Decision-Theoretic Approach to Designing Cyber Resilient Systems

Vineet Mehta, MITRE Corporation
Paul D. Rowe, MITRE Corporation
Gene Lewis, Stanford University
Ashe Magalhaes, Stanford University
Mykel Kochenderfer, Stanford University

***Abstract*—The increasing number of persistent attacks on computing systems has inspired considerable research in cyber resilience solutions. Resilient system designers seek objective approaches to aid in the comparison and selection of effective solutions. Decision theoretic techniques such as Markov decision processes can be leveraged for such comparisons and design decisions. Markov decision processes facilitate examination of uncertainty in system dynamics, diversity of responses, and optimization for operational objectives. This paper proposes a system design approach based in decision theory to achieve effective cyber resilience solutions. The prototypical example of a system with network intrusion detection and host reconstitution is used to illustrate this approach and highlight difficulties designers face due to the non-trivial coupling that may arise between response mechanisms.**

I. Introduction

Recently, the field of cyber resilience has received growing attention as the cyber defense community has pivoted from the pursuit of absolute protection to the exploration of methods for damage mitigation and system recovery. Cyber resilience has been defined as "the ability of cyber systems and cyber-dependent missions to anticipate, continue to operate correctly in the face of, recover from, and evolve to better adapt to advanced cyber threats" [1]; this is a paradigm shift away from the traditional focus of preventing intruders from entering a system. Recent work in cyber resilience has subsequently focused on the study of reconstitution, or actions that allow for the recovery of infected systems back to a normal operating state. Cyber resilient design attempts to blend traditional security measures with proactive strategies and adaptive components to create a system that can both defend and recover from malicious attacks conducted by an adversary.

The design of cyber-resilient systems is a non-trivial task. Even simple system architectures that operate under strong assumptions about adversary behavior are prone to volatile and unpredictable system dynamics due to inherent uncertainty in system inputs and the effects of component interactions. This difficulty is further exacerbated by the number and variety of defensive and resilience solutions available, coupled with diverse organizational objectives, and shifting adversarial tactics. Systematic analysis under a range of operational conditions is necessary for understanding the performance sensitivity of candidate system designs; to perform such an analysis, a flexible framework is needed within which system designs can be quantitatively modeled, evaluated and compared. Such a framework would ease the designer's task of identifying cyber resilient architecture options.

We contribute to the development of this framework by proposing a decision-theoretic modeling paradigm. Specifically, we illustrate the advantages of a Markov Decision Process (MDP) formulation [2], [3] for comparing cyber resilience of a variety of system designs and configurations. Through the use of an illustrative example, we demonstrate how an MDP system model can be used to account for the variety of mission objectives, input uncertainties, and design options in a unified manner. Each MDP formulation entails an optimal policy (a mapping from the system state to a recommended action). We study the sensitivity of the utility of these policies to changes in the adversary-controlled parameters of a model. Our illustrative example suggests how more complex systems can be analyzed through the composition of component models. The facility for composition allows for a unified examination of design options from diverse resilience techniques.

Previous work in the formation of mathematical frameworks of cyber resilience [4], [5] has focused on systems modeled as interdependent service components connected over a graph. Broadly, these approaches represent the system health as a collection of component states, with the system transitioning between states in response to both external influences and actions taken on components. Though these works are important steps in model development, their focus has been on recommending the best actions that return a system to normalcy during operation. There appears to be an absence of frameworks which address system designs with a mixed set of cyber resiliency goals (anticipate, withstand, recover, and evolve) in a unified manner, with the goal of supporting both feature selection during system design and sensible action recommendation during system operation. Our approach suggests a unifying framework within which features contributing to different resilience goals can be compared against each other; a system designer can therefore decide, for example, whether it is better to implement a feature that makes it easier to *withstand* an attack or a feature that reduces the time to *recover* from an attack. Furthermore, the system operator is provided with guidance on how to best employ the resilience features selected by the system designer.

Approved for Public Release; Distribution Unlimited. Case Number 16-2931. This technical data was produced for the U. S. Government under Contract No. FA8702-16-C-0001, and is subject to the Rights in Technical Data-Noncommercial Items Clause DFARS 252.227-7013 (JUN 2013).

978-1-5090-3216-7/16/$31.00 ©2016 IEEE

Concretely, the main goal of this paper is to illustrate the use of decision theory for a quantitative analysis of resilience as it relates to the security of computing systems. In support of this goal we provide:

- A formulation of systems analysis for cyber resilience within the context of a decision-theoretic framework.
- An application of this formulation to an example system with a specific set of resilience response mechanisms: network intrusion detection and host reconstitution.
- An illustration of design options analysis using this framework in comparing the resilience afforded by families of systems.

The paper is structured as follows. Section II describes previous work in support of cyber resilience. Section III describes the use of a POMDP framework for examining resilience for security of computing systems. Section IV introduces the formulation of an example system as an MDP model. Section V describes experiments aimed at comparing the resilience of different system configurations. Section VI compares results obtained from the calculation of optimal policies for a variety of system configurations, with different resilience features, and adversarial activity. The results illustrate how a designer can establish which resilience features are most effective. Section VII closes the paper with a discussion.

II. Related Work

Previous work in support of cyber resilience has focused on the development of a set of goals and a framework within which these goals may be pursued; examples include the Cyber Resiliency Engineering Framework (CREF) proposed by Bodeau et. al [1], [6] and the CERT Resilience Management Model [7]. Under these frameworks, generic cyber resilience goals are defined by a cyber system's ability to anticipate adversarial attacks, continue operation under distress, recover operation after sustaining an attack, and utilize previous encounters to better prepare for future attacks.

A. Mathematical Formulations of Cyber Resilience

Work in developing a mathematical formulation of modern cyber resilience can be traced to Ramuhalli *et. al* [4], who proposed a graph-based cyber resilience model for the study of reconstitution actions. They model their system as a connected network with nodes that host a set of dependent services; the system maintains a state related to its health, with the ability to transition between sets of fully operational, marginally operational, and compromised states in response to natural causes or the actions of an adversary. Reconstitution is treated as an optimization problem over the resulting graph, balancing the value of continued operation against the costs of reconstitution; resilience is achieved through complete reconstitution of lost services.

Choudhary *et. al* [5] expand this graph-based framework by modeling systems of interest as a collection of interdependent system components. Here each system component is modeled as a subgraph and assumed to provide a distinct service while being susceptible to a particular type of attack; the system as a whole is then modeled as the total set of connected sub-graphs. Resilience is achieved by selecting a subset of pre-defined mitigation actions. Choudhary *et. al* also provide an action recommendation engine that takes as input the current system state and provides as output an action recommendation. The recommendation engine compiles sufficient statistics related to performance and security measures of interest, and then solves an optimization problem to provide a sensible advisory. In contrast, our approach provides a recommendation engine in the form of a policy; advantages include the ability to examine policies without the need for a running system during design-time, and fast real-time action lookup for a given system state when in deployment.

B. Decision-Theoretic Approaches to Cyber Security

One well-explored approach to handling the inherent uncertainty in adversarial cyber attacks is to adopt a learning-based decision-theoretic model. These models are capable of leveraging data to derive optimal actions based on the state of the system. To the best of our knowledge, such models have not been employed as generalized frameworks in the design or orchestration of resilience for secure systems. Decision-theoretic approaches have been applied extensively to the adaptation of detection models in intrusion detection systems (IDS); examples include a multi-agent reinforcement learning approach considered by Servin and Kudenko [8], where agents cooperate to detect intrusions, and a partially observable Markov decision process (POMDP) formulation provided by Lane [9], where incoming partially labeled data is assumed to be generated by a normal user or an attacker. They have also been applied in the selection of suitable policies for guiding the operation of intrusion detection and response systems, as demonstrated in the analysis provided by Kriedl [10].

The mention of previous IDS related efforts here is motivated only by their connection to our own illustrative example of applying MDP models for design and operation of resilient secure systems. Our main aim, however, is to demonstrate the general utility of such models in providing a unified framework for design of resilient systems, and their subsequent operation. Since our illustrative example considers IDS models, we provide some details on the IDS attributes highlighted in our example. Despite the popularity of IDS systems, the inaccuracy of detections and large volume of false alarms (in some cases nearly 70% of all alerts) can limit effectiveness [11]. IDS systems also have difficulty keeping up with incoming network traffic which exceeds rates of a few megabits per second. Studies with Snort, an open source intrusion detection system, have reported dropped packets proportions of more than 70% [12] due to network packet throughput limitations.

In order to address this limitation, Bakhoum [13] disproved the common notion that a strong IDS must inspect every packet, showing instead that network security can be maintained while inspecting only a subset of incoming packets. However, such a system is still vulnerable to allowing hostile packets past the IDS and into the host, where an infection can

cause critical computing components to malfunction indefinitely until taken offline for maintenance. This result suggests that a controller taking actions over both the IDS and the system host is needed to achieve the cyber resilience goals of operation under distress and recovery after a sustained attack.

In order to highlight the use of decision theory for selection of features during design, we include a host capable of reconstitution as a separate component, connected to the IDS by a communications link. This allows us to illustrate the modular composition of a system from components and to examine non-intuitive responses that can arise from component interactions under a range of adversarial scenarios. We consider how different design choices augment the space of actions available for an operational system, and how the addition of such actions affects overall cyber resilience. We also illustrate the flexibility of decision theory based design for assessing resilience of alternate operational objectives through appropriate constructions of reward structures. This ability to assess alternate designs in the context of operational objectives allows for discriminating the effectiveness of resilience features. For instance, an IDS, which is often viewed as a valuable feature, might prove to be of little value for certain designs and operational scenarios.

III. MDP Framework for Cyber Resilience

When we talk of the resilience of a system we refer to the stability of that system's performance under shocks of various types. The performance is measured by some objective function f, and the shocks are represented as changes in the different inputs to f. There is no single right choice of objective function to measure a system's performance; in fact, there may be many such performance measures of interest for a given system. The system may similarly be more or less stable as different inputs vary, so we might be more interested in resilience to changes of particular inputs. In cyber resilience we are primarily interested in the stability of an objective function f under changes to inputs that are under the control or influence of a cyber adversary.

A. Decisions in Cyber Resilient System Design

An important question in system design is whether adding a given feature to a system will make it more resilient, and if so by how much. Similarly, a system designer may be constrained to add only one of two extra features, and so would like to know which one has a greater effect on the system's resilience. It is common for such features to provide system administrators with the ability to adaptively respond to adversary activity. An important aspect in increasing the resilience is to choose good policies for response. This suggests a connection between cyber resilience and decision theory.

To demonstrate this connection consider a standalone system that is connected to an external network on which an adversary may reside. The adversary can inject malicious messages into the stream of incoming messages; these malicious messages have some chance of infecting the system. The system incurs some cost for being infected. Common methods for making the system more resilient to incoming malicious messages include (a) employing an intrusion detection system (IDS) to filter out messages matching known malware signatures and (b) periodically resetting or reconstituting the system to a known good state. Both of these resilience mechanisms may introduce a new cost of their own. If message inspection cannot keep up with the rate of incoming messages, dropped packets will reduce the performance of the system. Similarly, resetting the system might entail taking it offline for some time, which itself may carry some cost.

These feature interactions increase the complexity of the system; by offering more actions to take in a given state (e.g. inspect a message vs. let it pass through uninspected), the variety of system behaviors depending on the policy chosen is correspondingly higher. Given the additional cost of some of the new actions, there is a chance that there are new policies available to deploy that perform worse than the original system. This suggests that techniques from decision theory can bring value to the problem of cyber resilience.

B. Markov Decision Process Formulation

A popular model used in decision theory is a (fully observable) Markov Decision Process (MDP) or its partially observable variant (POMDP) [2], [3]. In general a POMDP is described by the tuple $F_\Omega = (S, A, P_S, R, Z, P_Z, \Omega)$ where:

Parameter	
S	Set of states
A	Set of actions
P_S	Probability distribution over resulting states
R	Rewards associated with states and actions
Z	Set of observations
P_Z	Probability distribution over resulting observations
Ω	Set of system parameters

A POMDP model of the system described above can provide us with parameterized, quantitative dynamics. The set Ω can encode probabilistic parameters such as false positive and negative rates of the IDS, or the likelihood that a malicious message will infect the system. We can therefore investigate and compare the system's performance under changes to the underlying parameters, and consider variations associated with uncertainty of these parameters. Likewise, we can compare the system behavior with and without the two resilience mechanisms we described. Since such models admit solutions (i.e. policies that optimize the utility function) it is sensible to use the expected utility of the system under this optimal policy as a performance measure. That is, $\bar{V}_* = \mathbb{E}[V_*(s)]$ acts as a scalar metric for characterizing the performance of a system modeled by F_Ω. In this way the resilience of the system roughly corresponds to the stability of $\bar{V}_*$ under changes to the underlying parameters Ω.

IV. Exemplar Problem Formulation

In the sequel, we illustrate the practicality and power of our MDP cyber resilience framework via an examination of a prototypical system.

A. Model Overview

Our example model features an intrusion detection system protecting a host, as illustrated in Figure 1. The actions of the intrusion detection system and the host are guided by a controller. The action spaces over the IDS and host system is specified as $a_I \in A_I = \{\texttt{inspect}, \texttt{pass}\}$ and $a_H \in A_H = \{\texttt{wait}, \texttt{reset}\}$. Messages generated in the extranet

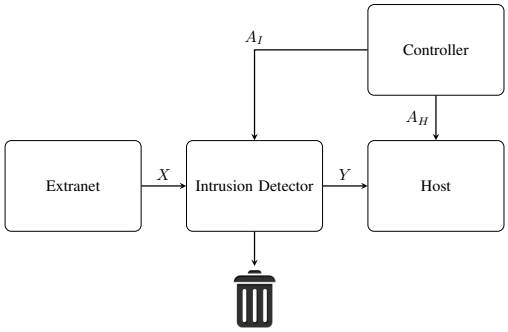

Fig. 1. Intrusion Detection and Response System

first pass through the intrusion detection system before arriving at the host system. A message's infection state is given by the random variable $x \in X = \{\texttt{benign}, \texttt{malicious}\}$. An incoming message is `malicious` with probability λ. The intrusion detection system can either pass the input message or drop it, guided by the controller actions $a_I \in A_I$. The random variable Y encodes the message output states as: $y \in Y = \{\texttt{benign}, \texttt{null}, \texttt{malicious}\}$. If a message successfully passes through the intrusion detection system it's outputs state y corresponds to that of the input x. Otherwise the output has state $y = \texttt{null}$. In the following we detail the internal operation of the intrusion detection system and the host, as influenced by the actions (a_I, a_H).

B. Intrusion Detection System Model

The state transition diagram in Figure 2 describes the operation of the intrusion detection system.

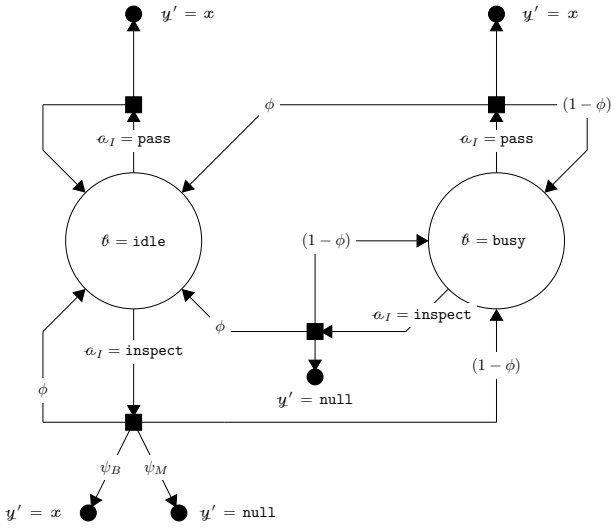

Fig. 2. Intrusion detector state transition diagram

The random variable B encodes this state as $b \in B = \{\texttt{busy}, \texttt{idle}\}$. The intrusion detection system is in the $b = \texttt{idle}$ state if it is not inspecting a message for malicious content, otherwise its state is $b = \texttt{busy}$. We employ the Markov assumption to model transitions between states. Therefore transition to a future state $B' = b'$ depends only on the current state $B = b$ and the action $A_I = a_I$. The intrusion detection system simply passes an input message without inspection if the current action is $a_I = \texttt{pass}$. Under this action, the output of the intrusion detection system at the next time step y' corresponds to the infection state of x. In this instance, the intrusion detection system returns to the `idle` state with certainty. If the intrusion detection system starts in the `busy` state, it transitions to the `idle` state with probability ϕ. The value of parameter ϕ governs the mean holding time in the `busy` state, and lower values of ϕ are interpreted as settings where the system performs deeper message inspection. When the system starts in the `idle` state, under the action $a_I = \texttt{inspect}$, it transitions to the `busy` state with probability $(1-\phi)$. In our model we make the simplifying assumption that the intrusion detection system provides an output which reflects the outcome of its inspection in the next time step, independent of the busy time induced by `inspect` action.

We model the intrusion detection system as an imperfect inspection system that can produces false positive and false negative results. The false positive and false negative probabilities for malicious message detection are specified by the parameters β_{FP} and β_{FN} respectively. When the intrusion detection system inspects a message, the message is classified as `malicious` with probability $\psi_M = \lambda(1-\beta_{FN}) + (1-\lambda)\beta_{FP}$ and `benign` with probability $\psi_B = \lambda\beta_{FP} + (1-\lambda)(1-\beta_{FN})$. When the intrusion detection system is in the `idle` state and it classifies a message as `malicious`, that message is always dropped and $y = \texttt{null}$ for the next time step. While the intrusion detection system is in the `busy` state, the action to `inspect` also results in that message being discarded and $y = \texttt{null}$. This behavior models computational limits typical of intrusion detection systems when attempting to keep pace under high traffic loads.

C. Host System Model

Figure 3 illustrates operational dynamics of the host, which are also modeled by a Markov Decision Process. The host system is described by two primary state variables that characterize the operational and infection states. The host has two operational states $w \in W = \{\texttt{full}, \texttt{reset}\}$. In the `full` state, the host is capable of fully processing incoming messages. The `reset` state provides an opportunity for the host to either avoid processing potentially malicious messages or allow for repairs if the host was previously infected. The state variable $h \in H = \{\texttt{clean}, \texttt{infected}\}$ specifies whether the host system is infected or not.

The state of the host system is influenced by the input message state y and action a_H. If the action state is `wait` and incoming message is not `malicious`, the system remains in the $(\texttt{full}, \texttt{clean})$ state. We allow the host to have some

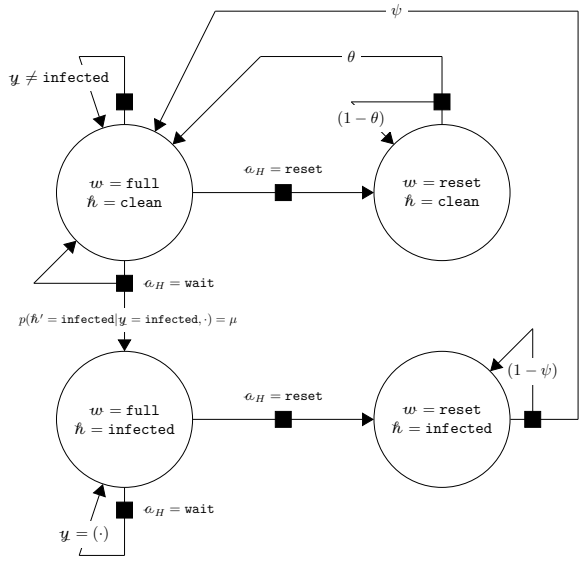

Fig. 3. Host state transition diagram

intrinsic resistance to `malicious` incoming messages. This ability is parameterized by μ: the probability of infection in the event of a `malicious` message. Incoming `malicious` messages cannot further infect a host, when the host state is (`full`, `infected`). The `reset` action forces a transition to $w =$ `reset` state. Our model allows different mean holding times in the `reset` state for the infection states `clean` and `infected`, through separate restoration probabilities θ and ψ.

D. Stochastic Model Dynamics

In order to examine the stochastic dynamics of our model as a whole we must consider the joint state of the system $S = (Y, X, W, H, B)$, under the influence of aggregate actions $A = \{A_I, A_H\}$. The transition probability from current state S to future state S' conditioned on action A is factored as:

$$P(S' \mid S, A) = P(Y' \mid X, B, A_I)P(X') \times P(W', H' \mid W, H, Y, A_H)P(B' \mid B, A_I)$$

The probability of a malicious incoming message is independent of other state variables, and given as a Bernoulli distribution with parameter λ:

$$P(X') = \text{Ber}(X' = x' \mid \lambda) = \lambda^{x'}(1-\lambda)^{1-x'}$$

Selected component probabilities are provided in Tables I and II. For brevity of presentation, we have only specified the non-zero probabilities in these tables. The probability function $P(Y' \mid X, B, A_I)$ is specified in Table I. This function models the output of our intrusion detection system at the next time step as dependent only on the input message state, the intrusion detection system busy state, and the controller action. The probability function $P(B'|B, A_I)$ is specified in Table II. This function models the intrusion detection system's next busy state B' as depending only on the current busy state and controller action.

TABLE I
INTRUSION DETECTOR OUTPUT TRANSITION PROBABILITY: $P(Y' \mid X, B, A_I)$

X	B	A_I	Y'	$P(Y' \mid X, B, A_I)$
`benign`	`idle`	`inspect`	`benign`	$1-\beta_{\text{FP}}$
`benign`	`idle`	`inspect`	`null`	β_{FP}
`malicious`	`idle`	`inspect`	`null`	$1-\beta_{\text{FN}}$
`malicious`	`idle`	`inspect`	`malicious`	β_{FN}
`benign`	`idle`	`pass`	`benign`	1
`malicious`	`idle`	`pass`	`malicious`	1
`benign`	`busy`	`pass`	`benign`	1
`malicious`	`busy`	`pass`	`malicious`	1
`benign, malicious`	`busy`	`inspect`	`malicious`	1

TABLE II
INTRUSION DETECTOR OPERATION TRANSITION PROBABILITY: $P(B' \mid B, A_I)$

B	A_I	B'	$P(B' \mid B, A_I)$
`idle`	`pass`	`idle`	1
`idle`	`inspect`	`busy`	$1-\phi$
`idle`	`inspect`	`idle`	ϕ
`busy`	`inspect, pass`	`busy`	$1-\phi$
`busy`	`inspect, pass`	`idle`	ϕ

The transitions of the aggregate host states are given by the probability function $P(W', H' \mid W, H, Y, A_H)$. This probability can be written in a fashion similar the other component probabilities. The form of this probability highlights that the host state is driven primarily by the message state Y and action A_H.

Our aim is to find optimal policies $\pi_*(s)$ for the MDP $(S, A, P, R \mid \Omega)$. We have specified the states S and actions A, as well as the transition probability P for the parameter tuple $\Omega = (\lambda, \mu, \phi, \theta, \psi, \beta_{\text{FP}}, \beta_{\text{FN}})$. The reward structure R remains to be specified for a complete description of the MDP. In this paper we will focus on the case where the rewards depend only on the aggregate state $S_H = (W, H, Y)$ of the host system. The rewards for the host states are specified in Table III.

TABLE III
REWARD STRUCTURE: $R^{(\alpha)} = R(S_H)$

W	H	Y	R
`full`	`clean`	`benign`	1.0
`full`	`clean`	`null`	$1.0 + \alpha\rho_+$
`full`	`clean`	`malicious`	$1.0 + \alpha\rho_-$
`full`	`infected`	`benign, null, malicious`	ρ
`reset`	`clean, infected`	`benign, null, malicious`	2ρ

In general the future-discounted expected return of applying a policy π when starting in state $S_0 = s$ is given by the value function:

$$V_\pi = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t)) | S_0 = s\right] \quad (1)$$

The discount factor γ is constrained such that $\gamma \in [0, 1]$. A solution to our MDP involves finding the optimal policy which

TABLE IV
FOUR SUBSYSTEMS

	A_H	
A_I	{wait}	{wait, reset}
{pass}	Σ_0	Σ_2
{pass, inspect}	Σ_1	Σ_3

maximizes the value function across all states. This objective is formally expressed as:

$$\pi_*(s) = \arg\max_a R(s,a) + \gamma \sum_{s'} P(s'|s,a)V(s') \quad (2)$$

A scalar metric for characterizing the resiliency of our system is the maximal expected utility $\bar{V}_* = \mathbb{E}[V_\pi | \pi = \pi_*]$. This expected value is calculated using the stationary state probability distribution $\varphi(s)$. The stationary distribution can be computed as the normalized left eigenvector for the unit eigenvalue of the transition probability matrix under optimal policy.

V. EXPERIMENTS

Our MDP model contains several sub-models that arise by restricting the action space $A = A_I \times A_H$. In particular, consider the four systems that arise by restricting the action spaces as described in Table IV. These systems are viewed as candidate design proposals with different resilience features.

Σ_0 represents a system without an IDS and with no ability to reset or reconstitute the system if it becomes infected. By adding the ability to inspect messages, Σ_1 represents a system with an IDS but with no ability to reset. Σ_2 contains no IDS, and so has no ability to inspect messages, but a reset action can restore it to a full and clean state. Finally, Σ_3 is the full system described in the previous section. Viewed as MDPs, these systems share the same underlying transition probabilities given a set of parameters Ω, but the restriction of the action spaces reduces the set of policies we can choose from when solving the MDP.

As discussed in Section III, we choose to measure the performance of these systems using the maximal expected utility under the optimal policy π_*, which we denote $\bar{V}_*$. Of course, $\bar{V}_*$ really depends on the underlying system (i.e. the set of available policies), on the parameter set $\Omega = (\lambda, \mu, \phi, \theta, \psi, \beta_{\text{FN}}, \beta_{\text{FP}})$, and on the discount factor γ used to discount the value of future rewards. We also might consider each system under several reward structures parameterized by ρ as described in the previous section. Thus, if we let Ω^+ denote the augmented set of parameters $\Omega \cup \{\gamma, \rho\}$ we may write $\bar{V}_*^i(\Omega^+)$ to denote the resulting maximal expected utility under the optimal policy for system Σ_i. Beyond comparing the performance (i.e. the values of $\bar{V}_*^i(\Omega^+)$) for the various systems Σ_i, we are also interested in the resilience of the systems. We want to know which systems are more sensitive to changes in the parameters Ω^+. We can thus examine the effect that each of the functional mechanisms (IDS and reset capability) has on both performance and resilience. Through this type of examination a designer can assess which features should be included in the objective design.

Ω^+ is a large domain, and some of the parameters are more likely to change over time than others. From the viewpoint of cyber resilience, we are most interested in detecting sensitivity to changes in parameters that are likely to be under the influence of an adversary. By adjusting the number of malicious packets sent to the system, the adversary has a strong influence on the λ parameter. Similarly, an adversary may have the ability to adjust the μ parameter governing the likelihood of compromise given a malicious message. For example, the adversary could send messages tailored to the particular system that are more likely to cause infections than generic attacks.

We therefore fix values for most of the parameters of Ω^+ while varying the values of λ and μ. The fixed parameters are $\phi = \theta = \psi = 0.9$ and $\beta_{\text{FP}} = \beta_{\text{FN}} = 0.01$. This assumes a system that is likely to complete packet inspection in a single time step, and is equally likely to recover from reset in a single time step (whether or not it was infected at the time of reset). The IDS is assumed to have very high accuracy with very low false positive and negative rates. We also set $\gamma = 0.95$ indicating that we only slightly discount the value of future states. Finally, we fix the parameter for the rewards to be $\rho = -1.5$.

Fixing these parameters means that $\bar{V}_*^i(\Omega^+)$ defines a surface over the unit square as λ and μ each vary over $[0, 1]$. We compute and plot various cross sections of this surface. In particular, we consider cross sections for $\mu \in \{0, 0.25, 0.5, 0.75\}$ letting λ range over $[0, 1]$ in increments of 0.01. The next section contains the results.

VI. RESULTS

In this section we present results for the experiments described is Section V. Our experiments cover a limited parameter space: $\lambda \in [0, 1]$ and $\mu \in [0, 1]$. The remaining parameters in the set Ω have values: $\phi = 0.9$, $\theta = 0.9$, $\psi = 0.9$, $\beta_{\text{FP}} = 0.01$, $\beta_{\text{FN}} = 0.01$. These parameters characterize a system that can quickly recover from `busy` and `reset` states. This system is also fairly accurate in its classification of inspected incoming messages. We will examine the resiliency of this system for two different reward structures: $R_H^{(0)}$ and $R_H^{(1)}$ for reward parameters $\rho_+ = -1.75$, $\rho_- = -2.75$, and $\rho = -1.5$.

A. Baseline System

We start by considering the system Σ_0 with the reward structure $R_H^{(0)}$. This system has a restricted action space $(A_I, A_H) = (\{\texttt{pass}\}, \{\texttt{wait}\})$. The reward structure $R_H^{(0)}$ favors the system's occupancy of state $(w, h) = (\texttt{full}, \texttt{clean})$. In the absence of any incoming malicious messages ($\lambda = 0$), the host system is always in this state, with expected utility $\bar{V}_* = 1/(1-\gamma)$. This value of expected utility serves as an upper bound on the performance the system can achieve. For non-zero probability of malicious incoming messages ($\lambda > 0$) the host system is always in state $(w, h) = (\texttt{full}, \texttt{infected})$,

with expected utility $\bar{V}_* = \rho/(1-\gamma)$. We note that the expected utility remains constant for values of $\lambda > 0$, because the reward structure favors the state $(w, h) = (\texttt{full}, \texttt{infected})$ irrespective of the value y. Due to the low utility value, this baseline system may be viewed as a poor resilience candidate.

B. Adding an IDS

System Σ_1 adds an intrusion detection capability whose objective is to intercept malicious messages before they reach the host system. However the host in this system lacks the ability to `reset` if it is infected. The presence of an intrusion detection system provides some protection by dropping malicious messages. However, some malicious messages do ultimately get past the intrusion detection system and successfully infect the host. Without the benefit of a `reset`, the host eventually enters the state $(w, h) = (\texttt{full}, \texttt{infected})$, which serves as an absorbing state for the system for $\lambda > 0$. Thus the expected utility for this system has the same behavior as that for system Σ_0, for reward structure $R_H^{(0)}$. This result appropriately suggests that little value is derived by adding an intrusion detection system, without a host based mitigation mechanism, based on long-term (stationary) behavior of the system.

C. Adding Reconstitutive Actions

Now we consider the systems Σ_2 and Σ_3, which include hosts that can `reset` in order to mitigate the effects of malicious messages. For the reward structure $R_H^{(0)}$, the intrusion detection system has been shown to add little value. We therefore expect benefit under this reward structure to be derived mainly from mitigation mechanisms at the host. Computed results of the expected utility under optimal policy are consistent with this intuition. The expected utility curves for system Σ_3 are shown in Figure 4 over a range of values for (λ, μ). The expected utility $\bar{V}_*$ reaches its maximum value

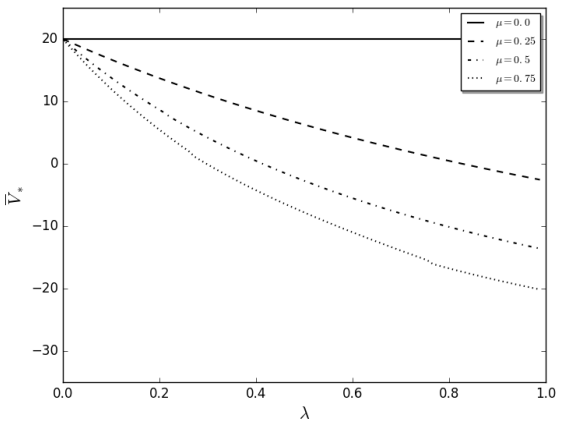

Fig. 4. Expected Utility for System Σ_3 with Reward Structure $R_H^{(0)}$

$(1/(1-\gamma) = 20)$ in the absence of incoming malicious messages ($\lambda = 0$), or when malicious messages have no impact on the host ($\mu = 0$). The expected utility diminishes quickly with increases in the probability of malicious messages (λ) and the probability of host infection (μ). The expected utility approaches its minimum value $(\rho/(1-\gamma) = -30)$ as $(\lambda, \mu) \to (1.0, 1.0)$.

With reward structure $R_H^{(0)}$ it is difficult to see the added benefit provided by an intrusion detection system, working in cooperation with host-based mitigation mechanisms. We can reveal this benefit by considering a modified reward structure $R_H^{(1)}$. In this modified structure we assign higher rewards for the absence of malicious passed messages, along with an uninfected and fully functioning host system (i.e. $y \in \{\texttt{benign}, \texttt{null}\}$ and $(w, h) = (\texttt{full}, \texttt{clean})$ respectively). The benefit of cooperative operation of an intrusion detection system with host mitigation mechanisms is highlighted by comparing the expected utility of systems Σ_2 and Σ_3 with the modified reward structure $R_H^{(1)}$. Results for system Σ_2 are shown in Figure 5. We note that this system lacks an intrusion

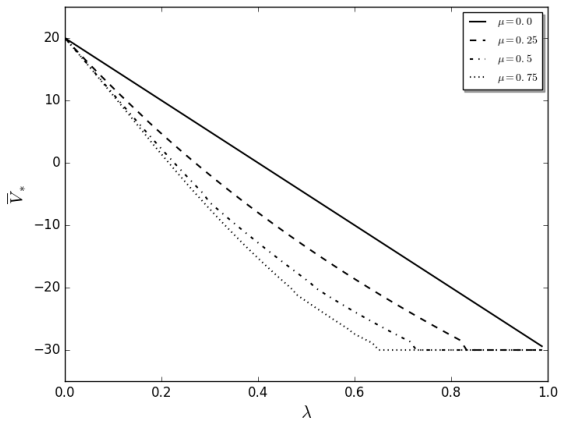

Fig. 5. Expected Utility for System Σ_2 with Reward Structure $R_H^{(1)}$

detection capability, but does possess host-based reconstitution. The observed monotonic reduction in this system's expected utility with increases in probability of malicious messages is consistent with the new reward structure, which favors the absence of such messages even for $\mu = 0$. A further decrease in utility is observed as the susceptibility of the host to `malicious` messages increases ($\mu > 0$). These results suggest the reset feature to be a valuable resilience design option for the system.

D. Combining Reconstitutive Actions with an IDS

The results for system Σ_3, which augments Σ_2 with an intrusion detection capability, are presented in Figure 6. The difference in resiliency offered by systems Σ_2 and Σ_3 can be understood by examining the plots in Figures 5 and 6 for corresponding values of μ. We recall that the intrusion detection system offers little benefit on its own. However, Figures 5 and 6 now reveal that the addition of intrusion detection capabilities provides an increase in expected utility. The cooperative operation of intrusion detection and host based mitigation capabilities is particularly evident for higher probabilities of incoming malicious messages (λ), and high values of host susceptibility ($\mu > 0.5$). It is interesting to

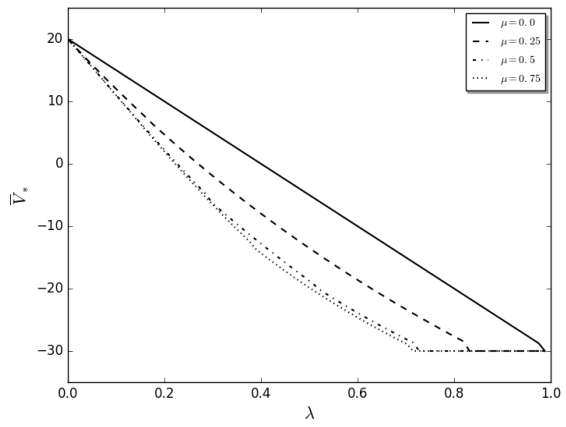

Fig. 6. Expected Utility for System Σ_3 with Reward Structure $R_H^{(1)}$

note that, for lower host susceptibilites, the host based `reset` mechanism offers the *best* mitigation to adversarial activity. As the host susceptibility increases, the intrusion detection system provides supplementary assistance against adversary attacks.

We note that the results shown in Figures 4-6 align well with our notion that resilience is connected to both the utility function's value and sensitivity (given by slope). As an example of this notion, note that we do not consider system Σ_0 resilient, despite the fact $\bar{V}_*$ is constant for all values of $\lambda > 0$; this is due to the utility over $\lambda > 0$ being at a minimum. These results highlight the value of our approach by revealing scenarios in which the inclusion of an IDS improves resilience.

VII. Conclusions

In this paper we have proposed decision theory as a suitably flexible framework for examining cyber resilience of computing systems. We have advocated the use of MDPs (or the more general variant POMDPs) for modeling a range of system designs, while incorporating uncertainty in system dynamics. MDPs also provide a flexible way of assigning value to desired system behaviors using rewards and a means of making optimal decisions that support resiliency goals.

In order to make our proposal concrete, we have chosen to analyze commonly used resiliency mechanisms: an intrusion detection system and host-based system reconstitution. The overall system has been modeled as a MDP. We have taken care however to treat the intrusion detection system and the host as separate components, with distinct state transition models. The coupling between the two components is via the messages that are passed from one to the other. Our model system has a number of tuning parameters, which facilitate the examination of a variety of behaviors and responses. However, we have chosen to focus on parameters that have a dominant connection with adversary behavior: the `malicious` message probability (λ) and the host infection probability (μ). Using this model we have illustrated how a comparative analysis of design alternatives can be formulated by examining different options for the action space and reward structure.

A quantitative comparative analysis of resiliency for a set of design alternatives has been performed for a common base system model that includes an IDS and host-based reconstitution capability. In this analysis we have selected reward structures that focus on the host's behavior, while allowing the coupling between the IDS and host to drive optimal actions for both components. The expected utility $\bar{V}_*$, derived from stationary probabilities based on optimal actions, has served as a scalar metric for understanding the resiliency of various design alternatives. We note that this particular metric is focused on assessing the defender's ability to weather adversarial activity.

In this paper we have examined resiliency from the perspective of the defender. For instance have not developed a metric that characterizes the amount of effort expended by an adversary to successfully impact system function. We have also assumed that state information about adversarial actions (`malicious` messages), and host infection state are known. The ability to jointly optimize actions of the IDS and host is also assumed. A natural extension of our work would seek to relax these assumptions.

References

[1] D. Bodeau, R. Graubart, W. Heinbockel, and E. Laderman, "Cyber resiliency engineering aid—the updated cyber resiliency engineering framework and guidance on applying cyber resiliency techniques," MITRE Corporation, Tech. Rep. MTR140499R1, 2015.

[2] L. Kaelbling, M. Littman, and A. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, no. 1–2, pp. 99–134, 1998.

[3] M. J. Kochenderfer, *Decision Making Under Uncertainty: Theory and Application*. MIT Press, 2015.

[4] P. Ramuhalli, M. Halappanavar, J. Coble, and M. Dixit, "Towards a theory of autonomous reconstitution of compromised cyber-systems," in *IEEE Conference on Technologies for Homeland Security*, 2013.

[5] S. Choudhary, L. Rodriguez, D. Curtis, K. Oler, P. Nordquist, P. Chen, and I. Ray, "Action recommendation for cyber resilience," in *Workshop on Automated Decision Making for Active Cyber Defense*, 2015.

[6] D. Bodeau, R. Graubart, and E. Laderman, "Cyber resiliency engineering overview of the architectural assessment process," *Procedia Computer Science*, vol. 28, pp. 838–847, 2014.

[7] R. Caralli, J. Allen, P. Curtis, D. White, and L. Young, "Cert® resilience management model," Software Engineering Institute, Tech. Rep., 2010.

[8] A. Servin and D. Kudenko, "Multi-agent reinforcement learning for intrusion detection: A case study and evaluation," in *Proceedings of the 6th German Conference on Multiagent System Technologies*, 2008.

[9] T. Lane, "A decision-theoritic, semi-supervised model for intrusion detection," in *Machine Learning and Data Mining for Computer Security*, 2006.

[10] O. Kreidl, "Analysis of a Markov decision process model for intrusion tolerance," in *IEEE/IFIP International Conference on Dependable Systems and Networks Workshops*, 2010.

[11] G. Tjhai, M. Papadaki, S. Furnell, and N. Clarke, "The problem of false alarms: Evaluation with Snort and DARPA 1999 dataset," in *International Conference on Trust, Privacy and Security in Digital Business*, 2008.

[12] F. Aleserhani, M. Akhlaq, I. Awan, J. Mellor, A. Cullen, and P. Mirchandani, "Evaluating intrusion detection systems in high speed networks," in *International Conference on Information Assurance and Security*, 2009.

[13] E. Bakhoum, "Intrusion detection model based on selective packet sampling," *EURASIP Journal on Information Security*, 2011.