INL/CON-06-01133
PREPRINT

Quantitative Cyber Risk Reduction Estimation Methodology For A Small SCADA Control System

Hawaii International Conference On System Science

Miles A. McQueen
Wayne F. Boyer
Mark A. Flynn
George A. Beitel

January 2006

The INL is a
U.S. Department of Energy
National Laboratory
operated by
Battelle Energy Alliance

Quantitative Cyber Risk Reduction Estimation Methodology for a Small SCADA Control System

Miles A. McQueen[1], Wayne F. Boyer[1], Mark A. Flynn[1], George A. Beitel[1],
[1] *Idaho National Laboratory, 2525 N. Fremont,*
Idaho Falls, Idaho, U.S.A. 83415
{miles.mcqueen, wayne.boyer, mark.flynn, george.beitel}@inl.gov

Abstract

We propose a new methodology for obtaining a quantitative measurement of the risk reduction achieved when a control system is modified with the intent to improve cyber security defense against external attackers. The proposed methodology employs a directed graph called a compromise graph, where the nodes represent stages of a potential attack and the edges represent the expected time-to-compromise for differing attacker skill levels. Time-to-compromise is modeled as a function of known vulnerabilities and attacker skill level. The methodology was used to calculate risk reduction estimates for a specific SCADA system and for a specific set of control system security remedial actions. Despite an 86% reduction in the total number of vulnerabilities, the estimated time-to-compromise was increased only by about 3 to 30% depending on target and attacker skill level.

1. Introduction

Control systems connected to public networks are at risk from cyber attack. Operators of these control systems need a measure of the risk associated with potential attacks if they are to effectively manage their resources. Cyber security evaluations are traditionally qualitative in nature such that recommendations are given for remedial actions with no quantitative measure of how the recommended actions reduce the risk of a successful attack.

In April 2005 our risk analysis team was asked to perform a quantitative estimate of the risk reduction on a partial Supervisory Control and Data Acquisition (SCADA) system referred to as CS60. The baseline system had already undergone a security review, been modified to enhance security, and then been retested; consequently, there was no opportunity to devise or execute any tests tailored for risk reduction estimation needs. The analysis was undertaken with the following goals in mind:

- The first goal was to find or create a suitable methodology for a quantitative risk reduction estimation of a specific SCADA control system. It needed to produce quantified estimates, believable to the customer and ourselves that were superior to the general qualitative assessments already made. It also needed to be defined well enough to provide a framework for discussion and improvement with a variety of personnel.
- The second goal was to apply the agreed-upon methodology to the CS60 system and assess the risk reduction accomplished by the security enhancements that had been applied as a result of the first security review.
- The third goal was to show that analysts and hackers could be integrated in such a way that the methodology could be executed efficiently and with an appropriate amount of give and take between group members with vastly different skills and interests.

2. Related Work

Researchers are testing the viability of different approaches for dealing with control system cyber security. Carlson et al. [3] describes a novel approach for applying Hidden Markov Models to an attack/defend scenario on an infrastructure system. The approach, based on sound statistical models, is flexible, but requires both detailed information about the system and significant set-up time. Madan et al. [6] apply a stochastic model to computer network system. It is used to determine steady-state availability of QoS attributes and also mean times to security

0-7695-2507-5/06/$20.00 (C) 2006 IEEE

failures based on probabilities of failure due to violations of different security attributes. The theory used is classic statistical stochastic modeling. Employing this type of model requires knowledge of the system in detail. Furthermore, Haimes [5] applied Hierarchical Holographic Models, event trees, and fault trees to a variety of applications, both models require specific details, are not dynamic, and rely on expert opinion.

Major [7] doesn't directly address cyber security, but instead, develops a simplified model for quantifying the risk to facilities from terrorist attacks. The focus of the work is on physical attacks. Initially the model calculates the probability of an attack escaping detection and the probability of the attack being successfully executed if not detected. These two probabilities are hypothesized to be dependent on the value of the facility, and both the attackers and defenders resources. The model then makes use of game theory to determine the probability of attack on a facility, which based on simplifying assumptions, is proportional to the square root of the facilities value. A mapping of the proposed physical model quantitative estimation to cyber security was not proposed and isn't obvious.

Taylor et al. [12] provide an interesting cyber security assessment process that combines techniques from Survivability System Analysis and Probability Risk Assessment. The proposed process has some significant advantages, but seems more suitable to complete and operational systems so that costs, attack scenarios, and critical system objectives may be fully explored. Further, the process is dependent on multiple iterations of expert elicitation, which are not available in many situations.

Dacier et al. [4] suggested the use of 'privilege graphs' to analyze security. Privilege graphs require modeling of vulnerabilities at a very low level, and, for a nontrivial sized system, would involve a graph of unmanageable size. Privilege graphs are transformed into Markov chains. But the assumptions underlying Markov chains are not necessarily applicable to an intelligent adversary.

Sheyner et al. [11] describe an automated technique for generating and analyzing attack graphs. They use a model checker as the core engine to comprehensively generate every attack path sequence that could lead to an undesired system state. They have developed a suite of tools to aid the process and suggested a technique for weighting transitions of the attack graph as part of a post processing analyzer. The tool suite was not immediately available for use. There is a question of scalability in using a model checker to generate the attack paths, and the level of attack and vulnerability abstraction may be at a lower level than optimal for an estimate of risk reduction.

Byres et al. [2] describe how the attack tree methodology may be applied to the common SCADA protocol MODBUS/TCP with the goal of identifying security vulnerabilities inherent in the specification and in typical deployments. Attack trees are a promising technology for aiding control system security analysts in the understanding and protection of their systems.

While a number of the above methods and techniques seem promising and merit future research, none could provide a quantitative measure of risk reduction for our case study.

3. Methodology Description

The proposed methodology is based on the assumption that risk is related to the elapsed time required for a successful attack. This methodology is briefly described by the following steps:

Step 1. Establish the system configuration.

Step 2. Identify applicable portions of the quantitative risk model.

Step 3. Identify and prioritize the security requirements of the primary target(s).

Step 4. Identify system vulnerabilities.

Step 5. Categorize vulnerabilities on each device by compromise type.

Step 6. Estimate time-to-compromise each device.

Step 7. Generate compromise graph(s) and attack paths.

Step 8. Estimate dominant attack path(s).

Step 9. Do Steps 3–8 for baseline and enhanced system.

Step 10. Estimate risk reduction.

The remainder of this section describes each step in more detail.

3.1. Step 1. Establish the system configuration

A control system typically consists of many interconnected computers customized and configured

to meet the requirements of a particular control process. The security of the system is highly dependent on the system configuration. The configuration determines the system perimeter and the potential attack targets. A primary target is defined as any control system device that can directly trigger a physical event, and a perimeter device as any device that is both part of the control system and can be directly reached with no routing, switching, forwarding, or inspection. A primary goal of system configuration analysis is the identification of primary targets and perimeter devices.

3.2. Step 2. Identify applicable portions of the quantitative risk model

The quantitative risk model used to define risk as the product of probability and consequence is

$R_e = P_e * C_e$

where R_e is the risk of an unwanted event, P_e is the probability of occurrence of the event, and C_e is the value of the consequence of the event typically measured in dollars.

3.2.1. Previous Decompositions of P_e. A variety of decompositions of P_e have been suggested in the literature [1,7,9].

Rinaldi [9] developed a partitioning of P_e that sets $P_e = P_A * (1\text{-}P_E)$, where P_A is the probability of an attack and $(1\text{-}P_E)$ is the probability of adversary success. Rinaldi further decomposes $(1\text{-}P_E)$ into the probability of the attack being interrupted and the probability of the attacker success being neutralized. Unfortunately, the meaning of the terms in this decomposition of $(1\text{-}P_E)$ is not clear, making it difficult to assess how one would measure or estimate their values.

Beitel et al. [1] set the initial decomposition identically to Rinaldi, where $P_e = P_A * (1\text{-}P_E)$, P_A is the probability of an attack, and $(1\text{-}P_E)$ is the probability of adversary success. They then make the assumption that adversaries are like managers of multinational corporations who make rational choices investments and expected returns. They go on to define investment and return measures and use a modified Balanced Scorecard Method to calculate probability of attack. The parameters of the underlying factor model are subjectively derived based on expert's knowledge of adversary goals and value systems.

Major's model [7] for decomposition of P_e sets $P_e = P_A * P_{ND} * P_{SE}$, where P_A is the probability of being attacked by the adversary, P_{ND} is the probability that the attack goes undetected, and P_{SE} is the probability that an undetected attack is successfully executed. Major worked with physical attack in mind; cyber attacks raise new issues, though the thought process is instructive. Some necessary changes in this model include the assumption: if an attack is detected it will be defeated and the consequence will not be generated. This assumption may be reasonable for a physical attack, but it is not justified for a cyber attack because, even if the attack is detected, the cyber attack may have proceeded to successful completion before mitigating measures could be taken.

3.2.2. New Decomposition of P_e. The above discussion and evaluation led us to a new decomposition that is specifically tailored for control systems, retains a focus on the intelligent adversary, and has intuitive meaning for the analysts, testers, and control system users. The following decomposition will be used as the foundation of the risk model used in risk reduction estimation work:

$P_e = P_t * P_a * P_b * P_s * P_c$

In this decomposition, P_e is expressed as a product of conditional probabilities where

- P_t = probability the system is on an attacker target list
- P_a = probability of being attacked given that the system is targeted
- P_b = probability of a perimeter breach given that the system was attacked
- P_s = probability of a successful attack given that there was a perimeter breach
- P_c = probability of damage given the system was successfully attacked.

The decomposition is clear, reasonably intuitive, and sufficiently well-defined to guide the analysis of the proposed method for reduction estimations on SCADA systems. Estimating these probabilities for a site is rather difficult. For example, the probability of a successful attack against a specific site is dependent on the security posture and potential consequences at other sites, the resources available and the consequence preference function of each attacker group, and a slew of intra-site technical and management details. Fortunately risk reduction estimates can be made without estimating absolute values for every component of R_e.

For our case study SCADA system, the security enhancements may affect P_b and P_s but have no affect on the other risk components. Total risk reduction is therefore estimated by estimating the relative change in P_b and P_s.

3.3. Step 3. Identify and prioritize the security requirements of the primary target(s)

The security requirements of a networked system determine the definition of a successful attack and are generally concerned with the attributes of Availability, Integrity, and Confidentiality. For a SCADA system, Confidentiality is of secondary importance while Integrity of the control signals and the Availability of the system are considered to be the highest priority attributes. Therefore SCADA system attacks of greatest interest are "Unauthorized Control" and "Denial of Service".

3.4. Step 4. Identify System Vulnerabilities

Tools that test for cyber security vulnerabilities are available in many varieties from commercial and free sources. The known vulnerabilities associated with each component of the system should be identified by testing and by a search of the vulnerability identification libraries such as Bugtraq (http://www.securityfocus.com), MITRE's Common Vulnerabilities and Exposures (CVE) (http://www.cve.mitre.org), and ICAT (http://icat.nist.gov/icat.cfm). The vulnerability identification libraries contain information about the publicly known vulnerabilities. There may be other vulnerabilities associated with the system that are not included in the public libraries. For example, there may be a vulnerability that is unique to control systems, whereas the libraries are oriented to generic IT vulnerabilities. The identification of vulnerabilities that are not in the public libraries will require expert knowledge of the system and its configuration.

3.5. Step 5. Categorize Vulnerabilities on Each Device by Compromise Type

After the vulnerabilities associated with a given machine have been identified, the plan is to categorize them according to compromise type so that the set of vulnerabilities associated with each edge of the compromise graphs (see Step 7) can be determined. A compromise graph is a directed graph where each node represents a potential attack state. Each node of the graph is one of the following types.

1. **Start** – In this state, nothing is known yet about the details of the target system. This is the single entry node of the graph.
2. **Launch** – Enough data has been collected to begin to develop an exploit or use a known exploit. There is one node of this type for each potential attack entry point on the perimeter.
3. **User_privilege** – This state applies to a particular machine; when the attack is in this state the attacker has gained user level privilege on that machine. There is one state of this type for every machine in the target system.
4. **Root_privilege** – This state applies to a particular machine; when the attack is in this state the attacker has gained root/admin level privilege on that machine. There is one state of this type for every machine in the target system.
5. **Target_node** – Any condition where the attack has succeeded.

The edges of the compromise graph represent a transition from one attack state to another; each transition represents a successful compromise. The value of each edge is an estimate of the time required to make the transition, which is related to the difficulty of exploiting the vulnerabilities associated with that edge. The transition time is a function of the existing vulnerabilities and the attacker skill level. The inverse of the edge value is roughly related to the likelihood that the transition will occur. Figure 1 is an example of a partial compromise graph and applies to our case study (See section 4). It is a partial graph because it does not include values for each edge, it does not show all inter-node edges and does not show all target nodes.

Each vulnerability is categorized into one or more of the following edge types.

1. Type **R** (Reconnaissance)
2. Type **B** (Breach) represents edges starting from a launch node.
3. Type **P** (Penetrate) represents edges starting from a user or root permission node and end on the same type of node,.
4. Type **E** (Escalation) represents an escalation of permissions on the same machine.
5. Type **D** (Damage) represents the transition to a target node.

A vulnerability is classified as the specified type for machine x if it can be exploited to realize that type of compromise on machine x.

3.6. Step 6. Estimate Time to Compromise Each Device

The time-to-compromise (**T**) is defined as the time needed for an attacker to gain some level of privilege on some system device. **T** depends on the nature of the vulnerabilities and the attacker skill level. The value of time-to-compromise is modeled as a random process that combines the following three subprocesses:

Process 1 is for the case where at least one vulnerability is known, and the attacker has at least one exploit readily available that can be successfully used against one of the known vulnerabilities.

Process 2 is for the case where at least one vulnerability is known, but the attacker does **not** have an exploit readily available that can be successfully used against one of the known vulnerabilities.

Process 3 is the identification of new vulnerabilities and exploits. Process 3 is a parallel process constantly running in the background. The attacker of a particular system may use the results of process 3 or may be part of process 3. That is, the attacker may wait for new vulnerabilities/exploits to be identified or probe for new ones.

Each of these processes has a different probability distribution. Process 1 and 2 are mutually exclusive. Process 3 is ongoing and in parallel with the other two processes.

The statistical properties of the above three processes are difficult to fully characterize and validate, however, in previous work [8] we proposed specific process model parameters based on the available data and current literature that yields the following formula.

$T = t_1P_1 + t_2(1-P_1)(1-u) + t_3u(1-P_1)$

where

T is the expected value of time-to-compromise

t_1 is the expected value of Process 1 (1 day)

t_2 is the expected value of Process 2 (5.8 *ET*)

t_3 = ((V/AM) - 0.5) 30.42 + 5.8 ≡ the expected value of process 3

$u = (1 – (AM/V))^V$ ≡ probability that Process 2 is unsuccessful (u=1 if V=0)

V is number of vulnerabilities

$P_1 = 1 - e^{-Vm/k}$

m = number of exploits readily available to the attacker, and k is total number of vulnerabilities in the CVE database.

$$ET = \frac{AM}{V} * \left(1 + \sum_{tries=2}^{V-AM+1} \left[tries * \prod_{i=2}^{tries} \left(\frac{NM - i + 2}{V - i + 1} \right) \right] \right)$$

ET is the expected number of tries

AM is the average number of the vulnerabilities for which an exploit can be found or created by the attacker given their skill level

NM is the number of vulnerabilities that this skill level of attacker won't be able to use (V-AM)

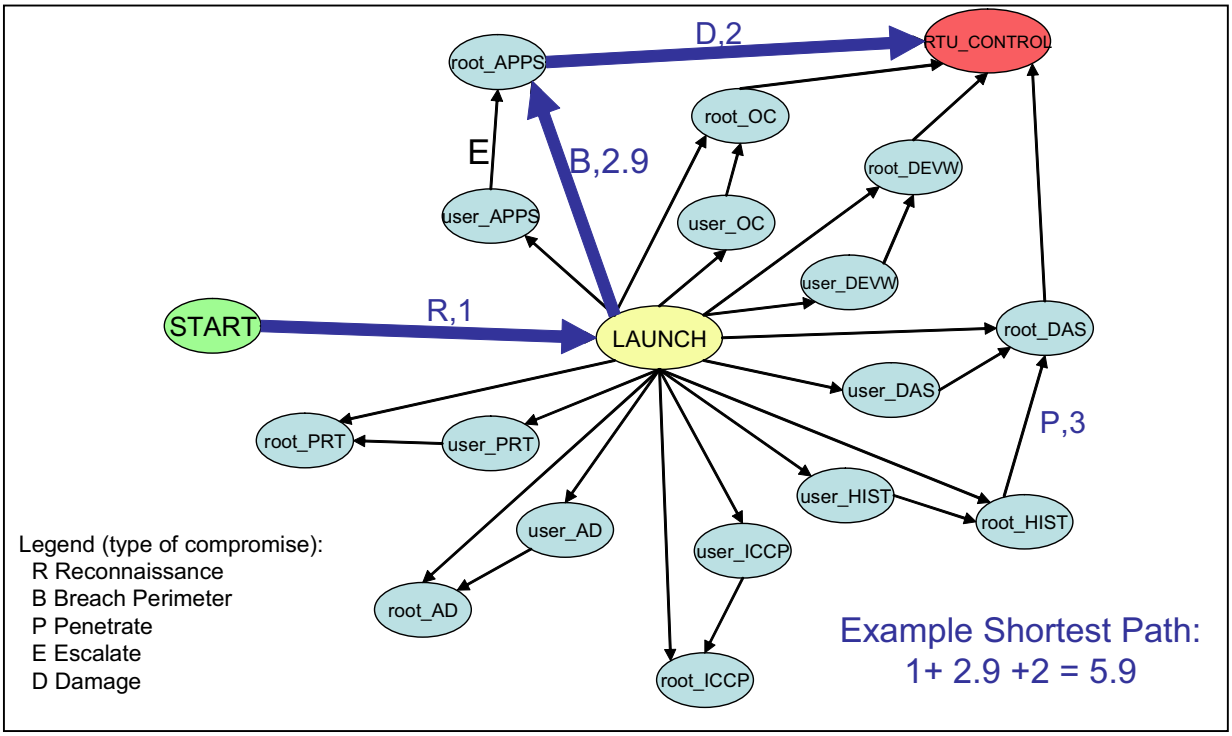

Figure 1. Partial compromise graph for CS60 system

3.7. Step 7. Generate compromise graph(s) and attack paths

As discussed above, a compromise graph is a directed graph where each node represents a potential attack state and each edge represents a successful compromise. The value of each edge is an estimate of the time required to make the transition and is related to the difficulty of exploiting the vulnerabilities associated with that edge. The values of the edges that terminate on a privilege node can be estimated by the method described in [8] using the number of vulnerabilities applicable to that particular type of compromise.

Table 1. Compromise graph compared to well known hacker methodology.

Action step	Description	Compromise graph mapping
Footprint	Casing the establishment	'Reconnaissance' Transition from start state to a launch state
Scan	Look for open or unlocked doors	
Enumerate	High level of intrusiveness information gathering	
Penetrate	Authenticate to remote machine. Gain a foothold.	'breach.' Transition from a launch state to a user_privilege or root_privilege state. Or 'penetrate' a different machine on the control system by transition to user_privilege or root_privilege on that other machine.
Escalate	Obtain super user privileges on the penetrated machine.	'escalate.' Transition from a user_privilege to a root_privilege state on the same machine.
Get interactive	Gain the ability to view and control the internal functions of the machine at will.	'damage.' Transition from a user_privilege or a root_privilege state to a target_node.
Pillage	Do whatever damage desired.	
Expand influence	Use the current position to launch other attacks.	
Cleanup	Install backdoors and erase evidence of entry.	Neglected by compromise graph

3.8. Step 8. Estimate dominant attack path(s)

The dominant attack path for each case was chosen to be the path with minimum value because minimum time-to-compromise implies maximum risk. For the model chosen, the minimum path for every case goes through the machine with the highest number of vulnerabilities.

The compromise graph is a model of a cyber attack where each edge is a transition that moves the attacker one step closer to success. Any sequence of edges that begins at the start node and ends at a target node is an attack path. Compromise graphs can be used to estimate the reduction in probability of a successful attack by comparing the minimum path of the baseline system to the minimum path of the enhanced system. Figure 1 is a partial compromise graph for the CS60 system where an example attack path is shown in bold arrows.

Consider the question of whether our binary model of attacker privilege has sufficient granularity. A binary privilege model applies for the following attack model. The attacker gains the desired level of access by entering at root level or at some low level of privilege and then escalates to root level. From root level, all other privileges can be gained. It is possible for an attacker to escalate to some intermediate level of privilege that leads to a target node without ever gaining root privilege. If that type of scenario is viable and if the probability of occurrence is significant, then the privilege level will need to be modeled explicitly. Modeling of every privilege level on every machine will probably lead to a model of unmanageable size.

Table 1 is a comparison between the proposed compromise graph and the attacker methodology described by Scambray and McClure [10]. It shows how the compromise graph models an attack progression in a manner that is similar to the attacker perspective. Compromise graphs are scalable to very large networks because the number of graph nodes is a linear function of number of machines in the network.

The creation of compromise graphs should take into account cases where there are multiple machines of the same type. Consider a group of machines with identical configurations on the same LAN. The group represents a machine-type and should be treated as a single entity because each has exactly the same set of vulnerabilities.

3.9. Step 9. Do steps 4-8 for baseline and enhanced System

The identification of vulnerabilities and the

estimation of dominant attack paths should be done for both the baseline and enhanced systems to provide a basis for comparison. If the enhanced system has fewer vulnerabilities than the baseline system and if the dominant attack paths for the enhanced system have larger expected times-to-compromise than the baseline system, for every attacker skill level, then risk reduction can be estimated.

3.10. Step 10. Estimate risk reduction

We propose to use estimated time-to-compromise as the primary measure of system security and therefore risk. Other researchers have suggested the use of time as a security metric. For example, Dacier et al. [4] observed that for some kinds of attacks security increases as the *time required* for the success of the attack increases" and also suggested that most attacks could be characterized by "time" and "effort" metrics. Time-to-compromise is a measure of the effort expended by an attacker for a successful attack assuming effort is expended uniformly. It is not known how an increase in the time-to-compromise impacts an attacker; therefore, this is not a precise measurement of risk. However, we believe that as the time-to-compromise is increased, the likelihood of successful attack, and therefore risk, tends to decrease. We make the assumption that risk is inversely proportional to time-to-compromise because it is consistent with the view expressed above and because it leads to the following simple formulation for risk reduction.

Pnew = Pold * (Oldtime/Newtime)

where

Pnew is the probability of a successful attack in the enhanced system,

Pold is the probability of a successful attack in the baseline system,

Oldtime is the expected time-to-compromise for the dominant attack path of the baseline system,

Newtime is the expected time-to-compromise for the dominant attack path of the enhanced system,

Risk reduction can be defined as $\Delta R = 1 - (Pnew/Pold) = 1 - (Oldtime/Newtime)$.

The relationship between relative risk and relative time-to-compromise as defined above is plotted in Figure 2. As expected, the risk reduction approaches 100% as the expected relative time-to-compromise increases.

Total time-to-compromise has variable and fixed components. The fixed component is the portion of the dominant attack path that is the same for the baseline and enhanced systems. The time for 'reconnaissance' and 'damage' may be fixed costs as assumed for this case study. If the fixed costs are a significant portion of the total costs of the baseline system, risk reduction is limited accordingly. Therefore, the analysis separates the effects of fixed vs. variable costs.

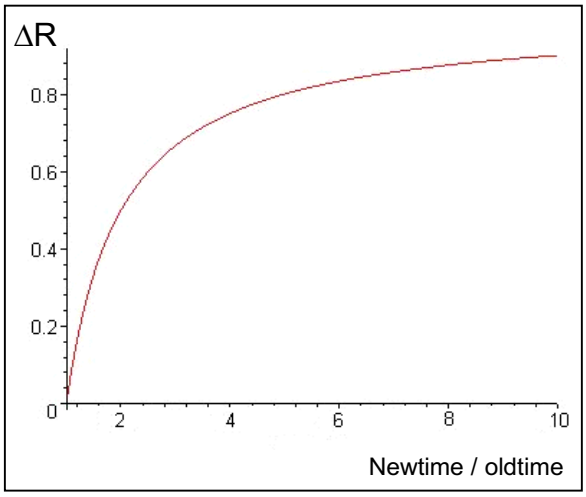

Figure 2. Estimated risk reduction versus time-to-compromise ratio.

4. Case study results

The proposed methodology was applied to a small SCADA system (CS60) consisting of 8 generic machine types connected to a local Ethernet LAN as shown in Figure 3. The attack target device with the highest potential for damage was identified as the RTU (Remote Terminal Unit) because it controls the physical state of equipment in the field. The security requirements for the CS60 system established the RTUs as the primary targets. The primary targets can be compromised by unauthorized RTU messages or by denying the normal flow of messages to and from the RTUs.

The system was tested as delivered from the manufacturer and did not include a firewall. The only perimeter device for the CS60 is the Ethernet switch that connects the system to the internet. For the purposes of testing, this perimeter device was assumed to be a simple switch that prevents locally addressed packets from external observation and prevents flooding of the local network from external sources. There are often additional perimeter devices associated with operational SCADA systems; for example: the ICCP server may have a separate dedicated external network link, and the serial link to the RTUs could be used as an attack entry point.

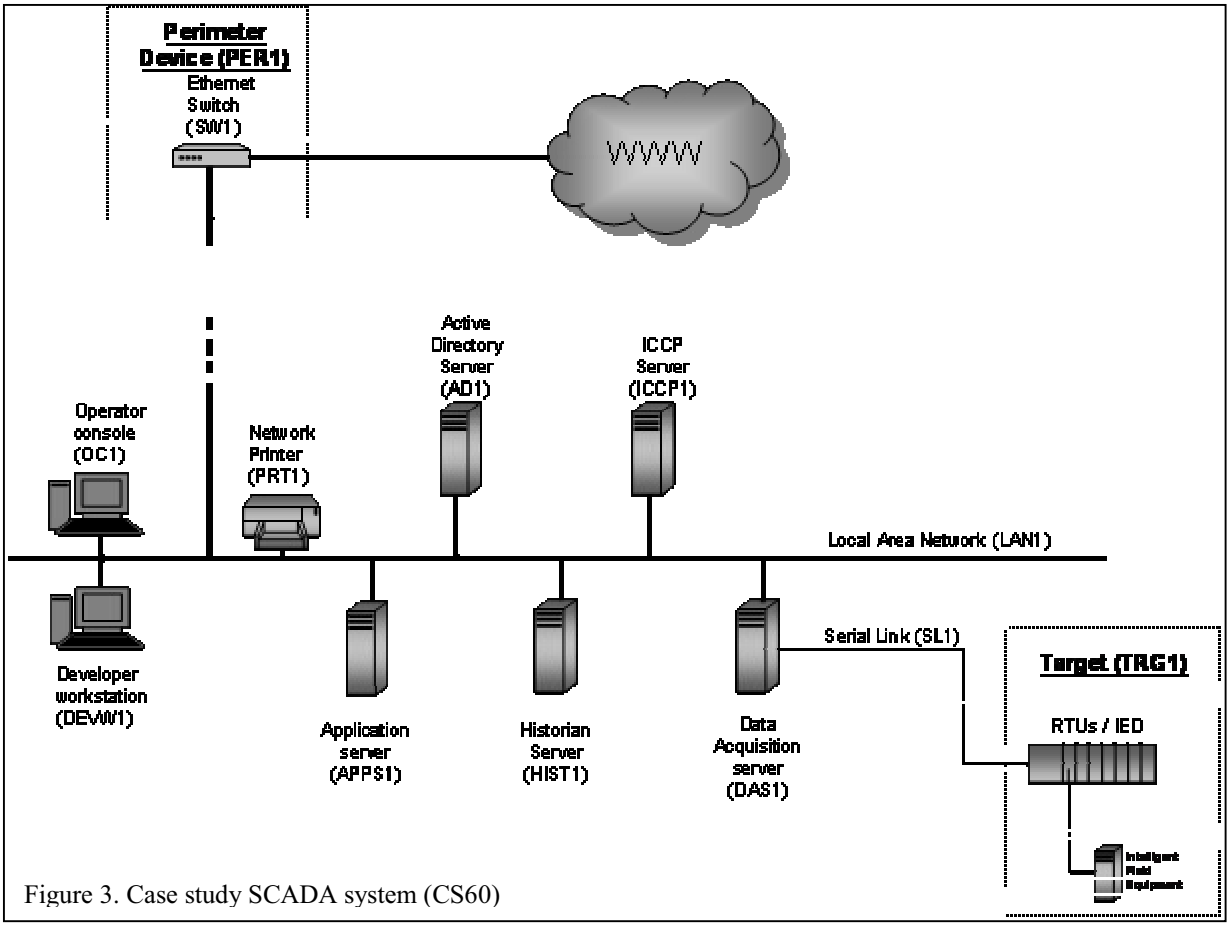

Figure 3. Case study SCADA system (CS60)

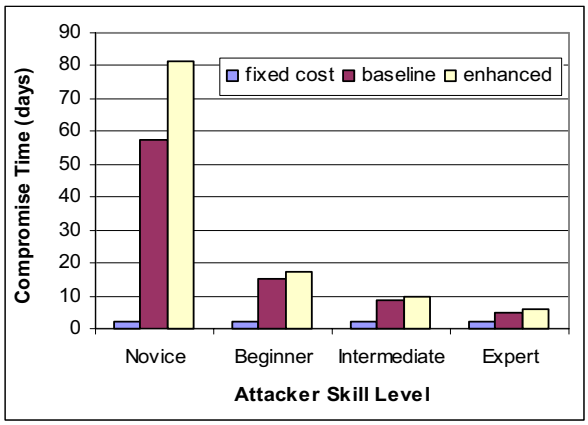

Figure 4. Estimated compromise time for denial of service attack.

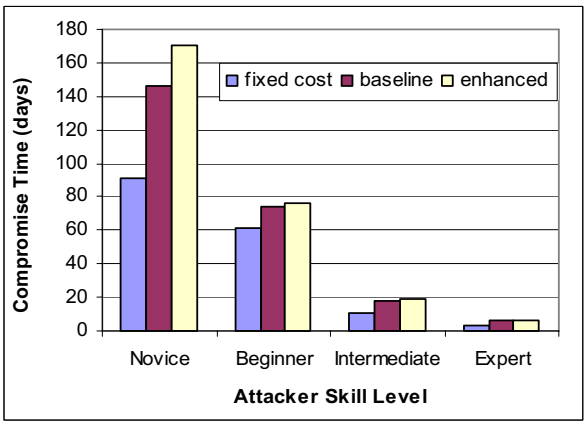

Figure 5. Estimated compromise time for RTU-control attack.

However, the testing of the CS60 system did not include any additional perimeter devices, so they were not included in the analysis.

The values of the "Reconnaissance" and "Damage" edges are expected to be unaffected by the security enhancements that were applied in our case study, therefore the values are considered to be "fixed costs". The time-to-compromise for reconnaissance was estimated to be 1 day for all attacker levels. Since many tools for reconnaissance are freely available and

easy to use we believe the time needed for reconnaissance for our case study system should be a small number in most cases. The time-to-compromise from root permissions to Denial of Service was estimated to be 1 day for all attacker levels because Denial of Service is considered to be a well known exploit that requires no special knowledge of the control system and is therefore expected to be a small number. The time-to-compromise from root permissions to RTU_CONTROL was estimated to be 2 days for expert, 10 days for intermediate skill, 60 days for beginner, and 90 days for novice skill level. These numbers are based on a recent research project at INL researchers who developed exploits of a similar type.

The total time-to-compromise was estimated for the baseline and enhanced versions of CS60 for each attacker skill level as described in sections 3.6 through 3.8. Total time-to-compromise is the value of the dominant attack path, and for the CS60 system it is the sum of the fixed costs plus the time-to-compromise the most vulnerable machine (APPS1). The shortest path example shown in Figure 1 is the dominant path for the case of an expert attacker and RTU_CONTROL target for the baseline system. The dominant attack paths were in every case determined by the maximum number of vulnerabilities per machine; 19 vulnerabilities in the baseline system and 11 vulnerabilities in the enhanced system. Estimated compromise time for a Denial of Service attack is shown in Figure 4 and for an RTU-Control attack in Figure 5.

Risk reduction was estimated as described in section 3.10 using the estimated time- to-compromise values of the dominant attack path for the baseline (old) enhanced (new) CS60 system. Risk reduction estimates are shown in Figures 6 and 7. Denial of Service attack risk reduction estimates range from 11.4 to 29.1%. For an RTU_CONTROL attack, risk reduction estimates range from 2.6 to 13.9%. Fixed costs are shown separately to emphasize how they affect the results. The fixed costs are the portion of the attack paths that were unaffected by the enhancements of the CS60 system and are the sum of the reconnaissance and damage parts of the attack paths, which were chosen somewhat arbitrarily. The estimated risk reduction is very small for the cases where the fixed cost is a significant portion of the baseline total time-to-compromise. Also, the absolute value of time-to-compromise is much lower for expert attackers than for novice attackers, even though the % change is not a strong function of the attacker type.

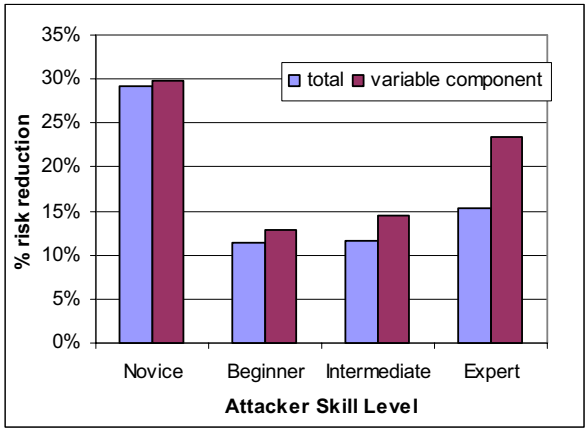

Figure 6. Estimated risk reduction for denial of service attack

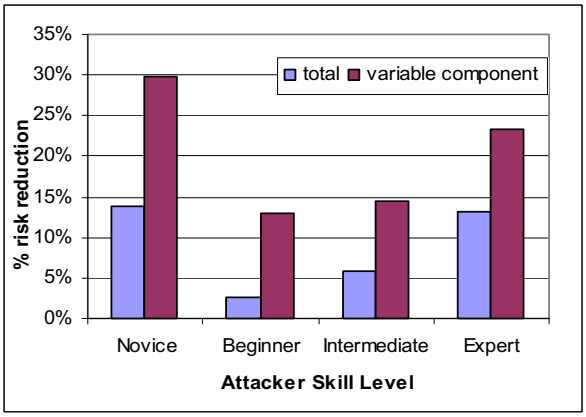

Figure 7. Estimated risk reduction for RTU-control attack.

5. Alternative simplistic risk reduction models/metrics

Consider some simplistic alternative quantitative risk reduction models/metrics. One such model is the binary open/closed door model in which any known vulnerability is considered an open door that a determined attacker will eventually enter. The application of this model to the case study yields a risk reduction of zero because there are known vulnerabilities (open doors) remaining that lead to a successful attack, even though many doors have been closed. This model has some merit, particularly if the attacker is highly skilled and is determined to attack that particular site, but is considered too pessimistic and too simplistic because it does not take into account the various types of potential attackers, the difficulty associated with the exploitation of existing

vulnerabilities, or the time required to complete a successful attack.

Another alternative quantitative risk reduction metric may be obtained by counting the reduction in number of vulnerabilities. This can be done in several ways. For example: the total number of vulnerabilities (before and after system enhancements) associated with all the machines in the system, or the number associated with the most vulnerable machine may be counted. An alternative view of vulnerabilities is the number of open TCP services rather than CVE entries. For this case study, the total holes found by Nessus (http://www.nessus.org) was reduced from 154 to 21 (86%), the number of vulnerabilities on the most vulnerable machine was reduced from 19 to 11 (42%), and the total number of open TCP services was reduced from 298 to 95 (68%). This model is also believed to be too simplistic and too optimistic because it implies a linear relationship between number of vulnerabilities and risk, and ignores other important considerations such as type of attacker and exploit difficulty. Also, a metric that uses total vulnerabilities does not take into account the multiple potential paths of an attack.

6. Conclusions

We proposed a methodology for obtaining a quantitative measurement of the risk reduction achieved when a control system is modified with the intent to improve cyber security defense against external attackers. The methodology employs a directed graph called a compromise graph, where the nodes represent stages of a potential attack and edges represent the expected time-to-compromise for several attacker skill levels. As part of the methodology, time-to-compromise was modeled as a function of known vulnerabilities and attacker skill level and the methodology was applied to calculate risk reduction estimates for a specific SCADA system and for a specific set of control system security remedial measures.

The nature of the numerical results obtained show that risk is related to system attributes in ways consistent with intuition, and reinforces the types of remedial actions that truly reduce risk. For example, the model emphasizes the dynamic nature of cyber security such that risk increases over time, unless there is constant effort to install patches or disable services as soon as new vulnerabilities are discovered. The model suggests the need to harden the weakest link in the attack path, but also emphasizes the need to harden many pathways because of the many potentials paths that could become the weakest link. The model suggests the importance of a firewall to reduce the number of vulnerabilities exploitable externally and suggests the security value of network partitioning within the control system to reduce the number of potential attack paths. The model also suggests the value of an intrusion detection mechanism so that remedial action can thwart an attack if it is detected in time.

The risk analysis model has the following drawbacks. The mapping of time to risk does not account for the intensity of the attacker. The model does not currently take into account dependencies between vulnerabilities on different machines, unless the machines are identical. For example, if two machines are not identical but have some of the same vulnerabilities, compromising them are not independent events. Also, it implies that, after reconnaissance, the attacker has a complete view of the system, but a more realistic attack model may have multiple reconnaissance activities.

The proposed methodology provides a uniform assessment mechanism that can be applied to the evaluation of security measures in other control systems. It provides a quantitative assessment of relative time for an attacker to generate an undesired consequence. The model provides risk estimation differentiation among several attacker skill levels and provides a framework for discussion. The level of abstraction is high enough to be manageable and detailed enough to provide useful security and defensive information. We believe that this methodology is extensible and can be useful for guiding risk assessments and mitigation strategies.

7. Future Work

The kind of data needed to effectively estimate control system cyber security risk is currently unavailable. For example: the industry needs a vulnerability library specific to control systems similar to the existing IT CVE vulnerability library. The existing CVE libraries do not always clearly identify the conditions under which a given vulnerability applies, nor do they indicate how difficult it is to exploit a given vulnerability. Existing vulnerability scanning tools do not clearly identify which vulnerabilities are tested and which are not. We would like to run experiments that measure the statistics associated with Processes 1 and 2. Validated statistical models may allow for a measure of the error bounds associated with future risk reduction estimates.

We would like to extend the compromise graph model to account for dependencies between vulnerabilities on different machines and include

multiple reconnaissance steps such that the path taken is the easiest node currently visible to the attacker. This will require establishing a method for estimating the damage and reconnaissance type edges of the compromise graphs and investigating how other parameters, besides attacker skill level and number of vulnerabilities, affect the compromise time of the edges. It may be valuable to match exploits to vulnerabilities and assess the difficulty in using/tweaking the exploit, but this may change too rapidly to be useful.

We suggest developing a tool to speed the development of user risk analysis models. The ideal tool would be useable by owner operators, so it needs to be intuitive and the components used to specify the system must map well to the users conception of the system. The underlying analytic engine would then transform the user specified model into the underlying analytic model. With future development, the model may be used for aiding secure system design and security requirements by providing better methods of measuring the potential risk reduction associated with design alternatives.

8. References

[1] Beitel, G. A., Gertman, D. I. and Plum, M. M., "Balanced Scorecard Method for Predicting the Probability of a Terrorist Attack," Presented at Risk Analysis 2004, September 27–29, 2004, Rhodes, Greece.

[2] Byres, E. J., Franz, M. and Miller, D., "The Use of Attack Trees in Assessing Vulnerabilities in SCADA Systems", International Infrastructure Survivability Workshop (IISW '04), IEEE, Lisbon, Portugal, December 4, 2004

[3] Carlson, R. E., Turnquist, M. A. and Nozick, L. K., Expected Losses, Insurability, and Benefits from Reducing Vulnerability to Attacks, SAND2004-0742, Sandia National Laboratories, Albuquerque, New Mexico, 2004.

[4] Dacier, M., Deswarte, Y. and Kaaniche, M., "Quantitative Assessment of Operational Security: Models and Tools" Information Systems Security, ed. by S. K. Katsikas and D. Gritzalis, London, Chapman & Hall, p.179-86, 1996.

[5] Haimes, Yacov Y., "Accident Precursors, Terrorist Attacks, and Systems Engineering," Presented at the NAE Workshop, 2003.

[6] Madan, B. B., Goševa-Popstojavova, K., Vaidyanathan, K. and Trivedi, K. S., "Modeling and Quantification of Security Attributes of Software Systems," International Conference on Dependable Systems and Networks, Washington, DC,, 2002..

[7] Major, J. A., "Advanced Techniques for Modeling Terrorism Risk," Journal of Risk Finance, Fall 2002.

[8] McQueen, M. A., Boyer, W. F., Flynn, M. A. and Beitel, G. A., "Time-to-Compromise Model for Cyber Risk Reduction Estimation", First Workshop on Quality of Protection, Milan, Italy - September 15, 2005.

[9] Rinaldi, S., "Modeling and Simulating Critical Infrastructures and Their Interdependencies," Proceedings of the 37th Hawaii International Conference on System Science, 2004.

[10] Scambray, J. and McClure, S., "Hacking Exposed Windows 2000: Network Security Secrets and Solutions," McGraw_Hill, 2001.

[11] Sheyner, O., Haines, J., Jha, S., Lippmann, R. and Wing, J. M., "Automated Generation and Analysis of Attack Graphs," Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy, Berkeley, California, May 2002, 273–284.

[12] Taylor C., Krings, A. and Alves-Foss, J., "Risk Analysis and Probabilistic Survivability Assessment (RAPSA): An Assessment Approach for Power Substation Hardening," Proc. ACM Workshop on Scientific Aspects of Cyber Terrorism, (SACT), Washington DC, November 21, 2002.