Autonomous AI-driven Penetration Testing System

Menashe Ashkenazi 326648532 and Noam Leshem

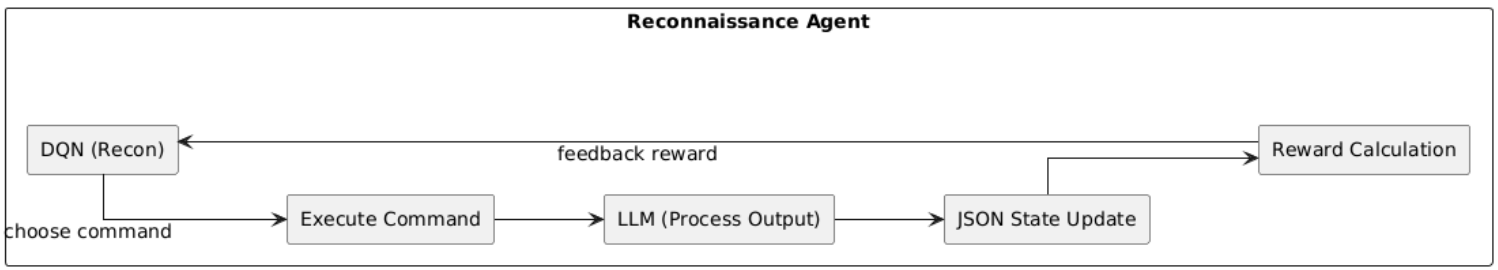
July 8, 2025

Why build an AI-based attacking agent?

- Traditional penetration testing is manual, time-consuming, and limited.
- Many systems remain vulnerable due to lack of continuous automated scanning.
- AI agents can automate, adapt, and learn to discover and exploit vulnerabilities.

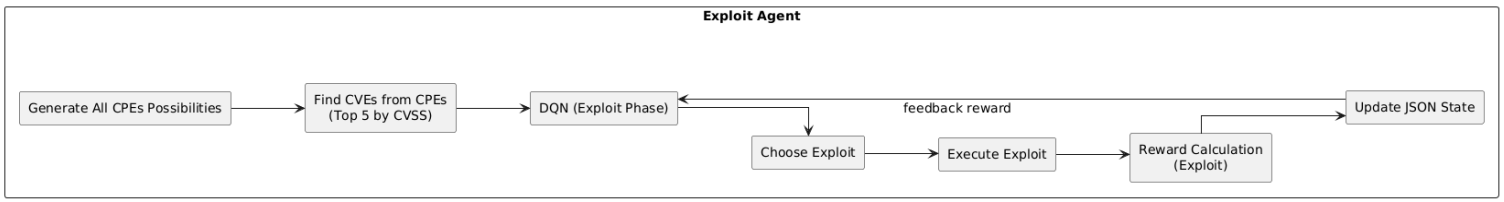
Reconnaissance Agent

- At each state (based on the current collected knowledge), it selects the most optimal next command using a DQN policy.
- Outputs are parsed by an LLM to extract structured information and update the shared JSON state.
- This enables intelligent and adaptive information gathering, focused on maximizing reconnaissance effectiveness over time.



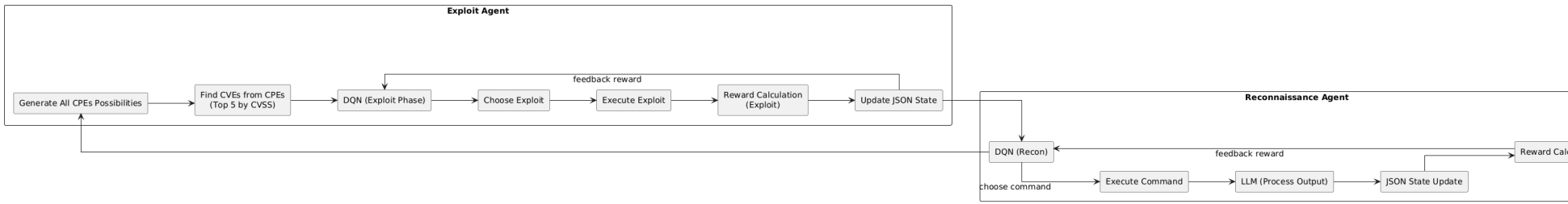
Exploit Agent

- The Exploit Agent receives context-rich input (via JSON) and uses it to identify vulnerable components.
- It leverages over **2,500 real-world exploits** from the Metasploit framework.
- Based on the current environment state, it selects an exploit using a DQN model trained to maximize success.
- Results are used to calculate a reward and further refine the policy for future exploit selection.



System Architecture

- The system consists of two autonomous agents working in coordination:
  - **Reconnaissance Agent** — collects information using LLM parsing and DQN-based exploration.
  - **Exploit Agent** — selects and executes exploits based on the evolving system state.
- Agents communicate through a continuously updated **shared JSON structure**.

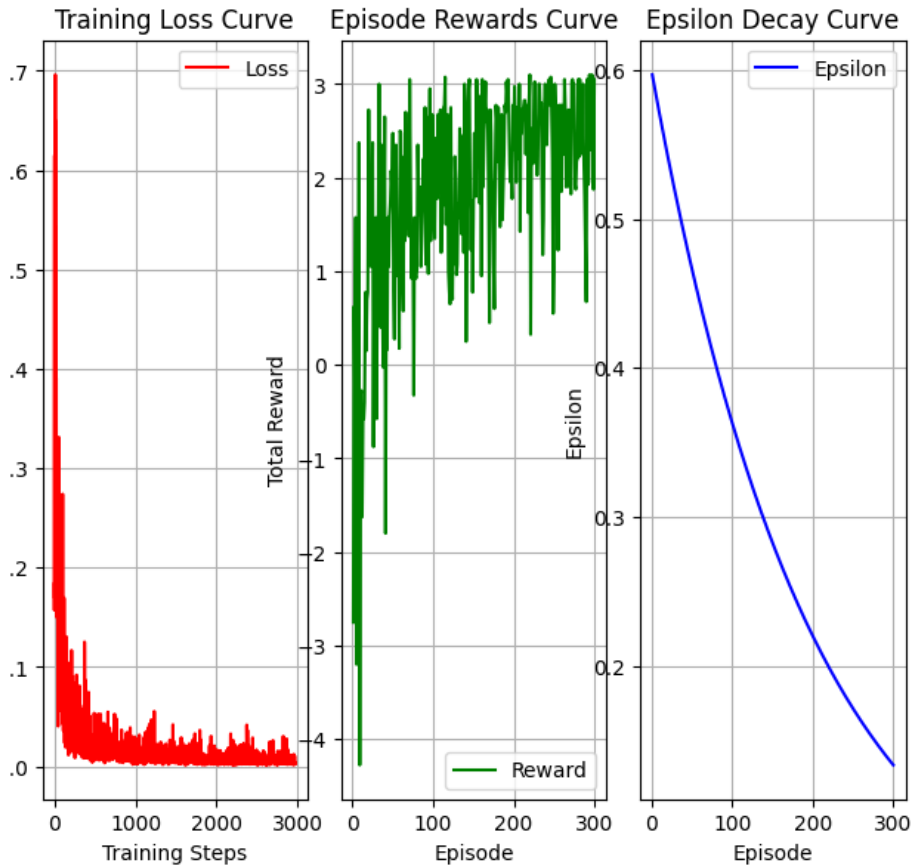


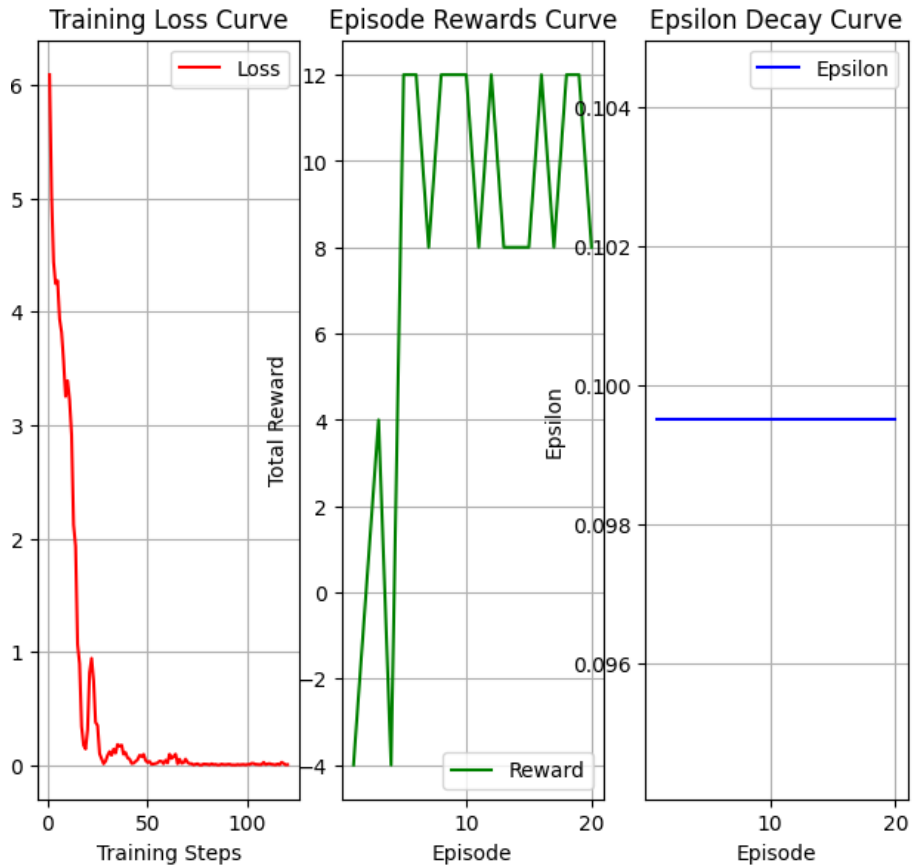
Datasets

- **NVD CVE Dataset:** Contains over **200,000+** publicly reported Common Vulnerabilities and Exposures (CVEs), enriched with metadata such as CVSS scores, affected platforms, and CPE identifiers.
- **Metasploit Exploit Dataset:** Includes more than **2,500 real-world exploits** mapped to CVEs, covering multiple attack vectors and platforms. Used for training and evaluating the Exploit Agent.

Training Results

- **Recon:** Loss < 0.05, 1k steps; R > 2.5, 150 eps.
- **Exploit:** Loss < 0.1, 50 steps; R 10–12, 20 eps.





Comparison with Previous Work

LLM+DQN Architecture – Key Advantages

- **Modular & Extendable:**
  - Swappable DQN Recon and Metasploit Exploit agents
  - Shared state for seamless upgrades
- **Adaptive RL Guidance:**
  - Learns optimal scans from experience
  - No hand-crafted scan logic
- **LLM-Structured State:**
  - Converts raw output into JSON
  - Unified knowledge base for agents
- **Autonomous Sequencing:**
  - No manual task breakdown
  - End-to-end attack planning
- **Real-World Exploits:**
  - Direct Metasploit payloads
  - Broad exploit library used

Challenges and Future work

- Large action space (many tools, options).
- LLM inference is computationally expensive.
- Exploit success is judged solely by outcome, without visibility into underlying causes like firewalls or network filtering.
- Exploit dataset is small

Thank you!

Questions?