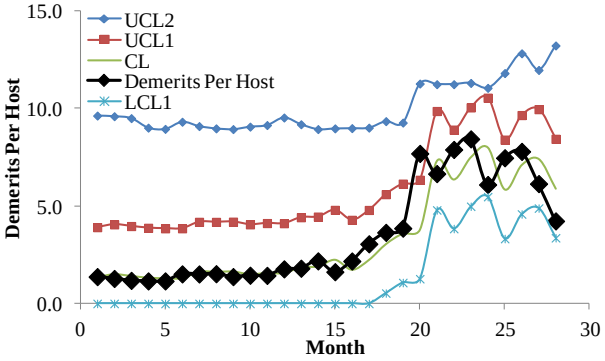
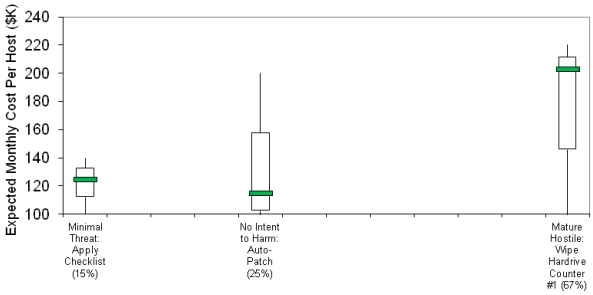
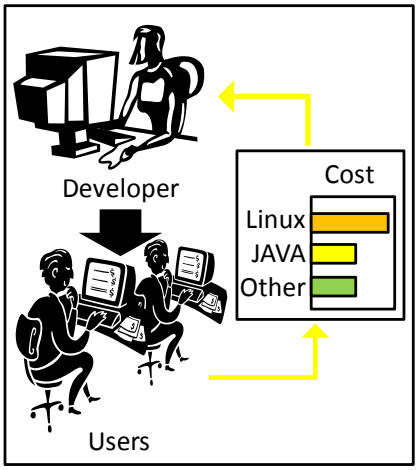
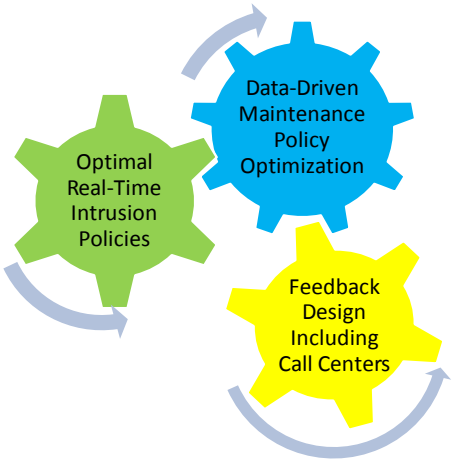
Sustainable Cyber Systems: Managing Life Cycles and Real-Time Risk

White Paper
September 2014









Contact Information:

Theodore T. Allen, Ph.D.
The Ohio State University
1971 Neil Avenue – 210 Baker Systems
Columbus, Ohio 43221
(614) 292-1793 (office)

www.blying.com

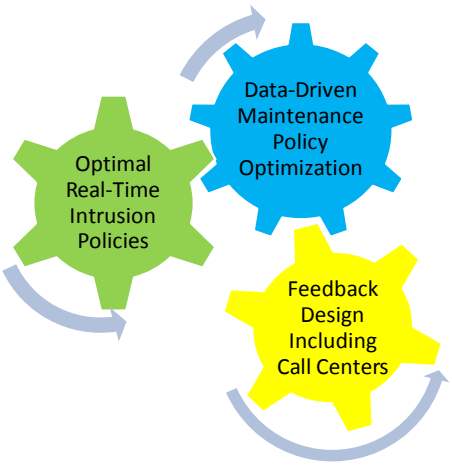


Figure 1. Interacting concentration areas.

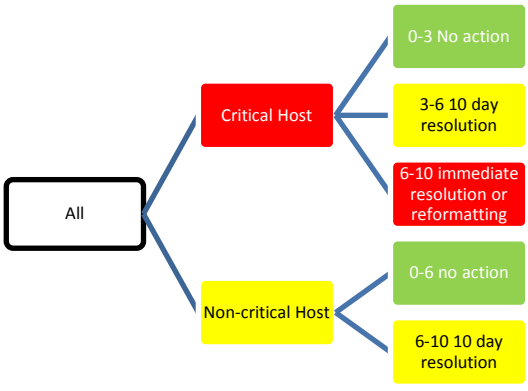


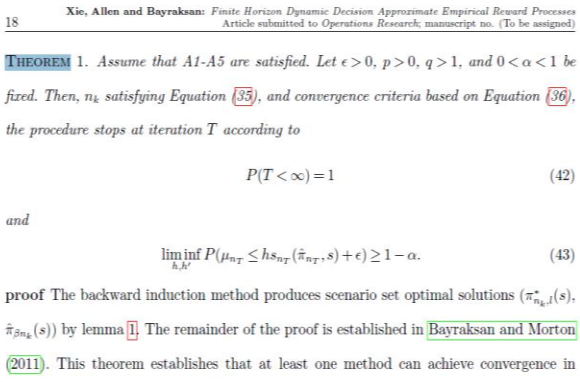
Figure 2. Example vulnerability policy.

1. Overview

Cyber security and the jobs of chief information security officers are complicated, providing many areas of interests and related data sources. In our work, we have focused on three interacting areas (Figure 1): The first area, cyber vulnerability maintenance and related policy optimization, offers easily accessible data with a relatively long time-scale (days or months) for policy optimization. Related issues include password policy optimization. The second area deals with network security. It aims to provide real-time decision-making and offers the possibility of developing cost-optimal rules that integrate other data sources including vulnerability scan data. A third area deals with feedback to software developers and implementers. Feedback from all the data sources can be integrated and presented so that, e.g., new project might use C# instead of JAVA to avoid costs over the long term.

2. Data-Driven Maintenance Policy Optimization

Organizations often develop control policies and documentation with an eye toward what other similar organizations are doing as well as to related cost data. Such costs include direct costs to pay for lost identities, administrator hiring costs, call center hiring costs, and disruption costs. Modeling the cost and benefits of alternative policies requires a sophisticated understanding of operation research. Our research group has developed data-driven formulations designed to help tune up existing policies to reduce costs and improve benefits, while taking into account data about past implementations. An example of a policy that could derive from our methods is shown in Figure 2. Decision variables include cutoffs and action options.



18 Xie, Allen and Bayraksan: *Finite Horizon Dynamic Decision Approximate Empirical Reward Processes*
Article submitted to *Operations Research*; manuscript no. (To be assigned)

THEOREM 1. *Assume that A1-A5 are satisfied. Let* $\epsilon > 0$, $p > 0$, $q > 1$, *and* $0 < \alpha < 1$ *be fixed. Then,* n_k *satisfying Equation (35), and convergence criteria based on Equation (36), the procedure stops at iteration* T *according to*

$$P(T < \infty) = 1 \tag{42}$$

and

$$\liminf_{k,k'} P(\mu_{n_T} \le hs_{n_T}(\hat{\pi}_{n_T}, s) + \epsilon) \ge 1 - \alpha. \tag{43}$$

proof The backward induction method produces scenario set optimal solutions $(\pi^*_{n_k,l}(s), \hat{\pi}_{\mathcal{S}n_k}(s))$ by lemma 1. The remainder of the proof is established in Bayraksan and Morton (2011). This theorem establishes that at least one method can achieve convergence in

Figure 3. Rigorous guarantee of solution quality.

Solving policy optimization problems can be challenging, even with modern computer power. Our algorithms offer efficient solutions with guaranteed quality. Proven claims include the theorem in Figure 3.

3. Parametric Uncertainty

The need to keep vulnerabilities and losses private, at least in part, and the relative rarity of intrusion experiences put limits on data available for analysis of selected policy options. To address the related "parametric" uncertainty, we have developed innovate formulations, algorithms, and visualizations.

The sufficiency model action clarification (SMAC) plot in Figure 4 is based on real-world data. Policy option 1 means using only automatic patching. Option 2 is manual intervention using a checklist approach. Option 3 is blocking the associated host from internet access. The SMAC plot in Figure 4 indicates that the most likely optimal policy is 11133 which means blocking critical and compromised hosts but performing only automatic patching on the others.

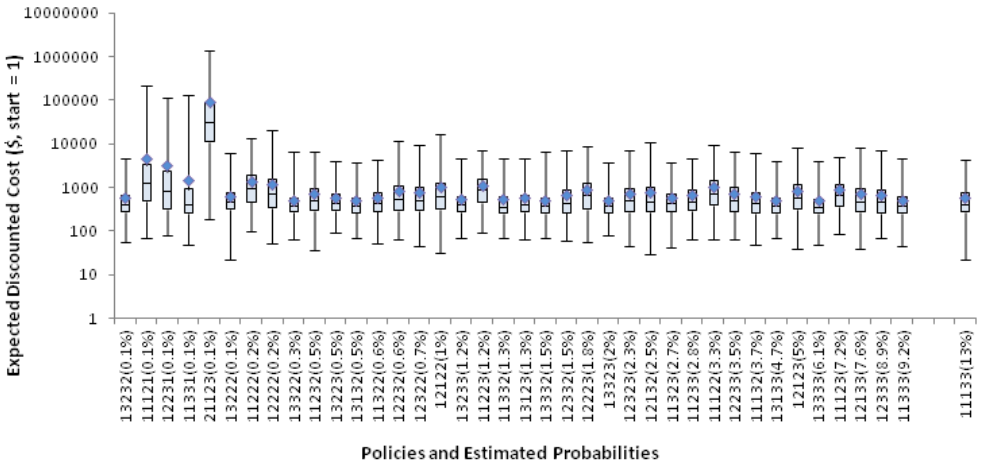


Figure 4. SMAC plot using real data for critical hosts under a single cost structure.

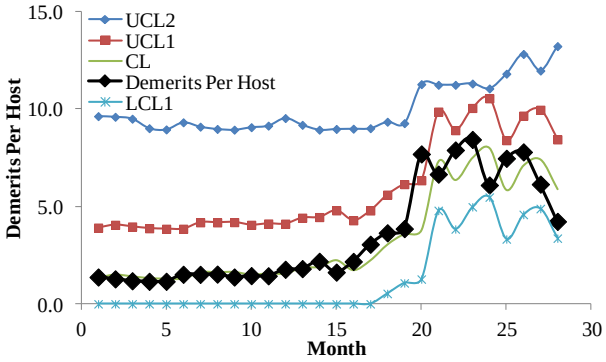


Figure 5. MCRAD control chart for monitoring.

4. Optimal Real-Time Intrusion Policies

Many researchers around the world have developed statistical and machine learning methods for detecting intrusions and automatic control. Relatively little research has focused on transparent and "directable" methods that facilitate administrator decision-making. One exception is our moving centerline residual-based and adjusted demerit (MCRAD) charts.

One MCRAD chart based on real (vulnerability) data is shown in Figure 5. The intent is to alert the human decision-maker that something unusual and possibly assignable or fixable is occurring. By combining MCRAD concepts and integrative metrics which synthesize real-time, vulnerability, and other sources of information, improved real-time control methods are being developed.

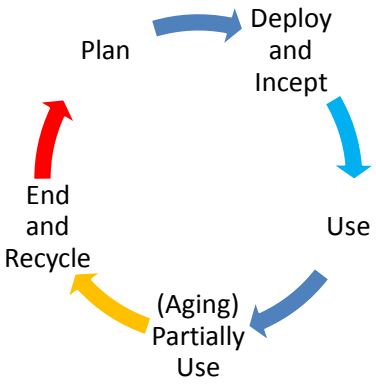


Figure 6. Software life cycle.

5. Sustainable Cyber Systems

The relationship between sustainability for physical and cyber systems has received little attention. Yet, when a developer picks a certain language or platform for development, there are long term implications for cost that are largely unknown.

Figure 6 indicates a life cycle for a specific software program in cyberspace. Note that there are organizational costs in all phases. The cost measures that we have used in vulnerability policy development attempt to address total life cycle cost but they ignore issues such as deployment, aging, and recycling. We are currently investigating more realistic formulations that include all stages in the life cycle.

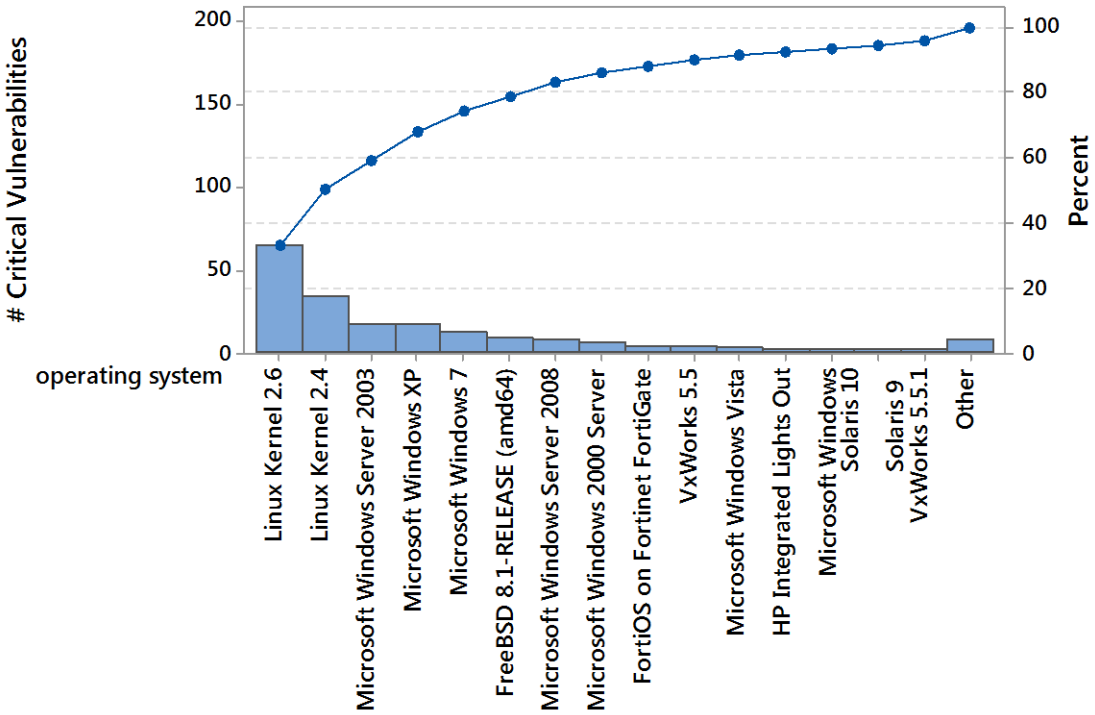


Figure 7. Pareto chart of critical vulnerabilities on a network.

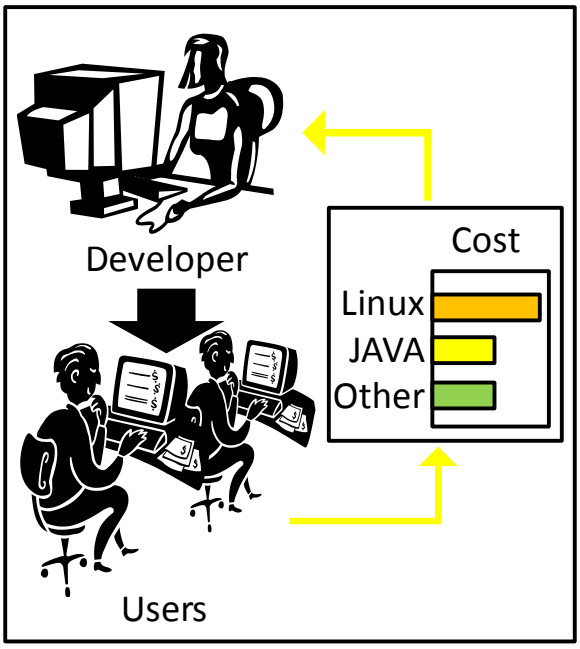


Figure 8. Life cycle analysis feedback loop.

In the physical world, there is growing consciousness that activities that we do such as eating beef and flying airplanes have surprisingly high costs in terms of tons of carbon emitted and water usage. Are there comparable activities in the software development world that designers should be aware of?

Our preliminary analyses of vulnerability data in Figure 7 suggest that aging Linux systems and reused JAVA code have surprisingly high organizational costs. Currently, we are investigating standardized cost metrics and visualizations that seek to increase awareness and feedback results to design communities and the public. Figure 8 sketches the possible form and function of such feedback.