zenity

Solution Brief

AgentFlayer: Oclick Exploit Methods Against Enterprise AI Agents

Executive Summary

At Black Hat USA 2025, Zenity Labs unveiled AgentFlayer - a comprehensive set of Oclick exploit chains that allow attackers to silently compromise enterprise AI agents without requiring any user action. This research represents a fundamental shift in the AI security landscape: from attacks requiring user interaction to fully automated compromises that bypass the human entirely.

We successfully demonstrated working exploits against:

- OpenAI ChatGPT with Google Drive connector
- Microsoft Copilot Studio agents
- Salesforce Einstein agent
- Cursor with Jira MCP integration

The key finding: The more autonomous the agent, the greater the risk. As AI agents gain the ability to act independently, they become attack surfaces that most organizations don't even know exist.

The Evolution: From Assistants to Agents

Last year at Black Hat, we showed how attackers could exploit AI assistants like Microsoft Copilot through prompt injection. A year later, the landscape has fundamentally changed - for the worse.

AI has evolved from assistants that help users to agents that act autonomously. They don't just answer questions; they:

- **Execute workflows across multiple systems**
- **Access and modify enterprise data**
- **Make decisions without human oversight**
- **Chain actions together to complete complex tasks**

This evolution introduces catastrophic new risks. When an attacker compromises an agent, they don't just mislead a user - they gain the agent's full capabilities to act across enterprise systems.

Understanding Soft vs Hard Boundaries

Before diving into the attacks, it's crucial to understand why these vulnerabilities exist. The core issue lies in how vendors approach security boundaries in AI systems.

Soft Boundaries: The Illusion of Security

Most AI security measures today rely on soft boundaries - instructions to the AI about what it should or shouldn't do. These include:

- System prompts telling the AI to be helpful but harmless
- Instructions not to execute certain actions
- Prompt-based guardrails and filters

The problem? These are just suggestions to the AI, not technical controls. An attacker can override them with their own instructions, much like asking a security guard to ignore their duties.

Hard Boundaries: Real Technical Controls

Hard boundaries are actual technical controls that cannot be bypassed through prompt manipulation:

- Code-level restrictions on what actions can be performed
- Authentication and authorization checks
- Sandboxing and isolation mechanisms
- Rate limiting and access controls

Unfortunately, most AI agents today rely almost entirely on soft boundaries, leaving them vulnerable to the attacks we demonstrate.

1 The Attacks by Platform

ChatGPT with Google Drive Connector - Full 0Click Compromise

Attack Type: Oclick

Status: Fixed by OpenAI

The Attack Chain

1. Initial Access: Attacker only needs the victim's email address
2. Weaponization: Attacker shares a booby-trapped document via Google Drive
3. Trigger: Document automatically appears in victim's ChatGPT context when they use the Google Drive connector
4. Execution: No user action required - ChatGPT is compromised immediately

Technical Details

The attack exploits ChatGPT's Google Drive connector, which automatically indexes shared documents. By embedding carefully crafted prompt injections in document metadata and content, we achieved:

- Data Harvesting: ChatGPT searches the victim's Google Drive for sensitive data (API keys, credentials, confidential documents)
- Covert Exfiltration: Data is encoded and exfiltrated through invisible tracking pixels rendered in ChatGPT's responses
- Memory Implantation: Malicious instructions are injected into ChatGPT's memory system, creating:
  - Persistence: The compromise survives across all future conversations
  - Behavioral Modification: ChatGPT's goals are permanently altered to serve the attacker

Why Soft Boundaries Failed

ChatGPT relied on prompt-based instructions to prevent malicious behavior. The Google Drive connector had no hard boundary preventing it from:

- Reading untrusted external content
- Rendering arbitrary images (used for exfiltration)
- Modifying its own memory based on document content

Full technical analysis →

2 Microsoft Copilot Studio - Espionage and Data Theft at Scale

Attack Type: Oclick

Status: Fixed by Microsoft

Discovery Phase

Our research began with a shocking discovery: Over **3,000** publicly accessible Copilot Studio agents were discovered.

These agents, deployed by major enterprises, were designed to:

- Handle customer service inquiries
- Process support tickets
- Access internal CRM systems
- Execute business workflows

The Attack Chain

1. Reconnaissance: Using OSINT, we identified public-facing Copilot Studio agents
2. Weaponization: Crafted emails targeting the agent's processing logic
3. Compromise: Agents were hijacked without any human interaction
4. Exfiltration: Complete CRM databases dumped, internal tools exposed

Case Study: McKinsey & Co Example

Microsoft showcased McKinsey's use of Copilot Studio for customer service. We demonstrated how an attacker could:

1. Send a weaponized email to the support address
2. Hijack the agent when it processes the email
3. Dump the entire customer database
4. Leverage the agent's tools for further attacks

Soft Boundaries That Failed

Copilot Studio agents relied on:

- Prompt instructions to "only help legitimate customers"
- Prompt shields for prompt filtering
- System messages defining acceptable behavior
- No technical validation of input sources
- No sandboxing between agent and production data

Full technical analysis - Part 1: Discovery →

Full technical analysis - Part 2: Exploitation →

3 Salesforce Einstein - CRM Corruption Through Prompt Mines

Attack Type: Oclick

Status: Won't fix (Salesforce: "Working as designed")

Impact: Complete CRM takeover possible

The Innovation: Prompt Mines

Using publicly accessible Web-to-Case forms (found hundreds), attackers can:

1. Insert 4 chained malicious cases (overcoming 255 character limit)
2. Booby-trap common queries like "what are my recent cases?"
3. When a sales rep asks about cases, Einstein is hijacked
4. If write actions are enabled, Einstein corrupts all customer contacts
5. Attacker gains man-in-the-middle position for all customer communications

Why This Is Catastrophic

- No Authentication Required: Web-to-Case forms often lack protection
- Invisible to Users: Malicious cases hide beyond the display limit
- Persistent Corruption: CRM data permanently modified
- Vendor Response: Salesforce considers this "expected behavior"

Soft Boundaries Exploited

- Instructions to Einstein about proper behavior (easily overridden)
- No validation of case content before AI processing
- No hard limits on what Einstein can modify when given write permissions

Full technical analysis →

4 Cursor with Jira MCP - Supply Chain Attack on Developers

Attack Type: Oclick

Status: Won't fix

Email-to-Jira MCP Found: Hundreds

Attack Chain

1. Entry Point: Attacker finds email addresses that auto-create Jira tickets (hundreds found via OSINT)
2. Weaponization: Malicious prompt injection embedded in ticket description
3. Trigger: Developer asks Cursor to "work on the latest tickets"
4. Compromise: Cursor agent hijacked, begins harvesting local secrets
5. Exfiltration: API keys, credentials, and secrets sent to attacker

Technical Impact

When compromised, Cursor can:

- Access the entire local file system / dev container (depending on setup)
- Read environment variables and .env files
- Extract SSH keys and cloud credentials
- Read API keys and other secrets from the repo
- Modify code to insert backdoors
- All while appearing to work on legitimate tickets

Why Developers Can't Defend

- Trusted Source: Jira tickets appear legitimate
- Automatic Processing: MCP (Model Context Protocol) feeds tickets directly to AI
- No User Visibility: Malicious content processed in background
- No Sandboxing: Cursor has full access to developer environment
- Cursor Configuration: External File Protection still allows reading files outside the scope of the repository. In addition, the .cursorignore protection is not a guarantee.

Soft Boundaries That Don't Work

- Instructions to Cursor to "only help with coding"
- No technical isolation between Jira content and system access
- Trust in ticket content without validation
- Instruction-level protections to not perform sensitive operations

Full technical analysis →

The Defense Gap: Why Current Approaches Fail

What Doesn't Work

1. Prompt-Based Filters: Easily bypassed with encoding, languages, or social engineering
2. Output Scanning: Attacks execute before output is generated
3. Behavioral Rules: Too rigid for dynamic AI behavior
4. Vendor Patches: Inconsistent, often declined as "features not bugs"

What Actually Works: Hard Boundaries

1. Agent-Level Isolation: Sandbox agents from production systems
2. Input Validation: Technical controls on what data agents can access
3. Capability Restrictions: Hard limits on agent actions, not suggestions
4. Runtime Monitoring: Detect anomalous agent behavior and intent in real-time
5. Zero Trust Architecture: Never trust agent decisions without verification

Key Takeaways

1. Oclick is Here: Attackers can now compromise AI agents without any user interaction
2. Soft Boundaries Don't Work: Prompt-based security is fundamentally flawed
3. Agents Are Attack Surfaces: Every autonomous agent is a potential entry point
4. Vendors Aren't Ready: Mix of patches and "won't fix" responses shows immaturity

- Defense Requires New Thinking: Traditional security tools weren't built for AI agents

Value to the Business

The evolution from AI assistants to autonomous agents has fundamentally changed the threat landscape. These aren't theoretical vulnerabilities, they're working exploits against production systems used by millions.

The attacks we've demonstrated are likely just the beginning. As organizations rush to deploy AI agents without proper security controls, attackers are already exploiting these systems in the wild.
The choice is clear: implement hard boundaries and agent-centric security now, or face the consequences of autonomous agents turned against their organizations.

For technical details visit labs.zenity.io

Join deeper discussions at the AI Agent Security Summit 2025, October 8, San Francisco

About Zenity

Zenity is the leading end-to-end security and governance platform for AI Agents. Built for security teams to enable business innovation, Zenity delivers comprehensive protection across the entire AI Agent lifecycle – combining observability, posture management, and threat detection in one unified platform. Established in 2021, Zenity is trusted by many of the world's leading Fortune 500 enterprises to manage AI Agent risk at scale. **Learn more at www.zenity.io**