Reinforcement Learning for Autonomous Cyber Defense

Dr. Ahmad Ridley

Department of Defense

Cyber Maryland

October 9, 2018
Cyber Resilience Research

**Problem:** Develop an autonomous cyber defense system that reasons and responds to improve the resilience of enterprise networks to advanced cyber-threats

**System should:**
- Enable network, and dependent missions/services, to anticipate, withstand, recover, and **evolve** from cyber attacks
- Perform automated, real-time reasoning and response
- Be adaptive to changing contexts and learn from past actions

**Key Questions:**
- When should system respond?
- Which responses most appropriate for current situation?
- What was actual effect of response?
- How can impact of responses be improved?
Cyber Resilience Challenges

• Real-time fusion of events from diverse and disparate data sources

• State representation and reasoning over state to choose defensive actions

• Developing secure design methodology for cyber defense systems

• Building human trust in autonomy of cyber defense system

• Creating metrics and experiments to measure impact of defensive actions on systems and adversaries
Cyber-Resilience Concepts

Figure 1. Resilience Base Metrics

Reiger, Craig C., Resilient Control Systems, Idaho National Laboratory, 2014
CYBER SECURITY
How the heck do you defend against this?

- Cyber defense at its most basic, involves:
  - Understanding what is running on your network, and
  - Putting it in the context of normal behavior.
- Examples include:
  - Network monitoring: what information is being sent where?
  - Process monitoring: what is being computed?
Network monitoring is hard

- A simple, completely benign internet connection might contain
  - DNS lookups.
  - Redirects
  - Encryption and security
  - Collection investigation of cookies
  - Sharing of this and other information with partners
  - Delivery of content and advertising from multiple sources.
Process monitoring is hard

• The human types:

```c
int main()
{
    x = get.computers();
    foreach(computer in x)
    {
        hack(computer)
    }
}
```

• The computer sees:

```
0xA412 0xBEEF 0x1231
0x9ACE 0x@437 0xF00D
 .
 .
 .
```
Ok, so it’s hard. Then what?

- There are some simple strategies that can be effective:
  - Train employees on good operational security practices
  - Be especially suspicious of new things.
  - Keep software up to date
  - Keep a record of bad things, and flag if you see them again.
This strategy has some problems.

- Flagging new things annoys your users
- Overly reliant on vendors patching their software
- Slight changes by the adversary re-expose you to known threats.
ARTIFICIAL INTELLIGENCE
Artificial Intelligence describes a variety of computational techniques.

Some of the most popular ones now are:

- Machine Learning
- Reasoning and Perception
- Planning, Adaptation, and Reaction
Machine Learning

• Unsupervised Learning: Group like things together. Hope that this likeness helps solve a problem.

• Supervised Learning: Try to identify observations according to some pre-specified class label.

• **Goal:** Automate decision making based on input data.
Deep Neural Networks

- A set of algorithms which were invented in the 1940s.

- The idea is that we can use a computer to mimic how a brain learns.

- We ‘train’ by strengthening the connections between ‘neurons’ that produce the desired output.

- This is very hard, even for a computer, but computers are a lot faster than they were in the 40s.

- Now, Deep Neural Networks can give you very good performance on difficult classification problems.
Examples of Supervised Learning in the Wild

• Amazon Ads

• Netflix and Spotify Recommendations

• Vending Machines

• Speech understanding

• Personal Assistants (siri)
MACHINE LEARNING AND CYBER SECURITY
Even thought it’s natural, doesn’t mean it’s easy

• ‘Malicious’ may mean different things to different people. Moreover, it can be hard to determine whether something is ‘benign.’

• Labeling data can be highly labor intensive.

• How to extract features, measure performance, and use effectively are unsolved problems.
Advances in machine learning can also advance cyber security

- Feature extraction is a major problem, but deep neural networks may be able to do that automatically.

- Algorithms that can model human language, may also be able to model networks or programs.

- Perhaps these approaches can be combined to automatically understand network traffic.
Types of Adversarial Machine Learning

• Types of adversarial machine learning include:
  • Manipulating training data so that the machine learning algorithm learns to wrong patterns
  • Understanding the decision boundary of the machine learning algorithm, so that the machine learning algorithm can be defeated
  • Changing the statistical distribution of the observed data, so that the algorithm becomes incorrectly biased in retraining
AUTONOMOUS DECISION-MAKING
Autonomous Cyber Defense

• **Intuition:** System performs reasoning and response, where goals are well defined by human and some bounded range of actions is available.

• **Features:**
  • Automated reasoning, and
  • Adaptive response

• Automated reasoning and adaptive response needed to mitigate damage from attacks at machine speed and enterprise scale.

• Automated responses can also reduce uncertainty and increase situational awareness
Autonomous Cyber Defense Challenges

• Cyber training data and realistic testbed for repeatable, interactive experiments

• Speed and scalability of reasoning methods, esp. over large state spaces

• Decisions made on uncertain and incomplete data
  ➢ Adversary influence on computer network
  ➢ Users, admins, and defenders influence on computer network and adversary

• Reasoning across different time scales (e.g., correlation among historical, current, and future decisions and responses)

• Interactions between global and local decisions and responses (e.g., central vs. distributed system)
Why Reinforcement Learning (RL)?

- Solve sequential decision-making, or control, problem for cyber defense domain
  - Achieve directed goal in adversarial cyber environment
  - Outcome of actions are unpredictable
  - Feedback, i.e., reward signal, from environment required

- Training sets are either not available or difficult and expensive to obtain

- Challenge to accurately label situations to effectively predict labels for new situations in cyber defense (Supervised Learning)

- Difficult to learn structure from cyber data in a dynamic environment (Unsupervised Learning)

- Solution requires fundamentally different class of algorithms than just supervised or unsupervised machine learning ones
Reinforcement Learning (RL) Overview

1. RL agents learn to map situations to actions to maximize numerical reward
   - Agent must be able to learn from own experience
   - Trial-and-error search
   - Tradeoff between exploitation and exploration
   - Delayed reward

2. RL policy maps perceived situations, i.e., states, to actions
   - Probability distribution across actions

3. Reward function defines the RL problem goal
   - Function assigns each state-action pair a numerical reward
   - Indicates intrinsic, immediate desirability of state

4. Value function indicates long-term reward
   - Cumulative, expected discounted reward for each state
   - Allows agent to reason about long-term consequences of actions

5. Agent learns (near) optimal policy that maximizes expected long-term reward from a given state

---

RL Agent-Environment Interaction

\[ S_t \xrightarrow{R_t} A_t \xrightarrow{R_{t+1}} S_{t+1} \]

\[ S_{t+1} \xrightarrow{R_{t+1}} A_t \xrightarrow{R_t} S_t \]

RL Existing Theory and Applications

- **Theory:**
  - RL Algorithms (Value iteration, policy iteration, Deep Q-learning)
  - Markov Decision Processes (MDPs)
  - Optimization (Deterministic and Stochastic)
  - Dynamic Programming

- **Practical applications:**
  - Robotics
  - Autonomous Vehicles
  - Airline flight scheduling
  - Maintenance and repair planning
  - Games (backgammon, checkers, AlphaGo)
  - Limited cyber applications:
    - Cyber-analyst scheduling,
    - Distributed Denial of Service (DDOS)/DNS attacks on availability
Given possible cyber-attack conditions, train RL agent to mitigate attack while maintaining network performance level (preferably high)

Network of eighteen (18) equally valuable nodes

Node types:
- Assigned 0: Safe
- Assigned 1: Unsafe & not isolated
- Assigned 2: Unsafe & and isolated

State of system (i.e., network) = total number of safe nodes => $3^{18}$ possible states
**RL Cyber-Defense Scenario**

- Attacker can compromise any node and spread to nodes connected to compromised nodes

- RL agent actions:
  - Do nothing,
  - Patch node, or
  - Isolate & patch node

- Reward function: $R = \text{Benefit of action} - \text{Cost of action}$:
  - Benefit = no. of safe nodes
  - Do nothing action has zero cost
  - Isolate action costs more than patch action
  - Use a tunable parameter to adjust cost
Experiment Details

- Train/test with 10k/1k episodes, with 100 turns per episode.

- Each episode starts from a clean slate.

- Use greedy-epsilon policy.
  - Train: start with epsilon = 1.0, decrease to 0.1
  - Test: fix epsilon = 0.05

- Evaluate RL agent performance based on:
  - Average over the number of safe nodes each turn to get episode's performance.
  - Average over episodes to get one scalar score for experiment
First Experiment Results

- Trained RL agent for 10,000 episodes

- Evaluate results based on number of safe nodes

- Results are good:
  - Agent with epsilon = 0.05:
    - Mean no. of safe nodes = 16.3, Median no. of safe nodes = 16.6
  - Agent with epsilon = 1.00 (i.e., random policy):
    - Mean no. of safe nodes = 9.1, Median no. of safe nodes = 8.9
Effect of Greedy-Epsilon Policy

- Epsilon (ε) indicates probability of taking random action, instead of greedy (best) action

- Increasing epsilon results in worse performance

- Best is ε = 0.0 with Mean=16.6 & Median = 16.8

- But ε = 0.05, which we use, is almost as good with Mean = 16.4 and Median = 16.7
Effect of Scaling Up Training

- Increased number of training episodes to 200k

- Results still good but not better:
  - Mean no. of safe nodes = 15.9,
  - Median no. of safe nodes = 16.0,
  - Only slightly worse than 10k trials.

- RL agent learns well enough with 10k trials.
Autonomous Cyber Defense Research Impacts

• Efficient, scalable cyber defense system to mitigate impact of cyber-attacks
  • Is RL effective in training agents at required speed and scale?
  • How does performance vary as attacker complexity increases?
  • How transferrable is the learning between environments?

• Improve understanding of response effects on:
  • Mission/services,
  • Performance of network, and
  • Cyber-attackers

• Path towards improved trust in, and acceptance of, autonomous cyber defense systems
Questions ???

Contact Info: Dr. Ahmad Ridley,
adridle@tycho.ncsc.mil