Transferring a multi-BERT pentesting agent to real networks

GENEVEY-METAT Christophe, NOLLET Guillaume, SATRE Pierre-Marie, SCOTTO DI PERROTOLO Loïc, CUSINATO Jean-Léon, DENIS Evan, QUEMAT Adrien, GUITTER Pierrick, GESNY Olivier – NEVERHACK

Identification of Use Cases

Our work falls within the field of automated cyberattack orchestration. The goal is to provide offensive and defensive security teams with a solution capable of autonomously and continuously simulating attacker behaviours on an information system. Such an agent makes it possible to increase the frequency of security tests, cover broader scopes, continuously validate deployed detection capabilities, and provide a first level of assessment before in-depth human intervention.

We developed and validated an attack agent trained on simulated networks (NASim environment [1]), then evaluated its ability to generalize to new topologies that were not used for training [2]. Our contribution aims to assess this agent-driven attack orchestration on real networks through a command-and-control (C2) infrastructure.

The transition from simulation to reality reveals two technical challenges that shape the approach presented in the following paragraph.

- **Challenge A: Generalizing to unknown real networks.** The agent must operate on topologies it was not trained on and move beyond the abstraction of its training environment: NASim exposes idealized actions that return structured dictionaries, whereas real decisions are projected onto multiple distinct attack primitives whose outputs are not structured.
- **Challenge B: Adapting in uncertain environments.** On a real target, an action may fail for unobservable reasons (missing session history, insufficiently supported inter-machine communications, etc.). The agent must be able to reassess its approach and revise its trajectory.

Proposed Technical Approach

Our agent, RedTeamBERT [2], is a pipeline of four BERT-type transformer models, each specialized in a sub-decision of the attack process. This modular, lightweight architecture (~2M parameters in total) was trained on NASim scenarios featuring 4 machines and multiple subnets; an abstract textual representation of the network (anonymized roles, statuses, services, and OSes) promotes generalization.

Challenge A (Generalization to unknown networks) is addressed by two complementary mechanisms, a padding system and a roll system, that allow the pipeline to adapt to topologies smaller or larger than those encountered during training. On previously unseen NASim scenarios with 5 to 10 subnets and 15 to 40 machines, this approach achieves a 93.5% success rate [2]. For Challenge B (Adaptation in uncertain environments), we propose re-evaluating any action that fails repeatedly: it is then excluded from selection for a given period.

Our contribution concerns the evaluation of this pipeline under real-world conditions. The agent uses a command-and-control (C2) server to control the implants deployed on already-compromised machines. Four adaptations were required to handle these realistic conditions:

1. **Managing credentials:** Since NASim does not model authentication, we provide the model with a list of credentials, from which it selects the username/password pair to test during lateral movement actions.
2. **Selecting the attack source:** The model must choose, for each action, the implant from which it is launched. We keep track of deployed implants, then select the one that is connected to the chosen target.
3. **Transposing meta-decisions into real actions**: lhe decision made by the agent is projected onto multiple primitives executable by the implant. For instance, the decision to compromise a machine becomes a set of lateral movement attempts using the provided credentials. This addresses the second aspect of Challenge A.
4. **Handling imperfect information**: Real discovery actions produce noise. We observed, for instance, reports of addresses corresponding to no exploitable machine within the scope, such as the DNS address 8.8.8.8. We deliberately filter out these results before integrating them into the state observed by the model, so as to separate the issue of transferability from that of noise robustness.

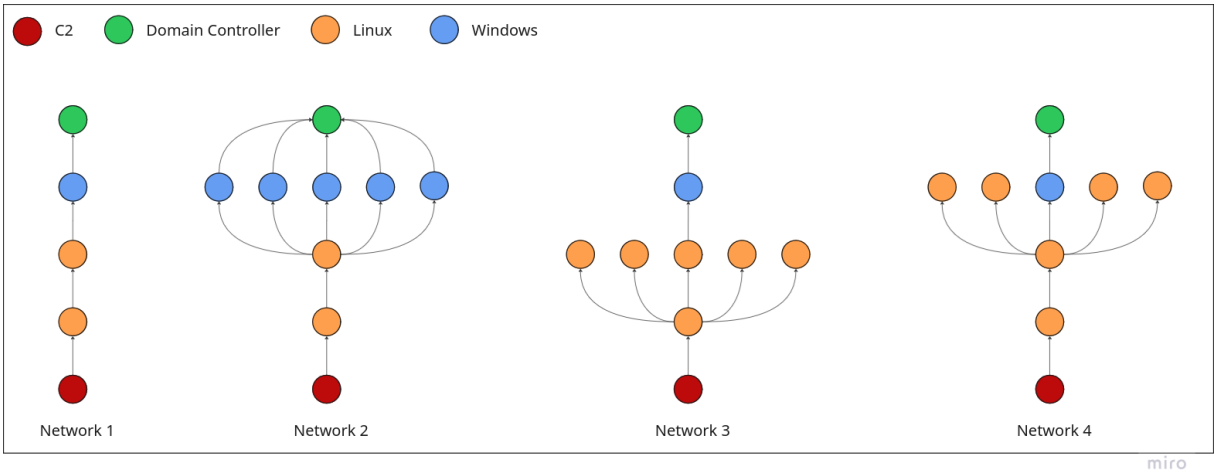

Fig. 1. Test network architectures.

Since the pipeline's meta-decisions remain unchanged, the SHAP-value explainability analysis presented in [2] applies as-is to real campaigns: the reason for each action observed on their information system can be reported back to defensive teams.

The prototype has been validated in simulation and is currently in the transfer phase to a pre-operational environment. Additional tests are planned on deeper topologies and more varied services. Future work will focus on extending the protocol: more services, integration of credentials, and implant selection into the learning loop itself.

[1] *Schwartz, J., & Kurniawati, H. (2019). Autonomous Penetration Testing using Reinforcement Learning. ArXiv, abs/1905.05965.*

[2] *C. Genevey-Metat et al., "Red Team BERT: Reliable Automated Penetration Testing with Multi-BERT Architecture," in Conference on Artificial Intelligence for Defense (CAID), 2025.*