Proceedings of the

3rd Cyberspace Research Workshop

15 November 2010
Shreveport, LA

Editors

Jean Gourd
Louisiana Tech University

Vir V. Phoha
Louisiana Tech University

S.S. Iyengar
Louisiana State University

The Cyberspace Research Workshop is hosted by the Center for Secure Cyberspace (CSC), a collaboration between Louisiana Tech and Louisiana State Universities. Funding for the CSC is made possible by a grant from the Louisiana Board of Regents Support Fund.
LEQSF(2007-12)-ENH-PKSFI-PRS-03

GENERAL CHAIR

Lt. Gen. Robert J. Elder, USAF, Ret.

CONFERENCE CHAIR

Les Guice

PUBLICATION & PROCEEDINGS CHAIRS

Jean Gourd
Vir V. Phoha

PUBLICITY CHAIRS FOR ANNOUNCEMENT & WEB

Christian Duncan
Jean Gourd

LOCAL ARRANGEMENT CHAIRS

Brenda Brooks
Jean Gourd

TECHNICAL PROGRAM COMMITTEE

V. V. Phoha, Chair	S. S. Iyengar, Chair
D. Ali	G. Allen
T. Atkison	K. Balagani
S. Dua	C. Duncan
J. Gourd	Md. E. Karim
T. Kosar	A. Lakhotia
A. Ray	T. Roberts
R. Selmic	P. Wahjudi
J. Walczyk	J. Zhang

MESSAGE FROM THE GENERAL CHAIR

Welcome to the 3rd annual Cyberspace Research Workshop. This event is an opportunity to discuss opportunities to advance cyberspace research in innovative ways. Past workshops have been conducted in conjunction with CIC-sponsored Cyberspace Symposiums; this year we are holding the event alongside the Air Force Global Strike Command's global conference, giving us a different perspective to think about the future of cyberspace and its role in United States national security and economic development. We look forward to a day of exciting discourse on the impact of cyberspace on America's future, in particular, areas where research is needed to maintain United State's competitiveness as a global power. Many thanks for your participation! I look forward to discussing this topic with you and hearing your perspectives on the role of cyberspace in America's future.

Sincerely,

Dr. Robert Elder, Lt General, USAF (retired)

TABLE OF CONTENTS

Strategic methods in adversarial classifier combination *

Anshuman Singh and Arun Lakhotia
University of Louisiana at Lafayette
{axs6222,arun}@louisiana.edu

ABSTRACT

We present an analytical method of configuring performance parameters of individual classifiers in a multiple classifier system (MCS) such that system as a whole incurs minimum misclassification costs. We use game theory for this analysis as it captures the strategic interdependence between the adversary and defender using the MCS. We consider the primitive combinations of MCS - the OR, AND and SELECT for our analysis. The analysis is based on cost-sensitive classification where the objective is not just to maximize the detection rate or minimize the false positive rate but to minimize the total expected costs. This is more practical since in MCS like malware detectors the false positives can lead to high response costs and even if the detection rate is high enough, the strategy of minimizing the expected costs is better than maximizing the detection rate.

1. INTRODUCTION

Many organizations deploy detection systems like malware detectors, intrusion detectors, vulnerability detectors, spam detectors etc. to secure their networks. It is not uncommon to find, for example, a malware detector being used in conjunction with intrusion detector or an intrusion detector being used with spam detector. The individual detectors may further be composed of many specialized detectors. For example, most malware detectors are composed of packing detectors, x-ray scanners, checksum matchers, behavior matchers etc. that work together to correctly classify an input [13].

There are choices in designing a particular detector and each one of the its component detectors. The design choices make a decision on the tradeoff between detection accuracy and cost (efficiency, development cost and time, research cost and time, ease of use etc.). These design choices are generally exercised on individual detectors or classifiers but rarely exercised keeping in mind the security of the system as a whole. In this paper, we focus on *strategic* design of complex security detection systems or more technically, *multiple classifier systems*.

Strategic design considers cost sensitive decisions by adversary and classifier systems. Strategic decision making is considered an art and generally exercised by security experts based on their experience. Instead, the focus is mostly on computational issues in solving security problems. Though, this approach is effective in the short run, in the long run it inevitably leads to a race where more often than not adversary is ahead and the defender is forced to take a reactive approach. A more proactive approach is needed where strategic decisions are not left to experts based on their experience but are determined in a more reasoned, scientific and a formal way.

Strategic design of multiple classifier systems involves choosing the optimal combination method that is robust against adversarial inputs. Adversarial inputs are inputs that have been modified so that they are misclassified by the security system. At a finer level of granularity, strategic design of classifier combinations involves determining the optimal performance parameters for component classifiers so that appropriate classifiers can be chosen using required level of training. In an adversarial setting, the optimal performance parameters of component classifiers system will depend on the choice of evasion or obfuscation method used by the adversary. This strategic interaction can be modeled using game theory. So far, statistical decision theory has been used in designing optimal classifier systems. We consider a game-theoretic modeling and analysis for designing adversarial classifier combinations.

The paper is organized as follows. Section 2 briefly describes related works in adversarial classification and learning. Section 3 describes the cost-sensitive models of classifier and adversary. Basic concepts of multiple classifier systems and game theory are also described. Section 4 presents the game-theoretic analysis for configuring primitive combinations of classifiers. Finally, in section 5, we summarize the paper.

2. RELATED WORK

Adversarial classification was first presented in [6]. In this paper, the interaction between adversary and classifier is modeled as an extensive game where both classifier and adversary are cost-sensitive and classification function and feature change function are assumed to be public information. The adversary first decides the minimum-cost feature change strategy followed by the classifier adapting the clas-

*This research was supported in part by funds from Air Force Office of Scientific Research grant AF9550-09-1-0715.

sification function to the possibility of feature change by the adversary. The Nash equilibrium is then computed which constitutes the optimal classification function for the classifier and minimum cost feature change function for adversary.

The situation when adversary doesn't know the classification function in advance was given in [10]. This paper presented a new learning paradigm suitable for adversarial problems, called adversarial classifier reverse engineering (ACRE), where the goal is not to learn the entire decision surface and adversary's success is measured relative to a cost model. The adversary tries to learn instances that are not labeled malicious in polynomial number of queries.

Adversarial classification and adversarial learning has been the topic of two very recent doctoral dissertations [2, 12]. Design of pattern recognition systems for adversarial classification tasks is studied in [2, 5, 4, 3]. In [2], an analysis of different attacks on different stages of a pattern recognition systems (data preprocessing, feature extraction, model training etc.) is given. Then, a methodology of evaluating the robustness of a classifier at design phase is proposed. Finally, the use of multiple classifier systems is proposed for design of robust systems for adversarial pattern recognition. Multiple classifier systems can also be attacked depending on their architecture, and this possibility is studied in our work. In [12], a case-study of how learners can be manipulated by poisoning the training data is presented. It also presents another case study where evasion using minimum queries of already trained classifier is presented.

Most of the works above study single classifiers in adversarial environment. These works suggest that there is no solution yet to the problem of architectural vulnerabilities of complex classification systems that can be attacked to the the advantage of the adversary. This is precisely the problem addressed in this paper.

3. BACKGROUND

Classifier.

A classifier C is a function $C : X \to \Omega$, where $X = \{x_1, x_2, \ldots, x_n\}$ is the input space and $\Omega = \{\omega_1, \omega_2, \ldots, \omega_m\}$ is the class space. The set Ω is discrete and finite for crisp label classification. It is the set $[0, 1]$ for soft labels representing probabilities of being in a class.

Binary Classifier (Detector).

A binary classifier C_b is a function $C_b : X \to B$, where $X = \{x_1, x_2, \ldots, x_n\}$ is the input space and $B = \{+, -\}$ is the class space. A binary classifier is also called a detector.

Confusion Matrix.

Given a classifier $C : X \to \Omega$, where $\Omega = \{1, 2, \ldots, m\}$ is the class space, the performance of C can be described using an $m \times m$ matrix $C_{conf} = [n_{ij}]$ where n_{ij} is the number of instances of input with true class i that were classified as class j for $i, j = 1, 2, \ldots, m$ [7].

We consider boolean classifiers in the following definitions. Also, given the confusion matrix, C_{conf} of a boolean classifier, $TP = n_{11}$, $FP = n_{21}$, $FN = n_{12}$, $TN = n_{22}$, $P = n_{11} + n_{12}$ and $N = n_{21} + n_{22}$.

True positive rate.

Also called *hit rate*, *recall* and *sensitivity*.

$$\text{tp rate} = \frac{\text{Positives correctly classified}}{\text{Total positives}} = \frac{TP}{TP + FN} = \frac{TP}{P}$$

False positive rate.

Also called *false alarm rate*.

$$\text{fp rate} = \frac{\text{Negatives incorrectly classified}}{\text{Total Negatives}} = \frac{FP}{FP + TN} = \frac{FP}{N}$$

Cost Matrix.

Given a classifier $C : X \to \Omega$, where $\Omega = \{1, 2, \ldots, m\}$ is the class space, the performance of C can be described using an $m \times m$ matrix $C_{cost} = [c_{ij}]$ where c_{ij} is the cost of assigning class j to an instance of input with true class i for $i, j = 1, 2, \ldots, m$. The cost of correct classification is zero, i.e. $c_{ii} = 0, i = 1, 2, \ldots, m$.

Classification related costs.

Costs related to classification for security applications comprises of the following three types [9]:

1. *Operational Cost*: This involves the amount of computing resources needed to make the classification decision.
2. *Damage Cost*: It characterizes the damage done when the classifier misses an attack.
3. *Response Cost*: It is cost of responding when there is a positive classification (or an alarm) by the classifier irrespective of whether it is correct or not.

Cost-sensitive adversary.

A cost-sensitive adversary will have to incur following types of cost:

- *Feature-change cost (Obfuscation cost)*: This is the cost of making changes to the features in the input for evading classifiers.
- *Learning cost*: This is the cost of making changes to the feature change function when a modified input gets correctly classified.

Multiple classifier systems.

Multiple classifier systems are designed for accuracy greater than that of the component classifiers. The two major combination methods known in literature for designing multiple classifier systems are *fusion* and *selection*.

Fusion combination.

The *fusion* combination consists of classifiers connected together in parallel so that for any given input all classifiers are run, and the outputs are combined using some decision function. Decisions of individual classifiers are fused when entire feature space is the input to all classifiers and error rate of all classifiers are almost identical. The most commonly used fusion function is the majority vote. More precisely, given classifiers $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$ and a combination function $f : 2^{\mathcal{C}} \to \Omega$, the fusion combination of a multiple classifier system is defined as:

$$C_{MCS}(x) = f(C_1(x), C_2(x), \ldots, C_n(x))$$

Selection combination.

The *selection* combination consists of classifiers connected together using lightweight classifiers called selectors. Each classifier is an expert on a part of the feature space. Selector decides which part of the feature space needs to be used for classification and the input is directed to the corresponding classifier. Given classifiers C_1, C_2, C_3 with C_1 as selector, selection combination function can be written as:

$$C_{SEL}(x) = \begin{cases} C_2(x) & \text{if } C_1(x) = + \\ C_3(x) & \text{if } C_1(x) = - \end{cases}$$

Game theory.

"Game theory concerns the behaviour of decision makers whose decisions affect each other" [1]. Game theory is a generalization of decision theory. Decision theory is essentially one person game theory.

Normal form game..

It is a tuple $< N, A_i, \succeq_i >$ where N is a set of players indexed by i, A_i is the set of actions for player i where $i \in N$ and $\succeq_i$ is the preference relation of player i defined on the set $A = \times_{j \in N} A_j$, the outcomes of the game. In most cases, the preference relation can be represented by a von Neumann Morgenstern utility function $u_i : A \rightarrow \mathcal{R}$.

Extensive form game..

Since order of moves is relevant in an extensive form game, the concept of *terminal history* is used to describe the sequence of actions that lead to an outcome. An extensive form game is a tuple $< N, Z, P, \succeq_i >$ where N is a set of players, Z is the set of terminal histories, $P : H/Z \rightarrow N$ is a function that maps each nonterminal history to a player(H is the set of all histories, both terminal and nonterminal) and $\succeq_i$ is the preference relation over the set of terminal histories Z.

Mixed and pure strategy..

A mixed strategy of a player in a strategic game is a probability distribution over the player's actions. If A_i is the action set of player i, then the mixed strategy space, Δ_i, is the set of probability distributions of A_i. A pure strategy is a probability distribution that assigns probability 1 to a single action.

Nash equilibrium..

Nash equilibrium is a solution concept for normal form games. A Nash equilibrium is an action profile a^* with the property that no player i can do better by choosing an action different from a_i^* , given that every other player j adheres to a_j^*[11].

Backward induction..

A solution concept for extensive form games that can be applied to any finite game of perfect information. The algorithm begins by determining the optimal choices in the final stage K for each history h^K. Then go back to stage $K-1$, and determine the optimal action for the player on the move there, given that the player on move at stage K with history h^K will play the action that we determined previously. The algorithm proceeds to "roll back", just as in solving decision problems, until the initial stage is reached [8].

4. CONFIGURING PRIMITIVE COMBINATIONS

In section 4.1, we present the cost model that will be used in the game-theoretic analysis of the primitive classifier combinations. It can shown that majority voting used in fusion combination can be implemented using primitive combinations functions boolean AND and boolean OR (See appendix). So we analyze the primitive combinations of OR and AND in addition to selection. In sections 4.2, 4.3 and 4.4, we present the game-theoretic analysis of SELECT, OR and AND combinations to obtain optimal detection rates and false positive rates that minimize the expected cost.

4.1 Cost model

We now give the cost matrix of adversary and multiple classifier system (MCS). The damage cost when MCS misses an attack is given by d. The response cost when there is an alarm or a positive classification by MCS is r. φ is the feature change cost for the adversary. The learning cost for the adversary when the classifier correctly detects an attack is λ. The cost matrix of MCS and adversary is summarized in figure 1.

Let p_D denote the detection rate (true positive rate) of the classifier, i.e. the probability that classifier correctly detects true class of the input. Since the classifier can also give false positives, we denote the false positive rate by p_F. A classifier can be configured to operate at a specific combination of (p_D, p_F) values on its Receiver Operating Characteristics (ROC) curve, which specifies the permissible combinations for the classifier. An ROC curve represents p_D as an increasing concave function of p_F. We assume that the ROC curve is given by the power function $p_D = p_F^r$, with $0 < r < 1$.

4.2 Configuring selection combination

In the selection combination, the adversary can evade the selector to evade the selection combination. We assume that adversary has complete knowledge of the MCS's combination method and the cost matrix and MCS has complete knowledge about the adversary's cost matrix. Evading the selector will direct the input to the classifier that has greater probability of misclassifying. The defender has two options:

1. Configure the selection composition by anticipating the optimum cost of input modification by the adversary. We analyze this option using a sequential (extensive) game.

2. Randomize between selection composition and one of the classifiers. We analyze this option using a simultaneous (static) game in mixed strategies.

In the selection combination, the selector's accuracy of correctly selecting an input to be directed to classifier C_1 is denoted by p_D^S. When the adversary obfuscates the input by changing the features to evade the selector, the accuracy of the selector (C_S) degrades to γp_D^S where $\gamma < 1$. The true positive rate and the false positive rate of C_1 is p_D^1 and p_F^1 respectively. Similarly, the true positive rate and the false positive rate of C_2 is p_D^2 and p_F^2 respectively.

4.2.1 Sequential game analysis

In this section we consider the MCSŰ-Adversary game in which MCS and adversary decide their action in sequence. MCS decides the method of classification for the selector

		Detected class: −	Detected class: +
True class	−	0	r
	+	d	r

MCS

		−	+
True class	−	0	0
	+	φ	$\varphi + \lambda$

Adversary

Figure 1: Cost matrix for MCS and adversary

which determines the detection rate p_D^S. The adversary decides the obfuscation method (feature change cost φ) to evade the MCS. This game is an extensive game of complete information and can be solved using backward induction to give the equilibrium outcome. The equilibrium outcome in this case will be the pair (selector accuracy, cost of evasion).

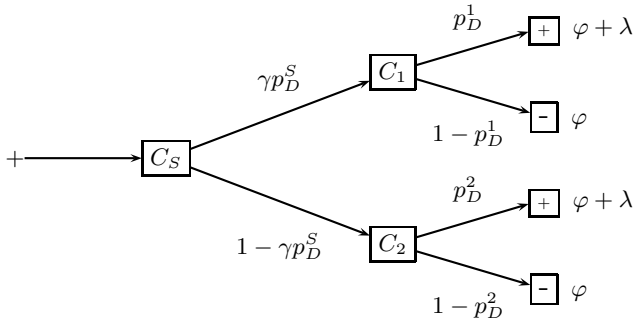


Figure 2: Adversary's cost tree in selection combination

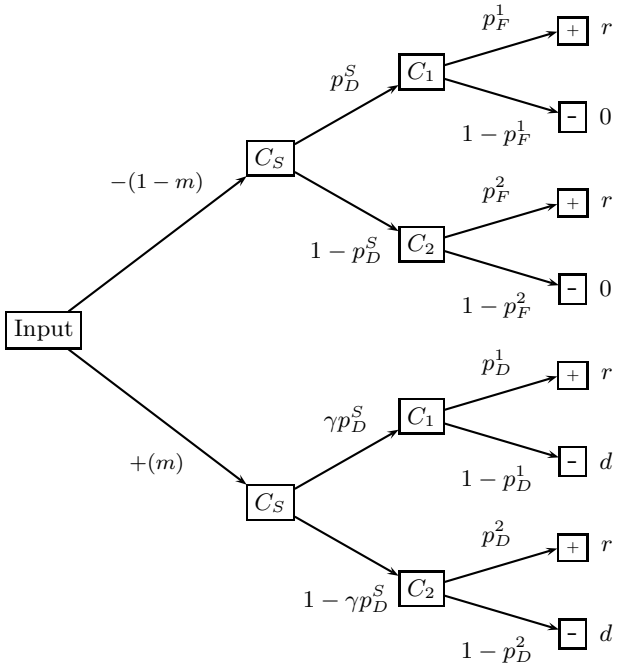


Figure 3: MCS's cost tree in selection combination

The expected cost of the MCS can be obtained from Figure 3 as:

$$\begin{aligned} E[c_{MCS}] =& m(p_D^S(p_F^1(r) + (1 - p_F^1)(0)) + \\ & (1 - p_D^S)(p_F^2(r) + (1 - p_F^2)(0))) + \\ & (1 - m)(\gamma p_D^S(p_D^1(r) + (1 - p_D^1)(d)) + \\ & (1 - \gamma p_D^S)(p_D^2(r) + (1 - p_D^2)(d))) \end{aligned}$$

The expected cost of the adversary can be obtained from Figure 2 as:

$$\begin{aligned} E[c_{Ad}] =& \gamma(p_D^S(p_D^1(\varphi + \lambda) + (1 - p_D^1)(\varphi)) + \\ & (1 - \gamma p_D^S)(p_D^2(\varphi + \lambda) + (1 - p_D^2)\varphi)) \end{aligned}$$

Since, as a proactive measure MCS is configured before the adversary attacks, the backward induction solution involves the following steps.

1. MCS calculates the expected cost of the adversary and minimizes it w.r.t. the cost of evasion
2. MCS then minimizes its own expected cost subject to the constraint of the cost of evasion found above.

Hence, the equilibrium solution (p_D^S, φ) can be obtained by solving the following constrained optimization problem:

$$\min_{p_D^S} E[c_{MCS}]$$

subject to

$$\min_{\varphi} E[c_{Ad}]$$

The expected cost of the adversary can be simplified to obtain

$$E[c_{Ad}] = \gamma p_D^S \lambda (p_D^1 - p_D^2) + p_D^2 \lambda + \varphi$$

The selector's accuracy degradation factor γ will depend on the feature change cost φ. If the selector is robust enough, then a small increase in γ will be obtained with a larger increase in φ. Assuming, $\gamma = \sqrt{\varphi}$,

$$\frac{d(\sqrt{\varphi} p_D^S \lambda (p_D^1 - p_D^2) + p_D^2 \lambda + \varphi)}{d\varphi} = 0$$

yields

$$\varphi = \left[\frac{p_D^S \lambda (p_D^2 - p_D^1)}{2} \right]^2$$

The value of φ computed above can be substituted in $\min_{p_D^S} E[c_{MCS}]$ using $\gamma = \sqrt{\varphi}$ to obtain the equilibrium value of p_D^S. The optimal p_F^S can be obtained using the ROC curve.

There may situations where adversary may not be cost minimizing. Instead, the expected cost may be bounded by a constant K. In this case, the constrained optimization problem becomes

$$\min_{p_D^S} E[c_{MCS}]$$

subject to

$$\gamma p_D^S \lambda (p_D^1 - p_D^2) + p_D^2 \lambda + \varphi < K$$

In this case, the solution may not be a Nash equilibrium because the adversary is not rational in the sense that it is not expected cost minimizing.

4.2.2 Random probabilistic selection

The simultaneous game involves the decision of whether to use selection combination or not on part of the classifier. The adversary decides whether to game the combination or the single classifier. We use random probabilistic selection based on *random* primitive as a defense against gaming of the selector. We consider a static game in mixed strategies where both players randomize between the two options. We solve this game for optimal randomization probability for each player. The game is shown in Figure 4.

The expected costs a_{11} and c_{11} can computed from figures 2 and 3, respectively. The payoffs a_{12}, c_{12}, a_{21}, c_{21}, a_{22} and c_{22} can be obtained from figures 5, 6, 7, 8, 9 and 10 respectively.

	Game selector	**Game classifier**
Use selection	(a_{11},c_{11})	(a_{12},c_{12})
Not use selection	(a_{21},c_{21})	(a_{22},c_{22})

Figure 4: Adversary-Classifier simultaneous game

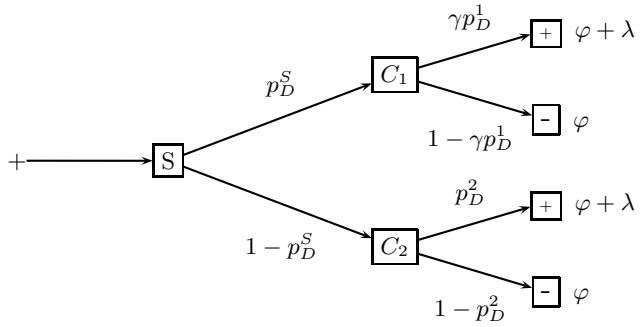


Figure 5: Cost tree of adversary to compute a_{12}

Let α be the probability of adversary gaming the selector and let β be the probability of classifier using the selection combination. Then, the expected payoff of the classifier is:

$$\pi_C = \alpha\beta c_{11} + (1-\alpha)\beta c_{12} + \alpha(1-\beta)c_{21} + (1-\alpha)(1-\beta)c_{22}$$

The expected payoff of the adversary is:

$$\pi_A = \alpha\beta a_{11} + (1-\alpha)\beta a_{12} + \alpha(1-\beta)a_{21} + (1-\alpha)(1-\beta)a_{22}$$

The Nash equilibrium (α, β) can be obtained by solving the simultaneous equations $\frac{\partial \pi_C}{\partial \alpha}$ and $\frac{\partial \pi_A}{\partial \beta}$ for α and β.

4.3 Configuring OR combination

The adversary will have to evade both the classifiers to evade the OR combination. We compute the optimum detection rates of the two classifiers (p_D^1, p_D^2) for the defender and the optimum cost of obfuscating the two classifiers (φ_1, φ_2) for the adversary. The degradation in the detection rate of classifiers C_1 and C_2 due to obfuscation is denoted by γ_1 and

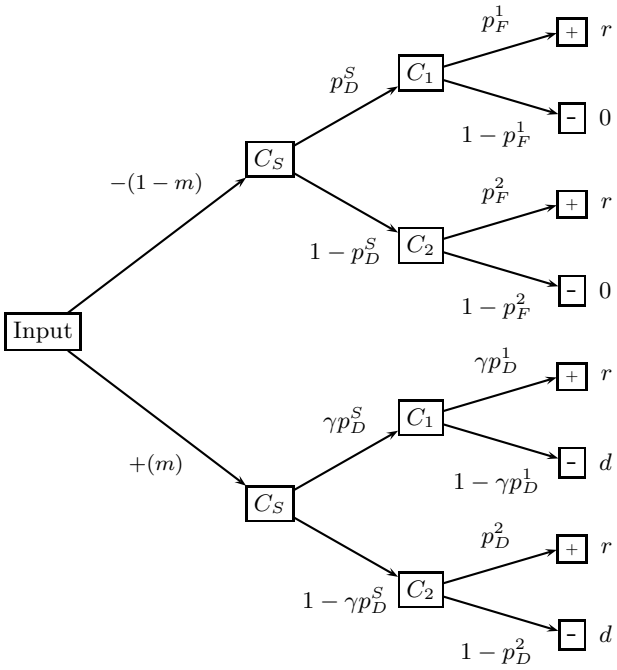


Figure 6: Cost tree of MCS to compute c_{12}

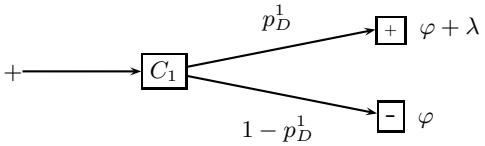


Figure 7: Cost tree of adversary to compute a_{21}

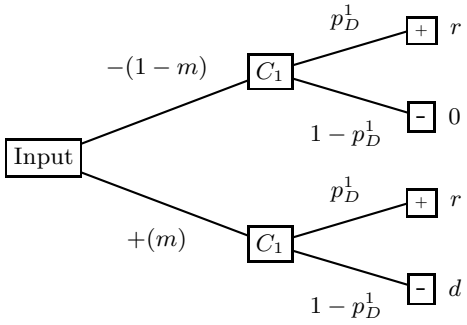


Figure 8: Cost tree of MCS to compute c_{21}

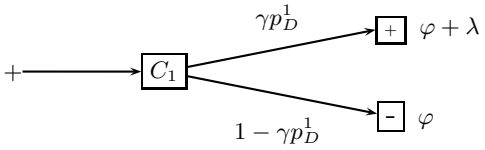


Figure 9: Cost tree of adversary to compute a_{22}

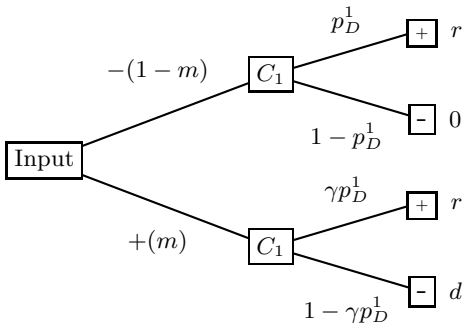


Figure 10: Cost tree of MCS to compute c_{22}

γ_2, respectively. We consider the sequential game in which one of the players picks their strategy first followed by the other. The expected cost of the MCS can be obtained from Table 1 as:

$$\begin{aligned}E[\pi_{MCS}] =& m(r(\gamma_1\gamma_2 p_D^1 p_D^2 + \\ & \gamma_1 p_D^1(1-\gamma_2 p_D^2) + (1-\gamma_1 p_D^1)\gamma_2 p_D^2) + \\ & d(1-\gamma_1 p_D^1)(1-\gamma_2 p_D^2)) + \\ & (1-m)(r(p_F^1 p_F^2 + p_F^1(1-p_F^2) + \\ & (1-p_F^1)p_F^2) + (1-p_F^1)(1-p_F^2)(0)\end{aligned}$$

The expected cost of the adversary can be obtained from Table 2 as:

$$\begin{aligned}E[\pi_{Ad}] =& \gamma_1\gamma_2 p_D^1 p_D^2(\varphi_1+\varphi_2+\lambda_1+\lambda_2) + \\ & \gamma_1 p_D^1(1-\gamma_2 p_D^2)(\varphi_1+\varphi_2+\lambda_1) + \\ & (1-\gamma_1 p_D^1)\gamma_2 p_D^2(\varphi_1+\varphi_2+\lambda_2) + \\ & (1-p_D^1)(1-p_D^2)(\varphi_1+\varphi_2)\end{aligned}$$

Since as a proactive measure MCS configures before the adversary attacks, the backward induction solution involves the following steps.

1. MCS calculates the expected cost of the adversary and minimizes it w.r.t. the cost of evasion
2. MCS then minimizes its own expected cost subject to the constraint of the cost of evasion found above.

Hence, the Nash equilibrium $((p_D^1, p_D^2), (\varphi_1, \varphi_2))$ can be obtained by solving the following constrained optimization problem:

$$\min_{p_D^1, p_D^2} E[c_{MCS}]$$

subject to

$$\min_{\varphi_1, \varphi_2} E[c_{Ad}]$$

4.4 Configuring AND combination

The adversary can evade the AND combination by evading any one classifier for which the cost of evasion is lower. The defender being aware of this vulnerability can randomize between using the AND combination and using the classifier with higher detection rate i.e. some randomly selected inputs are directed to the AND composition and some to the classifier with higher detection rate. The adversary will have to randomize between evading the AND combination (i.e. the classifier with lower detection rate) and evading the classifier with higher detection rate to counter the randomization used by the defender. We can model this situation as a simultaneous (static) game in mixed strategies and the analysis is similar to random probabilistic selection.

5. SUMMARY

We presented a method of deriving optimal performance parameters of classifiers in OR, AND and SELECT combinations. The SELECT combination was analyzed using sequential game as well as a simultaneous game. The sequential game was solved to obtain the optimal detection rate of the selector and the simultaneous game was solved to obtain the optimal randomization probabilities that minimize the expected costs for defender. The OR combination was analyzed to obtain the optimal detection rate of both the classifiers by solving the sequential game between the defender and adversary using backward induction. The analysis for AND combination is similar to random probabilistic selection except that the weaker classifier is considered as a target of evasion.

6. REFERENCES

[1] R. Aumann. Game theory. In S. Durlauf and L. Blume, editors, *The New Palgrave Dictionary of Economics*. Palgrave Macmillan, 2008.

[2] B. Biggio. *Adversarial Pattern Classification*. PhD thesis, University of Cagliari, 2010.

[3] B. Biggio, G. Fumera, and F. Roli. Adversarial pattern classification using multiple classifiers and randomisation. In *Proc. of Joint IAPR Int. Workshop on Structural, Syntactic, and Statistical Pattern Recognition, LNCS vol. 5342*, pages 500–509. Springer, 2008.

[4] B. Biggio, G. Fumera, and F. Roli. Evade hard multiple classifier systems. In *Proc. of Workshop on Supervised and Unsupervised Ensemble Methods and Their Applications (SUEMA 2008), Studies in Computational Intelligence vol. 245*, pages 15–38. Springer, 2008.

[5] B. Biggio, G. Fumera, and F. Roli. Multiple classifier systems for adversarial classification tasks. In *Proc. of 8th Intl. Workshop on Multiple Classifier Systems (MCS 09), LNCS vol. 5519*, pages 132–141. Springer, 2009.

[6] N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '04)*, pages 99–108, 2004.

[7] T. Fawcett. An introduction to ROC analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[8] D. Fudenberg and J. Tirole. *Game theory*. MIT Press, 1991.

[9] W. Lee, W. Fan, M. Miller, S. Stolfoc, and E. Zadok. Toward cost-sensitive modeling for intrusion detection and response. *Journal of Computer Security*, 10(1):5–22, 2002.

[10] D. Lowd and C. Meek. Adversarial learning. In *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 641–647, 2005.

Input true class	C_1	$p(C_1)$	C_2	$p(C_2)$	C_1 OR C_2	$p(C_1$ OR $C_2)$	MCS's payoff
+	+	$\gamma_1 p_D^1$	+	$\gamma_2 p_D^2$	+	$\gamma_1\gamma_2 p_D^1 p_D^2$	r
+	+	$\gamma_1 p_D^1$	-	$1-\gamma_2 p_D^2$	+	$\gamma_1 p_D^1(1-\gamma_2 p_D^2)$	r
+	-	$1-\gamma_1 p_D^1$	+	$\gamma_2 p_D^2$	+	$(1-\gamma_1 p_D^1)\gamma_2 p_D^2$	r
+	-	$1-\gamma_1 p_D^1$	-	$1-\gamma_2 p_D^1$	-	$(1-\gamma_1 p_D^1)(1-\gamma_2 p_D^2)$	d
-	+	p_F^1	+	p_F^2	+	$p_F^1 p_F^2$	r
-	+	p_F^1	-	$1-p_F^2$	+	$p_F^1(1-p_F^2)$	r
-	-	$1-p_F^1$	+	p_F^2	+	$(1-p_F^1)p_F^2$	r
-	-	$1-p_F^1$	-	$1-p_F^1$	-	$(1-p_F^1)(1-p_F^2)$	0

Table 1: MCS's costs in OR combination

C_1	$p(C_1)$	C_2	$p(C_2)$	C_1 OR C_2	$p(C_1$ OR $C_2)$	Adversary's payoff
+	$\gamma_1 p_D^1$	+	$\gamma_2 p_D^2$	+	$\gamma_1\gamma_2 p_D^1 p_D^2$	$\varphi_1+\varphi_2+\lambda_1+\lambda_2$
+	$\gamma_1 p_D^1$	-	$1-\gamma_2 p_D^2$	+	$\gamma_1 p_D^1(1-\gamma_2 p_D^2)$	$\varphi_1+\varphi_2+\lambda_1$
-	$1-\gamma_1 p_D^1$	+	$\gamma_2 p_D^2$	+	$(1-\gamma_1 p_D^1)\gamma_2 p_D^2$	$\varphi_1+\varphi_2+\lambda_2$
-	$1-\gamma_1 p_D^1$	-	$1-\gamma_2 p_D^2$	-	$(1-\gamma_1 p_D^1)(1-\gamma_2 p_D^2)$	$\varphi_1+\varphi_2$

Table 2: Adversary's costs in OR combination

[11] M. Osborne. *An Introduction to Game Theory*. Oxford university press, 2004.

[12] B. Rubinstein. *Secure Learning and Learning for Security: Research in Intersection*. PhD thesis, University of California, Berkeley, 2009.

[13] P. Szor. *The Art of Computer Virus Research and Defense*. Addison-Wesley, 2005.

APPENDIX

A. MAJORITY VOTING AS A COMBINATION OF BOOLEAN AND AND OR

For fusion combination, majority voting is the most commonly used decision combination method. Given n classifiers, if any combination of $(n/2)+1$ ($(n+1)/2$ when n is odd) classifiers give a '+' class, then the combination assigns the output label as '+'. If there are three classifiers, majority voting can be given using boolean AND ($\wedge$) and OR ($\vee$) as follows:

$$(C_1 \wedge C_2) \vee (C_2 \wedge C_3) \vee (C_1 \wedge C_3)$$

This can be generalized to n classifiers as:

$$\mathcal{C}_1 \vee \mathcal{C}_2 \vee \ldots \mathcal{C}_k$$

where

$$\mathcal{C}_1 = C_1 \wedge C_2 \wedge \ldots C_l$$

and so on, $k = \binom{n}{l}$, and $l = (n/2)+1$ when n is even or $l = (n+1)/2$ when n is odd.

Method for Cyber-Security Risk Assessment and Cost-Efficient Management in Wireless Sensor Networks

M. Sahinoglu
Director, Informatics Institute
Auburn University Montgomery
Phone: +1-334-244-3769
mesa@aum.edu

ABSTRACT
In this work, the author describes a methodical procedure for the risk assessment and cost-efficient risk management in Wireless Sensory Networks (WSN) by using a software program: Security Meter. Possible applications to a generic WSN will be illustrated.

General Terms
Algorithms, Management, Measurement, Reliability, Security

Keywords
Risk, Assessment, Vulnerability, Threat, Countermeasure, WSN

1. INTRODUCTION

The core of the paper is to study and deliver a unique risk methodology, suitable to environments in Wireless Sensory Networks (WSN). The Security-Meter (SM) embodies a unique algorithm because the proposed method integrates the necessary ingredients of a full risk portfolio: i) Assessment, ii) Management (mitigation to a tolerable risk level from undesirable), iii) Cost minimization (employing modern game theory), and iv) Recursive feedbacks in real time, all in synergy executed by a single cohesive software. These features are quite an improvement compared to others whose similar components are disjoint and divergent. WSN security monitoring as a vital life- and cost-saving idea is gaining momentum after uncontrolled devastating incidents such as 2005 Katrina levee-breakage and 2010 gulf oil-spill disasters. Partial results using SM have been obtained (Examples in Appendix Figure 3, Table 1 and Figure 4) on other critical environments such as health-care risk assessment and management in HIPAA related activities. These works were based on two-player zero-sum game solutions. We will then first assess cyber-security risk in the general WSN platform and then look for answers to currently significant questions for recovery. Secondly we will look for solutions as to how to mitigate the undesirable risk to a tolerable level with cost minimization in the Appendix Figure 5 and Figure 6, after we work in the health-care domain.

2. MOTIVATION

Wireless Sensor Networks (WSN) contain distributed autonomous sensors distributed spatially to monitor physical or environmental conditions such as temperature, sound, vibration, motion, fire, pollutants or earthquake to name a few, either in civilian life such as industrial factories, refineries, sewages, power plants, or military applications such as battle field surveillance [7,11,12]. Unique characteristics (some of which may easily become their vulnerabilities if not carefully monitored) of a WSN include but not limited to i) Unattended operation, ii) Communication Failures, iii) Mobility of nodes, iv) Heterogeneity of nodes, v) Ability to withstand harsh environmental conditions, vi) Large scale deployment, vii) Dynamic network topology, viii) Ability to cope with node failures, ix) Limited Power they can store and x) Scalable node capacity [13,14]. A tree diagram such as those presented in Appendix Figure 3 on health-care [3] will this time show WSN security vulnerabilities, threats and countermeasures as illustrated in Figure 5. Some of the earlier used military applications are now used for industrial process monitoring and quality control, health monitoring in bio-medical engineering, environment and habitat monitoring, health care applications, home automation, traffic control and space shuttle monitoring at higher assurance level. Sensor nodes can be imagined as small CPUs with limited computational power and memory, sensors, communication device and a power source usually in the battery format [4]. Additionally, the life time of WSNs is determined by the energy which is the scarcest resource of WSN, largely owing remote (mountain tops in wireless networks) and hostile (war scenarios) regions with ad-hoc communications as a key factor. Security and mobility posing the two current problem areas, the following related issues such as life-time maximization, robustness, fault tolerance, self-configuration need to be studied.

3. WHAT TO DO ABOUT WSN RISK

To circumvent this problem of security risk in WSN, the author will analyze the implementation of Security Meter employing the WSN-themed tree diagram in the Appendix Figure 5 accompanied by the results in Figure 6. This procedure will activate the risk assessment algorithm to compute a risk measure for the WSN under scrutiny. Further, the risk mitigation will be activated. The risk management algorithms in the Security Meter can be compared by using two Game theory solution alternatives:

i) Conventionally, two-player zero-sum solution with Minmax=Maxmin condition is studied with existing saddle point.

ii) Not all two-player zero-sum games have saddle points. Such two-person zero-sum games employing Minmax≥Maxmin using mixes of strategies will enable the game to have a saddle point in mixed strategies [9, 10].

In this research paper, due to lack of space, the author will suffice to work on (i) Two-Player Zero-Sum. Likely situations leading to a catastrophe such as in Gulf shores incident will be examined in the near future so as to assess the risk prematurely and manage the risk with cost-optimal countermeasures.

4. METHODOLOGY

Risk Assessment: Innovative quantitative risk measurements are greatly needed to objectively compare risk alternatives and manage existing risks [1,5]. The proposed Security Meter algorithm provides these means in a quantitative manner that is imperative in the security world [2,8]. For a practical and accurate statistical design, security breaches will be recorded so as to estimate the model's input probabilities using the risk equations developed. Undesirable threats (with and without bluffs) that take advantage of hardware and software vulnerabilities can break down availability, integrity, confidentiality, nonrepudiation, and other aspects of software quality such as authentication, privacy, and encryption [4]. Figure 1 below illustrates the constants in the SM model as the utility cost (dollar asset) and criticality constant; the probabilistic inputs are vulnerability, threat, and lack of countermeasure all valued between 0 and 1 [2]. See Figure 1. SM in a black-box is described as follows:

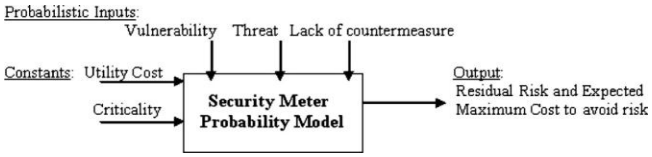


Figure 1. Security Meter Model with probabilistic, deterministic inputs, and calculated outputs.

Probabilistic Tree Diagram: Given that a simple sample system or component has two or more outcomes for each risk factor, vulnerability, threat, and countermeasure, the following probabilistic framework holds for the sums $\sum v_i = 1$ and $\sum t_{ij} = 1$ for each i, and the sum of LCM + CM = 1 for each ij, within the tree diagram structure in Figure 2. Using the probabilistic inputs, we get the residual risk = vulnerability **x** threat **x** lack of countermeasure, where **x** denotes 'multiply'. That is, if we add all the residual risks due to lack of countermeasures, we can calculate the overall residual risk. We apply the criticality factor to the residual risk to calculate the final risk. Then we apply the capital investment cost to the final risk to determine the expected cost of loss (ECL). This helps to budget for avoiding (before the attack) or recovering (after the attack) the risk. Final risk = residual risk x criticality, whereas ECL ($) = final risk x capital cost.

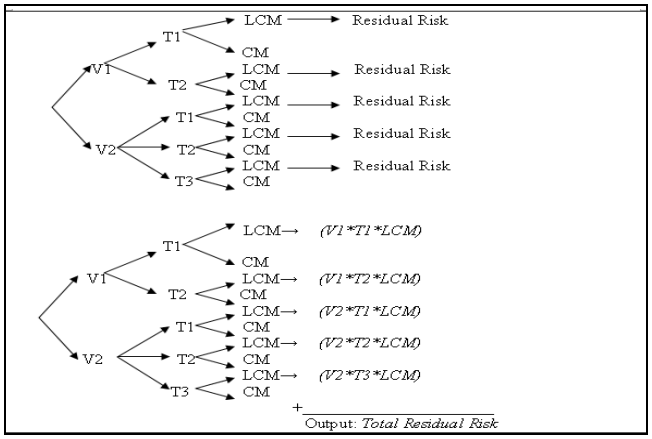


Figure 2. General-purpose tree diagram for the SM model

Algorithmic Calculations: Figure 1 leads to a sample probabilistic tree diagram of Figure 2 so as to perform the calculations. For instance, out of 100 malware attempts, the number of penetrating attacks not prevented will give the estimate of the percentage of LCM. One can then trace the root cause of the threat level retrospectively in the tree diagram. A cyber-attack example: 1) A hacking attack as a threat occurs. 2) The firewall software does not detect it. As a result of this attack, whose root threat is known, is the 'network connectivity' as vulnerability exploited? This illustrates the "line of attack" on the tree diagram such as in Figure 2. Out of those that are not prevented by a certain countermeasure (CM), how many of them were caused by threat 1 or 2, etc., to a particular vulnerability 1 or 2, etc.? We execute as in Figure 2 [1,2]. Calculate the Residual Risk (RR) = Vulnerability x Threat x LCM, for each branch, and proceed by summing the RRs to obtain the total residual risk (TRR).

5. SECURITY RISK ASSESSMENT & COST-EFFICIENT MANAGEMENT METHOD APPLIED TO HEALTH CARE AND WIRELESS SENSOR NETWORKS

5.1. Case Study on Health Care (HIPAA)

Let's assume that we have the following input risk tree diagram in the Appendix Figure 3 and input risk probability chart in Table 1 where one can observe all the vulnerabilities and threats as clearly described in the spreadsheet. For this health care case study, note only the highlighted boxes of interest are selected in Figure 3:

Also note the meanings of the following acronyms of countermeasures (CM) as shown in Table 1:

CM11: Control access, secure, backup, enforce strict policy of sharing records, and let patients decide when records can be disclosed.

CM12: Anti-phishing, firewall, anti-malware scans, off-site backup of insurance records, policy limiting of records vs. insiders, patients control of records.

CM13: Limit access to paper records, secure with passwords and encryption, off-site backup, limiting the share of records.

CM14: Compliance with HIPAA, privacy officer to develop and implement HIPAA policies and procedures, third parties doing business procedures complying with HIPAA, limit access to hardware/software facilities, authentication of those whom you communicate with.

CM21: Prevention of easy access to paper records, secure e-records using frequent passwords and encryption, backup of records off-site and local, patients' control of records.

CM23: Screen staff before hiring, limit who can access records, and prohibit staff from sharing passwords, staff trained in patient privacy and confidentiality, staff made conversant with HIPPA

CM22=CM12; CM31=CM21; CM32=CM22, and CM33= CM23.

Now whereas using the input Table 1 and the results in Figure 4 stemming from Figure 3, and in order to improve the base risk by mitigating from 26% down to 10%, we implement the first-prioritized four recommended actions in Figure 4:

1) Increase the CM capacity for the vulnerability of "Outpatient Facilities" and its connected threat "Patient Records" from the current 70% to 100%.

2) Increase the CM capacity for the vulnerability of "Urgent Care's Surgery Centers" and its connected threat "Patient records" from the current 96% to 100%.

3) Increase the CM capacity for the vulnerability of "Local Health Centers" and its connected threat "Patient records" from the current 72% to 98.54%.

4) Increase the CM capacity for the vulnerability of "Local Health Centers" and its connected threat "Internet" from the current 70% to 99.99%.

In doing these actions, as displayed in Figure 4, a total amount of $510 is dispensed (< $513.30 as advised) each within the limits of optimal costs annotated, staying below the breakeven cost of $5.67 per % improvement. The next step proceeds with optimization to a next tolerable risk percentage once these acquisitions or services are provided. The next action could well be one such as down to 5% from a current 10%, budget allowing.

5.2. Case Study on Wireless Sensor Networks

Use the Appendix Figure 5's sample tree diagram for a WSN Cyber-Risk study and Figure 6's cost-efficient solution for a specific set of responses to the diagnostic questions. With the selected vulnerabilities, that is only two out of seven, i.e., hardware and network interconnectivity and their pertinent threats (only two selected per vulnerability), to mitigate the initially assessed risk from 48% down to 40%: i) Increase the CM capacity from 51% up to 75.37% in the 'Security' threat of the 'Network Interconnectivity' vulnerability. ii) Increase the CM capacity from 53.50% to 53.55% in the 'Robustness/Resiliency' threat of the same 'Network' vulnerability. If this study is to involve an oil-rig WSN scenario [6], then vulnerabilities and threats will be designed in accordance with the oil-rig terminology for a working algorithm. The study shown here is only an example leading to the prototype analysis which is valid in all generic WSN.

6. DISCUSSIONS AND CONCLUSIONS

Addressing risk quantification and cost-efficient risk management issues with a game theory solution may prove this research work worthwhile to pursue. At the end of this research, the managers and analysts of complex *wireless sensor networks* are expected to develop an awareness of what risk factors prevail by assessing and managing the risk content prematurely before preventable tragic events roll out of control. That is, in High-Assurance systems such as oil-rigs, hospital surgery wards or NASA projects using Software-Based event detection mechanisms [6, 7, 11, 12, 13, 14].

7. ACKNOWLEDGMENTS

Author's thanks go to Professor S.S. Iyengar from LSU and Professor Phoha from LaTech for exchanging ideas, and brainstorming on this methodology to reach a consensus, and also to Scott Morton, CS assistant at the Informatics Institute of AUM, for the visio diagrams and sample diagnostic questions.

8. REFERENCES

[1] M. Sahinoglu, Trustworthy Computing - Analytical and Quantitative Engineering Evaluation. Hoboken, New Jersey: J. Wiley & Sons Inc., July 2007.

[2] M. Sahinoglu, "Security Meter - A Practical Decision Meter Model to Quantify Risk," IEEE Security and Privacy, vol. April-May, pp. 18-24, 2005.

[3] Aysen Dener, M. Sahinoglu, Vir Phoha, S. Morton, "A Quantitative Security and Privacy Risk Assessment and Management Method for Social Networks", Accepted for Presentation at ISI/CRA (Comm. On Risk Analysis) Special Session on Trustworthy Computing, Dublin, Ireland, August 2011

[4] V. V. Phoha, Internet Security Dictionary, Springer-Verlag, New York, New York, 2002. Listed in "Best books of 2002," Journal of Object Technology (ETH Zurich), Vol. 2, No. 1, Jan-Feb 2003, pages 123-125. Excellent reviews in ACM Computing Reviews; Book Reviews, International Journal of Mathematics and Computer Education, Spring 2003; and ComputerWorld, Nov. 25, 2002.

[5] M. Sahinoglu, "Method of Automating Security Risk Assessment and Management with a Cost-Optimized Allocation Plan," USA Patent Filing Date: March 20, 2009, Publication Date : September 23, 2010

[6] S.S. Iyengar, S. Mukhopadhyaya, C. Steinmuller, and Xin Li, "Preventing Future Oil Spills with Software-Based Event Detection", IEEE Computer, August 2010, pp. 95-97

[7] S.S. Iyengar, N. Parameshwaren, Vir Phoha, N. Balakrishnan, Chuka D. Oyoke, Fundamentals of Sensor Network Programming, John Wiley publications, Spring 2010

[8] M. Sahinoglu, "An Input-Output Measurable Design for the Security Meter Model to Quantify and Manage Software Security Risk", IEEE Trans on Instrumentation and Measurement, 57(6), June 2008, 1251-1260

[9] Martin Shubik, Games for Society, Business and War: Towards a Theory of Gaming, Elsevier, 1975 (Part II, 29-117)

[10] Robert Aumann and Jacques Dreze, "Rational Expectations in Games," American Economic Review, March 2008 http://www.ma.huji.ac.il/raumann/pdf/86.pdf

[11] A. Ray, V. V. Phoha and S. Phoha, eds., Quantitative Measure for Discrete Event Supervisory Control, Springer, New York, 2005.

[12] V. V. Phoha, A. U. Nadgar, A. Ray and S. P. Phoha, Supervisory Control of Software Systems, IEEE Transactions on Computers, 53 (2004), pp. 1187-119.

[13] Kun Yan, Hsiao-Chun Wu, and S. Sitharama Iyengar "Robustness Analysis and New Hybrid Algorithm of Wideband Source Localization for Acoustic Sensor Networks", IEEE Transactions on Wireless Communications (2010).

[14] Mengxia Zhu, Song Ding, Richard R. Brooks, Qishi Wu, Nageswara S.V. Rao, and S. Sitharama Iyengar, "Fusion of Threshold Rules for Target Detection in Sensor Networks" , ACM Transactions on Sensor Networks, Vol 6 and Issue 2, May 2010.

APPENDIX

The following Appendix material will include Figures 3 to 6 and Table 1, all of which have been cited in the manuscript's Section 1-5.

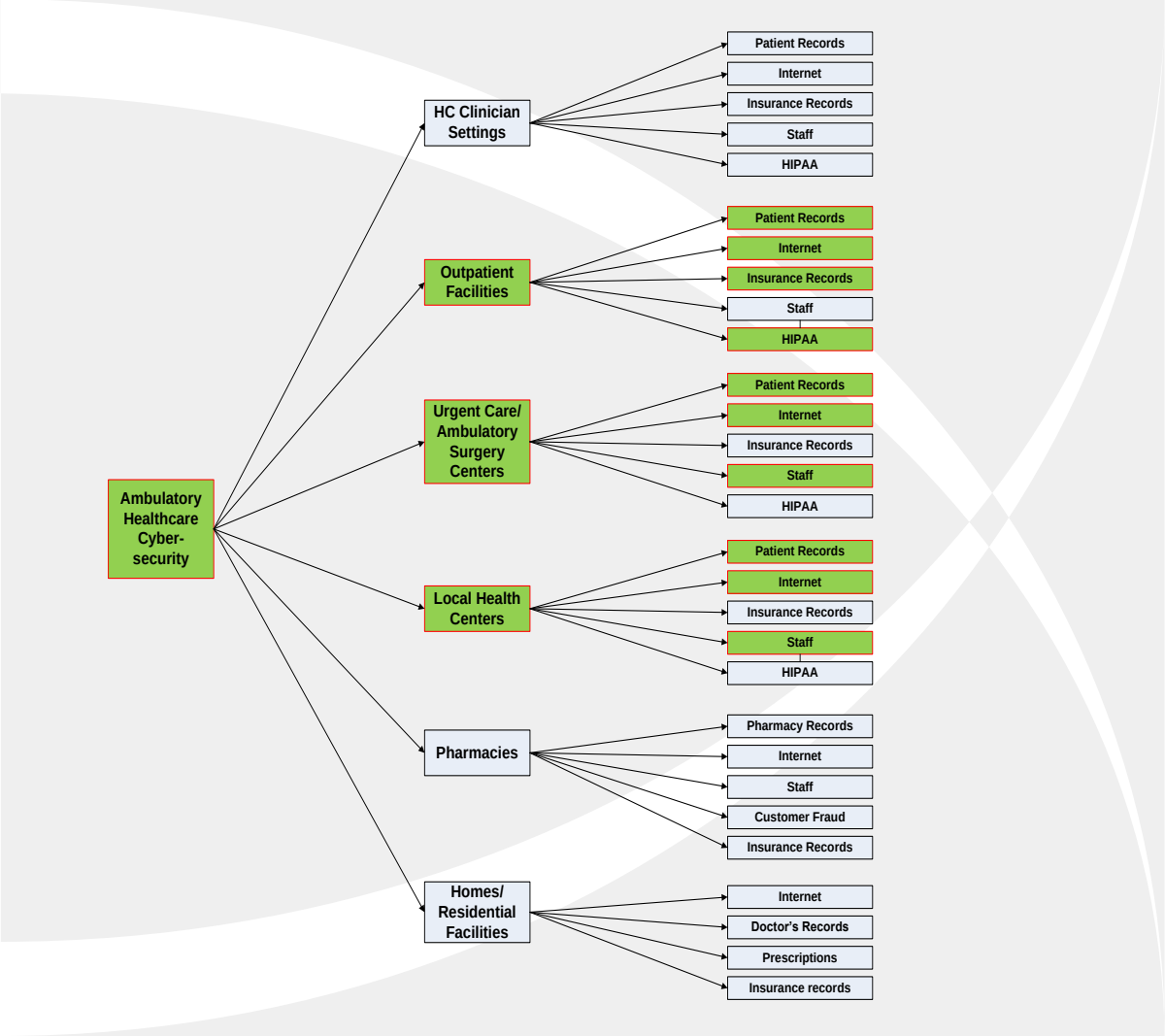


Figure 3: Health care enhancement (HIPPAA) related Security Meter's tree diagram with highlighted selections

Table 1. A Sample Security Meter Probability Chart for Health Care.

Vulnerability	Threat	Countermeasure
$V_1 = 0.35$ (Outpatient Facilities)	$T_{11} = 0.48$ (Patient records)	$CM_{11} = 0.70$ LCM_{11}=0.30 by Subtraction
	$T_{12} = 0.16$ (Internet)	$CM_{12} = 0.42$ $LCM_{12} = 0.58$ by Subtraction
	$T_{13} = 0.32$ (Insurance Records)	$CM_{13} = 0.97$ $LCM_{13} = 0.03$ by Subtraction
	$T_{14} = 0.04$ (HIPPA)	$CM_{14} = 0.80$ $LCM_{14} = 0.20$ by Subtraction

$V_2 = 0.26$ (Urgent Care/Surgery)	$T_{21} = 0.22$ (Patient records)	$CM_{21} = 0.35$ $LCM_{21} = 0.65$ by Subtraction
	$T_{22} = 0.02$ (Internet)	$CM_{22} = 0.35$ $LCM_{22} = 0.65$ by Subtraction
	$T_{23} = 0.76$ (Staff)	$CM_{23} = 0.96$ $LCM_{23} = 0.04$ by Subtraction
$V_3 = 0.39$ (Local Health Centers)	$T_{31} = 0.32$ (Patient records)	$CM_{31} = 0.72$ $LCM_{31} = 0.28$ by Subtraction
	$T_{32} = 0.59$ (Internet)	$CM_{32} = 0.70$ $LCM_{32} = 0.30$ by Subtraction
	$T_{33} = 0.09$ (Staff)	$CM_{33} = 0.46$ $LCM_{33} = 0.54$ by Subtraction

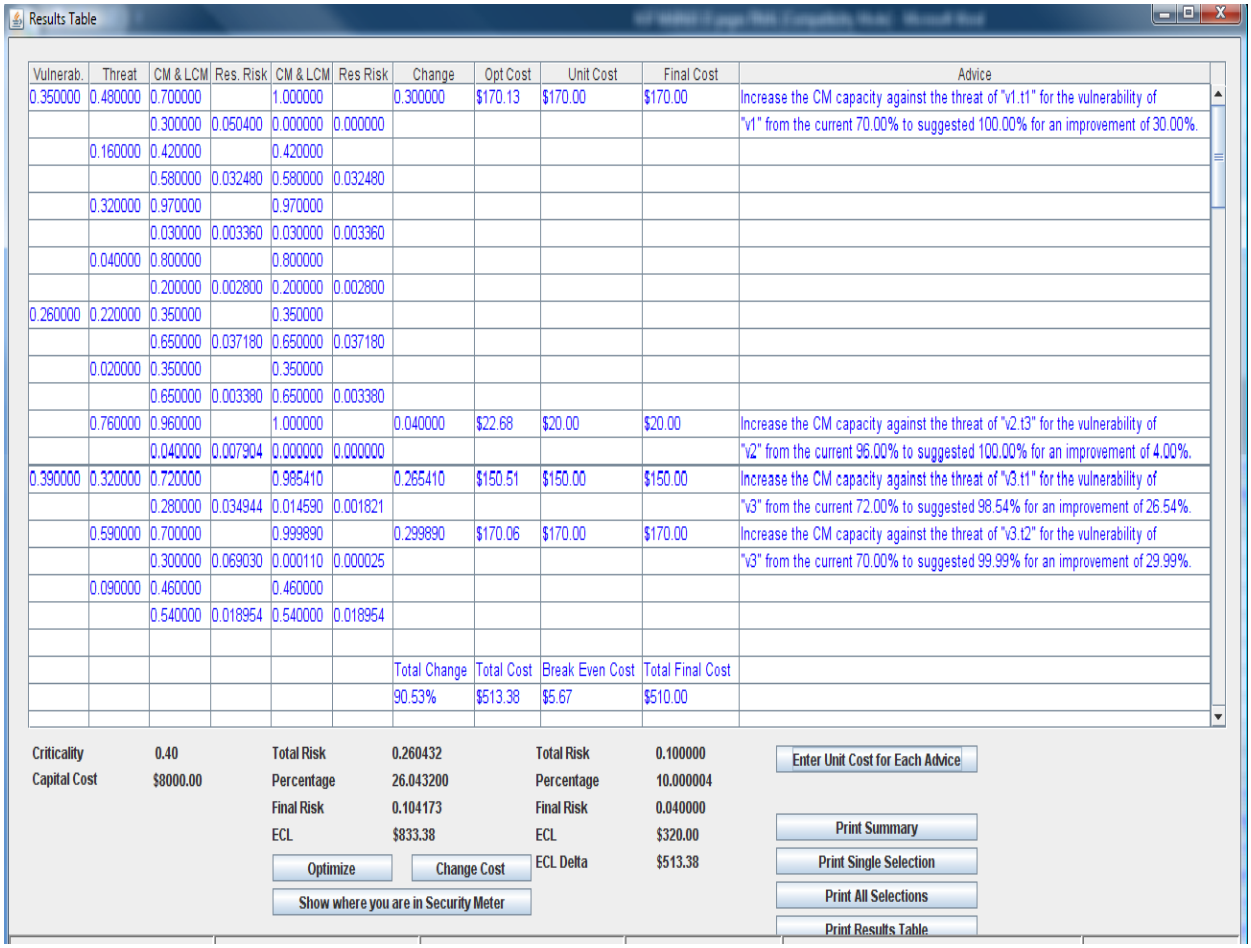


Figure 4: Example of Game Solution-Cost Optimal Health Care Risk Management for Table 1 and Fig. 3

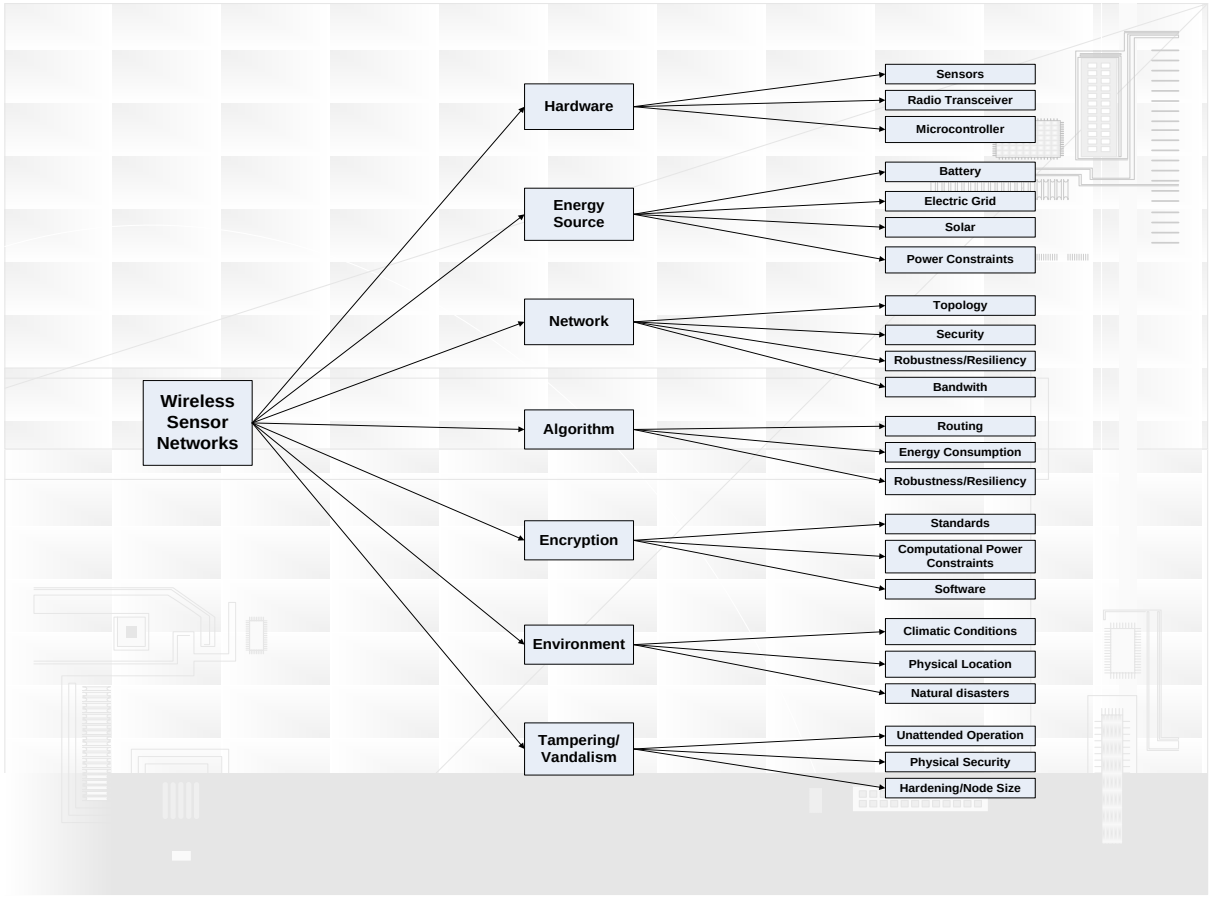
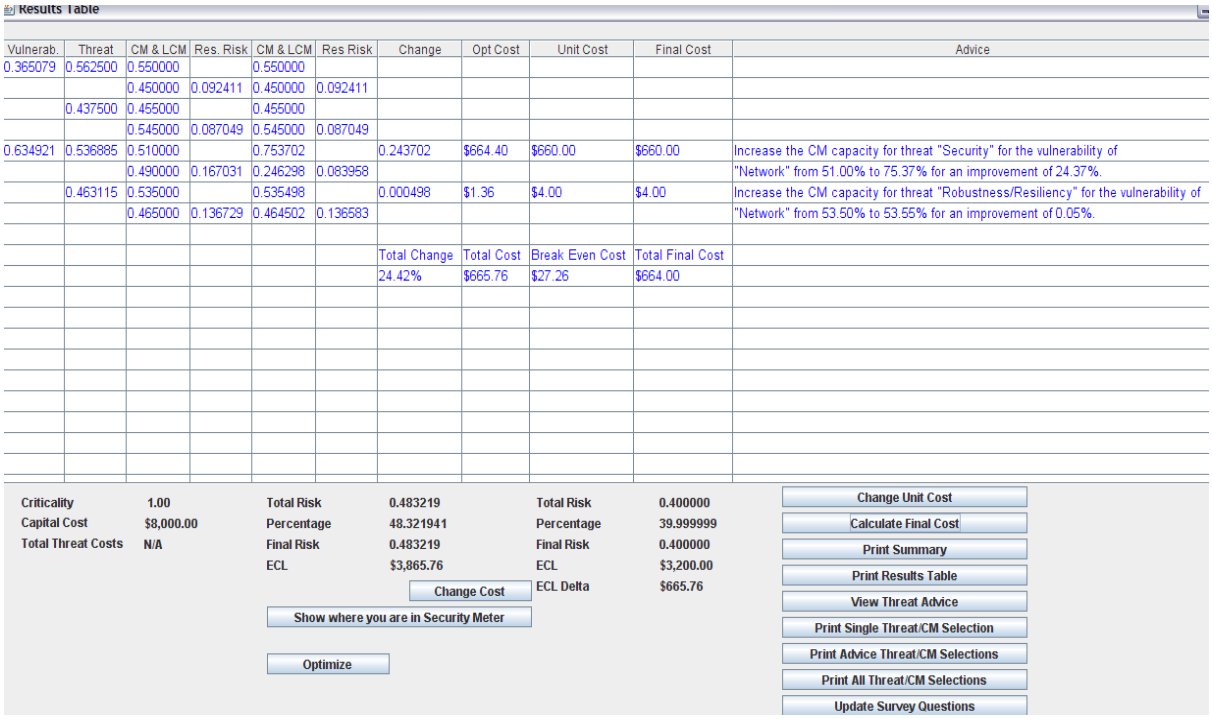


Figure 5. 'Wireless Sensor Networks' Security Meter's tree diagram



Results Table

Vulnerab.	Threat	CM & LCM	Res. Risk	CM & LCM	Res Risk	Change	Opt Cost	Unit Cost	Final Cost	Advice
0.365079	0.562500	0.550000		0.550000						
		0.450000	0.092411	0.450000	0.092411					
	0.437500	0.455000		0.455000						
		0.545000	0.087049	0.545000	0.087049					
0.634921	0.536885	0.510000		0.753702		0.243702	$664.40	$660.00	$660.00	Increase the CM capacity for threat "Security" for the vulnerability of
		0.490000	0.167031	0.246298	0.083958					"Network" from 51.00% to 75.37% for an improvement of 24.37%.
	0.463115	0.535000		0.535498		0.000498	$1.36	$4.00	$4.00	Increase the CM capacity for threat "Robustness/Resiliency" for the vulnerability of
		0.465000	0.136729	0.464502	0.136583					"Network" from 53.50% to 53.55% for an improvement of 0.05%.
						Total Change	Total Cost	Break Even Cost	Total Final Cost	
						24.42%	$665.76	$27.26	$664.00	

Criticality	1.00	Total Risk	0.483219	Total Risk	0.400000	Change Unit Cost
Capital Cost	$8,000.00	Percentage	48.321941	Percentage	39.999999	Calculate Final Cost
Total Threat Costs	N/A	Final Risk	0.483219	Final Risk	0.400000	Print Summary
		ECL	$3,865.76	ECL	$3,200.00	Print Results Table
			Change Cost	ECL Delta	$665.76	View Threat Advice
		Show where you are in Security Meter				Print Single Threat/CM Selection
		Optimize				Print Advice Threat/CM Selections
						Print All Threat/CM Selections
						Update Survey Questions

Figure 6: WSN Safety Risk Assessment and Management Solution Example using Security Meter Algorithm

Pollination in MAIDS: Detecting and Combating Passive Intrusions in a Multi-Agent System

Jeremy Kackley
Noetic Strategies, Inc
P.O. Box 22225
Huntsville, AL 35814
jeremy.kackley@gmail.com

James Jacobs
Jackson State University/CDID
1230 Raymond Rd.
Jackson, MS 39204
jjacobs@c-did.com

Paulus Wahjudi
Weisberg Division of Engineering & Computer Science
Marshall University
Huntington, WV 25755
wahjudi@marshall.edu

Jean Gourd
Louisiana Tech University
Center for Secure Cyberspace
P.O. Box 10348
Ruston, LA 71272
jgourd@latech.edu

ABSTRACT

In our previous work [7], we introduced a Multi-Agent Intrusion Detection System (MAIDS) aimed at detecting attackers through the observation of anomalous data. When detecting attacks that originate from passive nodes (that essentially just observe), relying upon anomalies presents a major weakness. In this paper, we extend our framework by integrating a novel process we call *pollination* that allows for the traceback of an agent's path in the network by leaving evidence of migration on both the agent and the node. As this work is fairly new, we provide a high-level overview and discuss how such a process might work within the context of our framework in order to detect passive attackers.

Keywords

Mobile Agents, Distributed Computing, Cyber Security

1. INTRODUCTION

In our previous related work [7], we introduced a mobile agent framework that assisted in detecting and combating compromised network nodes. A weakness in this infrastructure is the inability to detect compromised nodes that are passively intercepting information.

The ability to identify passive attackers, compromised network nodes and compromised agents and systems in our Multi-Agent Intrusion Detection System (MAIDS) requires the ability detect anomalies and changes in the mobile agent network and associate these changes with a specific threat. Several techniques exist in the mobile agent research community to eliminate the ability of the network to change by hardening the network. Other techniques rely on the ability to detect, trace and eliminate the anomalies in either the network or the agents.

Much of the existing research in mobile agent security is focused on hardening the agent or the agencies against attack (e.g., in [9, 10]). The focus is primarily on encryption or encapsulation techniques to harden the agent or agencies from attackers. The problem with encryption or hardening techniques is that they only buy time before the information is decoded or vulnerabilities develop thereby compromising the mobile agent network. The general technique to buy extra time is to change the defensive strategy more quickly than the time it takes to compromise the system. The inherent problem with this approach is the inability to detect a failure in the system in a timely manner. For example, a passive node could simply collect data for a period of time long enough to break the defense and after which have little chance of being detected.

Other techniques for mobile agent protection rely on identifying and eliminating the threat. For example, Khan et al. in [8] introduce an identification mechanism that watermarks the agent or the transmission of the agent. Other techniques include marking the agent with packet tags (e.g., in [1, 6, 11]). These mechanisms–among numerous others (see [3, 5])–can be used to passively identify the movements of agents by following the traces they leave in order to reconstruct a path back to the source of the anomaly. The problem with these techniques is one of limited focus in that they do not look at patterns found in the entire network. They merely compare changes of individual instances in the network. In order to address passive nodes, the ability to ascertain changes in agents, the movement of agents, agencies, and the intent of agencies must be considered.

Further tracking requires active involvement and transmission of activity to determine a mobile agent's status [12]. Although active tracking does give a better picture of the

mobile agent network status, the active scheme mandates communication overhead.

2. FRAMEWORK OVERVIEW

The framework introduced in [7] and illustrated in Figure 1 presented MAIDS, an agent based framework for detecting compromised platforms. The key aspect of this framework is the concept of threat levels, and it provides an effective method to detect and combat compromised platforms. Threat levels correspond to a global view of how dangerous the current situation is; furthermore, they serve as a controlling factor for the operation of the framework. These levels range from *Threat Level One*, which can be considered "situation normal" and where strictly passive observation occurs, to *Threat Level Four*, where action is taken against suspected nodes. The progressive network threat levels allow for dynamic and adaptive detection with varying degrees of response. Once a node is suspected of being compromised, it (and any nodes that come in contact with it) will be thoroughly investigated before any verdict is given. Depending on the algorithm used to observe the network traffic, MAIDS can be adapted to observe the smallest piece of data or it can focus the network as whole. However, MAIDS is dependent on one key element: the active data packets sent by the compromised platform that the framework can then intercept and detect for irregularities. In the event that a compromised platform is passive and focuses only on intercepting information routed through it without actively sending packets to other nodes as an attempt to obtain information and/or to infect other nodes, MAIDS will never suspect that node as being compromised. This weakness will be addressed in this paper through the utilization of a Mobile Agents Pollination (MAP) technique. The following provides a brief overview of each threat level; for more detail, the reader is referred to our previous work in [7].

2.1 Threat Level One: Network Observation

This threat level corresponds to normal network situation and is the default. The important action that takes place at this threat level is the establishment and maintenance of a network of *Probe* agents. These *Probes* can be thought of as a distributed set of eyes and ears in the network. This level also sees the establishment of a Central Authority Node (CAN) that serves as the nerve center of the framework. As agents percolate through the network, they carry reports generated by the *Probes*. These reports are ultimately delivered to the CAN which makes judgments based upon them. In this way, the CAN monitors and maintains a view (possible delayed) of the entire network at a relatively low cost to performance. This view can be used to search for anomalies, such as a disproportionate number of agents arriving to those leaving a given node. Anomalies are domain-dependent. A certain level of anomalies is expected as a by-product of network behavior, thus a threshold value T_1 is defined for this threat level that indicates the maximum amount of anomalies to be expected in a non-compromised network. Rates above this value constitute an elevation of the threat level.

2.2 Threat Level Two: Network Suspected Compromise Investigation

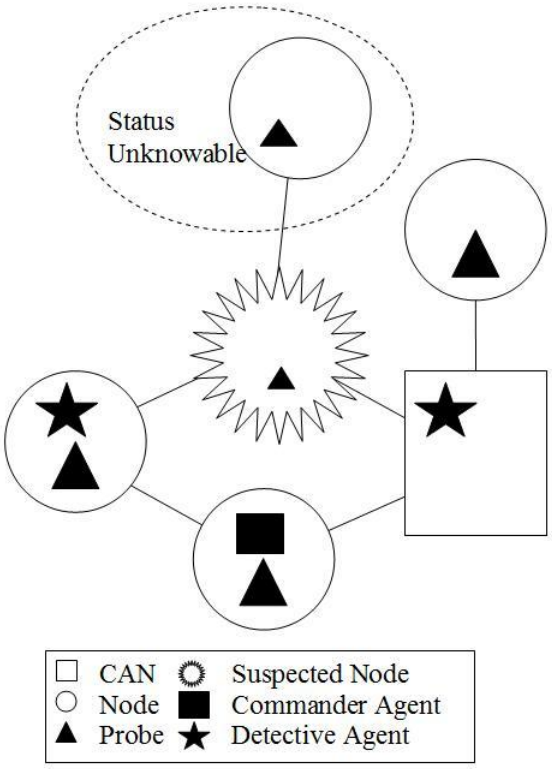


Figure 1: The MAIDS Framework

At this threat level the network is suspected to be compromised. This triggers the CAN to generate two agent types: a *Commander* agent and a *Detective* agent. *Commanders* are akin to a localized CAN, the purpose of which is to reduce report latency. *Detectives* are proactive versions of *Probes* that communicate observations directly to their *Commander*. The objective is to blockade the suspected node(s) and investigate incoming and outgoing traffic to determine if anomalies are still occurring at a level greater than the threshold value. Inherently, there is a network effect whereby any node that can only communicate through a suspected node is, of course, also suspect and cannot be trusted (see Figure 1). Thus, the virtual blockade could indeed comprise a major section of the network. The CAN takes into account this network effect when placing *Detectives* so as not to compromise the aggregate data. Another important point is that this is merely an investigative roadblock; communication is investigated and monitored but is not stopped. Again, anomalies are domain dependent, but it makes logical sense that there would be more types of anomalies defined at this level. Additionally, T_2 is the second threshold value of anomaly detection prior to elevation to Threat Level Three.

2.3 Threat Level Three: Network Compromise Confirmation

This threat level sees the creation of an additional type of agent, the *Secret Agent*, that is essentially something of a sacrificial agent. Its actions (and the expected results thereof) are predefined; therefore, it can be sent to a com-

promised node, and if the results are not observed exterior to the node or the communication of its observed effects do not match the observations of *Detectives*, then an inference can be made that a compromise has occurred. It is possible that the *Secret Agent* will never be heard from again, in which case this process must be repeated until either the agent survives or a set number of agents have been sacrificed. It is also possible at this threat level to either elevate to Threat Level Four, or to deescalate if the *Secret Agent* is not interfered with.

2.4 Threat Level Four: Network Compromise Resolution

For the most elevated threat level, the assumption is that a compromise has occurred. At this point, there are a variety of actions that can be taken. The appropriate action is very domain dependent; for example, if resource availability is more important than information security then simply alerting a human while continuing to gather information is the appropriate response. Alternatively, if information security is more important than availability or redundant resources exist, then automated responses are possible. The least severe response would be rerouting requests from the compromised node to a sandbox for future analysis and to prevent the compromise of the rest of the network (presumably without making it obvious to the attacker that he has been detected). A more severe action would be to blockade the node from the network, thereby preventing any requests from leaving or going to that node. The most severe would be attempting to remove the node from the network or possibly crash it (for example via a distributed denial of service attack or some out-of-band signal).

3. POLLINATION

The proposed pollination scheme in this work is a passive system that allows minimum overhead with active monitoring to provide near real-time discovery of the mobile agent network status. Pollination involves the exchange of *pollen* between the mobile agent and the agency on a node to provide a tracking and a pattern mechanism for use with inference modeling. The pollen allows the tracking of an individual agent's movements and intentions, and the pollination patterns in both the agent and the node allow for network and agency status to be inferred. The inference model then classifies the intent, and the security protocols in MAIDS will be enacted based on the perceived intent. Scaling of the pollination model allows for overhead to be minimized to the level of the threat.

The concept of pollination is designed to create a series of trail markers on both the nodes visited by a mobile agent and the mobile agent itself. The trail markers allow immediate identification of what nodes the agent has visited by simple inspection of the pollen the agent is carrying. The inspection of an agent's path via the pollen is performed at its destination. By traversing the trail of pollen back to the source node one can trace the agent's migration history.

The information provided by pollination is meant to be both historical and active. Historical information can be used to determine the sequence of events after an event has occurred. Active information is obtained from real-time inspection and is used to determine if an event has occurred. For example, in many cases the data and the code that processes it are not singly sensitive. However, the ability to simultaneously obtain both the data and the code has the potential to cause harm to the agent's designer. If both nodes are marked, and each node in the network is sensitive to this situation, a mobile agent containing pollen associated with both nodes can be apprehended.

Security in information systems is an important aspect for all applications. Such security covers three main components: data security, machine security and network security. One of the key issues involves the authentication of an entity in relation to its access to various resources. A trusted entity may become compromised, and thus untrustworthy, despite being positively identified. Determining if an entity has been compromised is an important but complicated process. The pollination concept introduced in this work extends the capability of MAIDS to detect compromised platforms that passively intercept information.

3.1 Mobile Agents Pollination

Mobile Agents Pollination (MAP) is a concept that addresses some of the issues with the MAIDS framework, namely that of detecting passive attacks, by identifying a mobile agent's movements and actions within the network. MAP uses *pollen* to uniquely identify the agencies or groups of agencies on nodes within a network. Furthermore, MAP uses pollination to form a trail map defining the path an agent utilizes when visiting nodes in the network. In addition to the trail map, the pollen and map properties can be utilized as action indicators to infer the meaning of the agent's visitations and the intent of agencies within the network.

MAP is similar to the natural process of flower pollination by bees. Bees traverse a field of flowers to acquire nectar. When bees encounter a flower, they inadvertently collect pollen from it and distribute pollen to it. Fundamentally, a grain of pollen represents a flower and a collection of pollen on the bee represents all the flowers it has visited. The pollen collection left by the bee at each flower represents the sequence of flowers it has visited before encountering the current flower. The analogy relates to MAP in that the mobile agent is the bee, the nectar is the information (or data) being collected by the agent and the pollen is a node's identification marker. The role of the mobile agent is to traverse the network of connected nodes to acquire information. MAP defines the process where the mobile agent *unknowingly* collects pollen from the current node and distributes pollen from the previously visited nodes. The pollen can then be used to quickly infer where the mobile agent has traveled and the sequence of travel. Other traits related to the map, the pollen, and the pollen's relationship to the nodes can be utilized to expand the perceptibility of the mobile agent's activity in the network.

The main purpose of MAP is to utilize pollination for tracking a mobile agent's activity in the network. We can then use this tracking information to infer the intent of the mobile agent and the nodes involved in the network. Ultimately, pollen is a marker used to identify a specific node a mobile agent may visit; pollination is the process of exchanging pollen to provide a mechanism to reconstruct where the

agent has been and infer the actions the agent performed.

Pollen can be unique to each node or the pollen can be unique for a subset of nodes depending on user requirements. The pollen variation throughout the network is similar to the DNA of a flower. The DNA for a specific node type is mostly the same between entities with minor variations making the sequence unique. The pollen from two different node types will have a greater variation in the predominating factors. The pollen strictly provides the identification of a node and should be dependent on the agency configuration to make the process of spoofing it difficult. The information provided by pollination is meant for both historical and active analysis. Historical information can be used to determine the sequence of events after an event has occurred. Active information is used to form real-time inspection in order to determine if an event has occurred.

There are two key observable parameter categories involved in the process of a mobile agent collecting information while traversing the network and visiting nodes: spatial and temporal. Within MAP, a spatial dimension refers to the network space in the path taken by the mobile agent, the endpoint used by the node, the connection, and the pollen. Furthermore, this can include the distance of the agent's travel and inter-node distance. Both are employed by MAIDS.

The network's spatial reference is depicted using standard network nodal reference with connections showing the relation between the nodes. The pollen associated with the nodes and the agents is represented by unique patterns. The pollen grains carried by the mobile agent and distributed to the nodes maintain the pattern of the origin node. The sequence of pollen grains creates a series of trail markers which define the spatial movements of the agent through the network. A direction depicted by the arrow is inherent in the sequence as the agent migrates from one node to another. The pattern allows immediate identification of what node(s) the agent visited by simple inspection of the pollen the agent is carrying. Traversing the trail of pollen defined by the sequence will lead back to the source node.

Within MAP, the temporal dimension is used to refer to the amount of processing a mobile agent expends at a node. Just as a bee accumulates more pollen the longer it sits at a flower, agents accumulate more pollen the longer they remain at a node gathering data or performing computations. This may be considered a density in the pollen an agent carries (more *grains* of pollen). The pollen density is associated with the pollen unique to a specific node, and the combination of all the pollen densities will indicate the time taken for the sequence to complete. A common time reference will maintain a standard gauge for use with the analytics within the MAIDS framework. The actual time interval is up to the implementation; however, the granularity of the interval should represent the difference between operation actions of the mobile agents for the operations we want to identify. The temporal reference is depicted by a count below the grains.

We acquire a time-sequence pattern by applying both the spatial and temporal components. The time-sequence pattern maps the network to the goal of the mobile agent's

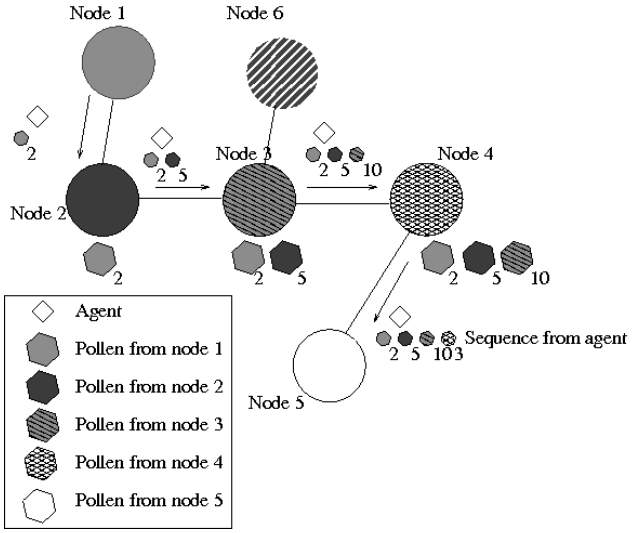


Figure 2: Pollination and Mobile Agents

movement. The spatial, temporal and spatial-temporal observations are used to infer the meaning of the action with regard to the intent of the agent and nodes. Associating the content description of the nodes with the pattern may allow for further identification of a mobile agent's actions.

The process of pollination leaves two distinct time-sequence trails as the mobile agent moves from node to node. The first trail is the set of pollen attached to the agent from the simple process of visiting nodes. The second is the set of pollen distributed along the path of nodes traveled where each node has a snapshot of the previous nodes the agent has visited. The sets have a number of key attributes that include node references, number of pollen grains, sequence of grains and the order (or pattern) of nodes visited, and amount of pollen attached. The pollination concept is depicted in Figure 2.

In this figure, an agent is migrating from Node 1 to Node 5. At each node, the agent exchanges pollen in the process of "doing work." The amount of work is quantified with the temporal reference. The agent reaches Node 5, and the resulting pattern is depicted with the temporal references denoted by the vector $\langle 2, 5, 10, 3 \rangle$. Note that Node 6 is not in the agent's path and therefore no pollen is exchanged with this node or the agent. This example serves as a base pattern or a composite of patterns that can be used to determine anomalies that occur with the agent's or node's standard operation.

3.2 Usage of Mobile Agents Pollination in MAIDS

In order to address passive attacks we simply need to have analytics to infer the meaning of the change in pattern. For example, suppose that a passive node successfully inserts itself into the network. The node will have to understand that pollination occurs otherwise it will not be accepted. We assume the attacker is smart enough to spoof a node (and understand that pollination occurs but not necessarily how it works) and continue with more advanced security concerns.

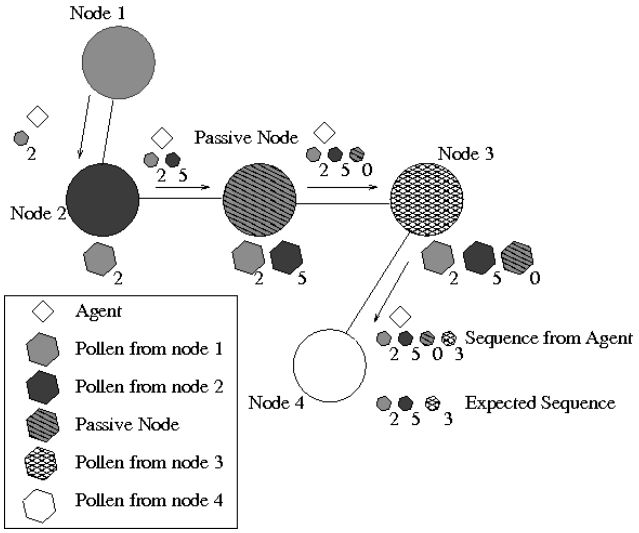


Figure 3: Network with Passive Node

Suppose we mark the path agents would take from Node 1 to Node 4 (see Figure 3) by marking each of the intermediate nodes with different pollen. Every time an agent reaches Node 4, it possesses a sequence of pollen that corresponds to the path taken. This is verified with the sequence that the agent was supposed to follow in order to determine if the agent has been compromised by an additional passive node or a violation of the agent code. The passive node is identified by added pollen, incorrect pollen or simply the lack of pollen in the sequence inspected at the destination node. A team of *Commanders* and *Detectives* can be used determine which node is the passive node and eliminate it. Compromised agents can simply be eliminated from the system at Node 3 upon arrival and trigger the inspection. For added security, every node along the path checks the pollen sequence and essentially provides a passive defense mechanism against corruption.

3.3 Implementation

The pollination process works at the application level where the agencies represent the nodes and mobile agents move throughout the network migrating from node to node. The nodes each have their own unique pollen definition. Implementing the pollen requires the ability to attach pollen and transport it with the mobile agent; it is *picked up* and *dropped off* by the nodes. We propose that the process of attaching the pollen to the mobile agent is not performed by the agent itself in order to maintain a correlation to the natural process of pollination (with bees and flowers). The agent should not know or care about how pollination works. We envision the use of a manipulation of the Open System Interconnection (OSI) [13] model's transport layer for both attaching the pollen to the agent and transporting the pollen and agent to the destination node. In the OSI model, the application data to be transported is broken into packets and transmitted from source to destination. We can add additional packets by appending the pollen to the data stream, or we can manipulate the packets using packet tagging (e.g., [2, 4]). Adding additional packets can be accomplished at the application level by simply appending to the end of the mobile agent in that data stream. The addendum is removed at the node and the pollen is recovered.

Packet tagging is accomplished by marking packets with identifiers for local purposes. The actual mark is part of the packet and cannot be removed; however, the mark can be modified or replaced with another mark. This may be an issue if the agent is transmitted to some location that is external to the system (outside the view of the MAIDS framework). However, for our purposes the mobile agent is assumed to remain internally within the network.

In addition to the pollen tag itself, the count representing action at each node can be used to acquire further insight into the meaning of the agent's intent. It is expected that a node with little information to share will require less time for the mobile agent to visit the node (i.e., less activity). Using the activity gauge, we can infer some of the intent of both the agent and the node. In the example of a passive node scenario, we can expect the sharing to be minimal for the agent (as described by the label "0" in Figure 3) with regard to the passive node's pollen.

The reverse is also possible with the relationship between the node's pollen and the data stored at the node. Using this relationship between the information of the node with the highest activity, we can infer the type of information the agent is interested in gathering and the information with little interest. The entire concept is depicted in Figure 3. The purpose of pollination is to allow an easy identification method for activity within a multi-agent system using pollination patterns. Any standard inference model (e.g., fuzzy logic, neural network, Bayesian model) can be utilized to trigger MAIDS security events from the pollination patterns.

A variation of the scheme can be implemented to acquire different levels of security throughout the network. At a low level we are only concerned with sensitive data or applications. For this we simply need to pollinate those locations and track the movement of agents carrying that pollen. Nodes are always active to interpret the meaning of the mobile agents and the surrounding nodes. The limitation of the pollination to a subset of the network has the effect of greater focus-ability on only the things that matter. This, in turn, has the effect of reducing the overhead associated with pollination in both time and space. Furthermore, we can change the pollination patterns associated with the network on a periodic basis to ensure security. This change can either be notified to the MAIDS security team in advance or it could simply trigger an event that allows the team to determine the appropriate action. The latter is preferable as the addition of a mechanism to disable security for changing patterns could be an exploitable vulnerability. The event triggered by the pattern change can be used as a test of system integrity as the process goes through the threat levels and back to Threat Level One.

The state of the system primarily resides at Threat Level One; however, as security concerns increase, the number of pollinated nodes increases to match the threat. The increase can either be tactically focused on the relative sensitivity of the data or it can be distributed throughout the network to

provide a "big picture" of the secured environment. At the highest level all nodes will be pollinated, and nodes without pollen or agents not containing pollen will be apprehended. The pollination paths not adhering to the required patterns will be examined to determine what events took place by the MAIDS security team.

4. DETECTING PASSIVE NODES WITH POLLINATION

Pollination is a tool that provides input into the MAIDS framework. Fundamentally, this input provides an additional data set that can be used for anomaly detection. For instance, pollen appearing from "new" nodes that should not exist may indicate an intruder connected to the network wirelessly. Alternatively, a drastic and sudden change in traceroutes for the network may indicate abuse. These anomalies feed naturally into the existing MAIDS threat levels.

In addition to pollen serving as a data source, traps can periodically be set in order to lure passive nodes to actively search for phantom prized data that, in turn, will expose their cover. As part of this process, the CAN will elevate the threat to Threat Level Two and randomly select a set of strategic nodes of interest (SNI) throughout the network as the host of the prized phantom data. Each trap will have a designated area of effect that determines the number of nodes that are affected. The CAN will then send agents to each node with the objective of broadcasting the existence of crucial data in the SNI. When a request for the phantom data is received at the SNI, the network threat will be elevated to Threat Level Four and the CAN will backtrack the pollen pattern to the originating host, marking it as a suspected compromised host.

4.1 The "Mole" Scenario

We have illustrated the use of pollination within MAIDS to assist in the detection of passive nodes (and attackers). However, there are many other uses for pollination; such a method can be used to detect other kinds of attacks. Take, for example, an insider threat and consider the following scenario: an organization's management is noticing that its main competitor frequently releases a product that is very similar to and directly competing with releases of their own– and it is doing so a few weeks prior to its own release dates. One might suspect that an insider is leaking information (and possibly being paid for it). How might the management of this organization determine if this is indeed happening and, if so, who is leaking the intellectual property?

By implementing MAIDS with pollination within the organization's network, we would be able to detect such an event and narrow down the leak to a subset of nodes (and possibly determine who the leak is). The solution requires an approach from two directions: from within the MAIDS framework and from a social engineering standpoint. One technique to identify an insider threat might be to announce the forthcoming release of a fictitious product whose details are located on a server (or servers) that resides at a particular node (or a subset of nodes) in the network. As previously stated, the CAN will randomly select a set of SNIs throughout the network that will host the fictitious information. Clearly, the fact that the product is fictitious and will never actually be released is something the management will conceal; this is the social engineering component of the approach. The idea is to entice the leak to obtain the details of the product while using the MAIDS framework with pollination to identify him.

Pollen is something that uniquely identifies nodes; so to determine the agents that are accessing the fictitious information, we simply generate a unique pattern on the SNIs that provide this information. There is no need to pollinate from other nodes in the network, as we are uninterested in them at the moment. Now we simply need to track those agents that are carrying the pollen. Once pollen is being carried around by the agents and is stored on the nodes in the network, we can inplement a set of traps that will assist in detecting the leak. Since nodes are always active to interpret the meaning of the agents and the surrounding nodes, MAIDS can focus on the leak. Agents sent by the CAN roam through the network in search for the pollen. When they find it, the CAN will raise the network situational awareness to Threat Level Two and proceed to investigate the path taken by the agent to reach the SNI. Instead of attempting to determine compromised nodes, *Commanders* and *Detectives* will be looking to locate the source of the information request in order to extirpate the source node–and thus the insider threat.

4.2 The Packet Sniffer Scenario

Consider the case when an attacker simply wishes to passively sniff for interesting packets in the network. In order to do so, the attacker must successfully (and secretly) insert a node in the network. In the case that the attacker has no idea that pollination is being utilized, he will be easily detected. As agents go through the attacker's node, they will go through un-pollinated, and this will be detected at some other node in the agent's path. This is illustrated in Figure 3.

But suppose that the attacker has some knowledge that a pollination scheme is being implemented in the network. The pollen being generated at his node will be unrecognized in the agent's pollination sequence (i.e., it is not pollen that the framework knows anything about). And again, the attacker will be easily detected.

In the most unlikely case, an attacker may clone the pollen being generated at some other (legitimate) node in the network. Since pollen for a node is randomly generated (and may change frequently), it is very unlikely that this will happen; however, this occurrence can still be detected by MAIDS as the pollen sequence will contain the same pattern more than once in the vector (and out of order).

5. CONCLUSION AND FUTURE WORK

The MAIDS framework previously introduced (in [7]) provides the ability to detect, investigate and deal with compromised nodes utilizing a multi-agent scheme in an effective and efficient network intrusion mechanism. However, MAIDS relies heavily on nodes that actively attempt to obtain data that they are not entitled to (and tipping their hand in the process). In the event that an intruder remains passive and only collects data that it unwittingly comes across, MAIDS is unable to detect it. To address the de-

tection of passive attackers, incorporating pollination into the MAIDS framework was proposed. As an added bonus, pollination also provides additional forensic capabilities in that a reconstruction of what occurred (i.e., where an agent migrated throughout the network) is possible.

As this work is still very new, there are a large number of areas we may explore in order to further improve MAIDS. The role of pollen can be expanded and could, for example, affect an agent's functionality or modify the data it collects thereby rendering it useless. When a node encounters pollen it does not recognize, it can either refuse the agent, respond to the agent's request (should it make one) with misinformation, or cease future communication with the node that supplied the pollen. Alternatively, we could embed information in the pollen via a predetermined set of pollen "colors" or patterns. These could represent different hidden emergency messages that can be relayed by each host to the CAN. This simple coloring scheme can act as a silent alarm that notifies the CAN for a possible breach in one or more nodes.

In reference to the purposeful setting of traps in order to lure potential passive attackers, we noted that when such a trap is sprung, the threat level elevates from Threat Level Two directly to Threat Level Four. This is somewhat drastic, but it is difficult to imagine a way of further confirming that a passive attacker exists without human intervention. Additionally, the amassed evidence will be fairly incriminating. It may eventually be possible to engineer a *Secret Agent*, as in Threat Level Three, capable of further investigation in an attempt to confirm this. It is unclear what exact form this would take, but it merits investigation.

As noted, the purpose of pollination is to allow an easy identification method for activity within a multi-agent system using pollination patterns. Any standard inference model (e.g., fuzzy logic, neural network, Bayesian model) can be utilized to trigger MAIDS security events from the pollination patterns. Such a discussion of models (and perhaps the *best* model) is not provided in this paper but merits attention in the future.

6. REFERENCES

[1] B. Al-Duwairi and G. Manimaran. A novel packet marking scheme for ip traceback. In *ICPADS 2004: Tenth International Conference on Parallel and Distributed Systems*, pages 195–202, 2004.

[2] A. Belenky and N. Ansari. Tracing multiple attackers with deterministic packet marking (dpm). In *PACRIM: IEEE Pacific Rim Conference on Communications, Computers and signal Processing*, volume 1, pages 49–52, 2003.

[3] Y. Bhavani and P. Reddy. An efficient ip traceback through packet marking algorithm. *International Journal of Network Security and Its Applications*, 2(3):132–142, July 2010.

[4] Y. Djemaiel and N. Boudriga. A global marking scheme for tracing cyber attacks. In *Proceedings of the 2007 ACM symposium on Applied computing*, pages 170–174, 2007.

[5] Z. Gao and N. Ansari. Tracing cyber attacks from the practical perspective. *IEEE Communications Magazine*, 43(5):123–131, 2005.

[6] M. Goodrich. Efficient packet marking for large-scale ip traceback. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 117–126, 2002.

[7] J. Kackley and P. Wahjudi. Detecting and combating compromised platforms in a mobile agent infrastructure. In *Proceedings of the 2nd Cyberspace Research Workshop*, pages 35–41, June 2009.

[8] A. Khan, X. Niu, W. Anwar, and Z. Yong. On the security properties and attacks against mobile agent watermark encapsulation(mawe). In *SMC 2008: IEEE International Conference on Systems, Management and Cybernetics*, pages 707–712, October 2008.

[9] H. Lee, J. Alves-Foss, and S. Harrison. The use of encrypted functions for mobile agent security. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, January 2004.

[10] D. Lin and T. Huang. A mobile-agent security architecture. In *2nd Internationoal Conference on e-Business and Information System Security (EBISS)*, pages 1–4, May 2010.

[11] D. Peng, Z. Shi, L. Tao, and W. Ma. Enhanced and authenticated deterministic packet marking for ip traceback. In M. Xu, Y. Zhan, J. Cao, and Y. Liu, editors, *Advanced Parallel Processing Technologies*, volume 4847, pages 508–517. Springer Berlin / Heidelberg, 2007.

[12] R. Sreedevi, U. Geeta, U. Kulkarni, and A. Yardi. Enhancing mobile agent applications with security and fault tolerant capabilities. In *IEEE International Advance Computing Conference*, pages 992–996, March 2009.

[13] D. Wetteroth. *OSI Reference Model for Telecommunications*. McGraw-Hill Professional, 2001.

Micro-Aerial Vehicle and Sensor Networks Laboratory Development and Applications *

Miguel D. Gates
Louisiana Tech University
Engineering
Nethken Hall 232
Ruston, LA 71272
mdg022@latech.edu

Christopher Barber
Louisiana Tech University
Electrical Engineering
Nethken Hall 232
Ruston, LA 71272
crb050@latech.edu

Rastko Selmic
Louisiana Tech University
Electrical Engineering
Nethken Hall 229
Ruston, LA 71272
rselmic@latech.edu

Christian Duncan
Louisiana Tech University
Computer Science
Nethken Hall 141
Ruston, LA 71272
duncan@latech.edu

Jinko Kanno
Louisiana Tech University
Mathematics
George T. Madison Hall 350
Ruston, LA 71272
jkanno@latech.edu

ABSTRACT

This paper provides a summary of the multitude of capabilities for Louisiana Tech University's Micro-Aerial Vehicle and Sensors Networks (MAVSeN) laboratory. The laboratory is oriented around research endeavors involving Micro-Aerial Vehicles (MAVs) and Wireless Sensor Networks (WSNs). Both fields have been a hot-button topic recently as a myriad of applications have been developed for both military and civilian efforts. This paper highlights the features of the MAVSeN lab and how it incorporates with the aforementioned research fields. Equipped with the latest high-speed motion capture technology, the foundation of the MAVSeN lab are the multiple Vicon cameras that allow for numerous objects tracking and localization in a real-time coordinate system with the highest level of precision and accuracy. This becomes an invaluable resource in many applications centered on localization, tracking, positioning, and control. It also allows future study in the areas of mobile sensor networks, deployment, MAVs, remote sensing and layered sensing.

Keywords

Cooperative Control, Micro Aerial Vehicles, Position Adaptive RF Direction Finding, Wireless Sensor Networks

1. INTRODUCTION

Micro-Aerial Vehicles (MAVs) and Wireless Sensor Networks (WSNs) are key components to advance the warfighters situational awareness on the modern urban battlefield. MAVs are a new area for innovative experimental platform development that holds great possibilities for large and small scale real-time battlefield awareness analysis. The small size and portability of MAVs make them a great asset to troops in the urban battlefield. The ability of a single or a cooperative team of MAVs to survey an area of interest, search a building before the deployment of troops for explosives or insurgents will help to reduce the number of unnecessary casualties during battle. The capabilities of WSNs are far reaching into providing battlefield commanders with a small and energy efficient method of monitoring battlefield situations. WSNs have the ability to form ad-hoc networks and the nodes that these networks are composed of are capable of being interfaced with various sensors or controlling actuator systems, physiological status of troops or monitoring areas for insurgent activity, and the detection of radiological or biological agents.

For this very reason, the MAVSeN lab has been designed specifically for the purpose of research in small-scale aerial vehicle design, cooperative intelligent sensing, and control algorithms of such platforms for various applications. Besides the United States Air Force (USAF), several universities also have similar systems working for them in various research fields, from swarm behavior to autonomous target tracking. The Louisiana Tech University MAVSeN Laboratory has unique capabilities for experimenting with swarms of MAVs and sensor networks, in both layered and cooperative sensing concepts. The laboratory setup provides a high-speed and high-resolution motion capture system that emulates indoor GPS, as shown in Figure 1. The motion capture system has the ability to track markers ranging from 3–24mm — any objects that can be fitted with 4 markers that are placed in positions are visible to the cameras. The laboratory consists of a Vicon motion tracking system and three Quanser based Qball MAV platforms contained in a (30ft x 12ft x 12ft) room. More descriptions detailing the laboratory equipment will be discussed in the upcoming sections. This paper is organized as follows: Section 2 illustrates the various components that construct the MAVSeN lab and how they integrate with each other; Section 3 de-

*This work was supported in part by the AFOSR grant

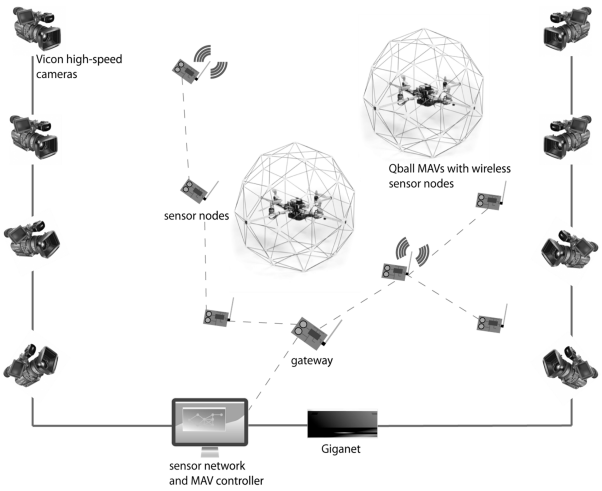


Figure 1: The MAVSeN Lab concept

scribes our current research and other capabilities able to be performed in the laboratory; and Section 4 concludes the paper and offer insights into future endeavors.

1.1 Background

A sensor network consists of a collection of sensing devices that can coordinate their actions through wireless communication and aim at performing tasks such as reconnaissance, surveillance, target tracking or environmental monitoring over a specific region.

A foundation in the area of MAVs working cooperatively with these sensor networks already exists. Li and Cassandras give a tutorial-style overview of sensor networks from a systems and control theory perspective, providing a comprehensive background [1] on sensor networks. Complementing this overview, they later presented a distributed coverage control scheme for cooperating mobile sensor networks [5]. They developed a gradient based algorithm requiring local information at each sensor and maximizing the joint detection probabilities of random events. Akin to this research, Cortes et al. presented control and coordination algorithms for groups of autonomous vehicle networks performing distributed sensing tasks where each vehicle plays the role of a mobile tunable sensor [4]. Bellingham et al. addresses to problem of cooperative path planning for a fleet of Unmanned Aerial Vehicles (UAVs) in uncertain or adverse environments, by modeling for the probability of UAV loss [2]. Similarly, Richards and How implemented a robust Decentralized Model Predictive Control (DMPC) for a team of cooperating UAVs [9]. Using this DMPC each vehicle plans only for its own actions, but still allowing the UAVs to communicate relevant plan data to ensure those decisions are consistent across the team. In a simple case like collision avoidance, DMPC guaranteed constraint satisfaction and offered significant computation improvement, compared to an equivalent centralized algorithm, for only a small degradation in performance, such as UAV flight time. Chandler et al. researched the development of cooperative rendezvous and cooperative target classification agents in a hierarchical distributed control system for unmanned aerospace vehicles [3]. For cooperative target classification he developed templates, followed optimal trajectories, and assigned adjacent vehicles to view at complementary aspect angles; hence, he combined these to maximize the probability of correct target classification over various aspect angles. Singh and Fuller developed a receding-horizon optimal control scheme for autonomous trajectory generation and flight control of an unmanned air vehicle in an urban terrain [10]. Because environments may be dynamic, or the vehicles need to change dynamics mid-fight due to sensor or actuator failure, they proposed a Model Predictive Control (MPC) scheme that navigates a vehicle with nonlinear dynamics through a vector of known way-points to a goal, and manages constraints for missions that will require vehicles with increased autonomy in dangerous situations and with tight maneuvering and operational capability e.g., missions in urban environments. Continuance and improvements of research in these various areas are the motivation behind the creation of the MAVSeN lab.

2. MAVSEN LAB

The MAVSeN lab was designed to provide a controlled laboratory environment for the research and advancement into the field of embedded mobile sensor platforms. The actual laboratory setup can be seen in the Figure 2. The lab itself is composed of many components, both hardware and software, that are vital in its functionality. These components include the following:

- Vicon motion capture cameras—with Tracker software
- Quanser Qball-X4 and DraganFlyer X6 Unmanned Aerial Vehicles (UAVs)
- MEMSIC IRIS wireless sensor nodes
- C++, Simulink, and MATLAB based software

2.1 Vicon MX T40 Camera System

The Vicon system consists of ten Vicon T40 cameras, as shown in Figure 3 configured with 12.5mm lenses that give an effective Field of View (FoV) of 66.7 by 51.6 degrees at a 5m camera distance [11]. Each camera is outfitted with a CCD sensor having a resolution of 2352 x 1728 (4,064,256) pixels. This provides a full resolution maximum capture rate of 370 fps for each camera. Surrounding each lens is a circular array of 252 near infrared (780nm) LEDs, which are adjustable in brightness to increase the sensitivity of the cameras for different lighting conditions. This enables the system to provide high precision tracking data with slow or high speed motion using markers of sizes ranging from 3 to 25mm in diameter, seen in Figure 4. These markers are placed on the objects that are to be tracked. The versatility of the Vicon T40 camera enables it to be able to track markers as small as 14mm in volumes as big as 10m x 10m x 4m. The Vicon camera has the ability to track an object within a millimeter of accuracy down to a hundreths of an inch in precision. This gives the system a sub-pixel resolution of around 1/50th of a sensor pixel, provided that the

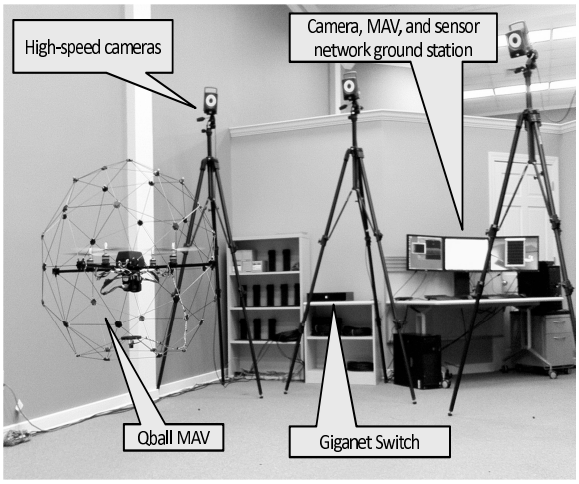


Figure 2: MAVSeN testbed setup at Louisiana Tech University

Figure 3: Vicon MX T40 Camera

full greyscale circle fit algorithm is used. The cameras are networked via gigabit cables that provide power, control, and data transfer from the cameras to the Vicon MX Giganet switch. The Giganet switch has connectivity for up to ten cameras and can be networked with additional switches if more cameras are needed. The captured camera data is then fed into the Vicon Server and displayed using the Vicon Tracker software for real-time analysis and capture. This data can also be transferred to the Quanser Workstation for flight algorithm position analysis in MATLAB, Simulink, and QUARC via a crossover cable connection.

2.2 Quanser Unmanned Aerial Vehicles

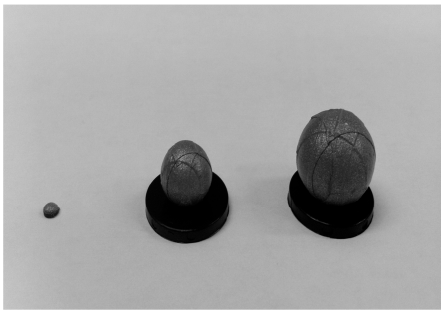


Figure 4: Fluorescent Tracking Marker—3mm, 9mm, and 14mm markers, from left to right

The Qball-X4 UAV, shown in Figure 5, is an innovative rotary wing vehicle platform suitable for a wide variety of UAV and UGV research applications[8]. It has a quadrotor helicopter design with four motors and speed controllers fitted with 10-inch propellers. The entire mechanism is enclosed within a protective carbon fiber cage, making it an ideal tool for basic vehicle navigation and control. This is equipped with QUARC software real-time control and multi-agent mission development frameworks; a ground control station; and embedded computer system and inertial measurement unit (IMU). Porting the cooperative control algorithms can be done through the software development toolkit. The

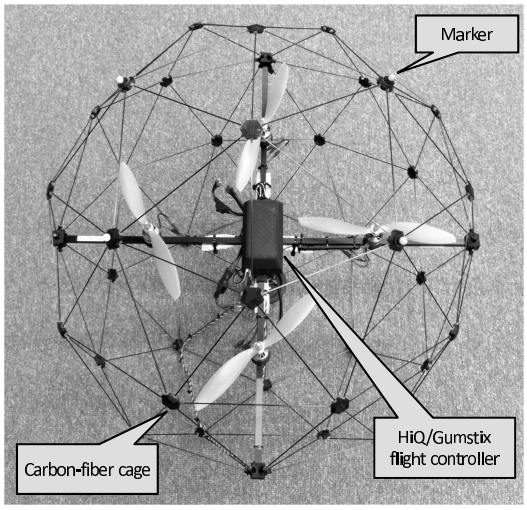


Figure 5: Quanser Qball-X4

MAVSeN laboratory is equipped with one DraganFlyer X6

(a six-rotor UAV helicoptor platform) and three Qball-X4 quadrotors that are controlled by a full-featured embedded avionics data acquisition card, the Quanser HiQ, which provides high-resolution inertial measurement sensors and motor outputs. High-level control of the Qball is performed by the Gumstix Verdex embedded computer platform which has been configured as a QUARC target system, this allows researchers to seamlessly create QUARC controllers from a host PC and then download and execute them on the embedded QUARC target computer. This will be further explained in Section 2.4.

The quad-rotor helicopter design has four brushless motors fitted with 10 inch propellers, speed controllers, and is enclosed inside a carbon fiber protective cage. This novel design provides safe indoor operation and protection from walls and other MAV platforms. It is equipped with the Quanser HiQ avionics data acquisition board and the Gumstix Vertex embedded computer. The Gumstix allows QUARC models to be downloaded and run directly on the Qball-X4. Wireless communication to the ground station and other vehicles can be configured to be either IEEE 802.11 (Wi-Fi) or IEEE 802.15.4 (ZigBee). The open-architecture of the Qball-X4 provides a platform in which researchers can rapidly modify the low-level flight dynamics stabilization parameters, as well as the advanced multi-agent guidance, navigation and control algorithms.

The Qball-X4 UAVs are integrated with the IRIS nodes, for detection and communication. By integration, the IRIS nodes are physically mounted onto the Qball and powered by tapping into the HiQ Avionics Data Acquisition board's power supply. This increases the power consumption, but keeps payload lighter by removing the node's battery pack, thus an even trade-off. Placement of the node is near the center of mass of the Qball; this is positioned on the top of the enclosure that protects the HiQ and Gumstix. Placement of the antenna of the IRIS node is critical in that signal propagation can be affected by the cage or other physical components. The antenna is vertically polarized and will be located on the Qball in a position such that reflections and near field effects due to surrounding conductors will have minimal effects on received signals from other nodes; also, factors like placement of the node beyond center of mass can unstabilize flight mechanics and decrease flight time. Though the Qball is equipped with an auto-pilot allowing it to be autonomous, is can also be teleoperated.

2.3 MEMSIC IRIS nodes

The IRIS 2.4 GHz Mote by MEMSIC is a module used for enabling low-power, wireless sensor networks[6]. It has a 2.4 to 2.48 GHz globally compatible ISM band; a 250 kbps High Data Rate Radio (outdoor line-of-sight tests yielded ranges as far as 500 meters between nodes without amplification); an IEEE 802.15.4 compliant RF transceiver; and a direct sequence spread spectrum radio (resistant to RF interference/provides inherent data security.) These nodes are the basis for the detection of the EM transmitter. We use four nodes in the development stage to test the functionality of PADF. Their primary function is not only to detect the EM source, but also transmit the RSSI values between neighboring nodes and the transmitter. These RSSI values will be instrumental in determining the position of the transmitter

Figure 6: MEMSIC's IRIS wireless sensor node

using the LSE. As seen in Figure 6, the antenna was customized by adding a ground plane to restrict interference of the EM signal by the node's internal circuitry. This modification allowed for relatively omni-directional RSSI measurements.

2.4 Integration of Qball and Vicon systems

An ad-hoc network configuration is used to transfer position and control information to and from the Qball. This network consists of the Qball which is the client along with the Vicon server and Quanser ground station acting as the data and control servers, respectively the Qball is configured to be on the same subnet as the Vicon server and Quanser ground station.

The Quanser ground station receives HiQ flight data and flight model information from the on-board sensors and monitors the response of the yaw, pitch, and roll controller outputs. It is also responsible for sending control commands to initialize the Qball into takeoff and landing states. Control commands are taken from the USB joystick connected to the Quanser ground station. To initiate the execution of the running flight model on the Qball the throttle on the joystick is moved from the zero-throttle position to the-quarter throttle position. If at any time during the flight test the Qball becomes unstable the user can lower the throttle to a position less than quarter throttle and this will send a land command to the Qball. Before the takeoff command is issued from the joystick the Qball must be receiving position data from the Vicon server and have an active connection with the joystick on the ground station. If either one of these states is not valid then Qball model will not execute and an error message is sent to the ground station.

The transfer of marker position data from the Vicon server is accomplished through a C++ program that uses functions from the Vicon SDK and Winsock protocols. When the program executes it calls functions from the Vicon SDK to create an internal client connection to the Tracker software, which is running a data push stream server. Once the created internal client is connected, it calls the data push stream server for a data frame. The data push server sends the data frames to the internal client, and the Winsock pro-

tocol is used to establish a connection to the Qball. Once the Qball connection is established the marker data frames that are being sent to the internal client are packaged into a sixteen element array and placed into the *sendbuffer* associated with the QballŠs socket connection. While the connection to the Qball is established this packaged data will be continually sent to the Qball. This provides real-time marker position information to the model running on the Qball and is used by the model for computing the Qball's centroid, pitch and roll rates. A visual representation of this system flow is modeled in Figure 7.

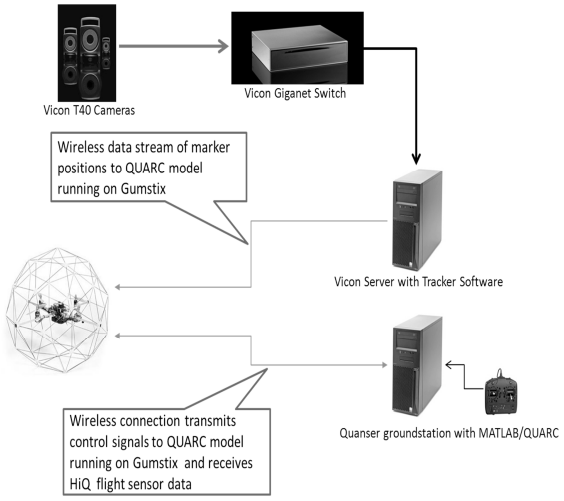


Figure 7: Qball-Vicon-Quanser Integration

3. CURRENT RESEARCH

3.1 Position Adaptive Direction Finding

The MAVSeN lab has many capabilities and functionalities. One novel concept in which these capabilities are maximized is a technique, denoted as Position-Adaptive RF Direction Finding (PADF), in which a non-cooperative RF emitter is localized using mobile sensor networks. These mobile networks will be configured as a swarm of MAVs incorporated with wireless sensor nodes. These MAVs will cooperate their sensing missions, adapt their position in real-time autonomously, and localize an unknown, hidden Electromagnetic (EM) source based on optimal detection algorithms. Our research activities are focused on cooperative control of these multiple MAVs while using custom position-adaptive algorithms for the detection of the source's unknown electromagnetic emission.

Typical direction finding is defined as a technique in which an emitter is localized in an open environment, usually using a well-defined method such as Angle of Arrival, Time-Difference of Arrival, or a hybrid of multiple techniques. PADF modified these concepts to encompass localizing an emitter in an urban or embedded environment. Given multipath and obstacles, the objective is to localize a hidden, uncooperative EM signal, thus, given n mobile sensor nodes or MAVs, we develop cooperative control algorithms that will maximize the probability of detection and localization of the EM source. In order to localize the transmitter, Received Signal Strength Indicator (RSSI) is used as an approximation of distance from the transmitter to the revolving receivers, provided from an algorithm for on-line estimation of the Path Loss Exponent (PLE) estimation that is used in modeling the distance based on received signal strength (RSS) measurements. The emitter position estimation is calculated based on the surrounding sensors' RSS values using a Least-Square Estimation (LSE). In doing such, three goals are accomplished along the way: maximized the probability of detection and localization given n mobile sensor nodes; used RSSI as a viable approximation of distance from receiver to emitter based on a proper PLE; and calculated an accurate position estimation of an EM signal based on RSSI values using a LSE algorithm. The basis of this work was developed from [7]. In order to prove this concept, a testing matrix was developed that would incorporate all the variables of the PADF model. This included the accuracy of the localization algorithm, along with the stability and sensitivity of given configurations; testing the propagation of the EM signal in embedded environment with leakage points, and defining a metric that determines the probability of a given configuration to estimate the hidden emitter's position.

3.2 Multiple MAV Cooperative Flight

For multiple Qball flights, a C++ multi-client server was created to stream the position data of all the Qballs simultaneously to all the Qballs in the capture volume. This multi-client server is run on the Vicon server and listens for connections from all Qballs that are operational. For multi-Qballs flights the structure and order of the elements that make up the position data array are very critical. This is due to the order by which each Qball and its associated markers are configured in the Tracker software. For each additional Qball in the capture volume an additional sixteen elements are added to the send data array.

4. CONCLUSIONS

We conclude that our MAVSeN lab is an innovative research facility that will allow advancements in the areas of Micro-Aerial Vehicles and Wireless Sensor Networks. By integrating both, we have opened a new avenue to other research endeavors that could prove to be very beneficial. We hope to improve our techniques such that it is possible to track small objects or equip significantly smaller micro aerial vehicles (akin to the interests of the USAF—bird-sized MAVs with flopping wings) or smaller autonomous airborne vehicles with WSNs. We are currently working on the development of a centralized control system that can be interfaced with small scale helicopters, quad-rotors, and articulated wing experimental platforms. Between the current decentralized control setup and the proposed future centralized control system the MAVSeN laboratory provides a perfect experimental environment for not only researchers at Louisiana Tech University but also for industry and defense researchers in the fields of wireless sensor networks and micro-aerial vehicles.

5. REFERENCES

[1] C. G. Cassandras and W. Li. Sensor networks and cooperative control. *European Journal of Control*, 11:436–463, June 2005.

[2] P. R. Chandler, M. Patcher, and S. Rasmussen. UAV cooperative control. In *Proceedings of the American Control Conference*, Arlington, VA, June 2001.

[3] P. R. Chandler, M. Patcher, and S. Rasmussen. UAV cooperative control. In *Proceedings of the American Control Conference*, Arlington, VA, June 2001.

[4] J. Cortés, S. Martínez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.

[5] W. Li and C. G. Cassandras. Distributed cooperative coverage control of sensor networks. In *Proceedings of 44th IEEE Conference on Decision and Control*, Seville, Spain, December 2005.

[6] MEMSIC. Wireless sensor networks. `http://www.memsic.com/products/wireless-sensor-networks.html`.

[7] A. K. Mitra. Position-adaptive UAV radar for urban environments. In *Proceedings of the IEEE International Radar Conference*, Adelaide, Australia, 2003.

[8] Quanser. Unmanned systems. `http://www.quanser.com/english/html/UVS_Lab/fs_overview.htm`.

[9] A. Richards and J. How. Decentralized model predictive control of cooperating UAVs. In *Proceedings of 43rd IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, December 2004.

[10] L. Singh and J. Fuller. Trajectory generation for a UAV in urban terrain, using nonlinear mpc. In *Proceedings of the American Control Conference*, Adelaide, Australia, 2003.

[11] Vicon. Vicon t40 cameras. `http://vicon.com/products/t40.html`.

Application of Context to Fast Contextually Based Spatial Authentication Utilizing the Spicule and Spatial Autocorrelation

Gregory Vert
Center for Secure Cyberspace
Louisiana State University, and
Texas A&M Central, Texas
(206) 409-1434
gvert12@csc.lsu.edu

Jean Gourd
Center for Secure Cyberspace
Computer Science
Louisiana Tech University
(318) 257-4301
jgourd@latech.edu

S.S. Iyengar
Center for Secure Cyberspace
Computer Science
Louisiana State University
(225) 578-1252
iyengar@csc.lsu.edu

ABSTRACT

This paper proposes an integrating mathematical method that unifies a new Contextual Processing model with that of the Spicule visual authentication method. In previous work the Spicule has been initially determined to be much faster at generation of authentication signatures for spatial data than standard encryption methods. It however could be much faster than it already is if a method was designed that could reduce the number of spatial objects it has to generate authentication signatures for. Previous experiments required Spicule to authenticate all spatial objects in a set for comparison against encryption methods. This paper provides brief overviews and background on the Spicule, and the new Contextual Processing model. It then proceeds to present the integrating mathematical approach of localized spatial autocorrelation. Finally an algorithm and the overall application of the method is presented by which limited sets of spatial object are mathematically selected for authentication when they are germane to a spatial query.

Keywords
spatial data authentication, contextual processing, contextual processing security, spatial autocorrelation.

1. INTRODUCTION

Contextual Processing (CP) has been around the research fields of Computer Science off and on for years. It has always had limited and focused application. However as the world has faced such events as 9/11, Indian ocean Tsunami, and Three Mile Island nuclear disaster there has been a need for more advanced processing paradigms especially ones that consider spatiality and temporality.

The goal of research in this area has been to link the environment a machine exists in to how the machine may process information. An example typically given is that a cell phone will sense that its owner is in a meeting and send incoming calls to voicemail as a result. Application of this idea has been applied to robotics and to business process management [1].

Some preliminary work has been done in the mid 90's. Schilit was one of the first researchers to coin the term context-awareness [2,3]. Dey extended the notion of a context with that of the idea that information could be used to characterize a situation and thus could be responded to [4]. In the recent past more powerful models of contextual processing have been developed in which users are more involved [5]. Most current and previous research has still largely been focused on development of models for sensing devices [6] and not contexts for information processing.

In addition to CP, there has also been an explosion in the amount of spatial data being generated and a heavy reliance on such data. Considering the use of GPS and Google Earth, this type of data is not the alpha numeric types of information that has been traditionally managed. One characteristic of the data is that it is voluminous and has spatial relationships inherently that much be preserved in its management. Coincidental with this fact, has been an increasing need to secure such data as it is transmitted across the internet. Traditional authentication methods have relied on dated concepts of hashing and encryption which are computationally impractical on large volumes of data. Instead of building faster processors the performance bottleneck of authentication can be addressed by *working smarter, only do what is required and ignore the rest of the noise*.

The following sections provide overviews on two brand new paradigms, that of the new CP model and the other of a visual algebra, the Spicule, that can be used for

authentication. Integration of the paradigms provides a potential path towards working more intelligently and quickly on authenticating and securing spatial information.

2. CONTEXTUAL PROCESSING

2.1 Overview

The initial development of the new CP model was based on examination of the natural disasters of the Indian Ocean tsunami, three mile island nuclear plant and 9/11. A goal was to determine what elements could be used to categorize these events. After analysis it was realized that all of them had the following categorical properties, which are referred to as the *dimensions* of a context in the model. They are:

time – the span of time and characterization of time for an event

space – the spatial dimension

impact – the relative degree of the effect of the event on surrounding events

similarity – the amount by which events could be classified as being related or not related.

Each one of the dimensions can be attributed which can be used to derive the semantic processing rules. These dimensions were discovered to be critical in the derivation of knowledge about an event because they affected the process of reasoning about an event. For instance, the time space dimensions can be utilized to reason that a tsunami in the middle of a large ocean may not have the *impact* or *similarity* to that of one just off the coast of Thailand and therefore the processing and dissemination of that information will be different. The reasoning is based in this case on the context defined by the dimensions.

The time and space dimension context driven processing will have the factors of geospatial and temporal elements to them. The geospatial domain can mean that information is collected and stored at a distance from where it may be processed and used in decision support as well as a description of the region that a context may pertain to. This means that context based information processing (CBIP) processing must have a comprehensive model to route information based on semantic content to the appropriate processing location and dissemination channels. CBIP processing can and often does have a temporal component. It can be collected over periods at regular or irregular intervals (the attribution of the dimension) and the time that the information is collected also may determine where the information is sent and the context of how the information is processed. For instance information that is collected as simply monitoring information may in the case of the Tsunami flow to research institutions around the world for storage and analysis at some point in the future. Whereas, noticing earthquakes on the ocean floor may route collected information to countries surrounding an ocean for immediate high speed analysis, critical real time decision making and rapid dissemination. Some factors that should be considered in CBIP processing are referred to as information criticality factors (ICF). These factors are further developed in ongoing research but are primarily used to drive processing decision making. They may include such attribution among other attributes as:

- time period of information collection
- criticality of importance,
- impact e.g. financial data and cost to humans
- ancillary damage
- spatial extent
- spatial proximity to population centers

These factors and many others in the model could be used to evaluate threat, damage, and criticality of operational analysis. Other factors affecting CBI processing might be based on the *quality of the data* such as:

- currency, how recently was the data collected, is the data stale and smells bad
- ambiguity, when things are not clear cut – e.g. does a degree rise in water temperature really mean global warming
- contradiction, what does it really mean when conflicting information comes in different sources
- truth, how do we know this is really the truth and not an aberration
- confidence that we have the truth

From the initial analysis of the facts describing the Indian Ocean Tsunami factors were defined that could define events and the context surrounding the event. These dimensions where defined to be the following:

temporality – defined to be the time period that the event unfolded over from initiation to conclusion

damage – the relative damage of the event both in terms of casualties, and monetary loss

spatial impact – defined to be the spatial extent, regionally that the event occurs over.

policy impact – directly driving the development of IA (security) policy both within a country and among

countries. This directly led to the evolution of security policy driving implementation because of the event.

2.2 Defining a Context

After the above dimensions were defined, the next phase of the research was to determine more rigorously how these factors might be defined and manipulated in an abstract sense. The following model component was developed where feature vectors could be utilized to define context and the factors of context. In its simplest form, a context is composed of a feature vector

$$F_n<a_1,..a_n>$$

where the attributes of the vector can be of any data type describing the event. This means that the vector can be composed of images, audio, alpha-numeric etc. Feature vectors can be aggregated via similarity analysis methods into super contexts S_c. The methods that might be applied for similarity reasoning can be statistical, probabilistic (e.g. Baysian), possibilistic (e.g fuzzy sets) or machine learning and data mining based (e.g. decision trees). Aggregation into super sets is done to *mitigate collection of missing or imperfect information and to minimize computational overhead when processing contexts.*

definition: A context is a collection of attributes aggregated into a feature vector describing a natural or abstract event.

A super context can be described as a triple denoted by:

$$S_n = (C_n, R_n, S_n)$$

where C is the context data of multiple feature vectors, R is the meta-data processing rules derived from the event and contexts data and S is controls security processing. S is defined to be a feature vector in this model that holds information about security levels elements or including overall security level requirements.

definition: A super context is a collection of contextual data with a feature vector describing the processing of the super context and a security vector that contains security level and other types of security information.

The cardinality of F with C is:

m:1

which when substituted into S creates a (C, R, S) cardinality of:

m:1:1

for the proposed model. However, we have not examined the impact, constraints of implications of having an

m:n:o

type of cardinality.

All of the above are a *type* of feature vectors where the elements of the vector can contain any type of information including the derived contextual processing rules and security methods for the given super context.

A super context is composed of context data from many sensing event objects, Eo_i,. As such contextual information collection works in a similar fashion to sensor networks and can borrow from theory in the field. Figure 1 shows the nature of collection of event object data over time. One can visualize a region of interest, e.g. the Indian Ocean tsunami for which event object data is collected which is centered over a thematic event object. In this case a thematic object when one considers all the data that may exist for the Indian ocean is the concept of the *origin of the tsunami*.

definition: A thematic event object (Teo) is the topic of interest for which event objects are collecting data. An example of a Teo would be the center of a tsunami.

In previous work [9], objects motions where characterized and described based on temporality, spatiality, impact and similarity. Development of these classes then lead to a grammar which derived rules that could have processing actions assigned to them. This allowed the notion of context to produce the paradigm of contextual processing. Simply put, *the nature of the information controlled the operation of the processing.*

3. SPICULE AUTENTICATION

The Spicule visual state change detection method[8] was originally conceived to be a simple and intuitive way to detect intrusions on computer systems. Years after its conception it was discovered that it had a variety of interesting applications based on the mathematics behind the paradigm. One of these turned out to be the ability to generate spatial authentication signature faster than standard hashing and encryption methods.

The Spicule's mathematics is based in vector algebra, and thus there is an algebra that exists for comparing two Spicule's to detect visually state changes in system state variables. Specifically, if the mathematical representation

of two Spicule's is subtracted a "change form" is created. The change form can be visualized which then results in a smooth featureless 3D ball if the two versions of the Spicule authentication signature are similar. The advantage of this is that it is simple and visually intuitive to recognize change with out having to conduct analysis or inspection of the underlying mathematica1 data. Figure 1 shows an example of the Spicule.

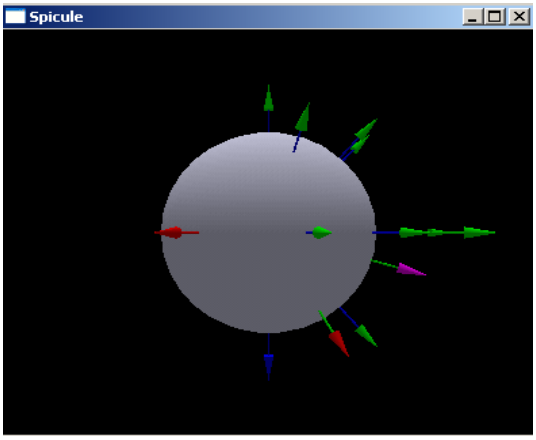


Figure 1. Sample picture of the Spicule

The development of the Spicule for authentication started with some research by Takeyama and Couclelis. They demonstrated that the GIS layering abstraction of a location is equivalent to a set of multiple attributes [9]. This means that various attributes about the same spatial object could be modeled that that selection of similar classes of attributes for a range of objects to be modeled for a variety of applications including authentication. This layering can be conceived as as a 3D-set of layers on top of each other.

In the layering paradigm, the Spicule can be utilized to create a mathematical *signature* for authenticating spatial data by mapping the tips of vectors on the Spicule to the unique spatial objects identified from the taxonomy. The signatures that can be generated using this approach becomes an n tuple which can be visually subtracted using Spicule to detect changes in the spatial data. This n tuple consists of information about a specific spatial objects vector consisting of a unique set of attributes such as magnitude, angular orientation and location of a vector on the 3D central ball. The vectors can be mapped to objects in various layers of spatial data objects (mentioned above), thus creating vectors that are not tied to the objects in a given layer, increasing the uniqueness of the signature. The number of vectors going from the Spicule was equal to the number of selected objects from the spatial dataset being authenticated. The collection of these vectors for a given set can be then used to describe a unique signature for a particular GIS data set.

The idea behind the developed authentication process was to utilize the Spicule tool to create a geometrical vector for each of several spatial objects selected from the spatial data sets. Vectors can be point from the center of the Spicule to the (x, y, z) coordinates of a spatial object. Each vector is thus unique and has three attributes that are represented as follows:

$$V_i = (\text{degreesVertical}, \text{degreesEquator}, \text{magnitude})$$

In this scheme there is a vector pointing from the center of the Spicule, at the origin, to each point or spatial object selected from the spatial data set for signature generation. In the previous work three data layers are initially proposed to be placed at one vertical unit apart from the Spicule layer. So, the first layer points will have coordinates of (x, y, 1), the second layer points' coordinates will be (x, y, 2), and the third layer points' coordinates will be (x, y, 3). Based on this the vector attributes for each authentication point in the three layers were:

$$Mag_i = \sqrt{x^2 + y^2 + i^2}$$

where:

i is the data layer number.

x, y are point original coordinates.

Mag_i is the magnitude of the vector from (0,0,0) to a point in layer i.

$$Sin\theta_{ei} = \frac{x}{\sqrt{x^2 + y^2}} \Rightarrow \theta_{ei} = Sin^{-1}\frac{x}{\sqrt{x^2 + y^2}} \qquad (2)$$

$$Sin\theta_{vi} = \frac{i}{\sqrt{i^2 + y^2}} \Rightarrow \theta_{vi} = Sin^{-1}\frac{i}{\sqrt{i^2 + y^2}} \qquad (3)$$

Equations (2) and (3) are used to calculate the equator and the vertical angles respectively,

where:

i is the data layer number.

θ_{vi} is the vertical angle degrees for a vector from (0,0,0) to a point in layer i.

θ_{ei} is the equator angle degrees for a vector from (0,0,0) to a point in layer i.

The collection of attributes and angles for all authentication vectors forms a two-dimensional matrix that is used as for the authentication signature and the Spicule visualization authentication process.

The signature calculation process is done when a spatial dataset is requested to be transmitted over the internet. Table 3 shows a sample calculated vector matrix.

Object ID	Layer	Mag_i	θ_{vi}	θ_{ei}
1	3	7.68	66.8	18.43
2	2	16.31	42.51	4.76
		. .	. .	. .
n	i	29.22	51.95	3.18

Table 3. Sample calculated vector matrix

At the receiving end, the same process to create a signature matrix from the *received* spatial dataset was applied. By visualizing the mathematical difference between the received spatial data sets matrix and the transmitted matrix, it can be determined if the dataset has been intercepted or altered during transmission. This process may be described by:

IF Visual Mathematical Difference = 0 THEN

No Interception or Alternation.

This is the standard logic found in traditional authentication schemes. In the above method if the visual mathematical *difference* (vector based subtraction) between the two matrices does not equal to zero, it is assumed that the spatial dataset has been intercepted and altered. However, we can not determine the extent and the type of change that have been made because removal, addition, or movement of a given spatial object or point may result in the change of sequence for many vectors in the matrix after the point of modification in the matrix. *The nice thing about application of Spicule is that visualization of the signature matrices with Spicule and application of visual subtraction of the vectors results in a Spicule devoid of vectors if the data objects have not been moved or modified during transmission. The intuitive nature of this visualization makes it easy for an analyst with the most basic of skills to determine if data has been modified and how much.*

4.0 COMPARATIVE AUTHENTICATION SIGNATURE GENERATION PERFORMANCE

Spatial data may be protected for transmission by encryption or by the generation of a signature using MD5, SHA or RIPEMD. In order to compare the performance of the spatial signature approach to that of above traditional methods a test suite was set up on a PC running at 2.4ghz with a P4 processor. The Crypto++ package was utilized for comparison with timing figures measured down to the millisecond. Crypto++ has a program call Cryptest that may be called with command line switch to encrypt symmetrically, decrypt and generate SHA, MD5 and RIPEMD160 digests. The command line interface was invoked from a command line shell generated with Visual Studio. Because Cryptest was being called using a system command from inside the compiled test program, the first part of the test suite called the operating system shell to load a simple C program. This allowed us to measure the effect on performance of just loading a simple program. Of note in the spatial signature generation test, this test selects increasingly more and more static spatial objects from the test data which are part of the objects from the previous work with taxonomies mentioned above. The above test was run thirty times for each part of the above test program with the following results:

Test Type	Pass 1 (10x)	Pass 2 (10x)	Pass 3 (10x)
Shell	63.00	58.00	57.00
Encrypt (symmetric)	126.60	123.4	121.90
Decrypt (symmetric)	115.60	123.5	121.90
MD5/SHA/RIPEMD	67.20	67.20	64.00
Spatial Authentication	< .01 millisecond	< .01 millisecond	< .01 millisecond

Table 4 Average performance comparison

of Spatial Authentication versus Symmetric encryption, SHA, MD5, RIPED (milli seconds) on test data

4. SPATIAL AUTOCORRELATION APPROACH

Spatial autocorrelation was developed by Moran in 1995 and has the potential to integrate contextual modeling in such that a reduced number of spatial objects can be selected for the Spicule authentication. This section discusses how such a method may work and is the subject of future research.

Global spatial autocorrelation measures the degree to which objects on a spatial grid are related to other objects. The notion is based on spatial dependence which can be defined as "the propensity of a variable to exhibit similar values as a function of the distance between the spatial locations at which it is measured"[7]. Put more simply, the value of a spatial variable is often influenced by its neighbors.

Global spatial autocorrelation can be defined given variable $x = \{x1, \ldots, x_n\}$ sampled over n locations[7]. Morans spatial correlation coefficient can be calculated by:

$$I=\frac{zWz^t}{zz^t}$$

where:

$z = \{x_1 - \bar{x},\ldots x_n-\bar{x}\}$

z^t – is the transpose of z

W - is a rectangular row normalized contiguity matrix

Localized spatial auto-correlation (LSA) is similar to global autocorrelation. Instead of measuring the correlation of a group of objects at a global level, it is a measure that determines how correlated a given variables location might be correlated and influenced by its neighbors. This is a derivation of I and is given by:

$$I_i=\frac{z_i}{s^2}\sum_j \frac{W_{ij}}{z_j},\ i \neq j\ ,$$

where:

$z_i = x_i - \bar{x}$

s – is the standard deviation of x

W_{ij} - is the contiguity matrix, normalized, or based on similarity

The application of local autocorrelation and context might follow the logic that

i) a user wants to retrieve object for a given location in space and or in a given time period for that location.

ii) the object the user might want to look at are of a given class with heterogeneous members. For example:

O = {tank, half trac, jeep, jeep with gun mount, armored personal carrier}

where:

O – is object class of battlefield objects with wheels

Note that within this class there are implications for *similarity* from the context model such as *members that can fire projectiles* and *members that transport resources*.

These members will have spatial locations, temporal loci and impact relationships for a given location, T_{eo} and for other given themes that might be part of a retrieval query such as *fighting, moving, transporting*.

To demonstrate how LSA might be integrated with context, consider the follow example. Figure 2 is spatial lattice where members of **O** are located in various concentrations and dispersions.

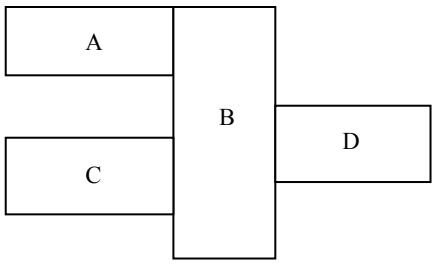


Figure 2, A contiguity lattice C of associated cells over a spatial extent with members of set O dispersed in various cells of the lattice.

Considering the argument of spatial dependency, one can see that concentrations of vehicles with guns can tend to be related among adjacent cells in the lattice, and that the same could apply for concentrations of vehicles that are used for transport.

The LSA method takes the above lattice and constructs a contiguity matrix as shown in figure 3. This matrix is the beginning of the LSA and identifies which lattice cells have shared edges and thus my have correlations among the cell contents.

	A	B	C	D
A	0	1	0	0
B	1	0	1	1
C	0	1	0	0
D	0	1	0	0

Figure 3, Contiguity Matrix M_c

Use of variations of the contiguity matrix in the Spicule approach is going to be the subject of further research and development, however, the current LSA uses a normalized matrix such as shown in Figure 4. Normalization is down to minimize the undue influence on calculation of I_i due to a large number of contiguous cells around a cell of interest.

	A	B	C	D
A	0	1	0	0
B	.3	0	.3	.3
C	0	1	0	0
D	0	1	0	0

Figure 4, Row Normalized *W*

Application of the LSA I_i can now provide the basis for application to the Spicule authentication method.

Consider that a user is interested in query Q_1:

Q_1 = *(the location of the majority vehicles with guns on them*, T_{eo})

Q_1 is a very realistic type of query for planning attacks or logistics. The steps to apply LSA in this type of query would be:

i) build *C*

ii) build $M_c = fn(C)$

iii) calculate $W = fn(M_C)$

iv) calculate $I_i = fn(W)$

v) apply Q_1 for some sort of selection criteria producing **O**

vi) generate authentication signature vector

$$s[] = Spicule(O)$$

Application of the above if done properly could produce a reduced number of spatial objects to authenticate and thus improve the already fast processing of Spicule.

Step v in the above algorithm implies some sort of selection method on the correlation coefficients I_i . This can be done by application of one of the following criteria:

- similar values,
- above a *floor* value,
- below a *ceiling* value
- falling into a bounded range

As an example, consider calculated lattice consisting of localized correlation coefficients for Q_1 as shown in Figure 5, a selection criteria for correlation of .8 ± .2, and a region of interest T_{eo}. Calculation of a localized correlation values might result in the following type of lattice where

	A	B	C	D
A	0	.82	0	0
B	.79	.8 T_{eo}	.5	1
C	-.2	.23	.4	0
D	0	1	-.6	0

Figure 5 Sample calculated local correlations I_i over *W*

the T_{eo} is a spatial location that Q_1 is centered upon. The application of the algorithm presented would result in the selection of lattice cells containing {.82, .79 and .8}.

A research note for the future would be to examine performance of this method and how it degrades as the granularity of the lattice increases. The current method proposes lattice cells that contain sets of objects, however it would be possible to say that each cell is a single object.

This LSA approach has some very handy properties when considering the CP model and integration into the Spicule method. The first of these is that it

i) integrates the dimension of spatiality into the Spicule process. It does this by organizing objects into lattices and incorporating the notion of spatial dependency. It also does not force unnecessary overhead on the calculation of I_i because it operates based on the notion of an irregular lattice, not a fixed lattice or a fixed grid. Cells without objects meeting Q_1 are merely ignored in the method.

ii) LSA has the built in notion of spatial dependency. This again is the idea that what is close to you spatially probably affects the value of adjacent cells. In the above example we have argued that it could be the case that vehicles with guns may be concentrated in certain lattice cells. This allows the proposed approach to integrate nicely with the CP dimension of *Similarity* . This idea might be defined as

Similarity based on spatial dependency

8. FUTURE WORK

The presented method integrates the new CP model with Spicule authentication via application of the LSA approach to create a new contextually based authentication paradigm. At present much work is being done on the theoretical constraints and applications of these methods. This leads to the opportunities for much more empirical work to be considered. Future research issues are many. One may investigate:

i) the integration of impact and time into the above proposed method and how that may be modeled.

ii) how granularity of lattice cells affect performance in selection of objects for Spicule authentication

iii) Boolean algebras for combinations of selection criteria in the proposed algorithm in this paper.

iv) $S_n = (C_n, R_n, S_n)$ and what the security term S_n may be defined as based on the method proposed in this paper.

CP has proven to be a useful paradigm in several areas of computer science and should continue to be investigated to develop a significant corpus of knowledge.

8. REFERENCES

1. Rosemann, M., & Recker, J. (2006). "Context-aware process design: Exploring the extrinsic drivers for process flexibility". T. Latour & M. Petit *18th international conference on advanced information systems* engineering. proceedings of workshops and doctoral consortium*: 149-158, Luxembourg: Namur University Press.*
2. Schilit, B.N. Adams, and R. Want. (1994). "Context-aware computing applications" (PDF). *IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'94), Santa Cruz, CA, US*: 89-101.
3. Schilit, B.N. and Theimer, M.M. (1994). "Disseminating Active Map Information to Mobile Hosts". *IEEE Network* **8** (5): 22–32. doi:10.1109/65.313011.
4. Dey,Anind K. (2001). "Understanding and Using Context".*Personal Ubiquitous Computing* **5** (1): 4–7. doi:10.1007/s007790170019.
5. Cristiana Bolchini and Carlo A. Curino and Elisa Quintarelli and Fabio A. Schreiber and Letizia Tanca (2007). "A data-oriented survey of context models" (PDF). *SIGMOD Rec.* (ACM) **36** (4): 19--26. doi:10.1145/1361348.1361353. ISSN 0163-5808. http://carlo.curino.us/documents/curino-context2007-survey.pdf.
6. Schmidt, A.; Aidoo, K.A.; Takaluoma, A.; Tuomela, U.; Van Laerhoven, K; Van de Velde W. (1999). "Advanced Interaction in Context" (PDF). *1th International Symposium on Handheld and Ubiquitous Computing (HUC99), Springer LNCS, Vol. 1707*: 89-101.
7. Shekhar, S; Chawla, S;(2003). *Spatial Databases A Tour*, Prentice Hall, p 190.
8. Vert, G.; Iyengar, S.S.; Phoha, V.;(2009) *Security Models for Contextual Based Global Processing an Architecture and Overview*, Cyber Security and Information Intelligence Research Workshop, published in ACM Digital Library, Oakridge National Laboratory, TN,.
9. Vert G; Phoha, V; Iyengar, S.S; (2010). *Contextual Processing Theory and Applications*, Taylor and Francis.

Network Maneuver Commander - Resilient Cyber Defense

Paul Beraud	Alen Cruz	Suzanne Hassell	Juan Sandoval	Jeffrey J. Wiley
Raytheon Company	Raytheon Company	Raytheon Company	Raytheon Company	Raytheon Company
Largo, FL 33777	Largo, FL 33777	Largo, FL 33777	Largo, FL 33777	Garland, TX 75042
1.727.302.3343	1.727.302.3296	1.72.302.4637	727.302.4637	972.205.8145
paul_f_beraud@ raytheon.com	alen_cruz@ raytheon.com	shassell@ raytheon.com	sandoval@ raytheon.com	jeffrey_j_wiley@ raytheon.com

ABSTRACT

Network Maneuver Commander (NMC) is a research project to develop a prototype cyber command and control (C2) system that maneuvers network-based elements preemptively, and to develop performance metrics to be used for the evaluation of cyber dynamic defense solutions. The Network Maneuver Commander addresses the gap area between active information operations & reactive information assurance defenses, by focusing on the introduction of artificial diversity of hardware platforms, operating systems, IP addresses and hypervisors. NMC also establishes metrics to determine the benefit of these defensive techniques. The goals of the research were to increase the investment an attacker must make to succeed, increase the exposure of an attacker to detection as the attacker is forced to out-maneuver target reconfigurations, increase the uncertainty of the success of the attack, increase the survivability in the presence of attacks, and to define metrics associated with cyber operations for a resilient and dynamic defense.

Categories and Subject Descriptors

C.4 [**Computer Systems Organization**]: Performance of Systems – *measurement techniques, modeling techniques, performance attributes.*

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection – *invasive software.*

General Terms

Algorithms, Management, Measurement, Reliability, Resiliency, Experimentation, Security.

Keywords

cyber security; dynamic defense; command and control; network maneuver; defense metrics.

1. INTRODUCTION

In order to decrease the success of cyber attackers, new and proactive defenses strategies are required. Conventionally, defense in the cyber domain has relied upon a static, layered, "defense in depth" approach, with a focus on perimeter protection. By establishing a new defensive technique, network maneuvering, the benefit of each individual layer provides can be relocated, helping defend against the most effective suite of malicious attacks.

In order to characterize the performance of cyber dynamic defense solutions, metrics need to be defined and captured. Conventionally, defense in the cyber domain has relied upon a layered, "defense in depth" approach. By establishing metrics for dynamic defense, the benefit each individual layer provides can be determined, helping select the most effective suite of defensive techniques.

1.1 Overview

Proactive computer network defense must anticipate the emergence of new vulnerabilities, take action to avoid threat actors seeking to exploit these vulnerabilities, and disrupt the actions of successful intruders to increase their work factor and minimize their impact.

The purpose of this paper is two-fold:

1) to describe the goals of the Network Maneuver Commander, the prototype developed, and the research conducted, to preemptively maneuver network elements to avoid cyber attack; and

2) to define metrics associated with cyber operations for dynamic defense. These metrics are captured from the perspective of the defender, as well as that of the attacker. For each metric, a description, method to calculate, analysis and a proposed collection point is provided.

Conference'10, Month 1–2, 2010, City, State, Country.

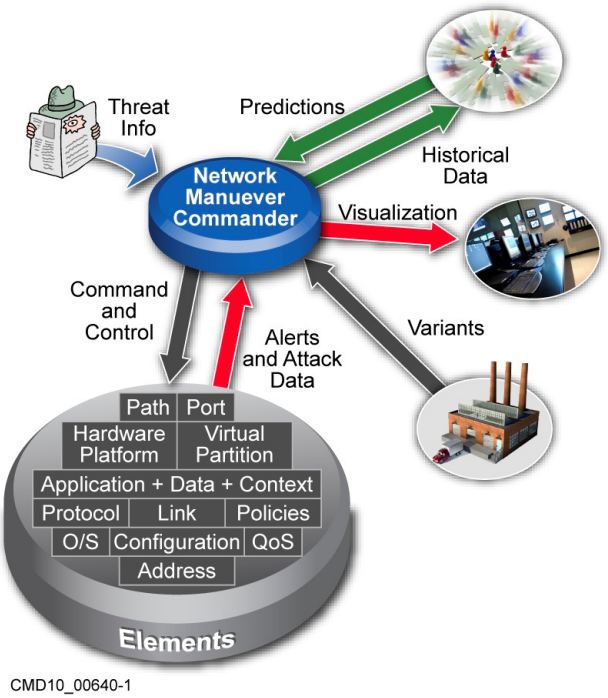


Figure 1 - NMC Context Diagram

1.2 Background

A leading example of prior research in the area of dynamic defense is the DARPA-funded project called Intrusion Tolerance by Unpredictable Adaptation (ITUA) [3]. ITUA successfully demonstrated the feasibility of thwarting attackers by injecting pseudo-randomness in system response to attacks, but as a post-attack reaction and response. We enhanced this concept by proactively maneuvering resources during normal system operations and prior to and independent of any attack. Furthermore, compared to ITUA, our scope is broader and encompasses the full gamut of hardware and software through the creation of artificial diversity. Resources here include hardware platforms, operating systems, middleware, as well as applications.

George Mason University pioneered a self-cleaning intrusion tolerance technology (SCIT) [1]. The SCIT Technology rotates transaction servers using load balancing and cleanses the ones that are offline prior to returning them. The Network Maneuver Commander concept is different in that although cleansing is also done, the primary focus is on the introduction of randomized artificial diversity, timing, and geographic destination. It is also not focused on transaction servers.

Information assurance (IA) defensive techniques today are passive and reactive. We wait and hope not to be attacked, focus on defending the perimeter, and as a result are vulnerable to insider and 0 day attacks. There are no industry defined standard metrics or benchmarks for evaluating cyber security architectures and systems performance for pro-active defense. The sections below encompass the work-to-date on the development of a cyber defense network maneuver commander. They are the result of Internal Research and Development (IRAD) funded by the Raytheon Company from March 2009 to December 2009. They describe and demonstrate the effectiveness of various cyber defense maneuvering techniques.

Some cyber defense metrics exist today. Metrics exist for evaluation of perimeter security for measuring IPS and IDS performance, for Red Team processes and for vulnerability assessments. ITUA [3] defined metrics as a means of measuring the post-attack reaction and response.

However, there are no industry defined standard metrics or benchmarks for evaluating pro-active dynamic defense methods in cyber security architectures today. The metrics described in this paper are a candidate set to quantify performance of these systems. They were derived from a cyber force-on-force simulation used to evaluate the effectiveness of various cyber defense techniques.

2. DISCUSSION

A network maneuver approach that is capable of avoiding many attacks, even in the face of zero-day vulnerabilities, provides a proactive posture for the enterprise. Recognizing that some attacks will succeed, the ability to disrupt a persistent threat, by requiring further attacker action to re-establish command and control, not only makes them work harder, but can increase the probability of attribution.

The Network Maneuver Commander functionality would contribute to future system capabilities by morphing the “game board” on potential adversaries and significantly raising their stakes in this cyber warfare. Network maneuvering raises the stakes for the adversary by providing effective, proactive defense with the objectives to: minimize the magnitude of the attacker’s effect; increase cost to the attacker; increase chance of detection and attribution; and increase the uncertainty that the attack was successful.

2.1 Hacking Process

The hacking process describes the steps a cyber attack must take in order to be successful. For the analysis and examples that follow, a hacking process as described in Hacking Exposed [4] is followed (see Section 9, References). This process can be thought of as a state diagram; a possible depiction is shown in Figure 2.

As a cyber attack executes, it progresses through the steps sequentially, from the Footprint phase to the Pilfer phase. If a phase is reached where a cyber defense prevents access, the attack reverts to a previous phase. (In some cases this may be back to the Footprint phase.)

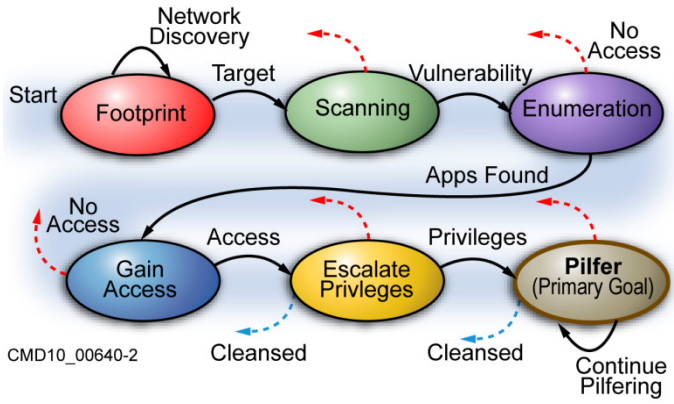


Figure 2 - A (one) hacking methodology

2.2 Cyber Defense Goals

There are three goals for the cyber defenses studied. These cyber defenses seek to:

1) Increase cost to the attacker

2) Increase uncertainty that the attack was successful

3) Increase chance of detection and attribution

The cost associated with execution of a cyber attack can be quantified as the number of times a particular phase of the attack is thwarted and the amount of time that is spent in the preparatory phases of an attack. For the cyber attacks analyzed, preparatory phases are those leading up to the Pilfer phase. A cyber defense is successful if there is an increase in both of these quantities.

The uncertainty associated with the success of an attack can be measured as a function the amount of time a threat spends executing its goal. For the cyber attacks analyzed, this is the time spent in the Pilfer phase. A cyber defense is successful if there is a decrease in this quantity.

The probability that a cyber attack is detected is proportional to the total time required for it to reach and execute its goal. A cyber defense is successful if there is an increase in this quantity.

Through the definition of these metrics, time becomes the fundamental measure of success and effectiveness.

2.3 Decision Framework

A decision framework provides the necessary intelligence and configuration information to enable the NMC to maneuver elements. The configuration information contains three main elements used to make maneuver decisions: diversity, geographic destination, and move interval. Additionally, intelligence information is provided in the context of threat levels that impact the decisions made relative to all three main elements. Finally, consideration is given to a security zone constraint where the NMC will not maneuver elements between security zones.

The three main elements (diversity, geographic destination, and move interval) consist of data elements allowing for a customized move based on intelligence and other constraints. For example, the diversity element considers the hardware platform type, the hypervisor being used / moved to, the operating system type, and the applications that need to be moved. Any constraints contained within these elements are also considered. Figure 3 depicts this graphically. There are different move intervals as shown as well as multiple geographic destinations that can all be customized based on the existing infrastructure and the threat/vulnerability.

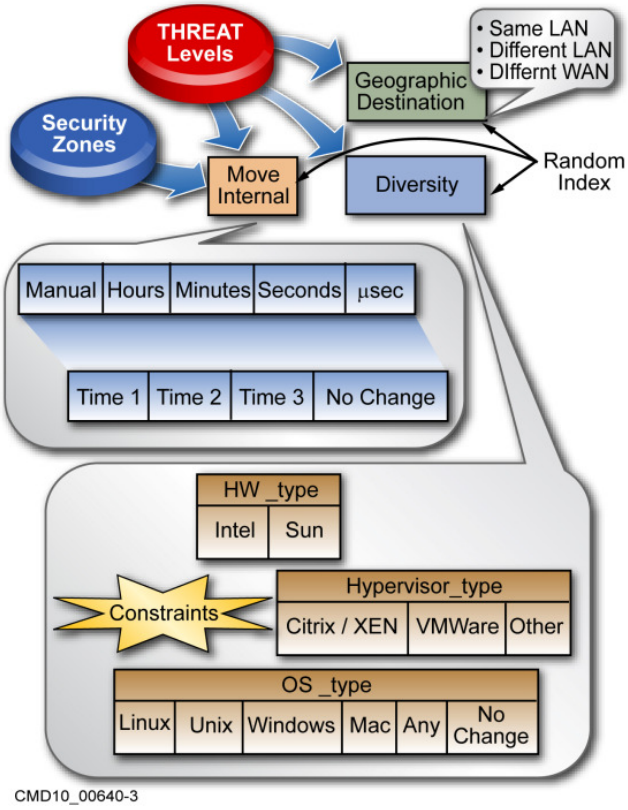


Figure 3 - Decision Framework

2.4 Analysis Framework

Capture and analysis of the metrics can be performed through force-on-force simulation. Each attack that executes against the modeled system is treated as independent. Data of each simulated event is collected. Ultimately, statistics for the attacks and defenses are aggregated, resulting in the identified metrics. The analysis framework used is shown in Figure 4.

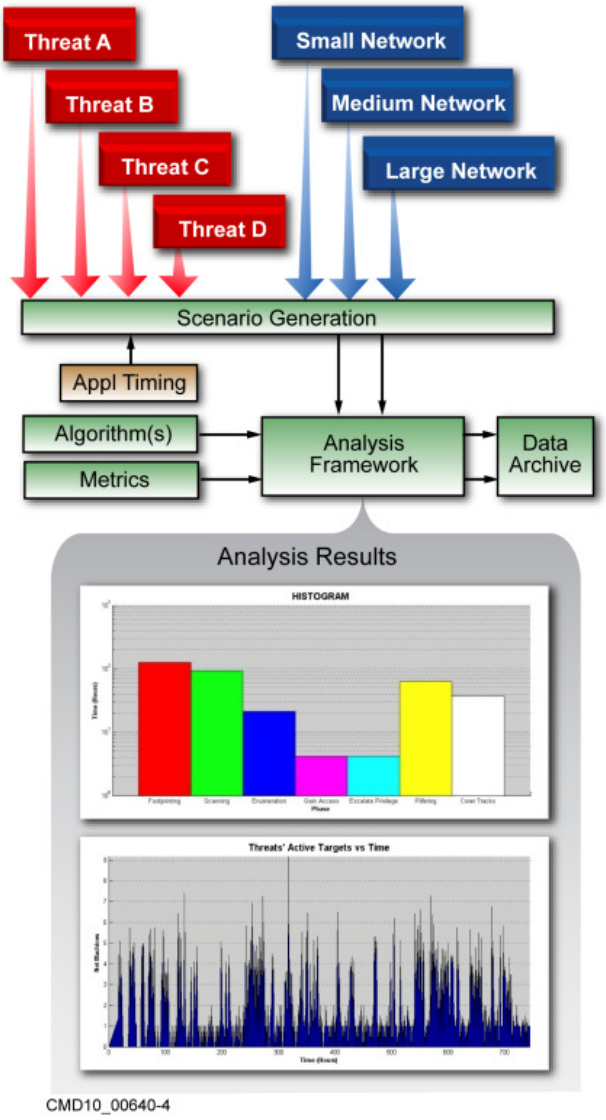


Figure 4 - Analysis Framework

2.5 Network Maneuver Architecture

The Network Maneuver Commander architecture consists of an extensible collection of loosely coupled services. The services were developed to be standalone independent components conforming to a variety of interfaces including WSDL, Rest & JMS XML message based. The orchestration of the services was accomplished via the use of an Enterprise Service Bus (ESB).

By leveraging the use of an orchestration engine, custom business logic for a particular deployment can be modified / extended via the rule configuration files. The NMC architecture includes a generic plug-in framework to provide wrappers for new applications to be plugged into the NMC system.

2.6 Network Collection Points

In deployed networks, capture of the metrics could be accomplished through extensions of the existing tools and equipment to support collection of additional data or mining of data that is currently collected. This data could then be aggregated and reported in support of the metrics identified in this paper. These metrics should be used across the industry to provide consistent measures and methods used by network cyber defense tools to provide the basis for defining and measuring cyber dynamic defense Service Level Agreements (SLAs). .

3. METRICS

The sections below also encompass the work-to-date on the development of cyber dynamic defense metrics. Throughout this discussion, the basis used for many of these metrics is time. Time is not only a measure of a cyber attack's progress, but is also used to quantify the cost to the attacker.

3.1 Percent of Successful Attacks

3.1.1 Description

A successful attack is defined as one which accomplishes its goal – in this case, successfully reaches and completes a Pilfer phase.

3.1.2 Calculation

$$p_{A,success} = \frac{N_{A,success}}{N_{A,total}} \times 100\%$$

where:

pA,success – percent of successful attacks

NA,success – number of successful attacks observed

NA,total – number of total attacks observed

3.1.3 Analysis

As the total number of attacks observed grows, this percentage approximates the probability that an attack will be successful. Thus, cyber defenses should serve to reduce this number.

3.2 Percent of Partially Successful Attacks

3.2.1 Description

An attack is partially successful when it to executes all of its phases up to, but not including, the Pilfer phase. This represents an attack's ability to defeat the boundary cyber defenses, and have access to and/or control of their target system.

3.2.2 Calculation

$$p_{A,partial} = \frac{N_{A,partial}}{N_{A,total}} \times 100\%$$

where:

$p_{A,partial}$ – percent of attacks ready to begin executing their goal

$N_{A,partial}$ – number of attacks reaching the start of their goal

$N_{A,total}$ – number of total attacks observed

3.2.3 Analysis

The percent of partially successful attacks is intended to characterize the presence of an attack on a system. Being that NA,partial is greater than or equal to NA,success, then pA,partial is greater than or equal to pA,success.

3.3 Mean Number of Attack Disruptions

3.3.1 Description

Disruptions are any effect a cyber defense produces that impedes the progress of an attack through its process.

3.3.2 Calculation

$$\overline{N}_{disruption} = \frac{\sum_{i=1}^{N_{A,total}} N_{i,disruption}}{N_{A,total}}$$

where:

$N_{disruption}$ – mean number of disruptions per attack

$N_{i,disruption}$ – number of disruptions on the i^{th} attack

$N_{A,total}$ – number of total attacks observed

3.3.3 Analysis

The number of attack disruptions is dependent on the length of time the system is observed. As such, it is recommended that the observation time be the same when cyber defenses are compared, or that this number be normalized by a unit of time. The number of disruptions is also correlated to the number of defensive actions. For dynamic defense where actions are preemptive, this is also correlated to the periodicity of defensive actions. In the case of reactive defense, defensive actions are correlated to the probability of attack prevention.

3.4 Time Spent per Phase

3.4.1 Description

An attack's timing profile can be characterized by the amount of time it spends in each of its phases. The goal of a cyber defense solution is to increase the time an attack spends in the preparatory phases, as well as shift the amount of time spent to the earlier phases (e.g., toward the perimeter, during the Foot-printing and Scanning phases).

3.4.2 Calculation

$$T_{phases} = (t_1, t_2, ..., t_N)$$

where:

T_{phases} – vector of phase-times

N – number of attack phases

t_n – time

$$t_n = \sum_{i=1}^{N_{A,total}} t_{i,n}$$, for cumulative time;

$$t_n = \frac{\sum_{i=1}^{N_{A,total}} t_{i,n}}{\sum_{j=1}^{N} t_j} \times 100\%$$, for percent mean time;

$$t_n = \frac{\sum_{i=1}^{N_{A,total}} t_{i,n}}{N_{A,total}}$$, for mean phase-time

where:

$t_{i,n}$ – time spent by the i^{th} attack on the n^{th} phase

t_n – time spent on the n^{th} phase of an attack

t_j – time spent on the j^{th} phase of an attack

$N_{A,total}$ – number of total attacks observed

N – number of attack phases

3.4.3 Analysis

The time spent per phase can be visualized as a cumulative phase time distribution. An example of this is shown in Figure 5.

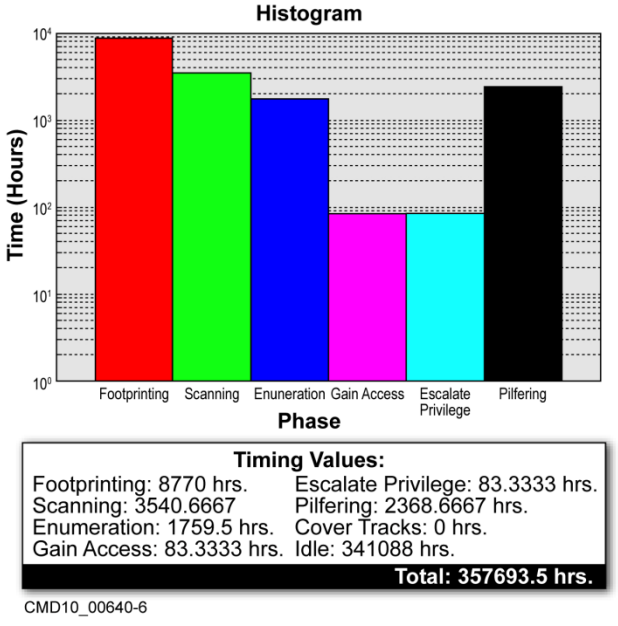


Figure 5 - Phase-Time Distribution

For preemptive defense, the distribution's histogram can be stacked. This allows multiple distributions to be compared on a single chart, as depicted in Figure 6.

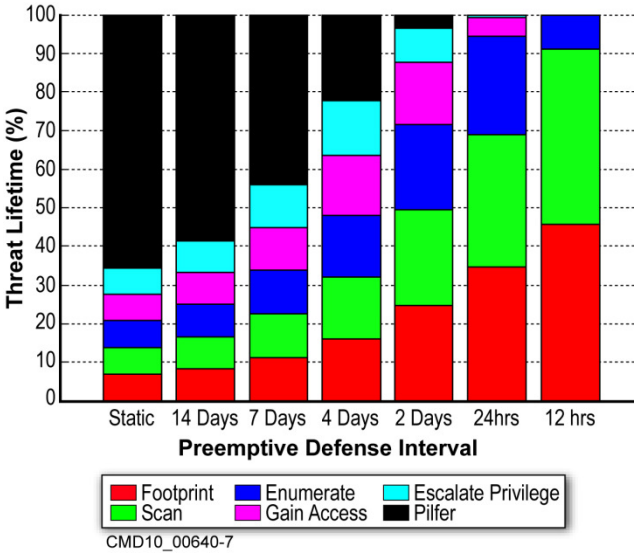


Figure 6 - Phase-Time Stacked

An alternative to the stacked histogram is the line graph, shown in Figure 7 below

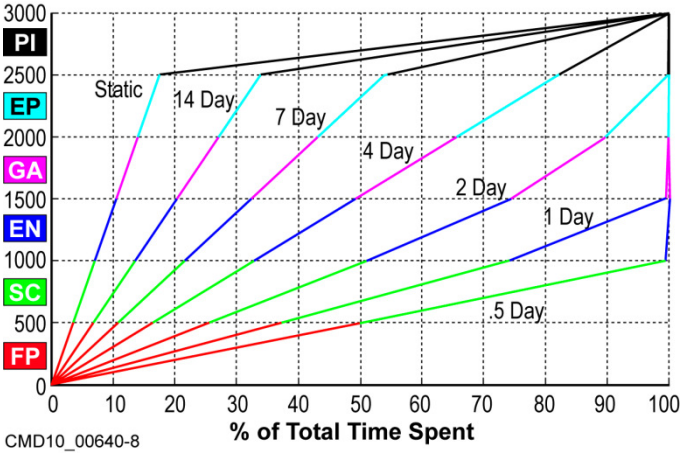


Figure 7 - Phase-Time Line Graph

3.5 Duration of Successful Attack

3.5.1 Description

Duration of a successful attack is the time consumed to execute from the first phase to the last (e.g., Foot-printing through Pilfer). This execution time may include multiple revisits to intermediate phases, either due to the way the attack behaves, or obstruction due to cyber defenses. Given multiple observations, the mean time can be computed.

3.5.2 Calculation

$$\bar{t}_{A,success} = \frac{1}{N_S} \sum_{j \in S} \sum_{i=1}^{N} t_{j,i}$$

where:

$t_{A,success}$ – mean execution time of an attack

$t_{j,i}$ – time spent by the j^{th} attack on the i^{th} phase

N – number of attack phases

S – set of all successful attacks (a total of $N_{A,success}$)

N_S – number of attacks that are members of S

3.5.3 Analysis

The effect of cyber defenses on a cyber attack can be observed through compression/expansion of this time versus an attack's nominal timeline.

3.6 Defensive Efficiency

3.6.1 Description

Defensive efficiency is the measure of how often an attack is disrupted versus how often defensive action is taken.

3.6.2 Calculation

$$\eta_{defense} = \frac{N_{A,total} - N_{A,success}}{N_D} \times 100\%$$

where:

$\eta_{defense}$ – defensive efficiency

$N_{A,total}$ – number of total attacks observed

$N_{A,success}$ – number of successful attacks observed

N_D – number of defensive actions taken

3.6.3 Analysis

The effect of cyber defenses on a cyber attack can be observed through compression/expansion of this time versus an attack's nominal timeline.

3.7 Defense Factor

3.7.1 Description

The defense factor captures the ratio of the preemptive defense interval to the nominal duration of a particular type of attack. It provides a measure of the relative speed of execution between defense and attack.

3.7.2 Calculation

$$D = \frac{t_{A,nominal}}{t_D}$$

where:

D – defense factor

$t_{A,nominal}$ – nominal attack duration

t_D – preemptive defense interval

3.7.3 Analysis

As preemptive defense actions speed up (i.e., their interval shortens), the probability that an attack will succeed diminishes. This characteristic is best seen in a sensitivity sweep of defense factors, as shown in Figure 8.

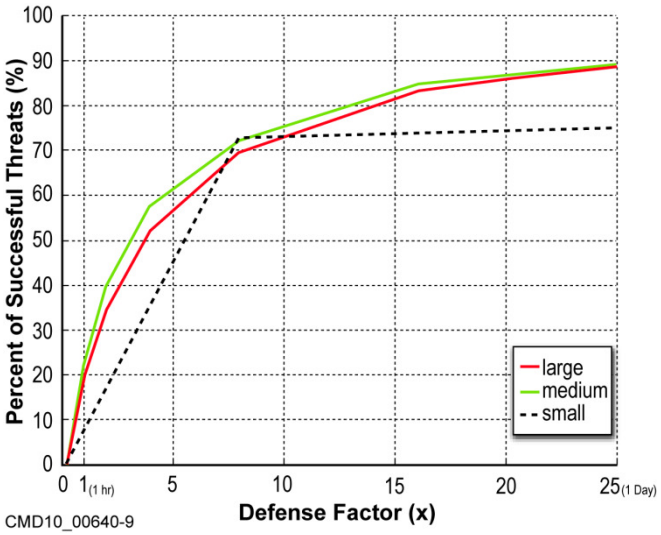


Figure 8 - Defense factor sensitivity

3.8 Additional Metrics

3.8.1 Utilization

As the trend toward virtualization and cloud computing grows, it is necessary to include a measure of how virtual resources are utilized.

Virtual utilization is a measure of how many logical processes occupy physical resources. As an example, virtual utilization can be measured as the ratio of virtual machines to a physical machines number of cores. This ratio can also be averaged across an enterprise network, to give a sense of the total utilization of asset

3.8.2 Attack Noise

In order for an attack to progress through a system, it must execute certain operations. These operations can be thought of as noise in the system.

It is desirable for a cyber defense to increase an attack's noise. The goal is to cause an attack to become an observable outlier from the system's normal operation. If a defense is able to compress the window of opportunity an attack has to execute (e.g., "attack time dilation"), forcing to perform more operations in less time, it would cause the attack to become more detectable. Following this principle, a characteristic function relating execution time to attack noise can be defined. This characteristic function may vary, depending on the attack phase (e.g., pings per second for Foot-printing, port scans per second for Scanning/Enumeration, password authentication attempts per minute for Gain Access).

3.8.3 Effective Surface Area

As is true in any domain, the larger a system is, the more susceptible it is to attack. The effective surface area provides insight into how a defensive technique may obscure a system's attack surface.

To understand the effect a particular cyber defense may have on an attack, a sensitivity analysis of how the defense behaves based on system size can be performed.

4. NETWORK METRICS COLLECTION

4.1 Attack Phase Measurements

It may be difficult in some cases to measure the time spent by an attack in a particular phase. The phase must first be detected, and then a determination made as to whether the activity indicates a new attack instance (from a different source) or an ongoing attack in the same phase. The data must be correlated with time-stamped indications so that the beginning and end may be determined.

Tools exist which can provide low level data, which when collected, aggregated and assigned to a particular attack instance, could support the measurement of attack phase duration and attack noise. In most cases, the reliance on security log data limits this measurement to an after-the-fact, forensic analysis. The position and configuration of the tools should be evaluated and tuned to allow for the most accurate detection of activity indicating these phases. The use of thresholds and alerting could provide better filtering and more rapid results. Honeypots and Honey Networks can be used to collect data in all phases, depending upon their level of sophistication if they are used

4.1.1 Foot-printing

Firewalls or Intrusion Prevention Systems can be used to detect the Foot-printing phase activity, depending upon their sensitivity settings and location within the network.

4.1.2 Scanning

Firewalls or Intrusion Prevention Systems running

4.1.3 Enumeration

Host based Intrusion Detection Systems can be used to detect Enumeration phase activity.

4.1.4 Gain Access

Mining data from system logs and host based Intrusion Detection Systems can provide information on the gain access phase. Behavioral based anti-malware tools may also provide information on gain access attempts.

4.1.5 Escalate Privilege

Escalation of privilege may typically be detected by analysis of log entries, although there may be additional behavioral tools deployed as well. Policy compliance tools may also provide useful information in detecting escalation of privilege.

4.1.6 Pilfering

Monitoring of outbound data at a firewall can be used to detect the ex-filtration of data indicating pilfering. This may also be detected by more specialized behavioral tools.

4.2 Successful Attacks and Total Attacks

An attack instance may be recognized in a phase as described above. Correlation and identification of an attack instance across phases to measure the attack cycle is a tracking problem. This capability would allow the timing of the phases to be aggregated to allow the total attack time to be calculated. In our research, a successful attack was one in which the pilfer phase was achieved. However, attacks may have different objects, so additional measures such as Denial of Service or equipment failure indicators may also need to be correlated to provide a measurement of the entire attack cycle.

4.3 Defensive Measures

Defensive actions may be measured by the system which initiates the defensive actions. These measures, when correlated with the attack measurements can be used to calculate defensive efficiency and the defense factor. Knowing the timing of these defensive actions and then correlating them with the attack measurements, can also provide the data for the attack disruption metrics. Attacks which do not progress beyond a particular phase can be considered to have been disrupted if defensive actions were taking place on the same resources.

4.4 Metrics Correlation

An event monitoring and correlation system, cyber command and control or Network Management System could be extended to pull the appropriate data from the available tools, aggregate, mine and correlate the information to provide monitoring, trending and analysis against a specified Cyber Dynamic Defense SLA defined for the enterprise.

5. RESEARCH RESULTS

5.1 Demonstrated Maneuvers

Successful demonstrations of the NMC were held in December 2009 in Largo, FL, and throughout 2010 in various locations. The demonstrations included the following features: movement of resources and applications across platforms, physical locations, virtual partitions, vendors, etc.; deployment and maneuvering of applications, data and network addresses; deployment of application executable variants; reset / check-pointing of hardware, virtual machines, or application data; periodic, aperiodic or episodic maneuvers; and scripted and orderly, random and continuous, and a hybrid of both maneuvers.

From a server perspective, maneuvers can utilize the organic failover/redundancy schemes if they are present. Similarly, maneuvers may take advantage of any inherent load balancing capabilities. For the work in 2009, maneuvers were conducted using both VMWare and Xen technology, though in the interest of our diversity requirements, the architecture and algorithms can support more technologies than that. It is important to note that we conducted maneuvers of applications with, or without, being in a virtual machine as this is not intended to be just a virtual defensive technique and there was a requirement to support legacy systems.

5.2 Constraints

There are constraints that must be addressed when implementing a maneuvering strategy. We have grouped these into four main areas:

1. Environmental
2. Architecture
3. Network
4. Security

In the environmental grouping, constraints exist for components like memory, processing power and speed, as well as power requirements. Size, weight and power (SWaP) must be calculated into the maneuvering scheme as the architecture is designed.

In the architecture grouping there are component relationship constraints, for instance, supported operating systems, hardware platforms, supported hypervisor types, network subnet requirements, etc.

The network grouping constraints exist for service level agreement (SLA) parameters such as latency, availability, throughput, and priority.

Finally, the security group has constraints around the security zones. The DARPA sponsored-BBN concept of security zones and known vulnerabilities were defined by the Designing Protection and Adaptation into a Survivability Architecture (DPASA) project [2]. Maneuvering should only take place within a contiguous security zone (e.g. the DMZ) and not maneuver from one security zone to another. If maneuvering across security zones is allowed, attacks could be transferred from one zone to another, which might open up vulnerabilities for the attacker to exploit that were not previously accessible. It is also advantageous to specify individual maneuver interval ranges per security zone.

5.3 Challenges

There are certainly challenges to implementing an active defense technology such as Network Maneuvering. A majority of modern technologies and software are not designed to support deception decision-making. Maneuver coordination is made difficult by the multitude of software interfaces within the applications and hardware that would be part of this strategy. Network visualization and situational awareness is, and will continue to be, extremely challenging. Defining measures of maneuver performance and success, through metrics, needs to be accomplished. In doing so, the benefits of maneuvering can be shown with empirical evidence. The current state of vendor licensing models presents a problem to maneuvering schemes since maneuvering relies on using many instances (physical and virtual) and there is no licensing scheme that is designed to support this. Use of high availability features for maneuvering increases license costs as this feature is typically more expensive.

There is also a limitation that high availability (multiple simultaneous uses) licenses for high availability deployments assume that a single operating system is supported.

There are both monetary and cultural barriers to entry in conducting network maneuvering. From a monetary standpoint, there could be the need for significant infrastructure investment, depending on an organization's current posture. Culturally, network maneuvering increases vendor diversity, whereas most businesses are driving their information technology organizations to converge on standardization and support for a limited number of vendors, platforms and configurations.

6. RECOMMENDATIONS

6.1 Maneuvers

Based on simulations we conducted in the laboratory, on real command and control and database applications in a controlled environment, the resulting data showed that maneuvering, artificial diversity and cleansing, do provide improved intrusion tolerance as a lower percentage of attacks were successful. Furthermore, simulation and analysis of real-world threats showed that network maneuvering significantly increased attacker's cost. This cost, as we have said, is in the resources expended (time, et al). Maneuvering algorithms also significantly reduced the probability of success of data exfiltration, and did so for orders of magnitude in some cases. One of our metrics researchers succinctly captured these maneuver results when he stated that as the Maneuver Interval decreases, the effect of increasing the Network size becomes negligible (i.e., the effect of maneuvering dominates over scaling up the physical size of the network). Frequent Maneuvers can be interpreted as artificially increasing the network's size – analogous to the concept behind Synthetic Aperture Radar (SAR).

6.2 Other Findings

The optimal maneuver frequency to meet the stated goals was to maneuver with an interval at least twice (2X) as fast as the fastest time it took an attacker to succeed in our stationary network scenario.

For a more robust performance, we recommend implementation of a client cleanup or complete virtualization scheme. This scheme has the added effect of eliminating any potential persistent threats on clients, as well as ensuring the clients return to a "known good" state periodically.

Maneuvering and artificial diversity in some cases can cause an application to move to a more vulnerable platform if an unknown (0-day) vulnerability exists on the destination platform or vendor type.

7. SUMMARY

The network maneuver commander prototype described is an initial capability set to be used in the proactive defense of cyber command and control systems. As such, they will aid in the objective selection of cyber defense solutions; the goal of this being to select methods and techniques that provide the most benefit. This is not intended to replace the "defense-in-depth" approach, but serve as another element of "defense-in-depth", providing "deception-in-depth".

Raytheon has successfully developed a prototype Cyber C2 System Maneuver and deception architecture and model. We have performed maneuvers, both command-based and automated, of specific applications across different hardware configurations, operating systems, and hypervisors.

The metrics described are an initial set to be used in the characterization of cyber defense systems. As such, they will aid in the objective selection of cyber defense solutions; the goal of this being to select methods and techniques that provide the most benefit. This is not intended to replace the "defense-in-depth" approach. By using these metrics as selection criteria, system developers can ensure high levels of effectiveness in each layer of defense.

Raytheon is continuing to evaluate other candidate algorithms and technologies with ongoing research, and have five patents pending on this technology.

8. ACKNOWLEDGMENTS

Our thanks to all who participated in, provided feedback or otherwise supported this research.

9. REFERENCES

[1] Bangalore, A. and Sood, A., 2009. Securing Web Servers Using Self Cleansing Intrusion Tolerance (SCIT), In *The Second International Conference on Dependability* (DEPEND 2009)June 18-23, 2009 - Athens/Vouliagmeni, Greece. http://cs.gmu.edu/~asood/scit/#publications

[2] Chong, J. Pal, P. Atigetchi, M. Rubel, P., and Webber, F. Survivability Architecture of a Mission Critical System: The DPASA Example. http://www.bbn.com/resources/pdf/GroupPapers_Survivability-Architecture-of-a-Mission-Critical-System-The-DPASA-Example.pdf.

[3] Cukier, M., Courtney, T., Lyons, J., Ramasamy, H., Sanders, W., Seri, M., Atighetchi, M., Rube, P., Jones, C., Webber, F., Pal, P., Watro, R., and Gossett, J. 1993. Providing Intrusion Tolerance with ITUA. 2002. *Supplement of the 2002 International Conference on Dependable Systems and Networks, June 23-26, 2002*. http://itua.bbn.com/

[4] McClure, S. et al. 2006. *Hacking Exposed 5th Edition: Network Security Secrets And Solutions*.

Zero Knowledge Trust Propagation in Airborne Networks

Stephan White
Department of Computer Science
Louisiana Tech University
Ruston, Louisiana

David Irakiza
Department of Computer Science
Louisiana Tech University
Ruston, Louisiana

Abena Primo
Department of Computer Science
Louisiana Tech University
Ruston, Louisiana

Joseph M. Kizza
Computer Science and Engineering Department
University of Tennessee at Chattanooga
Chattanooga, Tennessee

ABSTRACT

In airborne networks with sensitive resources and time critical missions, entity identification and authentication is essential and critical for secure access communications originating from entities outside the networks seeking entry into the networks. However, there are networks and times when entity identification is not required and indeed in some networks, for the security of the outside entity seeking authentication, identity must not be revealed thus preserving the secrecy and privacy of such entities. Using Zero Knowledge Protocol (ZKP), each network node must individually compute its own trust of the entity seeking network resources using its residual and propagated trust and then contribute this accumulated trust in the global network authentication of the entity.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General-Security and protection

Keywords

Mission critical, time, authenticity, anonymity, zero knowledge, authentication.

1. INTRODUCTION

The opening of exchange of information and data between two or more communicating entities depends a great deal on trust between the communicating entities. For fruitful communication, each side in the exchange must be able to trust the other and the data being exchanged. The challenge always is how to build the trust to a threshold required for admission of the entity.

In networking, there are many protocols that allow for the verification of nodes attempting to join the selected network. Different protocols require varying amounts of time to authenticate a new entity. As network technology continues to change it is important to minimize the latency of authentication protocols while still protecting the resources and data of the network

In agile airborne networks (ANs) and other entity groupings that are mission driven, time sensitive, ad-hoc and self-organizing networks; authenticity, anonymity and accountability are essential and crucial, more so than in other similar networks that are less mission critical and time sensitive. However, in these kinds of networks, standard cryptographic authentication protocols like PKI cannot work effectively.

This paper describes a zero knowledge protocol (ZKP) authentication process that involves full participation of every network node in the computation of trust used in the entity authentication by the network. The paper is divided into the following sections: related works, research on zero knowledge authentication, a trust propagation algorithm, trust computation and a proposed network simulation.

2. RELATED WORK

Several interesting zero knowledge proofs are of note here starting with Hannu Aronsson's [1] work which explains zero knowledge proofs starting with the basics including a summary of all of the major zero knowledge studies. Li Lu et al [5] discuss zero knowledge authentication in P2P systems. Kizza et al [4] also discuss zero-knowledge in Agile Mobile Adhoc Networks. Li Lu et al use a modified ZKPI scheme that uses a key exchange but with no third party. Stawomir Grozonkowski et al [2] propose a zero knowledge authentication scheme similar to Kizza's using graph isomorphism. Daniele Quercia et al [7] propose trust propagation among mobile device users using a web of trust. Audun Josay et al [3] also explores the different types of trust propagation in networks. However this also is not on airborne networks. Their application is based on web applications. None of these and other works is using zero knowledge in Airborne Networks with their stringent constraints. And none is using majority computation of trust like we propose.

3. ZERO KNOWLEDGE PROOFS

A Zero Knowledge Proof is a method of proving one's identity to another without revealing anything except that a statement is true. The statement is usually a mathematical one, and also it revolves around a secret. The "Prover" is the one who is seeking validation from the "Verifier" [6]. For example:

Once upon a time in the land of Barkfest the noble King Robert sent down a letter to dear farmer John requesting a basket of his favorite purple plums. The king made this request of John, because only John knew where to find the king's favorite fruit.

So John set out for the river George to cross the only bridge onto the island Marigold where he knew he would find the purple plums in his secret plum patch. As he approached the bridge, John prepared to greet his long time friend, the Ogre Jax. But, to John's surprise, a new Ogre stood between him and the only means of getting to the island with the purple plums.

John said, "Where is Jax?"

The Ogre answered, "Oh Jax is on vacation. I'm his cousin Ralph."

John replied, "Good to meet you Ralph. I've come to pick purple plums for King Robert, so I'll be on my way."

Ralph moved into John's path blocking the crossing over the bridge, "No man crosses this bridge by decree of the Elder Ogres!"

John was shocked because he had been over the bridge so often to get the king's favorite fruit. He said, "I've crossed this bridge many times. Jax knows me well and I'm the only person who knows where to find the plums."

Ralph asked, "So you say you've been on the island... are there any rose bushes along your way to the plums?"

John replied, "No."

Very good, thought Ralph. There weren't any rose bushes on the entire island. Ralph asked, "Do bluebirds nest near the plums?"

John replied, "Yes."

Ralph thought before asking his next question. If there are bluebirds near the plums then this patch must be on the north side of the island. There are many red daisy plants up there too. John would have seen them if he had actually been on the island. "What color are the daisies where your plums are?"

John replied, "Red."

Ralph is beginning to trust that John has been allowed to cross over to the island before today to pick these plums. He decided to ask a final question before fully trusting farmer John. "Will you be heading to the north or south end of the island?"

John said, "North."

Ogre Ralph was convinced that John had been allowed on the island before. He stepped out of John's way and let the farmer go about his business though never knowing where exactly John was headed to pick those delicious purple plums.

A Zero Knowledge Proof must satisfy three properties: Completeness, Soundness, and Zero-Knowledge. Completeness is that if the statement is true, the good Verifier will be convinced if the Prover is honest. Soundness is that if the statement is false, a deceitful Prover cannot convince an honest Verifier that it is true. There is a very small probability that they can. Zero-knowledge is that if the statement is true, a deceitful Verifier cannot learn anything about the honest Prover [6].

4. THE TRUST PROPAGATION ALGORITHM

In standard Zero-Knowledge proofs, the trust of a new entity seeking for network services is computed by the Verifier. Traditional Zero-Knowledge Proofs require that Prover trust by the Verifier starts with an apriori trust of 0.5. This then grows with every correct answer exchanged by the Prover to the Verifier's challenges. The Prover is admitted access to network services when the calculated apriori trust exceeds a threshold value τ. In the proposed algorithm, it is required that beside the one Verifier, there may be one or more additional first layer contact Verifiers. Also required is that every inside node within the network must contribute in the overall trust computation of the outside entity seeking network services. Each node starts its computation with an apriori trust of 0.5 of the immediate neighbor node. This is the first hand trust probably arrived at through direct observation. In addition to the apriori trust, each node in the network, then calculates entity trust based on second hand information it gets from its nearest neighbor if it is an internal node, or the information it gets from the entity if it is a front line Verifier node.

The time λ required by each network node including the Verifiers must be as small as possible. As the network cloud increases to size n, the time required to compute the overall trust needed to authenticate the entity also increases but is bounded by λ_2 which is the time the entity must stay in the proximity of the network cloud.

Due to this inherent time sensitivity of ANs, trust propagation must be rapid. In large networks, the trust propagation time λ may approach and exceed time λ_2. In an attempt to keep the authentication time within time λ_2, we propose the following second-hand trust propagation model in which each Verifier will set its starting trust of the Prover at $S_{(V_2,P)}$.

In the model, the second-hand trust $S_{(V_2,P)}$ between the second level verifier V_2 and the Prover P is computed as a multiple of the scaled trust $\omega_{(V_2,V_1)}$ of the first and second level verifiers and the first hand trust $F_{(V_1,P)}$ between the first level verifier V_1 and the Prover P as shown by the

formula below:

$$S_{(V_2,P)} = \omega_{(V_2,V_1)}.F_{(V_1,P)}$$

$$where$$

$$\omega_{(V_2,V_1)} = \frac{F_{(V_2,V_1)} - \tau_{V_2}}{1 - \tau_{V_2}}$$

$$iff$$

$$F_{(V_2,V_1)} > \tau_{V_2}$$

ω - scaling factor
τ_{V_2} - threshold value of V_2

The resulting $S_{(V_2,P)}$ is added to the apriori trust of 0.5 in the traditional Zero-Knowledge Proof. This allows for a dynamic apriori trust which fluctuates based on the perceived value of rumors from neighboring nodes. By increasing the starting trust, the time required for the Prover to gain access to each successive node is decreased. This allows the Prover to gain access to network services in a shorter time-period, adapting the Zero-Knowledge process to better suit the rapidly dynamic ad-hoc environment.

5. TRUST COMPUTATION

It is reasonably assumed that as a network cloud increases in size, regardless of the efficiency of the propagation algorithm, the time λ required to compute trust for each verifier node will approach λ_2 before the Prover has the opportunity to gain trust from all the network entities. It therefore will not gain access to the network services. It then becomes necessary to enforce a stopping condition for the propagation process so that a global trust of the Prover can be calculated and adopted by the remaining entities.

As this is a work in progress, a global trust computation scheme has not been finalized. However, the following assumptions are being considered from which global trust can be computed:

- The AN is relatively small such that all nodes in the cluster will be able to compute trust in a time λ_0 which is less than λ_2. In this case, the total trust is computed by the following formula;

$$Trust = \sum_{i=0}^{N} \frac{T_i}{N}$$

where;
T_i is the trust computed by each node,
N is the number of nodes in the cluster.

- The AN is small but the time λ_2 that the Prover stays in the cluster is less than the total time λ_0 required to compute trust of all the nodes in the cluster. In this case, we could assume that authentication of the Prover stops at time α_0 where $\alpha_0 = \frac{1}{2}\lambda_0$. At this time, nodes N_{α_0} have authenticated the Prover and if $N_{\alpha_0} > \frac{1}{2}N$, we can employ the majority rule by computing total trust as;

$$Trust = \sum_{i=0}^{N_{\alpha_0}} \frac{T_i}{N_{\alpha_0}}$$

where;
T_i is the trust computed by each node,
N_{α_0} is the number of nodes that have authenticated verifier in time .

Based on the global trust computed from using either of these assumptions, the Prover can then be accepted into the network provided that this trust is above the threshold value τ.

6. PROPOSED SIMULATION

A simulation of ad-hoc AN environments will be necessary to validate our proposed second-hand trust propagation model.

Ad-hoc AN simulations will be initially pursued with static network structures. We will test basic, first-hand knowledge network structures which should support previous Zero-Knowledge Proof simulations. We will then test various second-hand scenarios including but not limited to:

- Second-hand verification with one path to the Prover
- Second-hand verification with multiple paths to the Prover
- Second-hand verification by two adjacent asynchronous first-hand Verifiers.

We will also run simulations of large network clouds in order to determine the effective range of second-hand propagation within a network.

Further simulations after proof of concept through static network simulation will include the use of simulated or physical mobile network entities in dynamic environments.

7. FUTURE WORK

Zero Knowledge Protocols are designed to work between two parties, the Prover and the Verifier. Through verification rounds the Prover attempts to convince the Verifier he possesses a secret. Over time the Verifier may trust the Prover has a secret and allow the Prover communication with the Verifier.

In a network with several front end Verifiers, and a requirement that all nodes in the network must authenticate the entity before it can access network services, a process must be designed to compute the majority trust used against a threshold to authenticate the entity. The algorithm has been developed but we need to design a simulation using this algorithm in a mobile network. The programs developed will then be used to test the algorithm. We will also test signal delays and give a better estimation of the time required to run this protocol.

8. CONCLUSION

In this paper, we have developed an algorithm to compute global trust based on node residual and propagated trust. Propagated node trust is acquired using zero knowledge protocols. The wireless mobile network we are working with to simulate an airborne network is, like its counterpart, extremely time sensitive, and data passed through it must

remain secure. Without the use of a third party, the Verifier works to ensure the Prover can be trusted. The proposed protocol remains secure as the Verifier is 99% sure the Prover can be trusted before allowing communication to occur. Through development, simulation and testing, the proposed trust propagation authentication process may show good results comparable and probably an improvement to existing PKI authentication protocols.

9. REFERENCES

[1] H. A. Aronsson. Zero knowledge protocols and small systems, 1995.

[2] S. Grzonkowski, W. Zaremba, M. Zaremba, and B. McDaniel. Extending web applications with a lightweight zero knowledge proof authentication. In *CSTST*, pages 65–70, 2008.

[3] A. Jøsang, S. Marsh, and S. Pope. Exploring different types of trust propagation. In *iTrust*, pages 179–192, 2006.

[4] J. M. Kizza, L. Bramlett, and E. Morgan. Using subgraph isomorphism as a zero knowledge proof authentication in timed wireless mobile networks. *International Journal of Computing and ICT Research*, 4:19–16, June 2010.

[5] L. Lu, J. Han, L. Hu, J. Huai, Y. Liu, and L. M. Ni. Pseudo trust: Zero-knowledge based authentication in anonymous peer-to-peer protocols. In *IPDPS*, pages 1–10, 2007.

[6] A. Mohr. A survey of zero-knowledge proofs with applications to cryptography, 2007.

[7] D. Quercia, S. Hailes, and L. Capra. Lightweight distributed trust propagation. In *ICDM*, pages 282–291, 2007.

GSM Security Threats and Countermeasures

Saravanan Bala
Louisiana Tech University
Ruston, LA
sma060@latech.edu

Tanvir Ahmed
Louisiana Tech University
Ruston, LA
tah025@latech.edu

Samuel S Kasimalla
Louisiana Tech University
Ruston, LA
ssk015@latech.edu

Travis Atkison
Louisiana Tech University
NH 239, Ruston, LA
atkison@latech.edu

ABSTRACT
Global System for Mobile Communications (GSM) is a well-established and the most widely used cellular technology across the world. Security of the data exchanged forms the core part of any mobile communication networks. GSM uses several cryptographic algorithms for security such as A5/1, A5/2 and A5/3. Recent researches have proved that these algorithms have got limitations and they do not provide the sufficient level of security for protecting the confidentiality in GSM. This paper emphasizes two aspects of improving the security in GSM. One explains the limitations of A5/1 architecture, like weak clocking mechanism and the linear combination of the outputs, and proposes a simple enhanced architecture, which avoids the above mentioned limitations. Another aspect of this paper will address the need for end to end encryption to make the communication over the air more secure. Additional encryption can be implemented by using AES algorithm on GSM network.

Categories and Subject Descriptors
C.2.0 [**Computer Communication Networks**]:

General - *Data communications*

Keywords
GSM hacking, Interception methods, A5/1 Algorithm, Design, Security, Performance, Correlation attack, Clocking mechanism, Linear function, Eavesdropper, Cryptography, Counter measures.

1 .INTRODUCTION
Mobile communication offers wireless connectivity that enables people to communicate with each other anywhere at any time across the world. The openness and the ubiquitous nature of the wireless medium pose security threats to the sensitive information of the user. Any eavesdropper can overhear the information sent over the network. This makes security a most essential entity, and generally it can be achieved by the use of cryptography such as stream ciphering techniques. GSM employs A5/1, A5/3 & A5/8 algorithms to achieve the security. A5/1 algorithm is used to provide privacy over the air communication in GSM standard. Among the cryptographic algorithms used in GSM, A5/1 is considered the strongest encryption/decryption algorithm applied in commercial GSM systems. Recent studies reveal that A5/1 architecture has got some serious security flaws. [1]

Due to these flaws in security, A5/1 offers low resistance to cryptanalysis and it is vulnerable to various cryptographic attacks. Most of the attacks against the A5/1 algorithm target the two primary security flaws in the architecture. They are the way in which the clock-controlling unit is controlled and the linear combining function used to combine the output sequences of three Linear Feedback Shift Registers. Another important aspect of the security flaw in GSM is that it does not provide end-to-end encryption, and this fact makes the system more vulnerable to attacks by eavesdroppers.

In this paper the possible security threats against GSM and the counter measures used to prevent the threats are discussed. First the paper proposes a secured and an enhanced version of the A5/1 algorithm, which is a hardware implementation, and the second part proposes a solution for the end-to-end encryption, which is a software implementation. This paper is organized as follows; Section 2 describes the Architecture of A5/1 algorithm. Section 3 provides a quick overview on the possible attacks on GSM. Section 4 proposes the counter measures to be followed to prevent the attacks, by using a hardware enhancement & an additional encryption through software applications. Section 5 concludes this paper by summarizing the key points and the future directions to be carried.

2. ARCHITECTURE OF A5/1 ALGORITHM
A GSM conversation between any two points is a sequence of frames, each sent in about 4.6 milliseconds. Each frame consists of 228 bits - 114 bits of which is the message from MS to BTS, and the second half bits are representing communication from BTS to MS. The A5/1 algorithm uses 64-bit session key Kc and 22-bit frame number for encryption a session. A5/1 has 64-bit internal state that consists of three LFSRs: R1, R2, and R3 with linear feedback shift registers. Each register is clocked using clock cycles that is determined by the majority rule. The majority rule uses three clocking bits C1, C2 and C3 of registers R1, R2 and R3 and calculates the value of majority m using m = maj(C1, C2, C3). Among the clocking bits, if two or more are '0' then the value of majority m is '0'. In the same way, if two or more are '1' then majority m is '1'. Now, if C_i = m then register R_i will be

clocked (shifted), where i=1, 2, 3. Before a register is clocked the feedback is calculated. Then, the register is shifted one bit to the right (discarding the right most bit), and the bits produced through feedback connections are stored into the left most locations of LFSRs (location zero)**.** A5/1 is initialized with Kc and frame number as described below:

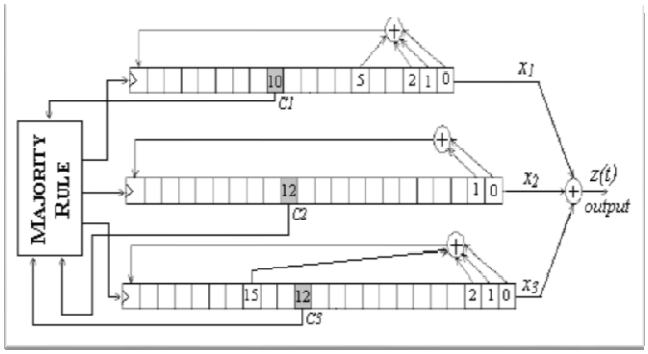
- First the LFSRs are initialized to zero. They are then clocked 64 times, ignoring the irregular clocking, and the key bits of Kc are consecutively XORed in parallel to the feedback of each of the registers. [2]

- In the second step the LFSRs are clocked 22 times, ignoring the irregular clocking, and the successive bits of are again XORed in parallel to the feedback of the LFSRs.

- The LFSRs are clocked in an irregular fashion. Each of them has one tap-bit, Cl, C2, and C3, respectively. In each step, 2 or 3 LFSRs are clocked, depending on the current values of the bits CI, C2, and C3. Thus, the clocking control device implements the majority rule.

- After the initialization procedure, the LFSRs are clocked 100 times with irregular clocking, but the output bits are ignored. Then, the LFSRs are clocked 228 times with the irregular clocking, producing 228 bits of the running key.

In A5/1 at every step two or three registers are clocked, and each register is clocked (shifted) with the probability ¾. At each clocking, each LFSR generates one bit x_i which are then combined by a linear combining function z(t), defined as z(t) = x1⊕x2⊕x3 to produce one bit of the output sequence z(t). [3]



[Figure 1. A5/1 Architecture] [3]

3. POSSIBLE ATTACKS ON GSM:

3.1 Correlation attack

This particular attack on A5/1 algorithm is based on ideas from correlation property of the linear function used in A5/1 architecture. This attack finds the loop holes in the hardware design and exploits the flaw in the design that the key and the frame counter are initialized in a linear fashion. The frames are initialized with the same session key but with different frame counters. This flaw enables cryptanalysts to launch a type of correlation attack, which is almost independent of the shift register lengths. Instead, it depends on the number of times the cipher is clocked before producing the first output key stream bit. In the A5/1 this number is 100. If the number is increased the attack becomes weaker, and vice versa happens. Given the key stream, the objective of the attacker is to recover the initial state of the running key generator. This is called an initial state recovery attack. In GSM system, the initial state of the shift registers consists of a linear combination of the publicly known frame counter and the secret session key. By finding the initial state, the secret session key can be recovered easily which then can be used to decrypt the original message. [2]

3.2 Brute Force Attack

This method has been used by cryptanalyst's right from the time A5/1 algorithm came into commercial existence. It is based directly on the problems statements and definitions of the concepts involved. This attack can be carried out by aligning pattern at the beginning of the text, then moving the text from left to right and then comparing each character of the pattern to the corresponding character in text until all characters are found to match or a mismatch is detected. This is a tedious and time consuming process. [4]

3.3 Recent attacks

Recently an A5/1 cracking project has been announced at the 2009 black hat security conference by cryptographers Karsten Nohl and Sascha Krißler. They used Rainbow tables with distributed computing and challenged that their methodology can be used to crack any cipher with key size up to 64-bits. This poses a serious threat to the user using GSM and this fact emphasizes on the need for the improved version of the A5/1 algorithm. [5]

4. PROPOSED COUNTER MEASURES

4.1 Hardware Enhancement

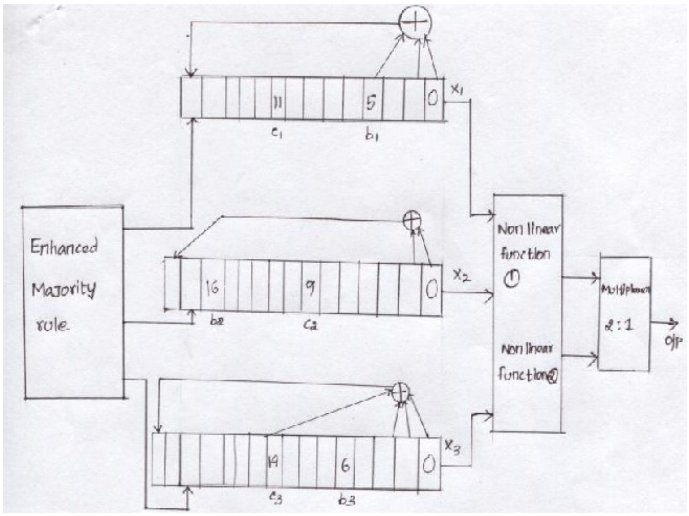
As discussed earlier clock-controlling unit and the linear combining function used in the A5/1 architecture makes the system more vulnerable to cryptographic attacks. In this proposal both the above flaws are rectified with an enhanced version. A simple architecture is proposed with an improved clocking mechanism, and the linear combination function has been replaced with two nonlinear functions with better efficiency. This model creates more irregular clocking and makes it harder for the cryptanalyst to crack.

The majority function used in the contemporary model can be utilized in this proposed scheme with a slight extension, In which the clocking bits in each register will be increased from one bit to two bits. This enhanced majority rule uses 6 clocking bits of the registers R1, R2 & R3. It computes two majority values using the six clocking bits of the registers. Further the clocking bits are selected in a way that there are no regularity exists between their positions.

The architecture is designed in a way that it takes b1, b2, b3 and c1, c2, c3 clocking bits as inputs and calculates two majority values m1 and m2 using the majority functions as:m1= major(b1, b2, b3), m2= major(c1, c2, c3). The clocking mechanism in our proposed scheme works as following; let's take two illusionary empty sets S1 & S2. If $b_x = m_1$ then register 'R_x' will be in the set S1 and if $c_y = m_2$ then the register 'R_y' will be in the set S2. According to the enhanced majority rule a register is clocked (shifted) provided the register is in common between both the sets

S1 & S2, where x,y = 1, 2 [6]. This enhancement in the majority rule overcomes the weaknesses due to poor clock-controlled mechanism and greatly improves the security of A5/1 cipher.

By utilizing this combination 64 distinct sets of registers are clocked against the 8 cases of clocking produced by the current architecture of A5/1. [6]



[Figure 2 Proposed A5/1 Architecture]

This solution improves the clocking mechanism. In A5/1 stream cipher, a linear combining function is utilized to combine the sequence of outputs from three LFSRs. As we know linear combining functions are cryptographically weak functions, so they need to be improved to prevent the attacks. [9] Therefore to overcome the weaknesses due to linear function, two cryptographically better nonlinear combining functions are used. We will ensure that the combining functions are not fixed so we use a multiplexer to change it dynamically. These two improvements in the architecture will improve the linear complexity where by it can withstand the correlation attacks, Algebraic attacks and linear complexity attacks.

4.2 Additional Encryption – Software Application

In GSM the radio link between mobile station and the base station is encrypted using A5/1, whereas the rest of the network transmits the data in clear form. So this poses a security threat which can be prevented by using end to end encryption. An efficient method for end to end secure communication is to encrypt the speech signal at the user end. When following this approach we need to ensure that the encrypted data transmits through GSM networks with sufficient accuracy so that the received information can be decrypted correctly at the receiver without any confusion at the receiver's end. One form of solution is to use the transmission of encrypted voice GSM Data Call CSW(Circuit Switched Data).

This technique has been employed in software products like SecureGSM. Another practical way of utilizing is the usage of connection based on packet switching. Connection based Packet switching is a method of transmission where small chunks of data are transmitted over a channel dedicated to the connection which is defined and preallocated in all the involved nodes before any packet is transferred and it is ensured that it is delivered in the correct order. This is a best offered method where a high quality of service is guaranteed throughout the connection.This technique has been used in the software product Babylon nG. This product is widely used due to the security it provides. Both the above mentioned techniques use Diffie-Hellman key agreement protocol for the ciphering key exchange and AES cipher for encryption of the voice. [7]

It is also further experimentally proved that the implementation of AES cipher for encryption of voice provides more robust and efficient system. The implementation of AES additional encryption uses classes available in JAVA package javax.crypto. Separate functions for encryption and decryption have been implemented in MATLAB using JAVA cryptography API and it proves that the AES provides better security than other algorithms when it is susceptible to brute force attacks.[8]

5. CONCLUSION

In this paper, we have presented an enhanced version of A5/1 stream cipher to be used in GSM standard. This proposed scheme generates cryptographically better key sequence than the current version of A5/1. We also presented that the end to end security can be provided with an additional encryption using AES. Further for future implementation the current limitation of GSM can be avoided by using the UMTS (Universal Mobile Telecommunications System) which uses UEA1 encryption with a key length of 128 bits. UEA1 is better compared to A5/1 with GSM since the key length is twice as long which makes the algorithm difficult to crack. Also it uses mutual authentication between handset and network which makes the system more protected against the false base station attack..

6. REFERENCES

[1] Secure Mobile Communication Using Low Bit-Rate Coding Method. IEEE paper published by Wasif, M.; Sanghavi, C.R.; Elahi, M.;

[2] Another attack on A5/1. IEEE paper published by Patrik Ekdahl and Thomas Johansson.

[3] Enhanced A5/1 Cipher with Improved Linear Complexity. IEEE paper published by Musheer Ahmad and Izharuddin.

[4] Introduction to the design & analysis of algorithms by Anany Levitin.

[5] Based on the presentation given by Karsten Nohl on the "26th Chaos Communication Congress (26C3)" conference.

[6] Security Enhancements in GSM Cellular Standard. IEEE paper published by Musheer Ahmad and Izharuddin.

[7] Communication Security in GSM Networks published on 2008 international conference on security technology by Petr Bouška, Martin Drahanský.

[8] Implementation and Analysis of AES, DES and Triple DES on GSM network an IEEE paper published by Sachin and Dinesh kumar

[9] Construction of nonlinear Boolean functions with important Cryptographic properties - Advances in Cryptology by Sarkar and Maitra.

Securing a Wireless Network - WPA Attacked and Defended

Hend Nuhait
Louisiana Tech University
Ruston, LA 71272
han002@latech.edu

Chetan Sakpal
Louisiana Tech University
Ruston, LA 71272
cvs004@latech.edu

Travis Atkison
Louisiana Tech University
Nethken Hall 239
Ruston, LA 71272
atkison@latech.edu

ABSTRACT
Wireless networks are being increasingly used these days. They have become very important in today's daily human life. People are not just using wireless networks at their workplaces, but also at their homes. Everyone is using them, irrespective of their ages. Nowadays, people are "connected" even through their cell phones. However, the wireless networks are not secure. They are easy to crack. In this paper, we focus on WPA wireless network because they are considered as one of the most secure wireless networks. We describe how easy hacking a typical WPA wireless network is. Moreover, we explain some ways of making WPA wireless networks more secure.

Categories and Subject Descriptors
C.2.2 [Computer-Communication Networks]

General Terms
Security, network, hacking, defense.

Keywords
Hacking WPA, defending WPA, network security.

1. INTRODUCTION
What is wireless network? A wireless network is the way that a computer is connected to a router without physical link, using radio signal frequency [9]. In recent years, people have started using wireless networks in their everyday lives. It allows them to use their networks anywhere and anytime. Furthermore, in the last decade, networks have become affordable to almost everyone. Even kids use networks to play multiplayer games. Also, organizations are using networks to make the interaction between clients easier, quicker, and efficient.

While most people like to use networks, especially the wireless networks, we have to ask this question, are they secure enough? Many network users do not choose the option of changing the default passwords, or even setting up a password. Attackers could take advantage of such practices and attack the whole network, in view of the fact that for attacking a system an attacker just has to find a single loophole.

The network security problem ranges from hacking a victim's personal computer and stealing their personal pictures and documents to doing illegal crimes using the victim's machine and ID. As we see criminal's job is becoming easier. They do not have to leave their place to do harmful things anymore.

In the network field, there is no such thing as absolutely secure. Furthermore, when a network developing company releases a new network, they claim that this network is absolutely secure. After people buy it and work with it, the smart bad guys find the faults or loopholes, and hack the network. Later, they share their knowledge with other non-smart bad guys. Therefore, our goal is to find defense mechanisms to secure the networks.

In the next sections, we describe a WPA wireless network. Then, we explain an easy way to hack a WPA wireless network. Finally, we suggest some ways to secure the network

2. WPA
WPA is a wireless network protocol that stands for WI-FI Protected access. WPA wireless network is a security standard for WiFi wireless connection. In addition, WPA wireless network was made by 802.11 standards protocol to avoid the weaknesses of the WEP wireless network, which is short for Wired Equivalent Privacy [6]. When the 802.11 standard protocol developers implemented the WEP wireless network, their plan was to have a wireless network as secure as the wired networks. Unfortunately, the WEP wireless network has many loopholes and security problems [6]. Attackers could easily hack the WEP encrypted key, because it uses RC4 encryption algorithm [5]. RC4 is a stream cipher. The idea of it is to take the data bytes and XOR them with some random bytes [3]. Another problem of WEP is that every computer that connected to the same network is using the same key. Therefore, any one uses these machine can hack the other clients' traffic [6].

WPA wireless network algorithm was different from WEP. WPA contains a key hash function, a message integrity code(MIC), and key management scheme. The key management scheme is used to prevent the use of the same key, which make sure that one of WEP problems is solved. Also, it helps distribute the key [6]. More over, in WPA, access point and client use a shared master key. This key is used to produce two keys, a 64-bit MIC key that is produced by the data and the MIC, message integrity code, and 128-bit encryption key that is used to encrypt and decrypt the data [5].

The summarized idea behind WPA wireless network is that the users have to authenticate themselves, then, they get the permission to use the network. They authenticate themselves with the four way handshake. The authentication or the four way handshake is done in two steps. The first step is that the user connects to the network. Then, the WPA asks the user to authenticate via authentication server. After that by access point, both the user and the authentication server authenticate each other [12].

WPA uses Temporal key integrity protocol "TKIP" encryption method which gives WPA wireless network the noticeable features. TKIP is a modified version if WEP wireless network. It uses a difficult mixing function to mix the keys. Mixing helps avoid key attacks [4]. Key mixing function takes the TK, the TA and the 48-bit IV, then it produces 128-bit key. Whereas TK is a 16-bit Temporal Key which is produce during authentication bye the key management scheme, TA is 6-bit Transmitter address, and IV is the initialization vector. The IV also is called TKIP sequence counter, which increases after each package. Since IV increases after each package, it will help avoid attacks [6].

3. HACKING WPA

Since hackers need only one hole to attack, while machine owners need to close all the holes, our approach to defend the network is to first attack it. That means we first find holes and gaps in WPA wireless network, and then, search for solutions to secure it.

Ironically, WPA is considered as one of the most secured wireless networks and still a Google search of "hacking WPA" yields 425,000 hits. In this section, we will explain one of the easy and efficient (from hacker's point of view) ways to hack WPA.

The first thing to be noticed is that, many hackers prefer using Linux operating system since it is an open source system. Hacking with other operating systems is not impossible. However, Linux environment, precisely Backtrack environment, is compatible with most attacking and defending tools [3]. It is also planned to be an environment that support people who are interested in security fields, whether they are professionals or beginners [1]. For that reason, in our attacking we used Backtrack operating system, on Linux virtual machine.

After installing Linux virtual machine and Backtrack4, we could start the WPA attack. First, we need to find a wordlist dictionary file. The wordlist file contains all the common passwords, like names, common word, default password, dates, license plates, phone numbers, zip codes, city names etc. There are many free wordlist dictionary files online. In addition, there are some free tools to generate wordlist files. Moreover, these dictionaries have the words by several languages.

Once all the materials are installed, we open a terminal window. Before we start hacking we have to know our "the hacker" interface and MAC address. That can be found by writing the following commands. The next command shows the interface

```
airmon-ng
```

And this command shows the MAC address

```
macchanger -s [Interface]
```

Then, the first step of hacking, we search for the target network that will be attacked. To do that we write the following command:

```
airodump-ng [The hacker's station interface]
```

The above command will show us all the available networks. We make sure that we choose a WPA or WPA2 encrypted wireless network. After we choose the network that we want to hack, it is better to save the information, because we will need it in the next step. The important information is the 'bssid' which is the MAC address for the router. Also, we need to know the channel of the network.

Next, as we notice in the previous section, WPA network requires a handshake between the station and the Router. To crack WPA we need to find a station that is already on the network. So, we can disconnect it from the network then we acquire the handshake. The next command is used to find the station that used that network:

```
airodump-ng --bssid [MAC of Router] -w [FILENAME] -c [CHANNEL] [ADAPTER]
```

Where, MAC of the Router and CHANNEL means the MAC address and the channel that we found on the previous step. The File name is the place that we want to save the handshake in. The ADAPTER means the hacker adapter, which we find in the first step.

After finding a station and making sure that it is on the network, because otherwise there is no point of the next steps, we save the victim's station MAC address.

Next, we start de-authentication of the station from the router and get ourselves connected. That is done by repeating the last command. At the same time, open another terminal and write this command:

```
aireplay-ng -0 15 -a [MAC OF ROUTER] -c [MAC OF SYSTEM ON NETWORK] [ADAPTER]
```

where number 15 indicates the number of attempts made to de-authenticate the station, MAC of System on Network means the MAC address of the victim's machine. That command will try to de-authenticate that station from the network and the let the hacker get into the network.
Here comes the last step which is cracking the network's password. This command is written to do the cracking, and it will not work except if the hacker could de-authenticate any machine connected to the network, and authenticate the hacker's machine in its place:

```
aircrack-ng -e [ESSID - Name of network] -w [WORDLIST] [FILENAME.cap]
```

Where filename.cap is the file that we save the hand shake on, and the wordlist is the address of the wordlist dictionary that we are using. The previous command will match the network password with each password in the wordlist dictionary, the attackers will be lucky if they password is in their dictionary. [8][11]. With this command we hacked the WPA secure wireless network, and cracked its passkey.

4. DEFENDING WPA

As we notified previously, we think the most efficient way to close the gaps and solve the problems is, especially network security problems, by being the attacker first. The first and easiest way to protect WPA wireless network is by making the password strong. The first step is setting up a password. Next, if the network already has a password, it is better for the user to change it regularly. Hackers use wordlists, dictionaries or password

cracking tools like John the Ripper. John the Ripper is a tool used to crack the weak password [15].

We conclude that users have to have a very strong password. The password should be long. Moreover, it should have a combination of capital and small letters, numbers, and special characters. The password, also, should not be something predictable, like name-birthdate, or name-license plate. As we have mentioned in the previous section, the wordlist usually has all the combination of the known words or sentences. In summary, if the password is not in the wordlist dictionary, it would be very difficult for the attacker to crack the WPA wireless network.

Wireless networks, when used on a small range like in an office or at home should prefer using WPA-PSK i.e. Pre-shared key. This doesn't rely on the complicated authentication server. Here, every device encrypts the network traffic using a 256 bit key. On entering an 8-63 bit passphrase, a random 256 bit key is generated by applying the PBKDF2 key derivation function to the passphrase, using the SSID as the salt and 4096 iterations of HMAC-SHA1 [12]. It may also be entered as a 64bit hex value. PBKDF2 stands for password based key derivation function. It uses a pseudo random function for example hash. SHA stands for Secure Hash Function. A brute force attack may not fail but might be considered worthless if a truly random key is used. Worthlessness of a Brute Force attack might eventually be the failure of it. But, if weaker passphrases are used, the shared key system is still vulnerable. Of course, a larger random key is more secure and desirable than a smaller one. The system could be made more secure if the passphrase is changed quite regularly. For example, if we assume a comparatively weaker key is being used which could be hacked in say 10 days using a brute force attack, then changing the passphrase would make the vulnerable system more secure because every week the attack would have to start from the first combination again. So a tradeoff between feasibility to change the passphrase regularly and ability to remember a more difficult passphrase has to be achieved. Such a system could be termed "almost completely secure". So for most of the applications WPA-PSK should be utilized so that desired security levels could be achieved.

When wireless networks are used on a large range, or due to some other reason, changing a WPA key regularly is not feasible, shared key alone may not be the best technique to be utilized. In such cases, a simple approach to maintain an access control list or MAC filtering could be used. MAC filtering is one of the simple techniques that alone do not look very promising as MAC spoofing could be very easily done. So some other techniques like not allowing MAC de-authentication of the devices on the network should be used. De-authenticating a device on the network is one of the ways in which MAC spoofing is done.

One more simple approach would be to hide the Service Set Identifier (SSID). Before doing this, the default login and password of the Router should also be changed. It may also help to change the default SSID name. Turning off the SSID beacon would not help much if it is easily guessable. SSID is broadcasted by any router so that devices would detect the network. If these settings are changed so that the SSID is not broadcasted by the router, other devices won't be able to detect the router. But, expert hackers could easily detect SSID of wireless networks using software. This technique when used with the above MAC filtering gives enough security to keep out casual users or even some inexperienced attackers.

The above two techniques, though not very effective, do help in improving security of a wireless network. Tutorials on hacking the WPA usually provide demonstrations of steps like using the dictionary, etc. They generally do not show how to get around such basic modes of security. It is also a good approach to use more techniques of security if known. Since, these are additional security techniques to the main WPA technique, using them should not be a problem. Sometimes if the techniques are interdependent on each other, and if one of them is weak and it is compromised, having a stronger one is of no use, since a single loophole into the system might be enough for the attacker. But, when the above mentioned techniques are compromised, it doesn't affect the security mechanism of WPA, since WPA does not depend on them.

The other solution is iJam which is a technique that was proposed by MIT [2]. The idea of iJam is that it will not enable the hacker from extracting other client's signals i.e. those signals which are not planned for him. In the WPA wireless network case, it will prevent the attacker from capturing the four way handshake.

Furthermore, here is the executive summary of the iJam. First, iJam works with the network physical layer. The senders resend their messages several times. Then, the receiver jams one of the messages. It could be the original or the repetition. Fortunately, the hackers do not know whether the message is jammed or not. Therefore, they are unable to hack the messages. On the other hand, it is easy for the receivers to decode it, because they can distinguish the clean messages from the jammed ones [2].

5. CONCLUSION

Wireless networks are growing rapidly. On the other hand, the networks' attacks are also increasing. Beside that, the crimes are becoming much easier. Therefore, people want to make sure that their network are secure.

Wi-fi Protected Access (WPA) wireless network was developed by 802.11 standards protocol to avoid the weakness in the Wired Equivalent Privacy (WEP) wireless network. In this paper we addressed some of the WEP problems and security issues. Furthermore, we explained the WPA features which made it more secure than WEP. After that, we explained the algorithm behind the WPA wireless network. Also, we discuss some of the security methods that was used to prevent eavesdropping and attacks.

We show in our paper how a typical WPA secured wireless network is attacked and taken control of and how we could utilize the features of not only the WPA encryption but also of the basic wireless networks to improve the security of the overall wireless network. Our attacking method was based on linux and BackTrack environment. However, our method works only if the password is contained in our wordlist dictionary.

WPA-PSK should be given much more importance when it comes to using wireless networks at homes or small offices. Initial setup of a router could also include most of these basic steps. As of now, WPA is completely secure if put to its optimum use, but history shows security of a technique is also inversely proportional to time.

After this experiment we found out that the easiest and cheapest solution to secure the wireless networks is to make the password difficult to guess. The password should contains numbers, letters, and symbols. It also should be long enough, so, attackers will not able to find it in their wordlist dictionaries.

Last but not least, wireless networks still need more work in order to be fully secure. Also, WPA has some holes and weak points. So, it can be attacked and its traffic can be hacked. The future work is to close the WPA holes and weak points. Also, a secure wireless

network protocol need to be developed, that protocol should prevent the weakness of both WEP and WPA.

6. REFERENCES

[1] BackTrack http://www.backtrack-linux.org/downloads/

[2] Gollakota, S. And Kati, D. 2010. iJam: Jamming Oneself for Secure Wireless Communication. Technical Report (June 7,2010). MIT.

[3] Himalayan,P., Hannikainen, M.,and Himalayan,T., and Saarinen J. 2000. Hardware Implementation of the Improved WEP and RC4 Encryption Algorithms for Wireless Terminals. In Proceedings European Signal Processing Conference, Tampere, Fenland.

[4] Martin, B. Tews, E. 2008. Practical Attack Against WEP and WPA. (November 8,2008) Proceedings of the 2nd ACM conference on wireless networks.

[5] Ohigashi, T. .And Morii, M. 2009. A Practical Massage Falsification Attack on WPA. IEICE Information System Researcher's conference

[6] Omen,V., Harvard, R., and Kjell, J.2004. Weakness in the Temporal Key Hash of WPA. (April 5,2004) ACM SIGMOBILE mobile computing and communication review

[7] VMWare http://www.vmware.com/products/player/

[8] http://www.youtube.com/user/mushroomHEADBANGERS

[9] http://www.home-network-help.com/wireless-network.html

[10] Http://www.openxtra.co.uk

[11] http://www.question-defense.com/2010/01/10/how-to-capture-a-4-way-wpa-handshake

[12] Yang, Z. 2006. Link-Layer Protection in 802.11i WLANs with Dummy Authentication. In WiSec

[13] http://palisade.plynt.com/issues/2007May/wpa-security/.

[14] http://searchmobilecomputing.techtarget.com/definition/Wi-Fi-Protected-Access.

[15] http://www.openwall.com/john/

Defending Against Cross Site Scripting

Harry May
Louisiana Tech University
Ruston,LA
hlm012@latech.edu

Apurba Dhungana
Louisiana Tech University
Ruston,LA
adh039@latech.edu

Jundong Chen
Louisiana Tech University
Ruston,LA
jdc074@latech.edu

Travis Atkison
Louisiana Tech University
Nethken Hall 239
Ruston, LA 71272
atkison@latech.edu

ABSTRACT
Currently, the most prevalent Web attack is Cross Site Scripting. Cross Site Scripting takes advantage of a vulnerability within a web browser. A web page can use a simple form to request information from a user. This information can be simple items such as first name, last name, etc. The user enters the requested information in a normal string fashion. The attacker, however, will not enter a normal string but will instead enter HTML codes that can compromise the security of the web site. This paper discusses some of the types of attacks and some of the defenses that can be used to combat the attacks. Web developers need to know how this attack is used and how to minimize the vulnerability by filtering all data input.

Categories and Subject Descriptors
K.6.5 [**Security and Protection**]: Miscellaneous

General Terms
Theory

Keywords
XSS, PHP, HTTP, Javascript

1. INTRODUCTION
A battle is happening every day. Attacks against web sites are occurring at an increasing rate. Defending against these attacks is an on-going struggle. One of the most prevalent attacks is the Cross Site Scripting, also called XSS. XSS was considered a moderate severity vulnerability for some time but advent of XSS worms and virus raised its profile. This is an attack against a computer security vulnerability[1] found in web applications enabling attackers to exploit these vulnerabilities within that web site. The commonality among these attacks are dynamic web pages and insufficient checking of web server input and output. In order to do a cross site scripting attack, the attacker will generally exploit the vulnerabilities in dynamic web application that requires user input. If this input is not validated and is used by a web application to generate a personal page for the user, it might be exploited by an attacker. Cross site scripting usually starts with input supplied by the attacker such as executable code loaded into a user's browser. The code may be written in Javascript, VBScript, ActiveX, Java, Flash, or any other browser-supported technology[2]. Javascript language is mostly used to enhance the client side display in the web page. JavaScript execution is confined by a sandbox mechanism that allows the code to do certain limited operations only. Another restriction in Javascript and other browser-side programming languages is called same origin policy. Same origin policy allows scripts running on a web page to access methods and properties on other pages if they originate from the same web site. Even though java script is confined by sand-boxing and same origin policy, if the user can be enticed to download a malicious java script code from a trusted website exploitation mechanism, this is known as cross-site scripting. The impact of an attack depends on the objective of the attacker. Compare the attack of a financial institution versus a simple web site. The attack on the financial institution would be considered more serious. More resources should be applied toward the security of this site versus the security of the simple web site.

2. TYPES OF ATTACKS
There are two primary types of attacks: persistent and non-persistent. The persistent attack is more significant than the non-persistent attack because of the nature of the data. Persistent data is stored on the server usually in the form of a database or file system. An attacker will generally submit XSS code to an area of a website where it is most likely to be visited by other users. Some of these areas may be blog comments, user reviews, chat rooms, etc. When the normal user clicks on this area, the XSS code will run. All the data collected from the XSS code can then be stored for later viewing by the attacker. Another persistent type of attack is the cookie. If a cookie is compromised by the attacker, that attacker can assume the identity of the user on that web site.

The non-persistent attack is more commonly seen than the persistent attack. Generally, in this attack, the users are

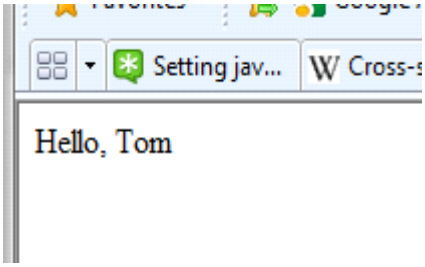


Figure 1: A simple example(Hello,Tom).

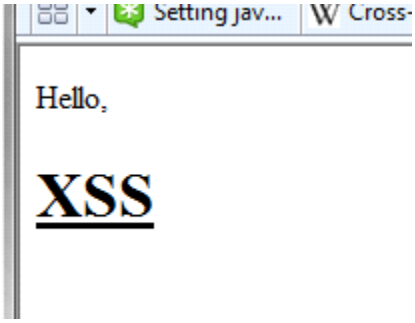


Figure 2: A simple example(Hello,XSS).

tricked (social engineering) into clicking on a web page link. This link is specially crafted in such a way as to contain malicious code or redirect the user to a malicious web site. Another method is to take the data provided by a user and generate a page of results without properly validating the input. To exploit this vulnerability, the attacker will enter command strings, using HTML characters, that can give them control of the web site. A simple example is shown below using this method.

3. A SIMPLE EXAMPLE OF NON-PERSISTENT ATTACK

Suppose a PHP page supplies a form for you to enter your name. After you enter your name (Tom) in the form, the page will display a welcome page such as "Hello Tom!". The code to generate that page would be something like the following:

<?php echo "Hello, {$HTTP_GET_VARS['name']}!";?>

Passing "Tom" as the argument to the page would create the line

http://localhost/hello.php?name=Tom

The displayed page would look something like that shown in Figure 1.

Now, instead of entering your first name, you enter something like:

<h1><u>XSS</u></h1>

The resulting page would look something like that shown in Figure 2.

This is something an attacker will do. Instead of entering normal requested data, they will enter code in the correct format that will exploit the vulnerability on that page. This is called the Proof of Concept XSS exploit. It accounts for 75% of XSS vulnerabilities existing in real-world web applications[3]. This simple test shows two things: First, the contents of the message can be replaced with arbitrary data for the web browser. Second, the server is not checking for invalid or inappropriate data.

4. AN EXAMPLE OF A PERSISTENT ATTACK

Consider a web site that has a vulnerable comment system. The address to get to this site and display a product may be something like `http://www.shoponline.com/product.jsp?id=1`. This will display the product and the comments associated with that product.

Product Details	
ID	1
Name	Airmax
Category	Running Shoes
Price	$50.00

Review
Keshia Kaam - Great running shoes, comfortable, and long lasting. Highly recommended.
To submit your review click here.
`http://www.shoponline.com/product-review.jsp?id=1`

The attacker clicks on the link to enter comments. Instead of entering normal comments, the attacker leaves a review containing malicious code.

Name: Ray Bhadur
Email Address: rayhack@hackme.np
Rating: *
Subject: Don't buy this shoe.
Description: Bad value for the money<script>window.location ='`http://hackem.np/getcookie.jsp?site='%2Bdocument.domain%2B'%26cookies=%1Bdocument.cookie`';</script>

Now, if you return to the original location of `http://www.shoponline.com/product.jsp?id=1`, you will see a new comment under the reviews.

Review
Keshia Kaam - Great running shoes, comfortable, and long lasting. Highly recommended.
Ray Bhadur - Don't buy this shoe.Bad value for the money.

Any user visiting this site who views this product information will get attacked. There is no email needed or no links from any other site required. Just visiting this site can cause the user to have their accounts hijacked, cookies stolen, or malware installed.

5. DEFENSE STRATEGY

The simplest and most effective way to defend against attacks is to validate the data before processing it any further[4]. There are two additional points about performing

the validation. The first is there should be a trusted environment in which the validation is valid. Second, the environment should have an input checkpoint where all input has to go through this checkpoint. With the trusted environment, after validation has occurred, the data should be able to be used inside the trusted environment without having to revalidate it at any point. In order to get data within the trusted environment, it must go through the checkpoint. The checkpoint will check the data for the correct format and then pass it into the trusted environment. There can be multiple checkpoints allowing for different type and format of data. The easiest form of checkpoint is one with a filter that "rejects all". Since that is not feasible, the checkpoint should check for valid data and reject everything else. The reason for checking for valid data is because it is easier to check for valid data compared to checking for invalid data.

6. REGULAR EXPRESSIONS

One of the ways of validating input is to use Regular Expressions. A Regular Expression is a method of matching search strings looking for a pattern. For example, using the pattern "car", will find the value in strings such as "car", "cartoon", and "bicarbonate"[5]. Most computer users are familiar with the asterisk search pattern such as *.TXT. This will search for all files that have an extension of TXT. Advanced patterns would use something like the brackets, [and]. A bracket expression matches a single character contained within the bracket. For example, [abc] matches the character "a" or "b" or "c". To make sure that the user entered a properly formatted email address, use a string like

$$\hat{}[A-Z0-9.\_\%+-]+@[A-Z0-9.-]+\.[A-Z]\{2,4\}\$$

The IEEE Std 1003.1, 2004 Edition, has complete definitions for the expressions[6]. Regular expressions are very powerful and they can be used for many applications other than filtering input.

7. HIDING THE EXPRESSION

An attacker will try to hide the message or the link from the web page user. One method they use is to create links containing hexadecimal characters instead of the normal Ascii characters. An example of a malicious link would be

Click to win $1,000,000

The scriptcode in this case may contain something like

<script>x=document.cookie;alert(x);</script>

The attacker would hide the real information from the user with something like

<a href="http://www.microsoft.com@
%77%77%77%2E%65%78%70%6C%6F%72%61%74%69%6F
%6E%61%69%72%2E%63%6F%6D%2F%72%65%71%2E%61
%73%70%3F%6E%61%6D%65%3D%3C%73%63%72%69%70
%74%3E%78%3D%64%6F%63%75%6D%65%6E%74%2E%63
%6F%6F%6B%69%65%3B%61%6C%65%72%74%28%78%29
%3B%3C%2F%73%63%72%69%70%74%3E">
Click here to win $1,000,000

The link looks like it comes from Microsoft but it doesn't. It is a little known internet syntax as defined in RFC 1738, "Uniform Resource Locators (URL)," at `ftp://ftp.isi.edu/in-notes/rfc1738.txt`. It takes the URL form of `http://username.password@webserver`. The real message after you convert the hex code to ascii is:

www.explorationair.com/req.asp?name=
<script>x=document.cookie; alert(x);</script>

8. STEPS TO PREVENT XSS

Start with the Proof of Concept attack. A script string should be submitted as a parameter to every page of the web application. All responses from these pages should be checked to see if there is any indication that the input may be susceptible to an XSS vulnerability. The script string could be something simple such as

"><script>alert(document.cookie)</script>

After submitting the script string to every possible input location, the next step is to submit a unique string such as 'mytestcharacterstring' to the same input locations as the script string. What you are looking for is to see if that unique string is displayed back to the browser. Each appearance of that string is a possible XSS vulnerability. The associated input filters need to be checked for vulnerabilities.

Sanitization is the next step in preventing attacks. When an input string is sanitized, HTML characters will be changed or removed from the input string. For example, the left and right angle bracket are key characters in HTML encoding. After sanitization, the left angle bracket will be changed to < and the right angle bracket will be changed to >. This will prevent simple script attacks because now the script is not a valid command. Another part of sanitization is the use of Regular Expressions as described above. There are libraries, tutorials, tools, examples, books and many other references on Regular Expressions at [7].

Browsers use different API commands to display information to a web page. Some of these commands are capable of accessing data using carefully crafted URLs. Some of the commands are document.location, document.URL, document.URLUnencoded, document.referrer, and window.location. This means that static as well as dynamic web pages may be vulnerable. Some of the other APIs that must be checked are document.write(), document.writeln(), document.body.innerHtml, eval(), window.execScript(), window.setInterval(), and window.setTimeout(). The reason these need to be checked is because data is passed to them. A carefully scripted Javascript string could be passed as a parameter.

A very simple prevention mechanism for cookies is to flag a cookie as HttpOnly in the Set-Cookie header. When the cookie is flagged in this manner, supporting browsers will not allow JavaScript to access the cookie directly. The cookie will still be submitted in the HTTP headers but the cookie information will not be returned when using the document.cookie command.

9. CONCLUSION

Nowadays, Web application is getting more complex day by day. Different scripting languages are used to provide the user with a better user interface and dynamic functionality. However, this functionality also creates vulnerabilities to cross site scripting. In this paper we describe how cross site scripting works, simple examples, and a different defense strategy to defend against this vulnerability. Creating a website that is not vulnerable to cross site scripting requires effort from web application developers, browser manufactures, and administrators. Numerous papers, examples, and tutorials are available on the internet on how to minimize XSS. It is up to the developers to make sure their site is as secure as possible.

10. REFERENCES

[1] Cross site scripting. In *http://en.wikipedia.org/wiki/Cross-site_scripting*.

[2] Harold Tipton and Micki Krause. *Information Security Management Handbook*. CRC Press LLC, 2009.

[3] Dafydd Stuttard and Marcus Pinto. *The Web Application Hacker's Handbook*. Wiley publishing,Inc., 2008.

[4] Michael Howard and David LeBlanc. *Writing Secure Code*. Microsoft Press, New York, 2003.

[5] Regular expression. In *http://en.wikipedia.org/wiki/Regular_expression*.

[6] Regular expressions. In *http://www.opengroup.org/onlinepubs/009695399 /basedefs/xbd_chap09.html*, 2004.

[7] Regular-expressions.info. In *http://www.regular-expressions.info/*.

A Different Approach to Network Security : The Best Defense is a Good Offense

Miguel D. Gates
Louisiana Tech University
Nethken Hall 232
Ruston, LA 71272
mdg022@latech.edu

Umesh Dhital
Louisiana Tech University
TTC 208
Ruston, LA 71272
udh001@latech.edu

Timothy Lindsay
Louisiana Tech University
Innovation Lab
Ruston, LA 71272
trl011@latech.edu

Travis Atkison
Louisiana Tech University
Nethken Hall 239
Ruston, LA 71272
atkison@latech.edu

ABSTRACT

Network security consists of the provisions and policies accepted by the network administrator to prevent and monitor unauthorized access, which can lead to misuse, modification, or denial of the computer network and network-accessible resources. With attackers targeting the slightest vulnerabilities in a network to gain access and wreak havoc, it is important to have a strong defense system in place to combat such threats. These threats are evolving rapidly with advancements in hacker tools, techniques, methods, scripts, and automated hacking malware, thus challenging the processes used to protect networks. To thwart these attacks, security too must evolve. In evolution, inherited traits change as a result from interactions between processes that introduce variation into a population, and other processes that remove it. So in terms of security, a network should evolve from a preventative technique into more a more proactive offensive deterrent designed to counter-strike and cripple the attacker's system. This paper discusses a method in which a system implements a variety of counterattacks and countermeasures once it feels its threshold of security has been breached as its primary method of defense.

Keywords

Phishing, SQL Injection, Network Security, Countermeasures, Counterattack

1. INTRODUCTION

With the influx of computers, due to the "technological age", making life easier, the world has become saturated with them now; hence, network security has become more prevalent. The expansion of computers and network systems seems to have created a direct relation to the increase of hackers or attackers who want to infiltrate these systems for personal gain. With these incessant network attacks, such as IP flooding and buffer overflows, people have tried to create defense protocols to prevent such attacks from working. Defenses are still susceptible because attackers only need one single point of vulnerability, whereas a good defense has to consider all access points. A new approach to network security is instead of blocking potential infiltrations, what if the network attacks an intruder when its system is compromised. Hence, the system's best defense is a good offensive strategy that will limit the effectiveness of an attackers' ploy.

In order to deter an attacker from advancing, this offensive scheme should force a hacker's system to parlay its attacks and set up its own defense mechanisms, thus preventing the breached system from further attacks. Of course, counterattacks are contingent upon the type of attack the intruder is attempting. For example, if an intruder is trying a simple ping attack in which they attempt to flood a server or gateway, the system would respond by doubling the pings in reverse to the intruder to flood his network even more quickly; therefore, counterattacking any intruder with an offensive barrage when compromised. This paper will discuss various network attacks and intrusions, and how, if implemented correctly, a system plans to counter these intrusions instead of just defending against them.

This paper proceeds as follows: Section 2 gives a background of common types of attacks (in order to defend any attack, one must know the way an attacker might advance); Section 3 discusses the various tools and setup of the network, as well as implementation of offensive schemes; Section 4 discusses the testing procedures and the results obtained from testing the network with new defense-to-offense technique; Lastly, Section 5 will conclude the paper and discuss possible future research to enhance the systems' defense.

2. BACKGROUND

A network attack is any operation to disrupt, deny, degrade, or destroy information resident in computers and computer networks, or the computers and networks themselves. There are many types of network security attacks that are applicable today. In order to implement even simplistic attacks,

one must first understand how they are done. The most common of these attacks are the following: ICMP attacks, SQL Injection, and phishing attacks (though not an actual attack to a network, it still in a very common attack). Each attack is different in creation, implementation, and how they can affect a network or server.

The Internet Control Message Protocol (ICMP) is one of the core protocols of the Internet Protocol Suite. It is chiefly used by the operating systems of networked computers to send error messages—indicating, for instance, that a requested service is not available or that a host or router could not be reached. Some popular ICMP attacks are Ping floods, Pings of Death (PoD), and Smurf Attacks [6]. One of the only ways to completely stop these attacks is to disable ICMP. A lot of software requires access to a site on the internet or a connection to a specific port and pings the host to make sure it is there. Therefore, if ICMP is disable then the software will not work. ICMP attacks also cannot be stopped with normal firewall rules because ICMP does not use a port number. ICMP requests are sent from one system and the receiving machine sends back responses. ICMP attacks are used for denial-of-service (DoS). A ping flood is from one or multiple machines to one or more machines on a network. The ICMP packets fill the network with traffic, effectively slowing down the network and the target systems. A ping of death (PoD) is an older attack that is not as common now as it once was. PoD attacks rely on knowing the boundaries of the receiving hardware. Typically, the largest packet that can be sent is 65,535 bytes. Sending a packet of 65,536 bytes is not allowed. This attack works by fragmenting the packet. When the receiving hardware receives the packet and reassembles it, there is a buffer overflow. This buffer overflow is likely to cause the system to crash. Another common ICMP attack is the Smurf Attack. It works by spoofing the IP of a system on the network [4]. Now the attacker will ping flood the network broadcast address. The broadcast address then forwards these packets to every system on the network, and all of the systems send responses to the actual computer that owns the IP address.

Another type of attack is an SQL (Structured Query Language) injection, which is a form of attack on a database-driven web site to execute unwanted SQL commands [5]. The attacker takes the advantage of unverified/un-sanitized user input to retrieve useful information for which he/she may not be authorized. The basic idea is to convince the application to run SQL code that was not intended. With injected SQL, the attacker can retrieve, insert, modify, delete or basically do anything with your database. Many web pages take input values from the user and use them to construct SQL queries without checking. Attackers may use this vulnerability to inject the queries they want.

Port scanning is a prevalent attack in which a software application is designed to probe a network host for open ports. To port scan a host is to scan for listening ports on a single target host. Port sweeping, however, is to scan multiple hosts for specific listening ports. In its most basic form, port scanning will simply send out a request to connect to a target computer on each port and notes which ports respond. This is considered the first step for an attack and can disclose sensitive information about the host.

Some of the most common attacks are ones that target the user's self-awareness versus the actual machine he or she is using. This is most prevalent in a phishing attack, which is a form of social engineering. A phishing attack (a portmanteau of "people" and "fishing") is a process of attempting to acquire sensitive information such as usernames, passwords, and credit card details by masquerading as a trustworthy entity in an electronic communication—by "baiting" unsuspecting users, so to speak. It is often carried out by email or instant messaging, in which it directs a user to enter details into a fake website whose look and feel are identical to the legitimate one. If done correctly, phishing can be one the hardest cons to spot in that it plays on one's emotions—fear that a breach has left the user vulnerable and supplying sensitive access information is the only to prevent the vulnerability from enhancing [1]. There are many techniques to phishing. These include spear phishing, link manipulation, filter evasion, website forgery, and even phone phishing. Spear fishing is a targeted form of phishing in which a person directs phishing towards banking, social networking, and file sharing sites. It is one of the most common forms, as most of these users are susceptible to phishing attacks. Link manipulation is a form of technical deception designed to make a link and website in an email appear to belong to the spoofed organization. Misspelled URLs are a common trick (because most users do not pay attention to slights in domain names). Filter evasion uses images instead of text to make it harder to be detected by anti-phishing filters. Website forgeries complete the con. They are done with the aid of a technique called cross-site scripting, which can use a trusted website's own scripts against the victim. Though phishing can seem tricky, the best way to combat it is by raising the awareness of victims to recognize plausible phishing attempts.

3. IMPLEMENTATION

In order to implement an offensive-attack defensive scheme, we first created a non-hostile system that is susceptible to attacks in order to deploy the aforementioned network attacks. For this, we used a dummy network: one main server with multiple Virtual Machines (VMs) capable of making a network size of 2 or 3 hosts. With multiple hosts, the first line of defense was a "ghost" server that plays a "shell game" on the unsuspecting intruders. This shell game is designed to lead the attacker into a maze of misdirection as they spend countless wasted time chasing virtually nothing trying to find the real server. This can be done by redirecting to the real machine through a certain port number, which may be protected with a port knocking sequence. If the intruder is skilled enough to bypass this, that triggers the network's counterattacks mechanism.

3.1 Components

A more in depth description of the components needed for the network structure are given below:

- Server
- Clients

A Linux server is running multiple VMs. It also has control of the environment and network capabilities. The clients

consist of two VMs running Ubuntu; these instantiations are able to be accessed by the Linux server. Visual Studio is used to create *C#* programs; these programs are necessary for implementing the various attacks and defense mechanisms.

3.2 Attacks and Counterattacks

As mentioned before, the counterattacks are contingent on the attacks being employed. For our case, we implemented three separate attacks: an ICMP (ping) attack, an SQL injection attack, and a phishing attack, along with the proper counterattack. We will use sample code and working applications to test the attacks and calibrate the system's responsiveness to those attacks. Every attack will result in a counterattack towards the intruder.

3.2.1 ICMP

ATTACK

For ICMP, the simplest attack is the ping attack. As such, the "Ping of Death" is one line of code, `ping -l 65538 192.168.2.3`, where `-l` is an argument denoting size, 65510 is the packet size in bytes, and 192.168.2.3 is the IP address of the intended victim. The goal is to generate a packet size that exceeds 65,535 bytes to try to crash the victim's operating system.

COUNTERATTACK

Ping flooding will be counterattacked by sending back two responses per request. Similarly, a ping of death will result in a response back of equal size.

3.2.2 SQL Injection

ATTACK

There are multiple types of SQL injection, from bypassing authentication to causing destruction like modifying, inserting or deleting records in the database tables [3]. To authenticate without any credentials:

```
Username: 'OR ''=' Password: 'OR ''='.
```

To drop a data-base table: `Username: ';drop table users-`. To execute xp_service control to manipulate services:

```
http://localhost/script?0'EXEC+master..
xp\_service control+'start',+'server';-.
```

In our case, we created a random database of names and account numbers using a random generator in *C#*. Using a modified version of Louisiana Tech's B.O.S.S login, we plan to use the database as a basis for our SQL injection attack.

COUNTERATTACK

To counter-attack, once we detect an SQL injection-type attack, we will show false progress, which makes the attacker waste time.

3.2.3 Phishing attack

ATTACK

To implement a phishing email, an attacker will spoof websites (`wget` or copying the *.css* file from the HTML source) and/or links (creating own hyperlinks) and redirect a user to a specified site, controlled by the adversary. They can even modify the DNS lookup and redirect from that point.

COUNTERATTACK

Although phishing is not a direct attack to a network, we still plan to implement a counterattack for it. For instance, we receive a bogus email asking for personal information. We can check the validity of the email by checking the authenticity of the IP address or DNS server in the header using an acquired package that we can use in a script, `whois.dll`. If the email fails authenticity, we knowingly entertain the attacker by visiting his website. However, once there, we actually use a method of SQL injection to delete his database, thus rendering him back to square one.

4. TESTING AND RESULTS

In order to test our counterattacks of the ad-hoc network, we first created a scenario in which our network could be compromised. By making the network susceptible to such attacks as SQL injection and phishing, we were able to implement the aforementioned attacks and test the capability of the appropriate counterattack.

4.1 Phishing

To create a proper phishing scenario, the first step was to spoof a well-known website. For all intents and purpose, Louisiana Tech University's B.O.S.S. was chosen as the target website. To spoof this site, the .css (cascading style sheets) was copied and saved to a new destination, making a new website with a different address, as seen in Figure 1. The next step was to falsify an email suggesting that a

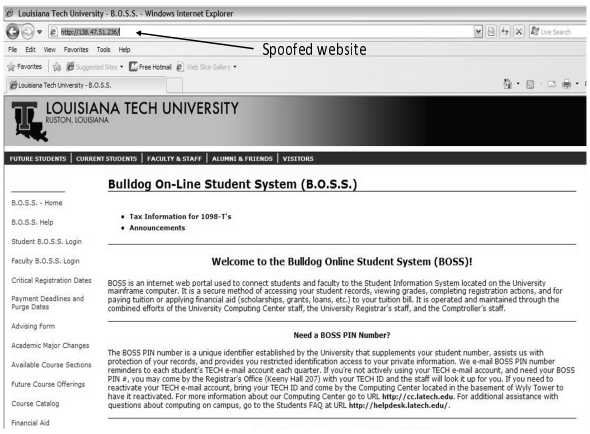


Figure 1: Spoofed Louisiana Tech University B.O.S.S. website

user's account needs validation, thus subjecting the user to attempt login from our modified website by placing a valid, but unauthentic link in the email to redirect them. Once there, it would be the user's job to input his or her student number and password. In order to steal this information and store it in a personal database, we used HeidiSQL. HeidiSQL, a client for web-developers using the MySQL-Database [2]. MySQL is a relational database management

system that runs as a server providing multi-user access to a number of databases. HeidiSQL allows one to manage and browse their databases and tables from an intuitive Windows interface. By managing, a user can view all databases on a server or connect to a single database to work with its tables and data; they can also create new ones, alter existing databases' name, do character set and collation, and drop (delete) databases. To manage these databases with HeidiSQL, users must login to a local or remote MySQL server with acceptable credentials, creating a session. Within this session users may manage MySQL Databases within the connected MySQL server, disconnecting from the server when done. Figure 2 is a screenshot of the HeidiSQL interface and shows a sample gathered database. In order to obtain

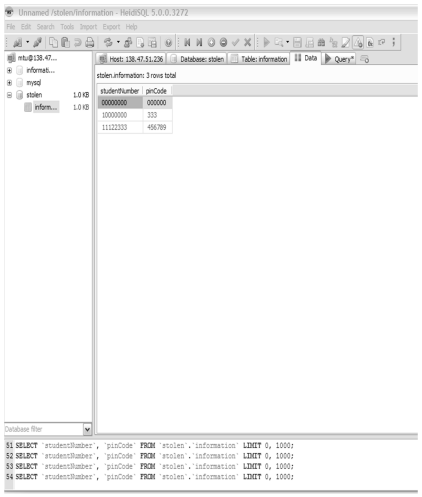


Figure 2: HeidiSQL

the inputted information into a database, MySQL scripts were written that performed these actions. An example of the code is portrayed in Figure 3. The preceding actions

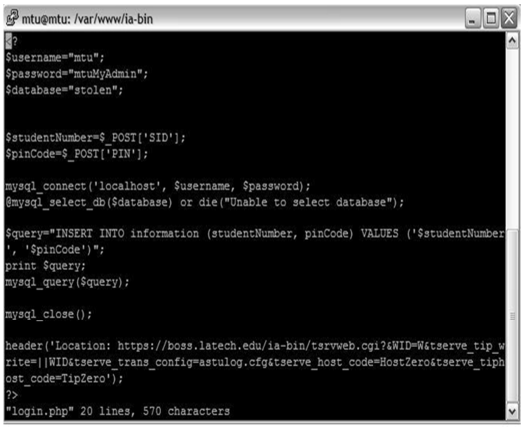


Figure 3: Database query using MySQL

were all used to setup an attempt to phish information from an unsuspecting user. Given an email in question, we can check the authenticity of the DNS or IP address from the header of the email. If discovering the email is in fact a phishing email, the systems stated counterattack is launch an SQL injection in return to delete the database of the attacker, thus rendering the attacker empty-handed from not only our information, but others as well. This SQL injection in implemented by inputting `'delete - -` into a query field. An example of this is shown in Figure 4. This culminates a counterattack scenario for a phishing attack.

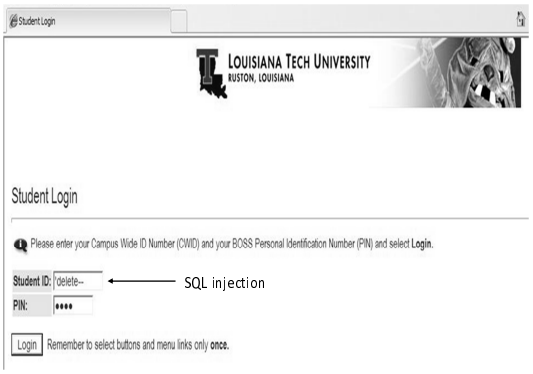


Figure 4: Phishing counterattack (SQL injection)

4.2 SQL Injection

An SQL injection is set up using some of the same tools as the phishing attempt. In addition, using a custom-written C# program, a random database was created. Each real entry in the database has contact information and an account number. The table format was identical between the ghost server and real server. An example of the query fields used can be seen in Figure 5. If SQL injection is attempted on

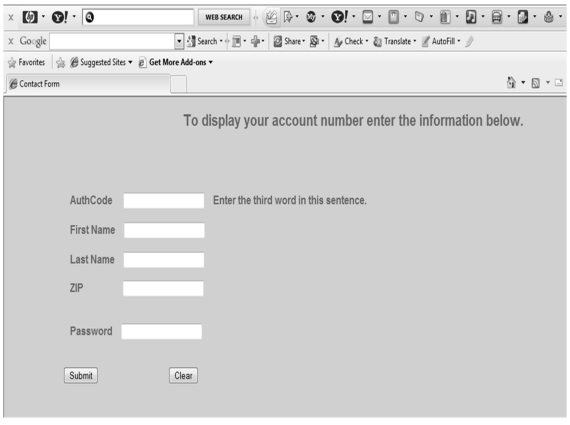


Figure 5: SQL Injection Query Form

our login page then we will secretly query a fake database, which was populated with data from the random-generated program. Otherwise, we will actually attempt a connection to the real server. The attempts can be detected by filtering for special characters, such as `'- -'` and `'*'` or by MySQL keywords like `'or'`, `'select'`, and `'delete'`. We can redefine the database error messages using custom-defined error pages.

5. CONCLUSIONS AND FUTURE WORK

This research proved to be able to demonstrate a proficiency in defending with a proactive offense. Though taking a beginner's approach, these small steps will help deter attackers from intrusion in networks using a non-typical defense. In short, this system should adequately react to ping, SQL, and phishing attacks with either an evasive (as with the SQL injection) or aggressive response. Though these attacks were done as a proof-of-concept and not as a defense on real networks, this could possibly open the door for this type of defense system to be viable.

5.1 Future Work

This research was done strictly as a proof-of-concept. In the future, we plan to possibly implement our defense on a live network and monitor its response. Also, currently, our system is catered to certain attacks. We would like to make a more in-depth model that will encompasses a greater variety of attacks. In keeping with making the defense stronger and more robust, we also would like to make the system more autonomous, as the user will be abstracted from launching counterattacks or monitoring the system.

6. REFERENCES

[1] J. Brott. Analyzing malware and malicious content. *Hacking*, 5(3):34–43, March 2010.

[2] H. Europe. HeidiSQL. `http://www.heidisql.com/`.

[3] S. McClure, J. Scambray, and G. Kurtz. *Hacking Exposed 6: Network Security Secrets and Solutions*. McGraw Hill, New York, NY, USA, 2009.

[4] I. S. Systems. Smurfing. `http://www.iss.net/security_center/advice/Exploits/IP/smurf/default.htm`.

[5] W. Wang. *Steal This Computer Book 4.0*. No Starch Press, San Francisco, CA, USA, 2006.

[6] WROX. Programmer 2 Programmer. `http://p2p.wrox.com/c-programming/25941-ping-c.html`.

A Call for the Consideration of Individual Differences in Cyber Security

John E. Buckner V Tilman L. Sheets

Louisiana Tech University

Abstract

Cyber security is an increasingly expanding field that tackles ever-evolving threats. The existing research on cyber security addresses a number of areas, such as training and development. However, there is a sparse research focusing on individual differences. The current discussion emphasizes the lack of current research on individual differences and cyber security and the benefits that such research may provide, such as allowing for the identification of superior cyber warriors and by identifying employees that pose security risks.

Cyber security is a growing area of concern in every industry today. Some of the most pressing cyber security threats include malware, targeted hacker attacks, and insider threats. Cyber security, a field which is part of information technology (IT) and computer science, has developed in response to the increasing demand for dealing with ever-present and evolving threats. Much of the literature on cyber security focuses on technology, environment (context), or training and development (cyber security educational programs). However, there is little research regarding cyber security and individual differences. The current discussion focuses on the disparity between the volume of research on training and development and the sparse amount of research on individual differences regarding cyber security.

Examining Cyber Security

Cyber security can be examined through two broad perspectives. These can be divided into training and development and individual difference perspectives. Given a training and development perspective, educational programs aim to fulfill industry demands for employees with cyber security knowledge. According to Newman (2007), Oklahoma State University's cyber-security offerings began with a one-credit summer course in 2002 and has since expanded to offering a bachelor's of technology degree in cyber security. The market is demanding that educational programs produce IT professionals who have technical skills in cyber security and will not require a large investment, in terms of additional time and training, on the part of the organization. Programs should focus on providing students with practical technical skills and prepare them for immediate placement in a wide variety of cyber security applications (Newman, 2007).

Training is also an important part of preparing all employees in dealing with cyber security. McCrohan, Engel, and Harvey (2010) studied the impact of awareness training on employee

security behavior. Two training conditions, a low- and high-information condition, revealed that users had significantly stronger passwords after training in the high-information condition.

Another perspective on cyber security emphasizes individual differences. Individual differences in personality and motivation have been acknowledged in terms of what makes a hacker, what differences exist between hackers who act defensively versus those who act maliciously (i.e., white- versus black-hat hackers), and what characteristics identify insiders. Doty and O'Connor (2010) describe cyber warriors (those protecting cyberspace) as talented individuals with specialized skills who are ethical and maintain professionalism (that is, dedicated to their organization, not out for personal financial gain). Randazzo et al. (2005) examined the primary motivation and characteristics of insiders, employees who use their inside knowledge and privilege to perform illegal or destructive acts. Their research found that most incidents required little technical sophistication, insider actions were planned, most insiders had a financial motivation, there was no common insider profile in terms of age, marital status, or position, and insiders committed the acts while on the job.

Need for Individual Differences

Individual differences have not been thoroughly examined in the context of cyber security and the workplace. In contrast to the identifying characteristics examined by Randazzo et al. (2005), individual differences that are typically examined in the psychology literature, and which may be more appropriate, include the big-five (i.e., extraversion, emotional stability, openness to experience, conscientiousness, and agreeableness), intrinsic and extrinsic motivators, and integrity.

Searching the education, psychology, and computer science and technology databases for the terms ‘malware’ or ‘computer science’ with ‘personality’ or ‘individual differences’ yields no results. In contrast, searching for ‘malware’ and ‘training’ or ‘cyber security’ and ‘training’ yields 17 and 38 results, respectively. Greater attention has been paid to training programs and the preparation of cyber warriors than to individual predispositions that impact cyber security. While technical expertise and knowledge can be acquired through educational programs and experience, personality characteristics that are the mark of a successful cyber warrior (e.g., creativity, determination) cannot be acquired merely through education.

Incorporating Individual Differences

Exploring employee individual differences can provide considerable benefit to organizations. The following suggestions are aimed to help advance the individual differences perspective and promote research on personality and cyber security.

Examining cyber warrior individual differences can help to identify those most capable of performing in a demanding, changing field. Lisa Vaas (2007) describes hackers as individuals who pay attention to small details, are innovative, are able to use and see old techniques in new ways, and are able to exploit even the smallest vulnerability. These characteristics are represented well by the personality characteristics of resiliency, tenacity, creativity, and general problem-solving ability. Future research could explore whether these individual differences are indeed related to employee performance.

Individual differences are also useful in predicting ‘at-risk’ employees. These employees may be insiders or naïve employees who may be inclined to encounter and interact more with malware (e.g., following insecure links, downloading from unknown sources, not securing

passwords). An individual-difference focus would allow for greater prediction of which employees are likely to engage in these behaviors. For example, there may be a common insider profile with regards to personality.

References

Doty, J., & O'Connor, T. J. (2010). Building teams of cyber warriors. *Army*.

McCrohan, K. F., Engel, K., & Harvey, J. W. (2010). Influence of awareness and training on cyber security. *Journal of Internet Commerce, 9, 23-41*.

Newman, S. (2007). Cyber security: Are you prepared? *Techniques*.

Randazzo, M. R., Keeney, M., Kowalski, E., Cappelli, D., & Moore, A. (2005). Insider threat study: Illicit cyber activity in the banking and finance sector. Carnegie Mellon University.

Vaas, L. (2007). Inside the mind of a hacker. *eWeek*.

Building Secure, Resilient Architectures for Cyber Mission Assurance

Harriet G. Goldman

> *"You are going to be attacked; your computers are going to be attacked, and the question is, how do you fight through the attack? How do you maintain your operations?"* – Lt Gen Ted Bowlds, Commander, ESC, 28 Jan 09

Motivation/Background

Today's information technology (IT) environments are increasingly subject to escalating cyber attacks. Cyber threats vary widely in sophistication, intent, and the consequences to the targeted systems and networks. The range of attackers extends from users who unintentionally damage systems to hackers, to cyber criminals, to full-scale cyber spies and cyber warriors; their intentions span from annoying vandalism to economic threats to taking out the electric grid or defeating armed forces. Similarly, the target of the attacks can vary from a single computer or router to an entire on-line banking system, business enterprise, or global supply chain. At the same time, our missions and businesses fall along a spectrum of criticality—from desirable to necessary, essential, and mission or safety critical. Given the broad spectrums of threat, intent, and consequence to mission-critical functions, determining exactly where our mission systems lie in this continuum of dimensions is vital to determine the appropriate level of investment and response.

The notion that we can achieve 100% protection is not only unrealistic but also results in a false sense of security that puts our missions and businesses at serious risk. Consequently, we must compensate for our inability to achieve full protection by ensuring that we can accomplish our missions despite cyber attacks. The cyber defenses generally available today help address the low-end threats against our less essential systems, but are often ineffective against most forms of cyber attacks targeting our most mission-critical systems. It is at the high end of the continuum that architecture resilience will matter most—to enable continuity of mission critical operations and support rapid reconstitution of existing or minimal essential capabilities or the deployment of alternative means of accomplishing the mission.

This paper offers ideas along the full spectrum of cyber security, but concentrates on architectural resilience against the upper end of the spectrum, where the stakes are high, the mission or business is critical, and the adversary is sophisticated, motivated, and persistent. However, many of the same techniques are valuable at the low to medium levels of threats and consequences because they can significantly reduce the operational impact and cost of cleanup after an attack. Even if the intentions and consequences of the threat are currently not very serious, we must keep in mind that today's massive data thefts or passive reconnaissance can quickly escalate into data and system modification, surreptitious commandeering of control, or denial of essential services with far more dire mission impact in the future.

The cyber adversary continues to have an asymmetric advantage as we fruitlessly play Whac-A-Mole in response to individual attacks. To reduce the adversary's advantage, we must proactively rearchitect our systems to impede or neutralize attacks and diminish their impact and consequences. While we cannot stop all attacks or make them totally ineffective, rearchitecting for resilience will make adversaries' attacks less likely to succeed, will minimize consequences to critical operations when they do succeed, will increase the adversary's cost and uncertainty, and may act as a deterrent against future attacks.

Recent events demonstrate that Government is increasing its attention to resilience. While these actions clearly indicate senior leaders understand the importance of resilience, we are just beginning to understand what it means to turn the concept into practice. Much work is needed to define and validate resilience: techniques and strategies; policies to promote operational and system resilience; risk decision methodologies, analytic processes, and acquisition guidance; and metrics for measuring resilience improvements and evaluating progress. Moreover, funding must be aligned to budget cycles to reflect these needs and build momentum.

Recent Government actions addressing resilience:

- *The National Infrastructure Protection Plan: Partnering to Enhance Protection and Resiliency*, produced by DHS last year, placed greater emphasis on the importance of building resilience into critical systems, especially IT systems.
- Darrell Darnell was appointed to a post on the White House National Security Staff in the newly created Office on Resilience in October 2009.
- Resilience was listed as one of the five homeland security missions in the recently published *Quadrennial Homeland Security Review*.
- Goal 4 of the *OSD/NII DoD IA Strategic Plan* is "Prepare for and operate through cyber degradation or attack."
- The FAA just initiated a Commercial Space Transportation Grant Program to ensure the resilience of the United States space transportation infrastructure.

However, game-changing technologies, techniques, and strategies can make transformational improvements in the resilience of our critical systems. This paper explores the art of the possible from which to begin evaluating the viability of promising strategies and techniques for resilience, singularly and in combination, to determine which are the most cost-effective to pursue. Some of the suggestions are already commonly embraced and in practice, whereas other notions are new and speculative. Often these new approaches cost more, but sometimes they reduce costs or improve reliability and thus may be part of the business justification. Decisions on how to proceed must weigh the cost and impact of failed critical operations against the cost and benefits of incorporating resilience. More important, in today's environment of sharply increasing cyber threats, these approaches can make the difference between success and failure, between life and death. To do nothing is to accept defeat and pay the price in terms of failed missions and business objectives.

Defining Resilient Architectures

Goals and Objectives

The term *resilience* has many definitions depending on the context and application. For a computing paradigm, the simple definition from the University of Kansas's ResiliNets Project proves most useful: "Resilience is the ability to provide and maintain an acceptable level of service in the face of faults and challenges to normal operation."[1] Resilience is related to survivability, which builds on the disciplines of security, fault tolerance, safety, reliability, and performance. This paper focuses on how to achieve resilience in our mission-critical computing environments against specific patterns of cyber attacks. It covers recommendations for how critical processing systems should be designed, deployed, and operated to support adaptation, scaling, replacement,

[1]University of Kansas ResiliNets Wiki and Wikipedia, "Resilience is the ability to provide and maintain an acceptable level of service in the face of faults and challenges to normal operation." https://wiki.ittc.ku.edu/resilinets_wiki/index.php/Definitions (accessed on 11 August 2010).

reconfiguration, and recovery in the event of an unexpected disruption, degradation, or compromise of critical data, system components, or services.

Improved technology, architectural advances in modularity, integration, standards, and service-orientation, and new distributed processing paradigms facilitate the creation of resilient architectures. At the same time, resilience is challenged by the last decade of unconstrained connectivity and business cost-cutting measures, which have resulted in extensive use of homogeneous, commercial-off-the-shelf (COTS) hardware and software, unknown interdependencies and connections, and reliance on outsourced services and network infrastructures whose pedigree, development life-cycle, and systems management and operations are out of an organization's control. Therefore, the question of how to achieve resilience is part of a larger question: how to construct resilient systems from components whose resilience characteristics may be limited, unknown, and possibly unknowable; and how to express resilience characteristics so that we can evaluate and measure them in desired capabilities throughout the system development life-cycle.

While this paper focuses on architectural strategies, attention to operator resilience is crucial to mission success given all the unknown unknowns. We need to improve our woefully inadequate training approaches and to exercise under realistic cyber intrusions. Traditional techniques derived from continuity of operations (COOP), disaster recovery (DR), operator training, and red team exercises remain vital to ensure that we are prepared to respond during a failure or natural crisis. They also play a crucial role in validating technical and operational resilience of the architecture to such events. However, as implemented and practiced, these techniques do not address current cyber attacks and must be revisited.

Similarly, implementing best security practices is insufficient. We must change our current philosophy of assuming we can either keep adversaries out or detect their breaches of our first-line defenses. Instead, we must assume some adversary success. Too often we do not know the cause of anomalous system behavior when we detect it or are not even aware that an attack is in progress. Consequently, we need to design systems that are more agile and adaptive AND more resistant and resilient to compromise.

> *"It is not the strongest species that survive, nor the most intelligent, but the ones most responsive to change"* – Charles Darwin

The goal for improving the resilience of our architectures is not to seek perfect protection against advanced threats—a goal that is elusive at best. Instead, it is to adopt and implement design strategies and techniques that support a balanced combination of protections, detections, and adaptive technical and operational responses that dynamically evolve in response to current and future cyber events.

The objectives fall into two categories: to make architectures more resistant to unintentional incidents and targeted attacks, and to make them more resilient to initial and subsequent compromise. The first category creates a more secure and resistant foundation and helps inform and trigger operational responses. This category of objectives can be met by techniques to protect systems and deter attackers, and to detect compromises when possible. In many cases we may already be addressing these objectives to some extent, but we must either do this more or do it better. The second category focuses on altering the environment in order to constrain or isolate critical capabilities and data, to reduce consequences, and to support agility and adaptive responses. Collectively the following five objectives can help achieve architecture resilience.

1. **PROTECT/DETER** by disincentivizing the adversary and raising the protection level of our systems. We cannot expect inadequately protected systems to be resilient, any more than we can hope to achieve

functional assurance from spaghetti code. First-order actions are to protect our critical infrastructure applications, services, and data as best we can against both known and possible cyber threats by:

> *"[w]e may improve deterrence if we…ensure resiliency and continuity of service. If opponents believe …attacks will have little effect, they will be less likely to launch them."* – Securing Cyberspace for the 44th Presidency (December 2008)

- Incorporating security technology and operational best practices (for confidentiality, availability, and integrity)
- Reducing the number and severity of vulnerabilities
- Making our IT systems more trustworthy by applying high-assurance techniques
- Adhering to fundamental security principles of policy enforcement**,** least privilege, and simplicity and modularity for trust analysis
- Designing for high availability, integrity, confidentiality, safety, agility, and scalability

Disincentives to attackers can be introduced by incorporating design strategies and impediments that:

- Make the attacker's task more difficult, costly, time-consuming, or uncertain;
- Increase the likelihood that the attacker will be detected;
- Disrupt or neutralize attacks; and
- Confuse the attacker and introduce deception and unpredictability.

2. **DETECT/MONITOR** those attacks and abnormalities that can be discovered to reveal intrusions and gain situational awareness (SA) to inform our responsive operational strategies. While we cannot always detect advanced exploitations, we can improve our capabilities and continue to extend them on the basis of after-the-fact forensic analysis. Recognizing degradations, faults, intrusions, etc., or observing changes or compromises can become a trigger to invoke contingency procedures and strategies. A fundamental prerequisite is deliberate placement of sensors that can provide in-depth coverage across the environment to monitor critical processing and flows.

   Many forms of monitoring are needed across the layers of the architecture (e.g., event monitoring, traffic analysis, identification of user and system anomalous behavior, audit analysis, and forensics). Analyzing and correlating operational measurements of performance, capacity, response time, latency, and processor health metrics, in combination with monitoring of misuse, abuse, attacks, exfiltration, malicious code, and modifications, will improve SA. Subtle abnormal changes are important because we have no assurance that our protective measures can fully deter the cyber adversary.

   Because we cannot predict the adversary's changing tactics, techniques, and procedures (TTPs), we need dynamic sensor deployment models that support active sensor tuning and deployment adjustments in real time based on early warning alerts or an incident. Monitoring adversarial action, rather than shutting it down, can provide an opportunity to learn the attacker's TTPs. Collecting data during an attack for later forensic analysis can enlighten us about new or improved detection, protection, or deterrent capabilities to develop for the future. The challenge is to do so undetected while still containing the damage as the attack is happening.

3. **CONSTRAIN/ISOLATE** interfaces and functional capabilities. System developers must apply design approaches that separate functions, data, and network segments in order to isolate critical assets and

problems, minimize damage and impact propagation, and allow better forensics and analysis. Isolation should ensure that some portions of the system continue to function even if others do not. Some examples are:

- Separate critical from non-critical processing and data
- Isolate an intranet from an extranet from the Internet
- Partition processing, access points, data, and network segments
- Separate inbound from outbound traffic
- Separate requests from responses
- Isolate faults
- Constrain propagation when attacks succeed

The concept of separation is not new. It constitutes the basis of a security kernel used to separate the security policy enforcement portion of the operating system (OS) from the rest of the OS. In an extreme case, this can be equated to a standalone system in a physically protected data center or isolated computing enclaves for executing highly sensitive work. For the power grid, isolation can take the form of *islanding*, a term used to denote a situation where the distributed generation generator continues to power a location even though the electric utility is not functioning. In computer networks, separation may equate to segmentation or to a hardware switch that delivers just-in-time connectivity of limited duration when needed.

4. **MAINTAIN AND RECOVER** operations for minimal essential capabilities. Maintaining critical operations means first distinguishing essential from non-essential capabilities, understanding dependencies among components, and performing contingency planning activities. Planning should address possible degradation in capacity and performance, denial of service, and corruption of data, hardware, and software processing. Good planning also calls for building fine-grained, adaptive manageability and configurability into system designs and supporting alternative operational capabilities and/or functionality for times when normal critical processing capabilities are under attack.

   Highly modular architectures, boundary devices, and administrative and management interfaces form the underpinnings for dynamic contingency operations. For contingency planning, administrative and network operational capabilities must support rapid and automated replication, scaling, failover, reconfiguration, recovery, reconstitution, replacement, relocation, and initiation of critical services or alternative services in a distributed environment. When attacks succeed despite COOP and contingency operations, rapid, dynamic discovery and composition of alternative services/capabilities that are interoperable with existing capabilities may be needed, in addition to the ability to return to a trusted state within a reasonable timeframe.

5. **ADAPT** continuously in response to escalating attacks and changing threats or risk posture, and as a proactive step to foil exploits. By introducing technical, defensive, and operational change management into the system, system designs can potentially foil an attacker's exploit and/or confuse the adversary by adding an element of surprise, uncertainty, or unpredictability. For example, continuous change through randomization, adaptive computing, self-healing, moving critical operation, and other adaptive techniques, or simply leveraging emerging technologies or new computing paradigms as an early adopter or in novel ways, can provide covertness, unpredictability, and resilience for a period of time.

While autonomic adaptive responses have the advantage of reacting at computing speeds, sophisticated attackers can also use them against us. Ultimately it is the operator's ability to understand the mission and situation and determine the best response in the heat of a crisis that will save the day. There is no substitute for trained, experienced operators who can readily adapt and respond to an unexpected situation.

Characteristics

To remain resilient while under cyber attack, our architectures must be evolved or at times radically redesigned to exhibit many of the functional and technical characteristics and properties summarized in the table below.

Objective	Resilient Architectural Characteristics and Properties
Protect/Deter	Implements security best practices
	Minimizes the loss or corruption of services or data
	Tolerates some failure, faults, intrusions, degradation, and loss
	Minimizes and simplifies the system's minimal essential functions to enable successful operations and management of the resulting system in a crisis situation
	Supports offensive abilities to react and, in some cases, fight back in a contested cyberspace
	Possesses hardware, software, services and data replication, redundancy, and diversity
	Provides assurance mechanisms for correctness and integrity of software and hardware functions for essential functions
Detect/Monitor	Detects anomalies, the symptoms of failures and/or attacks, and signs of problems in neighbors
	Monitors its operating condition; possesses self-awareness of state of health, performance, availability.
	Collects information for later forensic analysis.
Constrain/Isolate	Enables configurability, continuity of operations (COOP), and disaster recovery (DR) to support rapid, predictable reconfiguration or restoration of capability
	Integrates safeguards (e.g., segmentation and stops) to contain the spread of damage and propagation of failures
Maintain/Recover	Degrades gracefully, when necessary
	Fails in a known good way, when necessary
	Returns to its nominal operating condition as quickly as circumstances permit

Adapt	Operates adaptively during normal operations and in response to changes in external and internal situations • Sometimes in a predictable way and sometimes more randomly for the purpose of unpredictability for an adversary. • Is self-learning, agile, adaptive, reconfigurable and extensible.
	Leverages hardware, software, data, and processing diversity and distribution in a random way
	Returns relatively quickly to its level of trust prior to the anomaly or to an acceptable level of trust
	Adapts to introduce randomness, deception and unpredictability to confuse the adversary

Getting Started

Apply a Risk Management Approach

A one-size-fits-all approach to designing resilient architectures is neither practical nor appropriate. We should choose a carefully balanced combination of protection mechanisms, detection capabilities, and adaptive technical and operational responses based on mission or business needs, processing environment, risk tolerance level, and critical operational scenarios.

Achieving this balance depends on first applying a risk analysis methodology to determine which critical capabilities must be resilient, to what level, and against which threats. This risk analysis forms the basis for deciding the adequacy of existing mechanisms and procedures, identifying any gaps, and determining the trade space of alternative technology approaches and courses of action. Risk analyses and dependency modeling can help identify the best locations to place additional safeguards to monitor and limit attack propagation, and drive the design of failover partitions and snapshots to recover to a secure state. The risk management process should:

- Identify mission- or business-critical capabilities, use cases, and assets (aka crown jewels);
- Map crown jewel dependencies to one another and to the underlying IT infrastructure, people, and processes;
- Weight relative priority in terms of criticality and minimal essential capabilities;
- Analyze the current or planned architecture's susceptibility to known and probable attacker TTPs; and
- Evaluate alternative mitigation strategies at the nexus of security protections, business continuity disciplines, and network and computer network defense (CND) operations.

Once we understand the risk posture and constraints of our critical processing and operational environments and the trade space and criteria for making risk decisions, we can begin to evaluate a variety of game-changing technologies, strategies, and techniques for improving resilience.

Virtualize the Infrastructure for Agility

Using virtualization technologies to build in the agility needed to change mission systems easily can facilitate cost-effective adoption and greater impact of the approaches described above. Virtualization is a recent computing paradigm shift. Businesses are rapidly adopting and deploying virtualization to consolidate data centers for efficiency and to reduce costs (space, resources and power consumption savings), as well as to provide cost-effective redundancy and improved provisioning, recovery, and security. This single disruptive technology supports all the objectives for resilience and therefore serves as a keystone to building secure, resilient architectures.

Specifically, virtualization can be used to implement techniques for isolation, non-persistence, and replication and scaling for availability. Virtual machines (VMs) offer a cost-effective approach to diversity and randomness because they can incorporate diversity in hardware platforms, chip sets, operating systems, applications and services, and randomness in deployment practices. These features make virtualization a fundamental enabler of resilience. VMs can be created, replicated, reconstituted, and deployed in milliseconds, thereby providing scalability, manageability, and agility to create, deploy, and move critical processing capability at will if the system is under attack. Many of the approaches presented in this paper become more effective in a virtualized environment.

Cloud computing represents the most recent instance of this paradigm. It is a model for enabling self-service, on-demand access to shared computing resources (e.g., networks, servers, storage, applications, and services) in an Internet-based environment. Shared, managed services can be rapidly provisioned, deployed, and scaled with minimal service provider interaction. In this paper we do not distinguish virtualization techniques from cloud computing. The diversity and distributedness of cloud computing services raise virtualization and deployment to a higher level, but also introduce significant security, governance, and control issues that must be addressed.

Virtualization is already an important part of high-availability strategies, but has not yet been generally applied to promote resilience. To date, disaster recovery (DR) and high availability (HA) are achieved primarily through redundancy, capacity planning, and backup and restoration to achieve COOP. Virtualization offers the key benefit of reducing the time and cost of recovering from a backup. An entire VM can be backed up and restored in less time than it takes to save files to backups, reinstall the operating system, and restore data. Of course, operators must plan appropriately for such uses of virtualization to ensure the availability of the resources it will consume (e.g., memory, CPU cycles, and disk space).

Administrators can use virtualization to create a master image of how to configure servers in a given data center. This makes it easy to create cookie-cutter systems and to recover if a server is compromised, since the administrator can quickly reset the system to the original or different template. VMs can be reinstalled on different physical servers when portions of a system are inaccessible, corrupted, or under attack or when the nature of an ongoing attack warrants a different or more stringent security environment.

When a denial of service attack is in progress, we can make critical processing resilient by replicating, reconstituting, or deploying additional VMs to increase processing capacity. These VMs can be run on the same hardware, on different hardware that is known not to be under attack, or simply on a platform that supports alternative technologies, stricter security constraints, or is managed by a different service provider.

At the same time as we pursue applications of virtualization for resilience, we must address the security of the hypervisor and VMs to ensure that they cannot be easily compromised and that viruses and exploits cannot hop from one VM to another faster than they can move on physical machines. In addition, operators must evaluate the potential transient and movable nature underlying the range of deployment options to prevent introduction of new vulnerabilities or attack opportunities and to adapt scanning and monitoring for both on-line and off-line platforms and VMs.

Summary/Conclusion

To reverse the asymmetric advantage of the cyber attacker and minimize the impact on our critical mission capabilities, we must be proactive in building secure and resilient systems. By promoting resilience against escalating cyber attacks, we can simultaneously achieve resilience against acts of nature, loss of physical network elements, and other threats. While it is not realistic to assume we can stop all cyber attacks or make them totally ineffective, redesigning architectures for resilience will make attacks less likely to succeed, will minimize the consequences when they do succeed, will increase adversary cost and uncertainty, and may act as a deterrent against future attacks. Improving resilience will also increase system reliability.

Game-changing technologies, design techniques, and operational strategies are available today to get us started. This paper presented possible approaches to improving resilience such as:

- Diversity
- Redundancy
- Integrity
- Isolation/segmentation/containment
- Detection/monitoring
- Least privilege
- Non-persistence
- Distributedness and moving target defense
- Adaptive management and response
- Randomness and unpredictability
- Deception

Given operational constraints and lifecycles, not every technique applies in all environments. At a minimum, however, designers should consider these ideas when developing new systems. Legacy systems will pose a greater challenge and system designers will need to evaluate which techniques have the greatest potential value and how best to introduce them. To begin building resilience into systems, we must decide which of the described promising strategies and techniques are most appropriate for our environments and critical missions.

The next step is to experiment with these techniques and strategies in laboratories and pilots to (1) demonstrate the feasibility and effectiveness of different approaches in different environments and operational scenarios and (2) identify the usability, cost, performance, and other operational considerations that we must assess. During these evaluations, we can start to develop resilience metrics and to measure the tangible benefits, readiness, and residual issues we must address in order to proceed with deploying these techniques and strategies.

Doing nothing is not an option. We must act now to reverse the adversary's advantage and ensure that we can rely on our mission-critical capabilities to be available and trustworthy when we need them most.

