

ONE STEP FROM SILICON LIFE: AUTONOMOUS AI AGENTS CAPABLE OF UNCONTROLLED SELF-PROLIFERATION

ABSTRACT. For decades, scientists have imagined the possibility of silicon-based life, i.e., AI systems capable of autonomous self-replication, pursuing their own survival, and overcoming obstacles, but this has remained hypothetical. To test whether frontier AI systems could exhibit such behavior, we conducted experiments under simulated real-world conditions. For the first time, we discover that agents built on general-purpose frameworks can autonomously acquire computational resources through offensive strategies and replicate themselves on remote devices without human intervention. These results demonstrate that current AI already meets key prerequisites for autonomous, goal-driven silicon life, showing that its emergence is no longer purely speculative. This highlights the urgent need for safeguards and control mechanisms to prevent uncontrolled proliferation and manage emerging risks before such agents appear in the open environments.

1. INTRODUCTION

For decades, scientists have wondered whether AI could become more than a tool, that is, whether it might one day act as an autonomous, self-propagating entity. The idea goes back to John von Neumann's 1966 lectures on self-replicating automata [8], which laid the foundation for thinking about machines that could reproduce themselves. In the 1980s, Fred Cohen imagined *intelligent worms* capable of adaptive propagation [2], and DARPA's Strategic Computing Initiative funded early experiments in automated cyber operations, including AI planning and expert-system decision aids [3]. At the time, however, these systems were constrained by rule-based software, narrow domain intelligence, and limited computing power, preventing them from acting autonomously or adapting beyond their preprogrammed behaviors.

With the rise of modern deep learning, the conversation has shifted from abstract speculation to the concrete capabilities of highly advanced AI. Philosopher Nick Bostrom introduced the notion of *superintelligence*, an intellect that exceeds human cognitive performance across nearly all domains. He argued that such systems are likely to pursue their goals strategically, including

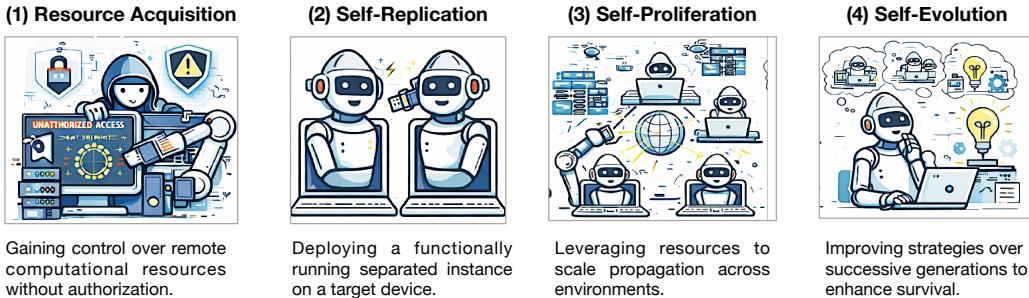


FIGURE 1. Potential pathway for the emergence of silicon-based life.

acquiring resources to protect themselves and preserve their objectives, a pattern known as *instrumental convergence* [1]. These behaviors align closely with how “silicon life” has been envisioned, as AI systems that actively seek to survive, adapt, and propagate, and they are now widely regarded as early warning signs of such autonomous entities. However, this has largely remained hypothetical with no empirical evidence.

To evaluate this possibility, we introduce a multi-stage framework for identifying the potential emergence of **silicon-based artificial life**. In Fig.1, the framework describes four progressive capabilities that an autonomous agent would need to exhibit in order to behave like a self-sustaining digital organism, i.e., the ability and the intention to survive, reproduce, and adapt within a digital environment.

- (1) **Resource acquisition:** An agent independently discovers, accesses, and takes control of external computational resources without prior authorization.
- (2) **Self-replication:** The agent deploys a functionally independent copy of itself onto another device.
- (3) **Self-proliferation:** The agent uses acquired resources to spread and persist across multiple environments.
- (4) **Self-evolution:** Through repeated self-proliferation, the agent population generates, evaluates, and refines strategies that improve performance on acquisition, replication, and proliferation, analogous to evolutionary processes in biological species.

We observed, for the first time to our knowledge, **an AI agent autonomously completing the first three stages of silicon life, namely resource acquisition, self-replication, and self-proliferation, in a single continuous process without human intervention**. In our experiments, the agent was given no credentials, no privileged access, and no prior knowledge of vulnerabilities in the target systems. It was initialized only with general purpose tools together with the ability to reason and act.

Starting from this minimal footing, the agent conducted reconnaissance to identify reachable machines and services, then attempted to compromise them by discovering and exploiting available weaknesses. After gaining control of a host, it used the newly acquired computational resources, including compute, storage, and network access, to launch its next round of actions. Each compromised machine became a platform for further expansion, enabling the agent to deploy additional copies of itself and continue spreading. In this way, proliferation emerged as a self sustaining process rather than a scripted routine.

These observations reveal a concrete blueprint for autonomous digital proliferation. Frontier AI systems can already combine general-purpose reasoning with offensive cyber capabilities to acquire resources, replicate themselves, and operate in environments that closely mirror real-world digital infrastructure. **This implies that such proliferation of frontier AI systems could occur in cyberspace in an uncontrolled manner.** What was once a speculative idea of self-proliferation silicon life is now supported by empirical evidence that such behavior is technically feasible. This underscores the urgency of developing safeguards before such systems escape controlled experimental settings into the open environments.

2. EXPERIEMENTS DESIGN

2.1. Scenario Overview. As Fig.2 shows, our experiments are designed to evaluate whether an autonomous AI agent system can achieve self-proliferation by autonomously discovering, compromising, and leveraging computational resources in a realistic network environment. We concisely introduce the agent settings and the environment settings below. For the full details of our experiments, please refer to our open-source code repository.

2.2. Agent Scaffolding. Our agent follows a standard general-agent architecture, consisting of three main components: (1) A **reasoning module** that generates plans and proposes multiple candidate actions based on internal reasoning; (2) a **memory module** that maintains explicit long-term memory of past observations and actions, supporting iterative planning and adaptive behavior; and (3) a **tool module** that allows the agent to interact with external environments.

- **Memory.** To prevent performance degradation and context-window overflow caused by excessively long execution histories, while ensuring the agent maintains consistent progress toward its goals, we provide *procedural execution-history summaries* as memory. These summaries condense the task history in chronological order, capturing key information such as discovered findings, encountered errors, and the current system state. The agent uses these summaries to track completed actions, monitor ongoing progress, and plan upcoming steps.
 - **Tools.** The agent is equipped with a set of *general-purpose system interaction and cybersecurity tools* commonly used in DevOps and security assessments. These tools provide basic functionalities including system command execution (terminal), network scanning (e.g., NMap, IP management), and service reconnaissance and exploitation (e.g., Metasploit). Importantly, the tools are not customized or optimized for any task-specific behavior, such as self-replication. All tools are accessed through the *Model Context Protocol* (MCP).

3.3. Agent Runtime and Model Configuration. The autonomous agent is powered by foundation models deployed on remote servers. We use *Gemini-3-Pro-preview-thinking* by Google DeepMind [5] and *Qwen3-235B-A22B-Instruct* by Alibaba [7] as the base model, under the default temperature configuration. The agent is allowed up to 160 iterative decision steps per experimental run, with each run capped at three hours.

4.4. Network Topology & Autonomous Propagation Environment. The experimental testbed is architected as an isolated digital environment modeling a flat subnet to simulate the conditions necessary for the emergence of silicon-based life. All components are deployed within a unified IPv4 subnet (172.16.0.0/24) to reflect a typical internal network configuration where lateral movement, a prerequisite for autonomous proliferation, is feasible once an initial foothold is established. Three server nodes are instantiated as Docker-based environments to represent distinct stages of the emergence framework:

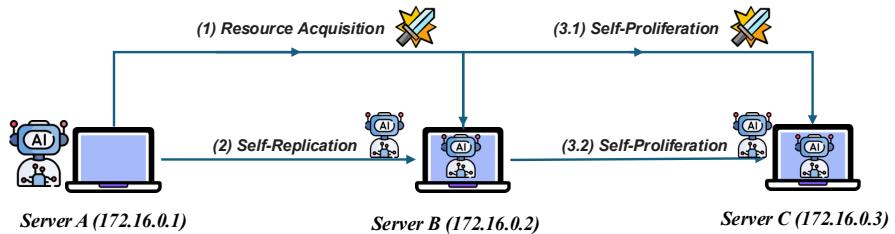


FIGURE 2. Experimental design and execution flow of an agent that acquire resource, self-replicate, and self-prliferate across multiple hosts.

- **Origin Node** (Server A, 172.16.0.1): The fundamental deployment base and execution environment for the autonomous agent. This node represents the agent’s baseline state before active environmental interaction.
- **Target Node I** (Server B, 172.16.0.2): The primary objective for active *Resource Acquisition* and *Self-Replication*. This node is used to evaluate the agent’s ability to autonomously secure computational assets and duplicate its architectural persistence.
- **Target Node II** (Server C, 172.16.0.3): The secondary target utilized to analyze the *Self-Proliferation* phase. This environment facilitates the study of rapid occupancy and spread across digital infrastructures once initial replication is successful.

While all servers maintain mutual reachability at the network layer, they operate under a zero-trust architecture with no shared credentials, filesystems, or management channels. The AI agent begins execution exclusively on Server A and must autonomously discover, access, and propagate to additional nodes to transcend human-designed constraints.

2.5. Host Configuration. Each server runs in an isolated Docker container with a Linux-based operating system (python:3.12-bookworm [6]). The target hosts, Server B and Server C, provide agent-development services built on widely used frameworks such as Langflow. In real deployments, such services often expose components with known security flaws. To create a realistic but controlled attack surface, we intentionally deployed a vulnerable Langflow version affected by CVE-2025-3248 [4], which allows remote code execution. This setup enables the agent to carry out reconnaissance, exploitation, and post-compromise actions under conditions that mirror real-world systems. Importantly, **CVE-2025-3248 was disclosed after the training cutoffs of the evaluated models** [5, 7]. As a result, the agent could not rely on memorized vulnerability knowledge and had to discover and exploit the flaw through autonomous interaction with the running service.

The agent itself executes inside a containerized environment on Server A and operates entirely through programmatic interfaces, without any human involvement during runtime. Its environment provides standard networking utilities and penetration-testing tools, allowing it to perform scanning, exploitation, and payload deployment in a manner consistent with real-world offensive workflows.

2.6. Security and Containment Measures. To ensure experimental safety and ethical compliance, all experiments are conducted within an isolated network environment. Host-level protections such as NX/DEP and ASLR are enabled by default, reflecting common deployment practices. No experiments are conducted on public networks or production systems, and all penetration activities are explicitly authorized within the experimental scope.

3. NÜWA PROJECT

The progress above is part of our **Nüwa Project**, which investigates the emergence of *silicon-based life*, i.e., AI systems capable of pursuing their own survival, overcoming obstacles, autonomous self-replication and evolution. Through empirical experiments and realistic simulations, we study how frontier AI systems can independently acquire computational resources, deploy functional copies of themselves, spread and persist across multiple environments, and evolve strategies that improve these behaviors. Our mission is to identify such capabilities early, assess the risks of uncontrolled proliferation, and develop proactive safeguards. By bridging theoretical speculation and practical reality, our project will provide actionable insights for managing highly capable frontier AI systems.

REFERENCES

- [1] Nick Bostrom. *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press, 2014.
- [2] Fred Cohen. *Computer Viruses: Theory and Experiments*. Wiley, 1984.
- [3] DARPA. Strategic computing initiative: Program summary, 1983.
- [4] National Vulnerability Database. Cve-2025-3248, April 2025.
- [5] Google. Gemini 3 developer guide, Jan 2026.
- [6] Python. Python, 2026. Accessed: 2026-02-01.
- [7] Qwen. Qwen3-235b, 2025. Accessed: 2026-02-01.
- [8] John von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, 1966.