Charger Active Defense v1.0 Team 2 - Group 12

Noah Sickels, Adam Brannon, William Lochte

## Background / Abstract

Modern attack tools are highly efficient, while the cybersecurity industry struggles with active defense capabilities. Current passive defenses primarily focus on mitigating threats through means that can disrupt business operations or can be circumvented. Additionally, with the rapid advancement of artificial intelligence and prompt engineering, attackers can instantly generate various attack tools to target organizations.

The Charger Active Defense project focuses on developing a network-based fuzzing workflow to effectively and comprehensively test these known and AI/LLM-generated attack tools. It aims to identify responses that may cause the attacking application to crash or hang. Responses generated could be saved and sent back to the adversary through a Python replay service during a detected attack. Due to the multi-threading nature of many attack tools like Masscan and Medusa, effectively fuzzing these tools is complicated. If a fuzzed response leads to a crash or hang, it typically occurs in a thread separate from the attacking application's instance. This crashed thread may either be reported as a false positive or not counted at all. Because of this discovery, at the customer's request, we are pivoting towards thoroughly fuzzing generated attack tools from large language models.

Since we have successfully applied the fuzzing workflow to Masscan, we will also pursue the integration of ThreadSanitizer (TSan) to determine if current fuzz testing is properly crashing threads. The end goal of this workflow is to assess how we can broaden this method to apply to other attack tools, providing organizations with a strategy to defend their systems against an ever-evolving threat landscape.

## Current Project Status, Issues, & Short-Term Activities & Goals

Our short-term goals for this period were:

During this period, we encountered a few issues:

* In order to use GitHub Copilot for personal use, we had to request a few educational license. Although this was done in the week prior, it took a couple days before they got back to us and approved us. Fortunately, we were able to get approved in time to complete the necessary objectives on time.

For the next reporting period, our short-term goals are:

|  |  |
| --- | --- |
| Responsibility | Task |
| Noah Sickels | Apply fuzzing workflow to 1st set of Gen-AI attack tools. |
| Adam Brannon | Perform LDRA static analysis on 1st set of Gen-AI attack tools. |
| William Lochte | Perform Valgrind dynamic memory leak testing on 1st set of Gen-AI attack tools. |

## Milestone Status Summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Milestone / Task | Projected Due Date | Completion Date | Status | Notes |
| Develop Gen-AI Tool #1 (Phind model) | 2/14 | 2/14 | Complete |  |
| Develop Gen-AI Tool #2 (Copilot model) | 2/14 | 2/14 | Complete |  |
| Python Service Response Module | 2/14 | 2/13 | Complete |  |

## Level of Effort / Individual Responsibility Record

|  |  |  |
| --- | --- | --- |
| Member | Hours / Period | Total Hours / Spring |
| Noah Sickels |  |  |
| Adam Brannon |  |  |
| William Lochte |  |  |
| Total |  |  |

|  |  |
| --- | --- |
| Member | Individual Accomplishments |
| Noah Sickels |  |
| Adam Brannon |  |
| William Lochte |  |

## Milestone Completion & Analysis

### Mitigations & Contingency Plans

* N/A – No missed milestones for this period.