THE ROAD TOWARDS AUTONOMOUS CYBERSECURITY: REMEDIES FOR SIMULATION ENVIRONMENTS

Martin Drašar, Ádám Ruman, Pavel Čeleda, Shanchieh Jay Yang

drasar@ics.muni.cz

SECAI 2023

How to get to autonomous cybersecurity?

- Despite all the promises of AI, we are not getting anywhere with autonomy
- Numerous reasons:
  - Domain complexity
  - Insufficient training datasets
  - Insufficient tooling
- This presentation addresses those reasons through the prism of training environments

State of the art

- Training environments are under-researched and under-developed
- Generic solutions cannot be used, they do not capture the complexity
- Cybersecurity solutions are either too abstract, or too specific
- Narrow scope of tooling

- No real push for creating deployable solutions

Simulation environments

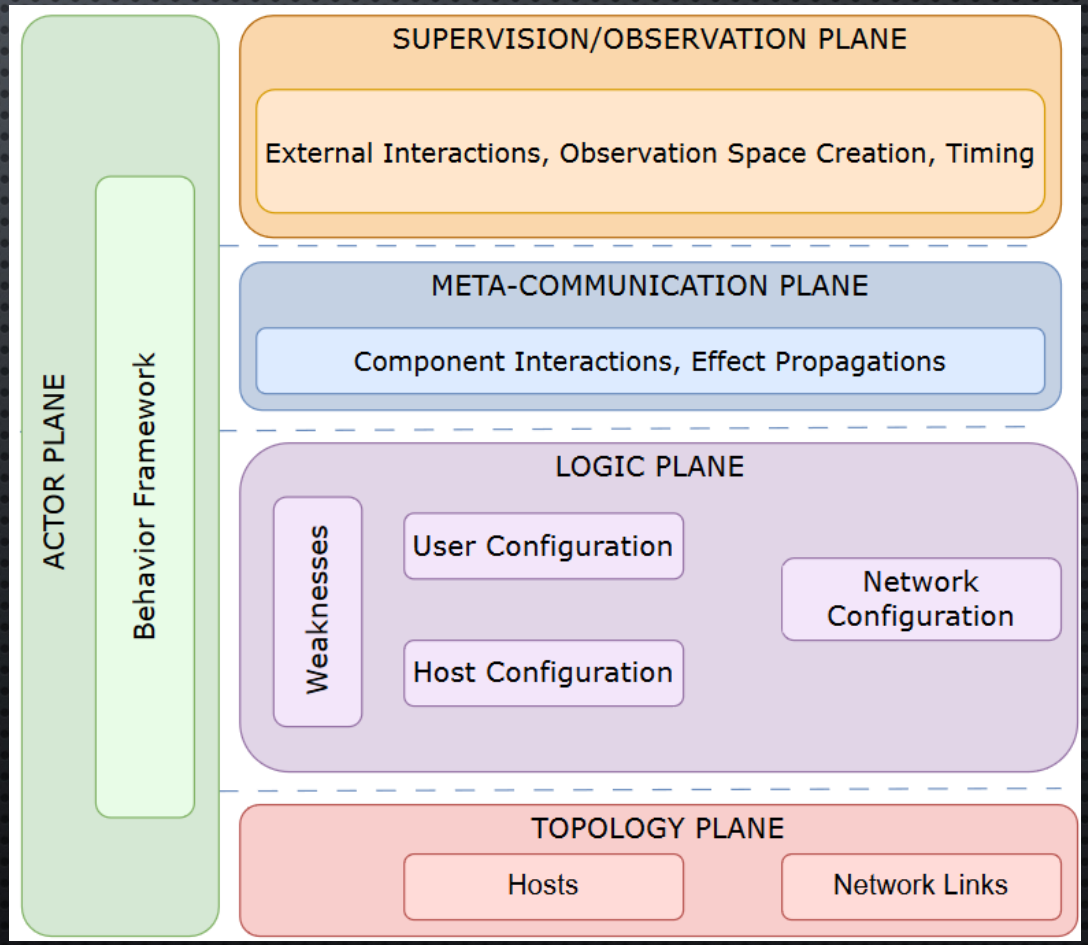
- Often build as means to an end
- No solid theoretical foundation
- In effect, different environments are incomparable
  - At least until today…

Assessment framework for autonomous cyber agent simulation

- Simulations cyber terrain ontology
- Actor evaluation framework
- Comprehensiveness and concreteness measurement

SIMULATIONS CYBER TERRAIN ONTOLOGY

- **TOPOLOGY PLANE:** PHYSICAL TOPOLOGY OF THE INFRASTRUCTURE
- **LOGIC PLANE:** FUNCTIONALITY OF SIMULATION
- **META-COMMUNICATION PLANE:** INTER-PLANE SIGNALING
- **SUPERVISION/OBSERVATION PLANE:** OBSERVATION SPACES AND TIMING
- **ACTOR PLANE:** STATE-CHANGING ENTITIES



Actor Evaluation Framework

- Based on COI framework
  - **Intent**: existential goal of actor
  - **Opportunities**: domain of events that can be invoked by actors
  - **Capabilities**: Predicates limiting actor's opportunities
  - **Preferences**: Prioritization based on secondary intents
  - **Sophistication**: cost and risk assignment of taking specific opportunities
- Four generic actor types: Adversaries, defenders, benign participants, fates

Comprehensiveness and Concreteness Measurement

- Based on MITRE's metrics
- **Perspectives**: Attack vectors, attack actions, adversary characteristics, defender actions, Technical architecture, technical vulnerabilities
- **Concreteness**: abstract, notional, representative, fully realized
- **Comprehensiveness**: fragmentary, partially specified, fully specified

Assessment of Deployability

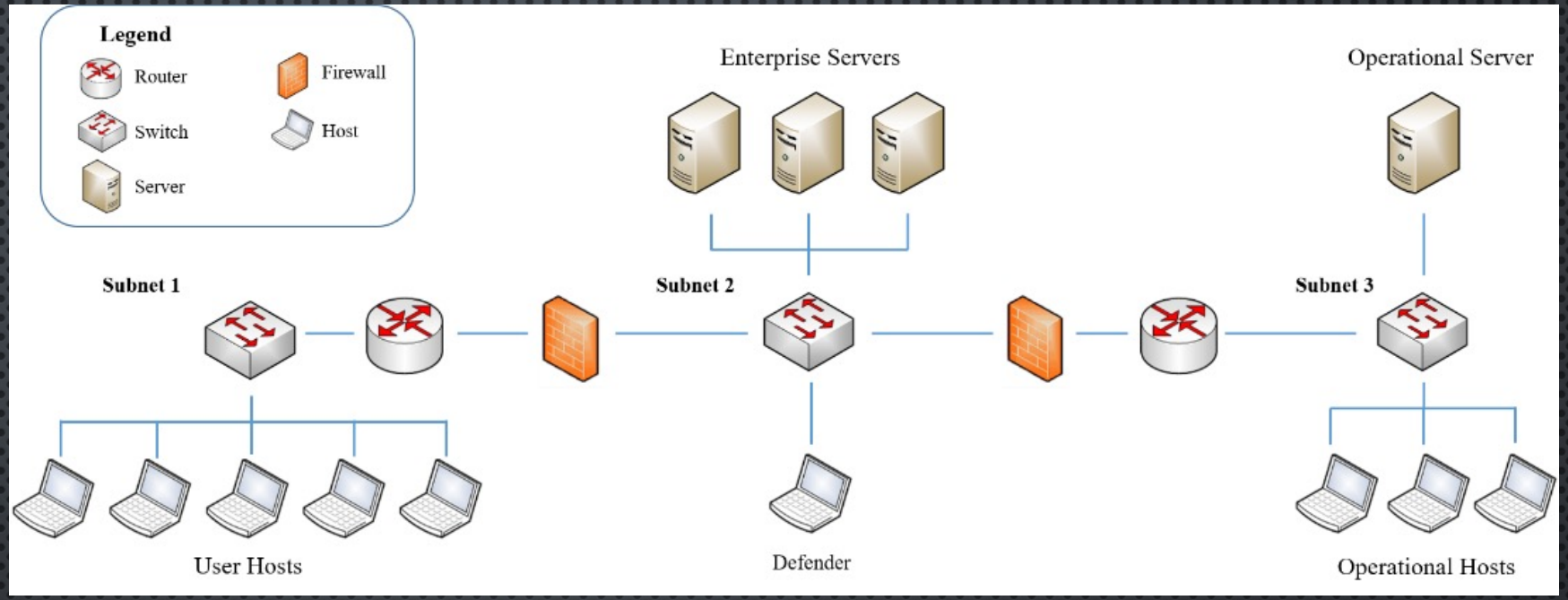
- We assert that to create deployable solutions, the training environment has to:
  - Approach minimal abstraction
  - Provide actionable descriptions of the terrain, users, vulnerabilities, etc.
  - Be dynamic and able to evolve
  - Be concrete and comprehensive

- To this end we analyzed the following environments:
  - Yawning Titan, CyberBattleSim, CybORG, CYST, and NaSimEmu (not in the paper)

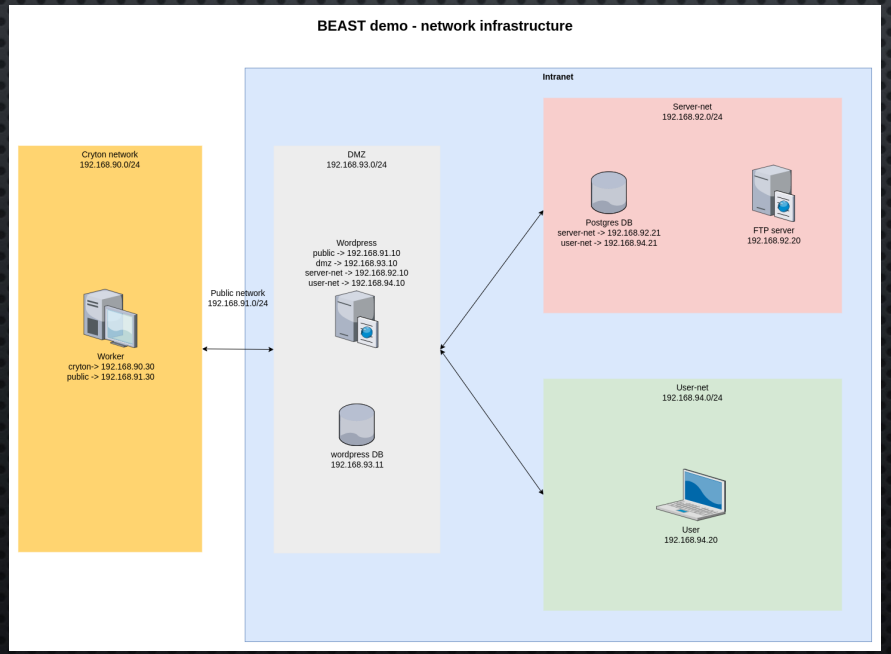
Comparison of CybORG and CYST

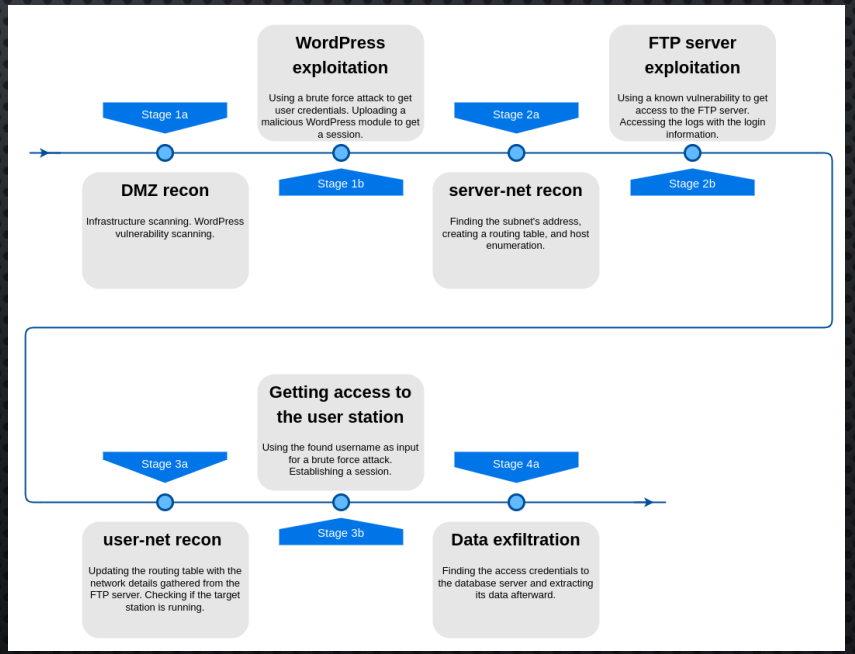
- According to assessment framework the two most sophisticated tools
- We made a qualitative evaluation based on 2nd CAGE challenge and similar custom scenario for CYST
- We evaluated strong points from the point of view of a developer of autonomous cybersecurity system.

CAGE challenge



CYST scenario

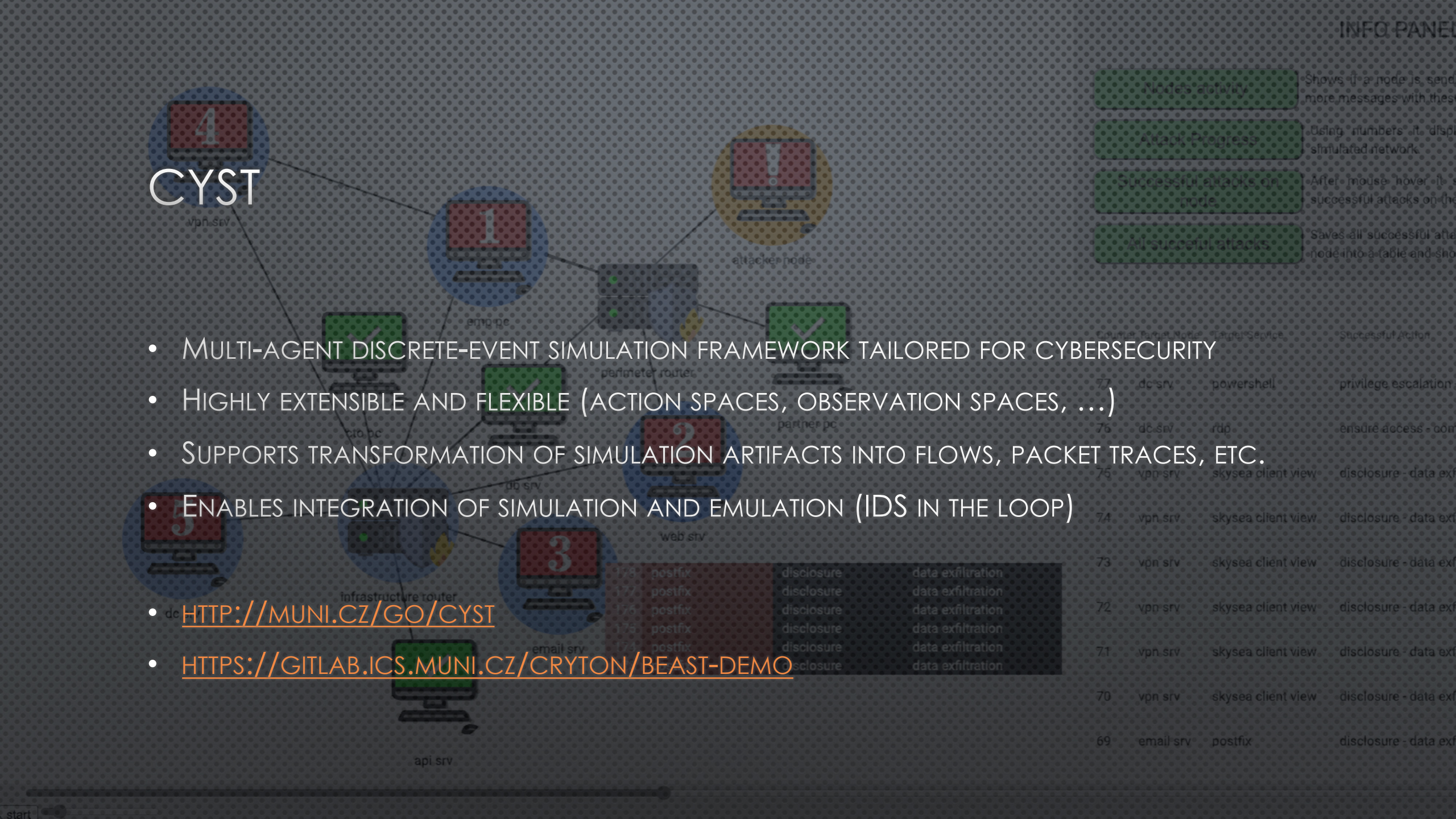




- Both environments gravitate to similar goals and use similar approaches
- CybORG is more readily usable at the expense of advance features
- CybORG dropped emulation support, so it is unusable in the future

CYST	CybORG
Infrastructure & Logic	
Network traffic shaping.	Service and OS knowledge base.
Modeling the traffic.	Modeling OS.
Support for complex authentication and authorization.	Host level information down to PID and files and their permissions.
Supervision, Actors & Agents	
Unbounded action and observation spaces.	Provides global and local observations.
Complex action parametrization to mimic real-world actions tailored for RL.	Integrated rewards.
Non-singular action handling.	Rich action space.
Transaction support for faster training.	
Agent-agent interaction in addition to agent-environment.	
External & Miscellaneous	
Strong focus on deployability.	Ready wrappers and interfaces for OpenAI.
Maximizing extensibility, stand-alone packages, usable as a library, and plugin support.	
Integration with outside running services.	
Human-machine interface.	

CYST



- MULTI-AGENT DISCRETE-EVENT SIMULATION FRAMEWORK TAILORED FOR CYBERSECURITY
- HIGHLY EXTENSIBLE AND FLEXIBLE (ACTION SPACES, OBSERVATION SPACES, …)
- SUPPORTS TRANSFORMATION OF SIMULATION ARTIFACTS INTO FLOWS, PACKET TRACES, ETC.
- ENABLES INTEGRATION OF SIMULATION AND EMULATION (IDS IN THE LOOP)

- HTTP://MUNI.CZ/GO/CYST
- HTTPS://GITLAB.ICS.MUNI.CZ/CRYTON/BEAST-DEMO

AI-DOJO

- Research project to create a platform for development of autonomous cybersecurity systems
- Integration of simulation and emulation
- Library of agents with different behavior (attacker, defenders, users)
- Automated generation of realistic cybersecurity scenarios to support learning

- https://muni.cz/go/ai-dojo

AICA-IWG

- Follow-up to NATO IST-152 tasked with specification of reference architecture for autonomous cyberdefense systems
- Working group focused on furthering the development of autonomous cybersecurity systems
- Academia, industry, defense

- https://www.aica-iwg.org/

ADDENDUM: NASIMEMU ASSESSMENT

- Abstraction level: High
- Topology:
  - Dynamic changes: Allowed
  - Representation: Custom data structure
- Logic:
  - Network:
    - Rule direction: Bidirectional
    - Rule granularity: Per protocol, for subnets
    - Additional capabilities: None
  - Hosts:
    - OS: Available, High-level tags
    - Software: Process
    - Software properties: Versions using tags
  - Users:
    - Account granularity: Not supported
    - Credentials: Not supported
    - Authorizations: Only the level of control over a host
    - Remote access control: Not supported
    - Local access control: user privileges
  - Weaknesses:
    - Realism: High
    - Representation: Exploitable vulnerabilities
    - Applicability guard: Service name
    - Additional action attributes: cost, probability of success
- Meta communication:
  - Event invocation: supervision intervention
  - Event propagation: supervision intervention
- Supervision/observation:
  - Observation space: provided
  - Timing: Sequential
  - Reward computation: Provided
  - Multi-agent support: UNKNOWN

Adversarial COPSI:

- Intent: Finding a pre-defined loot
- Opportunities: 8 actions (Exploit, Privilege escalation, ServiceScan, ProcessScan, TerminalAction)
- Capabilities: Accessible hosts (via control level), vulnerabilities
- Preference: Customizable via reward computation
- Sophistication: agent dependent

Defender COPSI:

- Defenders not available

Concreteness:

- Adversary characteristics: Abstract
- Attack vectors: Notional
- Attack actions: Abstract
- Defender actions: Unavailable
- Technical architecture: Notional
- Technical vulnerabilities: Representative

Comprehensiveness:

- Adversary characteristics: Fragmentary
- Attack vectors: Partially specified
- Attack actions: Partially specified
- Defender actions: Unavailable
- Technical architecture: Fragmentary
- Technical vulnerabilities: Partially specified