FRAMEWORK FOR TRUSTWORTHY AUTONOMY

36th Soar Workshop
Ann Arbor, Michigan

SoarTech
Modeling human reasoning.
Enhancing human performance.

Scott D. Lathrop
Director, Secure Autonomy

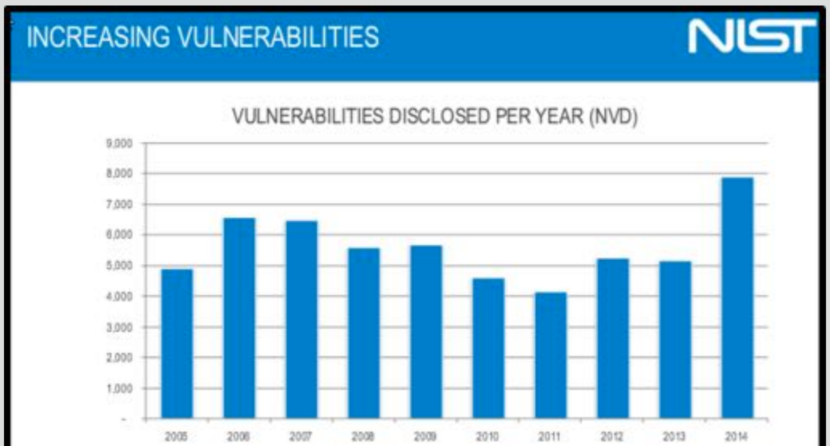
6/9/16

Problem – Cyberspace Related Issues

Increasing Complexity

- Space Shuttle: ~400K LOC
- F22 Raptor fighter: ~2M LOC
- Linux kernel 2.2: ~2.5M LOC
- Hubble telescope: ~3M LOC
- Android core: ~12M LOC
- Army Future Combat Sys.: ~63M LOC
- Connected car: ~150M LOC
- Autonomous vehicle: ~300M LOC

Increasing Vulnerabilities



In 2014:

- Over 7,900 new vulnerabilities disclosed & catalogued
- ~4,300 in Open Source, ~3,600 in commercial software

Increasing Threats

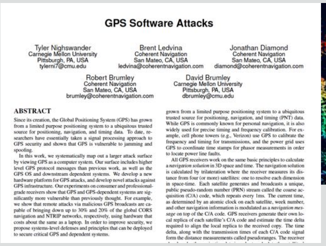
SOARTECH

ROBOTICS PLATFORMS NOT EXEMPT

Sensors

- Integrity attacks (i.e. spoofing), e.g.
  - GPS PNT attacks
  - Lidar spoofing
- Availability attacks (i.e. Denial of Service)

GPS spoofing (Nighswander, 2012)



GPS Software Attacks

Tyler Nighswander
Carnegie Mellon University
Pittsburgh, PA, USA
tylerni7@cmu.edu

Brent Ledvina
Coherent Navigation
San Mateo, CA, USA
ledvina@coherentnavigation.com

Jonathan Diamond
Coherent Navigation
San Mateo, CA, USA
diamond@coherentnavigation.com

Robert Brumley
Coherent Navigation
San Mateo, CA, USA
brumley@coherentnavigation.com

David Brumley
Carnegie Mellon University
Pittsburgh, PA, USA
dbrumley@cmu.edu

ABSTRACT

Since its creation, the Global Positioning System (GPS) has grown from a limited purpose positioning system to a ubiquitous trusted source for positioning, navigation, and timing data. To date, researchers have essentially taken a signal processing approach to GPS security and shown that GPS is vulnerable to jamming and spoofing.

In this work, we systematically map out a larger attack surface by viewing GPS as a computer system. Our surface includes higher level GPS protocol messages than previous work, as well as the GPS OS and downstream dependent systems. We develop a new hardware platform for GPS attacks, and develop novel attacks against GPS infrastructure. Our experiments on consumer and professional-grade receivers show that GPS and GPS-dependent systems are significantly more vulnerable than previously thought. For example, we show that remote attacks via malicious GPS broadcasts are capable of bringing down up to 30% and 20% of the global CORS navigation and NTRIP networks, respectively, using hardware that costs about the same as a laptop. In order to improve security, we propose systems-level defenses and principles that can be deployed to secure critical GPS and dependent systems.

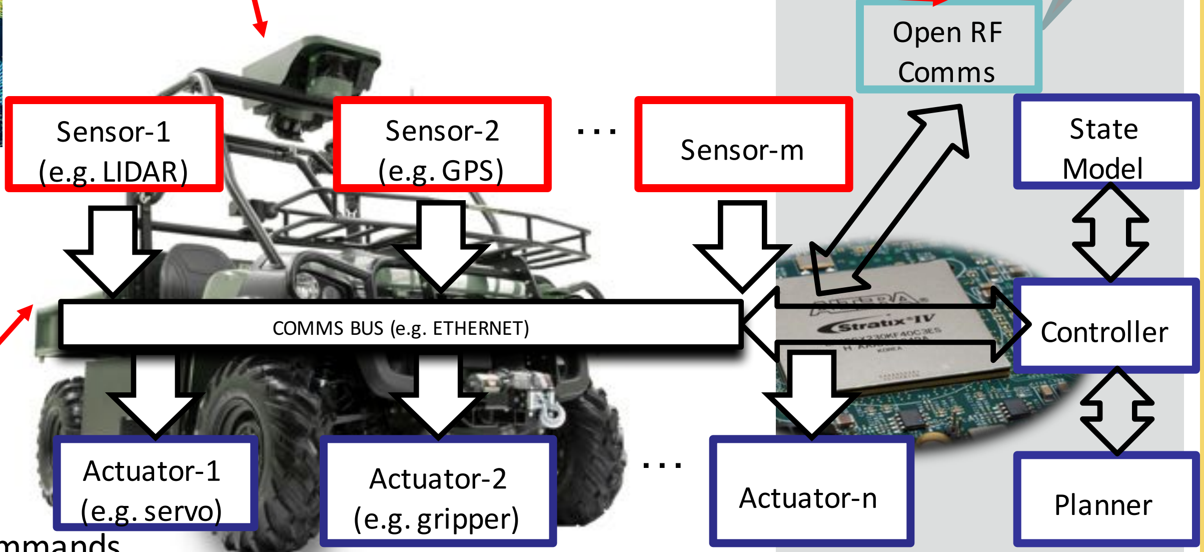
grown from a limited purpose positioning system to a ubiquitous trusted source for positioning, navigation, and timing (PNT) data. While GPS is commonly known for personal navigation, it is also widely used for precise timing and frequency calibration. For example, cell phone towers (e.g., Verizon) use GPS to calibrate the frequency and timing for transmissions, and the power grid uses GPS to coordinate time stamps for phasor measurements in order to locate power line faults.

All GPS receivers work on the same basic principles to calculate a *navigation solution* in 3D space and time. The navigation solution is calculated by trilateration where the receiver measures its distance from four (or more) satellites: one to resolve each dimension in space-time. Each satellite generates and broadcasts a unique, public pseudo-random number (PRN) stream called the coarse acquisition (C/A) code, which repeats every 1ms. The current time, as determined by an atomic clock on each satellite, week number, and other navigation information is modulated as a *navigation message* on top of the C/A code. GPS receivers generate their own local replica of each satellite's C/A code and estimate the time delta required to align the local replica to the received copy. The time delta, along with the transmission times of each C/A code signal form the distance measurements called pseudoranges. The receiver

Lidar spoofing (Petit, 2015)

Communication

- Confidentiality attack – e.g. Traffic eavesdropping
- Inadequate key management/poorly implemented encryption algorithms
- Integrity attacks – e.g. Buffer overflow/remote code execution, code injection
- Availability attacks – Denial of Service/Jamming
- Over the air (OTA) software updates



Onboard processing

- Integrity - Unauthenticated commands
- Broad attack surface – Little to no IP Port/Protocol restrictions
- Availability attack against legitimate commands
- Close access attacks
  - USB ports
  - Maintenance laptops
  - Cell phone
  - Physical Insider

Other potential threat vectors

- Supply chain threat – e.g. FPGA bitstream files
- Software repositories
- Legacy components => frequency of patching & refresh of hw/sw
- *Unique AI algorithmic vulnerabilities associated with autonomous systems*

INSIGHTS

- **General Principles**
  - Cybersecurity != Cyberspace defense--cannot defend everything – focus on "key terrain"
  - Must be able to detect, characterize, respond, and adapt *within mission context*

- **Adversary actors**
  - Multiple "online" personas associated with one physical identity
  - Tactical actions derived from goals/intents
  - Both parallel (e.g. reconnaissance, DDOS) and sequential (e.g. delivery/exploitation) action
  - Cognitive, Logical, and Physical indicators

Cyberspace Layer	Indicators	Detection Difficulty (Relative)	Adversary Cost to Change (Relative)
Persona/Cognitive	• Personas and Identities • Intent/Goals • Tactics, Tech., Procedures + C2 • Social Presence and communication	Hard	Medium (more difficult after foothold is gained)
Logical	• Malware variants • IP addresses/TCP Ports • Configurations/Logs • File hashes	Low->Medium (depending on adversary sophistication)	Low
Physical	• Infrastructure • Computing nodes • Electromagnetic Spectrum • Geo-Location • Persona biometrics (key stroke, mouse patterns, facial recognition)	Medium	High (lower after foothold is gained)

INSIGHTS

- **General Principles**
  - Cybersecurity != Cyberspace defense--cannot defend everything – focus on ”key terrain”
  - Must be able to detect, characterize, respond, and adapt

- **Adversary actors**
  - Multiple ”online” personas associated with one physical identity
  - Multiple tactical actions (derived from goals/intents) to achieve objectives
  - Both parallel (e.g. reconnaissance, DDOS) and sequential (e.g. delivery/exploitation) action
  - Cognitive, Logical, and Physical indicators

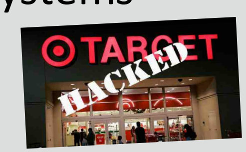
Generation Gap Could Lead to a Cybersecurity Worker Shortage

Schools are scrambling to provide courses that emphasize cybersecurity, an element traditional computer science tracks have not included.

- **Shortfall of expertise**
  - Well documented shortage of cyber expertise
  - Combat units do not have cognitive resources to fight kinetic and non-kinetic fight simultaneously
  - Demands some autonomy (*but there is a complexity tradeoff*)

- **Autonomous systems present new attack vectors**
  - Key benefit to autonomy – system’s ability to ”decide what to do next”
  - Decision knowledge emerges from perception and memory – both subject to compromise

- **Trustworthiness & Trust** - Key obstacle to employment of autonomous systems



SOARTECH

Conceptual Approach Trustworthy Framework for Autonomy

Hypothesis: Trustworthy framework for autonomy composed of three characteristics

Trustworthiness
&
Trusted

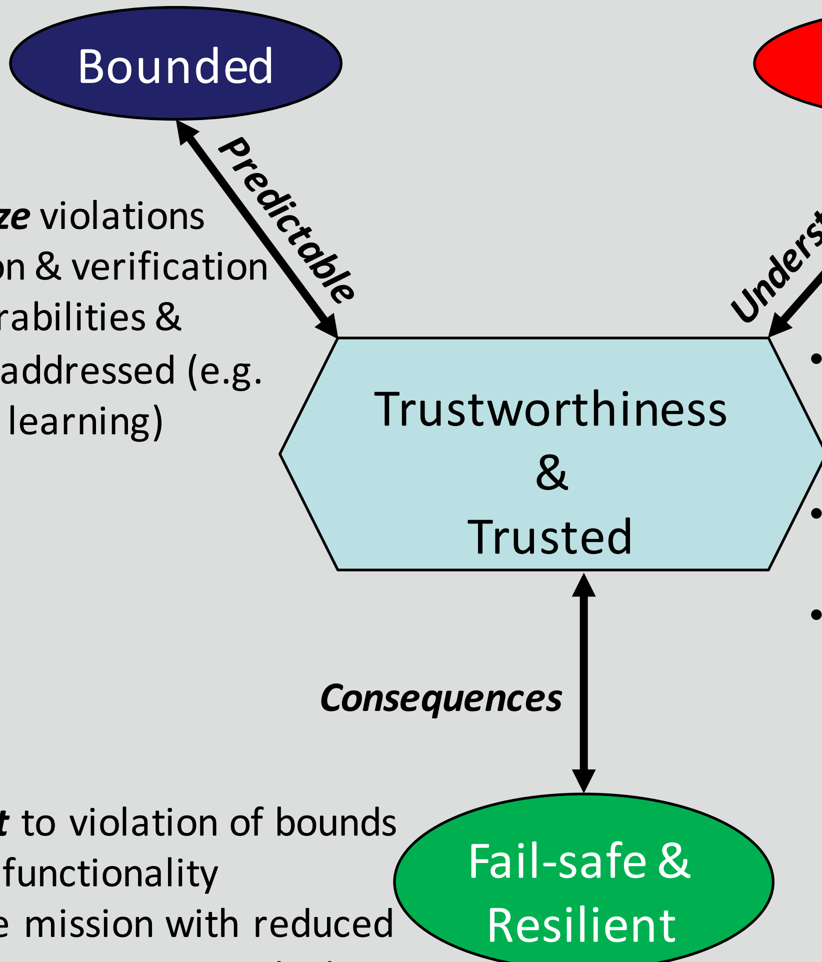
Trust Models*			
Ratnasignham, 1998	Deterrence	Knowledge	Identification
Lewis & Weigert, 1985	Cognitive	Emotional	Behavioral
Fahrenholtz, 2001	Habits	Passion	Policy

Common traits:
- Predictability
- Understanding
- Similarity
- Consequences

*From Wallace, 2007

Conceptual Approach - Trustworthy Framework for Autonomy

Hypothesis: Trustworthy framework for autonomy composed of three characteristics



- ***Detect & characterize*** violations
- Continuous validation & verification
- AI algorithmic vulnerabilities & mitigations must be addressed (e.g. adversarial machine learning)

- ***Support transparency*** through appropriate representations & processes
- Model user state/comprehension
- Multi-modal, adaptive, & interactive user interfaces

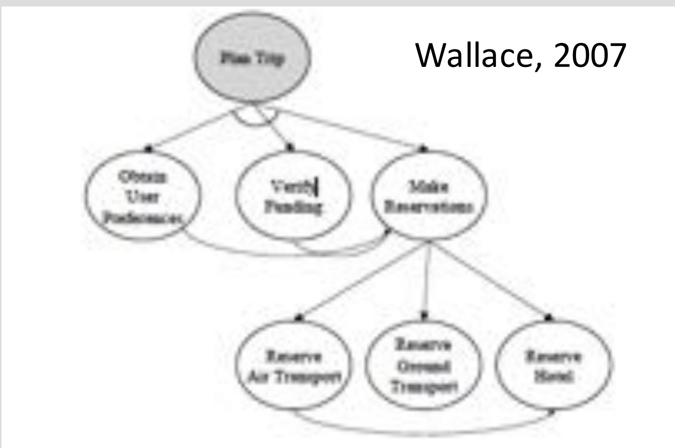
- ***Respond* and *adapt*** to violation of bounds
- Fail-safe – reduced functionality
- Resilient – Continue mission with reduced functionality (mission success may be less *efficient* and *effective*)

CHALLENGES & POTENTIAL APPROACHES

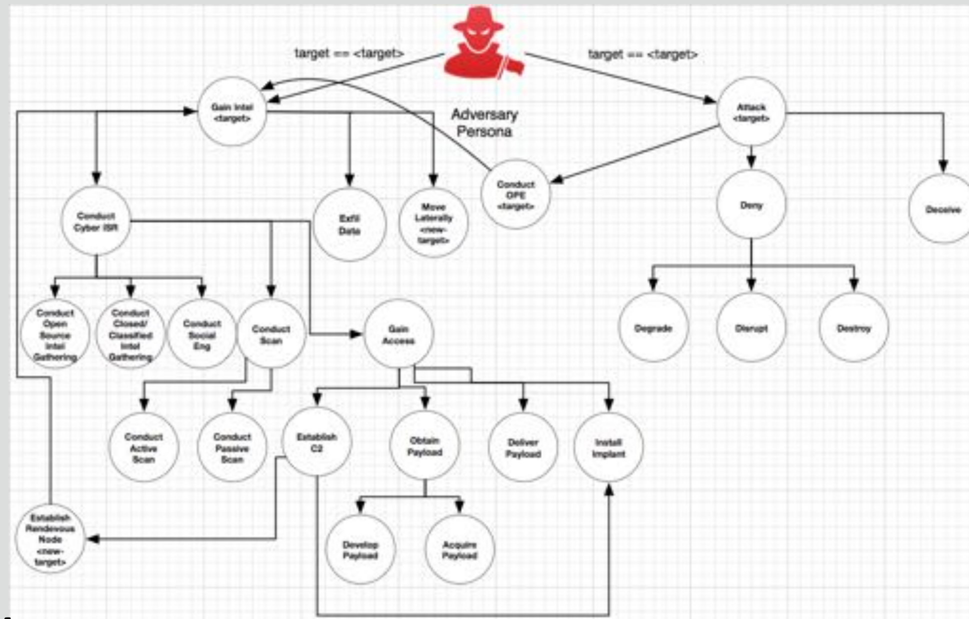
> “Trust but verify”
> - Army leadership philosophy

- **Bounded behavior – *detect & characterize***
  - ➢ Behavioral meta-models (Wallace, 2007)
  - ➢ Monitoring and Validating Synthetic Behavior (Jones, 2015)
  - ➢ Top-down, Abductive Reasoning for Behavior Detection (Crossman, 2011)
  - ➢ Ethics (Arkin, 2012)
  - ➢ Safety Envelope for Security (Tiwari, 2014)
  - ➢ **Cyber (?) – *Research Gap***

Friendly Behavior Envelope



Adversary Behavior Envelope



- ***Explainable - Support Transparency***
  - ➢ Episodic Memory (Nuxoll, 2007)
  - ➢ Model of User state/comprehension + multi-modal interfaces (Taylor, 2012)
- ***Fail-Safe & Resilient - Respond* and *adapt -- Research gaps***
  - ➢ What/Who makes decision to move to a fail-safe state?
  - ➢ What are the space of actions?

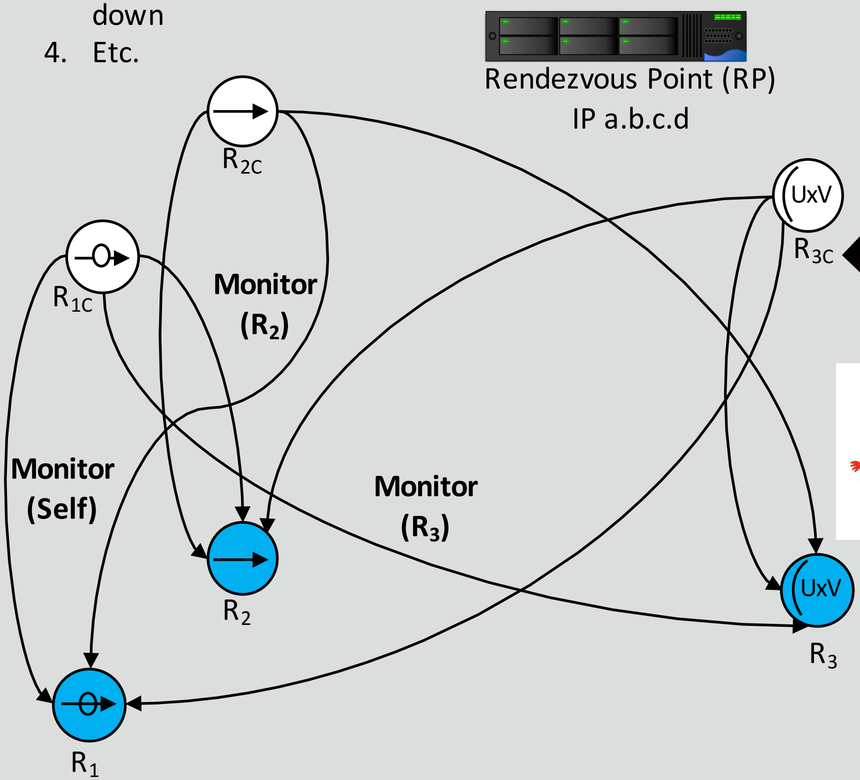
CYBER DEFENSE BATTLE BUDDY CONCEPT

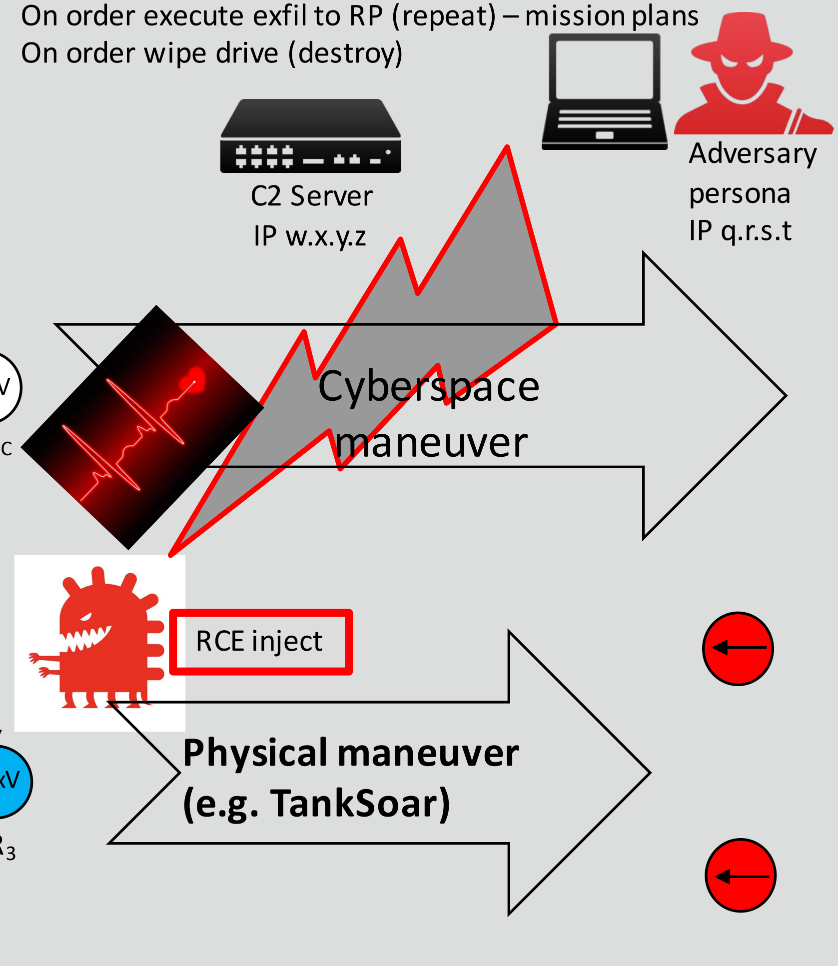
USE CASE (Friendly)

1. R_{1C}, R_{2C}, R_{3C} observe multiple R_3 connections to a.b.c.d/443 via logged connections
2. R_{3C} directs collection of physical signal emissions emanating from R_3 to confirm/deny
3. R_{1C}, R_{2C}, R_{3C} (majority) agree that R_3 has a boundary violation (transmitting to unknown IP) and recommend/decide on one of following actions (situation dependent (*cyberspace maneuver)*
    1. Block IP connections to a.b.c.d (via R_3 iptables)
    2. Repurpose R_3 as R_{3C} (and vice versa) to enable communication to continue and observe
    3. Hunt for communicating process on R_3 and shut down
    4. Etc.

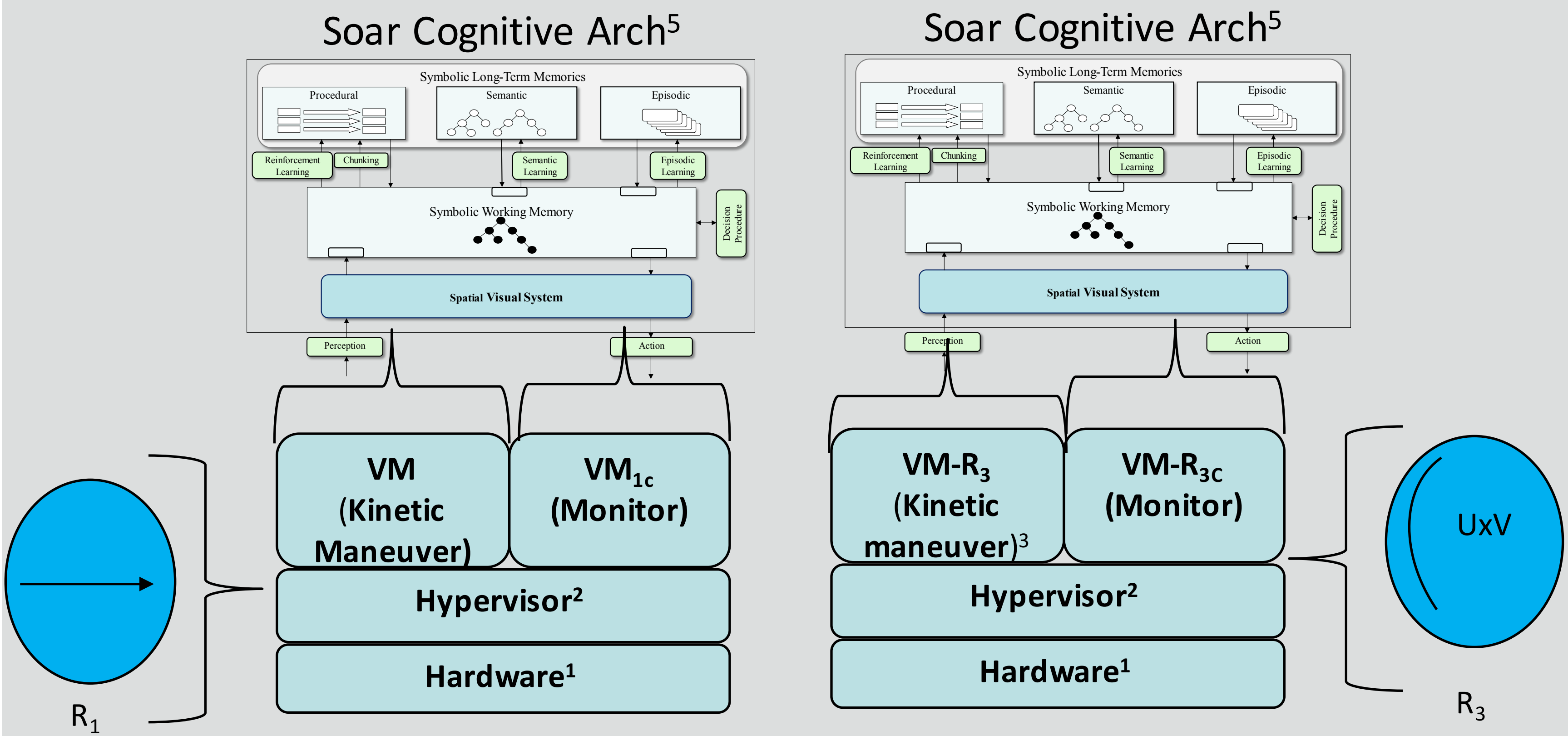
USE CASE (Adversary)

1. Gain access to R_3 via remote code exploit (RCE) through RF inject into vuln. P2P software (e.g. a ROS Node)
2. Decrypt install binary and write to disk
3. Execute install to extract in-memory implant/backdoor
4. Send heartbeat to C2 server and receive instructions for rendezvous collection point; Remove install binary
5. (Persona through C2 server) recon file system for relevant plans
6. On order execute exfil to RP (repeat) – mission plans
7. On order wipe drive (destroy)





Cyber defense Battle Buddy Technical Approach



NOTES:

[1]General Purpose Processor (GPP) or embedded system with ability to partition address space

[2]Hardware based hypervisor for efficiency and to support out-of-band processing.

[3]VM_1 (or more) – focused on the tactical behaviors to support synchronized kinetic + non-kinetic maneuver

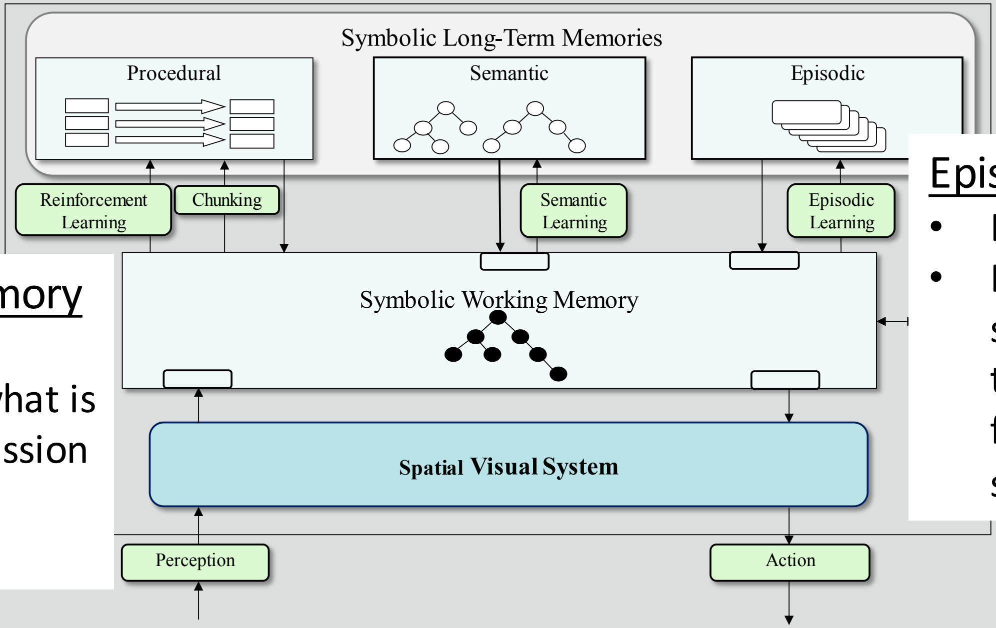
[4]VM_2– focused on behavior monitoring (communicate with other monitors preferable using out-of-band, non-operational link).

[5]Tactical Behavior implementation for kinetic/non-kinetic maneuver and cyber monitor

What Does Soar Have to Do with This Approach?

Procedural
- Hierarchical control & reasoning
- Abductive reasoning (hypothesis testing)
- Transitions to fail-safe states (policies)

Semantic
- Adversary attack graphs (doctrinal templates)
- Compute network nodes and connections
- Friendly tools, techniques



Working memory
- Situational context - what is broader mission context?

Episodic
- Explaining behavior
- Reduce hypothesis search space (these are the indicators I looked for last time in this situation)

SVS
- Physical indicators (e.g. geo-location of threat vectors)
- Integration of kinetic/non-kinetic maneuver (in order to exploit through RF, must have transmitter within radius x)

EVALUATION – NONE
{SOME RESEARCH & EVALUATION QUESTIONS}

- What are the design space tradeoffs?
  - Number and types of monitoring agents?
  - Self-monitoring or group monitoring with voting (majority) algorithm
  - Soar controlling both tactical kinetic/non-kinetic behavior and cyber defense monitoring agents? If separate, how/when do they interact?
  - What is CPU overhead? Communications overhead?

- What cyber-related knowledge is most useful for detection?
  - Cognitive – are behavior envelopes sufficient for tracking adversary behavior?
  - Logic - OS/App logs, file hashes, security tools' output
  - Physical emissions, spatial (e.g. geolocation) and temporal

- What are the unique vulnerabilities associated with AI systems? What are potential mitigation countermeasures?

- What is necessary for supporting infrastructure?
  - Modeling and simulation environment and tools to support development and experimentation
  - Physical platforms, space, and cyber/EW tools to support live experimentation

Nuggets & Coal

Nuggets	Coal
Exploring Soar applicability in a new domain (Cyberspace)	No design, implementation, evaluation ☹
Exciting, explosive area	Unclear of right approach – much hype around AI and "cognitive" approaches
A lot of interest (+Work)	A lot of work