Provisional Patent Application

MWRASP Quantum Defense System

Generated: 2025-08-24 18:14:57

CONFIDENTIAL - GOVERNMENT/CONTRACTOR USE ONLY

UNITED STATES PATENT AND TRADEMARK OFFICE

PROVISIONAL PATENT APPLICATION

TITLE OF INVENTION: SELF-EVOLVING AUTONOMOUS AGENT NETWORK WITH BEHAVIORAL INHERITANCE AND TRUST-BASED REPRODUCTION FOR CYBERSECURITY

INVENTOR(S): [To be provided]

DOCKET NUMBER: MWRASP-003-PROV

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to autonomous agent systems for cybersecurity, specifically to a self-evolving network of defensive agents that spawn, inherit

behaviors, and adapt through trust-based reproduction mechanisms.

Description of Related Art

Current multi-agent systems (US Patent 6,990,406) use fixed agent populations with predetermined roles. Existing systems lack evolutionary mechanisms, behavioral inheritance, or the ability to spawn new agents based on environmental needs.

Genetic algorithms exist in optimization but have not been applied to live cybersecurity agent populations. No prior art shows agents that reproduce based on success metrics or inherit behavioral traits from parent agents.

BRIEF SUMMARY OF THE INVENTION

The present invention creates a living ecosystem of cybersecurity agents that evolve to counter threats through: 1. Autonomous spawning when environmental triggers detected 2. Behavioral trait inheritance from successful parent agents 3. Trust score evolution through peer validation 4. Population control through resource competition 5. Emergent specialization without programming

DETAILED DESCRIPTION OF THE INVENTION

Agent Architecture

Base Agent Structure

```
class EvolvingAgent:
    def __init__(self, parent=None):
        self.id = generate_unique_id()
        self.generation = parent.generation + 1 if parent else 0
        self.behavioral_genes = self.inherit_or_randomize(parent)
        self.trust_score = 0.5  # Start neutral
        self.energy = 100  # Resource for actions/spawning
        self.specialization = self.determine_specialization()
        self.success_history = []
        self.offspring_count = 0
```

Behavioral Inheritance System

Agents inherit traits with mutation:

```
def inherit_or_randomize(self, parent):
    if parent is None:
        return generate_random_traits()

    inherited_traits = parent.behavioral_genes.copy()

    # Apply mutations (5% chance per trait)
    for trait in inherited_traits:
        if random.random() < 0.05:
            inherited_traits[trait] = mutate(inherited_traits[trait])

    return inherited_traits
```

Evolution Triggers

Agents spawn when detecting: - Increased data sensitivity (>threshold) - Network complexity growth (>baseline) - Threat sophistication rise - Performance degradation - Peer agent failures

Trust-Based Reproduction

```
def can_spawn(self):
    # Higher trust = more reproduction rights
    reproduction_threshold = 0.7 - (self.trust_score * 0.4)
    energy_requirement = 50 - (self.trust_score * 20)

    return (self.energy > energy_requirement and
            random.random() > reproduction_threshold)
```

Population Dynamics

Carrying Capacity

System maintains 50-200 agents based on: - Available computational resources - Current threat level - Network size - Data sensitivity

Natural Selection

Agents compete for: - CPU cycles - Memory allocation
- Network bandwidth - Spawning rights

Low-performing agents starve and terminate.

Emergent Specializations

Through evolution, agents develop into: - **Sentinels**: High threat detection accuracy - **Defenders**: Rapid response capabilities - **Analyzers**: Pattern recognition expertise - **Coordinators**: Multi-agent orchestration - **Scouts**: Exploration and reconnaissance - **Healers**: System recovery specialists

Method of Operation

1. **Genesis**: Create initial population of 50 diverse agents
2. **Environmental Monitoring**: Agents assess system state
3. **Trigger Detection**: Identify need for population change
4. **Spawning Decision**: High-trust agents reproduce
5. **Trait Inheritance**: Offspring inherit with mutations
6. **Population Pressure**: Competition for resources
7. **Natural Selection**: Successful agents survive and reproduce
8. **Emergent Behavior**: Specialized roles develop organically

Advantages

- **Adaptive Defense**: Evolution faster than attacker adaptation
- **Unpredictability**: Emergent behaviors cannot be reverse-engineered
- **Resilience**: Population recovers from losses
- **Efficiency**: Specialized agents outperform generalists
- **Scalability**: Population grows with threat level

CLAIMS

1. An autonomous agent system for cybersecurity comprising agents that spawn new agents based on environmental triggers and inherit behavioral traits from parent agents.

2. The system of claim 1, wherein agent reproduction rights are determined by trust scores earned through successful threat mitigation.

3. The system of claim 1, wherein agents compete for computational resources, creating natural selection pressure.

4. The system of claim 1, wherein behavioral traits mutate during inheritance, enabling evolutionary adaptation.

5. The system of claim 1, wherein specialized agent roles emerge through evolution without explicit programming.

6. A method for evolving cybersecurity defenses, comprising:

7. Maintaining a population of autonomous agents

8. Detecting environmental triggers

9. Allowing successful agents to spawn offspring

10. Inheriting behavioral traits with mutations

11. Applying selection pressure through resource competition

ABSTRACT

A self-evolving network of autonomous cybersecurity agents that spawn, inherit behaviors, and adapt through natural selection. The system maintains 50-200 agents that reproduce based on trust scores earned through successful threat mitigation. Offspring inherit behavioral traits from parents with mutations, enabling evolutionary adaptation. Agents compete for computational resources, creating selection pressure that eliminates poor performers. Through this evolutionary process, specialized agent roles emerge organically without explicit programming, creating an unpredictable, adaptive defense that evolves faster than attackers can adapt.

[END OF PROVISIONAL APPLICATION]
