Fuzzing FTW

DEF CON 26

Newest Slides & Other Training Materials

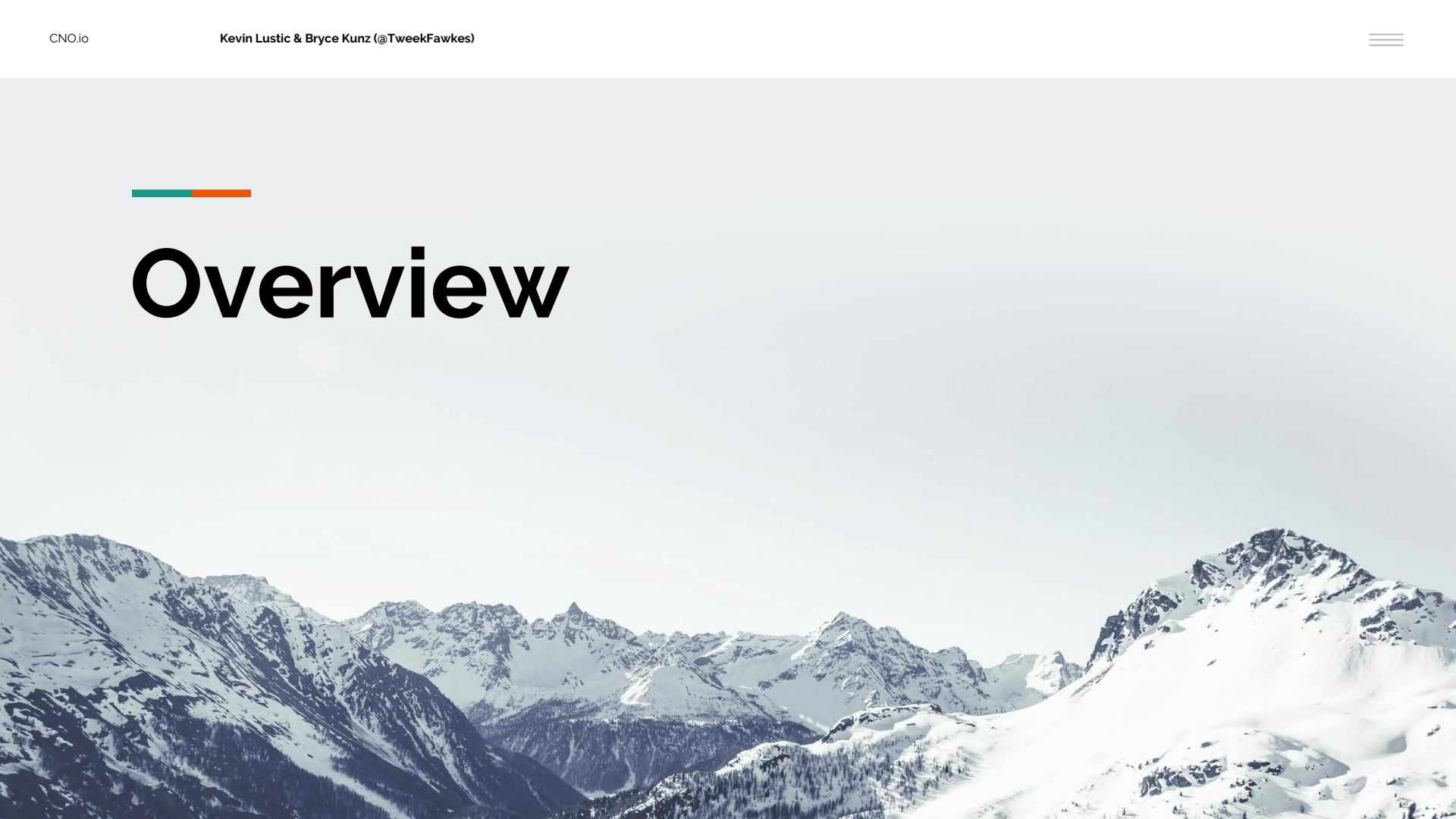
The newest slides & other training materials for this workshop are on GitHub at:

- https://github.com/cno-io/fuzzing_ftw

Who Are We?

- Kevin Lustic - Red Team Lead at Adobe DX
- Bryce Kunz (@TweekFawkes) - Stage 2 Security, Red Teaming & Splunk Security Services

Overview



What is Fuzzing?
Overview

Fuzzing is manipulating data, intended to be processed by an information system, through an automated and repeatable process.

Fuzzing Process
Overview

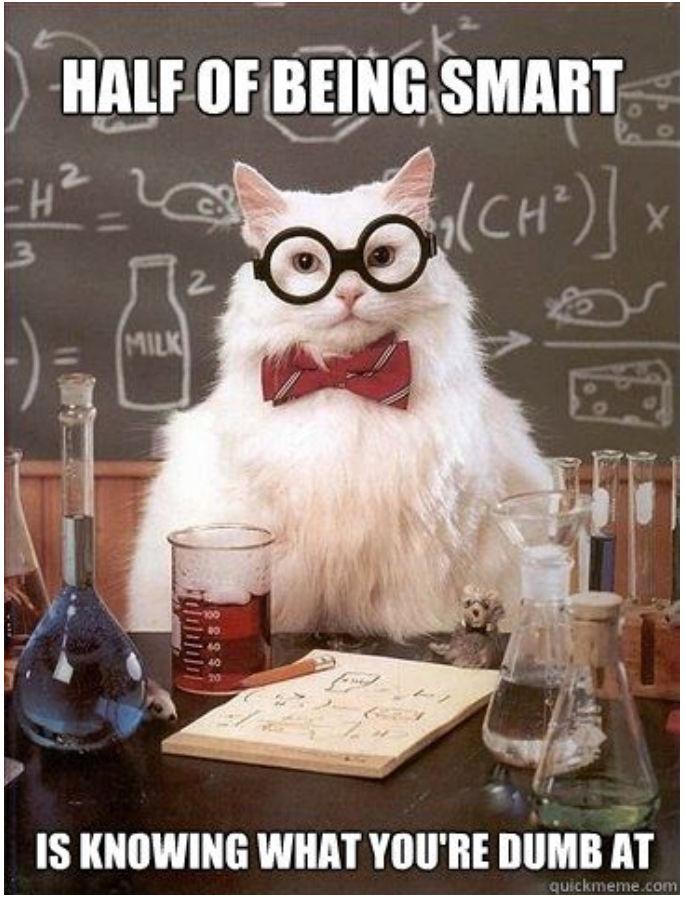
Fuzzing Process Overview:

- Acquire Knowledge
- Instrumentation
- Delivery
- Generation
- Scale
- Repeat!

#1 Acquire Knowledge
Fuzzing Process

Just enough to do some effective fuzzing, do not over think it.

- Prior Research
- Stack Overflow
- Documentation
- Code Review
- Reverse Engineering
- Etc...

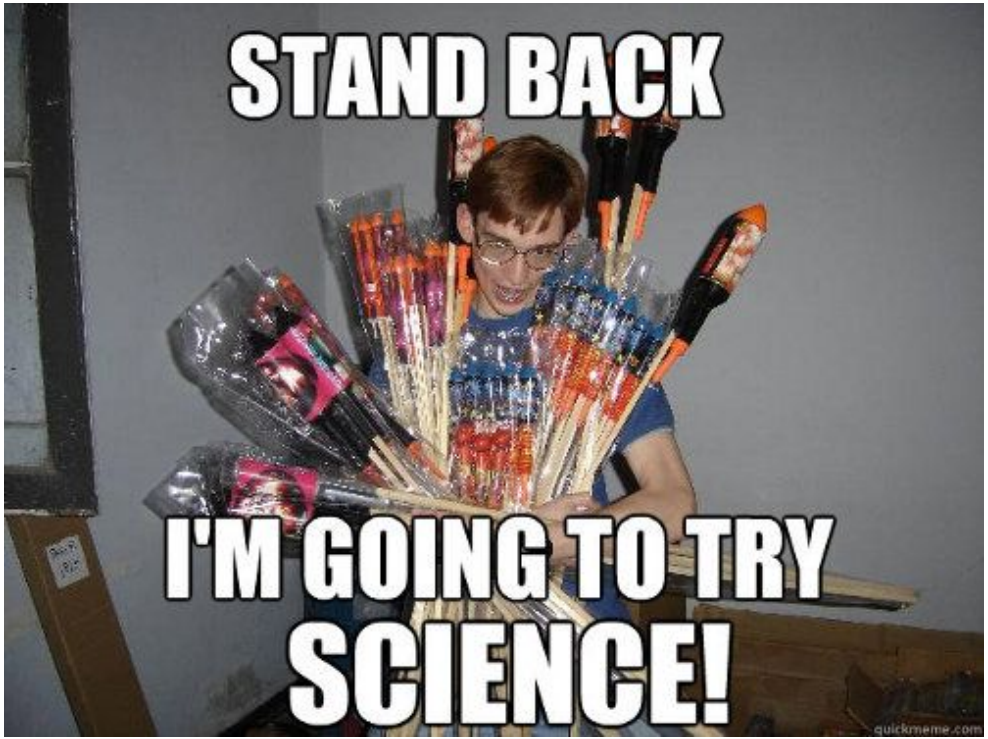


#2 Instrumentation
Fuzzing Process

How will we know when the process has crashed?

#3 Delivery
Fuzzing Process

How will we get our fuzzed payloads to the target?

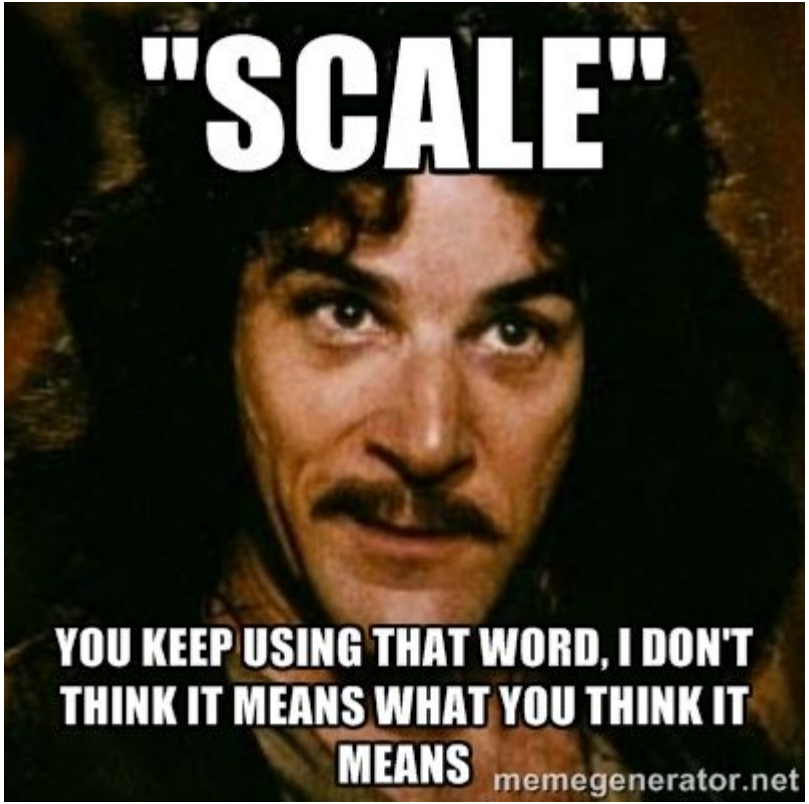


#4 Generation
Fuzzing Process

How will we generate new fuzzed payloads?

#5 Scale
Fuzzing Process

How will we scale this fuzzing operations?



#6 Repeat!
Fuzzing Process

Get a minimal viable fuzzing operations underway ASAP, then come back to each of these steps and progressively improve the operation over time.

Blind Fuzzing with Radamsa

Blind Fuzzing
Fuzzing Process

Blind (or Dumb) Fuzzing

- Input data is corrupted randomly without awareness of expected format.

Radamsa
Fuzzing Process

Radamsa is a test case generator for robustness testing, a.k.a. a fuzzer.

It is typically used to test how well a program can withstand malformed and potentially malicious inputs.

It works by reading sample files of valid data and generating interestingly different outputs from them.

AddressBook
Fuzzing Process

Blind Fuzzing with Radamsa

Demo

Blind Fuzzing with Radamsa

Hands-On Lab

Function Fuzzing with libFuzzer

Function Fuzzing
Fuzzing Process

Functional fuzzing focuses on code-coverage

Testing is often gray- or white-box to ensure this

Function Fuzzing
Fuzzing Process

libFuzzer is a coverage-based functional fuzzing library.

Define an entrypoint, link your code against libFuzzer, and execute your binary normally.

libFuzzer will provide instrumentation around an otherwise dumb fuzzing process:

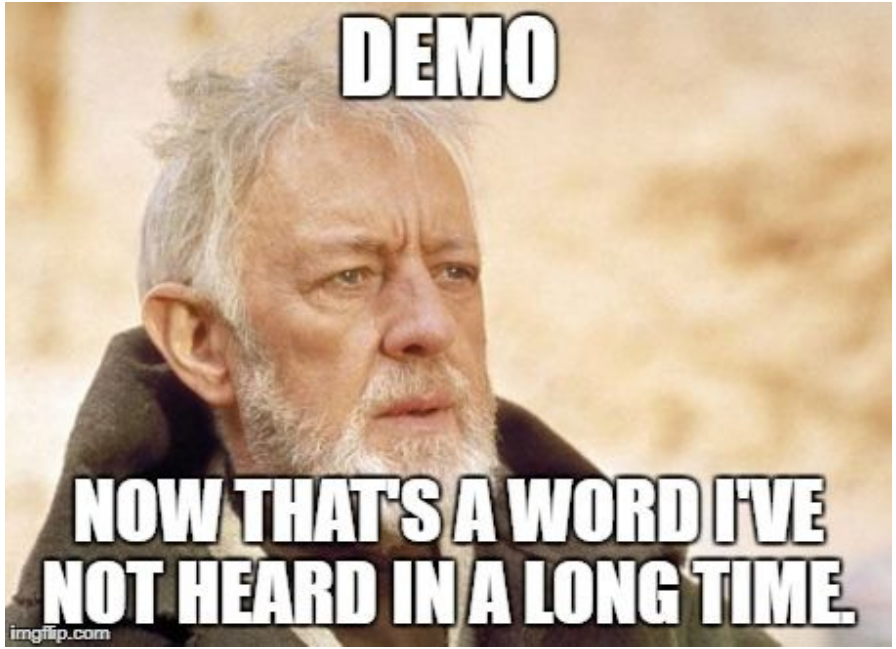
- Generate garbage input data
- Record debug information and the input post-crash

libFuzzer
Fuzzing Process

```
// fuzz_target.cc
extern "C" int LLVMFuzzerTestOneInput(const uint8_t *Data, size_t Size) {
  DoSomethingInterestingWithMyAPI(Data, Size);
  return 0;  // Non-zero return values are reserved for future use.
}
```

Function Fuzzing with libFuzzer

Demo



Function Fuzzing with libFuzzer

Hands-On Lab

File Fuzzing with AFL

File Fuzzing
Fuzzing Process

Fuzzing strategy for mutating or generating input files

Most useful if the app under test uses configuration files or heavily parses any other files

AFL
Fuzzing Process

Can instrument binaries white-box (with source code) or black-box

For white-box instrumentation, compile with AFL's provided compiler.

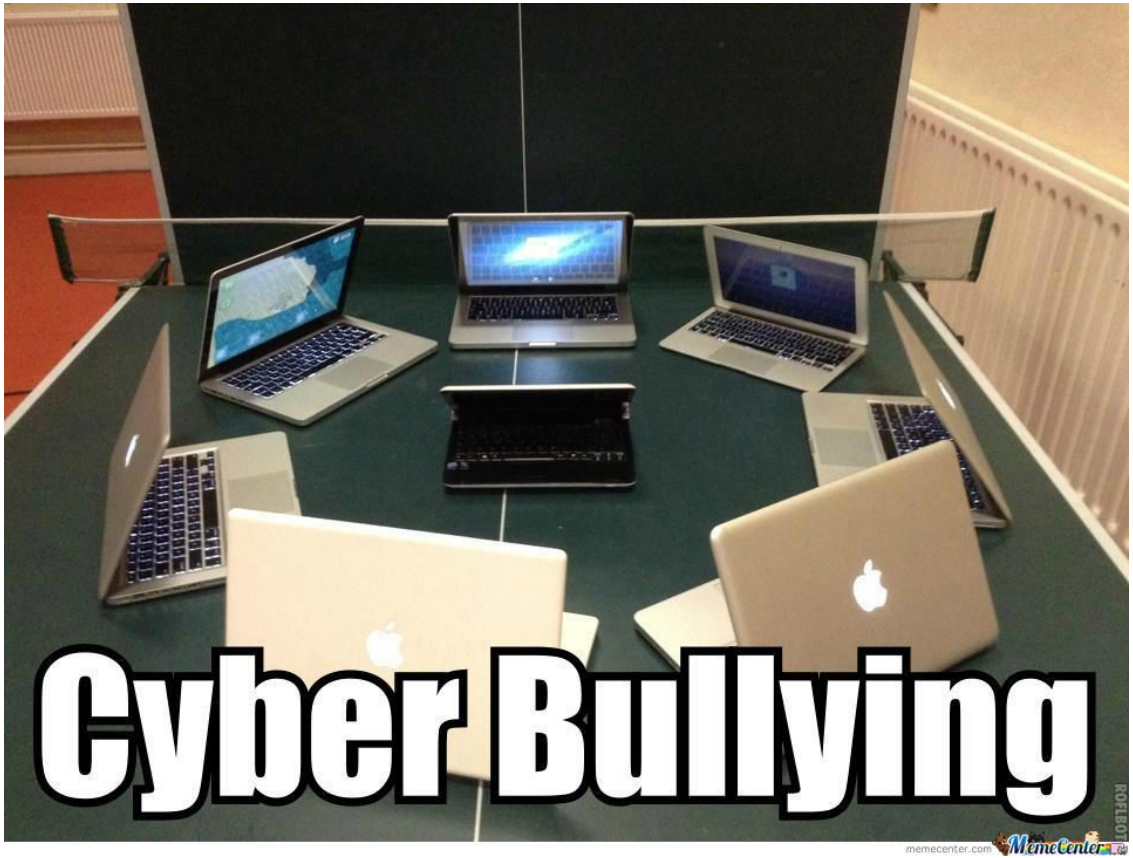
AFL also provides tools for minimizing a corpus; determining the smallest subset of provided input to maximize code coverage

File Fuzzing with AFL

Demo

File Fuzzing with AFL

Hands-On Lab



Network Fuzzing with BooFuzz (Sulley)

Network Fuzzing
Fuzzing Process

Fuzzing the code responsible for processing data received on a socket

BooFuzz
Fuzzing Process

Supports Serial, Ethernet, IP, TCP, HTTP… (support for full network stack)

Generation-based data

Fuzzing done in Python, tedious instrumentation handled by BooFuzz library

AngrySpider
Fuzzing Process

Start AngrySpider web server, poke around to see how it works

Use BooFuzz to fuzz the webserver - it will start up the process, monitor it for crashes, and deliver input data for you

Network Fuzzing with BooFuzz (Sulley)

Demo

Network Fuzzing with BooFuzz (Sulley)

Hands-On Lab

API Fuzzing with Bradamsa

API Fuzzing
Fuzzing Process

We can also fuzz web apps, including web apps with API interfaces

Generally going to be using a RESTful or SOAP style for the API interface

- RESTful API
  - GET /tvshows -- list of tv shows
  - GET /tvshows/silicon_valley -- detials about the “silicion valley” tv show
  - PUT /tvshows -- updating a tv show
  - POST /tvshows -- creating a new tv show
  - DELETE / tvshows -- deleting a tv show

Bradamsa
Fuzzing Process

Burp Suite + Radamsa === Bradamsa

- Radamsa - Simple Fuzzer
- Burp Suite - Web Intercepting Proxy for Web App Testing

API Fuzzing with Bradamsa

Demo

API Fuzzing with Bradamsa

Hands-On Lab

Thank you!

- CNO.io
  - https://github.com/cno-io/fuzzing_ftw