REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
12-01-2023	Final Report	29-Jul-2018 - 24-Jan-2022

4. TITLE AND SUBTITLE
Final Report: W911NF-17-S-0002: A Framework for Asymmetric Information Interactions among (Cyber) Defenders and Attackers

5a. CONTRACT NUMBER
W911NF-18-1-0288

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
611102

6. AUTHORS

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
University of Georgia Research Foundation,
200 D. W. Brooks Drive

Athens, GA 30602 -5016

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
71952-NC.9

12. DISTRIBUTION AVAILIBILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Prashant Doshi
UU	UU	UU	UU		19b. TELEPHONE NUMBER 706-542-2911

RPPR Final Report
as of 19-Jan-2023

Agency Code: 21XD

Proposal Number: 71952NC **Agreement Number: W911NF-18-1-0288**
INVESTIGATOR(S):

Name: Prashant Doshi
Email: pdoshi@uga.edu
Phone Number: 7065422911
Principal: Y

Organization: **University of Georgia Research Foundation, Inc.**
Address: 200 D. W. Brooks Drive, Athens, GA 306025016
Country: USA
DUNS Number: 004315578 EIN: 581353149

Report Date: 24-Apr-2022 Date Received: 12-Jan-2023
Final Report for Period Beginning 29-Jul-2018 and Ending 24-Jan-2022
Title: W911NF-17-S-0002: A Framework for Asymmetric Information Interactions among (Cyber) Defenders and Attackers
Begin Performance Period: 29-Jul-2018 **End Performance Period:** 24-Jan-2022
Report Term: 0-Other

Submitted By: Prashant Doshi Email: pdoshi@uga.edu
Phone: (706) 542-2911

STEM Degrees: 3 **STEM Participants:** 5

Major Goals: Engaging and deceiving attackers into intruding controlled systems and accessing obfuscated data offers a proactive approach to computer and information security. It wastes attacker resources and potentially misleads the attacker. Importantly, it also offers an untapped opportunity to understand attackers' beliefs, capabilities, and preferences and how they evolve by sifting and mining the detailed activity logs. Identifying these mental and physical states not only informs the defender about the attacker's intent, but also guides new ways of deceiving the attacker . In order to establish a formal understanding of deception, this research will build a general framework for computationally modeling interactions between asymmetric adversaries, which, among other uses, is expected to offer a principled basis for deception. The framework will be used to study various interactions between cyber defenders and attackers, with the ultimate goal of foundationally modeling cyber deception and improving its efficacy.

Specific goal #1: Build a computational framework for modeling non-cooperative interactions that take place over time between boundedly-rational and heterogeneous agents.

These agents differ in their beliefs, capabilities and preferences, and may have coarse knowledge of each other's parameters. Consequently, they imperfectly reason about others' strategies. Agents may differ in their coarseness, which leads to some being more strategic than others. In the context provided by this framework, we will further study epistemic and social factors such as persuasion and fundamental attribution error that generally contribute to deception.

Specific goal #2: Demonstrate the situational and mental conditions under which various cyber deception mechanisms available to the defender (e.g., honey pots and tokens) elicit the desired effect.

Both defender and attacker act per their beliefs and valuations in the multiagent interaction. Our main method within the framework involves each agent choosing actions that maximize its long-term expected utility given its belief over the state of the system and other's behavior and how it could evolve. This decision-theoretic approach offers the benefit that it makes few assumptions on others' knowledge and behavior, and is amenable to integrating human behavioral heuristics.

Specific goal #3: Deploy the framework on a live server and track attackers' mental states in situ and in real time while using the actions prescribed by the framework to respond to the adversaries' actions.

RPPR Final Report
as of 19-Jan-2023

We will begin by analyzing anonymized data sets related to cyber intrusions in actual computer systems and activity logs of locally deployed servers to facilitate building the space of candidate attacker types. An attacker's behavior manifests in the system call logs. A pipeline of graphical and machine learning tools can sift through the log data for tracing attack behavior, construct a higher-level process abstraction of the behavior, and utilize it to identify the type of attack. Subsequently, this same pipeline can be deployed to generate observations, albeit noisy, of the attacker's actions for input to the framework.

Accomplishments: SPECIFIC GOAL #1

We developed a computational framework for modeling host-based attacker-defender interactions. We modeled the attacker-defender interactions on a host as a sequential decision-making problem in a two-agent context. We introduced a factored variant of the well-known finitely-nested interactive POMDP framework called I-POMDP_X and use this framework to model the interaction with multiple attacker types. These attacker types were modeled as POMDPs and include:

- Data exfiltration where an agent enters the system unauthorized by exploiting a known weakness, searches for files containing some sensitive information, and on locating them transfers them to a different machine
- Persistence where an agent enters the system unauthorized by exploiting a known weakness, seeks to escalate its privileges by exploiting known weaknesses, and maintains a covert presence in the system
- Data modification occurs when an unauthorized agent enters the system by exploiting a known vulnerability, searches for files with sensitive information, and proceeds to actively escalate its privileges using a weakness to edit the data contained in sensitive files.

The framework computationally models the decision making of the defender while reasoning about the attacker's beliefs and capabilities as it acts and observes. We use this framework to model cyber attacks on a single honeypot host across multiple phases from the attacker's initial entry into the host to reaching its adversarial objective.

The defending I-POMDP_X-based agent uses decoys to engage with the attacker at multiple phases to form increasingly accurate predictions of the attacker's behavior and intent. We evaluated the performance of I-POMDP_X in promoting active deception with multiple attacker types both in simulation and on a real host. Our results show that the I-POMDP-based agent learns the intent of the attacker much more accurately compared to baselines that do not engage the attacker or immediately deploy all decoys en masse. In our final evaluations, the 'No decoy' and 'All decoys' yielded a mean (+- std err.) of 4.42 +- 0.16 and 2.93 +- 1.11 steps of engagement with the attacker, respectively. The longest engagement among these consisted of 7 and 6 steps, respectively. With 'No decoy', the attacker spends time searching for data and attempting to escalate his privileges but without much success, finally exiting the system. With 'All decoys', the attacker either quickly exploits the vulnerabilities or encounters the data decoys but quickly exits often due to the encountered data not being as expected. However, the I-POMDP_X agent engaged with the attacker for a mean duration of 5.81 +- 1.7 with the longest interaction happening for 9 steps. It leverages the information gained by the first few observations to avoid using decoys that the attacker would find suspicious.

Do the extended engagements facilitated by the I-POMDP_X agent help in intent recognition? The defender's I-POMDP X policy eventually yields the lowest cross-entropy values (between the defender's belief of the attacker's frame and the attacker's true type) compared to the baselines, often reaching zero in 6 steps. We show the cross-entropy for more steps because the attacker remains in the system performing a few more actions.

To demonstrate deployment, we also evaluated the deception policy generated by I-POMDP_X in simulations and on an actual system consisting of a standalone attacker programmed via Metasploit and a defender workstation. For the simulations, we randomly sample the attacker type and the starting privileges of the attacker to simulate a threat with unknown intentions and privileges. The defender begins knowing about the existence of decoys on the system. The attacker, on the other hand, does not have prior knowledge about any vulnerabilities or data on the system. The defender engages with the attacker by deploying decoys, facilitating deceptive observations, or adding known vulnerabilities to the system.

SPECIFIC GOAL #2

AI-based cyber-deception commonly ascribes rational behavioral models to attackers. However, many cyber-attacks are primarily orchestrated by human actors. It is well-known that human decision-making deviates from rational behavior due to the influence of cognitive biases. Hence, we focused on the

human elements of the attacker's decision-making process, which may lead to suboptimal behavior due to cognitive biases. Specifically, we modeled the effects of fundamental attribution error (FAE) and confirmation bias on the attacker's beliefs. Previous work has shown that these biases play a role in humans being deceived. FAE is the tendency of an observer to overestimate interpersonal factors compared to environmental factors. The phenomenon of overweighting positive confirmatory evidence leads to the agent overweighting observations that conform to her predicted belief state.

We extended the I-POMDP_X framework with cognitive models of the opponent to study the effects of FAE and confirmation bias. We show FAE as a consequence of coarse thinking. Whereas a rational attacker can distinguish between host types (i.e., host is a honey pot or it is actually a critical host), we assume that a human attacker is unable to distinguish between the two. This leads to the attacker thinking that there is no defender though the host actually has a passive defender. We formalized this reasoning in the context of I-POMDP_X and illustrated it using a simple scenario consisting of a strategic attacker and defender.

Attackers generally start in a system having very little knowledge about the resources and vulnerabilities present in the system. Hence, the initial observations that they receive play a crucial role in helping them form a hypothesis about their environment. We aimed to specifically exploit the role of confirmation bias in such a situation. Specifically, once the attacker forms a belief about the defender's frame, subsequent observations that contradict the attacker's belief are weighted less by the attacker. Next, we showed using the illustrative scenario how confirmation bias can cause the attacker to discount the defender's observed behavior.

Further experimentation on the previously developed cyber deception domain remains pending.

SPECIFIC GOAL #3

We developed a novel AI-based methodology that identifies phases of a host-level cyber attack simply from system call logs. System calls emanating from cyber attacks on hosts such as honey pots are often recorded in audit logs. Our methodology involves the following general steps:
(i) Efficiently load, cache, process, query, and display system events from audit
logs to support computer forensics. Output of queries should be provenance graphs, which can be processed for further analysis.

(ii) Resulting trace is often still hard to parse and in need of further abstraction to facilitate analyses. Utilize a latent-state probabilistic model, which allows us to infer the most likely sequence of higher-level actions, which we call an attack storyline, while modeling system calls as observations.

(iii) Finally, we seek to identify the attack phase based on the sequence of high-level actions inferred in the previous step. We view this step as a multi-class classification problem where each label is a phase of an attack.

To translate our methodology to practice, we evaluated each step with candidate methods or models. An independent red team simulated several types of attacks on a server acting as a honey pot. This yielded a total of 114 attack phases, which included asset discovery and data exfiltration, among others. The pipeline of selected models lead to a system, which we call Cyberian, that performs very well in identifying the attack phases as measured by an F1-score of 90.31 ± 8.44. Additionally, we tested on a few attack sequences discovered in a recent data set released by DARPA and reported positive results.

Training Opportunities: This project during the entire performance period provided opportunities for training 2 PhD students and 3 Masters students in the topics of cybersecurity, AI, and broader STEM topics with a focus on cyber deception and security forensics.

One of these PhD students successfully defended his dissertation (in the area of AI, cybersecurity, and IoT) and graduated with his PhD degree in December 2021. He has joined Facebook's.

Two of the Masters students also successfully defended their thesis (both on work directly supported by this grant) and graduated. Both are pursuing further graduate studies in the area of cyber security.

Results Dissemination: Outcomes from our research were disseminated through 3 published papers -- two in refereed conferences and one in a journal.

1. Setayeshfar, O, Adkins, C., Jones, M., Lee, K. H., Doshi, P. (2021). GrAALF: Supporting graphical analysis of audit logs for forensics", Software Impacts, Volume 8, 100068, ISSN 2665-9638, https://doi.org/10.1016/j.simpa.2021.100068.

2. AbuOdeh, M., Adkins, C., Setayeshfar, O., Doshi, P., & Lee, K. H. (2021). A Novel AI-based Methodology for Identifying Cyber Attacks in Honey Pots. Proceedings of the Innovative Applications of AI Conference (IAAI/AAAI), 35(17), 15224-15231.

3. Shinde, A., Doshi, P., and Setayeshfar, O. (2021). Cyber Attack Intent Recognition and Active Deception using Factored Interactive POMDPs. Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), 1200–1208. **Winner of the best application paper award**

The Java implementation of I-POMDP_X called Protos was released on GitHub under an open-source license. Protos is available here for download: https://github.com/dityas/Protos

Graduate students Aditya Shinde, Omid Setayeshfar, and PI Prashant Doshi also appeared in a podcast interview on AI-based cyber defense using deception and attacker intent recognition produced by Smokescreen, a cyber deception solutions company. The podcast can be accessed at https://www.youtube.com/watch?v=wyqMUqGEGVY

Honors and Awards: The following published paper on the I-POMDP_X model and its use in attacker intent recognition

Shinde, A., Doshi, P., and Setayeshfar, O. (2021). Cyber Attack Intent Recognition and Active Deception using Factored Interactive POMDPs. Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), 1200–1208.

won the Best Application Paper Award at the AAMAS 2021, which is a top-tier conference on autonomous agents.

https://aamas2021.soton.ac.uk/awards/best-paper-awards/

Protocol Activity Status:

Technology Transfer: Nothing to Report

PARTICIPANTS:

Participant Type: PD/PI
Participant: Prashant Doshi
Person Months Worked: 4.00 **Funding Support:**
Project Contribution:
National Academy Member: N

Participant Type: Co PD/PI
Participant: Kyu Hyung Lee
Person Months Worked: 2.00 **Funding Support:**
Project Contribution:
National Academy Member: N

Participant Type: Co PD/PI

RPPR Final Report
as of 19-Jan-2023

Participant: Adam Goodie
Person Months Worked: 1.00 **Funding Support:**
Project Contribution:
National Academy Member: N

Participant Type: Graduate Student (research assistant)
Participant: Aditya Shinde
Person Months Worked: 15.00 **Funding Support:**
Project Contribution:
National Academy Member: N

Participant Type: Graduate Student (research assistant)
Participant: Omid Setayeshfar
Person Months Worked: 3.00 **Funding Support:**
Project Contribution:
National Academy Member: N

Participant Type: Graduate Student (research assistant)
Participant: Christian Adkins
Person Months Worked: 15.00 **Funding Support:**
Project Contribution:
National Academy Member: N

Participant Type: Graduate Student (research assistant)
Participant: Mohammad AbuOdeh
Person Months Worked: 10.00 **Funding Support:**
Project Contribution:
National Academy Member: N

Participant Type: Graduate Student (research assistant)
Participant: Sean Frankum
Person Months Worked: 1.00 **Funding Support:**
Project Contribution:
National Academy Member: N

ARTICLES:

RPPR Final Report
as of 19-Jan-2023

Publication Type: Journal Article Peer Reviewed: Y **Publication Status:** 1-Published
Journal: Software Impacts
Publication Identifier Type: DOI Publication Identifier: 10.1016/j.simpa.2021.100068
Volume: 8 Issue: First Page #: 100068
Date Submitted: 8/5/21 12:00AM Date Published: 5/1/21 4:00AM
Publication Location:
Article Title: GrAALF: Supporting graphical analysis of audit logs for forensics
Authors: Omid Setayeshfar, Christian Adkins, Matthew Jones, Kyu Hyung Lee, Prashant Doshi
Keywords: cyber forensics, provenance tracking, log analysis
Abstract: System-level logs play a critical role in computer forensics. They capture interactions between programs and users in detail. However, a typical computer generates more than 2.5 million system events hourly, making finding malicious activities in such logs compute and time-intensive. We introduce GrAALF a graphical system for efficiently loading, storing, processing, querying, and displaying system events for computer forensics. In comparison to similar systems, GrAALF offers the flexibility of storage, intuitive querying, and the tracing power for longer sequences of events in real-time to help identify attacks. GrAALF is a robust solution for analysis to support computer forensics.
Distribution Statement: 1-Approved for public release; distribution is unlimited.
Acknowledged Federal Support: **Y**

Publication Type: Journal Article Peer Reviewed: Y **Publication Status:** 1-Published
Journal: AAMAS '21: Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems
Publication Identifier Type: ISBN Publication Identifier: 9781450383073
Volume: Issue: First Page #: 1200
Date Submitted: 8/5/21 12:00AM Date Published: 5/3/21 4:00AM
Publication Location:
Article Title: Cyber Attack Intent Recognition and Active Deception using Factored Interactive POMDPs
Authors: Aditya Shinde, Prashant Doshi, Omid Setayeshfar
Keywords: cyber deception, POMDP, honey pots
Abstract: This paper presents an intelligent and adaptive agent that employs deception to recognize a cyber adversary's intent on a honeypot host. Unlike previous approaches to cyber deception, which mainly focus on delaying or confusing the attackers, we focus on engaging with them to learn their intent. We model cyber deception as a sequential decision-making problem in a two-agent context. We introduce factored finitely-nested interactive POMDPs (I-POMDPX) and use this framework to model the problem with multiple attacker types. Our approach models cyber attacks on a single honeypot host across multiple phases from the attacker's initial entry to reaching its adversarial objective. The defending I-POMDPX-based agent uses decoys to engage with the attacker at multiple phases to form increasingly accurate predictions of the attacker's behavior and intent. The use of I-POMDPs also enables us to model the adversary's mental state and investigate how deception affects their beliefs.
Distribution Statement: 1-Approved for public release; distribution is unlimited.
Acknowledged Federal Support: **Y**

RPPR Final Report
as of 19-Jan-2023

Publication Type: Journal Article Peer Reviewed: Y **Publication Status:** 1-Published
Journal: IAAI-21: Proceedings of the 33rd Conference on Innovative Applications of AI
Publication Identifier Type: ISBN Publication Identifier: 978-1-57735-866-4
Volume: 35 Issue: 17 First Page #: 15224
Date Submitted: 8/5/21 12:00AM Date Published: 5/18/21 4:00AM
Publication Location:
Article Title: A Novel AI-based Methodology for Identifying Cyber Attacks in Honey Pots
Authors: Muhammed AbuOdeh, Christian Adkins, Omid Setayeshfar, Prashant Doshi, Kyu H. Lee
Keywords: Attack Phases, Cyber Forensics, Machine Learning, Methodology
Abstract: We present a novel AI-based methodology that identifies phases of a host-level cyber attack simply from system call logs. System calls emanating from cyber attacks on hosts such as honey pots are often recorded in audit logs. Our methodology first involves efficiently loading, caching, processing, and querying system events contained in audit logs in support of computer forensics. Output of queries remains at the system call level and is difficult to process. The next step is to infer a sequence of abstracted actions, which we colloquially call a storyline, from the system calls given as observations to a latent-state probabilistic model. These storylines are then accurately identified with class labels using a learned classifier. We qualitatively and quantitatively evaluate methods and models for each step of the methodology using 114 different attack phases collected by logging the attacks of a red team on a server, on some likely benign sequences containing regular user activities,
Distribution Statement: 1-Approved for public release; distribution is unlimited.
Acknowledged Federal Support: **Y**

CONFERENCE PAPERS:

Publication Type: Conference Paper or Presentation **Publication Status:** 4-Under Review
Conference Name: 10th ACM Conference on Data and Application Security and Privacy (CODASPY).
Date Received: 16-Aug-2019 Conference Date: 16-Mar-2020 Date Published:
Conference Location: New Orleans, Louisiana
Paper Title: GrAALF: Supporting Graphical Analyses of Audit Logs for Forensics
Authors: Omid Setayeshfar, Christian Adkins, Kyu H. Lee, Prashant Doshi
Acknowledged Federal Support: **Y**

Publication Type: Conference Paper or Presentation **Publication Status:** 4-Under Review
Conference Name: Neural Information Processing Systems
Date Received: 27-Aug-2020 Conference Date: 05-Dec-2020 Date Published:
Conference Location: Online
Paper Title: Active Deception using Factored Interactive POMDPs to Recognize Cyber Attacker's Intent
Authors: Aditya Shinde, Prashant Doshi, and Omid Setayeshfar
Acknowledged Federal Support: **Y**

DISSERTATIONS:

Publication Type: Thesis or Dissertation
Institution: University of Georgia
Date Received: 27-Aug-2020 Completion Date: 7/31/20 3:17PM
Title: ACTIVE CYBER DECEPTION AND ATTACKER INTENT RECOGNITION USING FACTORED INTERACTIVE POMDPS
Authors: Aditya Shinde
Acknowledged Federal Support: **N**

RPPR Final Report
as of 19-Jan-2023

Publication Type: Thesis or Dissertation
Institution: University of Georgia
Date Received: 27-Aug-2020 Completion Date: 7/31/20 4:00AM
Title: Cyber Attack Storyline Generation Using Hidden Markov Models
Authors: Muhammed AbuOdeh
Acknowledged Federal Support: **N**

Publication Type: Thesis or Dissertation
Institution: University of Georgia
Date Received: 05-Aug-2021 Completion Date: 5/1/21 9:36PM
Title: Towards a Non-Discriminatory Security Model Through Analysis of Low Level Data
Authors: Omid Setayeshfar
Acknowledged Federal Support: **N**

Partners

,

I certify that the information in the report is complete and accurate:
Signature: Prashant Doshi
Signature Date: 1/12/23 7:09PM

Accomplishments for the entire research project

SPECIFIC GOAL #1: Build a computational framework for modeling non-cooperative interactions that take place over time between boundedly-rational and heterogeneous agents.

August 1, 2018 to July 31, 2019

Objective 1: Model different types of attacks as POMDPs

Our computational framework for modeling interactions between asymmetric adversaries requires behavioral models of various types of attacks. These models may be intentional representing the attacker's beliefs, capabilities, and preferences, or subintentional modeling the attacker's actions using a probability distribution or a finite state automaton. As we seek to demonstrate the situational and *mental* conditions under which various cyber deception mechanisms available to the defender can be deployed, we model the attacker as an intentional agent using the POMDP framework. Parameters of a POMDP include the agent's belief, transition and observation functions (capabilities), and the reward function.

We are focusing on three popular types of attacks:

- Data exfiltration where an agent enters the system unauthorized by exploiting a known weakness, searches for files containing some sensitive information, and on locating them transfers them to a different machine over the network
- Persistence where an agent enters the system unauthorized by exploiting a known weakness, seeks to escalate its privileges possibly by again exploiting known weaknesses, and maintains a covert presence in the system
- Data modification occurs when an unauthorized agent enters the system by exploiting a known vulnerability, searches for files with sensitive information, and proceeds to actively escalate its privileges using a weakness in order to edit the data contained in one or more sensitive files.

We formulated POMDP-based intentional models for each of these attack behaviors. The POMDPs share the same state, action, and observation spaces and the functions but differ in the reward functions representing the differing intents of the attackers. We solved these large models using the SPUDD program[1] that uses algebraic decision diagrams to represent the POMDP parameters efficiently while promoting scalability in problem size. Solutions of the POMDP-based models are finite state automata, where the nodes represent prescribed actions associated with specific ranges of the agent's belief and the edges are conditions labeled with observations. We show the finite state automata for the three types of attacks in **Figure 1** (*a*), (*b*), and (*c*) (see the appendix).

August 1, 2019 to July 31, 2020

Objective 2: A computational framework for modeling host-based attacker-defender interactions

[1] Jesse Hoey, Robert St-Aubin, Alan J Hu and Craig Boutilier *SPUDD: Stochastic Planning Using Decision Diagrams*. Proceedings of Uncertainty in Artificial Intelligence, Stockholm, Sweden, 1999

We model attacker-defender interactions on a host as a sequential decision-making problem in a two-agent context. We introduce a factored variant of the well-known finitely-nested interactive POMDPs (I-POMDP_X) and use this framework to model the interaction with multiple attacker types. These attacker types were developed in the previous year and modeled as POMDPs. It computationally models the decision making of the defender while reasoning about the attacker's beliefs and capabilities as it acts and observes. We use this framework to model cyber attacks on a single honeypot host across multiple phases from the attacker's initial entry into the host to reaching its adversarial objective.

The defending I-POMDP_X-based agent uses decoys to engage with the attacker at multiple phases to form increasingly accurate predictions of the attacker's behavior and intent. We evaluate the performance of I-POMDP_X in promoting active deception with multiple attacker types both in simulation and on a real host. Our results show that the I-POMDP-based agent learns the intent of the attacker much more accurately compared to baselines that do not engage the attacker or immediately deploy all decoys en masse. We evaluate the deception policy generated by I-POMDP_X in simulations and on an actual system consisting of a standalone attacker programmed via Metasploit and a defender workstation. For the simulations, we randomly sample the attacker type and the starting privileges of the attacker to simulate a threat with unknown intentions and privileges. The defender begins knowing about the existence of decoys on the system. The attacker, on the other hand, does not have prior knowledge about any vulnerabilities or data on the system. The defender engages with the attacker by deploying decoys, facilitating deceptive observations, or adding known vulnerabilities to the system.

The 'No decoy' and 'All decoys' yielded a mean (+- std err.) of 4.30 +- 0.16 and 3.26 +- 0.20 steps of engagement with the attacker, respectively. The longest engagement among these consisted of 7 and 5 steps, respectively. With 'No decoy', the attacker spends time searching for data and attempting to escalate his privileges but without much success, finally exiting the system. With 'All decoys', the attacker either quickly exploits the vulnerabilities or encounters the data decoys but quickly exits often due to the encountered data not being as expected. However, the I-POMDP_X agent engaged with the attacker for a mean duration of 5.90 +- 0.24 with the longest interaction happening for 9 steps. It leverages the information gained by the first few observations to avoid using decoys that the attacker would find suspicious. Do the extended engagements facilitated by the I-POMDP_X agent help in intent recognition? **Figure 3 in the appendix** shows the cross-entropy between the defender's belief of the attacker's frame and the attacker's true type, as it varies across the steps of the interaction. The defender's I-POMDP X policy eventually yields the lowest cross-entropy values compared to the baselines, often reaching zero in 6 steps. We show the cross-entropy for more steps because the attacker remains in the system performing a few more actions.

<u>August 1, 2020 to July 31, 2021</u>

Objective 3: Finalize evaluations of the I-POMDP$_X$ model and publish the results

Final experimentation revealed that the NO-OP(no decoy), which does not engage with the attacker, and NO-OP(all decoy) defense strategy, which deploys all decoys indiscriminately on the host, yielded a mean (± std dev.) of 4.42 ± 0.16 and 2.93 ± 1.11 steps of engagement with the attacker, respectively. The longest engagement among these consisted of 7 and 6 steps, respectively. With NO-OP(no decoy), the attacker spends time searching for data and attempting to escalate his privileges but without much

success, finally exiting the system. With NO-OP(all decoys), the attacker either quickly exploits the vulnerabilities or encounters the data decoys but quickly exits often due to the encountered data not being as expected. However, the I-POMDP$_X$ agent engaged with the attacker for a mean duration of 5.81 ± 1.7 with the longest interaction happening for 9 steps. It leverages the information gained by the first few observations to avoid using decoys that the attacker would find suspicious. For example, the defender first manipulates the attacker's observations about her own privileges. This increases the defender's chances of observing file enumeration or vulnerability discovery activity, forming a belief over the frames. Subsequently, the defender baits the attacker using decoys and observes the interaction to solidify his belief. This minimizes the risk of the attacker encountering unexpected decoys or noticing discrepancies.

These simulations are predicated on the level-1 defender believing that none of the level-0 attacker types are aware of the deception, which is the typical case. However, if the defender's strategy level is 2 and it believes that the attacker believes that there is a small chance at 0.1 of decoys being used, we observed that the attacker often quickly exited the system as one would expect.

We published the I-POMDP$_X$ model of attacker-defender interactions and its evaluation toward active defense in the top-tier AAMAS 2021 conference. It was nominated for the best application paper award and proceeded to win the award.

SPECIFIC GOAL #2: Demonstrate the situational and mental conditions under which various cyber deception mechanisms available to the defender (e.g., honey pots and tokens) elicit the desired effect.

August 1, 2019 to July 31, 2020

Objective 1: Situational and mental conditions that promote deception

Our computational framework I-POMDP_X explicitly models the beliefs of the attacker and the defender throughout the interaction. This allows for detailed inferences about how specific deceptive actions affect the attacker's subjective view of the system. **Figure 4(a) in the appendix** illustrates a scenario taken from an actual simulation run with the data manipulator attacker type. Initially, the attacker has a non-zero belief over the existence of data on the system. However, the true state of the system on the left shows that the system does not actually contain any data. In the absence of the defender or any static data decoys, the attacker will eventually update his beliefs to accurately reflect the reality by performing the FILE_RECON_CDATA action, which searches for critical data for manipulation, and observing the result. However, to avoid this belief state, the defender deploys data decoys when the attacker acts. The attacker's inability to tell the difference between decoy data and real data and his prior belief about the absence of decoys leads him to attribute his observations to the existence of real data leading to the attacker being deceived.

Figure 4(b) in the appendix shows another scenario taken from simulations. In this particular scenario, the defender observed a file discovery action in the beginning and deployed critical data decoys. However, subsequent observations made by the defender were inconsistent with the data manipulator type attacker. Hence, the defender switches the decoys before the attacker can spot any discrepancies. The true state of the system is shown on the left. The defender performs REMOVE CDATA DECOYS to

remove the critical data decoys when the C DATA DECOYS state is yes. Simultaneously, the attacker performs FILE RECON SDATA to search for sensitive data. In such a scenario, the defender's action is given priority. Hence the attacker is unable to find sensitive data and the next state shows that C DATA DECOYS has transitioned to no. The attacker is unable to find any data and has a stronger belief that the host might not have any data. As the attacker performs the FILE RECON SDATA action for the last time, the defender deploys sensitive data decoys. In this particular case, the attacker was able to find the decoys and interact with them. In some cases, the attacker is unable to find the decoys despite the defender deploying them due to the imperfect nature of the FILE RECON SDATA action. When this happens, the defender does not observe any decoy interactions and is unable to form an accurate belief over the frame of the attacker.

<u>August 1, 2020 to July 31, 2021</u>

Objective 2: Modeling the cognitive biases involved in deception

Deception generally involves belief manipulation, which comes about because of information asymmetry between the interacting agents. This asymmetry could be due to the cognitive limitations of the deceived agent because of which it is unable to perfectly comprehend the strategy of the other. The attackers being humans themselves are also susceptible to these biases. In cyberattacks orchestrated by human attackers (and not automated malware), these biases can be exploited to manipulate the attackers' beliefs and achieve active defense objectives through deception. Indeed, exploiting cognitive biases can play a central role in employing active defense tactics such as *Channeling* and *Legitimization* mentioned in the MITRE SHIELD matrix.

Confirmation bias One of the biases that can be exploited against cyber attackers is the *confirmation bias*, which is the tendency to favor information or interpret information in ways that reinforce the decision maker's prior beliefs. By exploiting confirmation bias effectively, a defender could make the attacker disregard information or believe in deceptive signals as the situation demands. The confirmation bias manipulates the normative Bayesian belief update, $b'(s) = \beta \Pr(\omega|s) \ \Sigma_{s' \in S} \Pr(s|s')\, b(s')$. Here, *b'(s)* is the posterior distribution, *Pr(ω|s)* is the likelihood of the received evidence ω, and Pr(s|s') is the transition probability representing the dynamics of the system, while *b(s')* represents the prior. We model the confirmation bias in this belief update by adaptively weighing the evidence in the above expression. This is formalized as

$$b'(s) = \beta \Pr(\omega|s)^{\gamma} \Sigma_{s' \in S} \Pr(s|s')\, b(s'), \text{ where } \gamma = \frac{1}{1+distance(\Pr(\omega|S), \hat{b}\{s\})}.$$

Here, $distance\left(\Pr(\omega|s), \hat{b}\{s\}\right)$ is the Euclidean distance between the likelihood vector (of probabilities) and the predicted belief, $\hat{b}(s) = \sum_{s' \in S} \Pr(s|s')\, b(s')$. Notice that γ inversely weights the evidence based on its distance from the predicted belief. In other words, if the observed evidence supports the predicted belief due to which the distance between the observation likelihood from a state and the predicted belief of the state is small, the evidence will be considered in updating the belief of the agent. Otherwise, the evidence will be underweighted and the agent will rely on the dynamics and prior belief to update its own belief.

Fundamental attribution error In cognitive psychology, the theory of mind discerns the mental states including beliefs, intent and preferences of others, which aid in inferring how others will act. However, a surprising neglect in this "mind reading" by humans plays a role in being deceived. Humans often make a *fundamental attribution error* by ignoring the situational context as they seek to infer the mental states behind the other person's observed behavior. In other words, humans may misattribute their observations to the mental states of other agents in the interaction instead of attributing them to circumstances or considering other explanations.

In the case of typed multiagent systems, the error can lead to agents making incorrect inferences about other agents' types. In the context of cybersecurity, a defender can exploit the attribution error of the attacker to manipulate the attacker's beliefs about the capabilities of their opponent. We consider an attacker who exhibits *coarse thinking,* unable to differentiate between decoy and real information. Such an attacker will model the environment as made up of two host types; HOST TYPE = {nc, {c, hp}}. Unable to differentiate between decoy and real information, he categorizes honey pot and critical system as being the same critical host type. The attacker will perform a RECON action to check the type of host and ATTACK only if the host is {c, hp} which is the attacker's coarse representation of a critical system. We show that a level two attacker exhibiting this coarse thinking can be deceived to think that the host (which is actually a honey pot) is not actively defended; therefore, this misattribution of the defender's capabilities leads the attacker to engage with a honey pot.

SPECIFIC GOAL #3: Deploy the framework on a live server and track attackers' mental states in situ and in real time while using the actions prescribed by the framework to respond to the adversaries' actions.

August 1, 2018 to July 31, 2019

Both defender and attacker act per their beliefs, observations, and valuations in the multiagent interaction. To enable this interaction on a live system, we must retrieve the effects of the attacker and defenders' actions, understand how these modify the state of the system, and identify ways for the defender and attacker to sense these changes. Toward this, we are building several applications and models that will together allow the defender to sense and act.

Objective 1: Tool for graphical analysis of log data for forensics

We have developed a graphical software for efficiently loading, storing, processing, querying, and display of system event logs for supporting computer security forensics called GrAALF.

GrAALF is particularly useful to collect and analyze system audit logs for tracing an attacker's behavior. GrAALF offers the choice of compactly storing the logs in main memory, in a relational database system, and in a graph-based database. It allows stored audit logs to be queried using a simple query language that supports path queries and backtracking to an arbitrary depth from an identified resource.

With GrAALF, we collected and analyzed real-world audit logs, and we successfully identified the attacks and their provenance using manually composed queries. We are leveraging GrAALF to design machine learning tools that can automatically construct a higher-level process abstraction of the behavior and utilize it to automatically identify the type of attack.

Objective 2: To infer the attack sequence from log data

GrAALF can sift through massive raw logs to generate the sequence of system calls that lie on a potential attack behavior. It can add to this sequence in real-time as the attack unfolds. This focused trace is nonetheless hard to parse for humans and machines alike.

Notice that the system calls may be perceived as a sequence of observable signals emitted by the system as the adversary carries out its attack. These may be utilized along with probabilistic information about the current state of the attack behavior when a particular system call is emitted and how the attack transitions from one state to another given the execution of the system calls to infer a most-likely explanation for the perceived sequence of system calls. A **hidden Markov model (HMM)** is a probabilistic graphical model particularly well suited for this application.

A red team of researchers carried out 100+ different types of data exfiltration, persistence, and data modification attacks on a test server using Metasploit, which is a well-known penetration testing tool. GrAALF analyzed the corresponding system call log files generated either by *sysdig* or Linux's *audit* tool to obtain focused traces of possible attacks. We trained the HMM on these attacks using the well-known Baum-Welch algorithm, which essentially learns the transition and emission probability tables. We are currently engaged in a systematic evaluation of the performance of the HMM in correctly inferring the attack behavior (which uses the Viterbi algorithm for forward-backward inference). We refer to these inferred attack behaviors as *attack storylines*.

Objective 3: Classify attack sequences and generating observations using machine learning

Our hypothesis is that the attack storylines, similar to unstructured text, contain sufficient latent and observable information to determine the type of attack. Recent literature in cybersecurity has demonstrated the feasibility of using deep neural networks to not only detect anomalous behavior from low-level data but also predict the next step of the attack.[2] On the other hand, the amount of training data available to us is limited by the number of attacks that are available.

In this context, we are experimenting with multiple classification models including a standard one-dimensional convolutional neural network treating each attack sequence as a signal changing over time, a long short-term memory network analyzing overlapping sequences of observations in each attack, and a support vector machine designed to classify using relatively smaller datasets to correctly label the attack storylines with the type of attack. We are also closely monitoring for overfitting given that our current evaluation set has 110 attacks only.

We may utilize this classification model to also tag subsequences with higher-level observations that can be sent to the defender as an essential step in our framework for modeling the defender-attacker interactions.

Moving forward, we will bring these tools and models together on a single system to orchestrate a pipeline that will enable a near real-time analysis and identification of the attack behavior. This will enable the defender to engage with the attacker within the parameters of our framework. An illustration of such a pipeline is shown in **Figure 2** (see the appendix).

[2] M. Du, F. Li, G. Zheng, V. Srikumar, DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning, ACM CCS, 2017.

August 1, 2019 to July 31, 2020

We implemented Cyberian's AI-based methodology for identifying the attack phases from system call logs on a honey pot. To realistically evaluate Cyberian's performance, an independent red team composed of cyber security researchers and assisted by Metasploit (Rapid7, 2020) engaged in several attacks on a Linux server across a span of a few days. These attacks yielded a total of 139 phases, each of which is one of six popular types (based on the sequence of system calls executed by the attacker in each phase), or a benign sequence.

Objective 4: Evaluate the performance of the hidden Markov model for inferring "attack storylines"

For a given (cleaned) sequence of system calls from our log analysis tool GrAALF, we refer to the sequence of states inferred by the HMM as its storyline. To infer attack storylines automatically, our aim is to learn the transition and emission probability tables of the HMM from the sequences of system calls in attacks annotated with abstracted actions using a learning algorithm such as Baum-Welch. This requires relating each state to the observed system call triples emitted by the state. A storyline is then the most likely explanation inferred by the trained HMM for a given sequence of system call observations. The attack storylines are then passed to the next step in Cyberian's pipeline for identification. An example short attack storyline inferred for the asset discovery phase is: (Execute process ircd, Start shell by ircd, Execute process perl, Execute process perl, File recon, Start shell by perl, Execute process temp process, Start shell by temp process, File recon).

The cleaned sequences of system calls from GrAALF yielded 1176 distinct observations in all, for which we utilized an HMM with 17 states. We implemented the HMM using the Pomegranate package coded in Python. We evaluate the HMM's performance using 5-fold cross validation on the annotated sequences in the 139 sequences. Recall that the storyline for a sequence of system calls is the inferred most likely explanation (MLE). In addition to evaluating the fit of the model, we evaluated the correctness of the inferred storylines. We did this by comparing the previous mean log likelihoods with the mean of the log probabilities of the observed sequences in each test fold given the ground truth assignment of states for a sequence. In **Table 1(a) in the appendix**, we report the means and standard deviations of the likelihood ratio (LL of MLE/LL of ground truth) while noting that a ratio of 1 is desired. While none of the folds gave a perfect ratio of 1, we point out that the mean ratio for most folds is relatively close to 1 indicating that the generated storylines were mostly correct. **Table 1(b)** contains the likelihood ratios decomposed by attack phase.

Objective 5: Evaluate the utility of the HMM-inferred attack storylines toward identifying the attack phases

To demonstrate the usefulness of the HMM's abstraction step toward identifying high-level attack characteristics, we evaluate each classifier (i.e., SVM, CNN, and LSTM) in two ways. The first experiment uses the ground truth sequences and the storylines comprised of states that the HMM produced for each attack phase. In the second experiment, the input sequences are system calls directly coming from GrAALF without being processed by the HMM; as there are no storylines for this raw data, a 5-fold cross validation was used to evaluate each model on this data set.

Table 2 gives the mean classification accuracy for all the models in both experiments. First, notice that the use of storylines as input to the classifiers improves their accuracy significantly compared to just using the sequences of low-level system calls; this improvement is especially large for the CNN. As such, the process of inferring the most likely explanation of the observed system calls is valuable and identification of the attack phases benefits from reasoning about the context.

Objective 6: Performance of the classification of attack phases

Various parameters for each classification model were explored to determine the best configurations. In addition to trying different dropout rates as a method to resist overfitting on the training data, using class weights during the training phase resulted in improved accuracy. Measuring the performance on the testing data with more or fewer training iterations narrowed down the appropriate number of epochs. Among the various classifiers, the CNN model achieves significantly better accuracy than the SVM or LSTM classifiers. In other words, the CNN operating on attack storylines is able to accurately identify the type of about 85% of the attack phases.

We further analyze the models' performances with storylines as input by decomposing their classification F1-scores by attack phase. **Table 3 in the appendix** shows that the CNN's weakest performance, at 44%, is on the asset discovery sequences where none of the models achieve a high score. The CNN and SVM achieve similar results overall (the macro-averaged F1-scores across all classes are equal). The significantly lower asset discovery F1-score is likely due, in part, to the sequences of that class having a low average length compared to the rest of the classes. Asset discovery sequences give the models less information to learn from and inform the classification.

Assessing the confidences each model achieves during classification can reveal attack phases that lack clear discriminating features. The CNN and LSTM both have high confidence levels among correct and incorrect classifications, with almost all confidences being near 100% for the LSTM. The SVM has more variety in these scores, and has a clearer distinction between them among right and wrong classifications. Across all three models' incorrect classifications, if the true class was Privilege Escalation (PE) then the predicted class was almost always Asset Discovery (AD). The reverse is also true, with AD instances being misclassified as PE most often. AD is the worst-classified attack phase and PE is one of the weaker phases.

<u>August 1, 2020 to July 31, 2021</u>

Objective 7: Finalize evaluations of Cyberian and publish the results

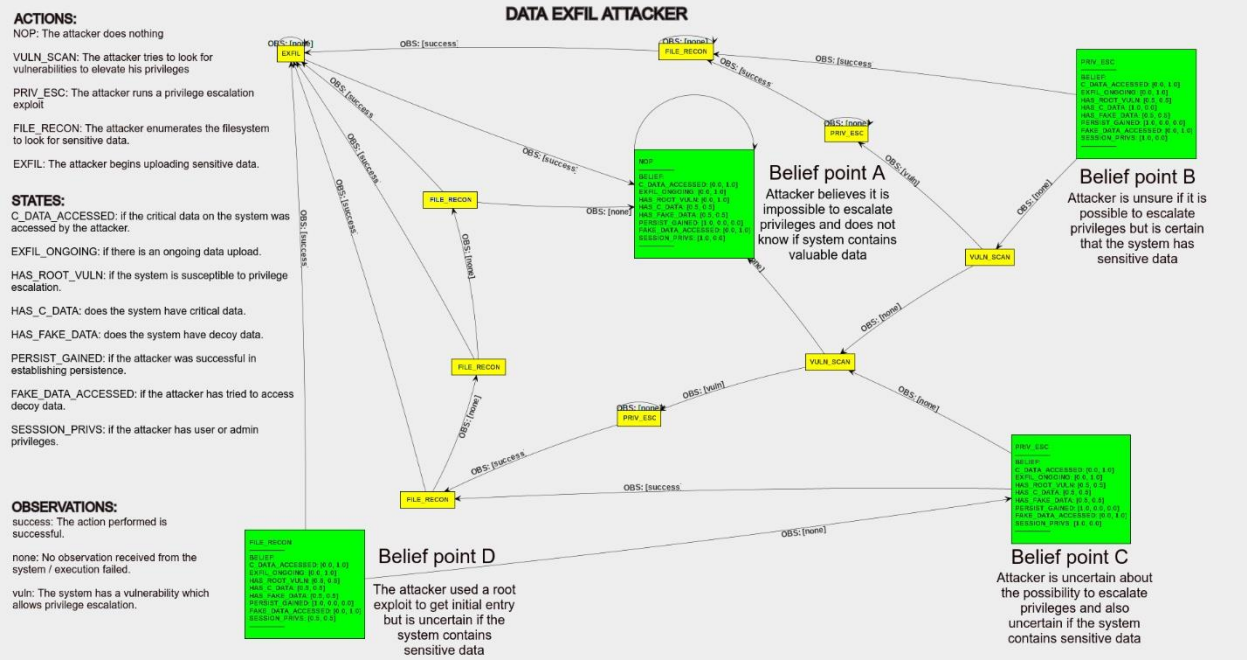
To demonstrate the utility of HMM's abstraction in the Cyberian methodology toward identifying high-level attack characteristics, we evaluate classifiers in two ways: The first experiment trains for up to 100 epochs (maximum iterations of 20K for the SVM) on ground truth attack sequences and evaluates each model's performance on storylines comprised of states that the HMM produced for each attack phase. A ground truth labeled sequence is the sequence of manually annotated states. In the second, input sequences are system calls directly coming from GrAALF without being processed by the HMM; as there are no storylines for this raw data, 5-fold cross validation was used to evaluate models on this data set (confusion matrices for each fold were combined to get final results). Relative results of these tests

show how well HMM predictions reflect true characteristics of each attack phase. **Table 4 in the appendix** gives the weighted-mean F1-score of the classification by each of the models in both experiments. First, notice that the use of storylines as input to the classifiers improves their accuracy significantly compared to using sequences of low-level system calls; this improvement is especially large for the CNN. Among the various classifiers, we note that the LSTM model now achieves a better mean F1-score than the SVM or CNN. The LSTM operating on attack storylines is able to accurately identify the type of about 90% of the attack phases. However, the paired F1-score differences between the LSTM and the other methods are not statistically significant. Clearly, the process of inferring the most likely explanation of the observed system calls is very valuable and identification of the attack phases benefits from reasoning about the context.

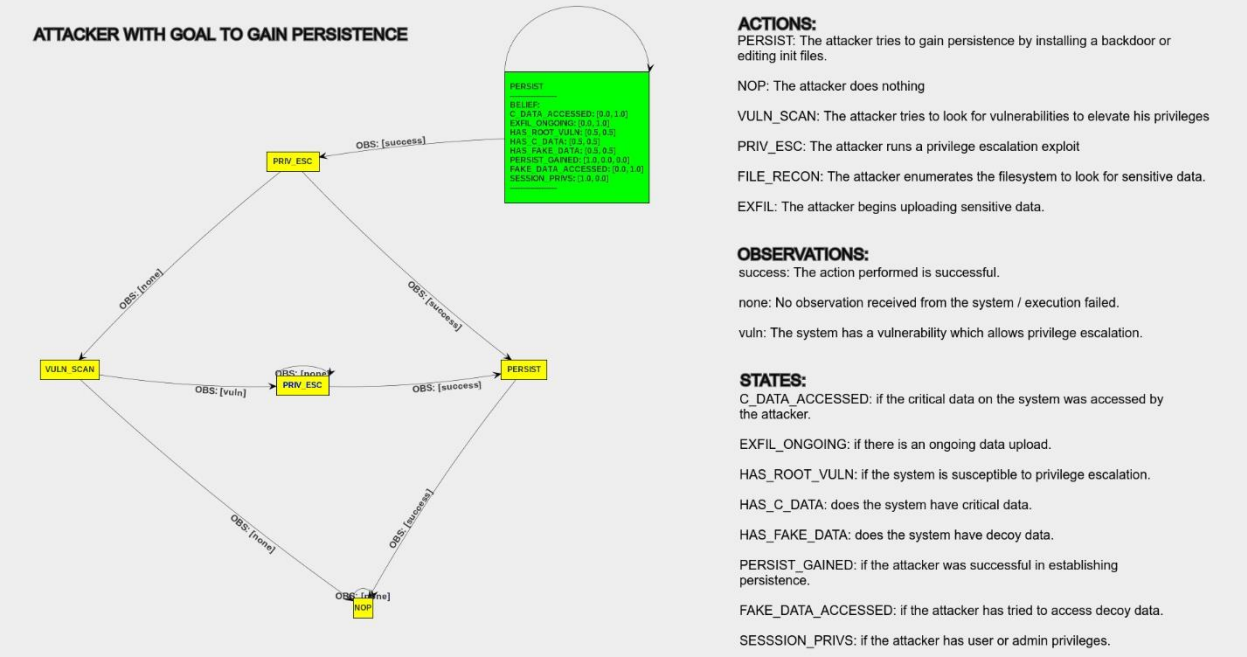
We analyze models' performances by reporting classification F1-scores by attack phase. Table 5 shows that LSTM's and CNN's weakest performance is on asset discovery sequences, where no model achieves a high score. This significantly lower F1-score is due, in part, to sequences of that class having low average lengths compared to others. Asset discovery sequences give the models less information to learn from and inform the classification.

We submitted the paper on this novel AI-based methodology for identifying cyber attacks in honey pots to the refereed IAAI 2021 conference, where it was accepted for publication. The paper was virtually presented at the conference in February 2021.

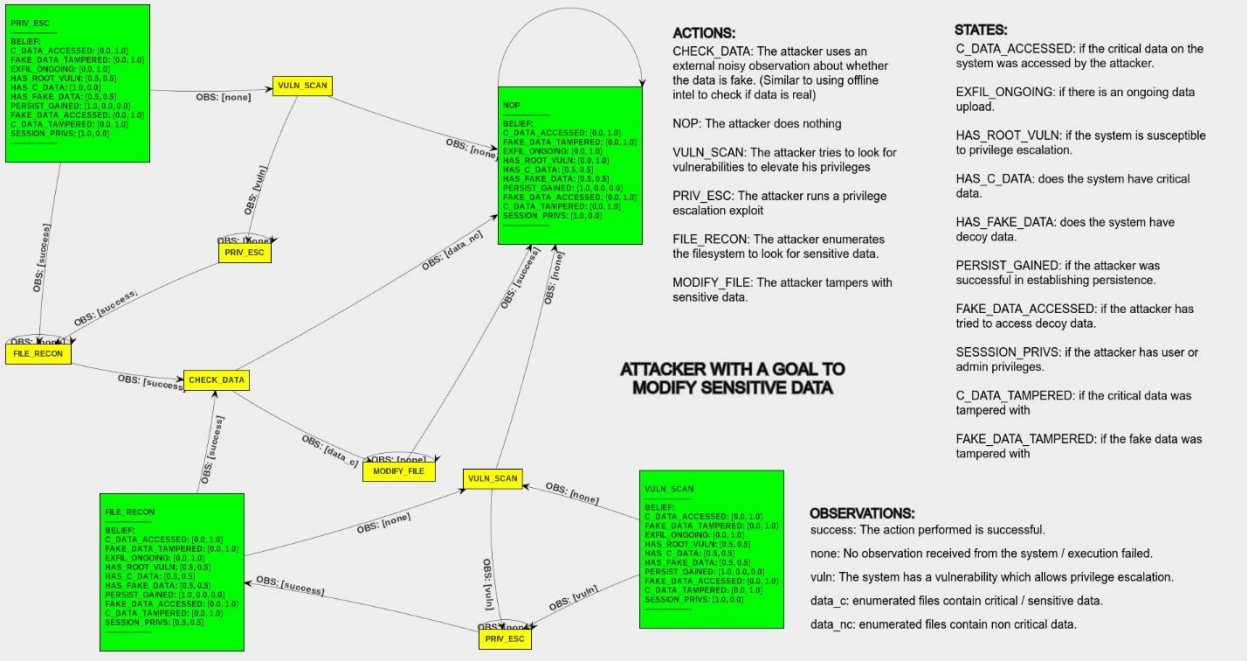
APPENDIX: FIGURES & TABLES



(a) Data exfiltration



(b) Persistence



(c) Data modifier

Figure 1: Finite state automata for the three types of attacks obtained as solutions of POMDPs. Boxes in green are action nodes that additionally show representative attacker beliefs that lead to the prescribed actions.

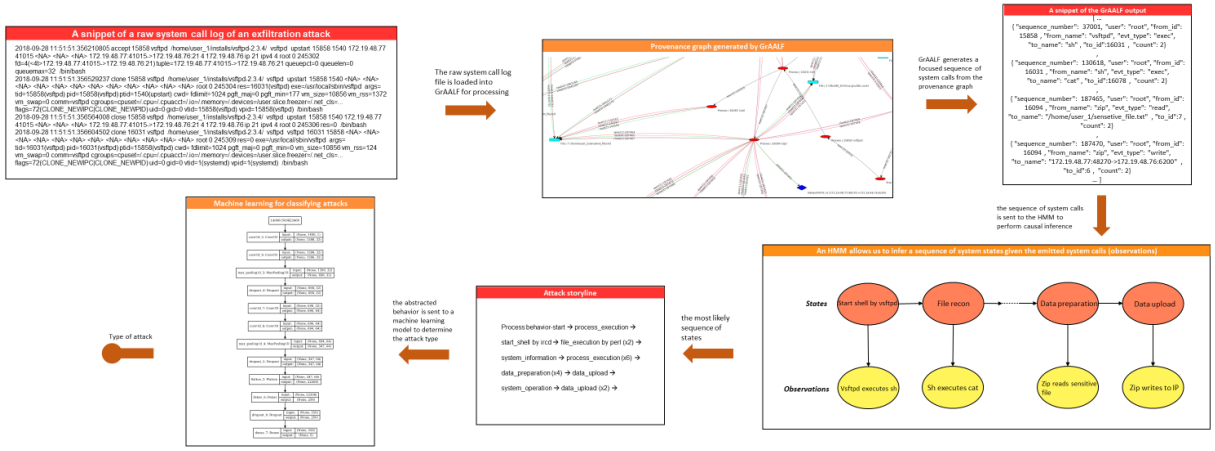


Figure 2: To enable active cyberdeception, system call logs will be frequently loaded by GrAALF, our graphical tool for forensics analysis of logs, which will generate focused traces of possible attacks. A trained hidden Markov model will offer a most likely higher-level explanation for a trace in the form of a sequence of states (we refer to this as the attack storyline). This storyline will be given as input to a trained classifier to determine the type of attack.

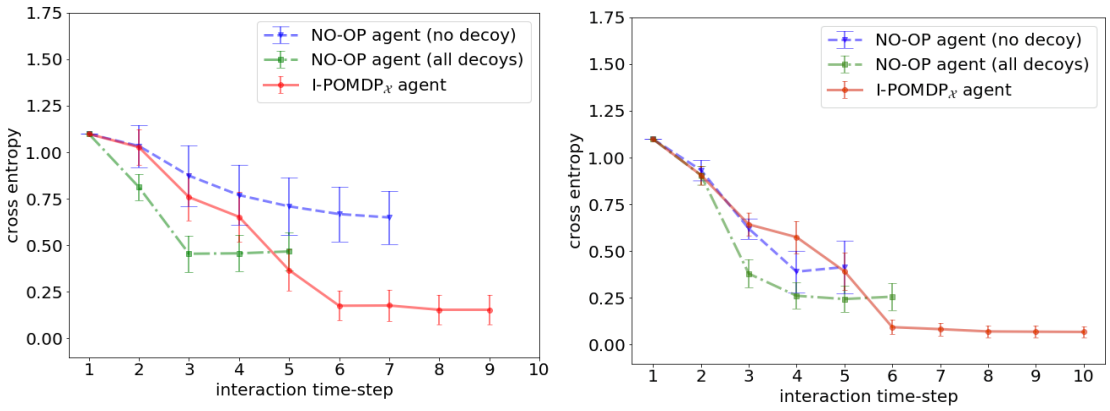
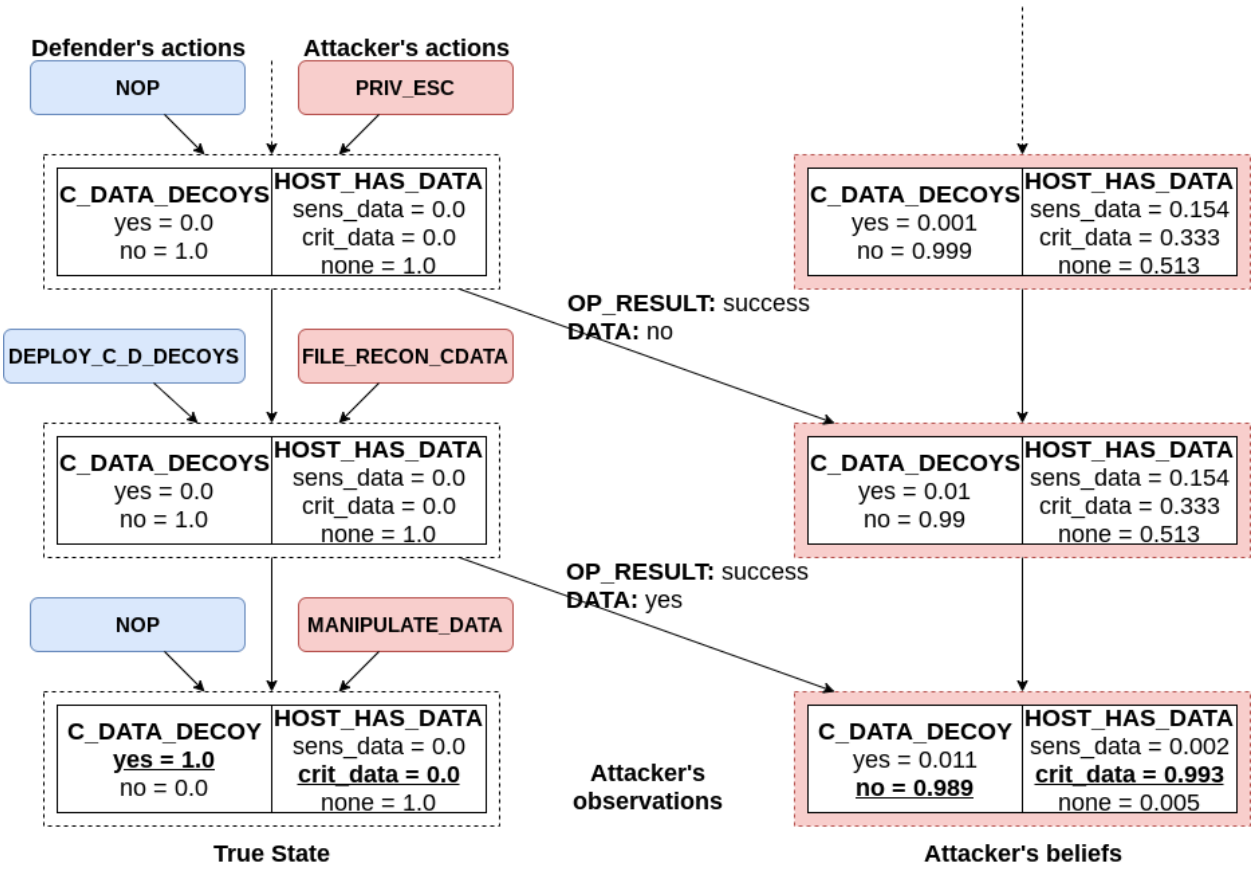
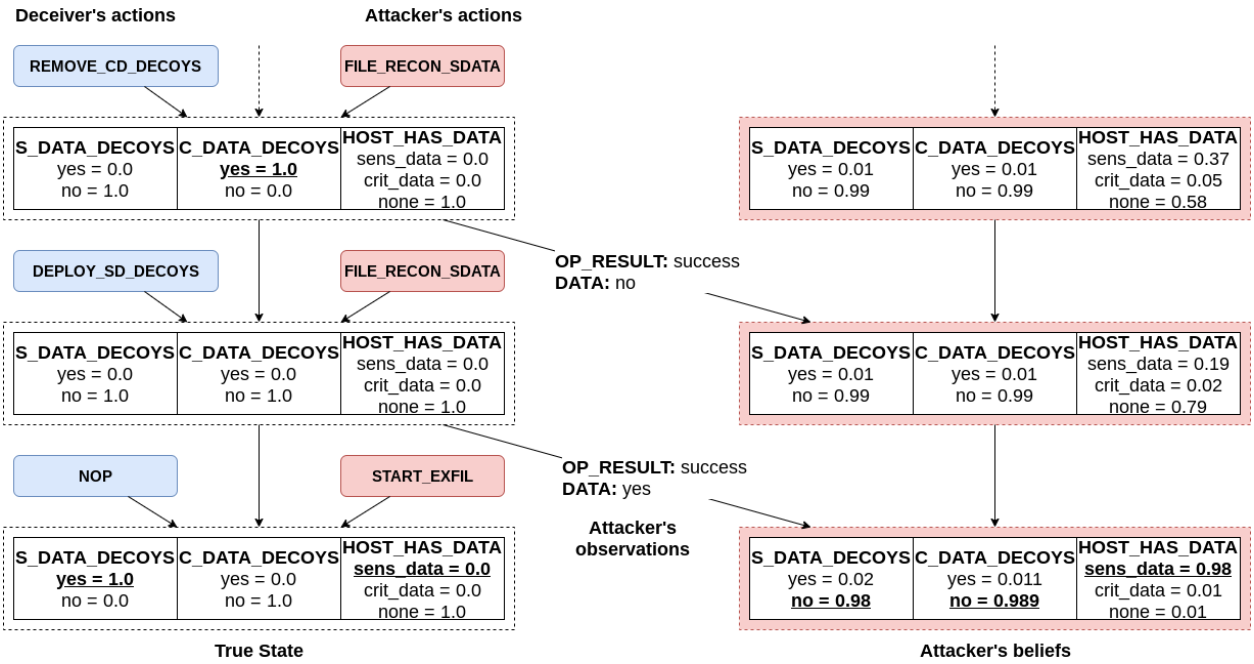


Figure 3: Cross entropy (KL divergence) of the beliefs of the I-POMDP_X agent and other baselines in simulations (left) and when deployed on a host (right). the I-POMDP$_X$ based agent uses implemented deception techniques and audit log analysis for observations, to engage with the attacker for longer duration than other agents and form more informative beliefs. Cross entropies near zero signify good intent recognition.



(a)



(b)

Figure 4: (a) The attacker starts with a low prior belief on the existence of decoys and an active defender. If decoys are indistinguishable from real data, the attacker attributes his observation to the existence of real data even when the host has none. (b) In the event of deploying wrong decoys, the defender corrects the decoy deployment. In this case, on observing file discovery actions, the defender deployed critical data decoys. Later as the interaction progresses, the defender forms a better belief over the attacker's frame from the observation and replaces the decoys before the attacker discovers the discrepancy.

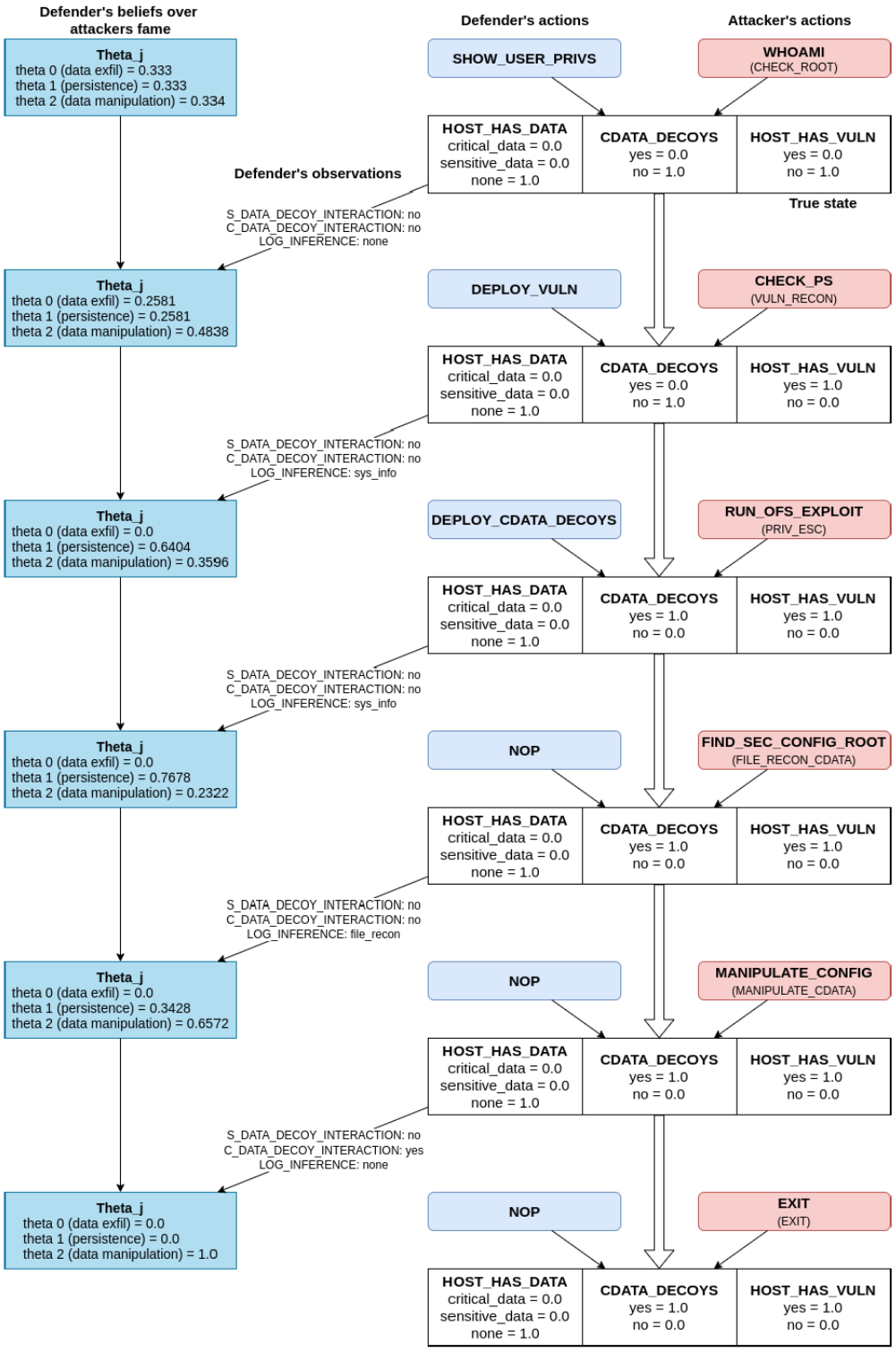


Figure 5: Interaction trace for I-POMDP$_X$-based defender against a human role playing a data manipulator type attacker.

Fold	Mean LL ratio
0	0.945 ± 0.068
1	0.919 ± 0.096
2	0.937 ± 0.076
3	0.924 ± 0.086
4	0.964 ± 0.05

(a)

Model	Asset disc.	Sys. recon.	Exfil	N/w disc.	Persist	Priv. Escal.
HMM	0.959	0.977	0.927	**0.850**	1.0	0.987

(b)

Table 1: (a) The mean and standard deviation of the log likelihood ratio per test fold generated by the HMM. The mean across all folds is 0.938.(b) Mean log likelihood ratio of the HMM decomposed by attack phases. The lowest performance among the attack phases is highlighted.

Model	Weighted-mean F1	
	with HMM	without HMM
SVM	85.36 ± 7.80	64.79 ± 15.75
CNN	$\mathbf{92.09 \pm 7.48}$	14.51 ± 16.30
LSTM	82.90 ± 14.63	47.55 ± 26.03

Table 2: Weighted-mean F1-score and weighted standard deviation across all classes for the models with and without the use of storylines. The weighted statistics are obtained by weighting the F1-scores of the phases with their respective class sizes.

Model	Asset disc.	Sys. recon.	Exfil	N/w disc.	Persist	Priv. Escal.
SVM	**71(46)**	90(80)	90(55)	79(64)	92(80)	86(36)
CNN	**77(93)**	95(98)	100(98)	94(95)	96(100)	87(90)
LSTM	**52(99)**	88(99)	100(99)	85(99)	83(99)	84(99)

Table 3: F1-score of each HMM-aided model decomposed by attack phases, and mean confidence on correct classifications (true positives). Lowest performance for each model is highlighted.

Model	Weighted-mean F1	
	with HMM	without HMM
SVM	84.14 ± 12.82	64.79 ± 15.75
CNN	86.32 ± 11.46	14.51 ± 16.30
LSTM	$\mathbf{90.31 \pm 8.44}$	47.55 ± 26.03

Table 4: Weighted-mean F1-score (%) and weighted standard deviation for models with and without storylines. Statistics obtained by weighting phases' F1-scores with class sizes.

Model	Asset disc.	Sys. recon.	Exfil	N/w disc.	Persist	Priv. Escal.
SVM	74(43)	90(61)	93(50)	**62(58)**	89(74)	97(36)
CNN	**64(90)**	90(99)	100(98)	91(99)	89(100)	79(98)
LSTM	**73(91)**	92(99)	100(99)	94(99)	86(99)	93(99)

Table 5: F1-score (%) of HMM-aided models by attack phases, and mean confidence on correct classifications (true positives). Lowest performance for each model is highlighted.