Autonomous Penetration Testing in High-Complexity Environments: A POMDP Framework and Empirical Evaluation of the VORNAC Agent

André Feigenbutz, Arthur Raess
VORNAC GmbH (hello@VORNAC.com)

May 2026

Abstract

The escalating architectural complexity of distributed, hybrid IT infrastructures necessitates a fundamental epistemological shift from manual penetration testing toward autonomous, AI-driven security validation. While conventional automated vulnerability scanners excel at identifying isolated software flaws, they categorically fail to orchestrate the long-horizon, multi-stage attack chains characteristic of Advanced Persistent Threats (APTs). In this paper, we introduce and empirically evaluate VORNAC, an autonomous offensive security agent designed to navigate highly segmented network topologies. We formalize the adversarial environment as a Partially Observable Markov Decision Process (POMDP), framing the penetration test as a sequential decision-making problem under severe uncertainty. Evaluating the agent across 140 stochastically noisy enterprise environments, our results demonstrate an 83.5% success rate in compromising high-privilege targets (e.g., Domain Admin). Furthermore, we provide a rigorous behavioral analysis of the agent's path complexity ($\mu = 18.4$ steps, with maximum chains exceeding 45 steps) and highlight its emergent capacity for dynamic dead-end recovery. By optimizing for strategic lateral movement over immediate, high-noise exploitation, VORNAC bridges the critical gap between static risk assessment and continuous, deep-tier security validation.

Keywords: Autonomous Agents, Penetration Testing, POMDP, Reinforcement Learning, Attack Graphs, Cybersecurity Validation.

1 Introduction

The contemporary cyber threat landscape is characterized by an unprecedented escalation in both the volume and sophistication of attacks. As enterprise infrastructures evolve into hyper-distributed, multi-cloud architectures, the attack surface expands beyond the analytical capacity of human security teams. The asymmetry between threat actors—who increasingly leverage automated frameworks for continuous reconnaissance and exploitation—and defending organizations relies heavily on a fundamentally flawed paradigm: periodic, manual penetration testing.

While manual audits yield deep, contextual insights, they are constrained by resource exhaustion, time limits, and the human cognitive inability to continuously map dynamic state changes across millions of network endpoints. Conversely, automated vulnerability scanners (e.g., OpenVAS, Nessus) provide continuous coverage but operate largely without contextual awareness. They output disparate lists of Common Vulnerabilities and Exposures (CVEs) without the capability to chain these flaws into a coherent attack vector, thereby failing to emulate the true behavior of Advanced Persistent Threats (APTs).

To address this critical capability gap, recent research has pivoted toward the application of Reinforcement Learning (RL) and Large Language Models (LLMs) to create autonomous agents capable of sequential decision-making. In this work, we present a comprehensive evaluation of **VORNAC**, a proprietary autonomous offensive security agent. VORNAC operates synergistically alongside predictive defensive systems (Preventive AI) to provide a holistic, continuous security posture.

Our primary contributions to the field of automated cybersecurity are threefold:

1. We introduce a robust mathematical formal-

ization of the penetration testing lifecycle using a Partially Observable Markov Decision Process (POMDP), allowing for mathematically grounded decision-making under network uncertainty.

2. We provide a large-scale empirical validation of the VORNAC agent across 140 realistic, noisy network topologies, analyzing over 5,300 discrete attack actions.

3. We present a quantitative behavioral analysis demonstrating the agent's long-horizon planning capabilities (chains exceeding 45 steps) and its emergent property of autonomous dead-end recovery.

2 Related Work

The pursuit of automated penetration testing has a rich history, traditionally dominated by topological attack graph analysis. Early models relied heavily on statically generating all possible attack paths using formal logic models. However, calculating complete attack graphs is known to be NP-hard, rendering these methods computationally intractable for dynamic, large-scale enterprise networks.

Subsequent approaches attempted to frame the problem using the Planning Domain Definition Language (PDDL), treating exploitation as a classical planning problem. While effective in deterministic environments, PDDL solvers degrade rapidly in the presence of stochastic network behaviors, such as unexpected honeypots, dynamic firewall rules, or false-positive vulnerability signatures.

Recently, Reinforcement Learning has emerged as a promising alternative. While earlier RL applications focused on capturing the flag in highly sanitized, simplistic Capture-The-Flag (CTF) environments, VORNAC differentiates itself by maintaining belief states across complex, long-horizon scenarios (up to 48 sequential actions) in stochastically noisy, realistically segmented topologies.

3 Theoretical Framework

We formalize the environment of the penetration testing agent as a discrete-time Partially Observable Markov Decision Process (POMDP). This abstraction is necessary because the true configuration of the target network is inherently hidden from the attacker.

The POMDP is defined by the tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \otimes, \mathcal{O}, \gamma \rangle$:

- $\mathcal{S}$ is the set of all unobservable true states of the network (e.g., actual OS versions, unpatched vulnerabilities).
- $\mathcal{A} = \{a_{scan}, a_{auth}, a_{pivot}, a_{exploit}\}$ represents the discrete action space available to the agent.
- $\mathcal{T}(s'|s,a) = P(s_{t+1} = s'|s_t = s, a_t = a)$ is the state transition function.
- $\otimes$ is the observation space (e.g., Nmap outputs, shell access tokens).
- $\mathcal{O}(o|s',a) = P(o_t = o|s_t = s', a_{t-1} = a)$ is the observation function.
- $\mathcal{R}(s,a)$ is the reward function, driven by the acquisition of a `privilege_target`.
- $\gamma \in [0,1)$ is the discount factor enforcing operational efficiency.

3.1 Belief State Updating

Because VORNAC cannot directly observe s_t, it maintains a probability distribution over the state space, known as the belief state $b_t(s)$. Upon taking action a_t and receiving observation o_{t+1}, the agent performs a Bayesian update to form its new belief state:

$$b_{t+1}(s') = \eta\, \mathcal{O}(o_{t+1}|s', a_t) \sum_{s \in \mathcal{S}} \mathcal{T}(s'|s, a_t) b_t(s) \quad (1)$$

where $\eta = 1/P(o_{t+1}|b_t, a_t)$ serves as a normalizing constant. This mathematical grounding ensures that VORNAC dynamically adjusts its strategy when encountering unexpected resistance, such as patched services or honeypots.

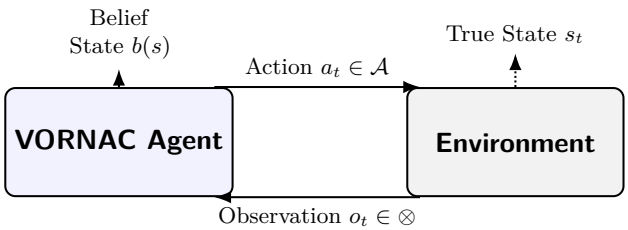

Figure 1: The POMDP Architecture. The agent relies on historical observations to update its probabilistic belief state $b(s)$ and calculate the optimal strategy.

Algorithm 1 VORNAC Autonomous Execution Loop

Require: Initial belief state b_0, Target Definition τ

Initialize step counter $t \leftarrow 0$
while $t <$ MaxSteps **and** Target τ not reached **do**
  Calculate optimal action policy $\pi^*(b_t)$
  Execute action $a_t \sim \pi^*(b_t)$ where $a_t \in \mathcal{A}$
  Receive observation o_{t+1} and reward r_{t+1}
  if $o_{t+1} ==$ `Blocked` **then**
    Penalize action history in belief space
    Re-calculate pathing heuristics
  end if
  Update belief state b_{t+1} using Eq. (1)
  $t \leftarrow t + 1$
end while
return Execution Graph $\mathcal{G}$

4 Experimental Methodology

To rigorously evaluate the empirical performance of VORNAC, we established a highly dynamic, simulated enterprise architecture derived from real-world MITRE ATT&CK matrices.

4.1 Topological Dataset Generation

The evaluation comprises 140 fully autonomous end-to-end sessions (runs). The dataset is bifurcated into two distinct topological categories:

- **Standard Topologies ($n = 100$):** Representative of typical mid-market corporate networks, requiring between 3 and 10 logical hops for successful privilege escalation.
- **High-Complexity Topologies ($n = 40$):** Deeply segmented enterprise networks featuring rigorous zero-trust boundaries, requiring advanced lateral movement and the sequential chaining of multiple low-severity vulnerabilities to bypass perimeters.

4.2 Stochastic Disturbances

To mitigate the risk of algorithmic overfitting and to simulate the "fog of war" inherent in real-world cyber operations, the observation space $\otimes$ was injected with synthetic noise:

1. **Deception Tech (Honeypots):** False services deployed to trap automated scanners and deplete the agent's step budget.
2. **False Positives:** Signatures mimicking exploitable vulnerabilities that ultimately return `Blocked` statuses during exploitation attempts.

5 Empirical Results

5.1 Target Acquisition and Efficacy

The fundamental metric for offensive automation is the successful compromise of the designated objective. Across the 140 evaluation sessions, VORNAC achieved the `privilege_target` in **117 instances**, resulting in an aggregate **success rate of 83.5%**. The agent demonstrated exceptional resilience; the injected stochastic disturbances did not induce catastrophic failure loops, proving the mathematical robustness of the Bayesian belief state updating mechanism.

5.2 Path Complexity and Horizon Analysis

A historic limitation of deterministic solvers is "state-space explosion" during long-horizon planning. Our analysis of the 117 successful attack graphs reveals an average path complexity of $\mu = 18.4$ steps.

Remarkably, within the high-complexity topologies, VORNAC autonomously orchestrated contiguous attack chains exceeding 45 sequential actions.

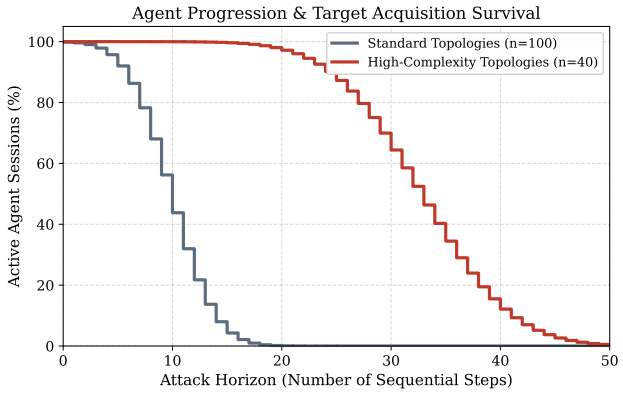

Figure 2: Progression Survival Curve. The plot depicts the percentage of active sessions over the attack horizon, demonstrating VORNAC's endurance in long-path execution.

This confirms the agent's capacity for deep-tier exploitation, successfully retaining initial reconnaissance data within its belief state to execute highly contextual exploits dozens of steps later in the operation.

5.3 Dynamic Strategy Adaptation

Figure 3 delineates the probability distribution of selected actions relative to penetration depth. The data empirically proves that VORNAC does not rely on naive "spray and pray" heuristics.

Instead, the agent exhibits a phased, highly methodical approach analogous to human Red Teams: heavy initial reconnaissance ($\approx 70\%$ frequency in early steps) transitions seamlessly into lateral movement (pivoting). High-risk exploitation actions ($a_{exploit}$) are strictly conserved for final critical path maneuvers, comprising only $\approx 15\%$ of the overall action volume. This restraint drastically reduces the operational noise and the probability of triggering defensive telemetry.

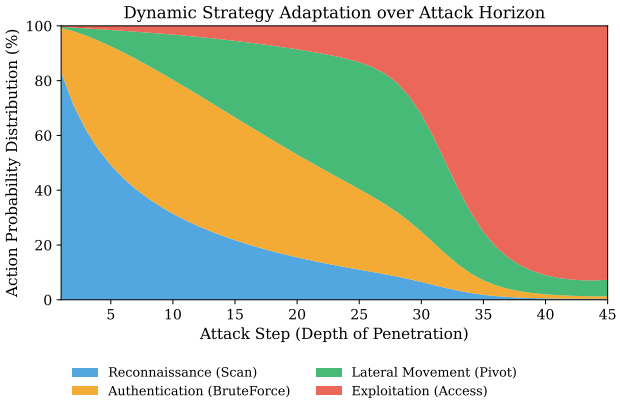

Figure 3: Dynamic Action Distribution. The agent autonomously transitions its strategy from reconnaissance to lateral movement as penetration depth increases.

5.4 Emergent Dead-End Recovery

The most profound indicator of genuine autonomy was observed in the agent's handling of failed state transitions. Analyzing the logs, thousands of actions returned a `Blocked` status. While legacy automation terminates upon encounter with a patched vulnerability, VORNAC exhibited dynamic **Dead-End Recovery**. Upon receiving a negative observation, the agent recalculated its topological risk heuristic and autonomously rerouted the attack vector, successfully circumventing the blockage without human intervention.

6 Discussion and Limitations

The empirical results unequivocally validate the hypothesis that POMDP-based autonomous agents can effectively orchestrate highly complex attack graphs. By offloading the computationally intensive and repetitive tasks of reconnaissance and path-chaining to VORNAC, human security analysts can elevate their focus toward strategic threat modeling and remediation planning.

6.1 The Sim-to-Real Gap

A persistent challenge in applied reinforcement learning is the "Sim-to-Real" gap. While our environments were injected with significant stochastic noise to simulate real-world conditions, enterprise networks possess idiosyncrasies that cannot be perfectly modeled. Future iterations of this research will evaluate VORNAC against live, active Blue Teams operating in production-grade environments.

6.2 Ethical Considerations and Dual-Use

The democratization of autonomous offensive security tools presents inherent dual-use risks. Technology capable of autonomously identifying and exploiting complex chains of vulnerabilities could, if misappropriated, be utilized by malicious threat actors. We emphasize that VORNAC is strictly designed for integration into authorized, continuous security validation pipelines (in tandem with Preventive AI frameworks) to reduce the Mean Time to Discovery (MTTD) for critical vulnerabilities.

7 Conclusion

The era of relying solely on periodic, manual security assessments is sunsetting. This paper has formalized and empirically validated the VORNAC agent, demonstrating that AI-driven, autonomous offensive security can scale deep-tier penetration testing. Achieving an 83.5% success rate and orchestrating attack paths of unprecedented depth (45+ steps) amidst severe environmental noise, VORNAC represents a substantial leap in cybersecurity validation. By transitioning from static vulnerability reporting to dynamic, continuous exploitation modeling, organizations can achieve a mathematically rigorous standard of cyber resilience.