

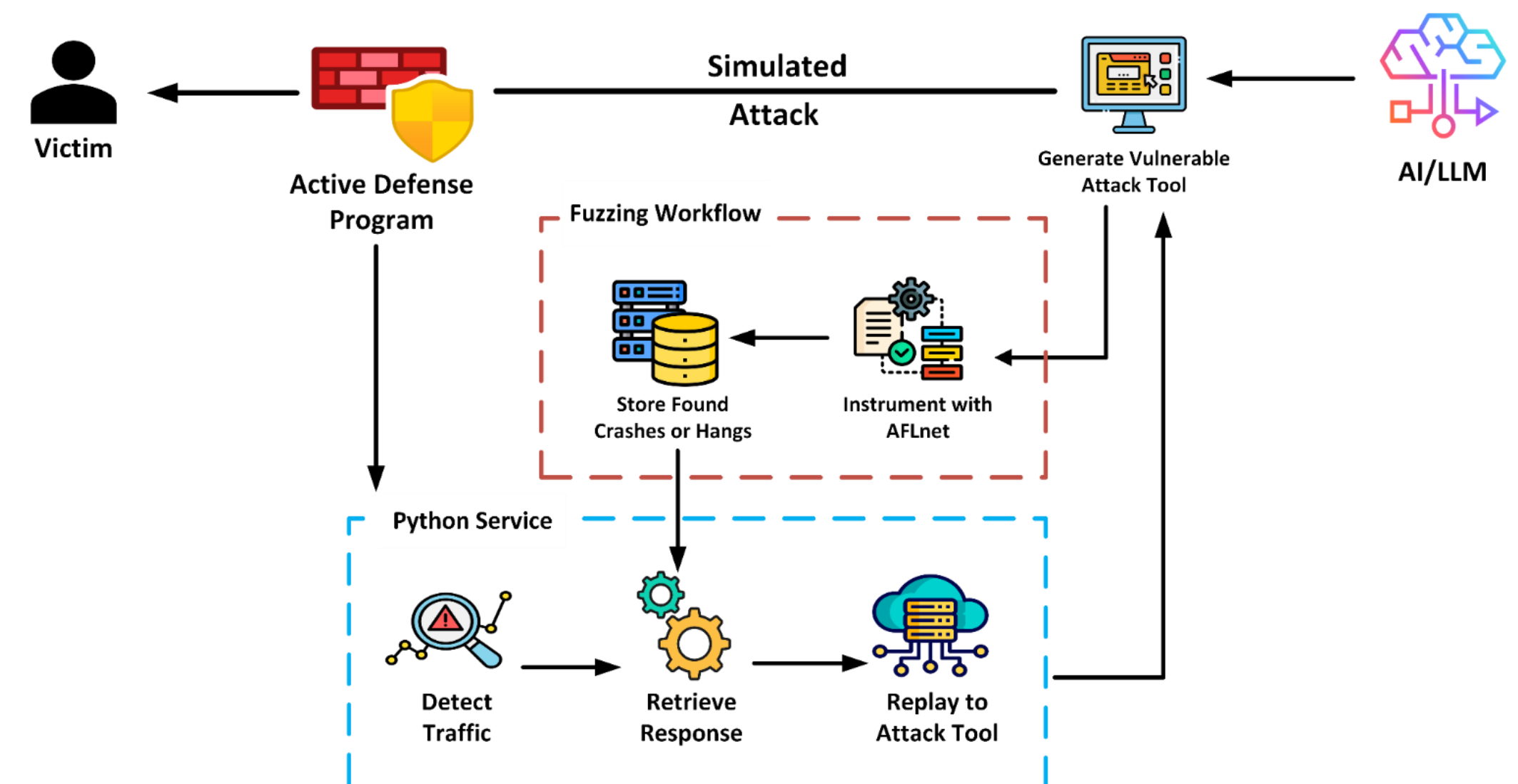
Charger Active Defense Proof of Concept Testbed

Adam Brannon, CPE, Noah Sickel, CPE, William Lochte, CPE

Project Sponsor: Dr. David Coe, Associate Professor, ECE, UAH

Overview of Project Work

- The goal of Charger Active Defense (ChAD) is to show the feasibility of active cybersecurity defense that crashes or hangs an attacker's tools using invalid network responses.
- We performed fuzz testing on both existing and AI generated attack tools to find network responses that crashed or hung the tools



Highlight of Requirements

M1	The ChAD fuzzing workflow must conduct network-based fuzzing to identify network responses, also known as Active Defense Responses, that can crash or hang adversarial attack tools.
M4	Must use two different AI/LLM models to generate additional attack tools.
M7	Demonstrate a fuzz testing workflow for Masscan and AI-generated attack tools.
M10	The ChAD program must provide an active defense response.

```
american fuzzy lop 2.56b (phind_brute)

process timing
  run time : 0 days, 0 hrs, 0 min, 1 sec
  last new path : none seen yet
  last uniq crash : none seen yet
  last uniq hang : none seen yet
cycle progress
  now processing : 0 (0.00%)
  paths timed out : 0 (0.00%)
stage progress
  now trying : bitflip 1/1
  stage execs : 59/96 (61.46%)
  total execs : 68
  exec speed : 34.44/sec (slow!)
fuzzing strategy yields
  bit flips : 0/0, 0/0, 0/0
  byte flips : 0/0, 0/0, 0/0
  arithmetics : 0/0, 0/0, 0/0
  known ints : 0/0, 0/0, 0/0
  dictionary : 0/0, 0/0, 0/0
  havoc : 0/0, 0/0
  trim : n/a, n/a
map coverage
  map density : 0.01% / 0.01%
  count coverage : 1.00 bits/tuple
findings in depth
  favored paths : 1 (100.00%)
  new edges on : 1 (100.00%)
  total crashes : 0 (0 unique)
  total tmouts : 0 (0 unique)
path geometry
  levels : 1
  pending : 1
  pend fav : 0
  own finds : 0
  imported : n/a
  stability : 100.00%

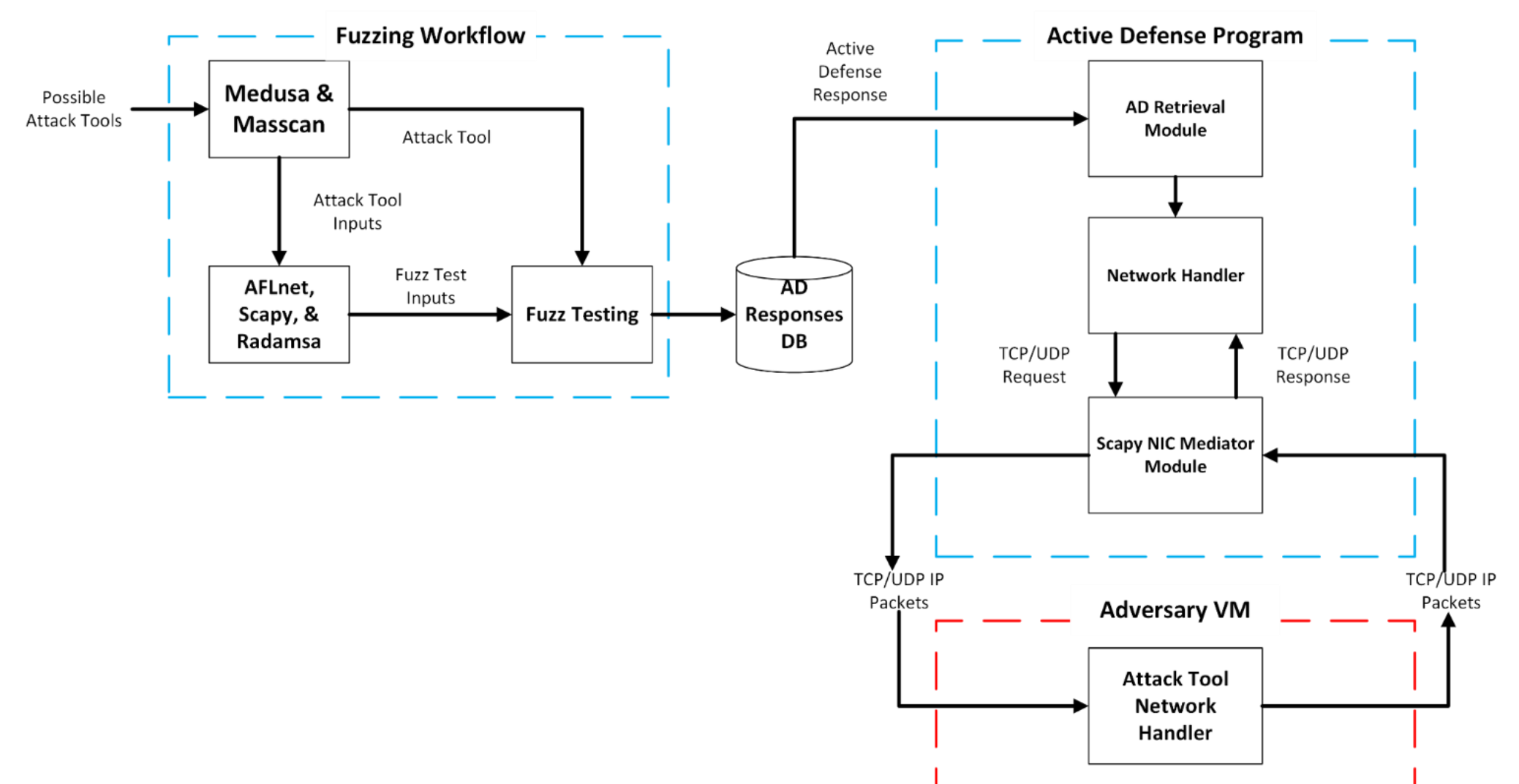
overall results
  cycles done : 0
  total paths : 1
  uniq crashes : 0
  uniq hangs : 0

[cpu000: 79%]
```

Design

Our design is split into two primary sections:

- Fuzzing Workflow:** Includes everything related to finding vulnerabilities within the open source tools
- ChAD Program:** The replay service for providing network responses to the attacking application



Results/Impact

- Our fuzzing workflow was able to crash 3 out of 6 AI generated attack tools and hang 1 of them.
- We were unable to crash or hang Masscan, our chosen well-known attack tool.
- Could lead to buffer overflow attacks which would allow nation-states to directly retaliate against the attacker (and collect more information on them).

