Using Deep Reinforcement Learning for Assessing the Consequences of Cyber Mitigation Techniques on Industrial Control Systems

Terry R. Merz and Romarie Morales Rosado
State University of New York, College of Homeland Security and Cybersecurity
Pacific Northwest National Laboratory

tmerz@albany.edu
romarie.morales@pnnl.gov

Abstract: This paper discusses an in-progress study involving the use of deep reinforcement learning (DRL) to mitigate the effects of an advanced cyber-attack against industrial control systems (ICS). The research is a qualitative, exploratory study which emerged as a gap during the execution of two rapid prototyping studies. During these studies, cyber defensive procedures, known as "*Mitigation,* were characterized as actions taken to minimize the impact of ongoing advanced cyber-attacks against an ICS while enabling primary operations to continue. To execute *Mitigation* procedures, affected ICS components required rapid isolation and quarantining from "healthy" system segments. However today, with most attacks leveraging automation, mitigation also requires rapid decision-making capabilities operating at the speed of automation yet with human-like refinement. The authors settled on the choice of DRL as a viable solution to this problem due to the algorithm's designs which involves "intelligent" decisions based upon continuous learning achieved through a rewards system. The primary theory of this study posits that processes informed by data sources relative to the execution path of an advanced cyber-attack as well as the consequences of deploying a particular *Mitigation* procedure evolve the system into an ever-improving defensive capability. This study seeks to produce a defensive DLR based software agent trained by a DRL based offensive software agent that generates policy refinements based upon extrapolations from a corrupted network state as reported by an IDS and baseline data. Results include an estimation rule that would quantify impacts of various mitigation actions while protecting the operational critical path and isolating an in-progress attack. This study is in a conceptual phase and development has not started. This research questions for this study are: RQ1: Can this software agent categorize correctly an in-progress cyber-attack and extrapolate the potential ICS assets affected? RQ2: Can this software agent categorize novel cyber-attacks and extrapolate a probable attack vector while enumerating affected assets? RQ3: Can this software agent characterize how operations are affected by quarantine actions? RQ4: Can this software agent generate a set of ranked recommended courses of action by effectiveness, and least negative effects on the operational critical path?

Keywords*:* Deep reinforcement learning, APT, ICS, Mitigation, AI based automated attacks, ICS cyber-attacks, energy security, cybersecurity

1. Introduction

The conceptual framework for this study emerged as a research gap during the execution of two Department of Defense rapid prototyping studies called JBASICS (2017) and MOSAICS (2021). The first study, JBASICS involved the development of written tactics, techniques, and procedures (TTP) that ICS operators could use to detect, mitigate, and recover from an advanced cyber-attack (J-BASICS 2017). The second study, MOSAICS sought to automate a set defined during the JBASICS study (DoD JCTD 2021). In both studies, *Mitigation* was characterized as actions taken to minimize the negative effects of an on-going advanced cyber-attack against an industrial control system (ICS) while simultaneously enabling the organization to maintain primary operations. However, *Mitigation* procedures were only lightly treated during the execution of the two studies as baselining and detection technologies for ICS were evolving and had not reached a maturity level that would allow the implementation of *Mitigation techniques* as defined by the programs (DoD MOSAICS JCTD 2021).

Lastly, to effectively execute a *Mitigation* procedure, ICS components compromised by an advanced cyber-attack would require rapid isolation and quarantining from "healthy" system segments. However, this would need a decision-making capability that operates at the speed of an automated decision-making process, yet has a human-like refinement, and nuanced understanding around such decisions (DoD MOSAICS JCTD 2021).

2. Research Design

As described by the CISA Alert AA22-103A (2022) adversarial tactics are continuously evolving. Therefore, the Merz/Morales research team hypothesizes that success of a *Mitigation* technology rests on adaptive processes. Additionally, the research team deducted this requirement precludes the use of rule-based algorithms such as

Decision Trees, or even potentially traditional Machine Learning, which requires training sets. However, both deep learning and reinforcement learning represent forms of autonomous learning that apply continuous learning (Nguyen, Reddi 2019).

With deep learning requiring pattern inputs to train the algorithm, reinforcement training offers a solution using trial and error with rewards (Nguyen, Reddi 2019). Both approaches have benefits and drawbacks. However, the research team hypothesized that when used in combination, i.e., Deep Reinforcement Learning (DRL) could potentially achieve the decision-making refinement described and required in the *Mitigation* conceptual framework used in both the J-BASICS and MOSAICS prototyping efforts (J-BASICS 2017).

DRL has demonstrated rapid decision-making capabilities in areas such as robotics, autonomous surgery, autonomous vehicles, biological data mining and drug design (Nguyen, Reddi 2019). Recent developments in cybersecurity research relative to the Internet of Things (IoT) have been centered around the use of DRL and have produced noteworthy results such as a DRL based resource allocation framework for "Smart Cities" that successfully integrates processes from networking, caching, and computing (Ferdowsi, et al., 2018). To that end, this study seeks to extend existing research using DRL in cybersecurity and to produce a software agent that generates policy refinements based upon extrapolations from a corrupted network state as reported by an IDS and baseline data. These corrupted states would have been produced by an offensive software agent also using DRL. The defensive software agent will include an estimation rule that quantifies the impacts of the various possible *Mitigation* actions on protecting the operational critical path while isolating the in-progress attack.

Success criteria: While the success of this exploratory study is framed within the context of the 4 research questions, the possibilities exist that an offensive software agent training a defensive software agent will generate previously undefined outcomes. Therefore, the success of this study is not limited to the research questions being answered in the affirmative but includes a collection of additional system behaviors that will be evaluated from a qualitative perspective.

Validation: After training the computational model for proper classification we will move to model validation. The computational model is verified against the first set of real data and re- verified against a second or subset of real data.

3. Research Team:

The research team involves a partnership between the College of Emergency Preparedness, Homeland Security and Cybersecurity (CEHC) at the University at Albany and the Pacific Northwest National Laboratory (PNNL). These organizations bring together a cross disciplinary team consisting of cybersecurity subject matter experts relative to industrial control systems, and subject matter experts in Machine Learning.

4. Limitations of the Study:

This study is limited to industrial control systems and more specifically to electrical systems. The limitation is driven by the ability to baseline and model system behaviors that have regular, predictable traffic patterns.

5. Research Plan:

While the research team has determined that DRL in general is the most logical approach to solving the *Mitigation* problem, there are many algorithms that can be used to implement it. The two approaches selected for this study are:

Table 1: Deep Reinforcement algorithm candidates (Nguyen, Reddi 2019)

Proposed Agent Name	Application Focus Area	Algorithm	Model	Actions	Rewards
Mitigation-TRPO	Increasingly improved defensive capabilities against cyber actors: devises filtering schemas to detect corrupted states and respond to adversarial actions.	Trust Region Policy Optimization (TRPO)	Model-Free	Determines which extrapolation rule to apply to generate an estimated viable operational state from a corrupted state.	Function: using state inputs
Mitigation-DDQN+A3C	Autonomous defense to include in Software	Double Deep Q Network and Asynchronous	Model-Free	Leverages dueling agents: attack agent learns to select most viable attack vector	Function: rewards based upon

Proposed Agent Name	Application Focus Area	Algorithm	Model	Actions	Rewards
	Defined Networks (SDN)	Advantage Actor-Critic Algorithm		while defending agent can take four actions: isolate, patch, reconnect or migrate to protect an asset and preserve as many nodes along the operational critical path.	the status of the assets along the critical path.

With the Mitigation DDQN+AC3 approach likely to produce the desired agent, the Mitigation-TRPO process would allow a Human-in-the-Loop to make an informed decision about proposed Mitigation actions. Therefore, we should explore using both to achieve different objectives of the Mitigation process.

6. Research Phases with Expected Outcomes:

The conceptual framework for this study is centered around the possibility that an industrial control system can be segmented in such as manner as to isolate an incoming cyber-attack while maintaining primary systems operations.

Using the Perdue reference model as the basis for the study, it is assumed the defensive software agent will focus on segmenting network traffic by one of the following methods (or a combination thereof) by user accounts, processes, applications, and protocols. It is also expected the defensive software agent may identify a physical action to be the most reasonable and will thus alert the operator to physically disconnect a component of the network.

Conversely, the offensive software agent, will mimic the behaviors of actual automated attacks having been trained in cyber-attack patterns typical of an advanced persistent threat. It will identify user accounts, processes, applications, and protocols for exploitation with the objective of terminating primary systems operations. If either software agent (offensive or defensive) are successful, a point award is applied to the software agent, and the agent "learns" the methodology used was successful and the method is captured as a sample of desired actions.

The inclusion of a reward system in training the respective software agents differs from traditional Deep Learning in that the software agents learn on a continuous basis. This approach represents a next level of intrusion detection that not only identifies activities that could lead to a corrupted network state (behavior observation), but also extrapolates courses of actions operators could immediately take to defend an industrial control system and maintain primary operations.

Figure 1 illustrates the communications process by which the defensive and offensive software agents interact with the industrial control system environment.

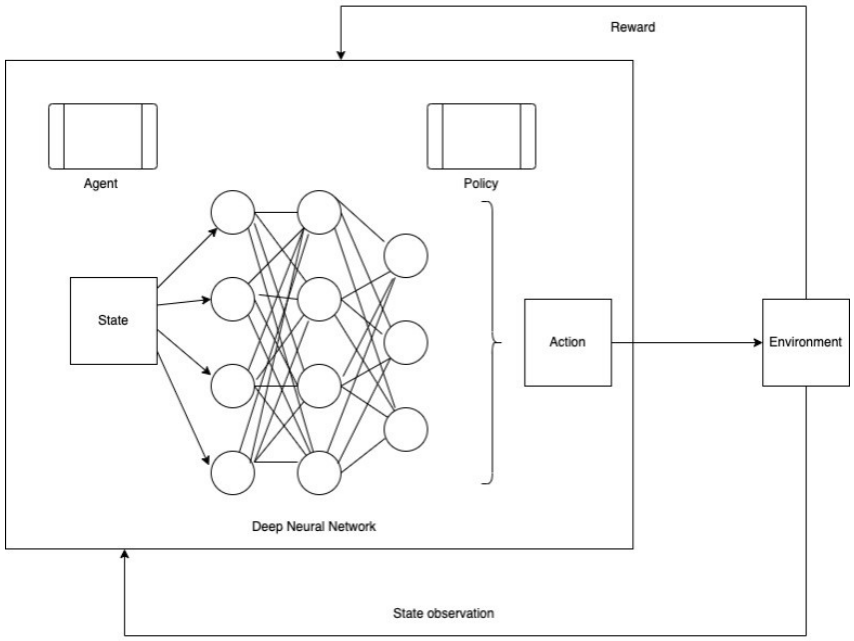

Figure 1: Conceptual Framework (Merz/Morales 2022) - Offensive/Defensive DRL Operations

7. In Conclusion:

Once funding is identified and secured, this study will produce 2 software agents using deep reinforcement learning: an offensive and a defensive agent. The project's success trajectory will be evaluated on a continuous basis throughout the execution of the study via a project risk management plan. Quarterly reviews will be conducted to determine the progress of the study's critical path, potential blockers, and mitigations to blockers. Since the project will be using an Agile development process, regular "hot washes" will be conducted at the conclusion of each sprint and procedures adjusted as needed. The strategic success of the project will be evaluated by the number and quality of peer-reviewed journal papers, follow on research, resultant thesis and dissertations, and acceptance into a transition to practice study resulting in the final fielding of the defensive software agent.

The project success metrics will be included in final reports, which will also address unexpected events, successes, failures, completeness of assumptions, and insights into future applications of the defensive agent.

The resultant models, artifacts and findings from this project are designed to first and foremost produce the foundations of a mitigation capability (in the form of a defensive software agent). This will position the defensive software agent for a transition to practice. A potential target framework for this transition is the Integrated Adaptative Cyber Defense framework, or IACD. The development of this framework was sponsored by the National Security Agency and developed by Johns Hopkins Applied Physics Laboratory (JHU APL N/D).

References

Cybersecurity & Infrastructure Security Agency (2022) "APT Cyber Tools Targeting ICS/SCADA Devices", Alert (AA22-103A), U.S. Department of Homeland Security, Washington D.C.

Department of Defense (DoD), Joint Technical Demonstration (JCTD) (2021) "More Situational Awareness for Industrial Control systems (MOSAICS)", Washington D.C.

Ferdowsi, A., Challita, U., Saad, W. and Mandayam, N.B., "Robust deep reinforcement learning for security and safety in autonomous vehicle systems," in 21st International Conference on Intelligent Transportation Systems (ITSC), 2018, pp. 307-312.

Johns Hopkins Applied Physics Laboratory (JHU APL), (N/D) "About IACD", https://www.iacdautomate.org/aboutiacd, Laurel, Maryland.

Nguyen, T.T., Nguyen, C.M., Nguyen, D.T., Nahvandi, S. (2019), "Deep Learning for deepfakes creation and etection: a survey" arXiv preprint arXiv:1909.11573.

Nguyen, T.T., Reddi, V.J. (2019) "Deep reinforcement Learning for Cybersecurity". IEEE Transaction on Neural networks and Learning Systems, 2021. https://doi.org/10.1109/TNNLS.2021.3121870.

United States Cyber Command, Department of Defense (2017) "Joint Base Architecture for Security Industrial Control Systems" (J-BASICS), Fort Meade, Maryland.