *The sauce of utter pwnage*, January 2011 http://thesauceofutterpwnage.blogspot.com/

[6] Eric Byres, Joel Carter, Amr Elramly, Dan Hoffman *Worlds in Collision: Ethernet on the Plant Floor*, ISA Emerging Technologies Conference, Instrumentation Systems and Automation Society, Chicago, October (2002).

[7] Eric Byres, Dan Hoffman, Nate Kube *On Shaky Ground - A Study of Security Vulnerabilities in Control Protocols* http://byressecurity.com/assets/pdf/On%20Shaky%20Ground%20-%20NPIC_HMIT_2006%20Paper.pdf

[8] Digital Bond: Securing The Critical Infrastructrue http://www.digitalbond.com/

[9] Byres Research,British Columbia Institute of Technology, *OPC Security Whitepaper #2 OPC Exposed* May 1, 2007

[10] Nicolas Falliere, Liam O Murchu, and Eric Chien, *W32. Stuxnet Dossier* , Symantec Security Response, Version 1.4 , February 2011, http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf

[11] Octavio Nieto-Taladriz Garcia *Security in Embedded Systems Challenges and Oportunities*, International Conference on Emerging Security Information, Systems and Technologies, Securware 2007

[12] Mark Grimes *SCADA Exposed* http://www.toorcon.org/2005/slides/mgrimes/mgrimes-scadaexposed.pdf

[13] Simon Hansman, Ray Hunt, *A taxonomy of network and computer attacks*, Computers & Security,DTD5, 2004

[14] John D. Howard, *An Analysis Of Security Incidents On The Internet 1989 - 1995*, dissertation, Carnegie Mellon University, April 1997

[15] Kevin S. Killourhy, Roy A. Maxion and Kymie M. C. Tan,*A Defense-Centric Taxonomy Based on Attack Manifestations*, Proceedings of International Conference on Dependable Systems & Networks: Florence, Italy, 28 June - 01 July 2004

[16] Ronald L. Krutz, *Securing SCADA systems* ,Wiley, 2006.

[17] Daniel Lough, *A Taxonomy of Computer Attacks with Applications to Wireless Networks*, Ph.D Thesis, Virginia Polytechnic Institute and State University, 2001

[18] Metasploit Blog, August, 2010 http://blog.metasploit.com/2010/08/vxworks-vulnerabilities.html

[19] HD Moore, *Fun with VxWorks*, http://dev.metasploit.com/data/confs/bsideslv2010/FunWithVxWorks.pdf

[20] Phil Muncaster, *Stuxnet-like attacks beckon as 50 new Scada threats discovered* 21st Apr., 2011, http://www.v3.co.uk/v3-uk/news/2045556/stuxnet-attacks-beckon-scada-threats-discovered

[21] Igor Nai Fovino, Alessio Coletta, Marcelo Masera, *Taxonomy of security solutions for the SCADA Sector*,Deliverable: D 2.2, Version: 1.1, A European Network For The Security Of Control And Real Time Systems, March, 2010 http://www.cen.eu/cen/Sectors/Sectors/ISSS/Focus/Documents/D22.pdf

[22] T. Paukatong,*SCADA Security: A New Concerning Issue of an In-house EGAT-SCADA* 2005 IEEE/PES Transmission and Distribution Conference & Exhibition: Asia and Pacific Dalian, China

[23] P.J. Pingree, *The Deep Impact Test Benches &# 8211; Two Spacecraft, Twice the Fun*, Proceedings of IEEE Aerospace Conference, Page 1–9, 2006

[24] Srivaths Ravi, Anand Raghunathan, Paul Kocher and Sunil Hattangady *Security in embedded systems: Design challenges*, ACM Transactions on Embedded Computing Systems (TECS), vol.3, no.3, pages 461–491,2004.

[25] Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, *Operating System Concepts*, 7th edition, Wiley & Sons, 2005

[26] MO Chun Man and Victor K. Wei, *A taxonomy for attacks on mobile agent*; EUROCON'2001, Trends in Communications, International Conference on. Volume 2, 4-7 July 2001 Page(s):385 - 388 vol.2

[27] Modbus IDA. *Modbus application protocol specification* v1.1a, June 4, 2004.

[28] Zili Shao, Qingfeng Zhuge, Yi He, Edwin H.-M. Sha, *Defending Embedded Systems Against Buffer Overflow via Hardware/Software*,Proceedings of the 19th Annual Computer Security Applications Conference (ACSAC 2003)

[29] United States Government Accountability Office, *Critical Infrastructure Protection Challenges and Efforts to Secure Control Systems*, Report to Congressional Requesters, March 2004,http://www.gao.gov/new.items/d04354.pdf.

[30] United Stateds Government Accountability Office (GAO), *Department of Homeland Securitys (DHSs) Role in Critical Infrastructure Protection (CIP) Cybersecurity*, GAO-05-434 (Washington, D.C.: May, 2005).

[31] Bonnie Zhu and Shankar Sastry, *SCADA-specific Intrusion Detection/Prevention Systems: A Survey and Taxonomy* In proceedings the First Workshop on Secure Control Systems (SCS'10), Stockholm, Sweden, 2010.