A Methodology for Estimating the Mean Time-to-Compromise of a System

David John Leversage
British Columbia Institute of Technology
3700 Willingdon Avenue
Burnaby, B.C., V5G 3H2, Canada
david_leversage@bcit.ca

Eric James Byres
Byres Security Inc.
PO Box 178
Lantzville, B.C., V0R 2H0, Canada
eric@byressecurity.com

Abstract: The ability to efficiently compare differing security solutions for effectiveness is often considered lacking from a management perspective. To address this we propose a methodology for estimating the mean time-to-compromise (MTTC) of a target device or network as a comparative metric. A topological map of the target system is divided into attack zones, allowing each zone to be described with its own state-space model (SSM). We then employ a SSM based on models used in the biological sciences to predict animal behavior in the context of predator prey relationships. Markov chains identify predominant attacker strategies which are used to build the MTTC intervals and can be compared for a broad range of mitigating actions. This allows security architects and managers to intelligently select the most effective solution, based on the lowest cost/MTTC ratio that still exceeds a benchmark level.

Keywords: Calculating Risk, MTTC, SCADA

1 Introduction

One of the challenges faced by any network security professional is providing a simple yet meaningful estimate of a system or network's security preparedness to management who are not security professionals. While it can be relatively easy to enumerate specific flaws in a system, seemingly simple questions like "*How much more secure will our system be if we invest in this technology*?" or "*How does our security preparedness compare to other companies in our sector*?" can prove to be a serious stumbling block to moving a security project forward.

This has been particularly true for our particular area of research, namely the security of Supervisory Control and Data Acquisition (SCADA) and Industrial Automation and Control Systems (IACS) used in critical infrastructures such as electricity generation/distribution, petroleum production/refining and water management. Companies operating these systems are being asked to invest significant resources towards improving the security of their systems, but management's understanding of the

risks and benefits is often vague. Furthermore, competing interests for the limited security dollars have often left many companies making decisions based on the best sales pitch rather than a well-reasoned security program.

The companies operating in these sectors are not unsophisticated – most have had many years of experience making intelligent business decisions on a daily basis on a large variety of multifaceted issues. For example, the optimization of hundreds (or thousands) of process feedback loops in the refining and chemicals industries (typically called control loops) is both extremely complex and critical to profitable operations. Yet, models based on the concept of Key Performance Indicators (KPI) have proven to be successful in simplifying the problem to the point where upper management can make well reasoned decisions on global operations without getting mired in the details. [1]

In our discussions with these companies, it was repeatedly pointed out that similar types of performance indicators could be very useful for making corporate security decisions. What was wanted was not a proof of absolute security but rather a measure of relative security.

To address this need, we propose the concept of a mean time-to-compromise (MTTC) interval as an estimate of the time it will take for an attacker within a specific skills level to successfully impact the target system. We also propose a simple state-space model (SSM) for estimating this interval for a given target system and point to a number of possible methods for determining MTTC values.

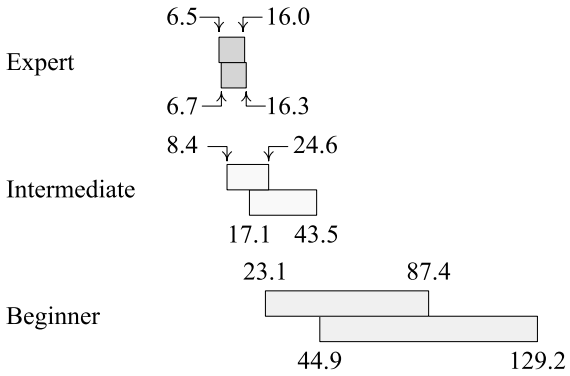

Figure 1 – Example of estimated MTTC intervals (in days) for the network shown in Figure 2. MTTC intervals are shown in pairs with the top interval representing the baseline system and the bottom interval representing the improved system (more frequent firewall rule reviews).

The concept of MTTC is not new – for example, Jonsson uses mean-time-to-breach to analyze attacker behaviors [2] and the Honeynet community uses MTTC as a measure of a system's ability to survive exposure to the Internet [3]. The key point with these works is that MTTC was seen as an observable variable rather than calculated indicator of relative security. McQueen et al [4] [5] moved toward the latter concept with a

methodology that employed directed graphs to calculate an expected time-to-compromise for differing attacker skill levels (The second paper also offers an excellent history of related work). Other works look at probabilistic models to estimate security. However, as McQueen et al points out, many of the techniques proposed for estimating cyber security tend to require significant detail about the target system, making them unmanageable as a comparative tool for multiple systems.

To address this, our model focuses on being a comparative tool and proposes a number of averaging techniques to allow it to become a more generally applicable methodology while still allowing meaningful comparisons. We also developed our model, along with its supporting methodology, with emerging industrial security standards in mind – specifically those being developed by the International Electrotechnical Commission (IEC) [6] and by the International Society for Measurement and Control (ISA) [7] [8].

2 Lessons Learnt from Physical Security

Determining the burglary rating of a safe is a similar problem to determining the security rating of a network. Both involve a malicious threat agent attempting to compromise the system and take action resulting in loss. Safes in the United States are assigned a burglary and fire rating based on well defined Underwriters Laboratory (UL) testing methodologies such as UL Standard 687 [9], which is summarized in Table 1.

UL Ratings	Description
B1	Theft Resistant (minimal security)
B2	UL Residential Security Container
B3	Non-rated anti-theft
B4	UL TL-15
B5	UL TL-30
B6	UL TL-30x6 or TRTL-30

TABLE 1 - UL Safe Burglary Ratings

This rating system is based around the concept of "*Net working time" (NWT),* the UL expression for the time that is spent attempting to break into the safe by testers using specified sets of tools such as diamond grinding tools and high-speed carbide-tip drills. Thus TL-15 means that the safe has been tested for a NWT of 15 minutes using high speed drills, saws and other sophisticated penetrating equipment. The sets of tools allowed are also categorized into levels - TRTL-30 indicates that the safe has been tested for a NWT of 30 minutes, but with an extended range of tools such as torches.

Our discussions with UL testing engineers confirmed that design level knowledge about the safe is used in planning and executing the attacks. They also confirmed that although there are maybe dozens of strategies (classified as attack types) that can be used to gain access to the safe, only a few are actually tried. Finally, each surface of the safe represents an attack zone which may alter the strategies used by the attacker.

There are a few observations about this process that merit mention:

1. There is an implication that given the proper resources and enough time, any safe can eventually be broken into.
2. A safe is given a burglary rating based on its ability to withstand a focused attack by a team of knowledgeable safe crackers following a well defined set of rules and procedures for testing.
3. The rules include using well-defined sets of common resources for safe cracking.
4. The resources available to the testers are organized into well-defined levels that represent increasing cost and complexity and decreasing availably to the average attacker.
5. Even though there might be other possibilities for attack, only a limited set of strategies will be used, based on the tester's detailed knowledge of the safe.

Most important, the UL rating does not attempt promise that the safe is secure from all possible attacks strategies – it is entirely possible that a design flaw might be uncovered that would allow an attacker to break into a given safe in seconds. However, from a statistical point of view, it is reasonable to assume that as a group, TL-30 safes are more secure than TL-15 safes. This ability to efficiently estimate a comparative security level for a given system is the core objective of our proposed methodology.

1. Learning from the philosophy of rating safes, our methodology for rating a target network makes the following assumptions:
2. Given the proper resources and enough time, any network can be successfully attacked by an agent skilled in the art of electronic warfare.
3. A target network or device must be capable of surviving an attack for some minimally acceptable benchmark period (the MTTC).
4. The average attacker will typically use a limited set of strategies based on their expertise and their knowledge of the target.
5. Attackers can be statistically grouped in to levels, each with a common set of resources such as access to popular attack tools or a level of technical knowledge and skill.

These assumptions allow us to calculate the MTTC using a variety of methods that we will outline below.

3 Attack Zones

Just like a safe has different sides that require their own attack strategies, we believe that networks have the same characteristic, namely that a complex network can be divide into zones that are generally homogeneous. Thus we begin by dividing a topological map of the target network into attack zones as is shown in Figure 2. In this particular case, the target of interest is Zone 1, a process control network (PCN) that is buried inside a corporate enterprise network (EN), which in turn is connected to the Internet[1]. Each zone represents a network or network of networks separated from other zones by boundary devices. Within a zone it is assumed that there are consistent security practices in effect such as operating system deployment, patching practices and communications protocol usage. These practices could be good or bad (i.e. patching is performed randomly by users), but they are consistent within the zone.

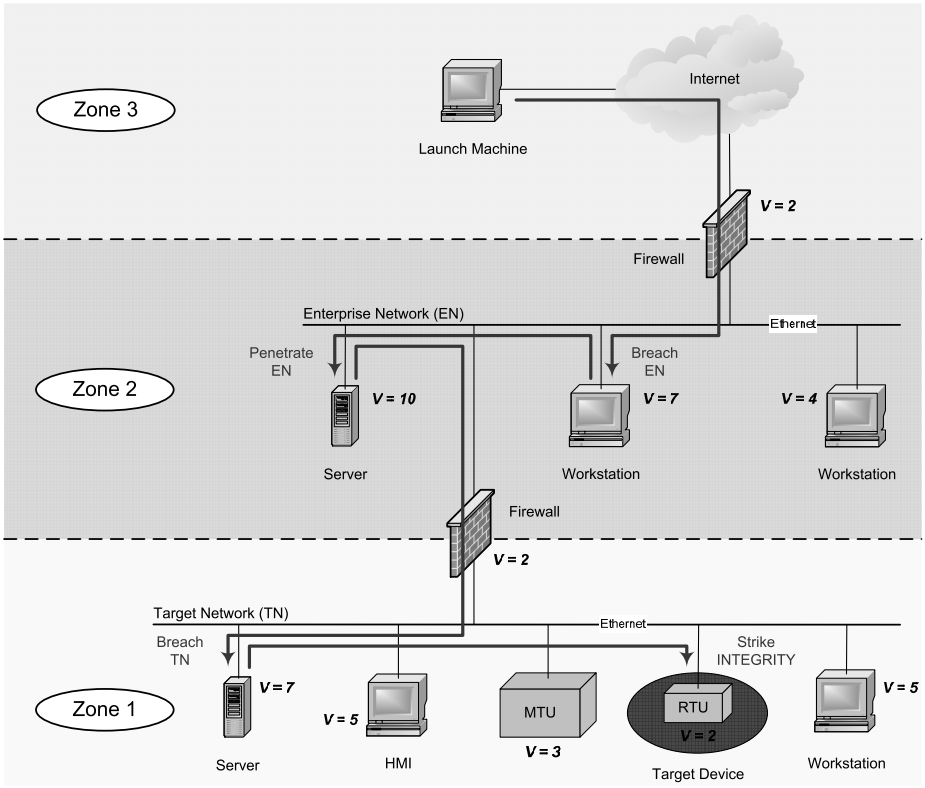

Figure 2 – An example attack zones and the attacker movement through the zones to strike a target device on the target network.

[1] This is a very common architecture in SCADA systems. For example, see "NISCC Good Practice Guide on Firewall Deployment for SCADA and Process Control Networks", www.niscc.gov.uk/niscc/docs/re-20050223-00157.pdf

The concept of zones is important for two reasons. First, an attacker staging an attack from within the target network will likely employ a different set of strategies than he/she would from the Internet and dividing the topological map into zones allows us to represent each zone with its own SSM. Second, by assuming consistent application of practice within a zone, we can make important simplifications to the model to keep it manageable.

4 A Predator Prey State-Space Model

Papers by Sean Gorman [10] and Erland Jonsson [2] provided the motivation and insight to pursue a predator prey-based SSM. For the purposes of this paper, our proposed SSM, shown in Figure 3, is for attacks launched from the Internet. In it we have defined three general states:

1. *Breaching* occurs when the attacker takes action to circumvent a boundary device to gain user or root access to a node on the other side of the boundary.
2. *Penetration* is when the attacker gains user or root access to a node without crossing a boundary device.
3. *Striking is taking action to impact the confidentiality, integrity (take unauthorized control) or availability (deny authorized access) of the target system or device.*

While it is possible to hypothesize many more states (and some may prove to be necessary), our experimentation indicates that having more than 5 states adds little to the output of the model yet greatly increases the complexity of the calculations. For example, McQueen and others suggests *Reconnaissance* states. However, we feel that this can add a significant level of complexity to the process since virtually all states will require some reconnaissance in order to be transited. Thus reconnaissance could just be considered a sub-state and included as part of a primary state's calculations.

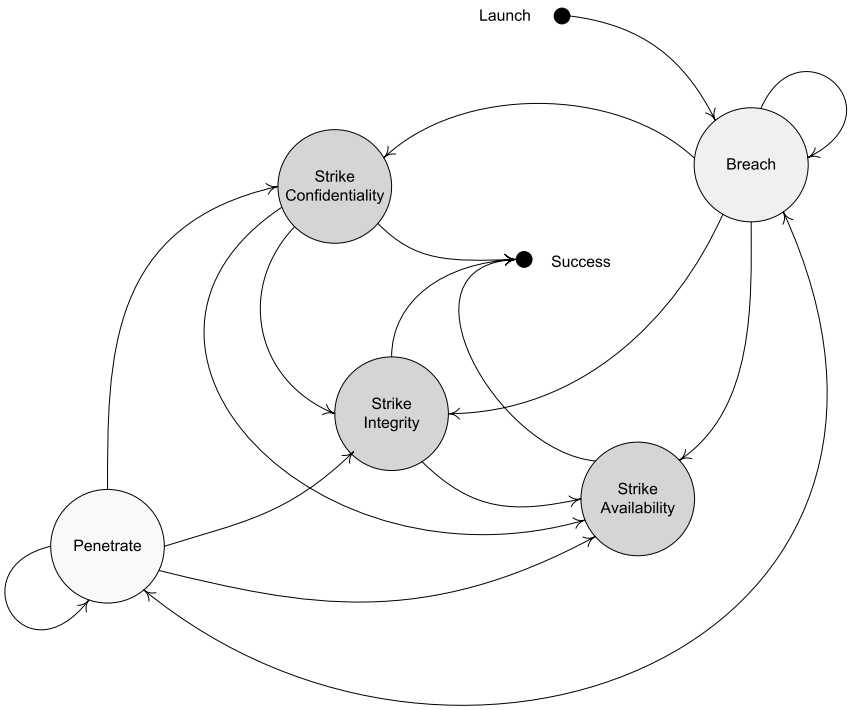

Figure 3 – SSM of attacker movement for attacks launched from the Internet.

The attacker compromises one or more nodes as he/she moves towards the target network as is shown in Figure 2. With layered network architectures, the resulting sequence of compromised nodes appears as movement towards the target and the attacker's strategy is betrayed by the sequence of states - a Markov chain.

5 Attacker Strategies

Since attacker strategy can be represented by a unique Markov chain, the time estimate for each strategy and attacker skill range can be calculated by adding the individual state times. For example, Figure 4 shows the Markov chain for the strategy of breaching the EN, penetrating the EN, breaching the target network (TN) and striking the target device for the network shown in Figure 2.

Similar to the situation with testing safes, the general strategies are identified by subject matter experts. The strategies chosen are strongly dependant on the target network's topology and communication protocols. While developing an exhaustive list of strategies does improve confidence, it is not mandatory in order to get a reasonable estimate of MTTC.

6 Estimating State Times

The next step is to estimate state times and there are numerous methodologies that can be used for this purpose. In this paper we present two; a statistical algorithm based on a modified version of McQueen et al's Time to Compromise Model [5] and an attack tree-based technique. The first allows us to estimate the duration of the breach and penetration states while the second is used to obtain a general and formalized estimate for the strike states in control systems where algorithms are not yet available.

6.1 The State-Time Estimation Algorithm (STEA)[2]

The attacker's actions are divided into three statistical processes:

- Process 1 is when the attacker has identified one or more known vulnerabilities AND has one or more exploits on hand.
- Process 2 is when the attacker has identified one or more known vulnerabilities; however, he does not have an exploit on hand.
- The attacker is in process 3 when there are no known vulnerabilities and no known exploits available.

The total time of all three processes is the time-to-compromise (T) as is shown in (1).

$$T = t_1 P_1 + t_2 (1 - P_1)(1 - u) + t_3 u (1 - P_1) \qquad (1)$$

6.1.1 Process 1

Process 1 is hypothesized to have a mean time of 1 day as is shown in (2). We expect this time to change with experience and we defer to McQueen et al [4] for supporting arguments.

$$t_1 = 1\,day \qquad (2)$$

The probability that the attacker is in process 1 is shown in (3).

$$P_1 = 1 - e^{-V \times M / K} \qquad (3)$$

Where: V = average number of vulnerabilities per zone

M = number of readily available exploits available to the attacker

K = total number of non-duplicate vulnerabilities

In the absence of statistical data, we hypothesize that the distribution of attackers versus skills levels to be a Normal Distribution and we introduce a skills indicator which

[2] To differentiate between original TTCM of McQueen et al and our modified version we call our version the State-Time Estimation Algorithm (STEA).

represents the percentile rating of the attacker and can take on any value from 0 (absolute beginner) to 1 (highly skilled attacker).

M is the product of the skills multiplier and the total number of readily available exploits available to all attackers (m). McQueen chose m to be 450 based on exploit code publicly available over the Internet through sites such as Metasploit. [11] We used the same value for "m" and multiplied by the skills multiplier to get "M" for both the breach and penetration states.

K represents the number of non-duplicate software vulnerabilities in the ICAT database for both the breach and penetration states. We hypothesize that it can be extended represent other classes of vulnerabilities, such as the number of non-duplicate vulnerabilities in the protocol being used to strike the target device.

6.1.2 Process 2

Process 2 is hypothesized to have a mean time of 5.8 days. Again we expect this time to change with experience and we defer to McQueen et al. for supporting arguments. [4]

$$t_2 = 5.8 days \times ET \qquad (4)$$

$$ET = \frac{AM}{V} * \left(1 + \sum_{tries=2}^{V-AM+1} \left[tries * \prod_{i=2}^{tries} \left(\frac{NM - i + 2}{V - i + 1} \right) \right] \right) \qquad (5)$$

Where: ET = expected number of tries

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

NM = number of vulnerabilities that this skill level of attacker won't be able to use

6.1.3 Process 3

This process hypothesizes that the rate of new vulnerabilities or exploits becomes constant over time. [12] To calculate this we need a probability variable u that indicates that process 2 is unsuccessful.

$$u = (1 - s)^V \qquad (6)$$

$$t_3 = ((1/s) - 0.5) \times 30.42 + 5.8 \text{ days} \qquad (7)$$

Equations (6) and (7) differ from the McQueen equations in that AM/V has been replaced with s (the skills factor).

The strength in the STEA model is that can be modified to include other time for sub-states (such as reconnaissance) and can also be adapted to incorporate environmental variables that effect the state times (such as patching intervals). As an example of this flexibility, the study team decided to include a rather abstract variable into the

calculation– the frequency of access control list rule reviews. To do this we first assumed that boundary devices like routers and firewalls offer security by reducing the number of vulnerabilities that are visible to the attacker. In other terms, only a portion of the network's attack surface is visible to the attacker. [13] We then assume that the effectiveness of any boundary device decays if its rule sets are not reviewed regularly [14]. We then incorporated this relationship to the Equations (3) and (6) to produce equations (8) and (9).

$$P_I = 1 - e^{-\alpha \times V \times M/K} \quad (8)$$

$$u = (1 - s)^{\alpha \times V} \quad (9)$$

Where: α = visibility (α = 1 when estimating penetration state times)

Finally we worked with a firewall expert at the British Columbia Institute of Technology to come up with a possible correlation between visibility and update/review frequency. His estimation is: No Reviews, α = 1.00, Semi-Annual, α = 0.30; Quarterly, α = 0.12; Monthly, α = 0.05. Further research is needed to provide support for these estimations but as a proof of concept they are sufficient.

This is one example of the opportunity to add environmental variables that may eventually prove to be important indicators of relative security performance. Other factors we have experimented with include patch intervals, operating system diversity and password policies. If industrial control loop optimization research is any indication, which indicators are truly important and how they affect the MTTC will be an area for considerable future research.

6.2 Estimating Strike State Times Using Attack Trees

In many cases analytical models are not yet available for a given state. For example, in the industrial controls world inherent vulnerabilities in the SCADA protocols themselves appear to have far more impact on the security that operating system or application vulnerabilities [15] and it is not clear if the STEA assumptions apply. To address this issue, our research activities have included exploring ways attack trees can be used to estimate state times.

We developed an attack tree methodology whereby the attacker's strategy maps to a forest of trees and yet remains bound by using a limited set of actions that can be taken at the end nodes based on Military lexicon. Figure 5 illustrates the forest for the strategy of: Breaching the EN, Penetrating the EN, Breaching the TN and then Striking Integrity of the target for the network shown in Figure 2 (which also illustrates the same strategy). Figure 4 also shows the Markov chain for the same strategy for comparison. Notice how the attack trees map to the Markov chain.

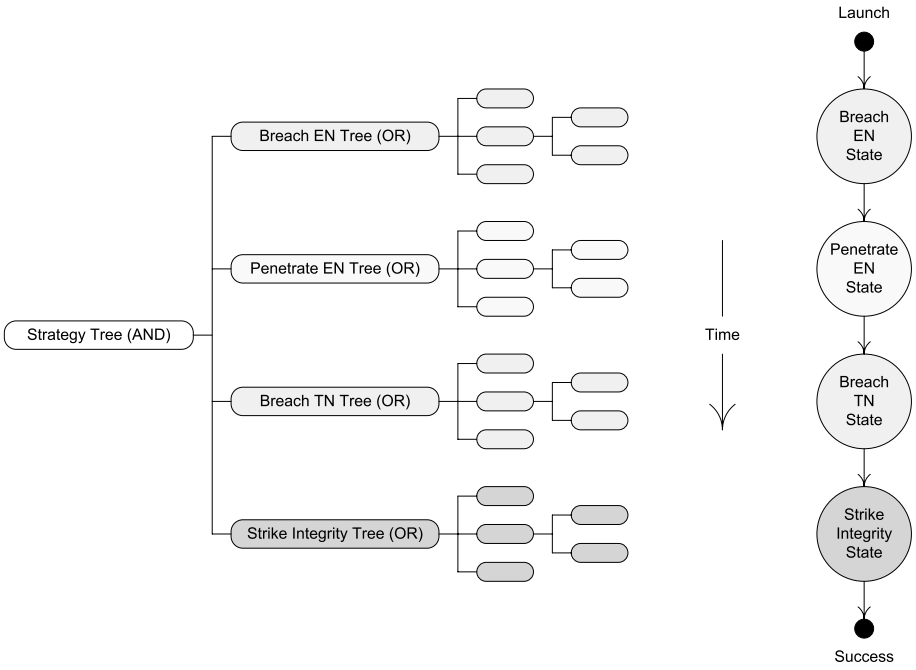

Figure 4 – The attack tree and Markov chain for the strategy of breaching the EN, penetrating the EN, breaching the PCN and then striking the target for the network shown in Figure 2.

Figure 5 illustrates a partial attack tree for breaching the EN by compromising Workstation #1 through software vulnerabilities. Notice that the root of the tree represents goal of attacker and the state. The next layer of nodes represents a physical device under attack. The third layer identifies the failure mechanism (the vulnerability) and the final layer represents the exploit capabilities of the attacker.

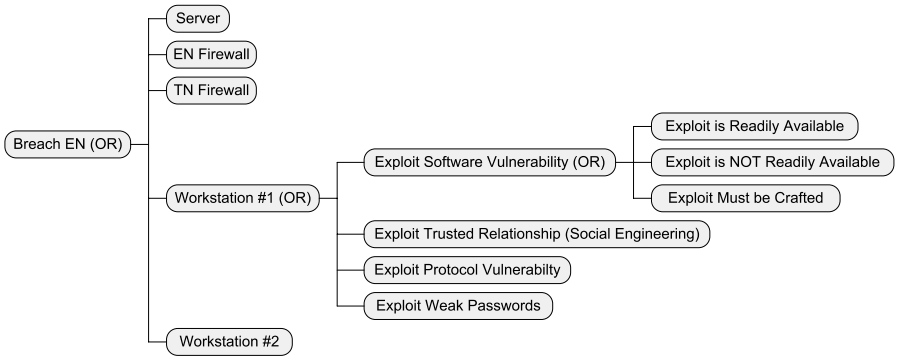

Figure 5 – A partial breach EN tree with software vulnerability exploits expanded.

Figure 6 illustrates a partial strike tree that focuses on vulnerabilities in the SCADA protocols found in the target network. The root of this tree also represents goal of attacker and the state. The next layer of nodes represents the protocol (or protocols) used to attack the target. Layer three identifies the failure mechanism (the vulnerability) based on data communication security goals as they are outlined in IEC/TR 62210. [9] The final layer represents the exploit capabilities of the attacker.

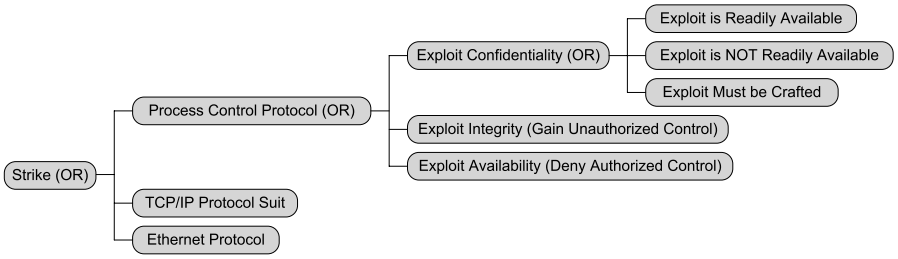

Figure 6 – A partial strike tree focusing on protocol vulnerability exploitation.

Notice the overall similarity and close mapping between Figures 5 and 6. The first layer of nodes represents the object (or objects) under attack. The second layer identifies the failure mechanisms (the vulnerabilities) and the third layer represents the exploit capabilities of the attacker.

We use attack trees to estimate the strike state's time for an attacker to: exploit confidentiality, exploit integrity or exploit availability. Child nodes are based on RFC 3552 [16] and US-CERT publications [17].

Unlike traditional capabilities based attack trees, subject matter experts estimate the time they would need to successfully craft a working exploit for attacks belonging to one or more of the strike state's categories. These times are used in calculating the strike state time when building estimated MTTC intervals and equation (10) is used to obtain an overall estimate.

$$Estimated\ State\ Time = \sum_{n=1}^{N} C_n \times E_n \qquad (10)$$

Where: C_n = weighting coefficient for expert E_n and

$C_1 + C_2 + \ldots + C_N = 1$.

E_n = time estimate made by expert n.

N = total number of experts.

7 Building Estimated MTTC Intervals

As mentioned previously, our MTTC intervals are based on a few attacker strategies selected by subject matter experts rather than exhaustive analysis of all paths. For the network shown in Figure 2, we choose three selected strategies:

1. Breach through EN Firewall to exploit TN Firewall vulnerability and Strike TN Availability.
2. Breach through EN Firewall to exploit EN node, then breach through TN Firewall to exploit TN Node. Strike TN Availability.
3. Breach through EN Firewall to exploit EN node, then breach through TN Firewall to exploit TN Node. Strike Integrity of control device via unauthorized SCADA commands.

Selection of the strategies is highly dependent on the analyst until we have gathered enough statistical data to identify the predominate strategies used by attackers. Results of honeynet research would be extremely useful for this task.

Each strategy will result in its own estimate for a given attacker skill level. The interval for each skill level can be built from only the shortest and longest interval, but our experience suggests that more information is obtained by identifying each strategy with a marker to give us greater insight into the strength of purposed mitigating actions.

8 Case Study Results

As a part of a separate research project into deployment vulnerabilities in the popular SCADA protocol DNP3 [18], we had assembled a team of experts to determine attack trees and strategies for attackers exploiting SCADA systems. We use this information to calculate the MTTC intervals for attacks to impacting these types of systems via the SCADA protocols.

The assumed environment was a control system with the zones and the number of vulnerabilities on each node as shown in Figure 2. The devices in the TN include a server, workstation and Human Machine Interface (HMI) running on Windows O/S, and two embedded controllers - a Master Terminal Unit (MTU) and Remote Terminal Unit (RTU). All use the DNP3 protocol embedded within a standard TCP/IP packet (port 20000) carried over Ethernet to control the process. Other protocols are active between the Windows-based devices, but can not directly affect the industrial process so they are not considered in the actual strike state. The firewall review policy is semi-annual.

First we calculated the MTTC for the three selected strategy markers using the STEA method for all Breach and Penetrate states and the attack tree method for the Strike state.

Next we calculated the security impact if management adopted a policy of monthly firewall reviews and updates. Figure 1 shows the three pairs of MTTC intervals where the top interval of each pair represents the baseline system with semi-annual reviews and the bottom interval of each pair represents the system with monthly reviews. This

indicates that changing from semi-annual to monthly has the most impact on the lower skill level attacker, a reasonable expectation since these attackers are more dependent on published exploits for well-known vulnerabilities. Assuming that management had already conducted a overall qualitative risk assessment and understood their likely threats sources, this information could be combined with cost/manpower estimates to decide if increased reviews was the most effective use of company resources.

We also tried comparing just the state time calculations of the Strike state using both the STEA and attack tree approach (i.e. we assume that all the previous states are completed and the attacker now has control of a non-critical device in the TN). We then compared these results to the results of an experiment conducted at Idaho National Labs where researchers developed exploits in a similar SCADA environment [5]. Table 2 shows the results of this analysis.

The attack tree values for striking Confidentiality and Availability might at first glance to be unrealistically low, but SCADA protocols typically have no confidentiality or access mechanisms available and thus can be snooped and DoS'd with ease [15]. However, exploit tools to inject traffic and gain control of a SCADA system are not widely available and would require considerable time to develop by a beginner or intermediate level attacker. This was shown in an experiment conducted at Idaho National Labs where researchers developed exploits in a similar SCADA environment [5]. Here, the measured times to develop exploits and gain control of the critical control device via the SCADA protocol were 2 days for an expert, 10 days for a intermediate and 60 days for a beginner.

Methodology	Skill Level		
	Expert s =1.0	Intermediate s = 0.9	Beginner s = 0.5
Attack Tree - Strike Confidentiality	1	1	1
Attack Tree - Strike Availability	1	1	1
Attack Tree - Strike Unauthorized Control	5	10	40
STEA	5	10	35
INL SCADA Attack Experiment	2	10	60

TABLE 2 – Strike State Times (in Days) for attacking a SCADA system via the SCADA protocol

While the high level of correlation between the different methods and measured results are encouraging, they are not the critical point. Like in the case of safe testing, the real strength of this methodology is not for obtaining absolute values of security, but rather relative values for comparing differing systems and solutions.

9 Future Research

Currently the STEA methodology focuses primarily on vulnerabilities of a software nature. We hypothesize that it can be further modified to estimate the state times for other vulnerabilities including human related vulnerabilities (i.e. poor password selection) and protocol vulnerabilities. Furthermore, it can be extended to account for a large number of environmental factors such as patch intervals, operating system diversity and password policies. Determining which indicators are truly important and how they affect the MTTC is an area for considerable research.

We envision a stochastic model that eventually leads to a Bayesian model. Relevant statistical data to set the MMTC intervals confidence levels also needs to be collected and promising sources for this statistical data are the Honeynet Project [19] and the results of penetration team testing in the field. Both will help us to improve our state time estimations and to identify predominant attacker strategies. Our experience with the Industrial Security Incident Database leads us to believe that this may even help identify how an attacker's strategies are modified according to environmental conditions (network topology, defenses, etc) and attacker skill levels.

We hypothesized that the distribution of attackers with skills ranging from beginner to expert to be normal distribution. Recent research has us pursuing key risk indicators to identify the key skills and resources used for each of the three attacker levels and to relate these to the attacker's skill level through learning curve theory.

10 Conclusions

The finding of this preliminary research indicates that MTTC could be an efficient yet powerful tool for comparative analysis of security environments and solutions. By deliberately restricting the variety of possible states (and nodes on attack trees) and selecting marker strategies rather exhaustive lists, the model allows reasonable comparisons for decision making purposes.

The selection of time as the unit of measurement is paramount to the model's strength. Time intervals can be used to intelligently compare and select from a broad range of mitigating actions. Two or more entirely different mitigating solutions can be compared and chosen based on which solution has the lowest cost in dollars per day and yet meets or exceeds a benchmark MTTC.

Another important relationship that can be realized is how hard or weak a system is as seen by the attacker compared with peer systems in the same industry. MTTC industry averages (and other averages) can be calculated over time giving and can be used for making peer comparisons. Having MTTC intervals above the average imply that an opportunistic attacker is more likely to move on to another target whereas MTTC intervals below the average imply the opposite.

About the Authors – David Leversage is a faculty member in the Department of Electrical and Computer Engineering at the British Columbia Institute of Technology, teaching courses in electronics and data communication. His research has included analysis of the vulnerabilities in the SCADA protocol DNP3, developing a qualitative risk analysis methodology for the Department of Homeland Security (through INL), and creating new techniques for analyzing Mean Time-to-Compromise in critical systems.

Eric Byres' work in Industrial Cyber Security spans both the academic and industry domains. As the founder of the BCIT Critical Infrastructure Security Centre, he shaped it into one of North America's leading academic facilities in the field of SCADA cyber-security, culminating in a SANS Institute Security Leadership Award in 2006. Mr. Byres has been responsible for numerous standards, best practices and innovations for data communications / control systems security in industrial environments. He is the chair of the ISA SP-99 Security Technologies Working Group and is the Canadian representative of the IEC TC65/WG13. Mr. Byres is currently the CEO of Byres Security Inc. and a Senior Partner in Byres Research.

REFERENCES

[1] Lane Desborough, Randy Miller, "Increasing Customer Value of Industrial control Performance Monitoring – Honeywell's Experience", Proc. 6th Int. Conf. on Chemical Process Control (CPC VI), Arizona, USA, 2001, 172–192.

[2] Erland Jonsson, Tomas Olovsson, "A Quantitative Model of the Security Intrusion Process Based on Attacker Behaviour", IEEE Transactions on Software Engineering, Vol. 23 No. 4, April 1997

[3] http://archives.neohapsis.com/archives/sf/honeypots/2002-q3/0032.html

[4] 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.

[5] Miles A. McQueen, Wayne F. Boyer, Mark A. Flynn, George A. Beitel, "Quantitative Cyber Risk Reduction Estimation Methodology for a Small SCADA Control System", *hicss*, p. 226, Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06) Track 9, 2006.

[6] IEC TR 62210 "Power System Control and Associated Communications – Data and Communication Security", International Electrotechnical Commission, May 2003

[7] ISA-99.00.01: "Security for Industrial Automation and Control Systems Part 1: Concepts, Terminology and Models (Draft)", International Society for Measurement and Control (ISA), Spring 2006.

[8] ISA-99.00.02: "Security for Industrial Automation and Control Systems Part 2: "Establishing an Industrial Automation and Control System Security Program (Draft), International Society for Measurement and Control (ISA), Spring 2006.

[9] UL 687: Standard for Safety Burglary-Resistant Safes, Underwriters Laboratories Inc., Oct 7, 2005

[10] Sean P. Gorman, Rajendra G. Kulkarni, Laurie A. Schintler, Roger R. Stough, "A Predator Prey Approach to the Network Structure of Cyberspace", ACM International Conference Proceeding Series; Vol. 58, 2004

[11] http://www.metasploit.com/

[12] Eric Rescorla, "Is Finding Security Holes a Good Idea", IEEE Security & Privacy, January-February, 2005.

[13] Pratyusa Manadhata, Jeannette M. Wing, "Measuring A System's Attack Surface", Technical Report CMU-CS-04-102, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 2004

[14] Avishai Wool, "A quantitative study of firewall configuration errors" IEEE Computer Magazine, IEEE Computer Society, June 2004, Pages 62-67

[15] 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

[16] RFC 3552, "Security Considerations Guidelines", Internet Enginering Task Force, July 2003

[17] http://www.cert.org/

[18] DNP3 Documentation Library, http://www.dnp.org/

[19] http://www.honeynet.org/