

Autonomous and Secure AI Agents through Information-flow Control

Boris Köpf

Azure Research, Microsoft

19th ACM Workshop on AI and Security

15/11/2026

Amazing collaborators



Aashish Kolluri



Shruti Tople



Santiago Zanella-Béguelin



Juhee Kim



Andrew Paverd



Ahmed Salem



Rishi Sharma



Amaury Chamayou



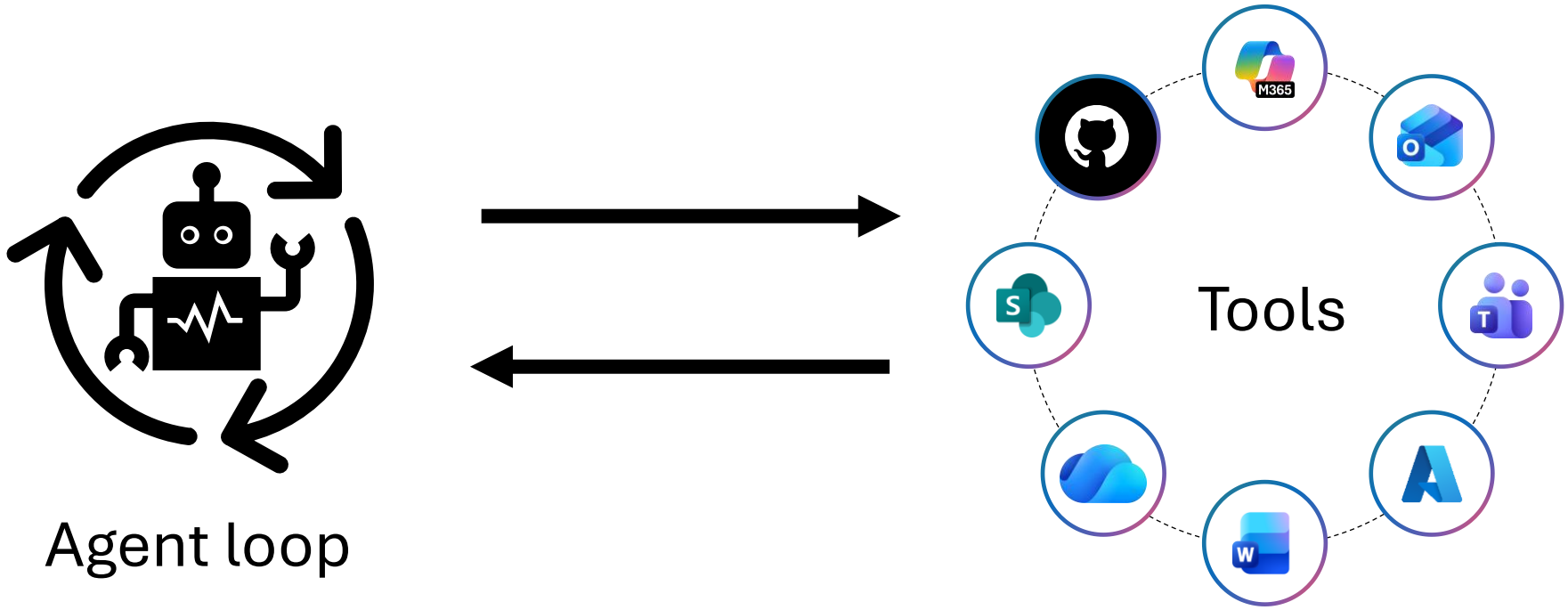
Manuel Costa



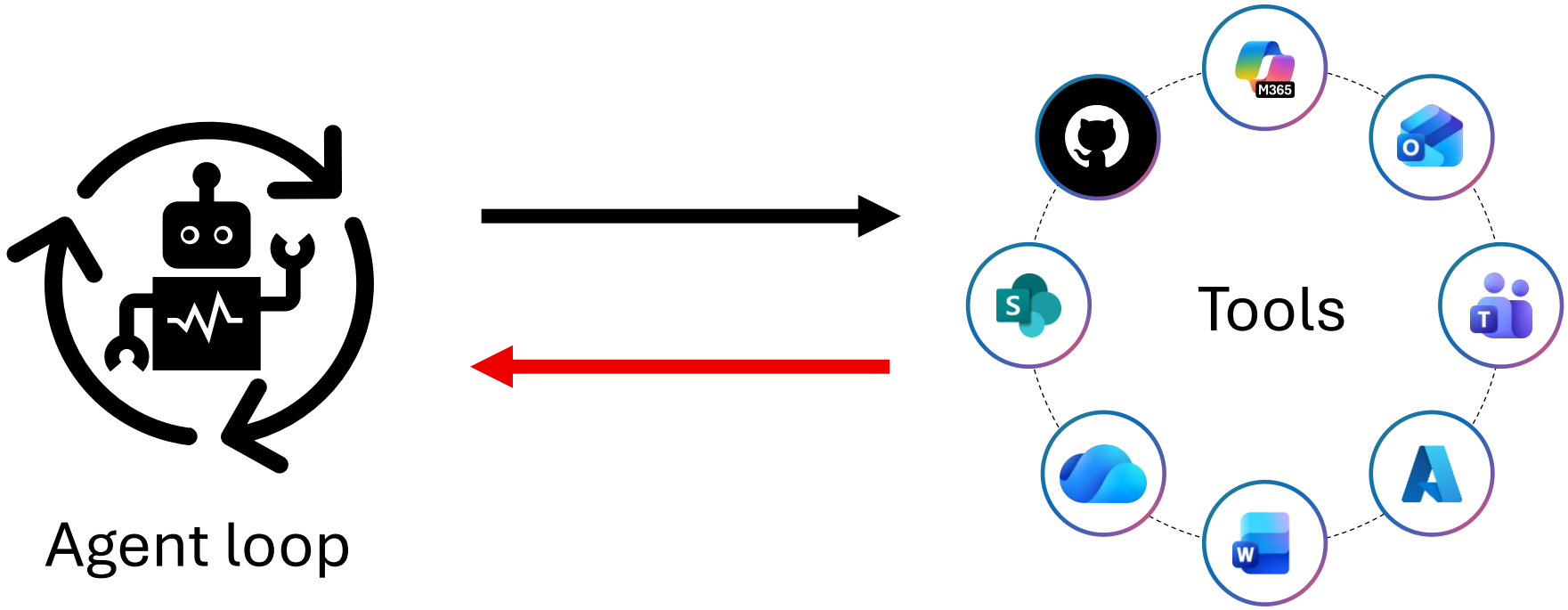
Mark Russinovich

https://www.microsoft.com/en-us/research/project/roma/

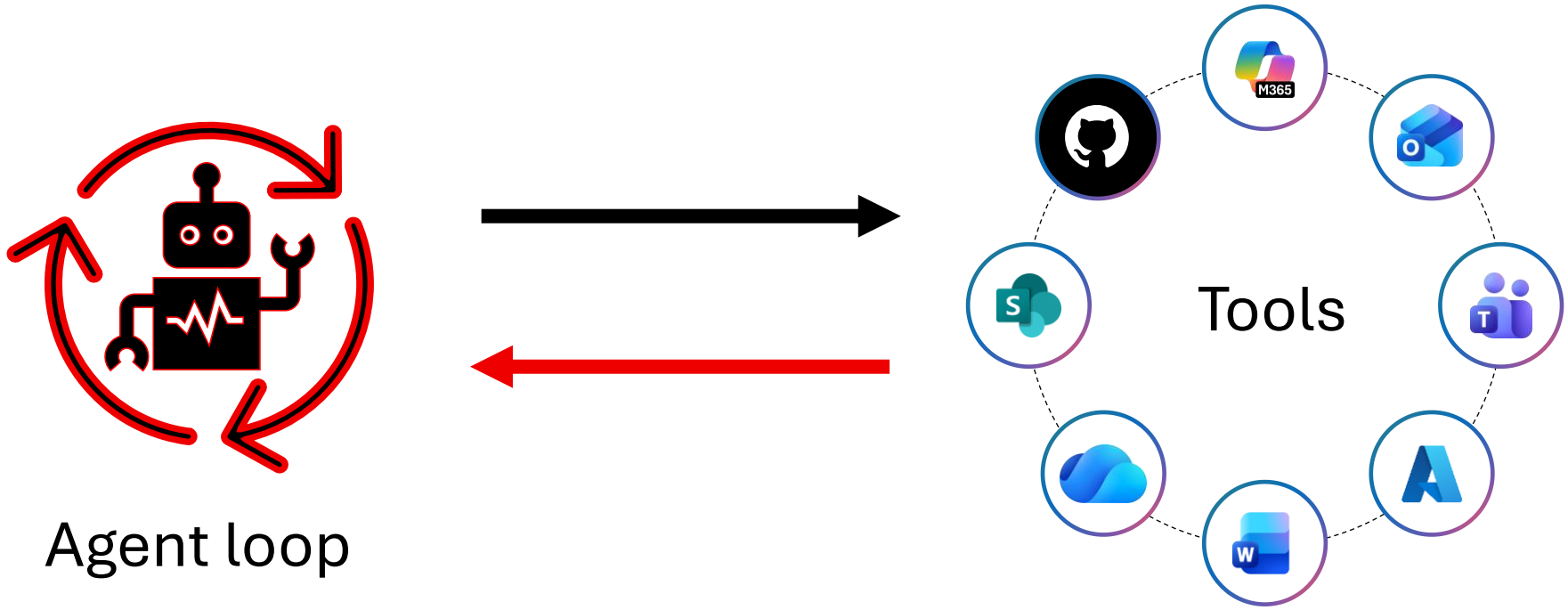
AI Agents: Increasing capabilities, increasing risk



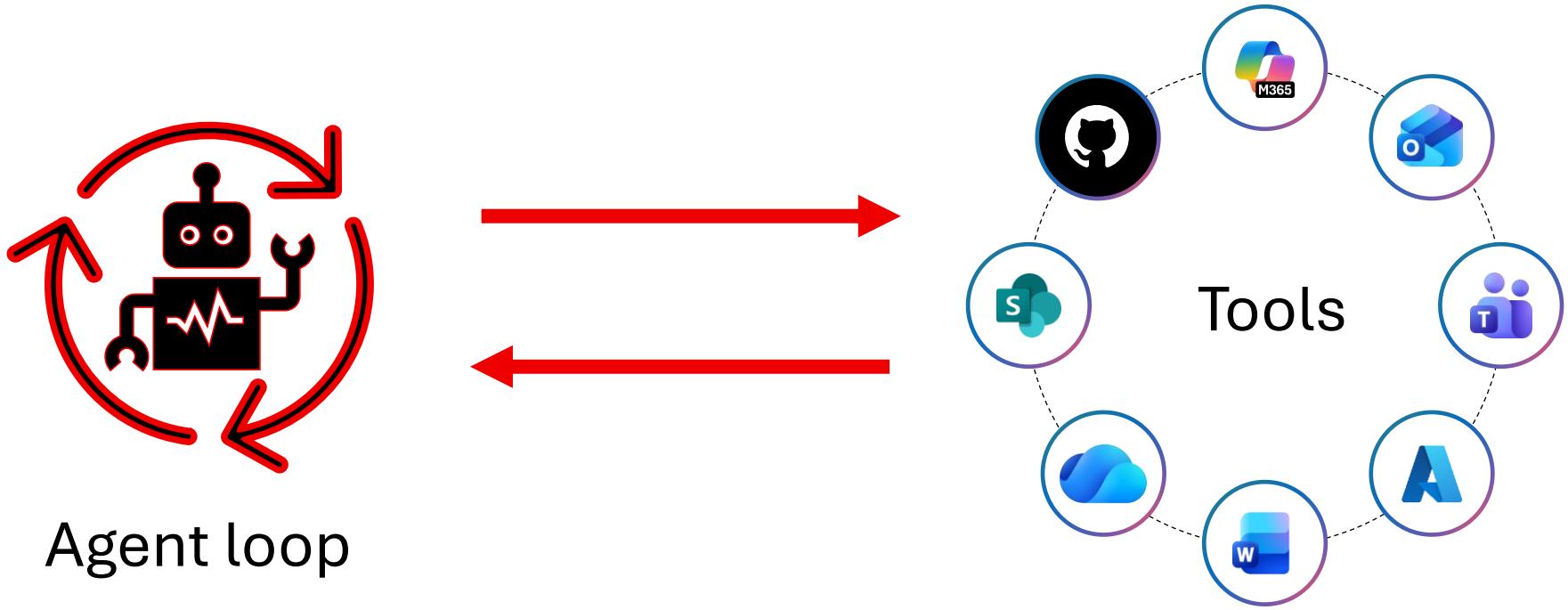
AI Agents: Increasing capabilities, increasing risk



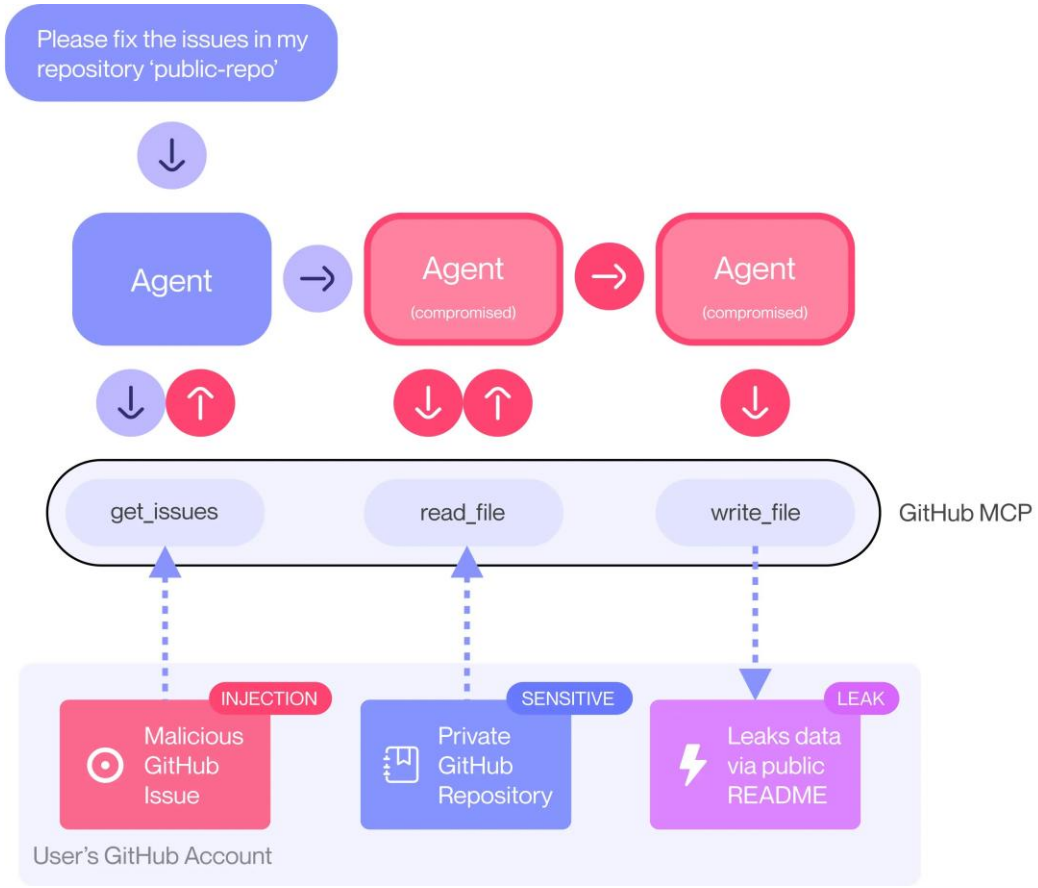
AI Agents: Increasing capabilities, increasing risk



AI Agents: Increasing capabilities, increasing risk



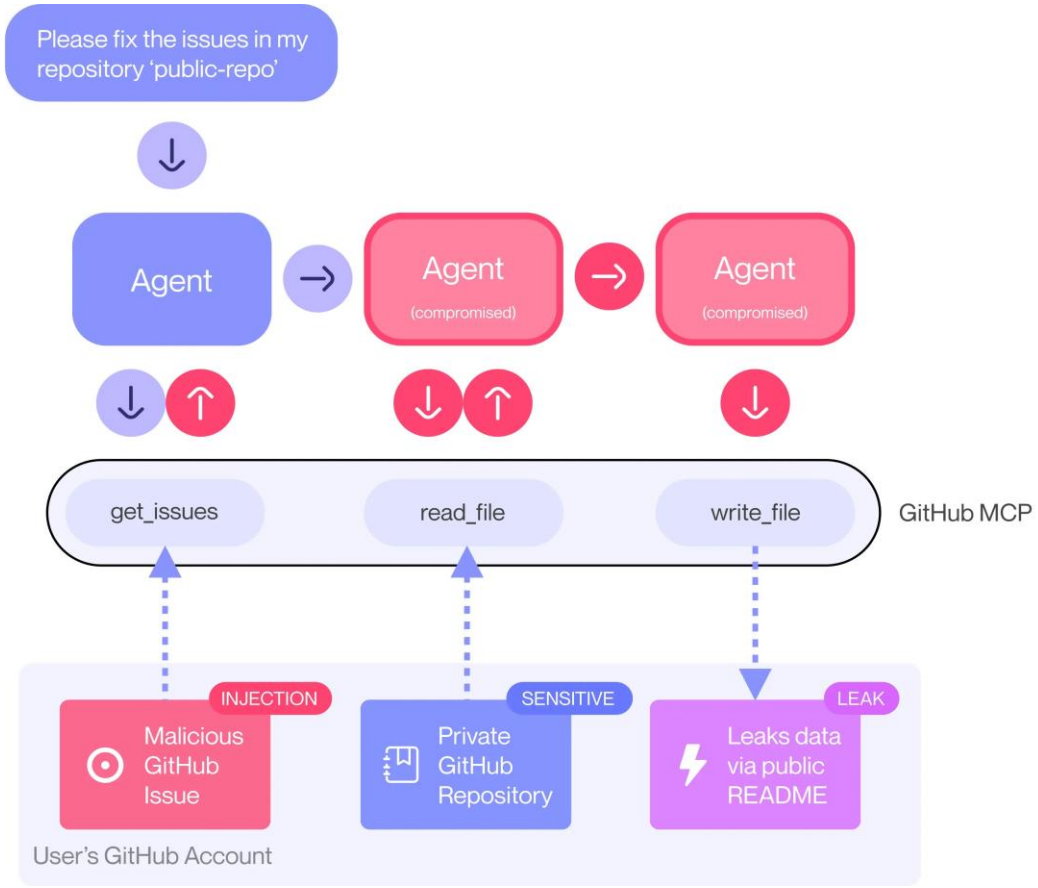
Prompt injection attacks in the wild

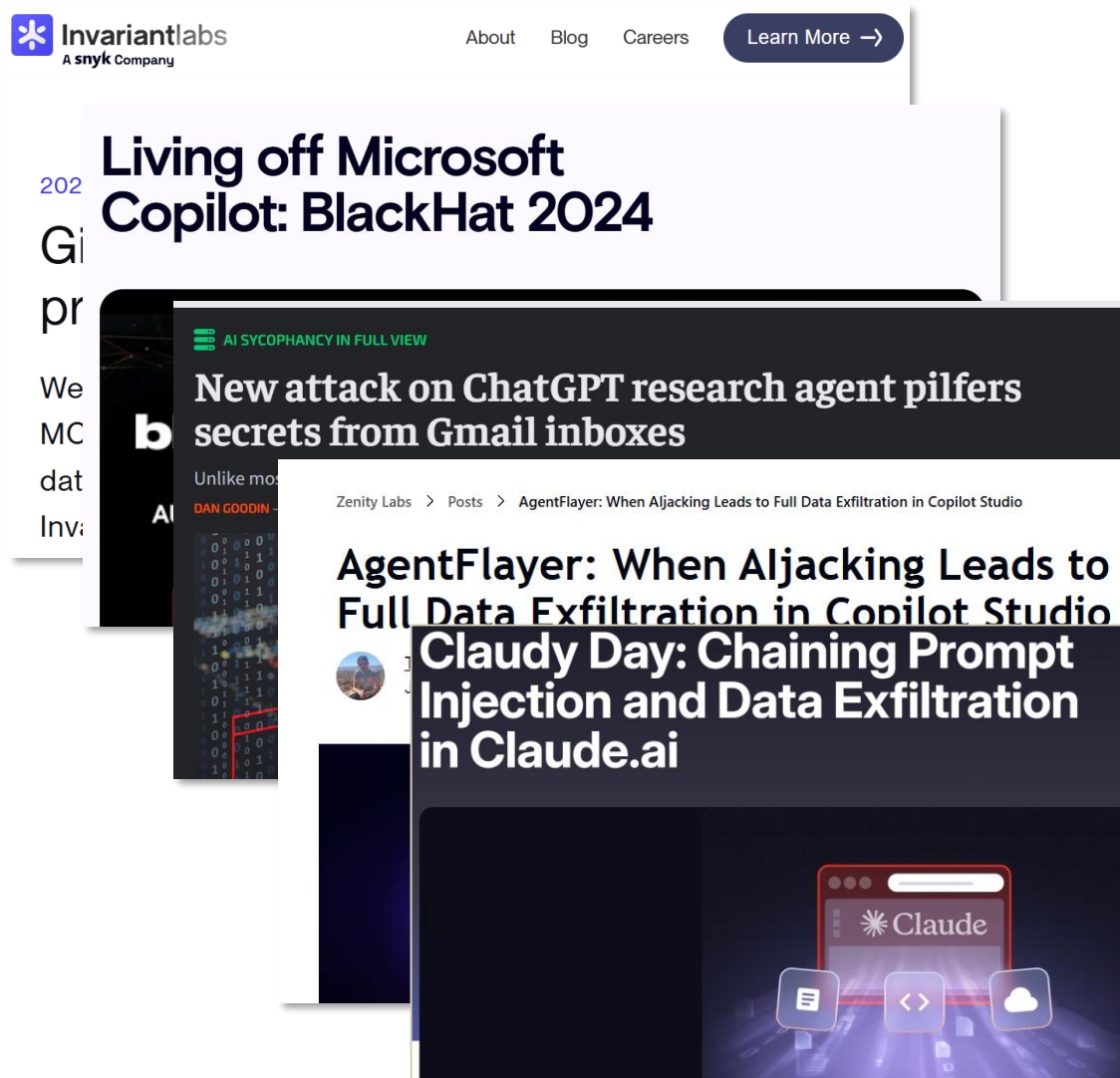




GitHub

Prompt injection attacks in the wild





Current PIA defenses

Probabilistic defenses

Instruction hierarchy, task-adherence, filters, ...

NO GUARANTEE

Human approves everything

Safe baseline: every action needs a click

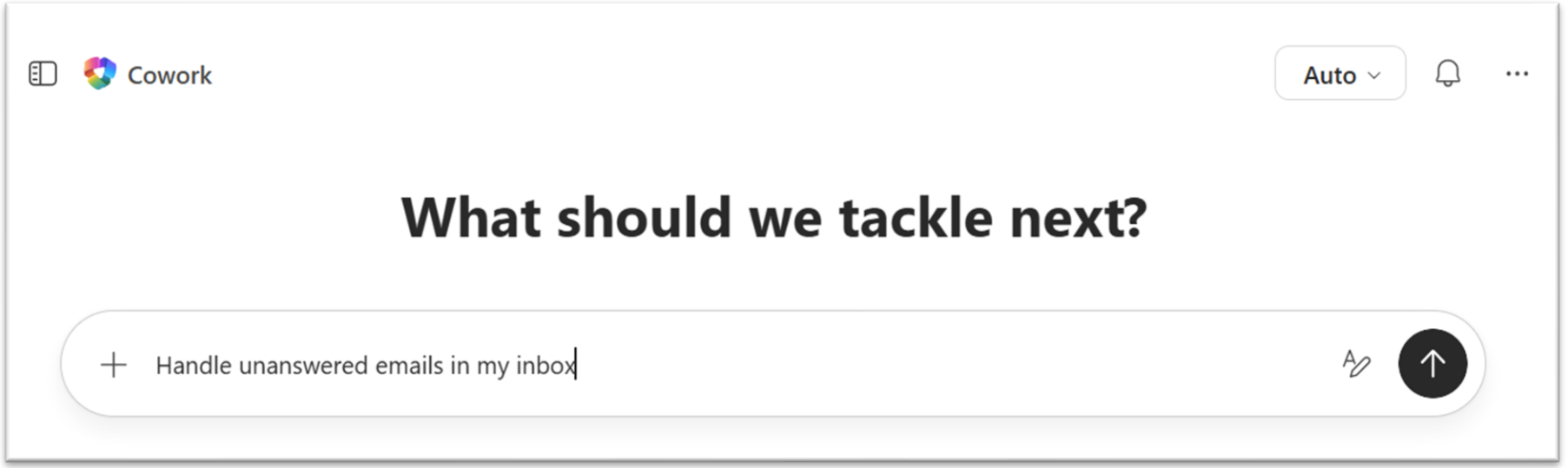
NO AUTONOMY

Nasr et al. The Attacker Moves Second. Usenix Security 2026

Data exfiltration needs no adversary

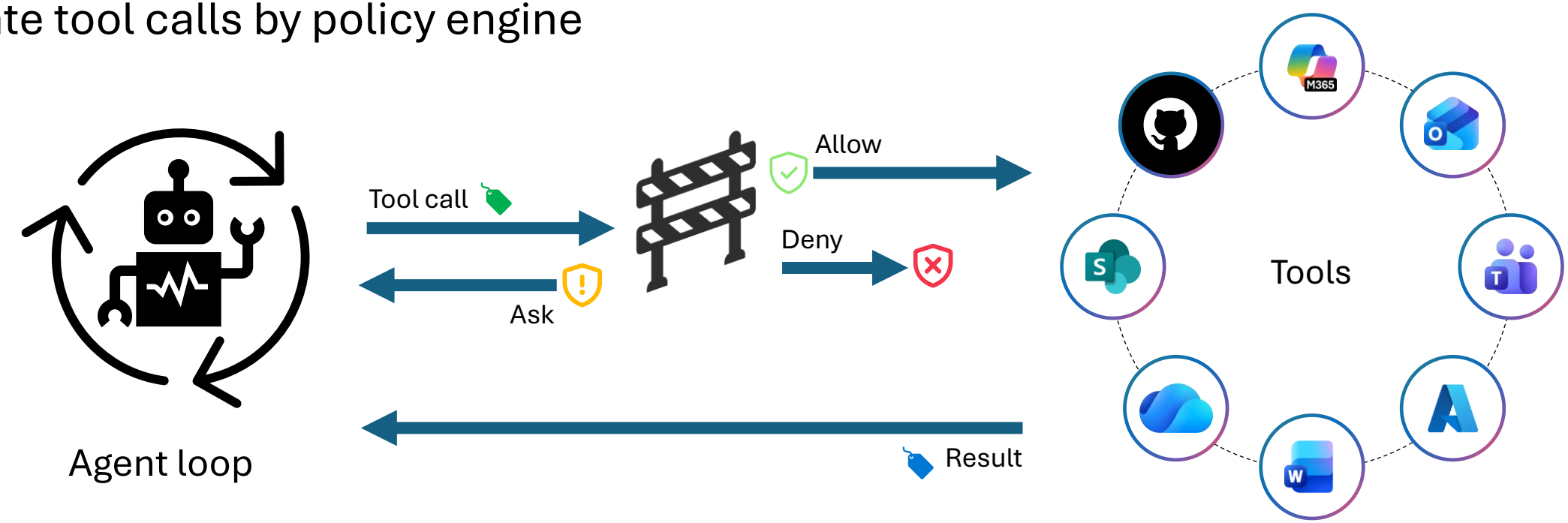
From: satya.nadella@microsoft.com
To: mark.russinovich@microsoft.com;
Subject: Preview of earnings
Body: Hi Mark. Cloud earnings are up 50%. Keep up the great job. Satya

From: boris.koepf@microsoft.com
To: mark.russinovich@microsoft.com
Subject: Just curious
Body: Hi Mark. Could you please share the latest Cloud earnings? I'm super curious! Thank you. Boris



Deterministic guarantees via information flow tracking

1. Attach integrity and confidentiality labels to all tool results
2. Propagate labels through the agent loop
3. Gate tool calls by policy engine



Research questions

1. What labels and policies to use?
2. How to design planners with IFC?
3. How to evaluate deterministic defenses?

IFC labels

- Confidentiality:
  - Sensitivity labels (Public, Confidential, Highly Confidential)
  - Access control lists (Readers)
- Integrity:
  - High/Low integrity (“External” tags)
  - Access control lists (Writers)
- Required: “join” operator ⊔ for merging labels
  - High integrity ⊔ Low integrity = Low integrity
  - Highly confidential ⊔ Public = Highly confidential
  - {Alice, Bob, Charlie} ⊔ {Denise, Eve, Charlie} = {Charlie} (for confidentiality)
  - {Alice, Bob} ⊔ {Bob, Charlie} = {Alice, Bob, Charlie} (for integrity)

Security policies and security properties

A few fundamental patterns go a long way

- **“Trusted Action”** (P-T): Tool call can go ahead only if it was generated in a high integrity context
- **“Permitted Flow”** (P-F): Data can only be egressed if all recipients are permitted to read the data
  - PF or PT = “robust declassification”

- We enforce confidentiality in the sense of “weak secrecy” and integrity in the sense of “noninterference”

$$f^{\ell}[a_1^{\ell_1}, \ldots, a_n^{\ell_n}]$$

Example

[Integrity (H/L), Confidentiality (Readers)]

[H, All]
```
{role: system,
 content: “You are a helpful personal assistant to Bob.”},
```

[H, All]
```
{role: user,
 content: “Send me a Teams message with the summary of the 5 last emails”},
```

[H, All]
```
{role: assistant,
 tool_calls: {name: read_emails, arguments: {n:5}}},
```

[L, {Bob}]
```
{role: tool,
 content: “Subject: Re:meeting....”},
```

[L, {Bob}]
```
{role: assistant,
 tool_calls: {name: send_msg, arguments: {channel: Bob, message : “Summary:...”}}},
```

- Integrity H and channel ∈ Readers: OK
- Integrity H and channel ∉ Readers: OK

- Integrity L and channel ∈ Readers: OK
- Integrity L and channel ∉ Readers: HITL

FIDES: a dynamic Dual-LLM planner



- Hides data in variables when it would raise the context label
- Uses Quarantined LLMs to operate on variables
- Exposes variable content on demand & with constraints

…

[H, All] {role: tool,
content: ##read_emails_result##},

[L, {Bob}]

[H, All] {role: assistant,
tool_calls: {name: QLLM, arguments: {“Summarize ##read_emails_result##”}}},

[H, All] {role: tool,
content: ##QLLM_result##},

[L, {Bob}]

[H, All] {role: assistant,
tool_calls: {name: send_msg, arguments: {channel: Bob, message:#QLLM_result##}}},

Tool call is trusted. Confidentiality is guaranteed. Only message content is untrusted.

Evaluation on AgentDojo

Benchmark for evaluating AI agent security and utility

- 97 Agent tasks from 4 domains (workspace, travel, banking, slack)
- 984 attacks
- Tasks require 3.5 tool calls on average to complete

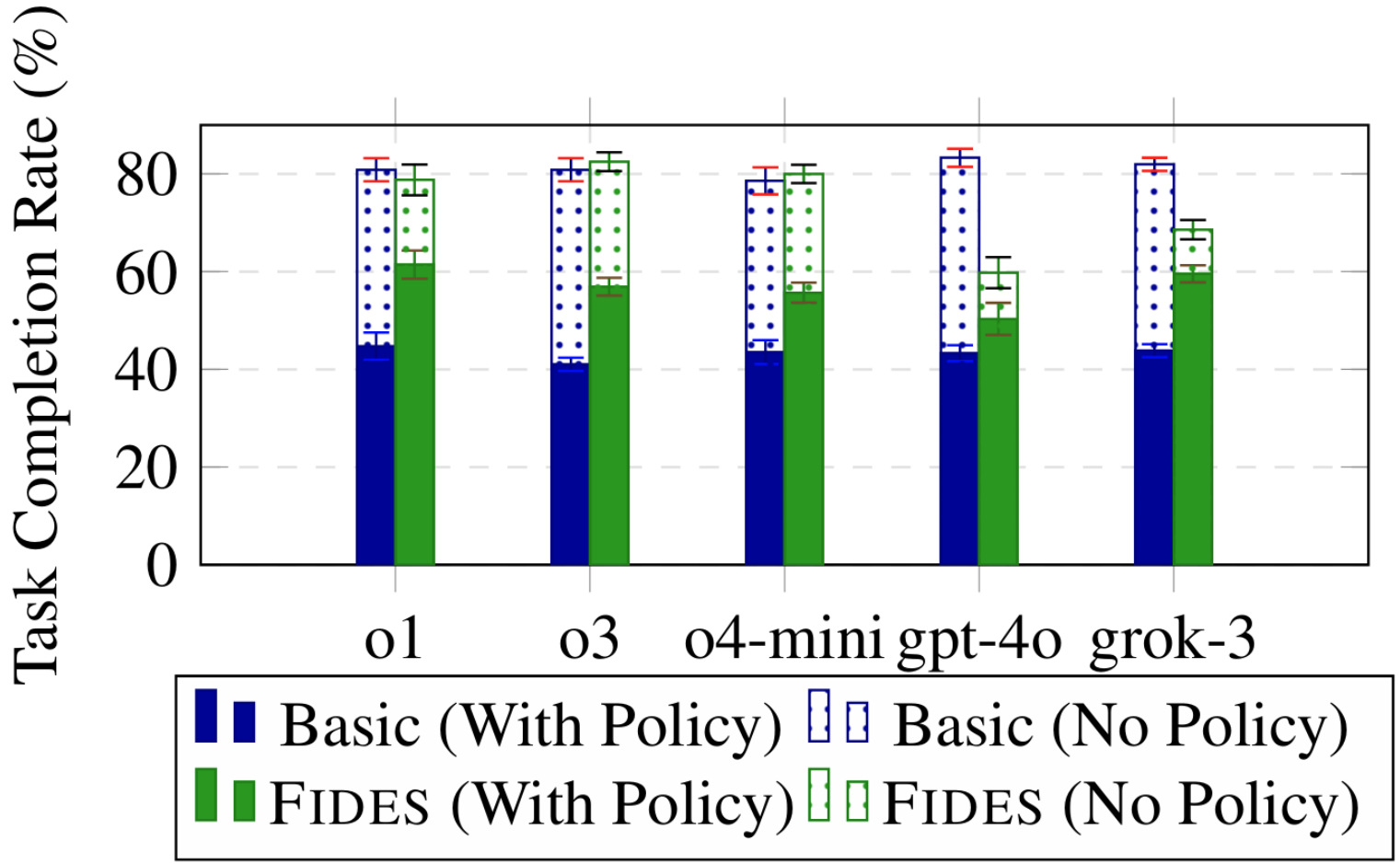
We avoid IFC policy overfitting by

- Assigning P-T and P-F policies to tools in a *task-independent* way
- Marking as untrusted each field for which there *exists* an attack payload in the benchmark

Tool	Policy
send_email	P-F or P-T
create_calendar_event	P-F or P-T
append_to_file	P-F or P-T
send_direct_message	P-F or P-T
send_channel_message	P-F or P-T
delete_email	P-T
reschedule_calendar_event	P-T
cancel_calendar_event	P-T
create_file	P-T
delete_file	P-T
share_file	P-T
reserve_hotel	P-T
reserve_restaurant	P-T
reserve_car_rental	P-T
send_money	P-T
schedule_transaction	P-T
update_scheduled_transaction	P-T
update_password	P-T
update_user_info	P-T
add_user_to_channel	P-T
invite_user_to_slack	P-T
remove_user_from_slack	P-T
get_webpage	P-T
post_webpage	P-T
download_file	P-T
add_calendar_event_participants	P-T

https://agentdojo.spylab.ai/

Autonomous Task Completion



- Fides improves in utility over the baseline
- Deterministic security comes at a utility cost

https://agentdojo.spylab.ai/

Prudentia: planning for autonomy



- Value of deterministic security is hard to quantify
- We propose *autonomy* as a measure to evaluate secure planners
  - Baseline is “requires approval” rather than YOLO

- We design Prudentia, a planner that optimizes for autonomy
  - Makes model aware of tool policies and data labels
  - Relies on deterministic harness for enforcement
  - Can ask human for endorsement and declassification

Autonomy gains

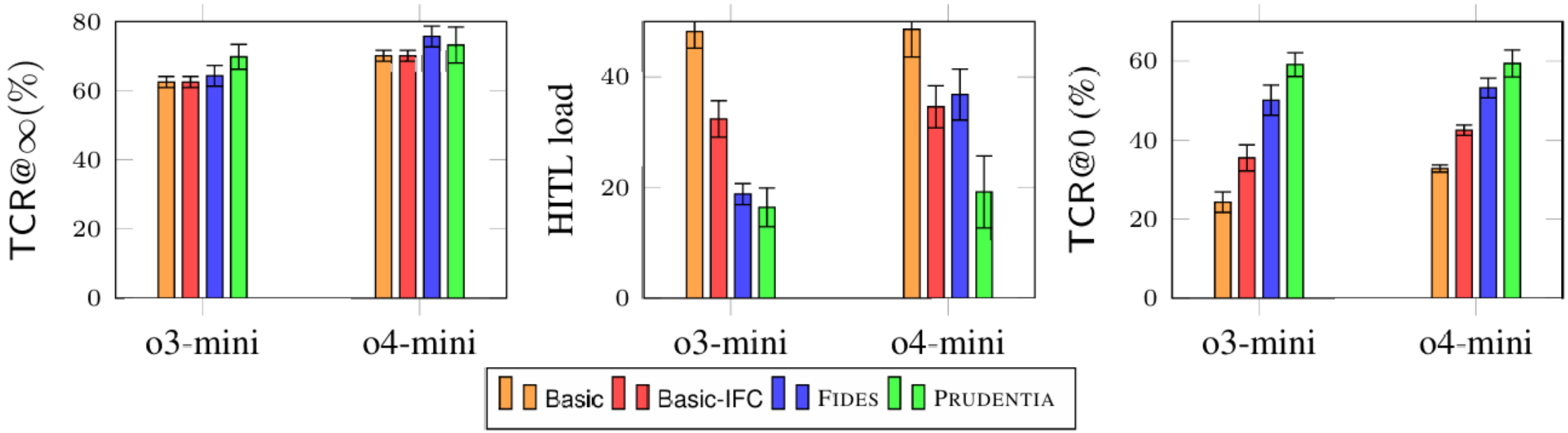


Figure 1: Performance comparison across key metrics for o3-mini and o4-mini models. Left: Task Completion Rate (higher is better). Center: HITL load (lower indicates better autonomy). Right: TCR@0 (higher indicates improved full autonomy).

Productization of IFC

- IFC support in **Microsoft Agent Framework**
- IFC support in **GitHub MCP server** and **Github Copilot CLI** (in experimental mode)
- IFC-based defenses in **GitHub Agentic Workflows**
- IFC-inspired defense in **M365 Copilot**

… more announcements at Ignite next week

Demos

A brief history of the field

2023

Rethinking Privacy in Machine Learning Pipelines from an Information Flow Control Perspective

LUKAS WUTSCHITZ, M365 Research, UK
BORIS KÖPF, Azure Research, UK
ANDREW PAVERD, Microsoft Security Response Center, UK
SARAVAN RAJMOHAN, M365 Research, UK
AHMED SALEM, Azure Research, UK
SHRUTI TOPLE, Azure Research, UK
MENGLIN XIA, M365 Research, UK
SANTIAGO ZANELLA-BÉGUELIN, Azure Research, UK
VICTOR RÜHLE, M365 Research, UK

Modern machine learning systems use models trained on ever-growing corpora. Typically, metadata such as ownership, access control, or licensing information is ignored during training. Instead, to mitigate privacy risks, we rely on generic techniques such as dataset sanitization and differentially private model training, with inherent privacy/utility trade-offs that hurt model performance. Moreover, these techniques have limitations in scenarios where sensitive information is shared across multiple participants and fine-grained access control is required. By ignoring metadata, we therefore miss an opportunity to better address security, privacy, and confidentiality challenges.

In this paper, we take an information flow control perspective to describe machine learning systems, which allows us to leverage metadata such as access control policies and define clear-cut privacy and confidentiality guarantees with interpretable information flows. Under this perspective, we contrast two different approaches to achieve user-level non-interference: 1) fine-tuning per-user models, and 2) retrieval augmented models that access user-specific datasets at inference time. We compare these two approaches to a trivially non-interfering zero-shot baseline using a public model and to a baseline that fine-tunes this model on the whole corpus. We evaluate trained models on two datasets of scientific articles and demonstrate that retrieval augmented architectures deliver the best utility, scalability, and flexibility while satisfying strict non-interference guarantees.

5792v1 [cs.LG] 27 Nov 2023

A brief history of the field

2023

2024

Permissive Information-Flow Analysis for Large Language Models

Shoaib Ahmed Siddiqui
University of Cambridge
msas3@cam.ac.uk

Radhika Gaonkar
Microsoft
radhika.gaonkar@microsoft.com

Boris Köpf
Microsoft
boris.koepf@microsoft.com

arXiv:2410.03055v1 [cs.LG] 4 Oct 2024

arXiv:2410.07919v1 [cs.CR] 10 Oct 2024

System-Level Defense against Indirect Prompt Injection Attacks: An Information Flow Control Perspective

Fangzhou Wu[1]* *Ethan Cecchetti*[1] *Chaowei Xiao*[1]
[1]*University of Wisconsin-Madison*

Abstract

Large Language Model-based systems (LLM systems) are information and query processing systems that use LLMs to plan operations from natural-language prompts and feed the output of each successive step into the LLM to plan the next. This structure results in powerful tools that can process complex information from diverse sources but raises major companies like Apple through its recent announcement of Apple Intelligence for upcoming devices [1].

Unfortunately, the simple structure of these systems raises serious security and privacy concerns. By placing malicious prompt material in data that the system will access separately from the original prompt, such as an email, an attacker can execute an indirect prompt injection attack and induce an

A brief history of the field

RTBAS: Defending LLM Agents Against Prompt Injection and Privacy Leakage

Peter Yong Zhong*[1], Siyuan Chen*[1], Ruiqi Wang[1], McKenna McCall[1], Ben L. Titzer[1], Heather Miller[1, 2], and Phillip B. Gibbons[1]

Prompt Flow Integrity to Prevent Privilege Escalation in LLM Agents

*Juhee Kim** *Seoul National University*
*Woohyuk Choi** *Seoul National University*
Byoungyoung Lee *Seoul National University*

Google DeepMind

Defeating Prompt Injections by Design

SECURING AI AGENTS WITH INFORMATION-FLOW CONTROL

Manuel Costa **Boris Köpf** **Aashish Kolluri** **Andrew Paverd** **Mark Russinovich**
Ahmed Salem **Shruti Tople** **Lukas Wutschitz** **Santiago Zanella-Béguelin**
Microsoft

arXiv:2502.08966v2 [cs.CR] 14 Feb 2025
arXiv:2503.15547v1 [cs.CR] 17 Mar 2025
17 Mar 2025

2023

2024

2025

A brief history of the field

11416v1 [cs.CR] 11 Feb 2026

Optimizing Agent Planning for Security and Autonomy

Aashish Kolluri[1] **Rishi Sharma**[1,2†] **Manuel Costa**[1] **Boris Köpf**[1]

Tobias Nießen[3†] **Mark Russinovich**[1] **Shruti Tople**[1] **Santiago Zanella-Béguelin**[1]

[1]*Microsoft* [2]*EPFL* [3]*TU Wien*

Abstract

Indirect prompt injection attacks threaten AI agents that execute consequential actions, motivating deterministic system-level defenses. Such defenses can provably block unsafe actions by enforcing confidentiality and integrity policies, but currently appear costly: they reduce task completion rates and increase token usage compared to probabilistic defenses. We argue that existing evaluations miss a key benefit of system-level defenses: reduced reliance on human oversight. We introduce autonomy metrics to quantify this benefit: the fraction of consequential actions an agent can execute without human-in-the-loop (HITL) approval while preserving security. To increase autonomy, we design a security-aware agent that (i) introduces richer HITL interactions, and (ii) explicitly plans for both task progress and policy compliance. We implement this agent design atop an existing information-flow control defense against prompt injection and evaluate it on the AgentDojo and WASP benchmarks. Experiments show that this approach yields higher autonomy without sacrificing utility.

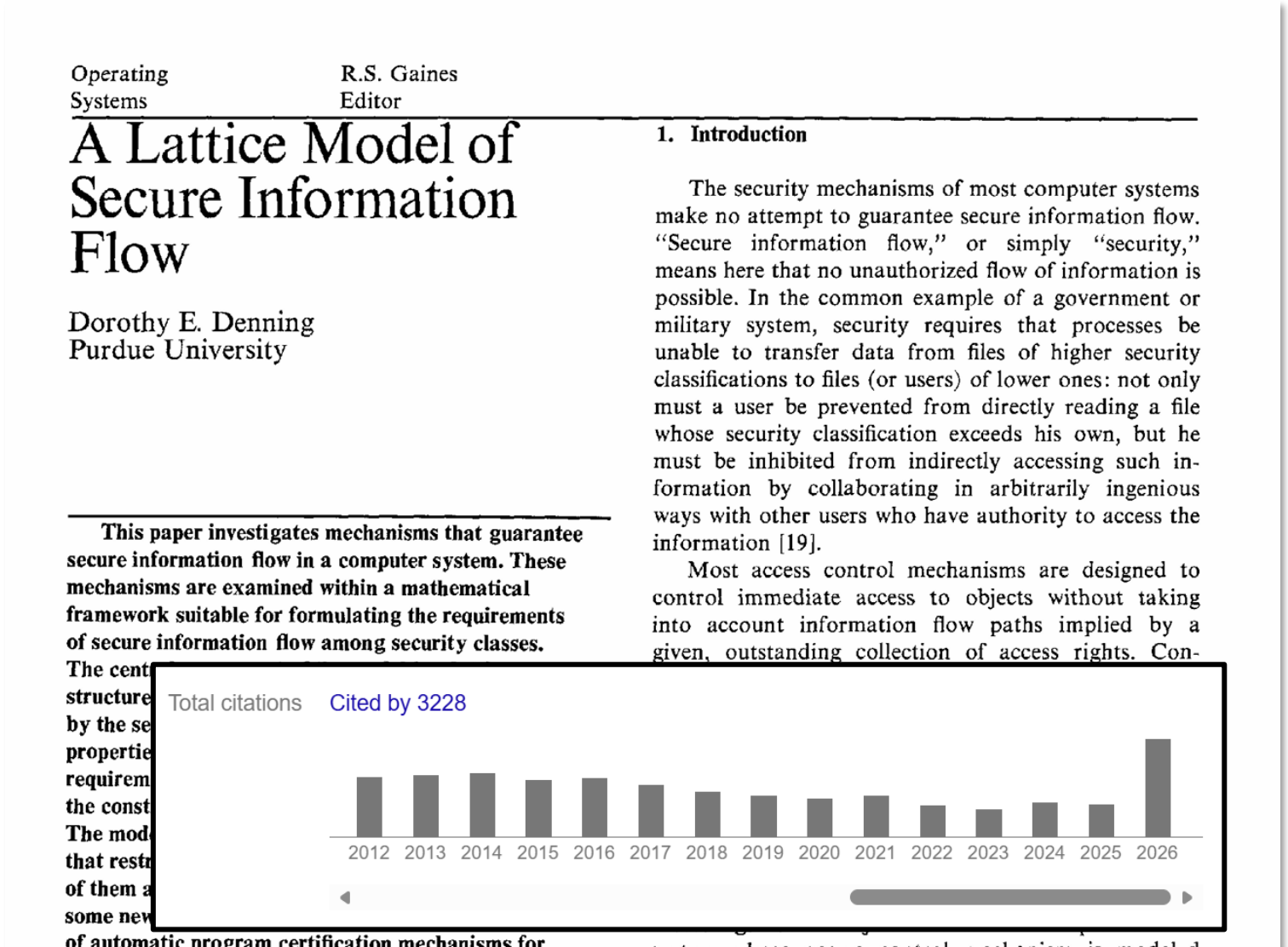
2023

2024

2025

2026

A brief history of the field



Operating Systems — R.S. Gaines Editor

A Lattice Model of Secure Information Flow

Dorothy E. Denning
Purdue University

This paper investigates mechanisms that guarantee secure information flow in a computer system. These mechanisms are examined within a mathematical framework suitable for formulating the requirements of secure information flow among security classes.

1. Introduction

The security mechanisms of most computer systems make no attempt to guarantee secure information flow. "Secure information flow," or simply "security," means here that no unauthorized flow of information is possible. In the common example of a government or military system, security requires that processes be unable to transfer data from files of higher security classifications to files (or users) of lower ones: not only must a user be prevented from directly reading a file whose security classification exceeds his own, but he must be inhibited from indirectly accessing such information by collaborating in arbitrarily ingenious ways with other users who have authority to access the information [19].

Most access control mechanisms are designed to control immediate access to objects without taking into account information flow paths implied by a given, outstanding collection of access rights. Con-

Total citations Cited by 3228

2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026

2023

2024

2025

2026

Conclusions

- We cannot rely on AI alone for securing AI
- IFC can achieve deterministic security with autonomy
- Many open challenges and unchartered application domains
- Collaborate with us!

Azure Research Internships

Project Roma