A multi-layered and kill-chain based security analysis framework for cyber-physical systems

Adam Hahn, Roshan K. Thomas, Ivan Lozano, Alvaro Cardenas

School of Electrical Engineering and Computer Science, Washington State University, Pullman, Washington 99164, USA
The MITRE Corporation, MS T340, 7515 Colshire Drive, McLean, VA 22102, USA
Erik Jonsson School of Engineering and Computer Science, University of Texas at Dallas, 800 West Campbell Road, EC31, Richardson, TX 75080, USA

Abstract

This paper introduces a novel framework for understanding cyber attacks and the related risks to cyber-physical systems. The framework consists of two elements, a three-layered logical model and reference architecture for cyber-physical systems, and a meta-model of cyber-physical system attacks that is referred to as the cyber-physical system kill-chain. The layered reference architecture provides a systematic basis for studying how the causal chain associated with cyber perturbations can be traced all the way to physical perturbations. The cyber-physical system kill-chain describes the progressive stages of attacks to illuminate the steps required for an attacker to launch a successful attack against a cyber-physical system. The proposed framework offers a novel approach for comprehensively studying the elements of cyber-physical system attacks, including the attacker objectives, cyber exploitation, control-theoretic properties and physical system properties. The framework is evaluated using a simulated unmanned aerial system and the results of the evaluation are discussed. The longer-term goal is to use the framework as a means to deduce cyber-physical system security properties and to enumerate the principles for designing systems that are resilient to cyber attacks.

Article info

Article history:
Received 13 January 2015
Received in revised form 9 August 2015
Accepted 15 August 2015
Available online 29 August 2015

Keywords:
Cyber-physical system security
Kill-chain
Cyber reconnaissance
Controllability
Observability

1. Introduction

Attacks on cyber-physical systems (CPSs) have been observed with increasing frequency and have attracted the attention of the research community and industry. However, a common attack analysis framework and related design principles for resilient cyber-physical systems have not yet emerged. Organizations such as NIST have emphasized that accurate security threat models are critical to designing secure cyber-physical systems [17].

This paper introduces a novel framework for analyzing cyber attacks against cyber-physical systems. The framework provides a foundation for understanding cyber attacks and the related risks to cyber-physical systems, and for articulating design principles that will help engineer resilient cyber-physical systems.

The framework incorporates two elements. The first element comprises a logical system model and reference architecture that express the architectural composition of cyber-physical systems in terms of three interrelated layers. The bottom physical layer models the physical dynamics and properties of a control system, including its sensors and actuators. The middle control layer embodies a mathematical model of the control logic and various control algorithms, state
estimators, error correction and sensor feedback mechanisms that are used to observe and manipulate the physical system. Finally, the top cyber layer expresses the control system in a computerized platform, such as a programmable logic controller. The cyber layer utilizes the Architecture Analysis and Design Language (AADL) to model communications buses, data, messages, processes and systems.

The second element of the framework is a cyber-physical system kill-chain for analyzing attacks and threat models in the cyber-physical system domain. This meta-model of cyber attacks helps understand the various lifecycle phases that make up an attack. It is analogous to the cyber kill-chain originally proposed by Hutchins et al. [10]. It clarifies the value of intelligence-driven defenses that emphasizes an understanding of adversarial tactics, techniques and procedures as the basis for designing system defenses. The intelligence-driven defense approach has been used to predict the vulnerabilities that could be exploited to launch attacks [8]. Meanwhile, the U.S. Department of Defense has used the kill-chain to design cyber security testing and evaluation activities [18].

By combining a layered reference model with the cyber-physical system kill-chain, the proposed framework supports the systematic analysis of causal chains and perturbations that emanate from the cyber layer all the way to the physical impact on the cyber-physical system. The utility of the framework is demonstrated using a case study involving the control system of a simulated unmanned aerial system (UAS). A number of cyber attacks were launched against various unmanned aerial system components and the resulting physical system impacts were evaluated. The simulations were then used to explore how the cyber, control and physical properties influence attack impacts. The analysis provides insights into defensive strategies that are effective during the various cyber-physical kill-chain phases.

2. Previous work

The last few years have seen the publication of numerous research papers that analyze cyber vulnerabilities in cyber-physical systems. Cardenas et al. [2] have demonstrated how attacks can impact different elements of a control loop in a cyber-physical system. Several research efforts have explored theoretical and real-world attacks against cyber-physical systems; these attacks have used a broad range of attack vectors and have targeted various elements of the control loop. Fig. 1 identifies the control loop components that can be impacted by cyber attacks, including measurements, actuator signals, controllers and reference signals.

By manipulating sensor signals, an attacker can corrupt the state estimates computed by a controller and then change the resulting control signals sent to actuators. Liu et al. [15] have shown that cyber attackers can manipulate power system state estimation algorithms by sending fraudulent measurements to state estimators. Other researchers have demonstrated that unmanned autonomous vehicles can be hijacked by spoofing GPS signals that cause controllers to send incorrect control actions [21]. Additionally, researchers have demonstrated that wireless signals can be used to manipulate medical devices and cause drug overdoses [14,19].

Numerous attacks have demonstrated the ability to manipulate control algorithms executing on controllers by abusing administrative rights or exploiting software vulnerabilities. By manipulating a controller, an attack can directly influence the control signals sent to actuators. The Stuxnet worm is an excellent example of an attack on a controller—it manipulated ladder logic device configuration code in a programmable logic controller to send malicious commands to frequency converters that directly controlled centrifuge rotors [4]. Similarly, the Aurora attack demonstrated that electric power generators could be physically damaged if their frequency control is made out-of-sync with the frequency control of the power grid [22]. Additionally, Halperin et al. [9] have demonstrated that controllers in implanted cardiac devices can be remotely reprogrammed to cause incorrect treatments and/or dosages to be given to patients.

An attacker can manipulate control actions by directly sending commands to actuators. For example, researchers have directly injected packets into the CAN buses of modern automobiles, resulting in malicious control actions being sent to critical actuators (e.g., engines and brakes) [3,13].

3. Cyber-physical system model

This section introduces a three-layer representation of a cyber-physical system $S$ as demonstrated in Fig. 2. The model layers include the physical layer $S_{phys}$, control layer $S_{con}$ and cyber layer $S_{cyb}$. Each layer is defined in detail below.

![Fig. 1 - Cyber-physical system attacks.](image1)

![Fig. 2 - Logical cyber-physical system model.](image2)
3.1 Physical layer

The physical layer represents the physical rendering of the cyber-physical system. It captures the physical properties of the system and the physical architecture, including the decisions involving the process variables that are measured using sensors and the manipulated variables that are controlled using actuators.

The physical properties of a system are characterized by the plant dynamics, which can be linear or nonlinear, deterministic or stochastic, time varying or time-invariant, hybrid or non-hybrid, and fast changing or slow changing (e.g., power grid voltages and currents can change and propagate in milliseconds while chemical processes can take hours to change their states).

The architectural properties of the physical layer depend on the numbers of measurements and actuation signals, and the specific level of the architectural hierarchy. For example, the massive scale of the electric power grid makes it impossible for a single authority to autonomously control the grid; as such, the power grid is controlled as a federated system, where each component subsystem is controlled by a single authority, but in a decentralized manner. In contrast, the water level in a tank can be maintained via centralized authority, but in a decentralized manner. In contrast, the water level in a tank can be maintained via centralized authority, but in a decentralized manner. In contrast, the water level in a tank can be maintained via centralized authority, but in a decentralized manner.

3.2 Control layer

The control layer takes sensor measurements $y(t)$ and produces actuator signals $u(t)$ that manipulate the system state $x(t)$. The control action $u(t)$ is typically a function of the system state, i.e., $u(t) = h(x(t))$. The system state is computed by an observer $k$ as an estimated state $\hat{x}(t)$ based on the sensor measurements. The observer usually has knowledge of the system parameters $f(\cdot)$ and $g(\cdot)$, the statistics of the noise signals $w(t)$ and $v(t)$, and has access to the control input and the sensed variables.

Controllability and observability are properties that do not directly affect the output, but are useful for analyzing control algorithms. These notions were originally defined for linear, time-invariant systems, but they have intuitive interpretations for general control systems. Note that observability and controllability are mathematical duals.

- Controllability: This is the ability of an external input to drive the internal system state from an initial state to another state in a finite time interval. A similar notion is output controllability, which describes the ability of an external input to drive the output from an initial condition to a final condition in a finite time interval. A system is controllable if it is possible to execute a control algorithm that can make the system stable.

- Observability: This is a measure of how well the dynamic behavior of a system (i.e., internal system states) can be inferred based on information about its external outputs (i.e., sensor measurements). In practice, a system is observable if it is possible to create an observer (also known as a state estimator) that can accurately track the system state given sensor measurements.

Each control architecture has unique security considerations. For example, in a linear system, an attacker can have a simple attack strategy to destabilize the system; however, in a nonlinear system, an attacker might be able to explore complex dynamics such as finding a resonant frequency of the system. Similarly, a centralized architecture may have a
single point of failure that an adversary could target, while a distributed architecture would allow an adversary to compromise a subset of agents, and any security model would have to incorporate the interdependencies of these agents and the overall system.

3.3. Cyber layer

The cyber layer is where the control aspects of a cyber-physical system are implemented as computerized control systems. This is typically realized through special-purpose hardware platforms. The hardware interfaces with electrical, communications and mechanical subsystems and must be optimized to satisfy real-time computing constraints.

The Architecture Analysis and Design Language (AADL) is used to model the cyber layer because it can capture the hardware and software architectures of embedded real-time systems. At a high level, AADL decomposes a system into hardware components and software components that correspond to the cyber components of a cyber-physical system [5]. The execution platform comprises hardware components such as devices, memory, buses and processors. These components represent the physical aspects of the system.

Software components, which model code execution, include processes, threads, data and subprograms needed to support execution. The properties of software components as well as hardware components are defined in the AADL standard [6].

AADL provides modeling constructs for expressing a number of descriptive and runtime properties of the software and execution platform (hardware) components of a system. For example, the execution properties that can be assigned to threads include timing, dispatch protocols, memory size and processor binding.

Of special interest with regard to the cyber layer is how confidentiality, integrity and availability attacks can negatively impact execution, access and other cyber system attributes. In the context of AADL, confidentiality attacks involve unauthorized access to memory and data while integrity attacks involve tampering with memory and data. Availability attacks perturb the execution characteristics of processes and threads, such as disrupting transmission and the data transfer characteristics of buses or delaying memory accesses.

By mapping the cyber layer to the control layer, this model provides the ability to evaluate how cyber attacks can perturb the control system and ultimately the physical system. Fig. 3 shows how various components of the AADL model of a system are mapped to the control layer. AADL subprograms are used to execute various observer and control action functions, while control variables are modeled as AADL data objects. These components represent the physical aspects of the system. The AADL model representation of the cyber layer is denoted by $S_{cyb}$.

3.4. Characterizing the attack space

The Cartesian product of cyber security properties, control properties, physical outcomes and extent characterizes the attack space of a cyber-physical system. The Cartesian product of these sets $AS = SECPROP \times CNTPROP \times EXTENT \times OUTCOME$ (3) enumerates all possible (but not necessarily viable) combinations of how a cyber attack vector may perturb control properties. The three sets are defined as follows:

- **SECPROP**: Cyber security properties, i.e., {confidentiality, integrity, availability}.
- **CNTPROP**: Control-theoretic properties, i.e., {controllability, observability}.
- **EXTENT**: Extent of the control properties gained by the attacker, i.e., {partial, full}.
- **OUTCOME**: Physical outcomes in terms of system effects, i.e., {stability, safety, efficiency}.

4. Cyber-physical kill-chain

This section presents a cyber-physical kill-chain that assists defenders in understanding and describing the cyber-physical system attack lifecycle and how attackers can carry out attacks. Fig. 4 compares the cyber-physical kill-chain with the traditional cyber kill-chain. The cyber-physical kill-chain specifies how the various kill-chain phases map to the cyber, control and physical properties of the targeted system.

4.1. Reconnaissance phase

The reconnaissance phase starts with a target system $S$ about which the attacker must gain information in order to perform the attack. Information about the system can be acquired by various means, including collecting publicly-available data, performing cyber exploitations or conducting physical observations. Reconnaissance can be modeled in each of the three layers of the framework.
4.1.1. Cyber reconnaissance
Reconnaissance in the cyber layer focuses on the operational and security attributes of a cyber system. Operational attributes describe how a system communicates and processes information to support the activities of the cyber-physical system. Security attributes describe the vulnerabilities of a system to attack and inform an attacker about the types of attacks that may be possible against the system. Operational attributes include the hardware and software platforms, communications protocols and processes. Security attributes include hardware and software vulnerabilities, cryptographic algorithms, authentication mechanisms and attack detection capabilities.

4.1.2. Control reconnaissance
Reconnaissance in the control layer seeks to understand the control model of the targeted system. The goal is to acquire information about the control algorithms, sensors and actuators and how they are used to monitor and control the cyber-physical system. This information is necessary to understand how a cyber attack can perturb the critical system based on information about the system state that may be available to the attacker and the attacker’s ability to influence the physical operations of the system.

Information about sensors and actuators is very valuable, especially the types of devices, the degree of redundancy and how the data provided can be used to infer the system state. The control algorithms, topology and dimensions are of interest as well; these provide insights into the attacks that are possible, how they might progress and if the system has synchronization constraints in the case of multiple interactive control loops (that can lead to cascading effects). If a system has multiple controllers, then understanding controller coordination is important to the attacker as well. This information allows the attacker to construct a view of the control algorithm that describes the behavior of the controller output \( u(t) \) given sensor values \( y(t) \).

4.1.3. Physical reconnaissance
Reconnaissance in the physical layer focuses on understanding the physical process \( f(t), g() \) that is being controlled. In order to achieve a particular objective, an attacker must understand how the physical process is influenced by various control actions as well as the physical properties and dynamics that govern the process. Also, an attacker must have a strong understanding of the stability, safety, efficiency and resource constraints. This enables the attacker to determine if the system is inherently stable, the degree of stability, the safety bounds and the resources required by the system to fulfill its mission at a given level of efficiency.

4.2. Weaponization phase
The goal of the weaponization phase is to develop a weapon based on the information about the three layers obtained in the earlier reconnaissance phase. The weapon designers would have to decide on the specific vulnerabilities to be exploited in the cyber and control layers based on the physical properties that need to be perturbed. They need to determine if the goal is to degrade some aspect of the physical operation of the cyber-physical system or to disrupt, or even completely destroy, the cyber-physical system. Other considerations include the quality and quantity of information about the internal workings of the cyber and control layers, the degree of precision of the physical effects and the available weapon delivery mechanisms. These considerations collectively determine the specific exploits at the cyber and control layers.

A weapon \( W \) employs one or more cyber exploitation vectors. Each cyber exploitation vector is an action that exploits a vulnerability in pursuit of a target at the cyber layer, with the end goal of perturbing the control and physical layers. Each cyber exploitation vector has a cyber impact that exploits a vulnerability in pursuit of a target at the cyber layer, resulting in a new control function for the system. An individual cyber exploitation vector is a tuple \( CEV = \{cybexp, cybtar, secprop, cntexp\} \) whose elements are defined as follows:

- **Cyber exploit (cybexp)**: The cyber exploit is the part of the weapon that provides the attacker with the ability to manipulate the operation of the cyber layer. The cyber exploit leverages a system weakness (e.g., software vulnerability or inadequate authentication) to gain influence over the cyber layer. Sources such as the Common Attack Pattern Enumeration and Classification (CAPEC) provide a comprehensive set of possible cyber exploits [16].

- **Cyber Target (cybtar)**: The cyber target can be any cyber component in the system architecture (e.g., bus, message, process or data) that contains an exploitable vulnerability. This ultimately determines how the attacker can impact the control of the system. Although the cyber exploit is applied to the cyber target, the particular target and vulnerability constrain the affected security property.
4.3. Delivery phase

The delivery phase encompasses all the actions that move a weapon from an attacker to a target. Examples include wireless transmission, supply chain infiltration or physical access to a dataport. A weapon may first have to be spread in unrelated networks in order to gain access to the targeted cyber-physical system. This spreading, and any cyber exploits required to achieve it, occurs during the delivery phase. The delivery phase ends and the subsequent cyber execution phase begins after the weapon begins to act directly on the cyber-physical system.

4.4. Cyber execution phase

The cyber execution phase focuses on how the cyber layer of the system $S_{cyb}$ is degraded by W.cybexp, W.cybart and W.secprom. This phase results in a modified cyber layer $S'_{cyb}$ in which the components have degraded confidentiality, integrity or availability (CIA) properties based on cybexp, cybart and secprom.

As mentioned in Section 3.3, the cyber layer can be described using an AADL model of the cyber components comprising the cyber-physical system. A cyber weapon contains an exploit against a particular target component that has a confidentiality, integrity or availability impact on the cyber layer. This impact leads to some effect on the control layer. Therefore, the cyber exploitation phase is modeled as a function that takes the current AADL model $S_{cyb}$ and the weapon $W$, and computes the impact of the weapon on the system model by degrading its confidentiality, availability or integrity properties to produce the attacked system model $S'_{cyb}$.

Algorithm 1. Cyber impact algorithm.

```
Algorithm 1. SystemImpact(System $S_{cyb}$, Weapon $W$)
List(Component c, CIA – effect e; totalImpacts = []
for all component, $c \in S_{cyb}$ do
    totalImpacts.addCyberImpact(c, W)
end for
return $S_{cyb} \rightarrow S'_{cyb} \setminus$totalImpacts
end function

function CyberImpact(Weapon $W$)
List( Component c, CIA – effect e; impacts = []
for all CEV, $c \in W$ do
    if $c = CEV, cybart$ then
        impacts.add(isVulnerable(c, CEV, cybexp))
    else if $c$ has Subcomponents then
        for all $c_{j} \in c$ do
            impacts.add(CyberImpact($c_{j}, W$))
        end for
    end if
end for
return impacts
end function
```

Algorithm 1 demonstrates how the cyber exploitation impact is computed using the system model. The CyberImpact function recursively explores all the subcomponents of the system to determine the cyber components that are targeted. When a target is found, function IsVulnerable determines the impact of the cyber exploit, taking into consideration the attack mitigation mechanisms that may be implemented for the component. For example, an attack against an encrypted bus enables an attacker to read data on the bus, but confidentiality is preserved because of encryption as long as the attacker does not have the decryption key. The impacts are then returned to the parent component and the full impact list is eventually returned to SystemImpact. The function SystemImpact, after calling function CyberImpact in the top-level AADL component (system itself), uses the resulting impact list to generate a post-exploit cyber model $S'_{cyb}$. The IsVulnerable function is not defined here.

### Table 1: Examples of cyber exploitation vectors.

<table>
<thead>
<tr>
<th>Cyber exploit</th>
<th>Cyber target</th>
<th>Security property</th>
<th>Control exploit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software exploit</td>
<td>Process, thread</td>
<td>[Conf., Int., Avail.]</td>
<td>$c(t, u, y, \dot{x}) \rightarrow (u_0, y_0, \dot{x}_0)$</td>
</tr>
<tr>
<td>Man-in-the-middle</td>
<td>Bus, data</td>
<td>[Conf., Int., Avail.]</td>
<td>$c(t, u, y, \dot{x}) \rightarrow (u_0, y_0, \dot{x}_0)$</td>
</tr>
<tr>
<td>Spoofing</td>
<td>Bus, data</td>
<td>[Int.]</td>
<td>$c(t) \rightarrow (u_0, y_0, \dot{x}_0)$</td>
</tr>
<tr>
<td>Communication flooding</td>
<td>Bus, data</td>
<td>[Avail.]</td>
<td>$c(t) \rightarrow (0.1)$</td>
</tr>
<tr>
<td>CPU consumption</td>
<td>Processor, process, thread</td>
<td>[Avail.]</td>
<td>$c(t) \rightarrow (0.1)$</td>
</tr>
</tbody>
</table>
because it is specific to the security protections available to the system (e.g., encryption and encapsulation) that may protect subcomponents although the parent component was successfully attacked.

4.5. Control perturbation phase

The input to the control perturbation phase is the degraded cyber model \( S_{\text{cyb}} \) (output of the previous phase) and the control exploit of the cyber weapon \( \text{cntexp} \). This phase attempts to assess the impact of a cyber attack based on how observable and controllable the system is to the defender and to the attacker, and then applies the control exploit \( \text{cntexp} \) to \( S_{\text{cyb}} \). This results in the attacker-manipulated control layer \( S_{\text{cnt}} \).

During this phase, the weapon begins to have an effect on the control layer by degrading integrity or availability in the cyber layer, which influences the operation of the control loop in the control layer. As demonstrated in Fig. 5, an attack that degrades the data and subprograms in the cyber model enables the attacker to manipulate the variables and functions defined in the control model. Additionally, this phase describes how an attack that degrades confidentiality, integrity or availability enables the attacker to gain observability and controllability, and whether the attack also degrades the defender’s observability and controllability.

As explained above, the security properties perturbed by a cyber attack determine the controllability and observability of the system to the attacker. However, the weapon’s control function \( \iota \) determines exactly how the control layer is manipulated by an attack to achieve its objective. An attack that targets a subprogram in \( S_{\text{cyb}} \) is modeled by encapsulating the control function in the subprogram within the function \( \iota \) and adding the control logic specified in the weapon’s control function. Similarly, an attack that targets data is modeled by encapsulating the data in function \( \iota \).

In this setting, a forced (i.e., non-homogeneous) system of linear differential equations is considered:

\[
\dot{x}(t) = Ax(t) + Bu(t) \quad y(t) = Cx(t) + Du(t)
\]

where \( x(t) \in \mathbb{R}^n \) is a vector of physical quantities representing the system state at time \( t \), \( u(t) \in \mathbb{R}^p \) is the control input at time \( t \), \( y(t) \in \mathbb{R}^q \) is a vector of sensor measurements at time \( t \), and \( A, B, C \) and \( D \) are matrices representing the system dynamics.

Assume that an attacker has access to a subset of control signals \( u_a \) (it does not matter which signals). Because the ordering of the vector \( x \) is arbitrary, it is always possible to partition the system as

\[
\begin{align*}
\dot{x} &= Ax + B_1 u_a + B_2 u_o \\
\dot{x} &= Ax + B_3 u_a + B_4 u_o
\end{align*}
\]

where \( B_1 = [B_{11} B_{12}]^T \), \( B_3 = [B_{12} B_{22}] \), \( u_o \) represents the first \( s \) rows of the vector \( u \) and \( u_a \) the remaining rows (which are assumed to be under the control of the adversary).

This model can be used to answer questions about the vulnerability of a system under attack. In particular, the cyber-physical system security axioms described in [1] are examined.

4.5.1. CPS security axiom #1: integrity – controllability

While obtaining a general result for controllability of a general system under attack is difficult, a common control algorithm that is representative of the majority of practical control systems is one that uses a state-feedback control signal, where the original control signal (control signal without an attack) is given by \( u = -Kx \).

Given that a state-feedback controller is used, if the system is under attack as described by Eq. (5), only a portion of the control signals maintain their integrity. Thus

\[
u = \begin{bmatrix} u_a \\ u_o \end{bmatrix} = \begin{bmatrix} K_s x \\ u_o \end{bmatrix}
\]

where \( K_s \) denotes the first \( s \) rows of the original matrix \( K \).

With state-feedback, the dynamical system under attack becomes

\[
\dot{x} = Ax + B_3 K_s x + B_4 u_o, \quad \dot{x} = Ax + B_4 u_o
\]

where \( A_s = (A + B_3 K_s) \). Therefore, an attacker can obtain complete state controllability of the system iff

\[
\text{rank}(B_4 A_s B_o \cdots A_n^{n-1} B_3) = n
\]

In some other cases, the attacker can disrupt system controllability via a simple state-feedback attack with an appropriate attack gain matrix \( K_a \). Replacing this state-feedback attack in Eq. (5) yields

\[
\dot{x} = Ax + B_3 K_a x + B_4 u_o, \quad \dot{x} = Ax + B_4 u_o
\]

In this setting, an attacker can completely disrupt system controllability via state-feedback integrity attacks iff

\[
\text{rank}(B_4 A_s B_o \cdots A_n^{n-1} B_3) < n
\]

4.5.2. CPS security axiom #2: integrity – observability

This section shows that, when the system defender uses state-feedback control, the attacker can use sensor measurements to reduce the problem to that of Eq. (5).

---

**Fig. 5 - Control impacts due to cyber attacks.**

---

Control Properties

<table>
<thead>
<tr>
<th>Observability</th>
<th>Controllability</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attacker Gains</strong></td>
<td>None</td>
</tr>
<tr>
<td><strong>Defender Loses</strong></td>
<td><strong>Attacker Gains</strong>, <strong>Defender Loses</strong></td>
</tr>
<tr>
<td><strong>Defender Gains</strong></td>
<td><strong>Defender Loses</strong></td>
</tr>
</tbody>
</table>
In a general linear, time-invariant system, the sensor measurements $y(t)$ are functions of the state and inputs. In order to deal with an attacker who compromises the integrity of the sensors, it is necessary to consider observability and the design of state estimators. This work assumes that $y(t) = x(t)$ (a valid assumption in many cases) and leaves the more general problem of attacks on system observability for future research.

Again, in order to obtain tractable results, it is assumed that the defender uses a state-feedback control law, i.e., $u(t) = Kx(t)$. Therefore, if the sensors are not compromised, the evolution of the system follows the equation:

$$\dot{x} = (A + BK)x$$  \hspace{1cm} (11)

However, if some sensors are compromised $y_d(t) = x_d(t)$, then the evolution of the dynamical system is given by

$$\dot{x} = A\dot{x} + \begin{bmatrix} B_{11}K_{11} & B_{12}K_{12} \\ B_{21}K_{21} & B_{22}K_{22} \end{bmatrix} x_d$$

where the fake sensor measurements $x_d(t)$ become, in practice, the control signal of the attacker $u_d(t)$.

Note that Eq. (12) has the same form as Eq. (5). However, in this case, the following equation holds:

$$A_d = A + \begin{bmatrix} B_{11}K_{11} & 0 \\ B_{21}K_{21} & 0 \end{bmatrix}, \quad B_d = \begin{bmatrix} B_{12}K_{12} \\ B_{22}K_{22} \end{bmatrix} x_d$$

where the fake sensor measurements $x_d(t)$ become, in practice, the control signal of the attacker $u_d(t)$.

### 4.6. Physical objective realization phase

This phase deals with the physical impact that an attacker intends to have on a system. In a traditional information technology system, the objective is generally to degrade system confidentiality, integrity and/or availability. Potential attack objectives are well defined in the military sphere and include deny, destroy, degrade and disrupt [11]. The objectives with regard to a cyber-physical system differ because they result in some impact to the physical system or $x$ as defined in Section 3.1. This section identifies multiple cyber-physical system objectives and specifies their impacts on the system state.

Let $t_f$ and $t_f$ denote the starting and ending times, respectively, let $X_s$ and $X_d$ denote the attacker’s and defender’s objective states, respectively, and let $\epsilon$ denote a small allowable error in the objective. As described above, the control system objectives are system stability, safety and efficiency. Hence, the following four physical attack objectives are considered:

- **Usurp**: Take complete control of the system such that an attacker can direct the system to the intended trajectory $X_s$ within some value $\epsilon$. This objective is defined as
  $$|x(t_f) - X_s| < \epsilon$$  \hspace{1cm} (14)

- **Destroy**: Damage the system to ensure that it cannot function properly because it becomes unstable or operates outside of the safe bounds. This objective is defined as
  $$x_{safe\_min} < x(t) < x_{safe\_max} \land x(t) \text{ is unstable}$$  \hspace{1cm} (15)

- **Disrupt**: Prevent the system from achieving its intended terminal state (e.g., mission) $X_d$. This objective is defined as
  $$|x(t_f) - X_d| > \epsilon$$  \hspace{1cm} (16)

- **Degrade**: Prevent the system from achieving its intended terminal state $X_d$ with optimal cost $\text{cost}(U)$. This objective is defined as
  $$|x(t_f) - X_d| < \epsilon \land \text{cost}(U) > \text{cost}(U)$$  \hspace{1cm} (17)

The properties specified above determine whether or not an attack would successfully realize the associated physical objective. For example, a less stable system may be more likely to meet the destroy objective than a more stable system. The following section demonstrates how different weapons can achieve the objectives.

### 5. Kill-chain evaluation

This section conducts a test case analysis of the cyber-physical system kill-chain. The model used for the analysis is based on an unmanned aerial system simulation developed by Goppert et al. [7]. Various cyber weapons are identified, applied to the test system and their physical impacts are then analyzed.

#### 5.1. Test system

This section describes the unmanned aerial system model used in the evaluation.

##### 5.1.1. Physical layer

A physical model was created of a MultiPlex Easy Star, a recreational unmanned aerial system with a small airframe. The control dynamics were modeled and simulated using the JSBSim flight dynamics engine [12], which was interconnected to a control system simulation component that received actuation signals and implemented the resulting changes in the physical simulation. The updated state information was sent back to the control simulation in order to compute the corresponding control actions. The actuators used by the unmanned aerial system included a throttle, rudder and elevators. The sensors included a GPS, inertial navigation system and magnetometer.

##### 5.1.2. Control layer

The control system for the unmanned aerial system was simulated in ScicosLab [20]. The state of the unmanned aerial system was obtained from JSBSim and was sent to an extended Kalman filter that computed the estimated unmanned aerial system state. The estimated state was sent to a waypoint guidance controller that determined the appropriate reference signal for the unmanned aerial system and computed the control error based on its current state. The error was then sent to a feedback control function that computed the
appropriate actuator signal using a separate PID controller for each of the three actuators mentioned above.

5.1.3 Cyber layer
The test system defined in [7] provided the control algorithms and physical system specifications, but did not provide the cyber components. Therefore, the AADL model shown in Fig. 6 was developed for the unmanned aerial system. Two processes were used to execute the unmanned aerial system, Nav and Fdbk. Both the processes had threads that executed the subprograms of the control algorithms discussed above. Additionally, both processes were executed on the same processor and resided in the same memory component. A bus was assumed to transmit messages between the components.

5.1.4 Objectives
In the simulation, the unmanned aerial system had a mission goal to reach a specified waypoint. The unmanned aerial system began the simulation at latitude–longitude (0.0, 0.0) and an altitude of 1000 feet. The objective was to fly to the waypoint at latitude–longitude (1.2 x 10^−4, 1.2 x 10^−4) with the same altitude of 1000 feet. The unmanned aerial system was also required to operate within a defined flight envelope to ensure safe operation.

The flight envelope was defined as |a| < λ, where a is the unmanned aerial system angular velocity. The cost of operating the unmanned aerial system was assumed to be equal to its fuel consumption cost. The unmanned aerial system attempted to meet its objective while minimizing fuel consumption.

5.2 Simulation results
The analysis of the kill chain examines various cyber-physical system weapons and evaluates their control and physical impacts on the test system. The weapons attempted to achieve each of the physical objectives identified in Section 4.6. In the case of the usurp objective, the attacker’s goal was to drive the unmanned aerial system to latitude–longitude (0.0, 1.2 x 10^−4) at 1500 feet instead of the defender’s waypoint (1.2 x 10^−4, 1.2 x 10^−4) at 1000 feet.

Fig. 7 presents the results obtained for each weapon. Attacks with the usurp, disrupt and degrade objectives are demonstrated by the latitude–longitude values of the flight trajectories and are shown as blue lines. Note that (0, 0) is the origin and the crosshairs indicate the intended waypoint. Fig. 7(a) shows a base run with no attack. Attacks with the destroy objective are graphed with their yaw, roll and pitch as a function of time. Fig. 7(b) also shows a base run with no attack.

Table 2 lists the various weapons launched against the unmanned aerial system and their physical objectives. Three weapons are evaluated – the first consumes CPU cycles and the other two are software exploits that target different unmanned aerial system processes.

The CPU consumption weapon prevents control algorithms in the unmanned aerial system from executing. However, because this attack only impacts availability, the attacker does not gain any controllability or observability of the system as demonstrated by the control impact column in Table 2. The control exploit is defined as

\[ u(t) = u(t) c(t) \]  \hspace{1cm} (18)

where c(t) is defined as

\[ c(t) = \begin{cases} 0 & \text{if } t_s < t < t_e \\ 1 & \text{otherwise} \end{cases} \]  \hspace{1cm} (19)

where t_s and t_e are the starting and ending times of the attack, respectively. The extent of the attack determines whether the attack results in a degrade, disrupt or destroy objective (Fig. 7 (c), (d) and (e)). A usurp attack is not possible because the attack lacks controllability and observability.

The second weapon targets the Nav process in the unmanned aerial system. The Nav process contains data about the current estimated state \( \hat{x}(t) \) and the desired waypoint \( X_d \). Since the software exploit on the Nav process degrades confidentiality, the attack can achieve full observability of the system. Additionally, by degrading the integrity of \( \hat{x}(t) \rightarrow \hat{x}(t) \), the attack can gain full controllability over the system. Thus, c(t, \( \hat{x}(t) \)) is defined as

\[ \hat{x}(t) = \hat{x}(t) + (X_d - X_s) \alpha \]  \hspace{1cm} (20)

where the difference between the defender’s and attacker’s waypoints is added to the estimated state, causing the estimated state and the actual state to diverge until the unmanned aerial system ends at the attacker’s objective state as demonstrated in...
Fig. 7(f). Note that the variable $\alpha$ is introduced to control how quickly the attack scales and is initially defined as

$$\alpha = \frac{t}{t_f - t_0}$$  \hspace{1cm} (21)

so that the scale of the attack gradually increases.

The degrade and disrupt objectives can be achieved by utilizing the same control exploit as for the usurp objective, but for a shorter period of time (Fig. 7(g) and (h)). Note that these attack objectives could also be carried out with partial observability and controllability. A destroy attack can be performed by setting $\alpha = 1$ to prevent gradual scaling, causing the unmanned aerial system to over-respond to the attack and become unstable (Fig. 7(i)).

The third weapon targets the Fdbk process, which does not have any state information, so the attacker does not have observability of the current system state. However, the attacker has full controllability because he can manipulate the integrity.
of all the actuator signals. Consequently, the usurp objective cannot be achieved reliably because it requires the attacker to probabilistically guess the state of the unmanned aerial system. However, the degrade, disrupt and destroy objectives can be achieved as demonstrated in Fig. 7(j), (k) and (l). To perform these attacks, the control exploit $\alpha(t, u(t))$ is defined as
\[ u(t) = u(t) + \alpha u(t) \] (22)
which scales the actuator signal by $\alpha$ for some period of time. In the case of the degrade and disrupt attacks, $\alpha = 0.5$ was used, but with a longer attack time for the disrupt objective. The destroy attack used a large value of $\alpha = 0.8$.

6. Conclusions

Developing and implementing successful cyber-physical system attacks require careful attention to the subtleties of how the cyber, control and physical layers interact and influence each other as well as how the various attack stages perturb these three layers. This paper lays the groundwork for a much-needed systematic and unified framework for analyzing attacks on cyber-physical systems. The framework enables the holistic analysis of cyber-physical system security because attacks against these systems require the manipulation of cyber, control and physical system properties.

The framework also supports the comprehensive analysis of the robustness of a cyber-physical system to cyber attacks with different objectives. The first element of the framework is a multi-layered reference architecture of the cyber, control and physical layers of a cyber-physical system. Cyber security and control-theoretic properties are leveraged to help analyze the interactions between these layers. The second element is a meta-model of cyber attacks called the cyber-physical system kill-chain. The kill-chain provides a basis for the systematic study of how the various cyber attack steps and phases can perturb the three system layers and eventually impact physical operations. Each phase in the kill-chain demands varying degrees of knowledge, precision and capabilities on the part of an attacker. Furthermore, depending on the specific security properties exploited, various control-theoretic properties are modulated differently.

The case study involving an unmanned aerial system demonstrates the effectiveness of the framework. In particular, the unmanned aerial system was subjected to a number of cyber exploit vectors; the framework was then used to decompose and analyze the threats, and assess the physical impacts.

Future research will examine various architectural designs and mitigation strategies to counter the threats discovered using the framework. This research will contribute to the long-term goal of articulating design principles for engineering secure and resilient cyber-physical systems.

REFERENCES