Chapter 2
Control-Theoretic Approaches to Cyber-Security*

Erik Miehling, Mohammad Rasouli, and Demosthenis Teneketzis

Abstract In this chapter, we discuss the control-theoretic approach to cyber-security. Under the control-theoretic approach, the defender prescribes defense actions in response to security alert information that is generated as the attacker progresses through the network. This feedback information is inherently noisy, resulting in the defender being uncertain of the underlying status of the network. Two complementary approaches for handling the defender's uncertainty are discussed. First, we consider the probabilistic case where the defender's uncertainty can be quantified by probability distributions. In this setting, the defender aims to specify defense actions that minimize the expected loss. Second, we study the nondeterministic case where the defender is unable to reason about the relative likelihood of events. The appropriate performance criterion in this setting is minimization of the worst-case damage (minmax). The probabilistic approach gives rise to efficient computational procedures (namely sampling-based approaches) for finding an optimal defense policy, but requires modeling assumptions that may be difficult to justify in real-world cyber-security settings. On the other hand, the nondeterministic approach reduces the modeling burden but results in a significantly harder computational problem.

Erik Miehling
Coordinated Science Lab, University of Illinois at Urbana–Champaign, Urbana, IL 61801
e-mail: miehling@illinois.edu

Mohammad Rasouli
Department of Management Science & Engineering, Stanford University, Stanford, CA 94305
e-mail: rasoulim@stanford.edu

Demosthenis Teneketzis
Department of Electrical Engineering & Computer Science, University of Michigan, Ann Arbor, MI 48109
e-mail: teneket@umich.edu

* This work is partially supported by the Army Research Office under grant W911NF-13-1-0421.

2.1 Introduction

The field of control theory studies how one can make a sequence of decisions in order to most efficiently guide, or *control*, a system toward a specified objective subject to some uncertainty regarding the system's evolution. Some examples of problems addressed by control theory include maintaining a system's output at a desired set-point in the presence of external disturbances, *e.g.*, an aircraft autopilot system responsible for maintaining speed and altitude in varying weather conditions, or tracking a path or trajectory subject to measurement noise and estimation errors, *e.g.*, an autonomous vehicle's road following algorithm tasked with translating noisy measurements from multiple sensors into real-time steering, acceleration, and braking decisions. Depending on the control environment, the information available for making decisions can take different forms. In some settings, the current status of the system is directly observable and can be used in the decision making process. In others, the uncertainty is not only due to the effect of the control action on the evolution of the system, but also includes the inability to perfectly observe the system's status, requiring control decisions to be made based on noisy observations or measurements. In either setting, sequential control decisions must be made based on new, potentially noisy, information that is revealed as the problem evolves. The precise topic that control theory addresses is the nature of this *feedback loop* – the influence of control decisions on the observable output and the dependency of revealed information on the choice of subsequent control actions – with the end goal of prescribing *optimal* control actions, that is, those that achieve the objective at the lowest operational cost.

In this chapter, we study the role of control theory in cyber-security. In particular, we focus on the (dynamic) defense problem: how a defender can prescribe actions in real-time, as a function of a stream of intrusion information in order to interfere with, and potentially mitigate, attacks carried out by one or more adversaries.[2] It is worth emphasizing the defining characteristic of control theory, namely the one-sided nature of the decision-making process. As such, the control-theoretic approach studied in this chapter considers the defender as the *only active decision-maker* in the system.[3] All other decision-making processes that may be present in the system, *e.g.*, actions of the attacker(s) or the behavior of trusted (non-malicious) users, are abstracted into the model of the cyber environment. The one-sided nature of the control theoretic approach is in contrast with the two-sided (or in general, many-sided) decision making environment of game theory which consists of many agents, each possessing different information and (at least partially) conflicting objectives. Game-theoretic tools, specifically how they can be used to address the problems in cyber-security, are discussed in-depth in Chapter 3. While modeling the cyber-

[2] Such systems are referred to as *intrusion response systems* in the cyber-security literature; see [1] for a review of the area.

[3] In some control settings, the "decision maker" may actually consist of a collection of agents making decisions based on their own localized information in order to achieve some common objective. Such problems still fall within the realm of control theory, due to all agents having an identical objective, but are referred to as *decentralized control problems* or *team problems* [2], [3].

security problem as a control problem is an approximation of the true problem, it is a valuable first step for addressing the full complexity of the game-theoretic approach. Indeed, many of the challenges of the cyber-security problem present in the control-theoretic approach also exist in the game-theoretic approach.

The defense problem presents many challenging requirements from both modeling and computational perspectives. The problem is inherently dynamic, evolving over time as a function of the defender's actions and (potentially unobservable) events from the cyber environment. New information is continuously revealed to the defender as the problem evolves, all of which, in general, must be used in the defender's decision making process. The model for the cyber environment, termed the *threat model*, must be sufficiently expressive to describe the complex nature of attacks. In particular, attacks are *progressive*, consisting of multiple stages and involving the combination of many vulnerabilities across multiple network elements, and *persistent*, with attackers continuing to attempt their objective, using various attack pathways, until they are successful. The defender, in its attempts to interfere with or mitigate attacks, must be aware of the conflicting effects of its defense decisions on the system. It is faced with an unavoidable tradeoff between security and availability; performing system modifications that lower an attack's chance of success also interfere with the normal functionality and usability of the system by trusted users. Beyond modeling challenges, the defense problem presents significant challenges from a computational perspective. The systems that are targeted by cyber attacks are large-scale, consisting of many hosts, each containing a wide-range of software and operated by a large collection of users. Reasoning about all possible ways such systems can be attacked often leads to a combinatorial explosion in complexity. As a result, scalable algorithms must be developed, often requiring approximations or novel solution techniques (such as sampling methods or system decompositions). One must also ensure that algorithms are able to meet the strict timing requirements of the system by prescribing defense decisions quickly. Oftentimes, defense decisions have a limited window of usefulness; prescribing a defense decision too late can be as ineffective as taking no action at all.

The tools offered by control theory are a natural fit for addressing the aforementioned requirements. First, quantifying the status of the system through assignment of a state allows one to formally describe the evolution of the system's level of security as a function of the defense actions and events from the cyber environment (*e.g.*, the description of the threat model). Furthermore, under the state-based approach, one can define an appropriate cost structure (costs for states and actions) that captures the desired tradeoff between security and availability. Defending the system then amounts to determining actions that ensure the system stays out of undesirable (high-cost) regions of the state space. In general, the defender's decisions must be made based on all available information. The notion of an *information state* from control theory allows for a compression of the available information into a summary that is sufficient for making optimal decisions. Once an appropriate information state for the problem is identified, one can cast the problem of determining the optimal defense policy (the sequence of functions mapping the information state to actions) as a set of sequential optimization problems (via dynamic programming).

Computational concerns can then be more directly addressed by investigating approximations to the dynamic programming recursion, leading to approximately optimal defense policies.

In what follows, we discuss the philosophy behind the control theoretic approach to cyber-security. First, in Section 2.2, we describe the assignment of a state to quantify the level of security of the system and how this state evolves as a function of the defender's actions and the events from the cyber environment. The defender's lack of perfect information regarding the state, and how this is addressed, is discussed in Section 2.3. Section 2.4 introduces the notion of defense policies and the computational procedure for obtaining them. Section 2.5 provides two model instances of the general control-theoretic approach, differing primarily in the assumed nature of the uncertainty in the problem (probabilistic vs. nondeterministic). The general idea of each approach is described, as well as each model's benefits and drawbacks. Concluding remarks are provided in Section 2.6.

2.2 The state-based approach to cyber-security

At the heart of any control problem is the notion of a state. The state describes the current operating status of the system, quantifying how the system reacts to the control input and events from the environment, and influencing how the control translates to the observable output. Viewing cyber-security as a control problem first requires that one defines a state that accurately quantifies the level of security of the system. To this end, the state, denoted by $x_t \in \mathscr{S}$ at any given time t, should reflect some aspect of the attacker's current capabilities. For example, the state could represent the permissions that the attacker possesses or its progress (in terms of compromised hosts) toward reaching a specific target host. In Section 2.5, we will define the state in the context of two formal security models; for the current discussion, however, consider the state to be abstract representation of the system's security level.

The next ingredient in the control-theoretic description of cyber-security is the specification of the control, that is, the defender's actions. Defense actions can take a wide variety of forms. One class of such actions is patches. A patch for a vulnerability renders the corresponding exploit(s) ineffective, offering an effective strategy for hardening the system and interfering with the attacker's goal. Unfortunately, the time between discovery of the vulnerability and the installation of a patch, termed the *vulnerability exposure window*, can be upwards of five months [4].[4] As a result, relying solely on patches would inevitably allow systems to be operational while exposed to vulnerabilities. Alternative defensive measures that operate on faster time-scale than patches are needed. The defense actions we consider throughout this chapter use known vulnerabilities and security alert information to actively interfere with the attacker's progression. Specifically, a defense action at time t, denoted by

[4] For a deeper discussion of this issue, see the related topic of vulnerability disclosure policies [5].

$u_t \in \mathscr{A}$, corresponds to system modifications that directly influence the ability of the attacker to induce a state transition, $x_t \to x_{t+1}$. For example, a defense action may disable the precondition of an exploit (such as connectivity between two hosts via a specific port) in order to block the attacker from using the exploit. While not a permanent solution, these defense strategies can be effective for interrupting an attack, buying useful time for forensic analysis and the development of a patch.

Defense decisions are made based on the predicted evolution of the state under various defense actions. In order to carry out such a prediction, one needs a model for the attacker. The concept of *threat modeling* [6] from the computer security community addresses precisely this task. Informally, threat models describe what the attacker can do given its current capabilities. More specifically, a threat model describes the various ways in which an attacker can infiltrate the system (attack vectors/pathways), what resources it finds valuable (the attacker's objectives), and what sort of security information is generated/detected during an attack (*e.g.*, via intrusion logs). In the context of the control-theoretic approach of this chapter, the threat model describes how the state evolves as a function of defense actions and events from the cyber environment, as well as what observations are generated during this evolution. For example, given a set of attacker capabilities (quantified by the current state) the threat model serves to define what exploits the attacker can attempt and, given the defense action, an updated set of attacker capabilities and any security alerts that may have been generated during the attempt of the exploits. It is important to note that while the control theoretic approach requires a well-defined threat model, it need not be completely known *a priori*. Simultaneous learning of the model and control of the system based on feedback information still falls within the realm of control theory (termed *adaptive control* [7] and *reinforcement learning* [8]).

While the defender's primary objective is to prevent the attacker from reaching its goals, it must also consider the effect of its defense actions on the normal operation of the system. Defenses that are most effective at interfering with the attacker also tend to be most disruptive to the normal operation of the system (*e.g.*, shutting down the email server to block phishing emails). On the other hand, prioritizing system availability unavoidably preserves attack pathways. In short, keeping the attacker away from its goals is largely in conflict with maintaining availability. Quantifying this tradeoff is achieved by assigning costs to both states and defense actions, via a cost function $c(x_t, u_t)$. High costs should be assigned to undesirable states, *e.g.*, the attacker possessing root access on a critical host, as well as to actions that significantly limit availability. Using the threat model, the defender can reason about costs of state-action trajectories which in turn guide the selection of defense actions that achieve the desired security-availability tradeoff.

2.3 The defender's information

A fundamental aspect of the dynamic defense problem is that the defender cannot perfectly observe the attacker's activity, *i.e.*, the events from the cyber environment. Instead the defender receives observations, denoted by $y_t \in \mathscr{O}$, generated as a function of the underlying events. The monitoring devices that generate the observations are inherently noisy. For instance, intrusion detection systems suffer from both missed detections (generating no security alerts when something malicious has occurred) and false alarms (triggering security alerts in the absence of malicious behavior). As a consequence of this imperfect detection, the defender has uncertainty over the true state of the system.

The control-theoretic concepts of *information structure* and *information state*[5] allow one to formalize the defender's lack of perfect information regarding the true state. The information structure of a problem is a formal description of the phrase "who knows what about the system and when" [10]. Under the centralized control theoretic approach of this chapter, the information structure of the problem has a straightforward interpretation; it simply describes the set of variables that the defender knows at any given time. Throughout the chapter, it is assumed that the information structure satisfies *perfect recall*, that is, the defender remembers all of its past observations and defense decisions. In other words, at time t the defender has access to the history $h_t = (u_0, y_1, \dots, u_{t-1}, y_t) \in (\mathscr{A} \times \mathscr{O})^t$. Given that there is only one decision-maker, the problem is said to have a strictly classical information structure [11]. This allows one to compress the history into a summary, termed an *information state* and denoted by I_t, that has a time-invariant domain $\mathscr{I}$ [10]. The information state is sufficient for making optimal decisions, *i.e.*, basing decisions on the information state, rather than the whole history, is without loss of optimality. Treating the information state as the state of the problem, one can formulate a completely observable decision problem that admits a dynamic programming decomposition. The evolution of the information state is dictated by the new information that is revealed as time progresses (defense actions and observations).

2.4 Computation of defense policies

The defense action at any given time is computed as a function of the defender's current information (given by the information state). Formally, the translation from information states to defense actions is specified by a *defense policy*, denoted by $g = (g_0, g_1, \dots, g_{T-1})$, where T is the decision horizon (the finite horizon case will be considered in this chapter; however T can also be infinite) and each g_t is a function from the given information state I_t to a distribution over defense actions, that is, $g_t : \mathscr{I} \to \Delta(\mathscr{A})$. Determining the *best* defense policy depends on the defender's model for how events are generated (*i.e.*, how the attacker chooses its actions). As

[5] For a deeper discussion of information structures and information states, see [9].

will be discussed in more detail in Section 2.5, the assumed nature of uncertainty in the problem dictates the cost criterion for the problem. For example, if uncertainty is quantified by probability distributions and the defender is risk-neutral, the defender's objective may be to minimize the total expected cost. On the other hand, under nondeterministic uncertainty, an appropriate criterion would be to minimize the worst-case cost, a so called *minmax* criterion. The best defense policy, termed an optimal defense policy denoted by $g^* = (g_0^*, g_1^*, \ldots, g_{T-1}^*)$, is a policy that minimizes the corresponding cost criterion.

In general, each defense action has a long-run impact on the evolution of the system. As such, defense decisions cannot be made in isolation; one must balance immediate costs with future costs to ensure that early defense decisions don't result in the system ending up in an undesirable or vulnerable state. Reasoning about sequences of actions is a computationally formidable task, especially when the time horizon, T, is long. Fortunately, results from control theory allow one to sequentially decompose the long-run optimization problem into a collection of simpler subproblems. The sequential decomposition, known as dynamic programming, relies on a concept known as the *principle of optimality* [12, 13]. A problem is said to satisfy the principle of optimality if, given a sequence of optimal control actions from time t onward, the remainder of the action sequence from $t+1$ onward will still be optimal for the problem that starts from the state resulting from the action taken at t. The cost of the remainder of the action sequence from a given state, termed the cost-to-go, is captured by defining a *value function*. The value function represents the best that one can do from the given state. The resulting recursive expression, termed the Bellman (or dynamic programming) equation is solved in the finite horizon case by starting from the final decision time and working backwards, a process termed backward induction. In the infinite horizon case, one must solve a fixed point equation [13]. The optimal policies are recovered from the value functions by finding the action, for a given state, that minimizes the cost criterion.

Dynamic programming is the predominant approach for solving centralized control problems (and thus the dynamic defense problem studied in this chapter); however, it suffers from computational challenges as the problem size grows. The main challenge arises from the need to compute and store the value functions for every possible state. As the state space grows, this procedure becomes increasingly burdensome (referred to as the *curse of dimensionality*). Due to the very large state space in many cyber-security settings, the curse of dimensionality becomes a significant issue for the dynamic defense problem This problem is further compounded by the fact that the defender possesses imperfect information of the state; the domain of the value functions is thus the set of information states $\mathscr{I}$, an uncountably infinite space. These challenges preclude the computation of optimal actions for every possible (information) state. One must resort to approximations of the dynamic programming recursion, resulting in approximately optimal defense policies. As will be illustrated in the following section, the information state of the problem provides guidance for an appropriate approximation, allowing for scalable and fast computation without significantly impacting decision quality.

2.5 Some models from the literature

There is a large body of research concerning the design of systems that prescribe automated defenses based on real-time intrusion information. Such systems are referred to by various names in the literature: automated intrusion response systems [14], autonomic & self-protecting systems [15, 16], and survivable systems [17, 18], among others [19, 20]. The seminal work of [18] was the first to investigate the design of such a system from a formal, control-theoretic perspective. More recent work has taken a similar approach, developing control-theoretic automated defense systems for completely observable [15, 16, 20] and partially observable settings [1, 19, 21–23].

This section will focus on the partially observable setting. In particular, we investigate two complementary approaches to modeling the defender's uncertainty: 1) probabilistic uncertainty, and 2) nondeterministic uncertainty. Probabilistic uncertainty quantifies all uncertainty in the problem via probability distributions. For instance, under a given defense action, the transition from one state to another is assumed to be dictated by probabilities. The second approach, nondeterministic uncertainty, considers a more coarse form of uncertainty where one only knows the possible events and not their specific probabilities. For each setting, the general decision environment and form of the information state is described. To aid in exposition, we draw upon two existing models developed in the literature, namely [1, 21] for the probabilistic approach and [22, 23] for the nondeterministic approach. In both cases, solving for an optimal defense policy is intractable, requiring solution techniques that yield approximate defense policies. Each section concludes with a general discussion of the benefits and drawbacks of the respective modeling approach.

2.5.1 Probabilistic uncertainty

The first approach assumes that the nature of the defender's uncertainty is probabilistic. Under probabilistic uncertainty, the state transitions and the generation of observations are assumed to be dictated by probability distributions. In particular, the state dynamics follow a controlled Markov chain[6] where the control is the defender's action, as illustrated by Fig. 2.1.

An implicit assumption in this setting is that the underlying distributions characterize, as a function of the defense action, all uncertainty associated with the attacker's behavior. In particular, given a current state $x_t = s_i$ and a defense action $u_t = a$, the transition to the next state $x_{t+1} = s_j$ is given by a fixed conditional probability $p^a_{ij} = \mathbb{P}(X_{t+1} = s_j \mid X_t = s_i, U_t = a)$.[7] Further, given a successor state

[6] This is a special case of a general probabilistic automaton where the dynamics are assumed to be Markovian.

[7] The uppercase notation, X_t, is used to represent a random variable.

$x_{t+1} = s_j$ and an action $u_t = a$, an observation $y_{t+1} = o_k$ is generated according to the conditional probability $r^a_{jk} = \mathbb{P}(Y_{t+1} = o_k \mid X_{t+1} = s_j, U_t = a)$.

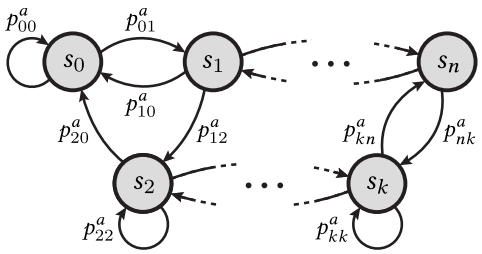

Fig. 2.1: Under the assumption of probabilistic uncertainty, the state dynamics evolve probabilistically. We represent this evolution as a controlled Markov chain where the control is the defense action.

The above described model is known in the literature as a *partially observable Markov decision process* (POMDP). It is well known that the information state in a POMDP is the conditional probability measure, that is, the probability mass function on the state space $\mathscr{S}$ conditioned on the history $h_t = (u_0, y_1, \ldots, u_{t-1}, y_t)$ [24]. The information state (also referred to as the *belief state*) is denoted by $I_t = \pi_t \in \Pi = \Delta(\mathscr{S})$ where $\Delta(\mathscr{S})$ is the probability simplex on the state space (the space of all probability mass functions on $\mathscr{S}$). The belief state π is updated via Bayes rule, as a function of the new information $(u_t, y_{t+1}) = (a, o)$, to $\pi' = (\tau_1(\pi, a, o), \ldots, \tau_n(\pi, a, o))$ where $\tau_j(\pi, a, o) = \sum_i \pi_i p^a_{ij} r^a_{jk} / \sum_i \sum_j \pi_i p^a_{ij} r^a_{jk}$.

Under probabilistic uncertainty, an appropriate performance metric is that of total expected discounted cost. The cost for a given defense policy $g : \Delta(\mathscr{X}) \to \Delta(\mathscr{A})$ is defined as

$$C(g) = \mathbb{E}\left[\sum_{t=0}^{T-1} \beta^t c(x_t, u_t) + \beta^T c(x_T)\right].$$

where $c : \mathscr{S} \to \mathbb{R}_+$ is a (terminal) cost function that only depends on the final state x_T, and $\beta \in (0,1)$ is a *discount factor* which serves to place more weight on immediate costs compared later costs. The expectation above is taken with respect to the joint probability distribution on trajectories $(x_0, u_0, \ldots, x_{T-1}, u_{T-1}, x_T)$ as a result of defense policy g. An optimal defense policy g^* is one that minimizes the total expected discounted cost $C(g)$, that is, $g^* = \inf_g C(g)$. Recalling the discussion of Section 2.4, optimal defense policies are computed from the value function. The value function in the probabilistic uncertainty case is defined on the space of beliefs $\Delta(\mathscr{S})$ and is denoted by $V : \Delta(\mathscr{S}) \to \mathbb{R}$. Using the likelihoods encoded by

the belief and the probabilities described by the model, one can write the dynamic programming equations, for every $\pi \in \Delta(\mathscr{S})$ and $t = 0, \dots, T-1$, as

$$\begin{aligned} V_t^*(\pi) &= \min_{a \in \mathscr{A}} \mathbb{E}\big[c(x,a) + \beta V_{t+1}^*(\tau(\pi,a,y))\big] \\ &= \min_{a \in \mathscr{A}} \left\{ \sum_i \pi_i c(s_i,a) + \beta \sum_k \sum_j \sum_i \pi_i p_{ij}^a r_{jk}^a V_{t+1}^*(\tau(\pi,a,o_k)) \big] \right\} \end{aligned}$$

with terminal value function $V_T^* = \mathbb{E}\big[c(x)\big]$. The solution of the above equations can, in principle, be obtained via a recursive computational procedure (*i.e.*, value iteration) which, in turn, yields a corresponding optimal defense policy g^*. Unfortunately, due to the scale of real-world cyber-security problems, one must resort to approximate procedures, as will be described later.

To provide context for the probabilistic approach, we review a model from the literature. The automated intrusion response system, developed in [1,21], models how a defender can optimally interfere with the progression of an adversary through a computer network. The progression of the attacker is described by a directed acyclic graph, termed an *attack graph*, that encodes the relationships between exploit preconditions (attacker capabilities that are needed to attempt the exploit) and postconditions (attacker capabilities that are realized upon success of the exploit). The state of the system at any given time is the set of currently enabled conditions. As the attacker attempts exploits and moves through the network, alerts are triggered via an intrusion detection system. The defender uses these noisy security alerts to construct a belief of the currently enabled conditions. Using the belief, the defender prescribes actions that induces system modifications that block exploits from being carried out. While these system modifications interfere with the progression of the attacker (the evolution of the state), they are also costly, requiring the defender to tradeoff between interfering with the attacker's progression and maintaining system availability.

The novelty of the model developed in [1,21] is the use of attack graphs to model the active progression of an attacker through a network. Prior work primarily considered attack graphs in the context of offline vulnerability analysis, *e.g.*, determining the minimum number of exploits to patch in order to maximize the number of blocked attack paths [25]. Introducing a state and using the attack graph to model the dynamics of the state process enables one to build a control (defense) problem and compute defense policies that optimally interfere with an attack as it is unfolding.

While one can write down the dynamic programming equations that characterize an optimal policy, offline computation for every possible belief that may be encountered during runtime is intractable. This is primarily due to the scale of real-world attack graphs and the size of the resulting state space. To avoid this challenge (termed the *curse of dimensionality*) we take advantage of the fact that the defender's uncertainty is described by probability distributions. In particular, the defender is able to forecast future possible attack pathways (chains of exploits) by sampling from the model's distributions. By conditioning on its current belief of the attacker's capabilities, the defender can reason about the likelihood (and expected costs) of various

state trajectories under different defense actions. This allows the defender to prescribe defense actions that guide the system to low cost regions of the state space and reach outcomes that balance between security and availability. Such an approach is termed an online algorithm [1, 26] since one is only concerned with prescribing actions from the current (belief) state. While the online defense algorithm requires one to continue to perform computation during runtime, it is much more scalable than offline approaches, yielding good quality defense policies in large domains. Additional details of the algorithm can be found in [1].

The benefit of taking a probabilistic approach to the cyber-security problem is primarily computational. Quantifying uncertainty via probability distributions enables the application of scalable computational procedures for computing defense policies. In particular, (provably convergent) solutions techniques based on sampling can readily be applied [1,26]. However, the probabilistic approach raises some concerns in real-world cyber-security settings. The primary concern is the specification of accurate probability parameters for describing the attacker's behavior. The usual justification for knowing the parameters in a general stochastic control setting is that one has learned them from existing data and previous runs of the problem. This is difficult to justify in the context of cyber-security: attacks are targeted and rarely repeated, leading to sparsity of useful attack data. That said, it is not necessary to specify accurate probabilities for the model to have value. The models can still provide useful qualitative insights that are not sensitive to the specific parameter values. For example, the sampling approach can identify, and focus defensive resources on, structural bottlenecks in the attack graph [1]. These structural properties of the problem are largely independent of the specific probability values. A secondary concern is the question of whether the assumed probabilities are informative for future evolution of the system. This requires that the statistics dictating the attacker's behavior do not change in time, *e.g.*, they are stationary, an assumption which may be difficult to justify in practice. This issue will be discussed in more detail in the following section.

One approach for addressing the above concerns is to consider a *set* of models, as done in [1]. Consideration of sets of models allows one to capture a wide-range of attacker behavior by not only updating the estimate of the attacker's evolution, but also the estimate of the true model. However, considering a large set of models further compounds computational difficulties. Obtaining an appropriate tradeoff between model expressiveness and computational tractability depends on the specific security setting.

2.5.2 Nondeterministic uncertainty

The probabilistic approach discussed in Section 2.5.1 is not the only way to reason about uncertainty [27]. A more coarse description of uncertainty, termed *nondeterministic uncertainty*, places no assumptions on how events from the cyber environment are generated. Under nondeterminism, one cannot reason about the probability

of events and thus cannot construct likelihoods of individual states. One can only reason about the set of possible states that are consistent with the available information [28–30]. In other words, one keeps track of the support of the distribution and not the likelihoods. Due to the lack of probabilities, the defender cannot differentiate between the set of possible states in the support. As a result, the defender adopts a worst-case cost criterion. Assuming the worst-case can be interpreted as the defender preparing for the attacker to perform the most damaging action (or even further, taking the most conservative viewpoint by assuming that the attacker is omniscient and is able to compute and execute this action). Throughout the discussion, we refer to the attacker as *nature*. We adopt this terminology since we are not modeling an explicit strategy for the attacker.[8] Throughout this section, the general model from the literature will be described along with a discussion of their application within specialized cyber-security models, in particular [22, 23].

The general system model consists of a finite set of states, $\mathscr{S} = \{s_0, s_1, \ldots, s_n\}$, where the state transition $x_t \to x_{t+1}$ is due to both the defender's action, $u_t \in \mathscr{A}$, and the event, $w_t \in \mathscr{E}$, from the cyber environment. Formally, we describe the dynamical system as a nondeterministic finite automaton (NFA). For any given state, the transition due to an action-event pair $(u_t, w_t) = (a, e) \in \mathscr{A} \times \mathscr{E}$ is in general nondeterministic, that is, the state may transition to one of a set of states, as illustrated by Fig. 2.2. The distinguishing feature of the nondeterministic case compared to the probabilistic case is that, in the latter, the defender cannot reason about the relative likelihood of transitioning to various successor states and must treat all successor states, from a given state, as possible.

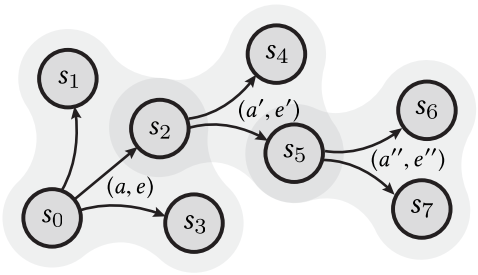

Fig. 2.2: Under nondeterministic uncertainty, the state dynamics are modeled by a nondeterministic finite automaton. For a given action-event pair, (a, e), state transitions are nondeterministic, meaning a given state can transition to one of a collection of states.

[8] Such settings are sometimes referred to as *games against nature* in the literature [31]; however, since no strategy is assumed for the attacker (nature), it is not viewed as an active decision maker, and thus we view the problem in the context of control theory.

The state dynamics encoded by the nondeterministic finite automaton are described by the function $f_t : \mathscr{S} \times \mathscr{A} \times \mathscr{E} \to \mathscr{S}$, that is, given an action-event pair $(u_t, w_t) = (a, e)$ the state $x_t = s$ follows the update $x_{t+1} = f_t(s, a, e)$. The defender does not perfectly observe the state or nature's events. Instead, it receives an observation y_t generated as a function of the true underlying state and the event, as described by the function $l_t : \mathscr{S} \times \mathscr{E} \to \mathscr{O}$. A slightly more general cost function is considered in this section, namely one that depends on nature's event in addition to the state-action pair, that is, $c : \mathscr{S} \times \mathscr{A} \times \mathscr{E} \to \mathbb{R}_+$. It is assumed that the cost function is bounded above by $\bar{c} < \infty$, that is, $c(s, a, e), c(s) \leq \bar{c}$ for all $(s, a, e) \in \mathscr{S} \times \mathscr{A} \times \mathscr{E}$. Define $\mathscr{C} = \left[0, \frac{1-\beta^{T+1}}{1-\beta}\bar{c}\right]$ as the possible range of cumulative costs accrued over the duration of the problem.[9]

The problem of decision-making under nondeterministic uncertainty has been extensively studied in the literature. Early work, see [32–36], has established a duality between probabilistic and nondeterministic uncertainty, proposing *cost measures*, *cost densities*, and *feared values* as analogous concepts to probability measures, probability densities, and expected values. Further connections to robust control and game theory have been established in [34, 37]. As illustrated in the literature, the relevant notion in the nondeterministic case is that of cost, rather than probabilities. In particular, one should base control decisions on the (worst-case) cost for reaching each state as opposed to reasoning about their likelihoods.

An appropriate information state in this setting is the *worst-case cost-to-come* statistic of [34, 36, 37], denoted by $I_t = \theta_t \in \Theta = (\mathscr{C} \cup \{-\infty\})^n$, defined as the maximum possible cost for reaching each state given the current history. That is, for any given time t, the information state θ_t consists of a collection of costs, one for each state, $\theta_t = \{\theta_t(s)\}_{s \in \mathscr{S}}$, where each $\theta_t(s)$ is defined as the maximum cost for reaching state s. If state s is not consistent with the current history then the corresponding $\theta_t(s)$ is assigned a negative infinite value. Given new information $(u_t, y_t) = (a, o)$, the information state θ_t is updated via the rule $\theta_{t+1} = \mu(\theta_t, a, o)$. To describe the update, define $\Omega(s', a, o) := \{(s, e) \in \mathscr{S} \times \mathscr{E} \mid s' = f_t(s, a, e), o = l_t(s, e)\}$ as the set of state-event pairs that are consistent with the new information $(u_t, y_t) = (a, o)$ given that the system is in state $s' \in \mathscr{S}$. Each component of the updated information state, $\theta_{t+1}(s')$, is computed by searching over all state-event pairs $(s, e) \in \Omega(s', a, o)$ in order to find the maximal cost for reaching $x_{t+1} = s'$ consistent with the new information $(u_t, y_t) = (a, o)$. Further details of the information state update in a general setting can be found in Ch. 6 of [37] and Sec. 2.3 of [36], as well as in the context of cyber-security in [23].

Since one does not have access to probability distributions in the nondeterministic setting, the notion of expected value is no longer relevant. An appropriate cost criterion in this setting is minimization of the worst-case cost. The worst-case cost for a given defense policy $g : (\mathscr{C} \cup \{-\infty\})^n \to \Delta(\mathscr{A})$ is

$$D(g) = \max_z \left[\sum_{t=0}^{T-1} \beta^t c(x_t, u_t, w_t) + \beta^T c(x_T) \right]$$

[9] For the infinite horizon case, $\mathscr{C} = \left[0, \frac{\bar{c}}{1-\beta}\right]$.

where the maximization is taken over all feasible trajectories of the form $z = (x_0, u_0, w_0, \ldots, x_{T-1}, u_{T-1}, w_{T-1}, x_T)$ as a result of defense policy g. An optimal defense policy is one that minimizes the worst-case cost $D(g)$, that is, $g^* = \inf_g D(g)$. As before, one can construct a dynamic programming recursion on the space of information states in order to recursively compute a value function, denoted by $W : (\mathscr{C} \cup \{-\infty\})^n \to \mathbb{R}$, and a corresponding optimal policy. Defining $\mathscr{O}_t$ as the set of observations that are consistent with the current information at time t,[10] one can write the dynamic programming equations, for each $\theta \in (\mathscr{C} \cup \{-\infty\})^n$ and $t = 0, \ldots, T-1$, as

$$W_t^*(\theta) = \min_{a \in \mathscr{A}} \max_{o \in \mathscr{O}_t} \left[W_{t+1}^*(\mu(\theta, a, o)) \right]$$

with terminal value function $W_T^*(\theta) = \max_{s \in \mathscr{S}} [c(s) + \theta(s)]$. Note that, unlike in the probabilistic case of Sec. 2.5.1, the cost function is embedded within the definition of the information state itself and does not explicitly appear in the dynamic programming equations.

The computational challenges are more pronounced in the nondeterministic case compared to the probabilistic case. The two main challenges are: i) complexity of maintaining the information state θ_t, and ii) solving the dynamic programming equations. To address the first challenge, the model of [22, 23] considered a simplified information state in which one only keeps track of the set of states consistent with the current history. That is, the information state θ_t is approximated by the set of states s that have a finite $\theta_t(s)$. The simplification leads to a much simpler information state but comes at the cost of optimality. The second challenge, solving the dynamic programming equations, cannot be addressed by the sampling-based approach outlined in the previous subsection (since we do not have access to probability distributions). Instead, the problem is approximated by (spatially) decomposing the system into a collection of sub-systems. By analyzing the functional dependencies between the state components, one can construct a graph that quantifies the strength of the coupling between states. One can then apply clustering algorithms to partition the graph into sub-systems, each associated with a local defense policy. Allowing defense policies to communicate the necessary security information via messages, the computation of the defense policy can be decomposed into the computation of multiple local defense policies. This improves scalability and permits computation in some moderately-sized settings. Additional details of the decomposition approach can be found in [23].

The main benefit of taking a nondeterministic approach to the defender's uncertainty is the increased modeling flexibility. Reasoning over possible transitions and computing the worst-case includes a wide-range of attacker strategies, even non-stationary behavior. Furthermore, the modeling task is greatly simplified, compared to the probabilistic approach, as one does not need to make claims about which states are more or less likely to be realized. The nondeterministic approach does come with some drawbacks. The main issue is computational – even after the sim-

[10] In other words, $\mathscr{O}_t$ is the range of the functions $w \mapsto l_t(x_t, w)$.

plification of the information state to describe the set of consistent states, the space of approximate information states is the power set of the state space and thus scales poorly. Furthermore, the defense policies computed under the minmax approach can be overly conservative. Indeed, always assuming the worst possible state transitions is a very pessimistic viewpoint. This can be problematic as the attacker may prescribes attacks for the sole purpose of triggering conservative defenses, causing the defender to essentially carry out a denial-of-service attack on itself. Integrating elements of the probabilistic uncertainty approach into the nondeterministic approach can help to alleviate this issue. In particular, considering a range of distributions over which the worst-case is taken (to obtain the *least favorable distribution* [37]) permits one to regulate the degree of pessimism in computing defense policies.

2.6 Concluding remarks

Fundamental to the control-theoretic approach is the assumption that the defense problem is one-sided, that is, the defender is the only active decision maker. As such, the threat model serves to absorb the attacker's behavior into the model of the cyber environment. Computational limitations preclude specification of a complete threat model, that is, a full representation of the system (*e.g.*, active services/software, all active users and associated privilege levels, network connectivity, and trust relationships). One must make approximations, specifying threat models that include coarser state (*e.g.*, attacker privilege levels) and observation processes (*e.g.*, noisy security alerts from an intrusion detection system). This unavoidably introduces uncertainty, requiring the defender to estimate the true security status of the system from the observable signals.

Two complementary approaches to handling the defender's uncertainty have been discussed, namely probabilistic uncertainty and nondeterministic uncertainty. Probabilistic uncertainty assumes that the defender's uncertainty can be quantified by probability distributions. While permitting efficient (sampling-based) computational procedures for determining defense policies, taking a probabilistic approach requires some assumptions, *e.g.*, stationarity, that may be difficult to justify in real-world cyber-security settings. Alternatively, the nondeterministic approach leads to both a simpler modeling task (there is no longer a requirement to specify probability parameters) and a more flexible model (allowing for a description of non-stationary behavior). However, these benefits come at a cost of a harder computational problem.

The control-theoretic approach discussed in this chapter provides a foundation for extension to more general settings. One natural extension is to consider the case where the model of the cyber environment is unknown. To address this problem, standard tools from Bayesian adaptive control and reinforcement learning [8] may not be sufficient. The main challenge arises from the need to obtain large quantities of useful attack data. Concepts such as *transfer learning* [38] and *generalization* [39] may be useful for dealing with the sparsity and reproducibility issues in

attack data. Another extension is the consideration of more complex threat models, primarily allowing for both the attacker and the defender to be active decision makers. Such a modification results in a game-theoretic interaction where both the attacker and defender optimally respond to each other's actions. The complexities associated with the game-theoretic approach to cyber-security, as well as some foundational results, are presented in the following chapter.

References

1. E. Miehling, M. Rasouli, D. Teneketzis, A POMDP approach to the dynamic defense of large-scale cyber networks, IEEE Transactions on Information Forensics and Security 13 (10) (2018) 2490–2505.
2. J. Marschak, R. Radner, Economic Theory of Teams, Yale University Press, 1972.
3. Y.-C. Ho, M. Kastner, E. Wong, Teams, signaling, and information theory, IEEE Transactions on Automatic Control 23 (2) (1978) 305–312.
4. B. Gorenc, F. Sands, Hacker machine interface: The state of SCADA HMI vulnerabilities, Tech. rep., Trend Micro Zero Day Initiative Team (2017).
5. A. Arora, R. Telang, H. Xu, Optimal policy for software vulnerability disclosure, Management Science 54 (4) (2008) 642–656.
6. A. Shostack, Threat modeling: Designing for security, John Wiley & Sons, 2014.
7. P. R. Kumar, P. Varaiya, Stochastic Systems: Estimation, Identification, and Adaptive Control, Prentice Hall, NJ, 1986.
8. R. S. Sutton, A. G. Barto, R. J. Williams, Reinforcement learning is direct adaptive optimal control, IEEE Control Systems 12 (2) (1992) 19–22.
9. A. Mahajan, N. C. Martins, M. C. Rotkowitz, S. Yüksel, Information structures in optimal decentralized control, in: 51st Annual Conference on Decision and Control (CDC), IEEE, 2012, pp. 1291–1306.
10. A. Mahajan, M. Mannan, Decentralized stochastic control, Annals of Operations Research 241 (1-2) (2016) 109–126.
11. J. H. van Schuppen, Information structures, in: J. H. van Schuppen, T. Villa (Eds.), Coordination Control of Distributed Systems, Springer International Publishing, 2015, Ch. 24, pp. 197–204.
12. R. Bellman, Dynamic Programming, Princeton University Press, 1957.
13. D. P. Bertsekas, Dynamic Programming and Optimal Control, Vol. 1, Athena Scientific, 1995.
14. A. Shameli-Sendi, N. Ezzati-Jivan, M. Jabbarifar, M. Dagenais, Intrusion response systems: Survey and taxonomy, International Journal of Computer Science and Network Security 12 (1) (2012) 1–14.
15. S. Iannucci, S. Abdelwahed, A probabilistic approach to autonomic security management, in: IEEE International Conference on Autonomic Computing (ICAC), IEEE, 2016, pp. 157–166.
16. S. Iannucci, S. Abdelwahed, A. Montemaggio, M. Hannis, L. Leonard, J. King, J. Hamilton, A model-integrated approach to designing self-protecting systems, IEEE Transactions on Software EngineeringEarly access.
17. S. M. Lewandowski, D. J. Van Hook, G. C. O'Leary, J. W. Haines, L. M. Rossey, Sara: Survivable autonomic response architecture, in: DARPA Information Survivability Conference & Exposition II (DISCEX), Vol. 1, IEEE, 2001, pp. 77–88.
18. O. P. Kreidl, T. M. Frazier, Feedback control applied to survivability: A host-based autonomic defense system, IEEE Transactions on Reliability 53 (1) (2004) 148–166.
19. S. Musman, L. Booker, A. Applebaum, B. Edmonds, Steps toward a principled approach to automating cyber responses, in: Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications, Vol. 11006, International Society for Optics and Photonics, 2019, pp. 1–15.

20. P. Speicher, M. Steinmetz, J. Hoffmann, M. Backes, R. Künnemann, Towards automated network mitigation analysis, in: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, ACM, New York, NY, USA, 2019, pp. 1971–1978.
21. E. Miehling, M. Rasouli, D. Teneketzis, Optimal defense policies for partially observable spreading processes on Bayesian attack graphs, in: Proceedings of the Second ACM Workshop on Moving Target Defense, ACM, 2015, pp. 67–76.
22. M. Rasouli, E. Miehling, D. Teneketzis, A supervisory control approach to dynamic cybersecurity, in: International Conference on Decision and Game Theory for Security, Springer, 2014, pp. 99–117.
23. M. Rasouli, E. Miehling, D. Teneketzis, A scalable decomposition method for the dynamic defense of cyber networks, in: Game Theory for Security and Risk Management, Springer, 2018, pp. 75–98.
24. R. D. Smallwood, E. J. Sondik, The optimal control of partially observable Markov processes over a finite horizon, Operations research 21 (5) (1973) 1071–1088.
25. M. Albanese, S. Jajodia, S. Noel, Time-efficient and cost-effective network hardening using attack graphs, in: 42nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), IEEE, 2012, pp. 1–12.
26. D. Silver, J. Veness, Monte-Carlo planning in large POMDPs, in: Advances in Neural Information Processing Systems, 2010, pp. 2164–2172.
27. T. R. Besold, A. A. Garcez, K. Stenning, L. van der Torre, M. van Lambalgen, Reasoning in non-probabilistic uncertainty: Logic programming and neural-symbolic computing as examples, Minds and Machines 27 (1) (2017) 37–77.
28. H. Witsenhausen, Sets of possible states of linear systems given perturbed observations, IEEE Transactions on Automatic Control 13 (5) (1968) 556–558.
29. F. Schweppe, Recursive state estimation: Unknown but bounded errors and system inputs, IEEE Transactions on Automatic Control 13 (1) (1968) 22–28.
30. D. P. Bertsekas, Control of uncertain systems with a set-membership description of the uncertainty, Tech. rep., DTIC Document (1971).
31. J. Milnor, Games against nature, in: C. H. Coombs, R. L. Davis, R. M. Thrall (Eds.), Decision Processes, John Wiley & Sons, 1954, pp. 49–60.
32. M. Akian, J.-P. Quadrat, M. Viot, Bellman processes, in: 11th International Conference on Analysis and Optimization of Systems Discrete Event Systems, Springer, 1994, pp. 302–311.
33. P. Bernhard, Expected values, feared values, and partial information optimal control, in: New Trends in Dynamic Games and Applications, Springer, 1995, pp. 3–24.
34. P. Bernhard, A separation theorem for expected value and feared value discrete time control, ESAIM: Control, Optimisation and Calculus of Variations 1 (1996) 191–206.
35. M. Akian, J.-P. Quadrat, M. Viot, Duality between probability and optimization, Idempotency 11 (1998) 331–353.
36. P. Bernhard, Minimax – or feared value – $L1/L\infty$ control, Theoretical Computer Science 293 (1) (2003) 25–44.
37. T. Başar, P. Bernhard, H-Infinity Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach, Springer Science & Business Media, 2008.
38. K. Weiss, T. M. Khoshgoftaar, D. Wang, A survey of transfer learning, Journal of Big data 3 (1) (2016) 9.
39. J. Oh, S. Singh, H. Lee, P. Kohli, Zero-shot task generalization with multi-task deep reinforcement learning, in: Proceedings of the 34th International Conference on Machine Learning, JMLR, 2017, pp. 2661–2670.