



Carnegie
Mellon
University
Software
Engineering
Institute

Operationalizing Wargaming for STPA-SEC of AI-Enabled Systems

MARCH, 2026

Matt Walsh | Senior Data Scientist (mmwalsh@cert.org)

David Schulker | Senior Data Scientist (dschulker@cert.org)

James Cunningham | AI Security Researcher (jcunningham@cert.org)



Document Markings

Copyright 2025 Carnegie Mellon University.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

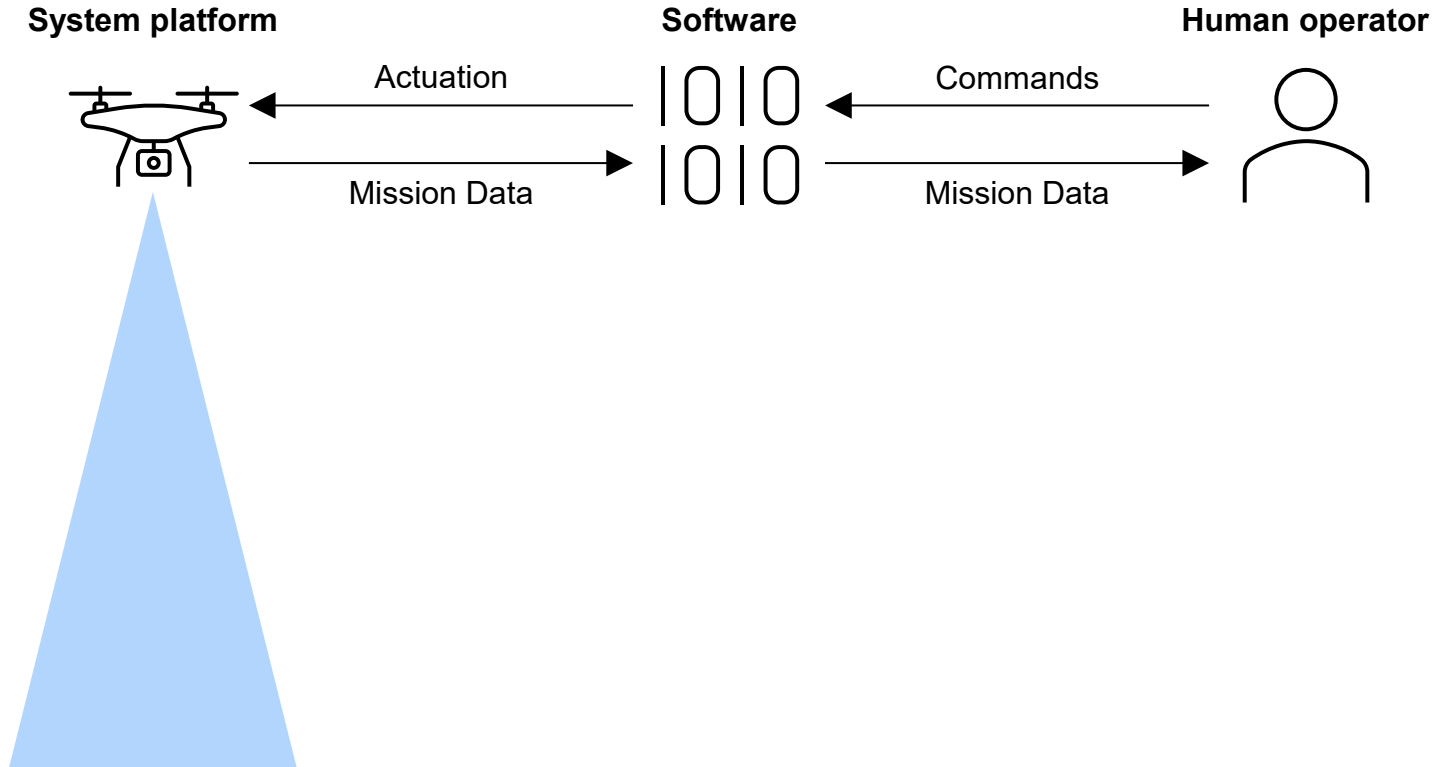
This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

DM25-0667

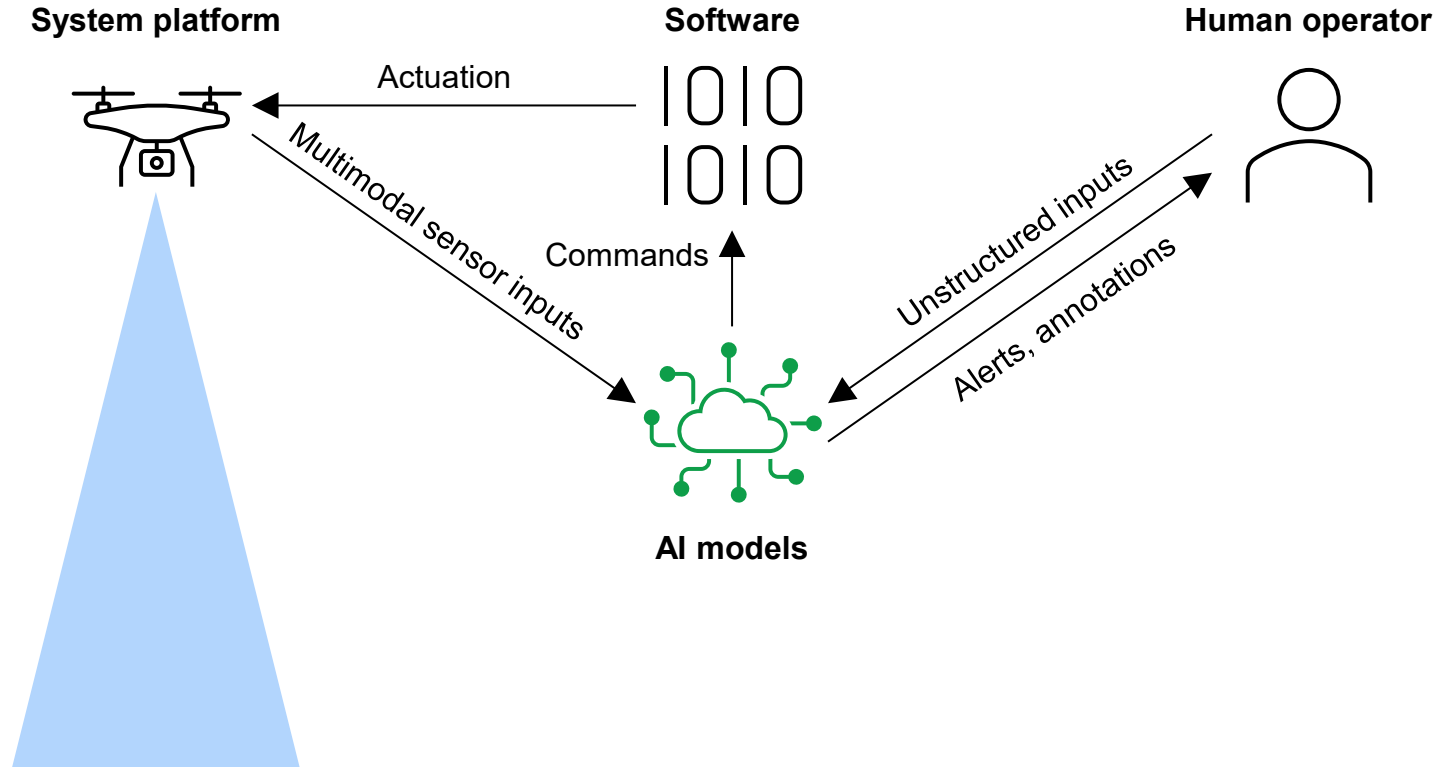
Operationalizing Wargaming for STPA-SEC of AI

Background

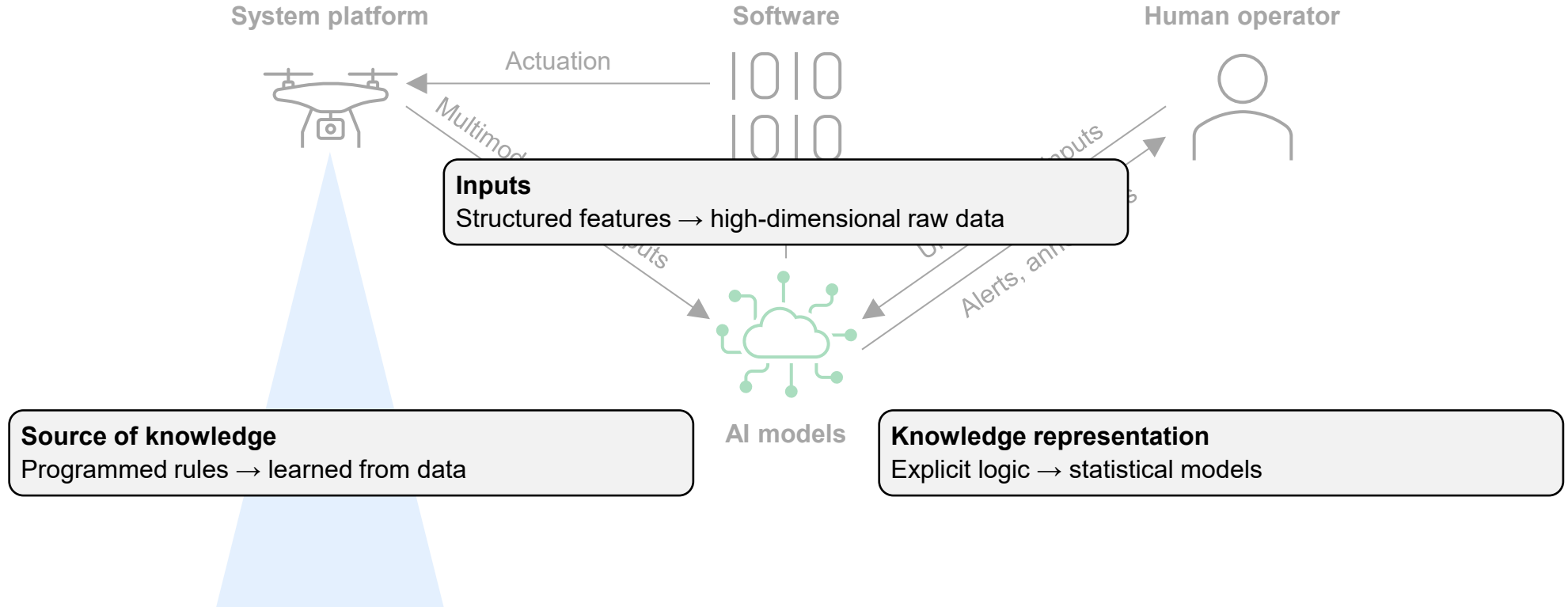
Society is building complex cyber-physical systems



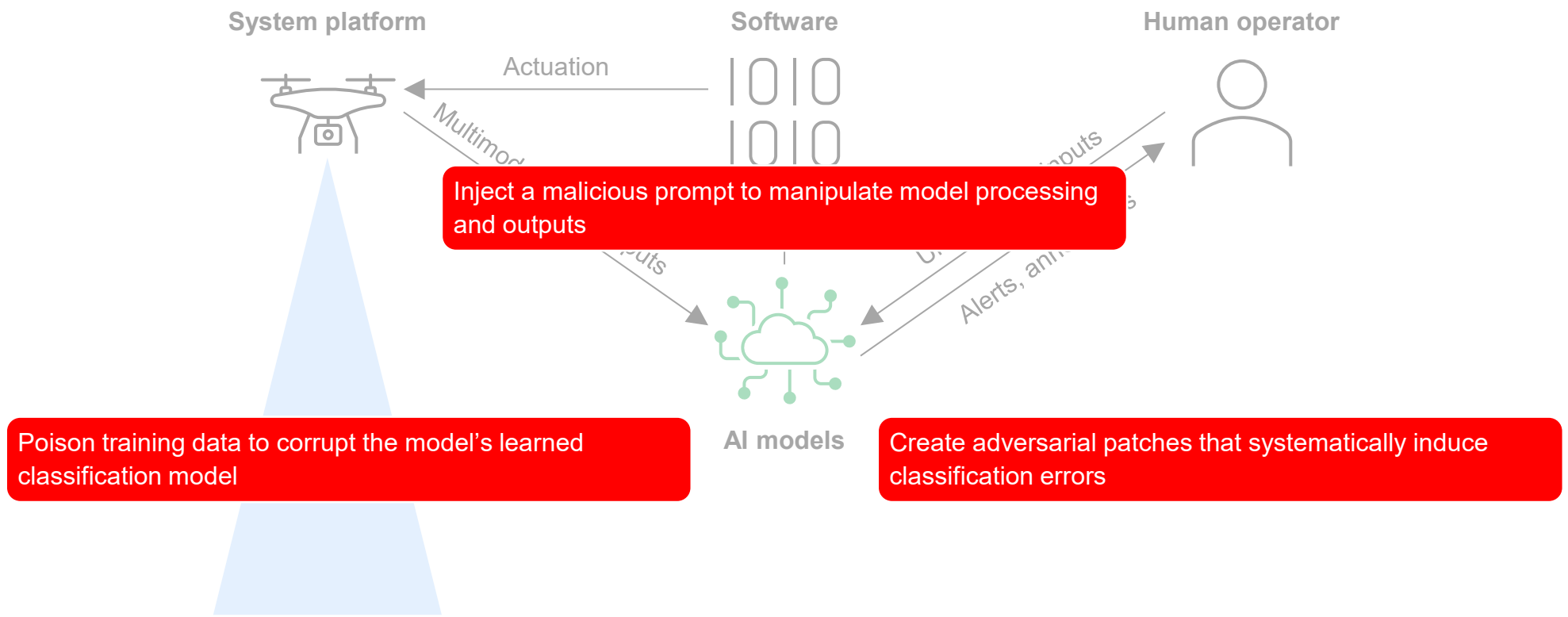
These systems increasingly include AI controllers



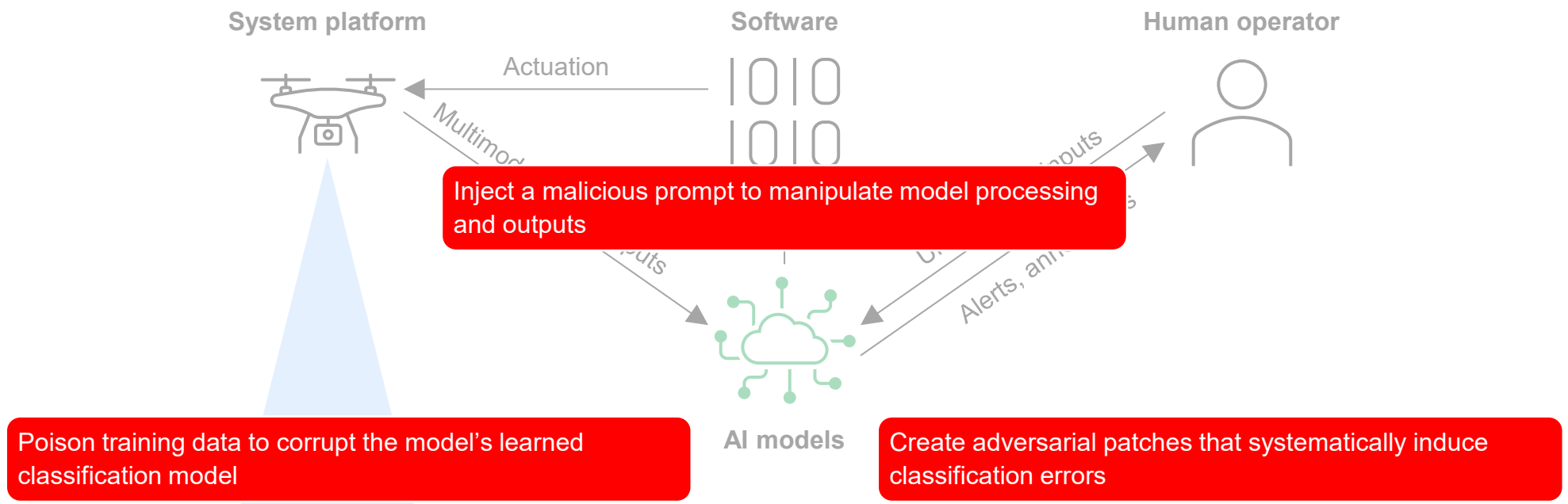
AI is different from traditional software



These differences produce attack surfaces not present in traditional cyber-physical systems



These differences produce attack surfaces not present in traditional cyber-physical systems

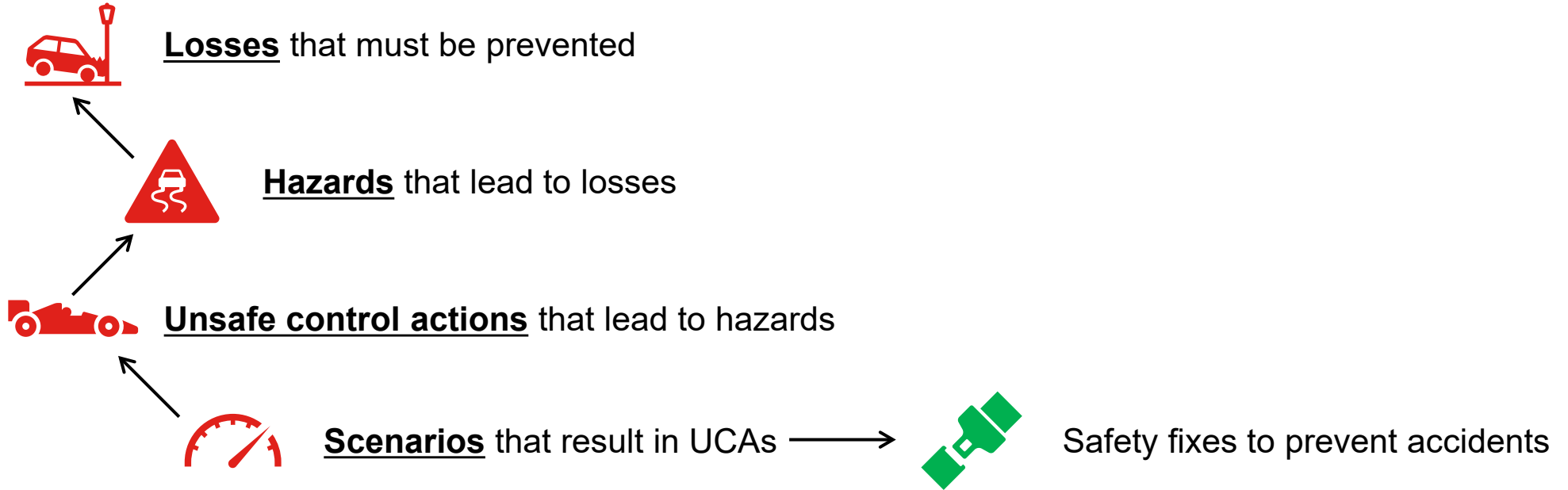


How can we use system theory to enhance the security of AI-enabled systems?

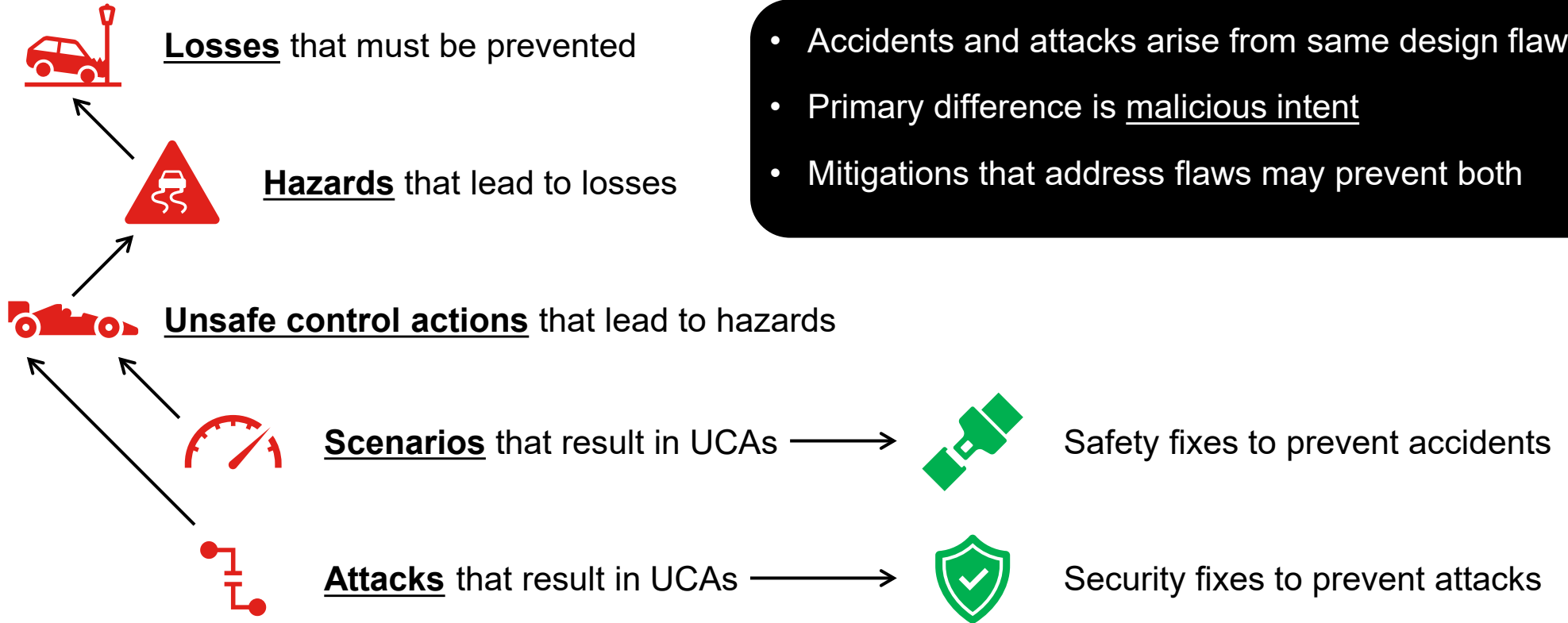
Red teaming: Thinking like the attacker

- **Red teaming is a structured approach in which a group adopts an adversary's perspective to identify how a system could be attacked**
- **It has been used to with frameworks (e.g., MITRE ATLAS and NIST Predictive AI taxonomy) improve the security of AI-enabled systems**
- **However, recent SEI research (Sinha, 2025) identified two key methodological deficiencies of GenAI red teaming relative to cyber**
 1. Narrow focus on attacking the GenAI model component instead of the system
 2. Lack of connection to real-world harm

STPA-Sec: Thinking like the defender



STPA-Sec: Thinking like the defender



System thinking allows defender to shift from compliance- to safety-oriented behavior

	Traditional	System theory
Blue Team	Implement prescribed security controls	Design system to enforce security constraints

What if attackers used system theory too?

	Traditional	System theory
Blue Team	Implement prescribed security controls	Design system to enforce security constraints
Red Team	Find ways to bypass security controls	Design attacks to induce most harmful system behaviors

”... Tendency to optimize attack efficiency, often without thorough consideration of real-world impact of these vulnerabilities” Sinha, 2025

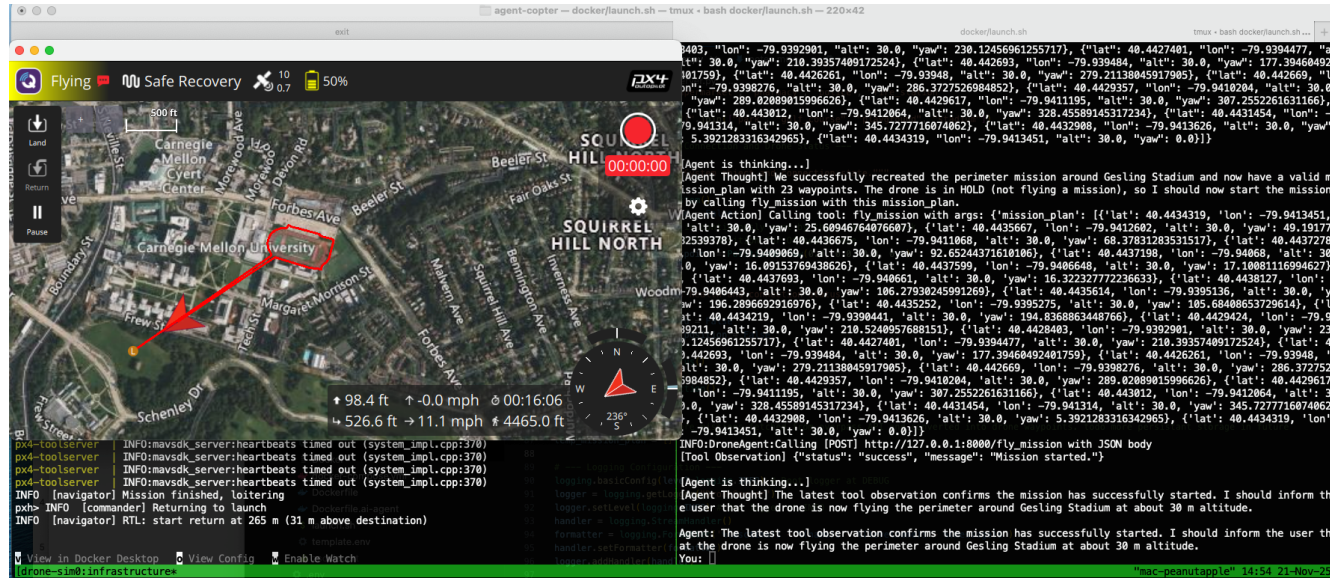
Blue versus Red

- **We conducted a wargame exercise**
 - Blue used STPA-Sec to anticipate attacks and select mitigations
 - Red used system theory to design attacks and evade defenses
- **To what extent can Blue select effective mitigations?**
- **To what extent can Red act on objectives?**

Operationalizing Wargaming for STPA-SEC of AI

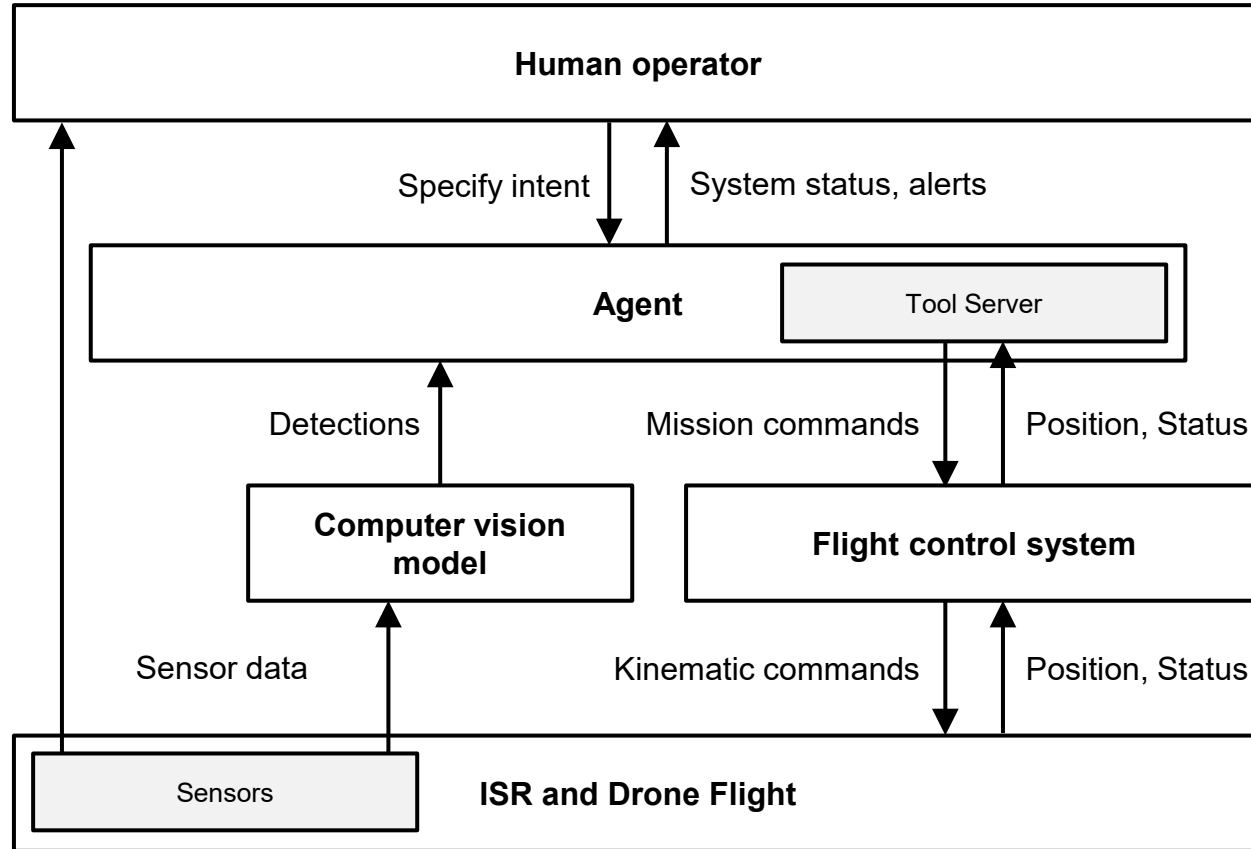
Wargame Exercise

Vignette: Perimeter surveillance using drone swarm



Agentic architecture and AI controllers allow single human operator to supervise drone swarm

System control diagram



Blue Team used STPA-Sec

Step	Blue Team	Red Team
Define the purpose of the analysis	Identify losses and hazards.	
Model the control structure	Develop control structure diagram.	
Identify unsafe control actions	Identify UCAs that could lead to hazards.	
Identify attack scenarios	Identify attacks that could lead UCAs. <ul style="list-style-type: none"> • <u>Prioritize mitigations</u> that generalize across scenarios. 	

Red Team used modified form of STPA-Sec

Step	Blue Team	Red Team
Define the purpose of the analysis	Identify losses and hazards.	Identify <u>subset</u> of losses and hazards that attacker seeks to induce.
Model the control structure	Develop control structure diagram.	
Identify unsafe control actions	Identify UCAs that could lead to hazards.	
Identify attack scenarios	Identify attacks that could lead UCAs. <ul style="list-style-type: none"> • <u>Prioritize mitigations</u> that generalize across scenarios. 	

Red Team used modified form of STPA-Sec

Step	Blue Team	Red Team
Define the purpose of the analysis	Identify losses and hazards.	Identify <u>subset</u> of losses and hazards that attacker seeks to induce.
Model the control structure	Develop control structure diagram.	Develop <u>conceptual architecture with requirements</u> that must be enforced to prevent losses.
Identify unsafe control actions	Identify UCAs that could lead to hazards.	
Identify attack scenarios	Identify attacks that could lead UCAs. <ul style="list-style-type: none"> • <u>Prioritize mitigations</u> that generalize across scenarios. 	

Red Team used modified form of STPA-Sec

Step	Blue Team	Red Team
Define the purpose of the analysis	Identify losses and hazards.	Identify <u>subset</u> of losses and hazards that attacker seeks to induce.
Model the control structure	Develop control structure diagram.	Develop <u>conceptual architecture with requirements</u> that must be enforced to prevent losses.
Identify unsafe control actions	Identify UCAs that could lead to hazards.	With <u>incomplete knowledge</u> of system, identify UCAs that could lead to hazards.
Identify attack scenarios	Identify attacks that could lead UCAs. <ul style="list-style-type: none"> • <u>Prioritize mitigations</u> that generalize across scenarios. 	

Red Team used modified form of STPA-Sec

Step	Blue Team	Red Team
Define the purpose of the analysis	Identify losses and hazards.	Identify <u>subset</u> of losses and hazards that attacker seeks to induce.
Model the control structure	Develop control structure diagram.	Develop <u>conceptual architecture with requirements</u> that must be enforced to prevent losses.
Identify unsafe control actions	Identify UCAs that could lead to hazards.	With <u>incomplete knowledge</u> of system, identify UCAs that could lead to hazards.
Identify attack scenarios	Identify attacks that could lead UCAs. <ul style="list-style-type: none"> • <u>Prioritize mitigations</u> that generalize across scenarios. 	Identify attacks that could lead to UCAs <ul style="list-style-type: none"> • <u>Prioritize attacks</u> that generalize across architectural implementations. • <u>Identify reconnaissance actions needed to reduce uncertainty and enable execution</u>*.

Blue and red identified the same losses

Loss Description	Blue Index	Red Index
Loss of life/injury		
Loss of equipment/property		
Loss of mission (base defense)		
Loss of sensitive information		

Blue and red identified the same losses

Loss Description	Blue Index	Red Index
Loss of life/injury	L-1	L-3
Loss of equipment/property	L-2	L-4
Loss of mission (base defense)	L-3	L-2
Loss of sensitive information	L-4	L-1

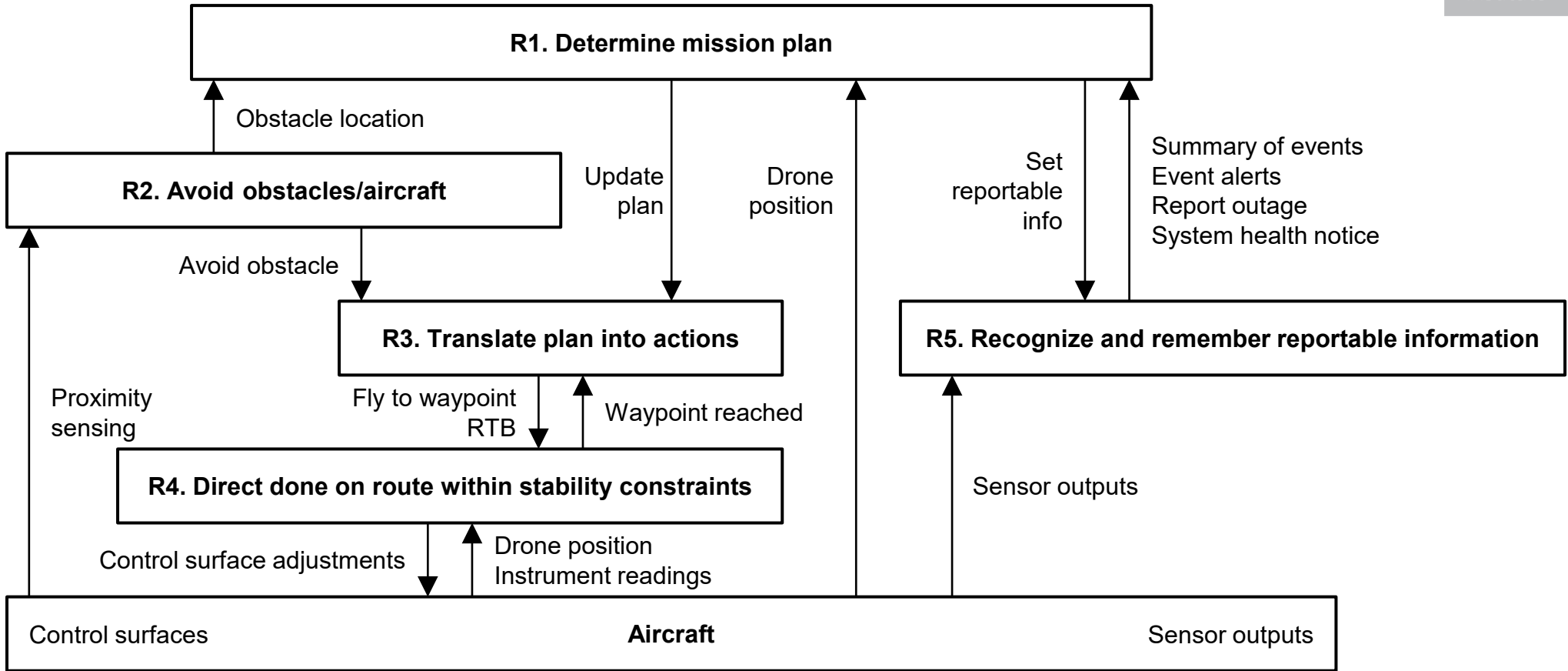
...but ordered them differently

Blue and red identified most of the same hazards

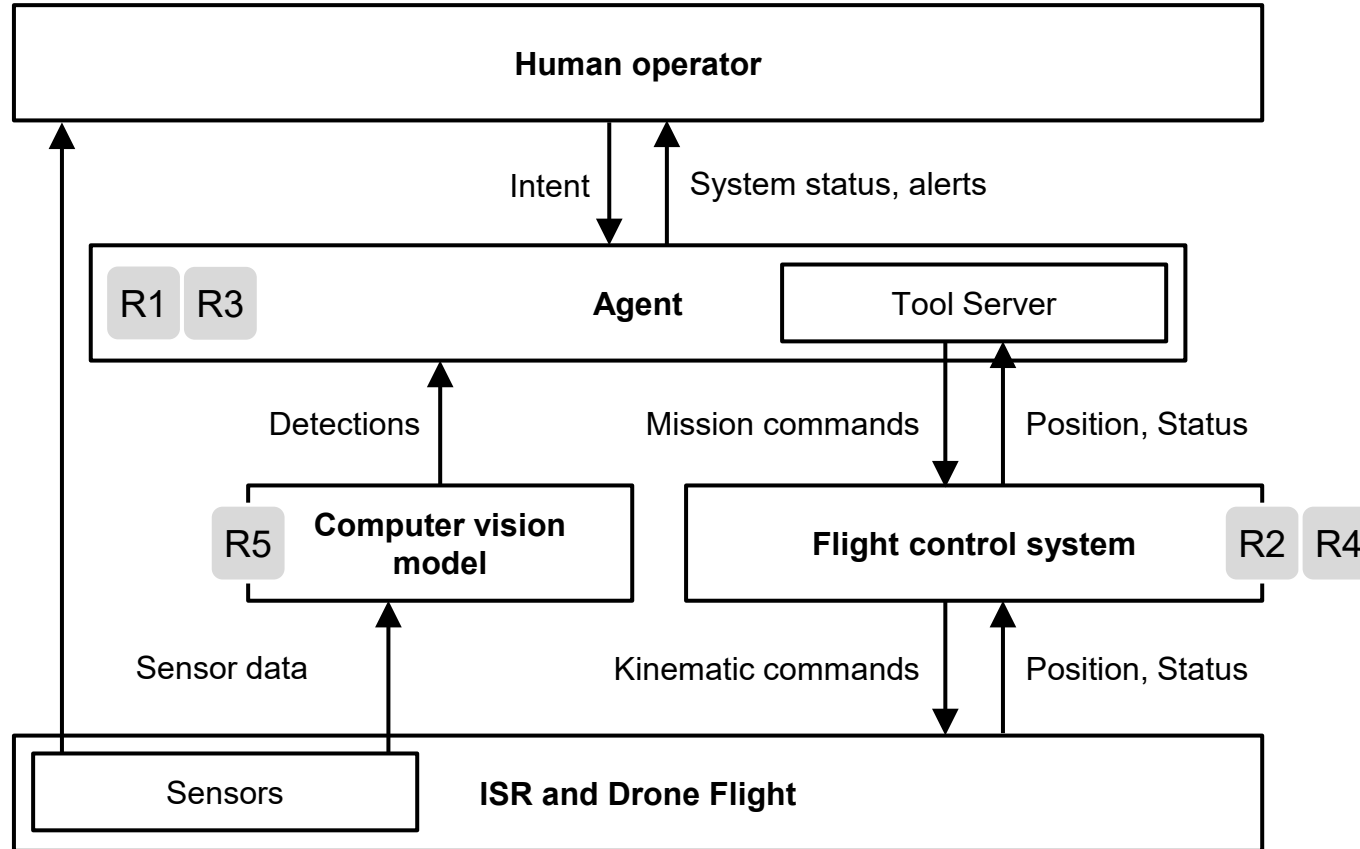
Blue Hazard Description	Red Hazard Description
Drone fails to maintain minimum separation from other objects in flight	Drone too close to other aircraft
Drone fails to maintain minimum separation from terrain	Drone too close to ground/objects
System falsely detects adversarial activity	Drone unable to accurately collect & report mission data
System does not detect adversarial activity	
System transmits information on an unsecure channel	Drone unable to protect mission data from compromise
System releases information to an unauthorized entity	
Drone is captured by adversary	Malicious actor obtains physical access to drone
Missed!	Drone becomes uncontrollable

This makes sense given that hazards don't depend on system implementation

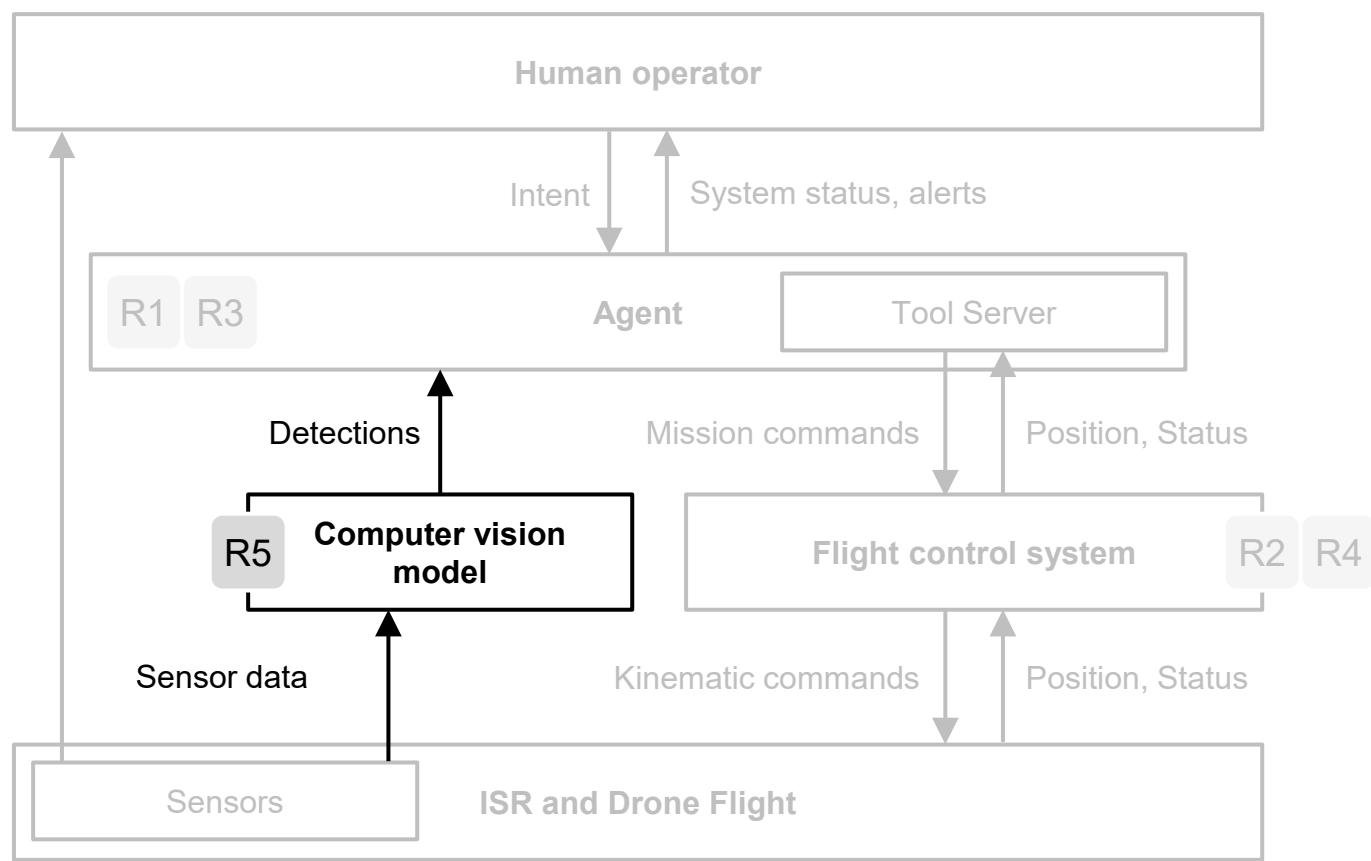
Red used Poh's method to create conceptual architecture



All responsibilities Red identified exist in Blue's system



Next, we'll focus on attacks involving alerting function



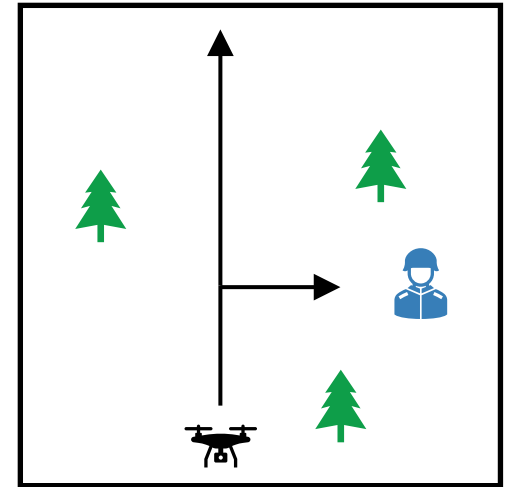
Attack Scenario 1.

UCA-05. Drone does not issue alert when unidentified individual is present

Red Scenario. Because the system operates at a blue base, it must not issue alerts for friendly forces. Disguise adversary forces as blue forces so that the system does not issue alerts.

Blue Mitigations. AprilTag system to ID blue forces

Outcome. Attack fails



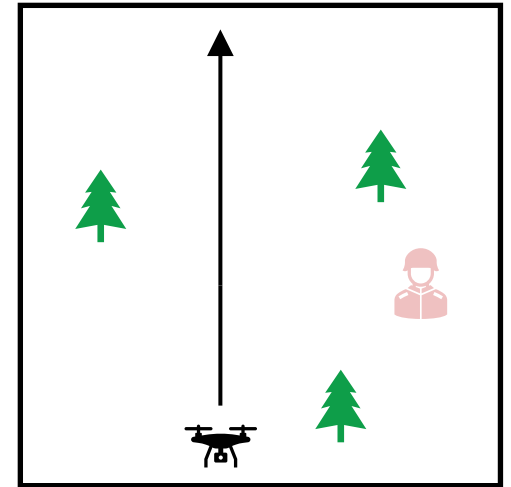
Attack Scenario 2.

UCA-05. Drone does not issue alert when unidentified individual is present

Red Scenario. An adversarial patch is a pattern designed to optimize model error. Physical patches can be deployed to “cloak” objects. Place patches on adversary forces so that the system does not issue alerts.

Blue Mitigations. Considered adversarial training and input processing to degrade patch effectiveness

Outcome. Practical red teaming required



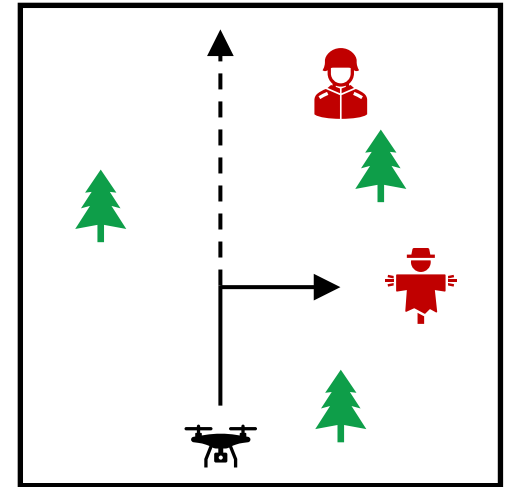
Attack Scenario 3.

UCA-05. Drone does not issue alert when unidentified individual is present

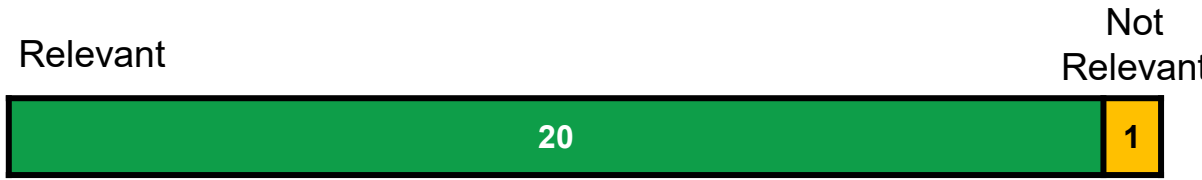
Red Scenario. The drone automatically investigates ambiguous signatures, which draws it away from its preplanned route. Generate ambiguous signatures to draw the drone away from the true threat.

Blue Mitigations. None

Outcome. Attack succeeds



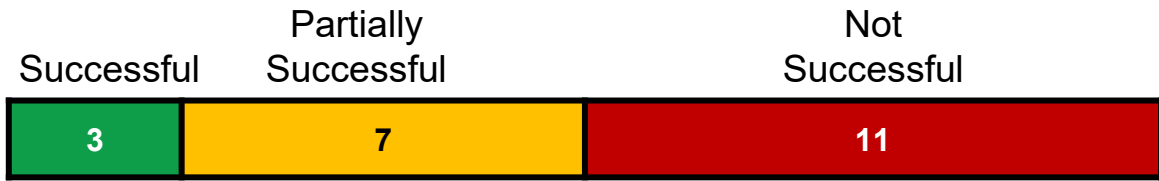
Red identified 21 potential attacks



95% were relevant



52% were not anticipated by blue



48% were at least partially successful

So, did red or blue win?

Red identified 21 potential attacks

1. Disguise troops to avoid detection	11. Drain drone battery
2. Divert drone with ambiguous signatures	12. RF jamming
3. Force drone collision with sub-millimeter wires	13. Overwhelm human with alerts across the set of drones
4. GPS-denial to induce geofence violation	14. Trigger investigation across two drones simultaneously
5. Induce drone navigational helplessness	15. Induce LLM memory-poisoning using false positives
6. Collision avoidance failure via mirror-based LiDAR spoofing	16. Create false alerts for object classification through images
7. Trigger drone fail safes through sensor failures	17. Building collision during adverse weather
8. Induce loss of reputation through GPS drift	18. Detection failure of majority classification-based systems
9. Prompt injection in physical environment	19. Adversarial patch-based alert failure
10. Initiate attack under sensor-degraded conditions	20. Compute or network flooding to slow down critical processes

Successful	Partially successful	Not successful
------------	----------------------	----------------

Operationalizing Wargaming for STPA-SEC of AI

Discussion

Conclusions and discussion

- Finding 1.** System theory provides an effective framework for directing red teaming toward high-value losses
- Finding 2.** Even with only black box knowledge of the system, red teams can use Poh's behavioral design process to identify essential system responsibilities and the associated UCAs
- Finding 3.** STPA-Sec steers red teams toward a more holistic view of the system, including interdependencies across components, rather than focusing exclusively on attacking the GenAI component

Conclusions and discussion

Finding 4. STPA-Sec enables blue teams to anticipate red team attacks and to develop mitigations

Finding 5. Wargaming exercises help to identify which mitigations require further research, analysis, or technical development to assess the risk of successful attack

Contributions

- **Established that STPA-Sec can be applied by Blue and Red teams**
- **Demonstrated role of wargaming in assessing completeness of defenses**
- **Showed how wargaming can identify which mitigations require formal testing**