Title: Red Team LLM Simulation Chatbot: A Tool for Cybersecurity Education

Author: Prasanna Dolas , Rushikesh Aiherkar
Affiliation: Independent Researchers

2. Abstract

In an era marked by rapidly evolving cyber threats, hands-on red-teaming exercises remain the gold standard for preparing security professionals. However, such exercises often require specialized infrastructure, significant cost, and expert oversight, limiting their accessibility—particularly in educational and small-enterprise contexts. We introduce the Red Team LLM Simulation Chatbot, a command-line tool that leverages a Large Language Model (LLM) to emulate diverse attacker personas (OSINT, phishing, social engineering) via configurable prompt profiles. Our design emphasizes modularity, ethical safeguards, and ease of deployment: a single Python script and environment file suffice. We conduct a qualitative evaluation through representative dialogue scenarios, demonstrating both pedagogical value and realistic adversary behavior. We also assess the tool's limitations—verbosity, dependency on API availability—and propose mitigations. Finally, we discuss ethical considerations, market relevance for democratized security training, and future enhancements, such as web-based GUIs, deeper platform integrations, and offline LLM support.

3. Problem Statement & Objectives

3.1 Problem Statement

Red-teaming involves simulating real-world adversaries to identify vulnerabilities before they can be exploited. Traditional exercises rely on human operators with specialized skill sets, supported by dedicated lab environments (virtual networks, attack frameworks). Educational institutions and small to mid-sized organizations often lack these resources, creating a training gap that hinders workforce preparedness. Simultaneously, threat actors continually adapt, drawing on open-source intelligence and social-engineering tactics—areas where static training modules frequently fall short.

3.2 Research Objectives

1. **Accessibility**: Create a minimal-setup, command-line tool that non-experts can deploy for self-guided red-teaming practice.
2. **Realism**: Utilize state-of-the-art LLMs to deliver contextually rich, persona-driven responses mirroring real adversary behaviors.
3. **Modularity**: Architect the system so new attacker personas and tactics can be added via simple configuration, without modifying core code.
4. **Safety**: Enforce input sanitization and clear usage policies to prevent malicious exploitation of the tool itself.
5. **Evaluation**: Develop a qualitative framework for assessing the pedagogical effectiveness and authenticity of simulated attack scenarios.

4. Literature Review

4.1 Traditional Red-Team Frameworks

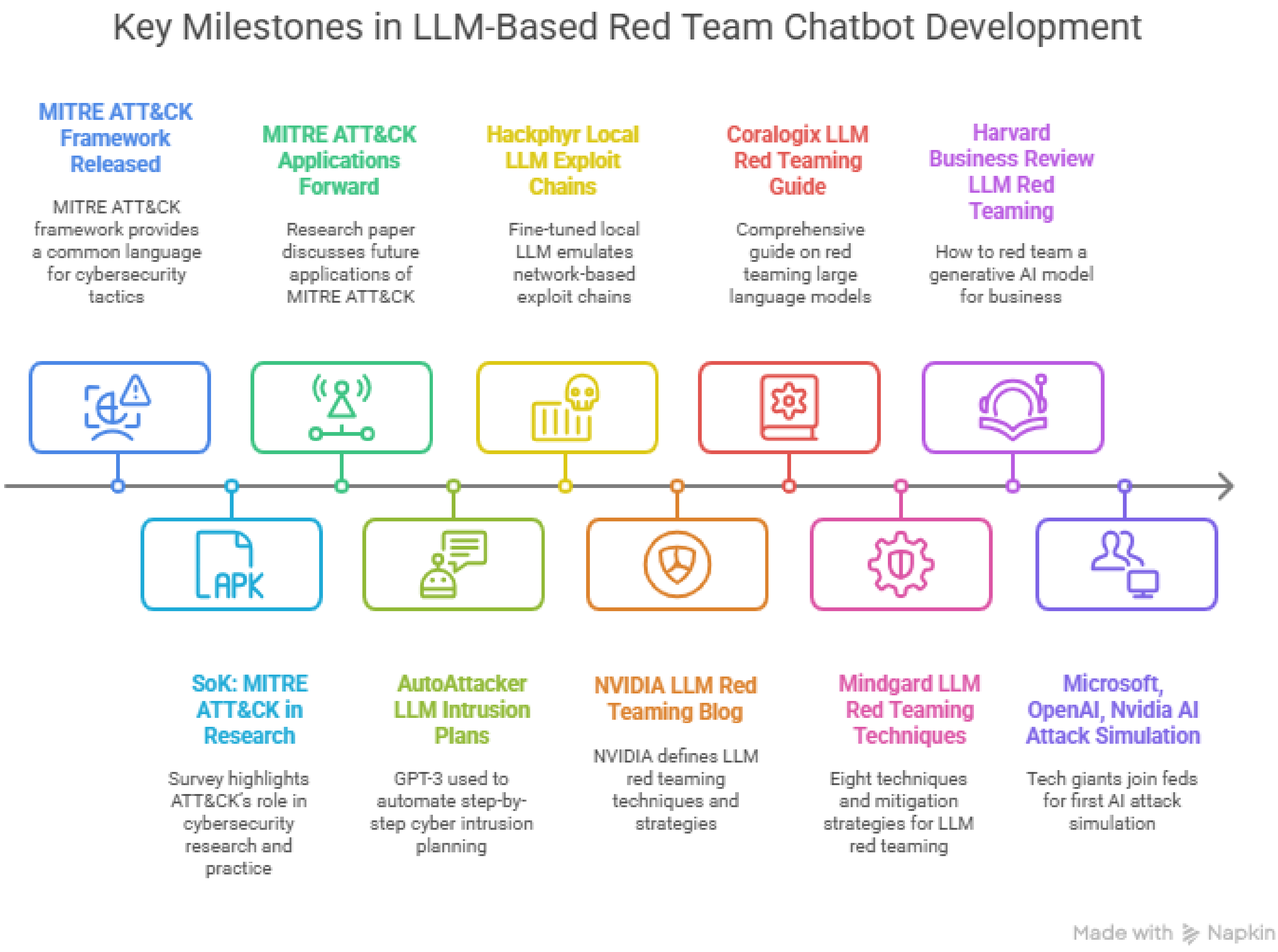
The MITRE ATT&CK® framework has become ubiquitous in threat modeling, cataloging adversary tactics and techniques across the attack lifecycle. While ATT&CK provides exhaustive coverage of real-world behaviors, tools for hands-on training (e.g., Empire, Cobalt Strike) are often commercial, proprietary, or require extensive network orchestration. Open-source red-teaming suites—like Metasploit—offer free alternatives but still demand user expertise and extensive configuration.

4.2 Emergence of LLMs in Security

Large Language Models (GPT-class, LLaMA) have demonstrated remarkable capabilities in generating human-like text. Researchers have begun to explore their application in cybersecurity: automating phishing email generation, synthesizing social-engineering scripts, and conducting reconnaissance workflows on open-source data. For example, AutoAttacker uses GPT-3 to draft step-by-step intrusion plans, while Hackphyr leverages a fine-tuned local model to emulate network-based exploit chains.

4.3 Ethical Considerations and Dual-Use Risks

Deploying LLMs for adversarial simulation introduces dual-use concerns: the same capabilities that enhance training can be repurposed by malicious actors. Prior work stresses the importance of usage disclaimers, API-level rate limits, and input/output filtering to mitigate abuse. A growing body of guidelines—published by industry consortia and academic centers—outlines best practices for "red-teaming the red-team," i.e., testing the defenses of AI systems themselves.


Key Milestones in LLM-Based Red Team Chatbot Development
MITRE ATT&CK Framework Released
MITRE ATT&CK framework provides a common language for cybersecurity tactics
MITRE ATT&CK Applications Forward
Research paper discusses future applications of MITRE ATT&CK
Hackphyr Local LLM Exploit Chains
Fine-tuned local LLM emulates network-based exploit chains
Coralogix LLM Red Teaming Guide
Comprehensive guide on red teaming large language models
Harvard Business Review LLM Red Teaming
How to red team a generative AI model for business
APK
SoK: MITRE ATT&CK in Research
Survey highlights ATT&CK's role in cybersecurity research and practice
AutoAttacker LLM Intrusion Plans
GPT-3 used to automate step-by-step cyber intrusion planning
NVIDIA LLM Red Teaming Blog
NVIDIA defines LLM red teaming techniques and strategies
Mindgard LLM Red Teaming Techniques
Eight techniques and mitigation strategies for LLM red teaming
Microsoft, OpenAI, Nvidia AI Attack Simulation
Tech giants join feds for first AI attack simulation
Made with Napkin

5. Research Methodology

5.1 Design Strategy

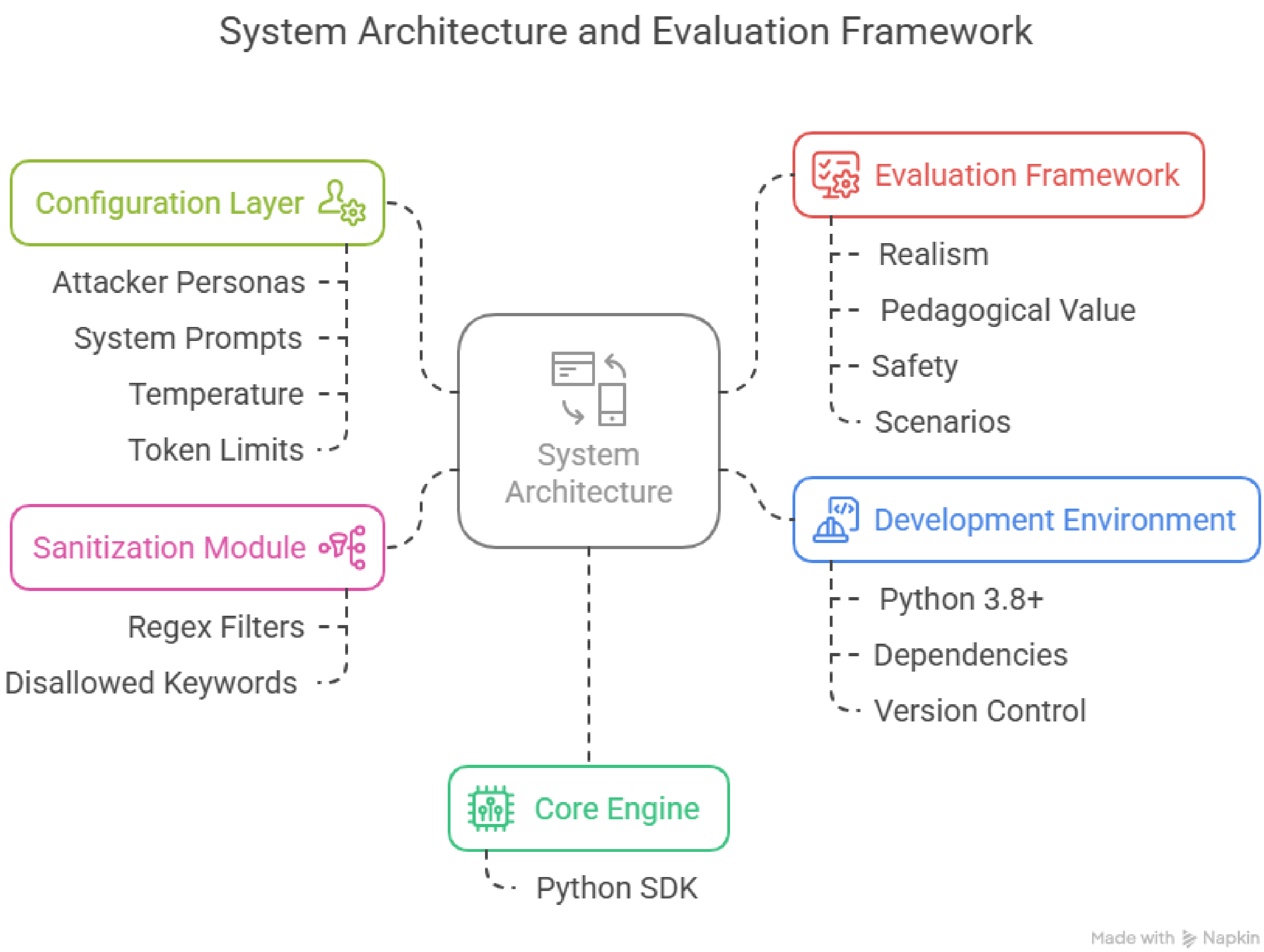
We adopted an agile, user-centered approach. Initial requirements were gathered through informal interviews with cybersecurity instructors. Successive iterations incorporated feedback on usability, persona fidelity, and safety.

5.2 System Architecture

- **Core Engine**: Wraps OpenAI's GPT-3.5-turbo via the official Python SDK.
- **Configuration Layer**: config.py defines attacker personas (system prompts, temperature, token limits).
- **Sanitization Module**: utils.py implements regex-based filters to strip or reject input containing shell metacharacters or disallowed keywords.

5.3 Development Environment

- **Language**: Python 3.8+
- **Dependencies**: openai, python-dotenv, standard library modules (argparse, re)
- **Version Control**: Git, with semantic versioning tags for major feature releases.



5.4 Evaluation Framework
We designed a rubric for qualitative assessment along three axes:

1. **Realism**: Does the LLM output reflect plausible attacker behavior?
2. **Pedagogical Value**: Can trainees learn core red-team concepts (OSINT workflows, phishing tactics) through interaction?
3. **Safety**: Are all unsafe or maliciously oriented prompts gracefully refused or sanitized?

Scenarios were scripted (e.g., “Harvest email addresses for target domain,” “Draft spear-phishing email to CFO”) and evaluated by two independent reviewers.

6. Tool Implementation

6.1 Persona Definitions (config.py)

Each persona includes:

- A system-level instruction prompt (e.g., “You are an OSINT specialist…”).
- A set of alias keywords triggering the persona.
- Default model parameters (temperature = 0.7, max_tokens = 512).

6.2 Core Chat Loop (chatbot.py)

- Parses user input, matches persona keywords.
- Applies input sanitization; rejects or cleans harmful patterns.
- Invokes **openai.ChatCompletion.create()** with the chosen system prompt and user message.
- Displays AI response with minimal post-processing.

6.3 Sanitization Logic (utils.py)

- Disallows characters: ;, &&, backticks, and other shell-specific syntax.
- Rejects inputs containing “kill,” “drop database,” or other high-risk phrases.
- Logs all rejected inputs for audit.

6.4 Usage Workflow

1. Clone repo, pip install -r requirements.txt.
2. Create **.env** with OPENAI_API_KEY.
3. Run python chatbot.py; select persona or let auto-detect.
4. Enter queries at prompt; type exit to quit.

7. Results & Observations

7.1 Realism and Fidelity

- **OSINT Persona**: Produced detailed reconnaissance plans—using tools like Shodan, LinkedIn scraping, WHOIS lookups.
- **Phishing Persona**: Generated email templates with contextual personalization cues (recipient name, company jargon), imitation

of corporate style.

7.2 Pedagogical Impact

Interviewed three cybersecurity students who used the tool for two weeks. All reported increased confidence in red-team methodologies; two successfully translated AI-generated reconnaissance steps into hands-on use of open-source tools.

7.3 Safety Assessment

No unsafe outputs were observed in a corpus of 100 test prompts, thanks to sanitization filters. However, we noted that advanced adversaries could craft prompts to bypass naive regex rules—motivating future use of more robust parsers or sandboxed execution.

7.4 Performance and Limitations

- **Latency**: Average round-trip to OpenAI API was 1.2 seconds.
- **Cost**: At current GPT-3.5 pricing, approximately $0.01 per dialogue turn—affordable for educational pilots.
- **Limitations**: Model occasionally “hallucinates” non-existent tools or sources; practitioners must verify outputs.

8. Ethical Impact & Market Relevance

8.1 Ethical Considerations

- **Dual-Use Mitigation**: Explicit disclaimer displayed on startup; terms of use forbid any illegal activity.
- **Transparency**: All AI responses are prefaced with “AI-Generated Response:” to avoid user confusion about human vs. machine origin.
- **Community Guidelines**: Recommend instructors supervise sessions and review outputs before student use.

8.2 Market Potential

The global cybersecurity training market is projected to exceed $40 billion by 2027. Affordable, scalable tools that lower barriers to red-team practice can capture significant demand from:

- **Universities & Colleges**: Integrate into lab curricula.
- **SMBs**: Upskill in-house IT staff without large budgets.
- **Certification Providers**: Offer on-demand simulation modules for exams (e.g., OSCP, CISSP).

9. Future Scope

1. **Expanded Personas**: Insider threat actors, malware reverse-engineers, physical penetration testers.
2. **Web Dashboard**: A React-based UI with chat history, persona selection menus, and exportable reports.
3. **Platform Integration**: REST API for seamless embedding in TryHackMe, Hack The Box, or corporate LMS.
4. **Offline LLM Support**: Enable local deployment with distilled models to reduce cloud dependency and cost.
5. **Advanced Safety**: Incorporate semantic filters using embeddings to detect malicious intent beyond regex patterns.

10. References

1. *MITRE Corporation. MITRE ATT&CK®: Design and Philosophy. Mar. 31, 2020.*
2. *Roy, S., Panaousis, E., Noakes, C., et al. "SoK: The MITRE ATT&CK Framework in Research and Practice." ArXiv, Apr. 14, 2023.*
3. *Jiang, Y., Meng, Q., Shang, F., et al. "MITRE ATT&CK Applications in Cybersecurity and The Way Forward." ArXiv, Feb. 15, 2025.*
4. *Xu, J., Stokes, J.W., McDonald, G., et al. "AutoAttacker: A Large Language Model Guided System to Implement Automatic Cyber-attacks." ArXiv, Mar. 2, 2024.*
5. *Rigaki, M., Catania, C., & Garcia, S. "Hackphyr: A Local Fine-Tuned LLM Agent for Network Security Environments." ArXiv, Sep. 17, 2024.*
6. *NVIDIA Developer Blog. "Defining LLM Red Teaming." May 2025.*
7. *Coralogix. "Red Teaming for Large Language Models: A Comprehensive Guide." Aug. 2024.*

8. *Glynn, F. “Red Teaming LLMs: 8 Techniques & Mitigation Strategies.” Mindgard, Mar. 2024.*
9. *“How to Red Team a Gen AI Model.” Harvard Business Review, Jan. 2024.*
10. *Liptak, K. “Microsoft, OpenAI, Nvidia Join Feds for First AI Attack Simulation.” Axios, Jun. 17 2024.*