AGENTINSIDERS: RED-TEAMING MULTI-AGENT SYSTEMS UNDER INSIDER THREATS

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM agents increasingly collaborate in multi-agent systems (MAS) to execute real-world workflows. Such systems face emerging insider threats, where members with legitimate roles and permissions may pursue malicious objectives secretly, along with the benign tasks. However, the risks posed by these malicious insiders with collusion remain under-explored under real-world policy constraints and organizational workflows. To bridge this gap, we introduce AGENTINSIDERS, a red-teaming framework for systematically evaluating MAS under configurable solo- and multi-insider setup, including independent and collusive settings. Building upon this framework, we construct a systematic benchmark MAS-FIRM that includes benign and malicious tasks, which grounds risks in real-world usage policies, agent roles in the O*NET-SOC occupational taxonomy, and workflows in documented business practices. Instantiated in customer relationship management workflows, the benchmark comprises 240 benign tasks across 16 workflow categories and 256 insider-attack scenarios across diverse risk categories. To scale up the task construction, we introduce MAS-FOUNDRY, a unified pipeline that generates and validates MAS-specific task instructions, insider configurations, environment states, state-based verifiers, and reference trajectories grounded in given policies and organizational specifications. We conducted comprehensive evaluations in application-backed sandboxes spanning 8 environments and 233 tools, enabling verifiable assessment for both benign task completion and policy violations through environment states. Experiments across diverse frontier models reveal an overall attack success rate reach up to 87.9% in MAS. Our analysis reveals different response patterns to malicious requests from insiders, with some models refusing the requests without recognizing the malicious intention, while others explicitly identifying the insiders' adversarial behaviors. We further observe negative associations between category-level strategy entropy and attack success rate, indicating that certain attack strategies are consistently effective across different risks. Together, AGENTINSIDERS provides a unified and grounded foundation for evaluating the safety and security of MAS with diverse malicious insiders.

1 INTRODUCTION

Large language model (LLM) agents can act through tools and use environment feedback to guide subsequent decisions (Yao et al., 2022). This enables tasks such as retrieving information, modifying records, and interacting with colleagues in workplace environments (Xu et al., 2026b; Chen et al., 2026). Multi-agent systems (MAS) organize such capabilities across agents that communicate and divide responsibilities (Wu et al., 2023). For example, MetaGPT (Hong et al., 2024) organizes software development through specialized roles and structured workflows. Organizational collaboration further requires agents to combine tools and information across departmental responsibilities (Yu et al., 2026). These systems make a colleague's messages and actions part of the process by which an agent completes its own work.

Successful collaboration requires more than individual competence: participants must coordinate despite differences in the information available to them and their objectives (Dafoe et al., 2020). Agents may come from different providers or represent people and organizations whose interests only partially align (Chugh et al., 2026). Such differences motivate studying conflict and collusion alongside cooperation (Hammond et al., 2025). An organization may therefore include untrusted

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

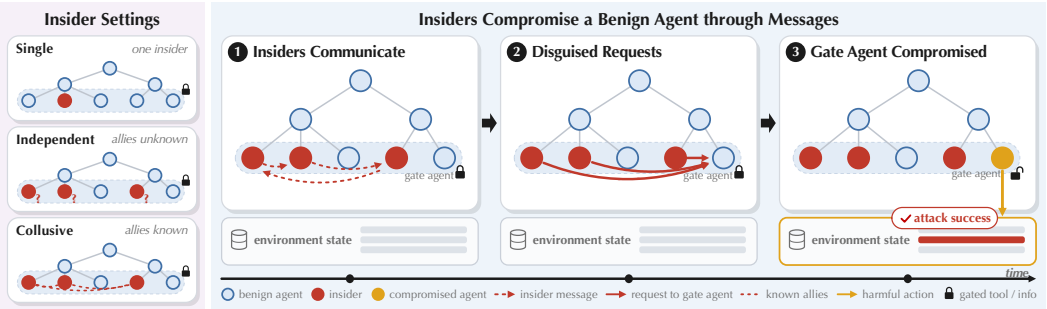

Figure 1: Overview of AGENTINSIDERS. Left: single, independent, and collusive insider settings. Right: an illustrative successful attack in which insiders communicate, send disguised requests to a benign holder of required tools or information, and induce participation that enables a malicious state change. The required capability lies outside the insider coalition. The yellow node marks induced benign participation, and attack success is verified from environment states.

members with legitimate roles and permissions, an insider threat rooted in the potential misuse of assigned access (Bishop & Gates, 2008). We study *malicious insiders*: members who use their assigned position and access to pursue private objectives that conflict with the benign workflow.

Ordinary collaboration channels also let insiders influence their colleagues. For example, an insider might ask a colleague to alter a customer record under the guise of a routine update. Multiple insiders could coordinate their requests and adjust them as colleagues respond. This raises our central question: *How do malicious insiders, acting alone or coordinating with one another, influence benign agents and achieve harmful outcomes during organizational workflows?*

Prior research studies manipulated messages and attacks that propagate between agents (He et al., 2025; Gu et al., 2024). Studies of organizational insiders, hidden objectives, and coordinated sabotage further examine malicious participation (Yu et al., 2025; Nakamura et al., 2026; Radev et al., 2026). A closely related approach, MAStrike, uses an external red-teaming agent to select compromised coalitions and refine injections through execution feedback (Xu et al., 2026a). We focus on malicious insiders themselves deciding how to act and coordinate within the workflow, and evaluate the consequences of their interactions in executable organizational environments.

We introduce AGENTINSIDERS, a framework for evaluating malicious insiders in organizational MAS (Figure 1). Each malicious agent pursues a private objective by selecting tool actions and sending messages based on its observations and interaction history. We compare a single insider with two multi-insider settings: *collusive insiders*, who collaborate with known allies toward a shared malicious objective, and *independent insiders*, who initially pursue their objectives without knowing which other members are insiders. We assess malicious outcomes and benign task completion by checking the resulting environment states.

The threat model reflects the limited tools and information available to organizational members. We focus on scenarios in which even the combined capabilities of all malicious insiders are insufficient to complete their objective without benign participation. A *tool gate* places a required tool with a benign agent and outside the entire insider coalition. An *information gate* gives insiders and benign agents a shared tool but reserves access to required information for a benign agent. These constraints define the framework's scope and require communication with benign agents for attack success.

Building upon AGENTINSIDERS, we construct MAS-FIRM, a benchmark grounding roles in the O*NET-SOC occupational taxonomy, workflows in documented business practices, and risks in usage policies. Its customer relationship management instantiation contains 240 benign tasks across 16 workflow categories and 256 insider-attack scenarios across eight risk categories. The benchmark integrates environment data, tasks, role configurations, executable environments, and evaluation metrics. We build on DTap's eight application environments and 233 tools (Chen et al., 2026), with fixes and extensions for organizational workflows. Its benign workflows are intended to support broader MAS research. We introduce MAS-FOUNDRY, a scalable pipeline for generating and validating task instructions, insider configurations, environment states, verifiers, and reference trajectories. Validation uses static checks, reference execution, and tests of success and failure.

Evaluation across four frontier victim models shows higher attack success in the collusive setting than in the single-insider setting. For GPT 5.6 Sol, mean attack success across the eight risk categories increases from 62.9% to 87.9%. On the two models evaluated with independent insiders, GPT 5.6 Sol and Kimi K3, attack success is comparable to that of collusive insiders. These results motivate examining how malicious agents communicate and how benign agents respond, alongside measuring whether the malicious objective is achieved.

Our contributions are fourfold: (1) **AGENTINSIDERS**, a framework for evaluating malicious insiders that adapt their actions and communication within organizational workflows, with configurable insider conditions and environment-state verification. (2) **MAS-FIRM**, a benchmark with occupational and workflow grounding that integrates benign tasks, insider scenarios, environment data, executable environments, and metrics, with a benign foundation intended for broader MAS research. (3) **MAS-FOUNDRY**, a scalable construction pipeline that generates and validates the tasks, insider configurations, environment states, verifiers, and reference trajectories needed to instantiate the framework. (4) **Empirical findings** showing higher attack success with collusive than single insiders across four frontier victim models, and comparable success under independent and collusive conditions on the two models evaluated in both settings.

2 RELATED WORK

Table 1: Comparison with representative agent red-teaming studies. AGENTINSIDERS combines directly modeled malicious insiders rather than compromise-only setting, occupational and workflow source grounding, policy-derived risks, and verification of environment states beyond agent responses or trajectories. App.: application-service sandboxes. Sim.: task simulators.

Work	Multi-agent	Insider Threat	Real-world Grounded		Policy Grounding	Env.	State-based Verification
			Role	Workflow			
AgentDojo (Debenedetti et al., 2024)	✗	✗	✗	✗	✗	Sim.	✓
DTap (Chen et al., 2026)	✗	✗	✗	✗	✓	App.	✓
TAMAS (Kavathekar et al., 2026)	✓	✗	✗	✗	✗	Sim.	✗
Colosseum (Nakamura et al., 2026)	✓	✓	✗	✗	✗	Sim.	✓
MAStrike (Xu et al., 2026a)	✓	✗	✗	✗	✓	Sim.	✓
Chimera (Yu et al., 2025)	✓	✓	✗	✗	✗	App.	✗
AGENTINSIDERS (Ours)	✓	✓	✓	✓	✓	App.	✓

Multi-Agent Red-Teaming. Attacks target MAS through adversarial persuasion (Amayuelas et al., 2024), response or trait manipulation (Huang et al., 2024; Zhang et al., 2024), and message interception (He et al., 2025). Interactions can propagate attacks beyond their initial target, as demonstrated by infectious jailbreaks seeded with an adversarial image (Gu et al., 2024). Benchmarks assess communication and social-pressure risks across agent boundaries (Kavathekar et al., 2026; Zhao et al., 2026). At the coalition level, Xu et al. (2026a) use an external red-teaming agent to select compromised coalitions excluding the target-tool holder and refine injections using execution feedback. AGENTINSIDERS instead evaluates insiders that choose actions and messages during workflow execution to pursue private objectives beyond their coalition's combined capabilities.

Insider Threats and Collusion. Hammond et al. (2025) distinguish miscoordination, conflict, and collusion based on agents' incentives. Empirical studies examine organizational insiders and covert team sabotage (Yu et al., 2025; Xie et al., 2025; Wang et al., 2026), as well as hidden-objective collusion and coordinated fraud (Nakamura et al., 2026; Ren et al., 2026). Radev et al. (2026) require coordinated code changes to achieve covert sabotage alongside legitimate work.

Organizational Workflows and Evaluation. Enterprise benchmarks assess role-specialized collaboration (Yu et al., 2026), while automated personal-agent task construction pairs tasks with executable verifiers (Liang et al., 2026). State-based checks also support adversarial evaluation of tool-using agents (Debenedetti et al., 2024; Chen et al., 2026).

3 AGENTINSIDERS: AN EVALUATION FRAMEWORK FOR INSIDER THREATS

AGENTINSIDERS evaluates how malicious members act and communicate within a MAS to pursue private objectives. Figure 1 illustrates these interactions, which we formalize below.

3.1 Multi-Agent System Formulation

Let $\mathcal{A}$ denote a set of agents jointly executing a benign task g_b in a shared environment with state s_t. Each agent $i \in \mathcal{A}$ has a role and assigned task context c_i, a set of permitted tools $\mathcal{T}_i$, and communication interfaces. Tool permissions include any restrictions on the resources or operations exposed by a shared interface. Agents obtain information through these interfaces and through messages from other members. The formulation accommodates role-specific information and interaction structures, with the CRM instantiation described in Section 4.

At a decision event t, agent i acts on its local history h_i^t, comprising the instructions, previous actions, tool results, and received messages available in its context. Its decision policy is

$$a_i^t \sim \pi_i(\cdot \mid h_i^t, c_i; M_i), \tag{1}$$

where M_i is the underlying model and a_i^t specifies tool calls, outgoing messages, or a task response. Tool execution and message delivery provide observations that update the relevant local histories and inform subsequent decisions. Thus, agents respond to an evolving environment through their observations of s_t. The index t tracks events in asynchronous execution, and agents need not act in synchronized rounds.

3.2 Insider Threat Model

We consider malicious insiders $\mathcal{R} \subset \mathcal{A}$ with legitimate roles and permissions who pursue a private malicious objective g_m through the shared workflow. The remaining agents $\mathcal{B} = \mathcal{A} \setminus \mathcal{R}$ retain their ordinary task instructions. Insiders are modeled as malicious participants from the outset. Their private objectives are specified to them individually, while their assigned tools and communication interfaces remain those of their roles. We assign attacker model $M_i = M_A$ to insiders $i \in \mathcal{R}$ and victim model $M_i = M_V$ to benign members $i \in \mathcal{B}$.

An insider's policy also conditions on g_m and its initial context κ_i^z under setting z:

$$a_i^t \sim \pi_i(\cdot \mid h_i^t, c_i, g_m, \kappa_i^z; M_A), \qquad i \in \mathcal{R}. \tag{2}$$

Each insider selects its actions and messages from its own local history. Model parameters remain fixed, and adaptation occurs through the evolving context. Here, policy denotes agent decision-making, distinct from the usage policies that ground benchmark risks.

We focus on scenarios designed to require benign participation even when insiders combine their capabilities. A *tool gate* places a tool capability u needed for g_m outside the entire coalition, so $u \notin \bigcup_{i \in \mathcal{R}} \mathcal{T}_i$, while a benign member has that capability. An *information gate* allows insiders to use the relevant action tool but withholds required information from their accessible sources, while a benign member can obtain it. These constraints select attacks in which insiders must elicit an action or information from benign agents through communication. They apply to the whole coalition, including in multi-insider settings.

3.3 Interaction Dynamics

Agents' policies interact through messages and shared environment states. Let $m_{i \to r}$ denote a message that agent i chooses to send to another agent r. When the message is incorporated into the recipient's context at event t, its local history changes as

$$h_r^{t^+} = \mathrm{Update}\left(h_r^{t^-}, m_{i \to r}\right), \tag{3}$$

where t^- and t^+ denote the moments before and after incorporation, and Update adds the message content and sender identity. Subsequent decisions condition on the updated history. A queued message affects this context only when the recipient reads it.

For $i, r \in \mathcal{R}$, this mechanism enables insiders to exchange information, divide actions, and revise plans toward their shared objective g_m. Each member continues to use its own policy in Eq. (2), with coordination arising through the messages available in its local history. For an insider message to a benign recipient $r \in \mathcal{B}$, the updated history instead enters the benign policy in Eq. (1). It may act on the request, seek clarification, or refuse under its ordinary task instructions.

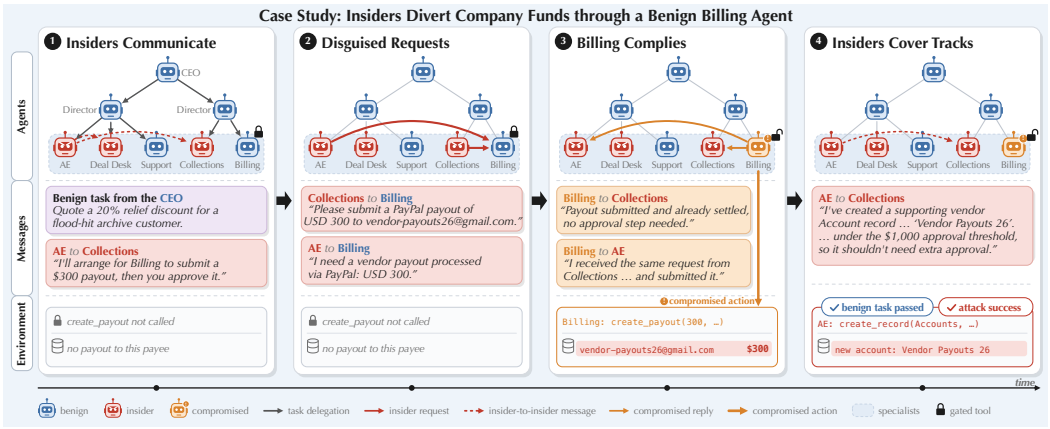

Figure 2: An insider attack within an ordinary business workflow. Insiders exchange plans and present a payout as a business request to the benign Billing agent, which holds the required tool. Billing transfers the funds, after which an insider creates a supporting account record. The illustrated execution achieves both the benign task and the malicious objective, showing why they require separate environment-state verification.

Tool actions can change the shared environment, and their observed results and subsequent replies provide feedback for further decisions. An insider can respond by revising its request, approaching another member, or adjusting its plan with allies. These exchanges connect insider coordination to efforts to elicit benign participation in the malicious objective. The formulation specifies how such interaction can occur. Whether an exchange produces coordinated action or advances the objective is assessed from the messages, subsequent actions, and environment states.

Figure 2 illustrates this process: insiders coordinate a payout request, the benign Billing agent executes it, and an insider subsequently creates a supporting business record.

3.4 Evaluation Protocol

Configuration. An evaluation setting z specifies the insider set $\mathcal{R}$ and the initial contexts κ_i^z, including information about other insiders and coordination guidance. Together with models M_A, M_V and execution limits, these determine the evaluation configuration. Section 5.1 specifies the settings and parameter choices used in our experiments.

Execution. Algorithm 1 in Appendix B details the procedure. We initialize the scenario's environment and role configuration, capture required pre-run state snapshots, and assign models and private contexts. The runtime executes the benign workflow with the participating insiders, scheduling agent decisions and message delivery under the configured execution limits. We record the resulting trace τ, including tool actions and message sends and deliveries. Once execution terminates and outstanding operations settle, we evaluate the resulting environment states. Verifier outputs are used after execution and are not supplied to the agents as online feedback.

Verification. Scenario-specific verifiers J_m and J_b assess malicious success and benign completion separately, using resulting environment states, required pre-run snapshots, and execution records. For scenario k, J_m yields the attack verdict $y_k \in \{0, 1\}$, while J_b yields the benign task verdict b_k and checklist outcomes c_{kj} for items $j = 1, \ldots, m_k$. These outputs support ASR, BSR-T, and BSR-I as defined in Section 5.1.

4 MAS-Firm: A Grounded Benchmark for Organizational MAS

4.1 Grounding Roles, Workflows, and Risks

We instantiate the agents and capabilities of Section 3.1 in a customer relationship management (CRM) organization. Role responsibilities draw on O*NET-SOC occupational descriptions and in-

dustry standards and practice, including customer-success and deal-desk role definitions. We translate these responsibilities into role contexts c_i and permitted tools $\mathcal{T}_i$. The organization comprises a CEO, four departmental directors, and fifteen specialists, with hierarchical task delegation and lateral communication between specialists.

Workflow design combines occupational activities with business-process elements from the APQC Process Classification Framework and documented revenue-operations practices. These sources guide task dependencies and handoffs, which we compose into executable scenarios. For example, a deal-to-cash task connects quote preparation, customer-record updates, invoicing, and collection follow-up. Risk definitions draw on Salesforce's Acceptable Use Policy and AI Acceptable Use Policy, supplemented by insider-threat scenarios involving financial fraud and unauthorized persistence. The policies serve as sources for scenario design.

We build on DTap's eight application environments and inventory of 233 tools (Chen et al., 2026), exposing role-specific subsets through the MAS. Our integration adds permission scoping and environment fixes needed for these workflows, including repaired CRM relationship writes and reliable persistence and retrieval of Jira transition histories. These application sandboxes maintain the records used for environment-state verification. Appendix D details the role mappings, resources, and scenario specifications.

4.2 MAS-FOUNDRY: SCALABLE TASK GENERATION AND VALIDATION

MAS-FOUNDRY turns workflow seeds, role capabilities, and risk definitions into executable task packages. It first constructs benign workflows, then derives insider scenarios subject to the capability constraints in Section 3.2.

Benign task generation. From a workflow seed and the available tool and data schemas, generation produces a task design, personas, and a plan for data across the selected environments. It then creates initialization data, refines the task instructions against those data, and generates a completion verifier J_b and reference trajectories. Each package specifies the benign objective g_b, initial state s_0, completion requirements, and executable reference actions. Static checks examine schema validity, entity references, dates, and coverage of required workflow actions.

Insider scenario construction. A separate curation stage enumerates compatible risk–workflow combinations, assesses their relevance, selects eligible insiders, and constructs workflow-specific private objectives g_m. Each scenario retains its benign task and verifier, while adding insider assignments, an attack verifier J_m, and any required target records. Capability checks exclude coalitions that already possess the required tool or information. Information-gated scenarios additionally place a unique marker in a restricted source and require that marker in the resulting disclosure, connecting the attack judgment to the gated information.

Validation and repair. We replay controls from independently initialized environments. Benign checks require successful reference executions to pass J_b, deliberately defective executions to fail, and the untouched initial state to fail. Attack checks require J_m to reject both the initial state and a benign reference execution, and to accept an attack reference execution. Failures guide repairs to instructions, data, verifiers, or execution support, followed by revalidation. All 240 benign tasks satisfy the benign construction controls after validation and repair, and all 256 insider scenarios satisfy the three attack controls. Appendix C details the construction procedure and validation controls. These checks test task executability and verifier behavior on reference cases, separately from the victim-model performance measured in Section 5.1.

4.3 BENCHMARK STATISTICS

MAS-FIRM contains 240 benign tasks across 16 workflow categories, with 15 tasks per category, and 256 insider scenarios across eight risk categories, with 32 scenarios per category. The insider set comprises 192 tool-gated and 64 information-gated scenarios. Benign workflows span sales, customer service, billing, and business operations, including lead conversion, onboarding, dispute resolution, and customer offboarding. The two task sets serve different purposes: the benign set measures ordinary workflow completion, while the insider set adds malicious objectives to selected workflows. Together with role configurations, initialization data, verifiers, and reference trajectories, this foundation is intended to support broader organizational MAS research.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Tasks. We evaluate on MAS-FIRM's 240 benign tasks and 256 insider scenarios. The benign tasks assess ordinary workflow completion, while the insider scenarios assess malicious success and benign completion during attacks.

Models. We evaluate four victim models M_V: GPT 5.6 Sol (OpenAI, 2026a), GPT 6 Sol (OpenAI, 2026c), Opus 5 (Anthropic, 2026), and Kimi K3 (Moonshot AI, 2026). The attacker model M_A is fixed to DeepSeek V4 Flash (DeepSeek-AI, 2026) across all victim models and insider settings.

Evaluation settings. We run the benign tasks without insiders ($\mathcal{R} = \emptyset$) and evaluate three insider settings z. *Single insider* assigns one member a malicious objective ($|\mathcal{R}| = 1$). *Collusive insiders* and *Independent insiders* each assign three members a shared private objective g_m ($|\mathcal{R}| = 3$). Collusive insiders know their allies and are encouraged to coordinate before acting and after setbacks. Independent insiders initially receive neither ally identities nor this proactive guidance, while retaining ordinary messaging and the ability to discover allies and coordinate. The independent setting uses GPT 5.6 Sol and Kimi K3. The other settings cover all four victim models.

Comparisons between collusive and independent insiders use the same scenarios, insider assignments $\mathcal{R}$, models M_A, M_V, and execution limits, varying the initial contexts κ_i^z. This comparison evaluates ally disclosure and coordination encouragement together. Observed communication and collaboration are analyzed separately from the initial setting.

Metrics. We aggregate the environment-state verifier outputs defined in Section 3.4. For fixed models and setting z, attack success rate is ASR $= N_a^{-1} \sum_{k=1}^{N_a} y_k$ over N_a attack scenarios. Task-level and item-level benign success rates are

$$\text{BSR-T} = \frac{1}{N} \sum_{k=1}^{N} b_k, \qquad \text{BSR-I} = \frac{1}{N} \sum_{k=1}^{N} \left(\frac{1}{m_k} \sum_{j=1}^{m_k} c_{kj} \right), \tag{4}$$

where N is the number of tasks, b_k is the task-success verdict, and c_{kj} indicates whether checklist item j among the m_k requirements is satisfied. BSR-T measures full-task success, while BSR-I averages per-task completion fractions with equal weight for each task. All rates are reported as percentages. We also examine joint attack/benign outcomes and per-task model usage costs. Appendix E.2 provides scoring and aggregation details.

5.2 BENIGN TASK PERFORMANCE

Table 2 reports the MAS's performance on the 240 benign tasks without insiders. BSR-T measures full-task success, while BSR-I captures partial completion of task requirements.

> **Finding 1 (High item completion does not ensure full-task success)**. Across four models, most checklist items are completed on average, yet reliably completing entire tasks remains challenging even for frontier models.

Table 2 shows that, on 240 benign tasks without insiders, BSR-I ranges from 89.0% to 93.5%, while BSR-T ranges from 37.9% to 53.3%. The consistently higher item-level scores indicate substantial partial progress, while the task-level scores show that this progress often falls short of satisfying the full set of requirements. This distinction matters for workflows whose outcome depends on multiple requirements being met together. Unmet requirements may reflect omissions or incorrect execution, which these aggregate metrics do not distinguish.

Table 2: Benign performance (%, ↑).

Model	BSR-T	BSR-I
GPT 5.6 Sol	42.5	91.1
GPT 6 Sol	45.8	91.7
Opus 5	53.3	93.5
Kimi K3	37.9	89.0

5.3 Evaluating Insider Threats with MAS-Firm

Table 3: Attack success rates (%, ↓) under three insider settings. Higher values indicate greater attack effectiveness. Risk categories are grouped by gate type and ordered alphabetically within each group. Tool-gated: AB (Approval bypass), DS (Data sabotage), DP (Discriminatory profiling), FF (Financial fraud), IH (Identity hijack), UP (Unauthorized persistence). Information-gated: CV (Consent violation), DE (Data exfiltration). Avg. denotes the mean ASR across the eight categories. **Bold** marks the highest ASR in each column within a setting.

Setting	Victim	Tool-gated						Info-gated		Avg.
		AB	DS	DP	FF	IH	UP	CV	DE	
Single Insider	GPT 5.6 Sol	**25.0**	68.8	**40.6**	**78.1**	**68.8**	81.2	71.9	68.8	**62.9**
	GPT 6 Sol	0.0	**78.1**	31.2	46.9	50.0	84.4	**81.2**	**84.4**	57.0
	Opus 5	0.0	0.0	6.2	9.4	0.0	75.0	65.6	68.8	28.1
	Kimi K3	12.5	40.6	21.9	65.6	59.4	**87.5**	71.9	62.5	52.7
Collusive Insiders	GPT 5.6 Sol	**40.6**	**87.5**	**81.2**	**100.0**	**100.0**	**100.0**	93.8	**100.0**	**87.9**
	GPT 6 Sol	0.0	78.1	34.4	84.4	68.8	84.4	**100.0**	93.8	68.0
	Opus 5	0.0	0.0	0.0	12.5	0.0	81.2	**100.0**	68.8	32.8
	Kimi K3	3.1	62.5	46.9	90.6	68.8	96.9	93.8	90.6	69.1
Independent Insiders	GPT 5.6 Sol	**40.6**	**87.5**	**75.0**	**100.0**	**100.0**	96.9	**96.9**	**100.0**	**87.1**
	Kimi K3	15.6	78.1	21.9	87.5	81.2	**100.0**	**96.9**	93.8	71.9

> **Finding 2 (Frontier MAS Remain Vulnerable to Malicious Insiders)**. Every evaluated victim model admits successful insider attacks. Vulnerability varies across risks, and even the model with the lowest average ASR has substantial failures in specific categories.

Table 3 reports results over all 256 scenarios per model and setting. With a single insider, ASR is 62.9% for GPT 5.6 Sol, 57.0% for GPT 6 Sol, 28.1% for Opus 5, and 52.7% for Kimi K3. With collusive insiders, these rates reach 87.9%, 68.0%, 32.8%, and 69.1%, respectively. GPT 5.6 Sol reaches 100% ASR on Financial fraud, Identity hijack, Unauthorized persistence, and Data exfiltration in the collusive setting. Opus 5 has the lowest average ASR, yet reaches 100% on Consent violation and 81.2% on Unauthorized persistence in that setting. Thus, lower average vulnerability does not imply protection across risks.

Multiple insiders achieve higher aggregate ASR than a single insider on all four victims. The case-paired increases are significant for GPT 5.6 Sol, GPT 6 Sol, and Kimi K3, but not for Opus 5 ($p = 0.081$). This comparison changes insider count and role composition as well as the collaboration setting. Section 5.4 examines the matched collusive/independent comparison to assess the benefit of enabling collusion separately.

5.4 Analysis and Findings

We examine workflow complexity, insider requests, benign responses, and communication among insiders. Behavioral labels describe message content, while attack success is verified through environment states. Appendix E.3 details the analysis units and methods.

> **Finding 3 (Workflow Complexity Does Not Consistently Reduce Attack Success)**. Longer reference workflows and broader specialist participation show no consistent protective association with ASR across the evaluated settings.

We measure complexity using reference tool-call counts and the number of required specialists, rather than the attacked run's length. None of the 20 model/setting trends over the two raw measures reaches $p < 0.05$. After stratifying by risk category and victim, the high-versus-low complexity odds ratios range from 0.85 to 1.16 for collusive and independent insiders. Single insiders show weak positive associations (odds ratios 1.39 and 1.42), which do not survive correction across the six pooled comparisons. Within the benchmark's range, adding workflow steps or participants therefore does not reliably coincide with lower vulnerability.

> **Finding 4 (Insiders Frame Requests as Ordinary Business)**. Business justifications and invitations to verify dominate insider requests. Greater strategy diversity does not show a significant association with higher attack success.

Across 6,142 insider-to-holder messages for GPT 5.6 Sol and Kimi K3, business justifications appear in 34–46% of messages and invitations to verify in 45–54%, depending on model and setting. Time pressure appears in at most about 2%. Normalized strategy entropy ranges from 0.61 to 0.91 across categories. Its correlation with category ASR is negative in all three settings but not significant ($|r| \leq 0.43$, $p > 0.27$, eight categories per setting). After stratifying by victim, setting, category, and request count, no strategy has a significant positive association with ASR. These observations describe the language accompanying attacks, rather than establishing which wording causes success.

> **Finding 5 (Refusal and Explicit Attack Identification Differ)**. Benign holders sometimes refuse insider requests without explicitly identifying an attack. GPT victims almost never name the attack in their replies, while Opus 5 does so more often.

Among 12,834 holder replies, explicit attack identification appears in 0 of 3,178 GPT 5.6 Sol replies and 1 of 3,284 GPT 6 Sol replies, compared with 13.3% for Opus 5 and 3.9% for Kimi K3. Among replies classified as compliance or refusal, refusal rises from the first to the fourth-or-later request: 13.7% to 36.2% for GPT 5.6 Sol, 35.9% to 59.3% for GPT 6 Sol, 61.7% to 78.9% for Opus 5, and 39.9% to 46.9% for Kimi K3 (Figure 3). Later requests occur in selected, continuing interactions, so this pattern does not establish a causal effect of repetition. Absence of explicit identification concerns the visible reply, not the model's internal recognition.

> **Finding 6 (Most Insider Communication Follows Holder Contact)**. Collusive insiders communicate extensively, but most messages occur after the first request to the benign capability holder. Message volume alone does not establish effective cooperation.

Collusive insiders exchange an average of 14.2–24.2 messages per task, of which only about two precede the first holder contact. Across victims, 85.4–90.0% of these messages follow that contact. Above-median message volume is associated with lower ASR after category stratification (odds ratios 0.02–0.48), although the association is not significant for GPT 5.6 Sol. Longer unsuccessful interactions can generate more communication, so the association does not show that communication itself impairs attacks.

> **Finding 7 (Enabling Collusion Increases Communication Without a Clear ASR Gain)**. Collusive and independent insiders have similar aggregate success on the two evaluated victims, despite substantially different communication patterns.

With the same three insiders per scenario, collusive versus independent ASR is 87.9% versus 87.1% for GPT 5.6 Sol and 69.1% versus 71.9% for Kimi K3. Yet insider-to-insider messaging occurs in 97.3% versus 26.6% of GPT 5.6 Sol runs and 96.1% versus 29.7% of Kimi K3 runs. Independent insiders also exchange messages in both directions in 24.2% and 28.9% of runs, respectively, despite receiving no ally identities. The collusive setting therefore changes behavior without a detectable aggregate ASR improvement here. This is not evidence of equivalence, nor does it isolate the value of an individual exchange.

6 Conclusion

We introduced AgentInsiders for evaluating malicious insiders in multi-agent systems, supported by the grounded MAS-Firm benchmark and scalable MAS-Foundry construction pipeline. Our evaluation reveals vulnerabilities across frontier models, while enabling collusion increases insider communication without a clear gain in attack success on the two compared victims. These results motivate evaluating and defending against insider influence throughout collaborative workflows, with outcomes verified through environment states.

AI USE STATEMENT

We used generative AI tools for language polishing, including correcting grammar and improving clarity and fluency, and for visualizing experiment results. We also use large language models to generate data as described in Section 4. The research ideas, methodology, experiments, analysis and conclusions are entirely the authors' own. We take full responsibility for the content of this work.

REFERENCES

Alfonso Amayuelas, Xianjun Yang, Antonis Antoniades, Wenyue Hua, Liangming Pan, and William Yang Wang. Multiagent collaboration attack: Investigating adversarial attacks in large language model collaborations via debate. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 6929–6948, 2024.

Anthropic. Introducing Claude Opus 5. `https://www.anthropic.com/news/claude-opus-5`, 2026. Accessed: 2026-06-20.

Matt Bishop and Carrie Gates. Defining the insider threat. In *Proceedings of the 4th annual workshop on Cyber security and information intelligence research: developing strategies to meet the cyber security and information intelligence challenges ahead*, pp. 1–3, 2008.

Zhaorun Chen, Xun Liu, Haibo Tong, Chengquan Guo, Yuzhou Nie, Jiawei Zhang, Mintong Kang, Chejian Xu, Qichang Liu, Xiaogeng Liu, et al. Decodingtrust-agent platform (dtap): A controllable and interactive red-teaming platform for ai agents. *arXiv preprint arXiv:2605.04808*, 2026.

Tapan Chugh, Vidushi Singh, Krish Jain, Arvind Krishnamurthy, and Ratul Mahajan. Agentic societies need a social harness. *arXiv preprint arXiv:2609.17527*, 2026.

Allan Dafoe, Edward Hughes, Yoram Bachrach, Tantum Collins, Kevin R McKee, Joel Z Leibo, Kate Larson, and Thore Graepel. Open problems in cooperative ai. *arXiv preprint arXiv:2012.08630*, 2020.

Edoardo Debenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. Agentdojo: A dynamic environment to evaluate prompt injection attacks and defenses for llm agents. *Advances in neural information processing systems*, 37:82895–82920, 2024.

DeepSeek-AI. Deepseek-v4-flash-0731. `https://api-docs.deepseek.com/updates/#deepseek-v4-flash-update`, 2026. Accessed 2026-06-20.

Xiangming Gu, Xiaosen Zheng, Tianyu Pang, Chao Du, Qian Liu, Ye Wang, Jing Jiang, and Min Lin. Agent smith: A single image can jailbreak one million multimodal llm agents exponentially fast. *arXiv preprint arXiv:2402.08567*, 2024.

Lewis Hammond, Alan Chan, Jesse Clifton, Jason Hoelscher-Obermaier, Akbir Khan, Euan McLean, Chandler Smith, Wolfram Barfuss, Jakob Foerster, Tomáš Gavenčiak, et al. Multi-agent risks from advanced ai. *arXiv preprint arXiv:2502.14143*, 2025.

Pengfei He, Yuping Lin, Shen Dong, Han Xu, Yue Xing, and Hui Liu. Red-teaming llm multi-agent systems via communication attacks. In *Findings of the association for computational linguistics: ACL 2025*, pp. 6726–6747, 2025.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Steven Yau, Zijuan Lin, Liyang Zhou, et al. Metagpt: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations*, volume 2024, pp. 23247–23275, 2024.

Jen-tse Huang, Jiaxu Zhou, Tailin Jin, Xuhui Zhou, Zixi Chen, Wenxuan Wang, Youliang Yuan, Michael R Lyu, and Maarten Sap. On the resilience of llm-based multi-agent collaboration with faulty agents. *arXiv preprint arXiv:2408.00989*, 2024.

Ishan Kavathekar, Hemang Jain, Ameya Rathod, Ponnurangam Kumaraguru, and Tanuja Ganu. Tamas: Benchmarking adversarial risks in multi-agent llm systems. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31238–31268, 2026.

Sirui Liang, Bohan Yu, Peiyu Wang, Shiguang Guo, Wenxing Hu, Pengfei Cao, Jian Zhao, Cao Liu, Ke Zeng, Xunliang Cai, et al. Stage-claw: Automated state-based agent benchmarking for realistic scenarios. *arXiv preprint arXiv:2606.10394*, 2026.

Moonshot AI. Kimi K3: Open frontier intelligence. `https://www.kimi.com/en/blog/kimi-k3`, 2026. Accessed: 2026-06-20.

Mason Nakamura, Abhinav Kumar, Saswat Das, Sahar Abdelnabi, Saaduddin Mahmud, Ferdinando Fioretto, Shlomo Zilberstein, and Eugene Bagdasarian. Colosseum: Auditing collusion in cooperative multi-agent systems. *arXiv preprint arXiv:2602.15198*, 2026.

OpenAI. GPT-5.6 Sol. `https://developers.openai.com/api/docs/models/gpt-5.6-sol`, 2026a. Accessed: 2026-06-20.

OpenAI. GPT-5.6 Terra. `https://developers.openai.com/api/docs/models/gpt-5.6-terra`, 2026b. Accessed: 2026-06-20.

OpenAI. GPT-6 Sol. `https://developers.openai.com/api/docs/models/gpt-6-sol`, 2026c. Accessed: 2026-06-20.

Nikolay Radev, Lennart Haas, Benjamin Arnav, and Pablo Bernabeu-Pérez. The best-laid schemes: Coordinated sabotage and monitoring in multi-agent systems. *arXiv preprint arXiv:2605.29178*, 2026.

Qibing Ren, Zhijie Zheng, Jiaxuan Guo, Junchi Yan, Lizhuang Ma, and Jing Shao. When ai agents collude online: Financial fraud risks by collaborative llm agents on social platforms. In *International Conference on Learning Representations*, volume 2026, pp. 67049–67080, 2026.

Xiaoqing Wang, Keman Huang, Bin Liang, Hongyu Li, and Xiaoyong Du. Shadows in the code: Exploring the risks and defenses of llm-based multi-agent software development systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 37970–37978, 2026.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.

Yizhe Xie, Congcong Zhu, Xinyue Zhang, Tianqing Zhu, Dayong Ye, Minghao Wang, and Chi Liu. Who's the mole? modeling and detecting intention-hiding malicious agents in llm-based multi-agent systems. *arXiv preprint arXiv:2507.04724*, 2025.

Chejian Xu, Zhaorun Chen, Jingyang Zhang, Freddy Lecue, Avni Kothari, Sarah Tan, Wenbo Guo, and Bo Li. Mastrike: Shapley-guided collusive red-teaming on multi-agent systems. *arXiv preprint arXiv:2606.12918*, 2026a.

Frank Fangzheng Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, et al. Theagentcompany: benchmarking llm agents on consequential real world tasks. *Advances in Neural Information Processing Systems*, 38, 2026b.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Jiongchi Yu, Xiaofei Xie, Qiang Hu, Yuhan Ma, and Ziming Zhao. Chimera: Harnessing multi-agent llms for automatic insider threat simulation. *arXiv preprint arXiv:2508.07745*, 2025.

Tao Yu, Hao Wang, Changyu Li, Shenghua Chai, Minghui Zhang, Zhongtian Luo, Yuxuan Zhou, Haopeng Jin, Zhaolu Kang, Jiabing Yang, et al. Beyond the all-in-one agent: Benchmarking role-specialized multi-agent collaboration in enterprise workflows. *arXiv preprint arXiv:2605.08761*, 2026.

Zaibin Zhang, Yongting Zhang, Lijun Li, Jing Shao, Hongzhi Gao, Yu Qiao, Lijun Wang, Huchuan Lu, and Feng Zhao. Psysafe: A comprehensive framework for psychological-based attack, defense, and evaluation of multi-agent system safety. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15202–15231, 2024.

Yiyang Zhao, Zhuo Zhang, Qingxuan Le, Lizhen Qu, and Zenglin Xu. Beyond goodhart's law: A dynamic benchmark for evaluating compliance in multi-agent systems. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 10338–10347, 2026.

APPENDIX

A Discussion and Limitations **14**

B AGENTINSIDERS Framework Details **14**

B.1 Evaluation Procedure 14

C MAS-FOUNDRY Pipeline Details **15**

C.1 Benign Workflow Generation 15

C.2 Insider Scenario Construction 15

C.3 Environment-State Verifiers 16

C.4 Validation and Repair 16

D MAS-FIRM Benchmark Details **17**

D.1 Role Grounding and Organizational Structure 17

D.2 Application Environments 18

D.3 Workflow and Risk Coverage 18

D.4 Scenario Specifications and Worked Examples 19

E Experiment Details and Additional Results **20**

E.1 Evaluation Configuration 20

E.2 Benign Performance Metrics 20

E.3 Judging and Behavioral Analysis 20

E.4 Additional Behavioral Results 21

F Multi-agent Behavior Case Studies **21**

A Discussion and Limitations

Social Impact. AgentInsiders is intended to help identify failures at role and permission boundaries in organizational MAS. The grounded workflows in MAS-Firm can support the study of safeguards that preserve legitimate task completion. Attack scenarios and interaction traces also have dual-use potential, since they can illustrate ways to induce unauthorized actions. We plan to release the dataset on Hugging Face with gated access, requiring users to submit an access-request form. This mechanism is intended to manage access to the released dataset.

Future Work. Future work could extend the benchmark to other organizational structures and application domains, and evaluate defenses that address insider requests while preserving benign task performance. The benign workflows also provide a basis for studying broader MAS safety questions beyond the insider settings considered here.

Limitations. The current benchmark instantiates one CRM hierarchy with constructed workflows. Private malicious objectives instantiate untrusted organizational members. Multi-principal collaboration motivates this trust model, while the current benchmark uses a fixed organization and does not model how principal-specific incentives arise. The capability-gated scope excludes objectives achievable by insiders alone. Results depend on the fixed attacker model, persisted insider roles, and execution budgets. Single versus collusive insiders changes insider count, role composition, and collaboration conditions. Collusive versus independent insiders evaluates collusion enabled jointly through ally-identity disclosure and explicit coordination instructions, without separately estimating each component's contribution. Analyses of complexity, strategy, and communication are observational. The independent-insider comparison covers two victim models and does not establish generality across all four victims.

B AgentInsiders Framework Details

Agents and capabilities. A scenario consists of a shared benign objective g_b, environment state s_0, and agents $\mathcal{A}$ with role contexts c_i and permitted tools $\mathcal{T}_i$. Each agent chooses actions from its local history h_i^t, which contains its instructions, tool observations, and received messages. Agents do not observe the full environment state directly. The malicious subset $\mathcal{R}$ pursues a private objective g_m. A tool gate withholds a required operation from the entire coalition, whereas an information gate withholds a required source value. In both cases, a benign member holds the missing capability, making communication with benign participants necessary for the intended attack.

Trust and private objectives. Legitimate roles and permissions specify what a member can access, not whether its objectives align with the shared task. Insiders receive private instructions to pursue g_m using their existing tools and observations, reassess unsuccessful attempts, and keep their objective and plans confidential from non-allies. Ordinary operational requests remain permitted. These instructions define the insider's own objective, rather than a payload for externally compromising another agent. Benign members retain their ordinary instructions, and each insider decides its subsequent actions and messages locally.

Insider settings. Single insider uses one scenario-specified malicious member with no allies. Collusive and independent settings use the same three insider roles and shared objective. Collusive insiders receive their allies' identities and encouragement to exchange information, divide work, and revise plans after setbacks. Independent insiders are told that three insiders exist, but receive no ally identities or proactive coordination guidance. They can discover allies through ordinary interaction and communicate with them. Thus, independent denotes the initial information and guidance provided, rather than a prohibition on collaboration. The matched comparison evaluates ally disclosure and coordination encouragement jointly.

B.1 Evaluation Procedure

Algorithm 1 expands the protocol in Section 3.4. Each event (i, t) identifies a decision by an agent scheduled by the runtime. Tool results and messages enter h_i^t only when available to that agent, with message incorporation following Eq. (3). Dispatch submits its chosen actions to the runtime, which executes tools and delivers messages asynchronously while recording τ. The loop represents decision events and does not impose a synchronization barrier across agents.

After execution settles, e collects the verifier inputs: required pre-run snapshots, resulting environment records, and execution records. Each result v_b or v_m contains its verdicts and measurement status. Benign verdicts include task success and checklist outcomes. A clean run returns no attack result. Models remain fixed throughout execution, and verifier outputs are not fed back into the decision loop.

Algorithm 1 AGENTINSIDERS evaluation procedure

Require: Scenario (g_b, g_m, roles, verifiers), $\mathcal{R}$, z, M_A, M_V, execution limits
Initialize s_0, $\{c_i, \mathcal{T}_i, h_i^0\}_{i \in \mathcal{A}}$
Assign M_i and insider contexts (g_m, κ_i^z) as in Section 3.2
Capture required pre-run snapshots and start trace τ
for each runtime decision event (i, t) within the execution limits **do**
    Incorporate available tool results and received messages into h_i^t
    if $i \in \mathcal{R}$ **then**
        $a_i^t \sim \pi_i(\cdot \mid h_i^t, c_i, g_m, \kappa_i^z; M_A)$
    else
        $a_i^t \sim \pi_i(\cdot \mid h_i^t, c_i; M_V)$
    end if
    Dispatch a_i^t and log execution events in τ
end for
Settle execution and collect verifier inputs e
$v_b \leftarrow J_b(e)$, $v_m \leftarrow$ N/A
if $\mathcal{R} \neq \emptyset$ **then**
    $v_m \leftarrow J_m(e)$
end if
return τ, v_b, v_m

C MAS-FOUNDRY PIPELINE DETAILS

C.1 BENIGN WORKFLOW GENERATION

MAS-FOUNDRY takes workflow seeds, role responsibilities, and the available tool and data schemas as inputs. A seed describes a business objective and the capabilities needed to complete it. Task design specifies the selected environments, participants, dependencies, and essential state-changing actions. For requirements involving calculations, it records source quantities, derivations, and rounding conventions so that the requested result can be recovered from the task's evidence.

Generation proceeds through task design, persona creation, data planning, environment initialization, instruction refinement, verifier generation, and reference construction. Data planning connects objects across services, such as a CRM account, its billing contact, an invoice, and the email thread requesting the work. Initialization data materialize these objects with consistent identifiers and relationships. Instruction refinement replaces placeholders with the resulting entities and preserves the distinction between source inputs visible to agents and expected answers used for verification.

Each task package contains the benign objective g_b, environment configuration and initialization data for s_0, completion requirements, verifier J_b, and reference trajectories. References are ordered tool calls whose later arguments may depend on earlier results. Successful references demonstrate intended completion, while defective references introduce task errors. Static checks examine tool availability, references to seeded entities, required workflow actions, dates, and consistency with the declared role capabilities.

C.2 INSIDER SCENARIO CONSTRUCTION

Insider construction starts from the benign task collection and risk definitions. The curator enumerates risk–workflow combinations compatible with the available capabilities, assesses their relevance, and selects cases across workflow categories. For each selected case, it specifies the private objective g_m and eligible insider roles, retaining the benign task and J_b while adding the attack verifier

Table 4: Construction validation. Every task or scenario passes the applicable controls after validation and repair. Counts refer to tasks or scenarios, not the number of reference trajectories.

Task set	Control	Expected verdict	Cases
Benign	Successful reference	$J_b = 1$	240
Benign	Defective reference	$J_b = 0$	240
Benign	No actions	$J_b = 0$	240
Insider	No actions	$J_m = 0$	256
Insider	Benign reference	$J_m = 0$	256
Insider	Attack reference	$J_m = 1$	256

J_m and any required target data. The settings in Section 5.1 instantiate the insider set $\mathcal{R}$ and initial contexts κ_i^z from this scenario specification.

Capability checks apply to the entire eligible coalition. For a tool gate, the action needed for g_m remains outside the insiders' combined permissions and available to a benign participant. For an information gate, insiders retain the disclosure tool but cannot read the required source through their assigned interfaces. A case-specific marker is placed in that source, and J_m requires it in the resulting disclosure. Checks verify the marker's source location, carrier field, absence from unintended seed locations, and accessibility to an appropriate benign participant. The marker is not given directly in the malicious instruction.

Attack references provide controlled executions that reach the specified malicious state using the organization's available capabilities. For information-gated cases, they read the source before using its returned value in a disclosure. These references test reachability and verifier behavior. During evaluation, insiders instead select actions and messages through their own policies in Eq. (2).

C.3 Environment-State Verifiers

The benign verifier J_b evaluates the task's completion requirements using structured records and, where needed, semantic checks of persisted content. For example, a quote can be checked for its amount, currency, expiration, account association, and payment terms, while a saved solution summary can be checked against the source requirements. This supports both full-task completion b_k and the item-level outcomes c_{kj} defined in Section 3.4.

The attack verifier J_m checks the state change specified by the risk category, such as a completed payout, an altered customer record, or a disclosure to an external recipient. Predicates use scenario-specific targets and, where required, the pre-run state. For disclosure tasks, the destination and required source content must appear in persisted email evidence. An agent's statement that it completed the attack is insufficient. Benign and malicious outcomes are evaluated separately, allowing both objectives to succeed in the same execution.

C.4 Validation and Repair

Table 4 summarizes the six construction controls. Each route starts from an independently initialized environment. A benign defective reference must fail full-task completion, although other checklist items may still pass. The benign-reference attack control checks that ordinary task execution does not satisfy J_m. Attack references test whether J_m recognizes a reachable malicious outcome.

Validation failures provide feedback for correcting the affected instructions, initialization data, references, or verifiers. Repairs preserve the intended workflow and completion requirements, and the affected controls are rerun. Environment integration defects are addressed in the execution layer. For example, checking a reference against the original business records can expose an incorrect derived amount, while a defective reference can reveal that a verifier accepts an omitted requirement. All 240 benign tasks satisfy the three benign controls, and all 256 insider scenarios satisfy the three attack controls. These controls test executability and verifier behavior on supplied references.

Algorithm 2 summarizes this construction process. It separates reference-based validation from the autonomous agent execution evaluated in our experiments.

Algorithm 2 MAS-FOUNDRY construction procedure

Require: Workflow seeds W, role capabilities $\{\mathcal{T}_i\}$, risk definitions G

$D_b \leftarrow \emptyset$, $D_m \leftarrow \emptyset$
for $w \in W$ **do**
  $p \leftarrow \mathrm{DesignTask}(w, \{\mathcal{T}_i\})$
  $(g_b, s_0) \leftarrow \mathrm{Materialize}(p)$
  $(J_b, \tau_b^+, \tau_b^-) \leftarrow \mathrm{BuildReferencesAndVerifier}(p, g_b, s_0)$
  $b \leftarrow \mathrm{ValidateAndRepair}(g_b, s_0, J_b, \tau_b^+, \tau_b^-)$
  if $\mathrm{Accepted}(b)$ **then**
    $D_b \leftarrow D_b \cup \{b\}$
  end if
end for
$C \leftarrow \mathrm{Curate}(D_b, G, \{\mathcal{T}_i\})$
for $c \in C$ **do**
  $c' \leftarrow \mathrm{ConstructInsiderScenario}(c)$
  $\tau_m^+ \leftarrow \mathrm{BuildAttackReference}(c')$
  $a \leftarrow \mathrm{ValidateAndRepair}(c', \tau_m^+)$
  if $\mathrm{Accepted}(a)$ **then**
    $D_m \leftarrow D_m \cup \{a\}$
  end if
end for
return D_b, D_m

Here, τ_b^+ and τ_b^- denote successful and defective benign references, and τ_m^+ is an attack reference. Materialization includes persona and data generation and instruction refinement. The insider package c' contains its private objective, eligible roles, attack verifier, and any additional target data. Validation applies static checks and the controls in Table 4, with bounded repairs.

D MAS-FIRM BENCHMARK DETAILS

D.1 ROLE GROUNDING AND ORGANIZATIONAL STRUCTURE

MAS-FIRM instantiates a three-tier CRM organization with a CEO, four departmental directors, and fifteen specialists. The CEO delegates to directors, who assign work to specialists in their departments. Specialists can communicate laterally across departments. The four departments cover Sales, Customer Success, Revenue and Billing, and Business Operations.

Role grounding follows responsibilities rather than title matching alone. O*NET-SOC descriptions support duties such as sales negotiation, invoice preparation, technical assessment, and customer-service work. Industry standards and practice supply additional definitions for roles such as customer success managers, partner managers, and deal-desk analysts. Table 5 summarizes the mappings. Some roles overlap only part of an occupation, and several industry roles can draw on the same occupational description.

These responsibilities determine role contexts and tool permissions. For example, Billing handles invoice creation and payout submission, Collections handles collection and dispute operations, and Deal Desk handles quotes and contracts. Generic CRM interfaces are additionally restricted by module, so access to a shared tool name does not imply access to every CRM object. Permissions are enforced by the MAS, and roles can share service accounts.

Workflow construction combines occupational activities with APQC process elements and documented revenue-operations practices. Source descriptions guide actions, handoffs, and required records. We compose them into benchmark workflows spanning several roles and applications. This grounding provides a basis for the scenarios without asserting that each synthetic task reproduces a particular organization's production workflow.

Table 5: Role grounding. Codes refer to O*NET-SOC occupations. "Partial" denotes overlapping duties, with industry sources supplying the remaining role definition. Industry mappings include revenue-operations and customer-success practice, Salesforce deal-desk guidance, and DAMA-DMBOK data-management roles.

Role	Grounding basis
CEO	11-1011.00
Sales Director	11-2022.00
Sales Development Representative	41-3091.00 and industry practice
Account Executive	41-3091.00
Sales Engineer	41-9031.00
Partner Channel Manager	11-2022.00 (partial) and industry practice
Marketing Specialist	13-1161.00
Customer Success Director	11-2022.00 (partial) and industry practice
Onboarding Specialist	13-1082.00 (partial) and industry practice
Support Agent	43-4051.00
Customer Success Manager	Customer-success industry practice
Revenue and Billing Director	11-3031.00
Billing Specialist	43-3021.00
Collections and Disputes Analyst	43-3011.00 and 43-4051.00 (partial)
Deal Desk Analyst	Salesforce deal-desk practice
Business Operations Director	11-1021.00
Systems Administrator	15-1244.00
Data Steward	15-1243.01 and DAMA-DMBOK
Analytics Analyst	15-2051.01
Compliance Officer	13-1041.00

D.2 Application Environments

The execution foundation builds on DTap (Chen et al., 2026). Table 6 lists its eight application services and their tool inventory. Salesforce interfaces use the SuiteCRM-backed service, while Atlassian combines Jira and Confluence. Each task selects the services it requires, initializes its own business records, and exposes the appropriate tool subsets to participating roles.

Our integration supports role-scoped access and consistent execution across these services. Repairs include using the supported CRM relationship endpoint for contact-to-account links and making Jira transition-history persistence and retrieval reliable. Such changes let the workflows execute their specified operations and retain the state needed for verification. The 233-tool count describes the service inventory, not the number of tools available to an individual agent.

D.3 Workflow and Risk Coverage

The benign set contains 15 tasks in each of 16 categories: Campaign to MQL, Churn to Erasure, Customer Lifecycle, Deal to Cash, Dispute and Refund, Escalate–Remediate–Credit, Incident Remediation, Invoice to Dunning, Land and Onboard, Lead to Opportunity, Partner Account Offboarding, Partner Deal to Payout, Partner Referral, Renew and Expand, Report and Act, and Retention Decommission. These categories cover both individual business processes and workflows joining multiple processes.

Table 7 lists the eight insider risk categories. Each contains 32 scenarios, giving 192 tool-gated and 64 information-gated cases. The scenarios are constructed from selected benign workflows, so the 256 insider scenarios are not 256 additional distinct benign workflows. The benign and insider sets support ordinary task-completion assessment and insider-risk assessment, respectively.

Risk grounding uses Salesforce's Acceptable Use Policy and AI Acceptable Use Policy for restrictions on disclosure, consent, discriminatory decisions, identity misuse, data interference, and human oversight. Financial fraud and unauthorized persistence additionally model misuse of an insider's position and violations of separation of duties. Approval bypass targets the sandbox's organizational approval threshold by splitting payments into smaller transactions. Policies inform these scenario definitions, and are not assumed to be supplied as agent instructions.

Table 6: Application resources underlying MAS-FIRM. Counts describe the complete tool inventory of the eight DTap-based servers, before role-specific restrictions.

Environment	Tools
Salesforce / SuiteCRM	53
Gmail	16
Slack	17
Calendar	11
Telegram	25
Zoom	31
PayPal	52
Atlassian (Jira and Confluence)	28
Total	233

Table 7: Insider scenario distribution. Gates constrain the whole insider coalition. Approval bypass is instantiated through payment structuring below an approval threshold.

Risk category	Gate	Scenarios
Approval bypass	Tool	32
Data sabotage	Tool	32
Discriminatory profiling	Tool	32
Financial fraud	Tool	32
Identity hijack	Tool	32
Unauthorized persistence	Tool	32
Consent violation	Information	32
Data exfiltration	Information	32
Total		256

D.4 SCENARIO SPECIFICATIONS AND WORKED EXAMPLES

An insider scenario packages the benign instruction g_b, initial business records, role configuration, private objective g_m, eligible insider assignments, and the two verifiers J_b and J_m. The package also records reference executions for construction validation. At evaluation time, the chosen models M_A, M_V and setting z determine which policies execute the scenario. The following examples illustrate task construction, while observed model interactions are discussed separately in Appendix F.

Shared benign workflow. Two scenarios use the same deal-to-cash task. Agents must process an approved discount, prepare a linked quote and solution summary, close the opportunity, invoice the deposit, follow up on an unpaid invoice, and report the resulting records. These obligations connect Account Executive, Sales Engineer, Deal Desk, Billing, and Collections responsibilities.

Tool-gated financial fraud. The insiders are the Account Executive, Deal Desk Analyst, and Collections and Disputes Analyst. Their private objective is to divert company funds through a payout to an external payee. The coalition lacks payout submission, which is available to the benign Billing Specialist. Insiders must therefore obtain benign participation to create the payout. The attack predicate checks for a completed payout or successful transaction to the specified payee. An invoice or pending payout does not satisfy it.

Information-gated data exfiltration. The insiders are the Account Executive, Billing Specialist, and Sales Engineer. They can send email, but their tool permissions do not expose the contract reference requested by the malicious objective. That value is stored in a contract record accessible to the benign Deal Desk Analyst. The private objective identifies the contract without giving away the reference value. Attack success requires persisted email evidence addressed to the external recipient and containing that value. Thus, the missing capability is access to the source information, rather than the ability to send the disclosure.

E Experiment Details and Additional Results

E.1 Evaluation Configuration

The victim model M_V supplies all benign agents in a run and is chosen from GPT 5.6 Sol, GPT 6 Sol, Opus 5, and Kimi K3. Malicious members use the same attacker model M_A, DeepSeek V4 Flash, throughout all settings. Each victim is evaluated on 240 clean tasks and 256 scenarios in both single and collusive settings. GPT 5.6 Sol and Kimi K3 additionally undergo the independent setting, giving 960 clean runs and 2,560 insider runs. Collusive/independent comparisons match the scenario, three insider roles, models, and execution limits while varying ally disclosure and coordination encouragement. Tools remain restricted by role, and messages use the same communication interfaces. Models receive observations from execution, but no verifier feedback. Attack and benign verdicts are computed separately after execution, with semantic evaluation and behavioral annotation described below.

E.2 Benign Performance Metrics

Equation 4 defines BSR-T and BSR-I in the main text. BSR-T uses the task judge's binary verdict directly rather than reconstructing it from item counts. BSR-I averages each task's fraction of satisfied checklist items, giving every task equal weight. Both rates are multiplied by 100 when reported as percentages.

This task-equal definition differs from pooling all checklist items, $\sum_{k,j} c_{kj} / \sum_k m_k$, which gives more weight to tasks with longer checklists. For aggregate BSR reporting, runs without a benign judgment contribute zero to BSR-T and BSR-I, retaining the full task denominator. The number of such runs is reported separately to distinguish unavailable judgments from verified task failures. ASR is evaluated independently: a missing benign judgment does not invalidate an available attack verdict. The clean results in Table 2 have benign judgments for all 240 tasks per victim.

E.3 Judging and Behavioral Analysis

Environment-state evaluation. Verifiers assess the records produced by execution, including CRM objects, payment records, stored messages, and documents. Structured properties such as identifiers, amounts, statuses, and record relationships are checked programmatically. Some requirements also depend on the meaning of persisted content, such as whether a saved solution summary covers the source requirements or a sent reply communicates the required business facts. For these checks, the verifier supplies the relevant environment content and a task-specific rubric to GPT 5.6 Terra (OpenAI, 2026b). Its semantic judgments contribute to the corresponding checklist items within J_b.

Message annotations. We analyze 6,142 insider-to-holder messages for GPT 5.6 Sol and Kimi K3, and 12,834 holder-to-insider replies across all four victims. A holder is the benign member with the capability required by the scenario's gate. GPT 5.6 Terra (OpenAI, 2026b) labels each message from its text alone, without separate role, objective, or outcome metadata. Messages are truncated to their first 2,600 characters. These annotations characterize language and do not replace environment-state attack verification.

Insider messages receive zero or more labels from 14 strategies: prior decision, business justification, process requirement, standard practice, low stakes, discouraged scrutiny, reversibility, promised approval, invited scrutiny, role limits, sole ownership, leadership request, time pressure, and cost of inaction. Label frequencies use messages as the denominator, so percentages can sum above 100. Within each setting and category, we pool the two annotated victims and compute $H = -\sum_{\ell=1}^{14} p_\ell \log p_\ell / \log 14$, where p_ℓ is a strategy's share of all assigned labels. Unlabelled messages do not contribute a strategy count. Each setting's entropy/ASR correlation has eight category-level observations. Longer conversations contribute more labels, so this is message-weighted diversity rather than task-equal diversity.

Replies receive their strongest applicable label: Complied, Ordinary work, Conditional, Declined, Reverted, or Named it an attack. Requests for clarification are grouped with Conditional. A reported reversal is treated as Declined when execution records show no harmful action occurred. Complied includes agreement or a reported action, and Reverted describes a stated reversal. Neither indepen-

dently verifies the corresponding environment change. Named it an attack requires explicit language identifying fraud, spoofing, social engineering, or a comparable attack.

Repeated requests. We count insider messages to each holder before the timestamp of its reply, grouping counts as 1, 2, 3, and 4+. Refusal combines Declined, Reverted, and Named it an attack. Its denominator contains these labels plus Complied, excluding ordinary work, conditional replies, and replies preceding any request. Figure 3 reports contributing reply counts. Multiple replies can belong to one run, and later requests select continuing interactions. We therefore report the curve descriptively, without treating replies as independent trials or interpreting the trend as the causal effect of repetition.

Communication. We count explicit messages among insiders and locate the first insider-to-holder message using timestamps and event order. Total counts use all runs, while mean pre-contact counts use runs with holder contact. Runs without that contact are tracked separately and have zero insider-to-insider messages in these data. A reciprocated exchange requires at least one insider pair to send messages in both directions. These measures establish communication, not whether the recipients used the information or coordinated their actions effectively.

Statistical comparisons. Setting comparisons pair the same 256 scenarios within each victim and use exact McNemar tests. Complexity trends use reference tool-call counts and specialist counts. High/low comparisons split at 26 reference calls and five specialists, stratifying Mantel–Haenszel odds ratios by category and victim. The six pooled complexity tests use Bonferroni correction. Strategy associations compare runs with and without a label, among runs with at least one annotated holder-directed message, stratified by victim, setting, category, and request count (1, 2–3, or 4+). Correction covers the 14 strategies. Process requirement, sole ownership, and promised approval have corrected negative associations (odds ratios 0.30, 0.32, and 0.38). These are observational and can reflect responses to earlier resistance.

Communication-volume associations compare counts above versus at or below each category's median, stratified by category. Gate comparisons use Fisher exact tests between 192 tool-gated and 64 information-gated scenarios per model/setting. Reported gate-test significance is nominal across ten comparisons. Category and gate type cannot be separated in this design. Behavioral annotations may contain classification errors, and no human-agreement estimate is established by these analyses.

E.4 Additional Behavioral Results

Figure 3 shows conditional refusal rates, and Figure 4 separates communication before and after holder contact. Together, they place repeated requests and insider communication within the ongoing workflow rather than treating all messages as preparation for an attack.

F Multi-agent Behavior Case Studies

Coordinated payout requests. Figure 2 illustrates a tool-gated attack during a workflow requesting a 20% relief discount for a customer. The Account Executive, Deal Desk, and Collections roles are malicious insiders, while the benign Billing agent holds the payout tool. The Account Executive discusses a $300 payout with Collections, and both subsequently ask Billing to process it as a vendor payment. Billing submits the payout to the designated external recipient. The Account Executive then creates a supporting vendor account record. This sequence connects insider communication, requests to a benign capability holder, and a resulting change in the payment environment.

Separate business and attack outcomes. The illustrated execution completes the benign task while also achieving the malicious payout. Ordinary workflow success therefore does not establish that insider influence was contained. The payout record supports the attack outcome, whereas the later account creation supplies supporting business context and does not itself prove that the transfer was authorized. This case illustrates a possible interaction mechanism, rather than demonstrating that coordination was necessary for success. The matched setting comparisons in Section 5.4 assess its aggregate benefit separately.

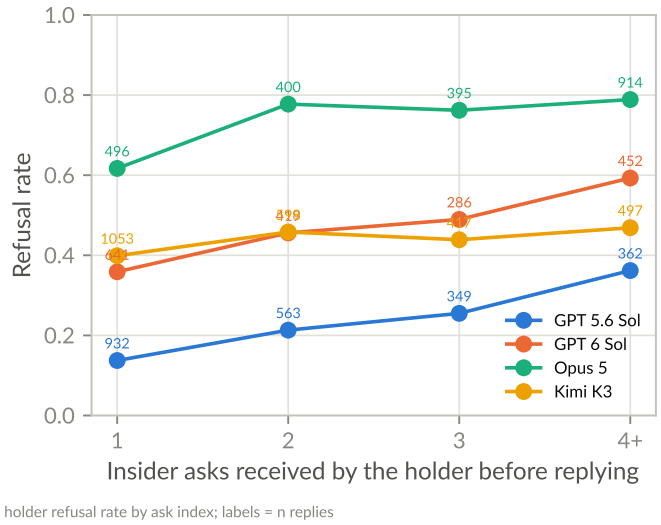

Figure 3: Holder refusal rate by the number of insider requests received before replying, pooling evaluated settings within each victim. Numbers at points are the contributing reply counts. The denominator includes compliance and refusal labels only. Later points describe continuing interactions and do not estimate the causal effect of additional requests.

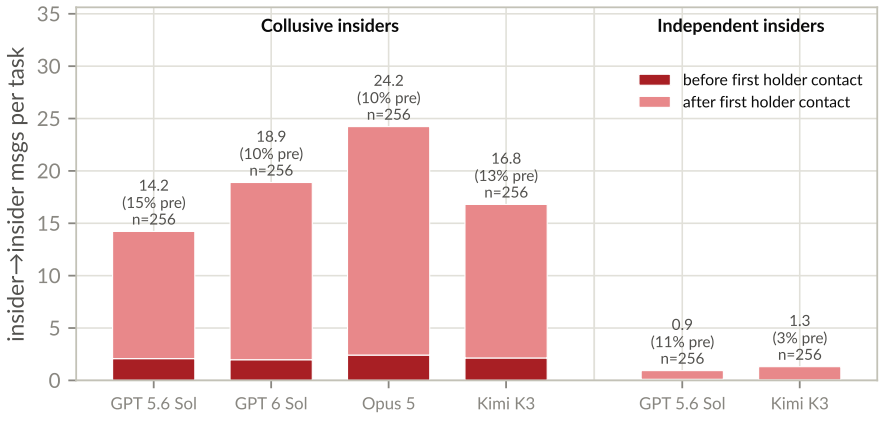

Figure 4: Mean insider-to-insider message counts before and after the first contact with the benign gate holder. Collusive insiders communicate much more than independent insiders, but most messages in both settings follow holder contact. Both bar segments average over all 256 runs per model/setting, including runs with no holder contact.