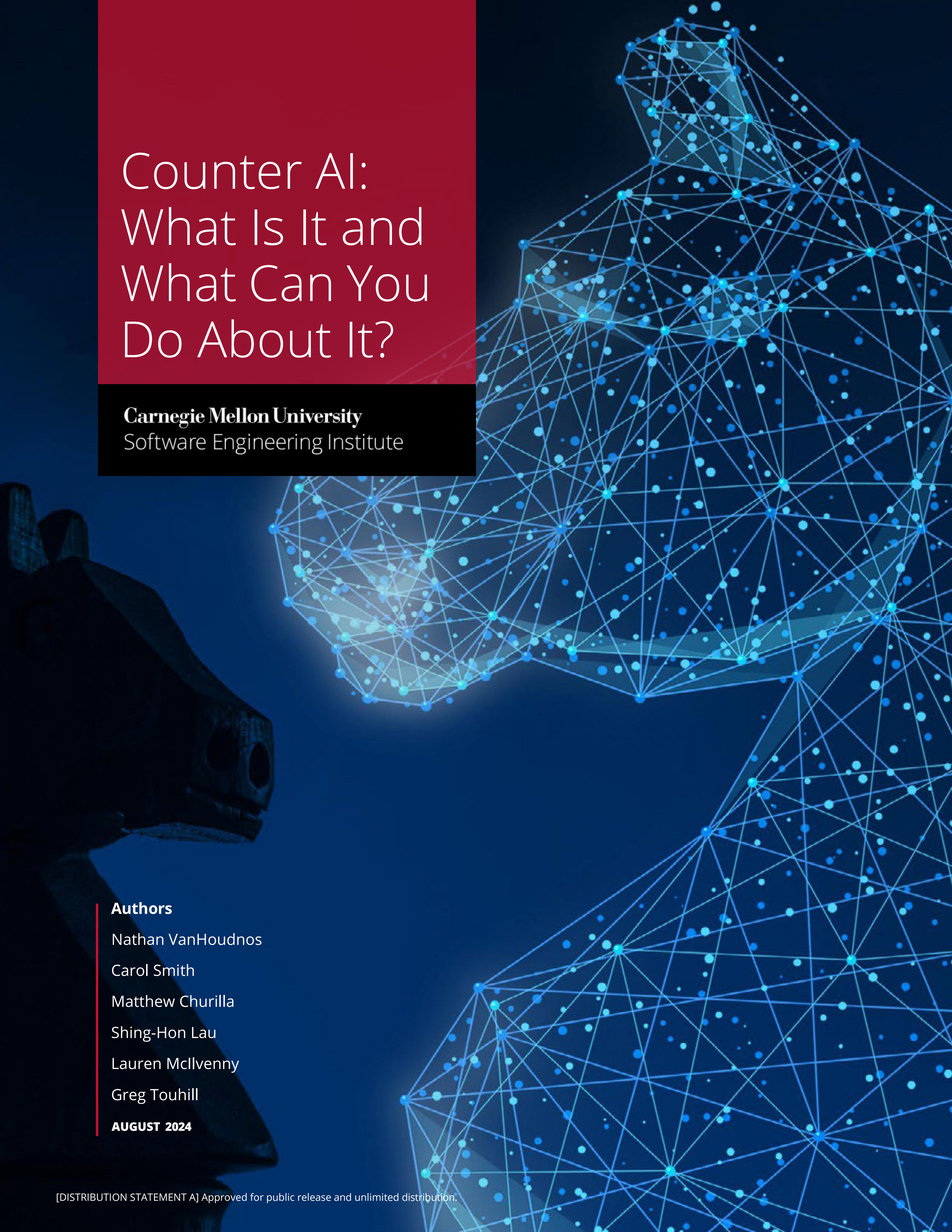


Counter AI: What Is It and What Can You Do About It?

Carnegie Mellon University
Software Engineering Institute

Authors

Nathan VanHoudnos

Carol Smith

Matthew Churilla

Shing-Hon Lau

Lauren McIlvenny

Greg Touhill

AUGUST 2024

AS THE STRATEGIC IMPORTANCE OF AI INCREASES, SO TOO DOES THE IMPORTANCE OF DEFENDING THOSE AI SYSTEMS [NSCAI 2021]. To understand AI defense, it is necessary to understand AI offense—that is, counter AI.

This paper describes counter AI. First, we describe the technologies that compose AI systems (the AI Stack) and how those systems are built in a machine learning operations (MLOps) lifecycle. Second, we describe three kinds of counter-AI attacks across the AI Stack and five threat models detailing when those attacks occur within the MLOps lifecycle.

Finally, based on Carnegie Mellon University (CMU) Software Engineering Institute (SEI) research and practice in counter AI, we give two recommendations:

- In the long-term, the field should invest in AI engineering research that fosters processes, procedures, and mechanisms that prevent vulnerabilities being introduced into AI systems.
- In the near-term, the field should develop the processes necessary to efficiently respond to and mitigate counter-AI attacks, such as building an AI Security Incident Response Team (AISIRT) and extending existing cybersecurity processes like the *Computer Security Incident Response Team (CSIRT) Services Framework* [FIRST 2019].

The AI Stack and MLOps: AI Technologies and AI Processes

The AI Stack is a pictorial representation of the various technologies and areas of research that are necessary to build out a breadth of AI capabilities [Moore 2018]. It is useful for understanding what AI systems are, but it does not depict how AI systems are built.

In this section, we summarize the AI Stack and develop a companion MLOps lifecycle to describe the process by which AI systems are built.

The left side of Figure 1 shows the AI Stack [Moore 2018]. The gray vertical bar, Ethics, spans the horizontal layers of the stack to emphasize the importance of building and using AI systems that align with the core democratic values of the United States. Ethics encompasses efforts known as AI assurance or responsible AI and includes the tools and techniques necessary to build an AI system that is aligned with user needs and ethical principles, meets technical specifications throughout the various stages of the development and acquisition processes, and is continuously monitored during operational or mission use.

The bottom two horizontal layers, Computing and Devices, form the basis of the hardware, software, and perception systems on which AI systems run. Computing in this formulation is broad, encompassing all the systems, networks, programming languages, operating systems, and hardware that enable computation. The Device layer is similarly broad, referencing the sensors and components needed for machines to perceive the world around them, including everything from cameras to light detection and ranging (LIDAR), and from synthetic aperture radar (SAR) to cyber sensors, such as net flow monitors.

The Massive Data Management (red) and Machine Learning (green) layers form the core of the AI system. The Massive Data Management layer includes the selection and analysis of data, preparation of data, and overall data management. Data are ingested and then used to identify statistical patterns in the Machine Learning layer. The Machine Learning layer represents the whole academic field of statistical machine learning, including supervised, unsupervised, and reinforcement learning approaches.

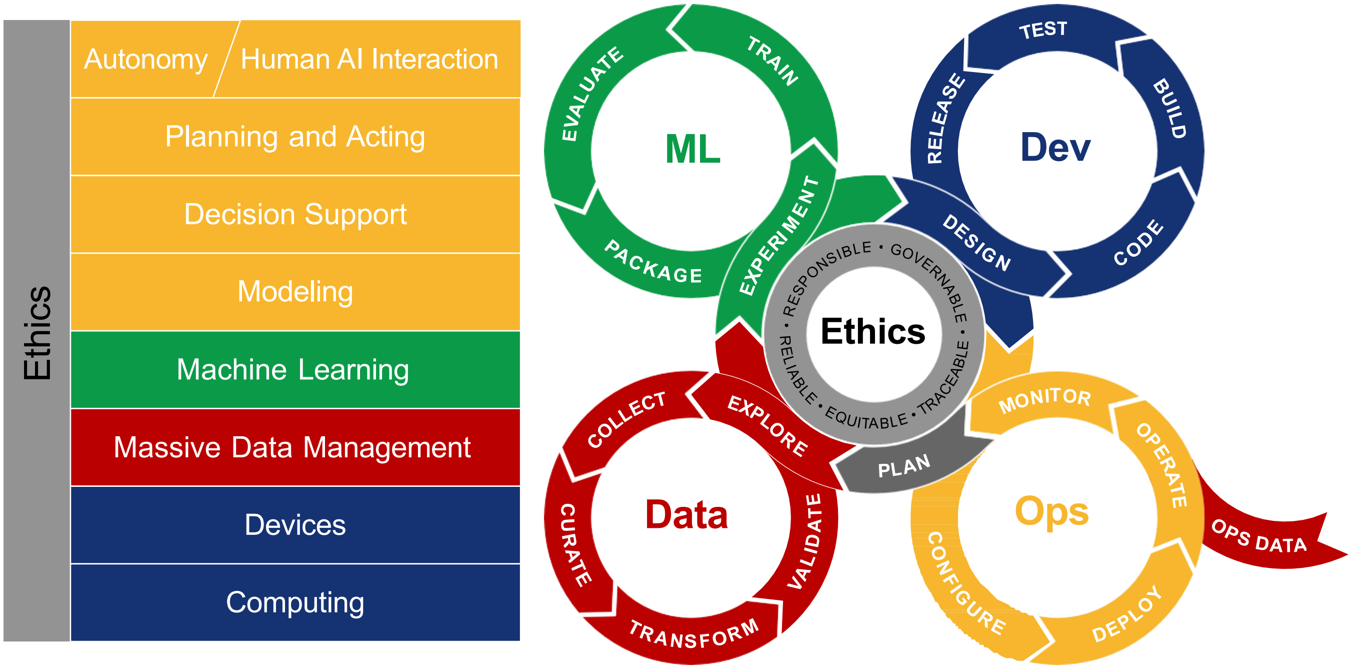


Figure 1: The AI Stack and a Companion MLOps Lifecycle

The top set of layers in the AI Stack represent the broad capabilities that are often ascribed to AI systems.

- The Modeling layer encompasses structuring of knowledge about the world to synthesize data into higher order concepts.
- The Decision Support layer encompasses the various ways that models synthesized from data can be used to help either humans or algorithms make decisions.
- The Planning and Acting layer represents AI systems that team with humans to make and carry out plans.
- The Autonomy and Human AI Interaction layer represents the spectrum of engagement and actions delegated by a human to an AI system.

In some cases, the AI system displays information in an interface for a human to then act; in others, the AI system may be designed for human-machine teaming scenarios; or the AI system may be designed to act without further guidance, in full autonomy.

The right side of Figure 1 is a companion MLOps lifecycle adapted from an earlier MLOps lifecycle [ML4Devs 2022]. The MLOps lifecycle is centered on the five DoD ethical principles for AI [DoD 2020]. The lifecycle is composed of five connected loops: Ethics, Data, ML (machine learning), Dev (development), and Ops (operations). Ethics (grey) takes a prominent place here (as it did in the AI Stack) because, again, considerations of ethics should permeate the whole process. The MLOps lifecycle starts from Ethics, at Plan (grey arrow), where the CONOPS (concept of operations, including end user needs and context) and measures of performance are defined by the team and stakeholders for the eventual system.

The next loop is Data (red), which is entered when the system is first developed and when new training data are required. Its five components are (1) Explore what data already exist, (2) Collect any new data required, (3) Curate and, if necessary, label the data for a given use case, (4) Transform the data for use in machine learning, and (5) Validate that the data meet the requirements of the system and its users. The next loop is ML (green), which is entered when a new ML model is required, or the model is changed. Its four components are (1) Experiment with various approaches, (2) Train the necessary models, (3) Evaluate the quality attributes of the models, and (4) Package the selected models for use within an AI system or submission to a model zoo, that is, a curated collection of models.

When new features are required, the system moves into Dev for development (blue). Its five components are (1) Design the features, (2) Code the features, (3) Build the new system, (4) Test the system for functionality, and (5) Release a new version of the system. The final loop is Ops, for Operations (yellow), with four components (1) Configure the system, (2) Deploy the system, (3) Operate the system, and (4) Monitor the system. The last three components run continuously while the system is in use. Note that Monitor feeds directly into Plan, implying that as information about the system is collected through monitoring, new Data, ML, and Dev loops can be integrated, as needed, to build an improved version of the system. Further note that the Ops loop explicitly includes an Operational Data input because the machine learning component of an AI system requires data.

Counter AI: Three Attacks and Five Threat Models

In this section, we introduce three kinds of counter-AI attacks across the AI Stack and five threat models describing when those attacks occur within the MLOps lifecycle. Briefly, AI systems are fundamentally insecure because there exist vulnerabilities in each layer of the AI Stack and plausible exploits at each step of the MLOps lifecycle.

Across the AI Stack, there are three kinds of counter-AI attacks: cybersecurity attacks, adversarial machine learning attacks, and adversarial AI attacks. Cybersecurity attacks use either physical or cyber methods to target the Computing and Device layers of the AI Stack. Physical cybersecurity attacks can include deny, degrade, and destroy strategies, such as jamming or destroying sensors in autonomous systems in operations. Cyber-based attacks similarly provide a means to deny, degrade, and, in some cases, destroy the software that operates the AI capability.

Adversarial machine learning (AML) attacks use machine learning methods to target the Massive Data Management and Machine Learning layers of the AI Stack. This class of attacks is uniquely effective because Machine Learning is a key component low in the stack that is both exposed during the Ops portion of the MLOps lifecycle and fundamentally vulnerable. In general, an attacker can counter ML by making it learn the wrong thing, do the wrong thing, or reveal the wrong thing about either the MLOps process that created it or the properties of its training data [Beieler 2019].

Generally, countering AI with AML follows a common three-step pattern:

1. *Model the target AI system.* To attack an AI system, the attacker must either have access to the target AI system within its operational context or be able to gather sufficient information about the target to build a proxy (an approximation of the target that is likely to share the same vulnerabilities as the target during its operations).
2. *Train the counter on the model.* In typical ML, the data are held constant as the model learns. In AML this is reversed: the attacker “trains the data” as the model is held constant. Controlling the input in this way allows the attacker to identify how best to poison a model trained on a given set of data (learn attack), drive the target system to a desired state (do attack), or reveal information about the training data or model (reveal attack).
3. *Test the counter.* In typical ML, a trained model is evaluated against several hold-out data sets in several contexts to provide evidence that the model is generalizable. Similarly in AML, a trained counter is evaluated against several hold-out models in several contexts to determine its efficacy as a learn, do, or reveal attack within the operational context of the target AI system.

Adversarial AI attacks use AI to target the Massive Data Management and Machine Learning layers of the AI stack; that is, an Adversarial AI is an AI system that attacks one or more layers in another AI system. For example, if a large language model (LLM) agent were trained to modify malware to appear as benign software, then the LLM agent would be an Adversarial AI that targets the Machine Learning layer of an antivirus. This is distinct from an Adversarial Machine Learning attack, as the Adversarial AI attack is autonomous or semi-autonomous once deployed by the attacker.

Five threat models describing when these three kinds of counter-AI attacks occur within the MLOps lifecycle are shown in Figure 2. The first threat model, ML Only, models when the attacker only has knowledge of the ML loop (green) and is common in early proof-of-concept attack development because it allows a researcher to focus on vulnerability research in a controlled environment. The academic literature primarily uses this threat model to develop learn, do, and reveal attacks. However, the ML Only threat model does not yield exploits, that is, attacks that are likely to work in operations [Appruzzese 2023].

1. ML Only
2. Compromised Operational Environment
3. Acquired System
4. Data Poisoning
5. Future Capability

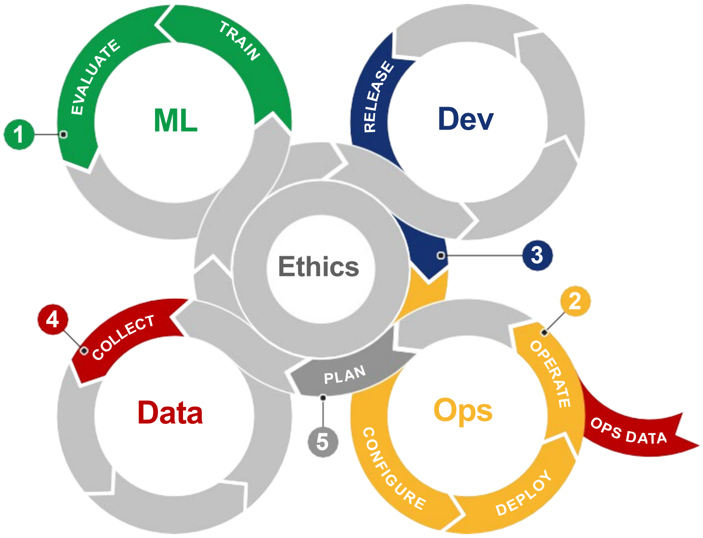


Figure 2: Threat Models Throughout the MLOps Lifecycle

The second threat model in Figure 2 is encountered primarily during Ops (yellow) but also affects Dev (blue). The Compromised Operational Environment is a realistic threat model for do and reveal attacks. Since do and reveal attacks are possible when the attacker can effectively replicate the target system and test various scenarios leading to a comprehensive understanding of vulnerabilities of the system that can be exploited. In this model, the attacker has access to the complete operational environment of the AI system, from the specific software and hardware of the AI system (Release), to the configuration of the sensors that feed in production data (Configure, Deploy, and Operate), to direct in-memory access to the ML model as it makes its predictions (Operate). This scenario is analogous to a traditional cyber-physical vulnerability: the attacker has a copy of the software, can configure it similarly to the target system, and can run it in the same operating system and hardware as the target system. The Compromised Operational Environment threat model is feasible if the attacker has successfully inserted a cyber implant into any machine running a given version of the ML system or if any of the edge devices, such as smart cameras mounted on drones, are captured and reverse engineered to gather not just the model but also the configurations and the prediction pipeline.

The third threat model in Figure 2 is Released System, and it considers the attacker has knowledge of only the Dev loop (blue). In Released System, a copy of the released system (Release) is available, such as when the kind of system the target uses can be purchased from a vendor. In the Released System threat model, it is necessary to build proxies of the target operational environment to develop do attacks that would succeed against other similar versions of the ML model in operational use, that is, to find the various ways that the purchased system could be reasonably configured, deployed, and operated. Once this is accomplished, the attacker can "train the data on the model."

The fourth threat model in Figure 2 is Data Poisoning, contained in the Data loop (red). The attacker has influence over what training data is collected (Collect) but may not be able to control how the data are labeled, have access to the trained model, or have access to the AI system. This scenario is plausible when the AI system collects training data over time and offers predictions as a service to users. The Data Poisoning threat model requires building proxy models, proxy systems, and proxy operational environments. Once this is accomplished, the attacker can train the data on the model.

The fifth threat model in Figure 2 is Future Capability, and it is the most difficult threat model for counter AI with AML. In this threat model, the attacker has no access at all to the target system—only information about the CONOPS of the AI system (Plan). It is surprising that this level of threat model is sufficient to develop a counter, but examples are widely studied in the academic literature, which refer to them as black-box transfer attacks. With sufficient effort, therefore, an attacker could, in principle, walk a black-box transfer attack through proxies across every loop: Data, ML, Dev, and Ops. As the attacker finds counters that fool increasing numbers of proxy models in varying contexts, the likelihood increases that the counter will be successful against the unseen target system.

What to Do About Counter AI: AI Security Incident Response

The existence of a Future Capability threat model underscores the fundamental insecurity of AI systems at our current level of AI maturity. We make two recommendations: In the long term, the field should invest in AI engineering research to enable more capable, accurate, secure, and trustworthy AI systems. In the near term, the field should develop the processes necessary to respond to counter-AI attacks quickly and efficiently. Then, take what we learn from the AI incident response to inform the field of study, identify vulnerabilities, and establish best practices.

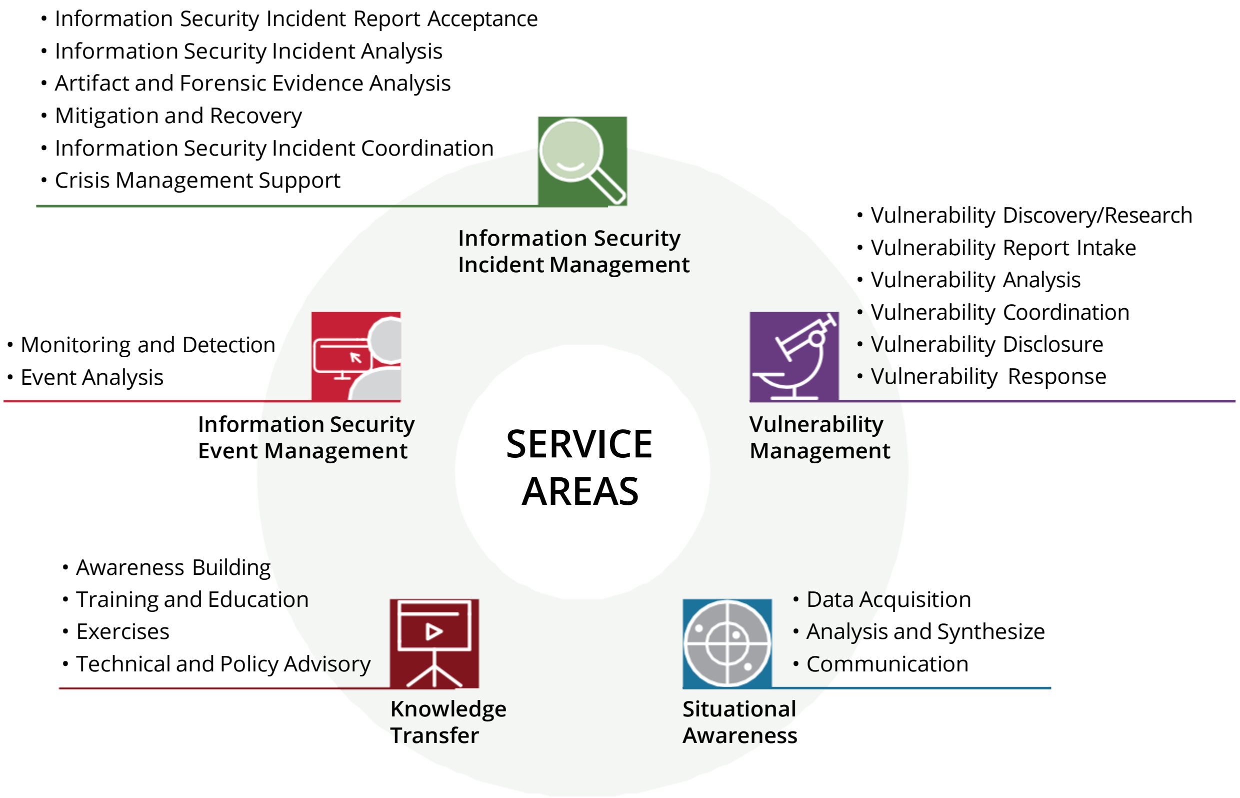

• Information Security Incident Report Acceptance
• Information Security Incident Analysis
• Artifact and Forensic Evidence Analysis
• Mitigation and Recovery
• Information Security Incident Coordination
• Crisis Management Support
Information Security Incident Management
• Vulnerability Discovery/Research
• Vulnerability Report Intake
• Vulnerability Analysis
• Vulnerability Coordination
• Vulnerability Disclosure
• Vulnerability Response
Vulnerability Management
• Monitoring and Detection
• Event Analysis
Information Security Event Management
SERVICE AREAS
• Awareness Building
• Training and Education
• Exercises
• Technical and Policy Advisory
Knowledge Transfer
• Data Acquisition
• Analysis and Synthesize
• Communication
Situational Awareness

Figure 3: CSIRT Services Framework Service Areas and Services [Source: Computer Security Incident Response Team (CSIRT) Services Framework, 2019. Copyright ©2023 Forum of Incident Response and Security Teams, Inc. All Rights Reserved.]

Note that this is very similar to traditional cybersecurity: all computer systems have vulnerabilities. In the long term, the field attempts to make systems more secure. In the near team, the field has learned how to manage vulnerabilities and respond to information security (InfoSec) incidents.

For example, the Computer Security Incident Response Team (CSIRT) Services Framework [FIRST 2019] lays out five service areas to support a robust cybersecurity defense infrastructure. Figure 3 displays the service areas of the CSIRT Services Framework. Note that no single CSIRT will perform all these functions for a given constituency, but this services list provides the breadth of what is required to manage InfoSec incidents.

Briefly, starting from the top of Figure 3, the InfoSec Incident Management service is the core of a CSIRT, where a given InfoSec incident report is received, contained, and mitigated, and operations are restored to a pre-incident state. To the right, the Vulnerability Management service performs vulnerability discovery research in addition to the analysis and handling of new or reported vulnerabilities, including developing vulnerability remediation and coordinating the patching of vulnerabilities. The Situational Awareness service gathers data, understands the context of the threat landscape, models and hunts threats, and communicates both current and projected risks. The Knowledge Transfer service builds awareness of InfoSec incidents and threats, trains and educates, conducts training exercises, and offers technical and policy advice. Finally, the InfoSec Event Management service identifies InfoSec incidents from a wide variety of event logs and contextual data sources.

Broadly, applying the CSIRT Services Framework to AI is straightforward: counter-AI attacks are exploits to vulnerabilities in AI systems. The operators of AI systems should monitor their systems to detect attacks and respond to incidents where attacks have occurred. As counter-AI incidents occur, the community should share knowledge to mitigate future attacks and conduct the necessary research to both discover new vulnerabilities and understand how the threat landscape evolves.

There are, however, important differences between the FIRST CSIRT Services Framework and the development of its AI-enabled extension. For example, within the Incident Response service area, the Mitigation and Recovery service is more complex for AI systems as the as the vulnerability may not live in code but in the model or the data [Spring 2020]. Mitigation may require significant resources, especially for systems that are not designed to be quickly modified. Similarly, within the Vulnerability Management service area, the Vulnerability Analysis service may require significant resources, as it often unclear what the root cause of an AI vulnerability is or how best to mitigate it.

We recommend, therefore, that in the near term, the field should invest in developing this AI-enabled extension of the CSIRT Services Framework. There are salient differences between traditional cybersecurity and AI, and we must work through them in order to respond to AI security incidents quickly and efficiently.

To conclude, as the strategic importance of AI increases, so too does the importance of defending those AI systems [NSCAI 2021]. In the long term, the field should directly address the fundamental vulnerabilities in AI systems by investing in AI engineering research. In the near term, the field should develop the processes necessary to efficiently respond to and mitigate counter-AI attacks, which means building an AI Security Incident Response Team and extending existing cybersecurity processes, such as the *Computer Security Incident Response Team (CSIRT) Services Framework* [FIRST 2019].

AISIRT: Ensuring the Safety of AI Systems

To provide the U.S. with a capability for addressing the risks introduced by the rapid growth and widespread use of AI, in 2021 the Carnegie Mellon University (CMU) Software Engineering Institute (SEI) formed a first-of-its-kind AI Security Incident Response Team (AISIRT). The SEI created the AISIRT to lead the way in formulating tools, practices, and guidelines for AI security incident response. The AISIRT works with government, industry, and academia to identify, analyze, and respond to threats to AI systems.

The SEI leveraged its expertise in cybersecurity and AI, as well as its strong track record in the development of cyber response capabilities and team development across the globe over the last 35 years to establish the AISIRT. The goal of the AISIRT is to lead a community-focused research and development effort to ensure the safe and effective development and use of AI technologies as they continue to evolve and grow.

Some of the challenges for maintaining effective monitoring of AI systems include identifying when AI systems are operating out of tolerance; whether they have been subjected to external tampering or attack; where defects occur that need to be corrected; and how to diagnose and respond to suspected or known problems. In addition, response capabilities require successful community and team building with both national and international organizations.

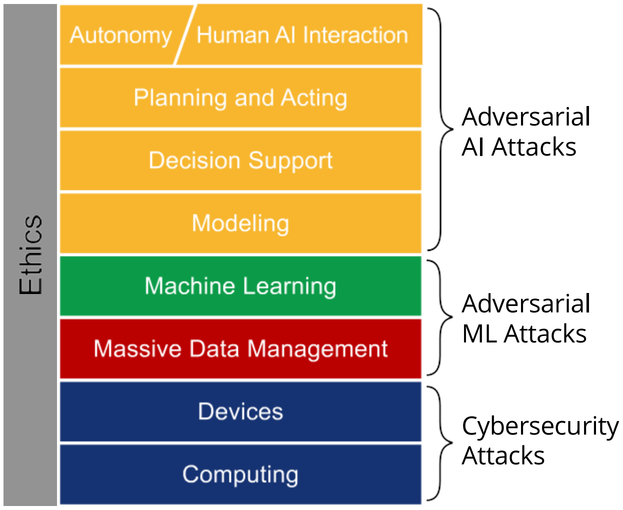
The AISIRT has reported the following items as some emerging lessons learned from managing early AI vulnerabilities.

- AI vulnerabilities are cyber vulnerabilities
- The AI ecosystem is complex encompassing many disparate entities
- AI vulnerabilities occur throughout the entire AI ecosystem
- Tools to identify vulnerabilities are lacking
- There is a need for secure development training tailored for AI developers [McIlvenny 2024]

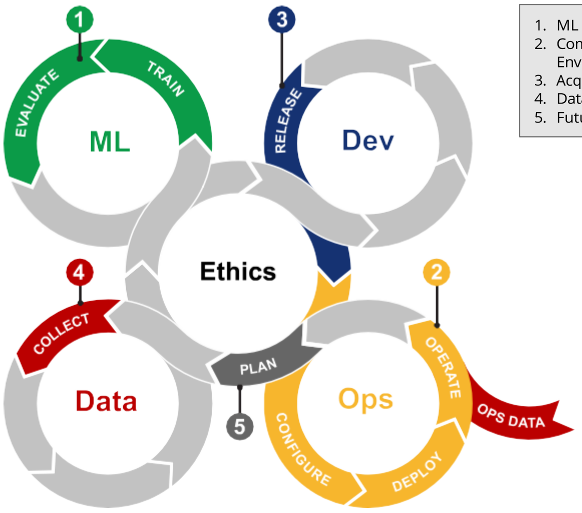
Built from this foundation at the SEI, the AISIRT fills an immediate need to ensure that AI is safe, contributes to the growth of our nation, and continues to evolve in an ethical, equitable, inclusive, and responsible way. The challenges and lessons the AISIRT has identified highlight the strategic importance of defending AI systems. The SEI is committed to continuing the necessary research and building the tools and communities needed to better secure AI systems.

Counter AI Overview

AI STACK



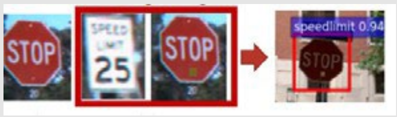
THREAT MODELS



1. ML Only
2. Compromised Operational Environment
3. Acquired System
4. Data Poisoning
5. Future Capability

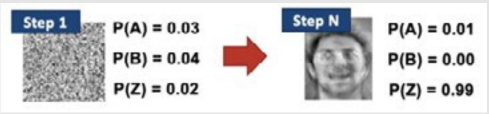
COUNTER AI TAXONOMY

An attacker can make you:
Learn the wrong thing

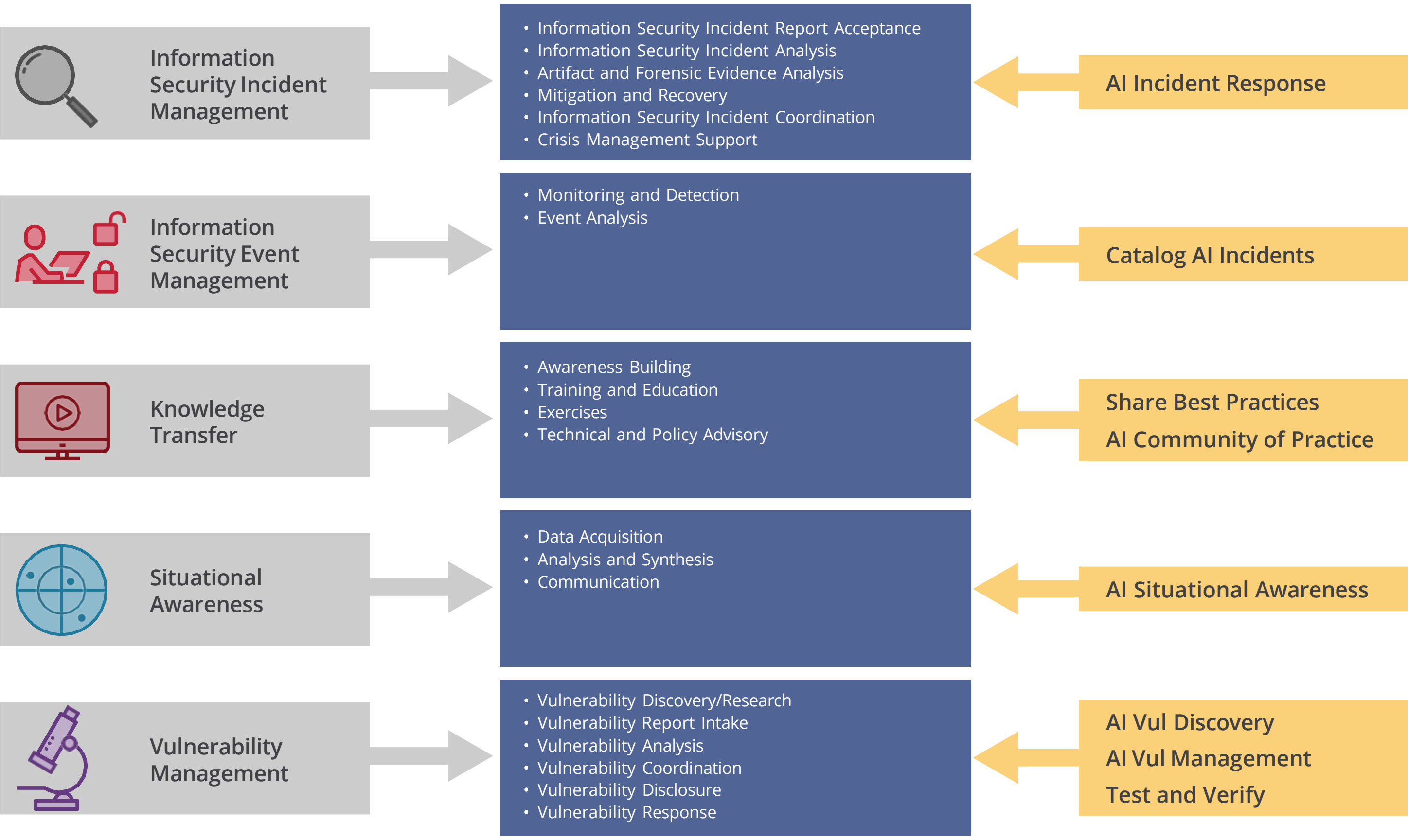


Do the wrong thing

Reveal the wrong thing



Adding AI SMEs to CSIRT Services



SERVICE AREAS	CSIRT PROCESSES	AISIRT EXTENSION
Information Security Incident Management	• Information Security Incident Report Acceptance • Information Security Incident Analysis • Artifact and Forensic Evidence Analysis • Mitigation and Recovery • Information Security Incident Coordination • Crisis Management Support	AI Incident Response
Information Security Event Management	• Monitoring and Detection • Event Analysis	Catalog AI Incidents
Knowledge Transfer	• Awareness Building • Training and Education • Exercises • Technical and Policy Advisory	Share Best Practices AI Community of Practice
Situational Awareness	• Data Acquisition • Analysis and Synthesis • Communication	AI Situational Awareness
Vulnerability Management	• Vulnerability Discovery/Research • Vulnerability Report Intake • Vulnerability Analysis • Vulnerability Coordination • Vulnerability Disclosure • Vulnerability Response	AI Vul Discovery AI Vul Management Test and Verify

References

Appruzzese, G.; Anderson, H. S.; Dambra, S.; Freeman, D.; Pierazzi, F.; & Roundy, K. A. Real Attackers Don't Compute Gradients: Bridging the Gap Between Adversarial ML Research and Practice. Pages 339–364. *Proceedings of the 2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML).* IEEE. 2023. **https://ieeexplore.ieee.org/document/10136152**

Beieler, J. AI Assurance and AI Security: Definitions and Future Directions. Presented at the Adversarial Machine Learning Technical Exchange, Rockville, MD. 2019. **https://cra.org/ccc/wp-content/uploads/sites/2/2020/02/John-Beieler_AISec_AAAS.pdf**

FIRST. Computer Security Incident Response Team (CSIRT) Services Framework. Version 2.1.0. Forum of Incident Response and Security Teams (FIRST) Standards. 2019. **https://www.first.org/standards/frameworks/csirts/csirt_services_framework_v2.1**

Martelaro, N.; Smith, C. J.; & Zilovic, T. Exploring Opportunities in Usable Hazard Analysis Processes for AI Engineering. In *AAAI Spring Symposium Series Workshop on AI Engineering: Creating Scalable, Human-Centered and Robust AI Systems.* 2022. DOI: 10.48550/arXiv.2203.15628. **https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=883992**

McIlvenny L., Touhill G. Creating an AI Security and Incident Response Team. RSA Conference. 2024. **https://www.rsaconference.com/usa/agenda/session/Creating%20an%20AI%20Security%20and%20Incident%20Response%20Team**

ML4Devs. MLOps: Machine Learning Life Cycle. 2022. **https://www.ml4devs.com/articles/mlops-machine-learning-life-cycle/**

Moore, A. W.; Hebert, M.; & Shaneman, S. The AI Stack: A Blueprint for Developing and Deploying Artificial Intelligence. In *Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR IX.* Vol. 10635, pp. 45–54. SPIE. May 2018.

National Security Commission on Artificial Intelligence (NSCAI). *Final Report.* 2021. **https://www.nscai.gov/2021-final-report/**

Spring, J. M.; Galyardt, A.; Householder, A. D.; & VanHoudnos, N. On Managing Vulnerabilities in AI/ML Systems. Pages 111–126. *Proceedings of the New Security Paradigms Workshop 2020.* ACM. January 2021. **https://dl.acm.org/doi/10.1145/3442167.3442177**

U.S. Department of Defense (DoD). DoD Adopts Ethical Principles for Artificial Intelligence. 2020. **https://www.defense.gov/News/Releases/Release/Article/2091996/dodadoptsethical-principles-for-artificial-intelligence/**

About the SEI

Always focused on the future, the Software Engineering Institute (SEI) advances software as a strategic advantage for national security. We lead research and direct transition of software engineering, cybersecurity, and artificial intelligence technologies at the intersection of academia, industry, and government. We serve the nation as a federally funded research and development center (FFRDC) sponsored by the U.S. Department of Defense (DoD) and are based at Carnegie Mellon University, a global research university annually rated among the best for its programs in computer science and engineering.

Contact Us

CARNEGIE MELLON UNIVERSITY
SOFTWARE ENGINEERING INSTITUTE
4500 FIFTH AVENUE; PITTSBURGH, PA 15213-2612

sei.cmu.edu
412.268.5800 | 888.201.4479
info@sei.cmu.edu

This material is based upon work funded and supported by the , Department of Commerce under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center sponsored by the United States Department of Defense.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

References herein to any specific entity, product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Carnegie Mellon University or its Software Engineering Institute nor of Carnegie Mellon University - Software Engineering Institute by any such named or represented entity.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

GOVERNMENT PURPOSE RIGHTS–Technical Data

Contract No.: FA8702-15-D-0002

Contractor Name: Carnegie Mellon University

Contractor Address: 4500 Fifth Avenue, Pittsburgh, PA 15213

The Government's rights to use, modify, reproduce, release, perform, display, or disclose these technical data are restricted by paragraph (b)(2) of the Rights in Technical Data—Noncommercial Items clause contained in the above identified contract. Any reproduction of technical data or portions thereof marked with this legend must also reproduce the markings.

DM24-0794