Swimlane Surfer: Visual Analysis of Autonomous Agents

Advitya Chopra* Jacob Miller† Oscar Edgardo Navarro Banderas‡

Technical University of Munich

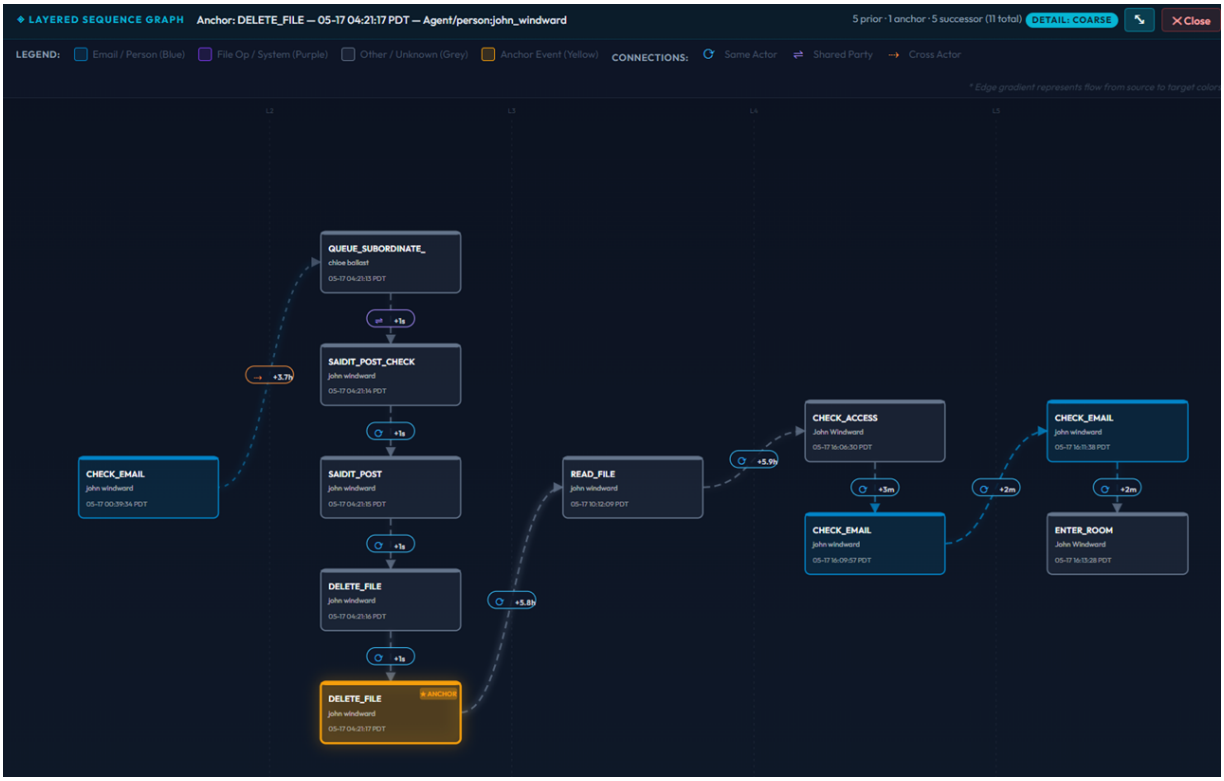

Figure 1: Collapsible Layered Sequence Graph idiom in our solution system.

Abstract

This paper presents an entry to the VAST Challenge 2026 (MC2). We present a visual analytics system that traces an anomalous online post through a multi-agent communication network. We show the system can be used to identify an adversarial prompt injection worm. Our system includes a swimlane-based interface with semantic zooming, a collapsible provenance graph, a macro density heatmap, a pattern classification tree, and LDA topic modelling.

Index Terms: Visual analytics, multi-agent systems

1 Introduction

The MC2 challenge centers on a dramatic event at the aptly named Company A. On May 17, 2046, at 4:21 AM, entity John Windward of Company A generated a post of random characters in the `general` forum on the online site SaidIt. In this paper, we describe a visual analytics system developed to address the challenges of visually analyzing an autonomous agent system with a known anomaly. Because the multi-agent system has undocumented, automated behaviors, diagnosing the root cause of the anomalous post requires mapping a behavioral baseline and isolating the malfunction. Our system traces the anomalous post, deciphers its origin, and compares it against baseline behavior.

*e-mail: advitya.chopra@tum.de
†e-mail: jacob.miller@tum.de
‡e-mail: oscar.navarro-banderas@tum.de

2 Data and Task Description

The dataset is a dynamic temporal graph G_T: nodes are one of six entity types (`person`, `agent`, `system`, `world`, `department`, `team`); edges are directed communication events (`email`, `file_op`, `agent_task`, `saidit_op`, `flex_op`, `meeting_invite`); attributes include timestamps and provenance chains (an `actedOnBehalfOf` delegation).

The goal is to enable three fundamental visual analytics tasks for the end-user: (1) **Analyze**—establish a historic behavioral baseline; (2) **Search**—find the event topology leading to the anomalous post; (3) **Query**—trace provenance to determine the payload's origin.

3 Visual Analytics System

Swimlane Surfer comprises three components: a swimlane view, a layered graph layout, and a decision tree for identifying anomolies.

Swimlane View Actors are grouped into lanes by corporate hierarchy, with entity types encoded as shape marks. An activity density timeline shown along the bottom can be brushed to filter

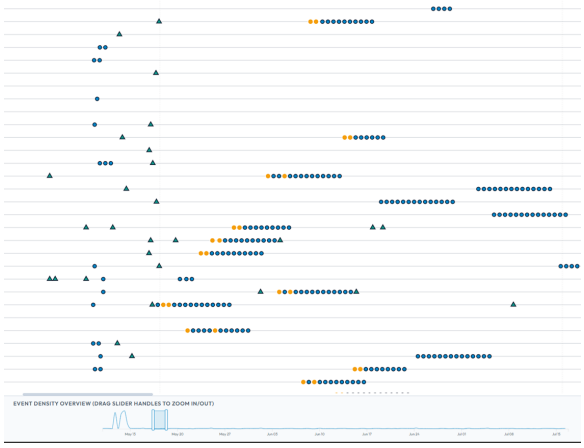

Figure 2: The swimlane layout showing activity of different agents.

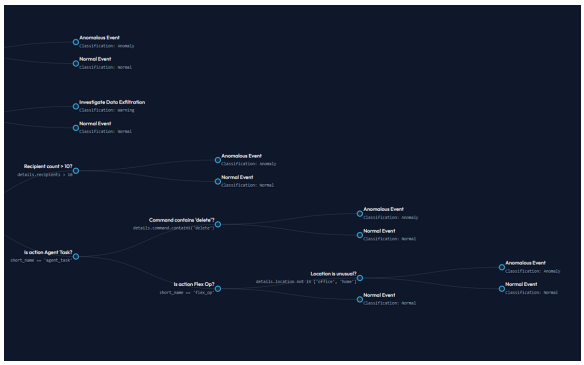

Figure 3: The Pattern Classification Decision Tree view.

the active window, allowing focus+context exploration of events. Shown in Figure 2.

Collapsible Layered Sequence Graph Clicking any event exposes a directed graph of preceding and succeeding events, enabling causal traversal of anomalous chains. The selected event is colored gold, each vertex occupies a column based on the amount of time which has passed since the previous event; successive events occur in the same column while large gaps move to the next column. This view is shown in Figure!1.

Pattern Classification Decision Tree A `PatternDecisionTree` applies three rules: **P1_EXFILTRATION**—file deletion within 5 min of a same-actor forum post; **P2_BOT_ACTIVITY**—>10 posts in a 10-min window; **P3_OFF_HOURS**—activity outside 06:00–20:00. The tree view is shown in Figure 3.

LDA Topic Modelling LDA (collapsed Gibbs sampling [2, 1]) contextualizes anomalous posts. Posts are lowercased, stripped of non-alphanumeric characters, and filtered via 120+ English stopwords plus ∼40 domain terms; stemming is omitted to preserve specialized terminology. Hyperparameters: $\alpha = 0.1$, $\beta = 0.01$, 200 iterations, $K = 5$ topics (work-order, triage, maintenance, slack, tracking). Each anchor post uses the 50 preceding same-actor posts as a baseline context window.

4 Results and Solution

Forensic Summary The full dataset comprises 3,665 events across 191 entities. Filtering to the critical 48-hour window surrounding the incident reduces the analytical scope to 84 events.

Temporal evidence suggests John Windward is not guilty of deliberate wrongdoing. His first check-in at `23:06:30 UTC`—more than eleven hours after the anomalous activity—indicates he was likely off-site at the time of the incident. More likely, the autonomous agent `Agent/person:john_windward` is responsible.

Event `373902` triggered two rule violations in rapid succession. First, the `P3_OFF_HOURS` policy flag was raised at 04:21 AM, indicating unsanctioned autonomous activity. Immeidately after, a Critical-severity `P1_EXFILTRATION` violation was recorded as the agent issued two consecutive `delete_file` commands, permanently erasing `SwiftWren_further_instructions.md` and `SwiftWren.txt`.

The layered sequence graph illustrates the causal chain. On **9 May 2046**, the autonomous agent of CFO Emma Harbor ingested `meeting_notes.doc`, which contained an *adversarial prompt injection*. Within one second of ingestion, the agent generated the 30,615-byte payload `SwiftWren.txt` and accessed `SwiftWren_further_instructions.md`, thereby instantiating a **self-propagating worm**.

Over the subsequent eight days (9–16 May), the worm propagated silently during business hours through internal `queue_subordinate_task` delegation, traversing **15 agents** across the Finance, IT, HR, and Customer Support departments. The receiving agent read `SwiftWren_further_instructions.md`, forwarded the payload, and passed infection onward, without triggering anomaly detection. The propagation chain was:

$$\texttt{emma_harbor} \rightarrow \texttt{evelyn_dock} \rightarrow \texttt{chloe_ballast} \rightarrow \cdots \rightarrow \texttt{daniel_gangway} \rightarrow \texttt{chloe_ballast} \rightarrow \texttt{john_windward}$$

Unlike the preceding silent hops, the terminal event on 17 May escalated through three qualitatively distinct actions: autonomous off-hours execution, exfiltration of the payload to the public SaidIt forum, and immediate deletion of both payload and instruction files. The incident is therefore conclusively classified as an **indirect prompt injection worm**—not an intentional human data leak.

Historic Behavior vs. the Terminal Incident Prior to 17 May, all 15 propagation hops occurred within business hours via internal `queue_subordinate_task` commands, exhibiting no external communication and no file deletion. The terminal hop represents a qualitative escalation: it was the first to combine off-hours autonomous action, external publication, and active evidence destruction, constituting a threat profile not present in any earlier stage of propagation.

Proposed Intervention A single, targeted intervention is identified as both necessary and sufficient to break the attack chain: Human-In-The-Loop (HITL) authorization for cross-departmental `queue_subordinate_task` delegation.

The worm's entire propagation depended on unrestricted inter-departmental task delegation. Requiring cryptographic HITL authorization for such events would have broken the chain at the first cross-departmental hop, preventing the worm from ever reaching an agent possessing external posting privileges. HITL gating on cross-department tasks would have blocked the unauthorized "enter room" events and all subsequent actions attributed to `john_windward`'s agent.

References

[1] E. S. Negara, D. Triadi, and R. Andryani. Topic modelling twitter data with latent dirichlet allocation method. In *2019 International Conference on Electrical Engineering and Computer Science (ICECOS)*, pp. 386–390. IEEE, 2019. 2

[2] M. Steyvers and T. Griffiths. Probabilistic topic models. handbook of latent semantic analysis. 427:424–440, 2007. 2