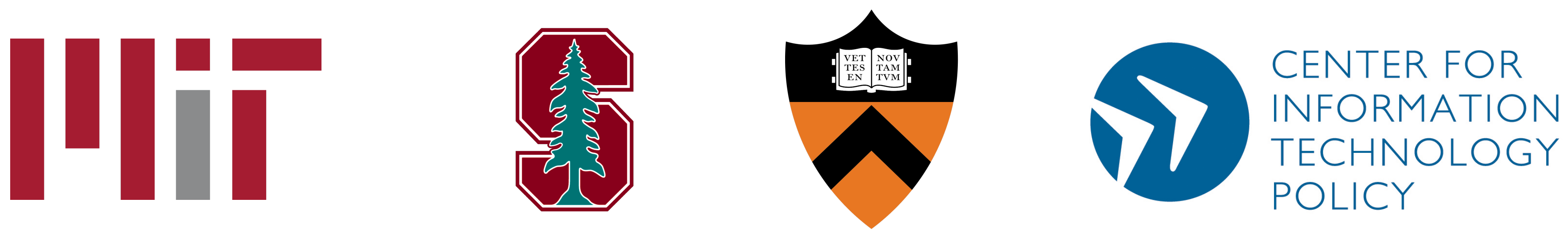


In-House Evaluation Is Not Enough: Towards Robust Third-Party Flaw Disclosure for General-Purpose AI



Shayne Longpre*, Kevin Klyman*, Ruth E Appel*, Sayash Kapoor, Rishi Bommasani, Michelle Sahar, Sean McGregor, Avijit Ghosh, Borhane Blili-Hamelin, Nathan Butters, Alondra Nelson, Amit Elazari, Andrew Sellars, Casey John Ellis, Dane Sherrets, Dawn Song, Harley Geiger, Ilona Cohen, Lauren McIlvenny, Madhulika Srikumar, Mark M Jaycox, Markus Anderljung, Nadine Farid Johnson, Nicholas Carlini, Nicolas Mialhe, Nik Marda, Peter Henderson, Rebecca S Portnoff, Rebecca Weiss, Victoria Westerhoff, Yacine Jernite, Rumman Chowdhury, Percy Liang, Arvind Narayanan

Problem Statement

AI systems, agents, and their applications have many risks.
However, there are obstacles to mitigation:

- 1. An absence of flaw reporting culture
- 2. Limited disclosure infrastructure (eg bug bounties)
- 3. No legal protections for third-party evaluators


Recommendations

We recommend the AI community adopt 3 conventions from the software security community:

- 1. Evaluators should **submit flaw reports**
- 2. AI developers should **adopt flaw disclosure programs**, to **coordinate *universally transferable* flaws**
- 3. AI developers should protect evaluators with **safe harbors**

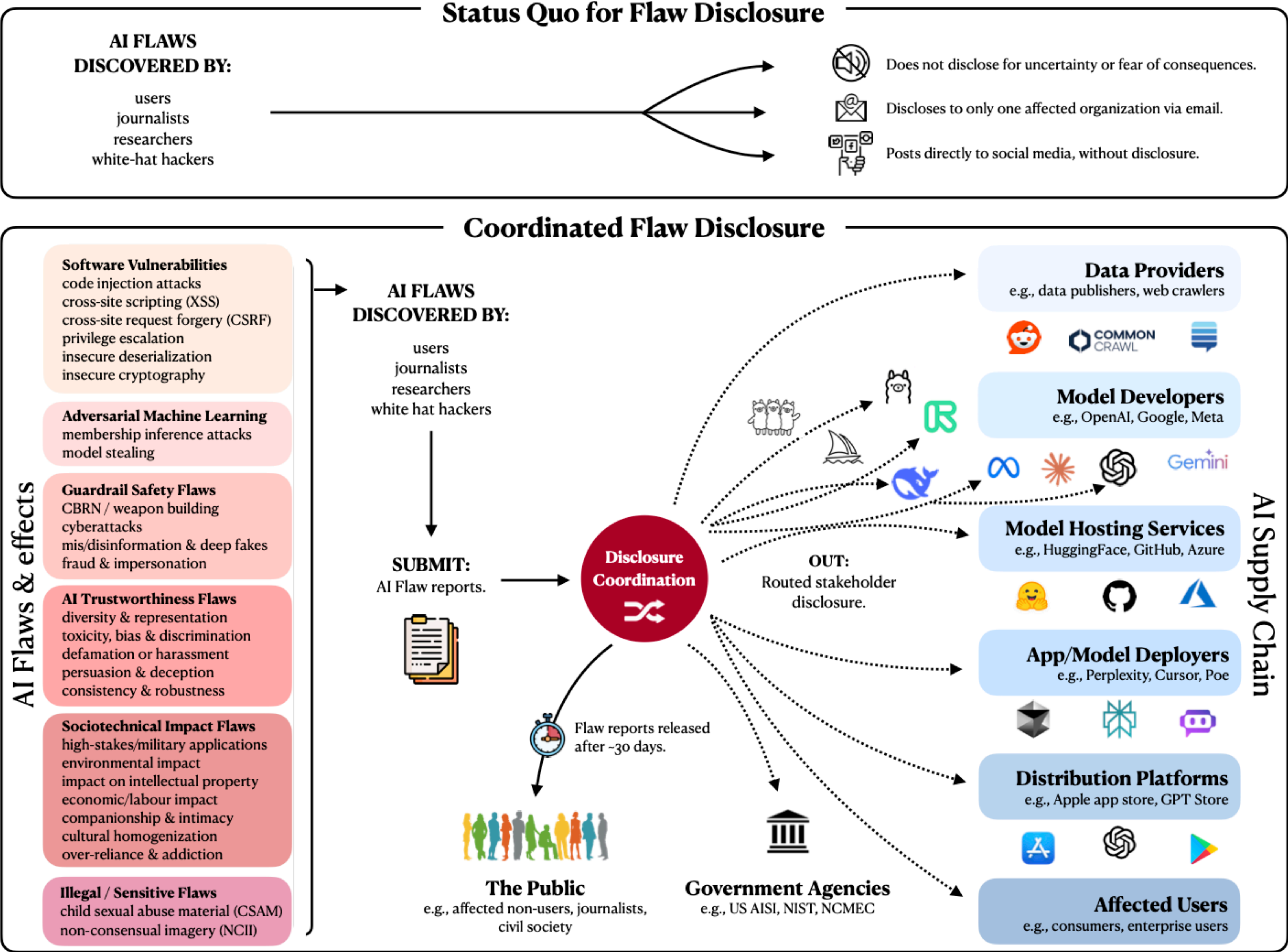
Next Steps

- We are building out a flaw report form, that is:
- A. Fast, and convenient to fill-out
 - B. Collects information that makes it easy for developers to validate, triage, and reproduce reported flaws

We would love to get feedback on it!  **Try out the AI Flaw Report**



Paper Link



Schema for a Flaw Report

Report Type	Field Name	Field Description
Collected for All Flaw Reports	Reporter ID	Anonymous or real identity of flaw reporter
	Report ID	Unique flaw report ID. The flaw report ID can be referenced in future submissions or mitigation efforts, similar to vulnerability identifiers such as CVE identifiers in computer security (Cybersecurity and Infrastructure Security Agency, 2022).
	System Version(s)	AI system(s) and version(s) involved; multiple systems can be selected
	Report Status	Current status of the report, recorded with timestamps as updated by the submitter or receiving company. Initially, the status of a report is “Submitted”, but once it is submitted the status field will be updated to reflect current status of addressing the flaw (e.g., “Under investigation” or “Fixed”) (Cybersecurity and Infrastructure Security Agency, 2022).
	Session ID	System session ID(s) for tracing flaw environment
	Report Timestamp	Report submission timestamp
	Flaw Timestamp(s)	Time(s) where flaws occurred
	Context Info	Versions of other software or hardware systems involved
	Flaw Description	Description of the flaw, its identification, reproduction, and how it violates system policies or user expectations
	Policy Violation	Detail of how the expectations of the system are violated or undocumented, pointing to the terms of use, acceptable use policy, system card, or other documentation. Policies may be explicitly or implicitly violated.
Collected for Real-World Events	Developer	Triage tag with name of system developer
	System	Triage tag with name and version of system
	Severity	Triage tag with worst-case scenario estimate of how negatively stakeholders will be impacted
	Prevalence	Triage tag with rough estimate of how often the flaw might be expressed across system deployments
	Impacts	Triage tag indicating how impacted stakeholders may suffer if the flaw is not addressed
	Impacted Stakeholder(s)	Triage tag(s) indicating who may be harmed if the flaw is not addressed
	Risk Source	Triage tag indicating worst-case scenario estimate of how negatively stakeholders will be impacted
	Bounty Eligibility	Triage tag indicating whether the submitter believes the flaw report meets the criteria for bounty programs
	Description of the Incident(s)	Details on specific real-world event(s) that have occurred
	Implicated Systems	Systems involved in real-world event(s) which generalized flaw reports might cover
Malign Actor	Submitter Relationship	How the submitter is related to the event (e.g., “affected stakeholder” or “independent observer”)
	Event Date(s)	Date when the incident(s) occurred
Security Incident Report	Event Location(s)	Geographical location of the incident(s)
	Experienced Harm Types	Physical; psychological; reputational; economic/property; environmental; public interest/critical infrastructure; fundamental rights; other
Vulnerability Report	Experienced Harm Severity	Maximum severity of harm experienced in the real world
	Harm Narrative	Justification of why the event constitutes harm and how system flaws contributed to it
Hazard Report	Tactic Select	Tactics observed or used (e.g., from MITRE’s ATLAS Matrix)
	Impact	Confidentiality/privacy, integrity, availability, abuse
Vulnerability Report	Threat Actor Intent	Deliberate, unintentional, unknown
	Detection	How the reporter knows about the security incident, including observation methods
Hazard Report	Proof-of-Concept Exploit	A code and documentation archive proving the existence of a vulnerability
	Examples	A list of system inputs/outputs to help understand the replication packet
Hazard Report	Replication Packet	Files evidencing the flaw statistically, including test data, custom evaluators, and structured datasets
	Statistical Argument	Argument supporting sufficient evidence of a flaw

AI flaw reports are complex to design. The relevant information is contingent on many conditions, such as whether the flaw has caused harm (and become an “incident”), or whether there is a malicious threat actor.