Comparing Electronic Battlefields: Using Mean Time-to-Compromise as a Comparative Security Metric.

David John Leversage[1] and Eric James Byres[2]

[1] British Columbia Institute of Technology, 3700 Willingdon Avenue,
Burnaby, B.C., V5G 3H2, Canada
david_leversage@bcit.ca
[2] 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 which 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.

Key words: Network Security, SCADA Security, Time-to-Compromise, Markov Chains, Predator Model, Attack Paths, Attack Zones, Attack Trees.

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

This paper is © Springer-Verlag Berlin Heidelberg 2007 and was Published in Volume 1 of the Communications in Computer and Information Science, Computer Network Security, Fourth International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security, MMM-ACNS 2007 St. Petersburg, Russia, September 13–15, 2007 Proceedings.

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.

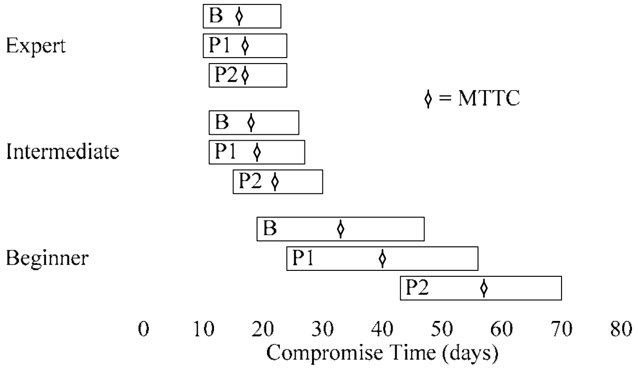

Fig. 1. Example of estimated MTTC intervals (in days) for the network shown in Fig. 2. MTTC intervals are grouped into threes for each attacker skill level and are for the case study given near the end of this paper. The top interval from each group (B) represents the baseline system, the middle interval (P1) represents more frequent patching of nodes on the primary enterprise network, and the bottom interval (P2) represents more frequent firewall rule reviews of the Internet facing firewall.

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]. A few selected UL safe burglary ratings are given in Table 1.

Table 1. Selected UL Safe Burglary Ratings

UL Rating	NWT (Min.)	Testing Interpretation
TL-15	15	Tool-Resistant (face only)
TL-30	30	Tool-Resistant (face only)
TRTL-15X6	15	Torch & Tool-Resistant (6 Sides)
TRTL-30X6	30	Torch & Tool-Resistant (6 Sides)
TXTL-60	60	Torch & Tool-Resistant

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 to 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.

Learning from the philosophy of rating safes, our methodology for rating a target network makes the following assumptions:

1. Given the proper resources and enough time, any network can be successfully attacked by an agent skilled in the art of electronic warfare.
2. A target network or device must be capable of surviving an attack for some minimally acceptable benchmark period (the MTTC).
3. The average attacker will typically use a limited set of strategies based on their expertise and their knowledge of the target.
4. 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.

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 divided 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 Fig. 2. In this particular case, the target of interest is Zone 1, is 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

[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", http://www.cpni.gov.uk/docs/re-20050223-00157.pdf

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.

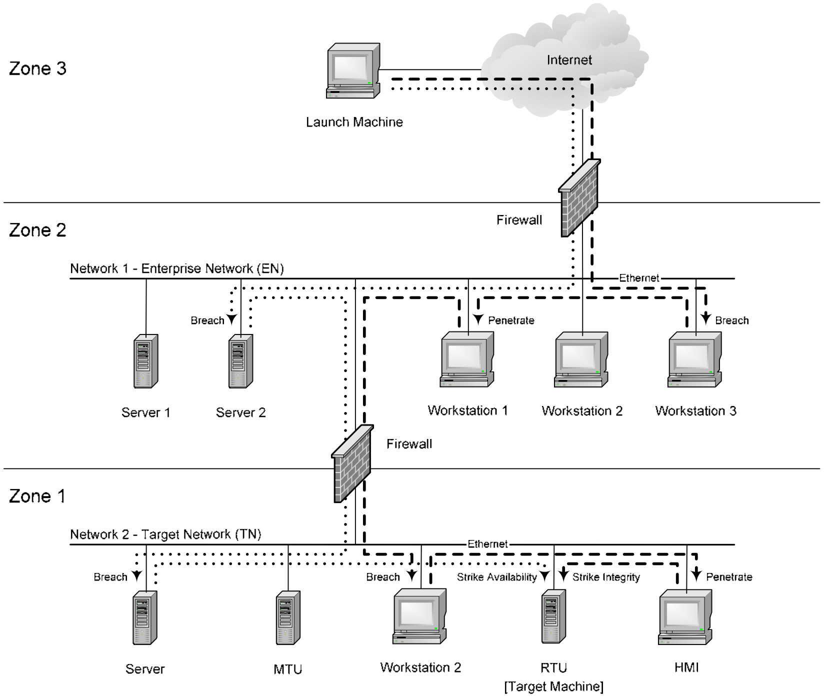

Fig. 2. An example illustrating attack zones and attacker movement through the zones to strike a target device on the target network. The dashed and dotted lines represent two different attack paths that are also represented by the same patterned lines on the attack path model of this system shown in Fig. 4. This topology is used for the case study presented near the end of this paper.

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 Predator 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 Fig 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 five 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 every state 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.

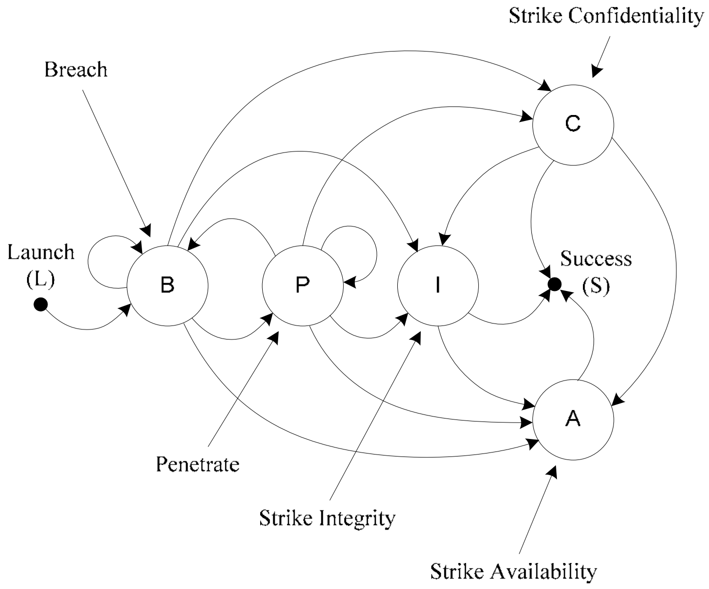

Fig. 3. SSM of attacker movement for attacks launched from the Internet.

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

5 Attack Path Model

The state-space predator model is used to map out the attack path model, a SSM of all possible attack paths from the launch node to the target device taking network topology and security policies into consideration. Consider the network shown in Fig 2. If we make the simplifying assumptions that: the attacker only moves forward towards the target, the firewalls cannot be compromised, and the target device cannot be compromised from a device outside of its zone, then the resulting attack path model is as shown in Fig 4.

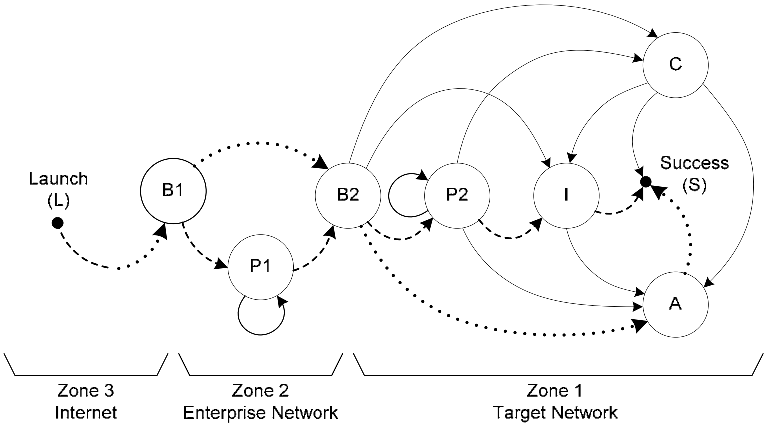

Fig. 4. Attack path model for the network shown in Fig 2 with simplifying assumptions. The dashed and dotted paths correspond to the same patterned attack paths as are shown in Fig 2. State times are given in tables 2, 3 and 4 for the case study given near the end of this paper.

The assumption that the attacker only moves forward towards the target reflects our philosophy that a motivated attacker will not deliberately increase their attack time with unnecessary actions. The second and third assumptions may not be typical of most networks and could be removed. However for the purpose of this paper, they simplify the attack path model to clearly illustrate its salient features.

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 (TTCM) [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 estimated state time (T) as is shown in (1).

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

Where: T = estimated state time
t_1 = mean time that the attacker is in process 1
P_1 = probability that the attacker is in process 1
t_2 = mean time that the attacker is in process 2
u = probability that the attacker is in process 3
t_3 = mean time that the attacker is in process 3

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.

$$t_1 = 1 \text{ day} \qquad \textbf{(2)}$$

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

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

Where: P_1 = probability that the attacker is in process 1
V = average number of vulnerabilities per node within a zone
M = number of readily available exploits available to the attacker
K = total number of non-duplicate vulnerabilities

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

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 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.

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]

$$\mathrm{ET} = \frac{\mathrm{AM}}{\mathrm{V}} * \left(1 + \sum_{\mathrm{tries}=2}^{\mathrm{V}-\mathrm{AM}+1} \left[\mathrm{tries} * \prod_{\mathrm{i}=2}^{\mathrm{tries}} \left(\frac{\mathrm{NM}-\mathrm{i}+2}{\mathrm{V}-\mathrm{i}+1} \right) \right] \right) \tag{4}$$

Where: ET = expected number of tries

V = average number of vulnerabilities per node within a zone

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

$$\mathrm{t}_2 = 5.8\mathrm{days} \times \mathrm{ET} \tag{5}$$

Where: t_2 = mean time that the attacker is in process 2

ET = expected number of tries

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.

$$\mathrm{u} = (1-\mathrm{s})^{\mathrm{V}} \tag{6}$$

Where: u = probability that the attacker is in process 3

s = attacker skill level (0 to 1)

V = average number of vulnerabilities per node within a zone

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

Where: t_3 = mean time that the attacker is in process 3
s = attacker skill level (0 to 1)

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).

$$P_1 = 1 - e^{-\alpha \times V \times M/K} \qquad \textbf{(8)}$$

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

Where: α = visibility ($\alpha = 1$ 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, $\alpha = 1.00$, Semi-Annual, $\alpha = 0.30$; Quarterly, $\alpha = 0.12$; Monthly, $\alpha = 0.05$. 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 than 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.

Fig. 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.

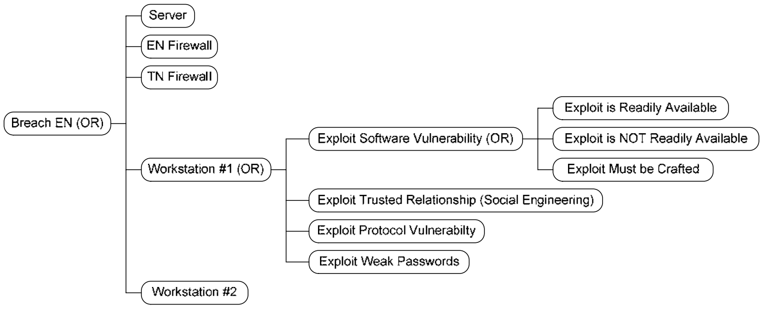
Server
EN Firewall
TN Firewall
Breach EN (OR)
Workstation #1 (OR)
Exploit Software Vulnerability (OR)
Exploit is Readily Available
Exploit is NOT Readily Available
Exploit Must be Crafted
Exploit Trusted Relationship (Social Engineering)
Exploit Protocol Vulnerabilty
Exploit Weak Passwords
Workstation #2

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

Fig. 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.

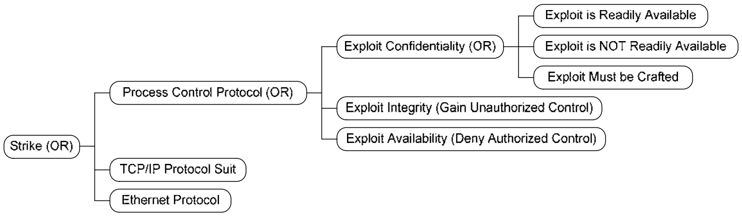
Strike (OR)
Process Control Protocol (OR)
Exploit Confidentiality (OR)
Exploit is Readily Available
Exploit is NOT Readily Available
Exploit Must be Crafted
Exploit Integrity (Gain Unauthorized Control)
Exploit Availability (Deny Authorized Control)
TCP/IP Protocol Suit
Ethernet Protocol

Fig. 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.

7 Building MTTC Intervals

Ideally, MTTC intervals should be based on predominant strategies used by attackers. Until reliable statistical data is available, each attack path is given an equal probability and the attack path model is truncated to allow only one penetration of each network. In practice we expect this not to be true; however, this suffices to provide a metric whereby two or more systems can be compared. Results of Honeynet research would be extremely useful for this task. Each attack path time is estimated for each attacker skill level and the interval for each skill level is built from the shortest and longest attack path time. The product of each attack path probability and its mean time are summed to produce a mathematical expectation for the MTTC itself.

8 Case Study

A utility company wanted to compare mitigating solutions at one of its facilities to determine how to best focus its resources. The system had a topology similar to the system shown in Fig 2 with an average of 6 and 10 vulnerabilities per node on the EN and PCN respectively. (Note: similar topologies are not required by the framework and are only used for illustrative purposes). Firewall reviews on both the Internet facing and Target Network facing firewalls were done on an annual basis. There is limited manpower and financial resources for security and management wants to evaluate two differing approaches. The first is to focus on patching systems on the primary enterprise network that makes up Zone 2. The second is to increase firewall rules reviews from yearly to quarterly on the Internet facing firewall. State times for the baseline system are given in table 2.

Table 2. State times (in days) for the baseline system.

	B1	P1	B2	P2	C	I	A
Expert	4.6	4.6	4.0	4.0	1.0	4.0	1.0
Intermediate	5.2	5.2	4.5	4.5	1.0	4.5	1.0
Beginner	9.5	9.5	8.6	8.6	1.0	8.6	1.0

IT security determined that the number of man hours it would take to reduce the average number of vulnerabilities on the enterprise network from 6 to 3 per node is about the same man hours as it would take to do firewall reviews on the Internet facing firewall on a quarterly basis. State times for both of these approaches are given in tables 3 and 4 respectively.

Table 3. State times (in days) for increased patching frequency of the enterprise network nodes reducing the average number of vulnerab ilities per node to 3.

	B1	P1	B2	P2	C	I	A
Expert	5.2	5.2	4.0	4.0	1.0	4.0	1.0
Intermediate	5.8	5.8	4.5	4.5	1.0	4.5	1.0
Beginner	13.9	13.9	8.6	8.6	1.0	8.6	1.0

Table 4. State times (in days) for qurterly firewall reviews on the Internet facing firewall.

	B1	P1	B2	P2	C	I	A
Expert	5.6	4.6	4.0	4.0	1.0	4.0	1.0
Intermediate	9.1	5.2	4.5	4.5	1.0	4.5	1.0
Beginner	33.0	9.5	8.6	8.6	1.0	8.6	1.0

MTTC levels were estimated for the baseline system and both proposals and are shown in table 2.

Table 5. Estimated MTTC values (in days) for each attacker skill level

	Expert	Intermediate	Beginner
Baseline	16.3	18.3	33.2
Increased Patching	16.8	19.2	39.8
Increased Rules Reviews	16.9	22.2	56.7

IT security determined that both approaches could be implemented using existing resources (primarily human) and each was estimated to cost about $15,000. The resulting cost per day of MTTC being bought (the cost / Δ MTTC ratio) for each attacker skill level is shown in table 6.

Table 6. Cost / ΔMTTC ratios for each attacker skill level

	Expert	Intermediate	Beginner
Increased Patching	$30,000	$16,667	$2,273
Increased Rules Reviews	$25,000	$3,846	$638

Within the framework of an overall qualitative risk assessment, this information could be used to decide if increased rules reviews on the Internet facing firewall is the most effective use of company resources. 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 which are exploited by attacks launched from the Internet. However, 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 resulting in MTTC intervals for a broad range of vulnerabilities and therefore mitigating actions.

Consider the scenarios where a threat agent breaches a plant's physical security and then logs onto an Human Machine Interface and strikes the confidentiality or integrity of the system. Or consider another attacker who takes a sledge hammer to a remotely situated target device and strikes availability. These scenarios involve four states: breach, strike confidentiality, strike integrity and strike availability - remarkably similar to the states in the SSM presented in this paper. We therefore expect that our SSM can be modified to identify attack paths for other attack classes such as social engineering or physical attacks. Similarly, we also expect that the STEA can be modified to estimate state times for other vulnerability classes such as protocol and human vulnerabilities. Identifying and describing a set of models that cover the entire attack surface of the target system is an area of considerable future research and this is where we are pursuing a Hierarchical Holographic Model which will act as the glue to unify our models.

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 a comparative analysis of security environments and solutions.

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 should imply that an opportunistic attacker is more likely to move on to another target whereas MTTC intervals below the average should imply the opposite. However, this is also an area of considerable research.

References

1. Desborough, L., Miller, R,: Increasing Customer Value of Industrial Control Performance Monitoring – Honeywell's Experience. Proc. 6th Int. Conf. on Chemical Process Control (CPC VI) (2001) 172–192
2. Jonsson, E., Olovsson, T.: A Quantitative Model of the Security Intrusion Process Based on Attacker Behaviour. IEEE Transactions on Software Engineering, Vol. 23 No. 4. (1997)
3. http://archives.neohapsis.com/archives/sf/honeypots/2002-q3/0032.html
4. McQueen, M., Boyer, W., Flynn, M., Beitel, G.: Time-to-Compromise Model for Cyber Risk Reduction Estimation. First Workshop on Quality of Protection (2005)
5. McQueen, M., Boyer, W., Flynn, M., Beitel, G.: Quantitative Cyber Risk Reduction Estimation Methodology for a Small SCADA Control System. Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS) (2006)
6. IEC TR 62210: Power System Control and Associated Communications – Data and Communication Security. International Electrotechnical Commission (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) (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) (2006)
9. UL 687: Standard for Safety Burglary-Resistant Safes. Underwriters Laboratories Inc. (2005)
10. Gorman, S., Kulkarni, R., Schintler, L., Stough, R.: A Predator Prey Approach to the Network Structure of Cyberspace. ACM International Conference Proceeding Series, Vol. 58 (2004)
11. http://www.metasploit.com/
12. Rescorla, E.: Is Finding Security Holes a Good Idea. IEEE Security & Privacy (2005)
13. Manadhata, P.: Wing, J.: Measuring A System's Attack Surface. Technical Report CMU-CS-04-102, School of Computer Science, Carnegie Mellon University (2004)
14. Wool, A.: A quantitative study of firewall configuration errors. IEEE Computer Magazine, IEEE Computer Society (2004) 62-67
15. Byres, E., Franz, M., Miller, D.: The Use of Attack Trees in Assessing Vulnerabilities in SCADA Systems. International Infrastructure Survivability Workshop (IISW), IEEE (2004)
16. RFC 3552: Security Considerations Guidelines. Internet Engineering Task Force (2003)
17. http://www.cert.org/
18. DNP3 Documentation Library, http://www.dnp.org/
19. http://www.honeynet.org/