Below is the outline of the runtime of our two-day training which includes the theory and the hands-on exercises. We have calculated this based on the content shared below with approximate timing for each section highlighted as “coverage & duration”.

== Day 1 ==

- Agenda & Objective of the training
- Introduction
- Why fuzzing
- Types of fuzzing
- Introduction to Whitebox fuzzing
- Types of AFL
- Introduction to AFL (american fuzzy lop)
- Working of AFL
- Understanding the core principles of AFL
- Understanding AFL mutation
- AFL strategy
- AFL utilities
- Prerequisite and installation

Coverage & duration: The above section takes minimum of 1:30 hours in which we cover the theory aspect of our training. During the above one of the trainers would be distributing the public server info and will ask if any troubleshooting is required.

A 10-minute coffee break

- Fuzzing with STDIO
- Finding targets
- Fuzzing ubuntu packages
- Not a pro tip
- Resolving dependencies

Coverage & duration: The above section takes 1:45 hour to complete, which includes minimum of 3 exercises. However, in this version (2023) we have added more programs to fuzz.

- Smart Fuzzing
- Instrumenting Binaries
- AFL Instrumentation
- Input Generation for Fuzzing
- Radamsa – Test case generation
- Coverage Guided Fuzzing
- Test Case Minimization

Coverage & duration: The above section takes ~2 hours to complete which includes 6 different complex programs such as coreutils, VIM. In this version (2023) we have also added fuzzing sudo and sudoedit, exiv2, hermes.

- Corpus Optimization
- Crash triage
- Effective ways of crash triage
- Exploitable or not?

Coverage & duration: The above section takes 30 minutes which completes a bit of theory such as types of registers and fundamentals of debugging. The practical done over here would be continuation from the above fuzzed binaries.

A 40-minute lunch break

- Introduction to ASAN/MSAN
- LLVM Symbolizer
- Domain-Specific Fuzzing

Coverage & duration: The above section takes 30 minutes which covers the fundamentals of sanitizers and symbolizers. There are 2 exercises covered in this section.

- Comparing trace bitmap
- Difference in yields, path
- Utilizing grammar for Fuzzing
- Hooking custom libraries

Coverage & duration: The above section takes 45 minutes which covers how grammars can be used in fuzzing and attendees learn how to write their own grammar to fuzz a program. Attendees can pick any target here from GitHub or which is provided in VM.

- Symbolic Fuzzing
- Real world examples for symbolic fuzzing
- Difference in coverage guided and symbolic fuzzing

Coverage & duration: This section is new for 2023. It would be completed in 1 hour. This section would cover theory of what symbolic fuzzing is and how we can enable it in our existing AFL framework.

- Generating graphs
- Optimizing the fuzzing hierarchy
- Primary and Secondary technique

Coverage & duration: So, this is our miscellaneous section where we discuss about challenges we face during fuzzing and how to overcome it. This section takes 30 minutes, this year we have added few more tips.

- Day 1 Exercise (Homework)

Coverage & duration: Attendees can participate in this; they have to replicate a UAF bug in VIM which would be discussed in day 2.

== Day 2 ==

- RECAP – Day 1
- Yields from Exercise #1
- Triage analysis (2)

Coverage & duration: The above section would cover a brief recap of day 1 and would be completed in ~30 minutes.

- AFL Persistence
- What is AFL Persistence
- Implementation of AFL Persistence
- . Commit patching for fuzzing

Coverage & duration: This section would cover how attendees can enhance their fuzzing approach via AFL persistence and understanding the while loop statement in C. This section would be completed in 45 minutes. This has 3-4 exercises which includes PHP and VIM.

A 10-minute coffee break

- Introduction to Blackbox fuzzing
- Setting up QEMU
- Fuzzing blackbox binaries with QEMU
- Real world examples and case studies
- Fuzzing stripped v/s non-stripped binaries
- QEMU and Address Sanitizers
- QEMU Persistence mode
- Fuzzing with nyx mode
- Snapshot Fuzzing in Nyx

Coverage & duration: This section would cover the fundamentals of blackbox fuzzing and how QEMU works. We also fuzz binaries such as busybox, coreutils and tcpdump and proceed to QEMU persistence mode. This section will take ~2:30 hours.

- Introduction to ARM
- Cross platform fuzzing

Coverage & duration: This section is new for 2023. It would be completed in ~45 minutes. We introduce the concept of cross platform fuzzing and see it in action. This includes 3-4 ARM based binaries as a part of the exercises.

A 40-minute lunch break

- Setting up WinAFL
- Corpus Utilization
- Corpus Mutation
- Fuzzing windows binaries
- Utilizing symbols for binaries
- Jackalope Internals

Coverage & duration: Here we switch the flavor from Linux to Windows. We have a look at how fuzzing works on Windows, we introduce WinAFL along with a few additional components such as DynamoRIO. Finally, we dive into fuzzing binaries for Windows x64 and x86 platforms. This section takes 2 hours.

- Overview of different fuzzers (Hongfuzz, Fuzzli and Grizzly)

Coverage & duration: Another new topic that we have introduced for our training this year. We have a look at how different fuzzing frameworks work in brief. This gives a comparison and a new outlook of how different fuzzing frameworks operate. This section will take 1 hour.

- Miscellaneous exercise
- Synchronization fuzzing
- Fuzzing VIM
- Regex engine
- Fuzzing OpenSSH x509 Parser
- Fuzzing cURL
- Fuzzing PuTTY
- Fuzzing PHP
- Serialize functions

Coverage & duration: This is the most interesting part in the course where we introduce fuzzing complex binaries. An attendee will emulate the techniques learnt so far and get their hands on fuzzing with exploring their own ways to fuzz. This section will take 1 hour.

- Fuzzing browser engines
- Best practices for fuzzing

Coverage & duration: We further dive into fuzzing browser engines such as JavaScriptCore & Chakra Core. This section will take 45 minutes.

- Utilizing slack webhooks for continuous fuzzing stats

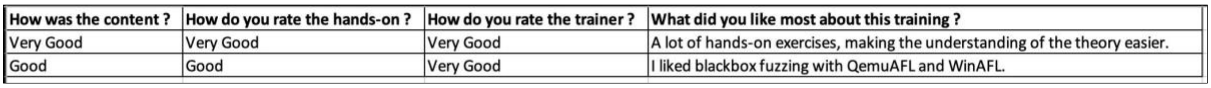
Coverage & duration: This section will take 15 minutes.

- CTC – Capture the crash – Exercise

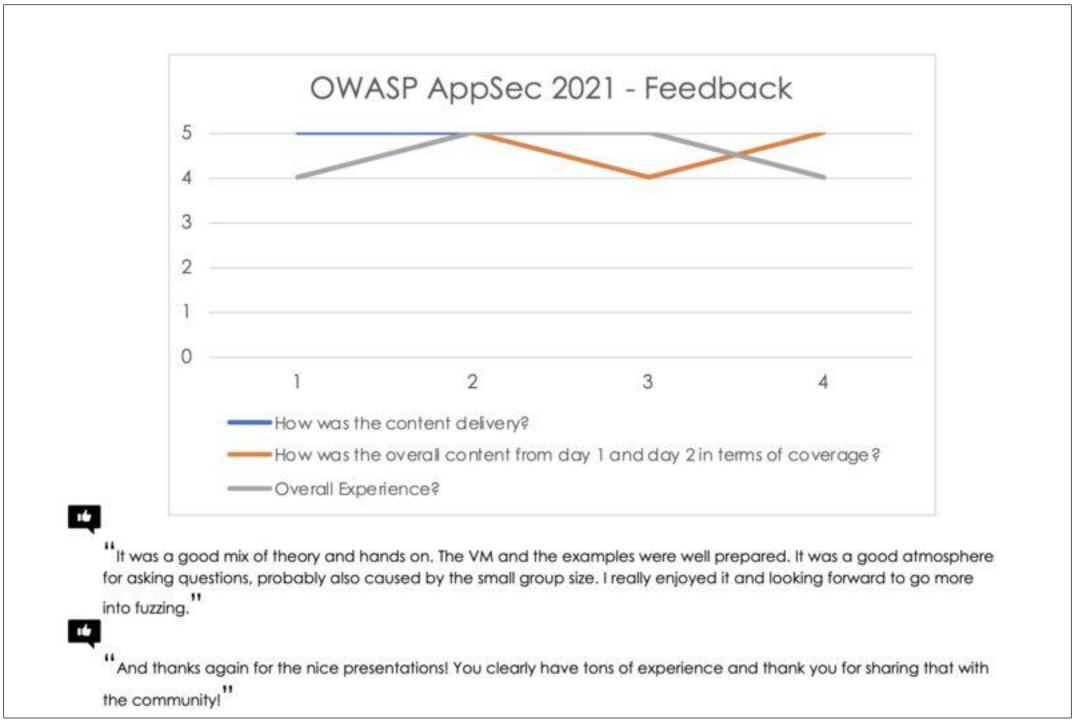
Coverage & duration: A custom binary is provided to the attendees to fuzz and identify crashes. This exercise will take 1 hour.

- Q/A & Feedback (15 minutes)

Additionally: Previous trainings feedback received (BlackHat EU, BruCON & OWASP AppSec)

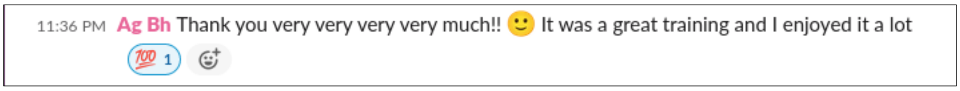


How was the content ?	How do you rate the hands-on ?	How do you rate the trainer ?	What did you like most about this training ?
Very Good	Very Good	Very Good	A lot of hands-on exercises, making the understanding of the theory easier.
Good	Good	Very Good	I liked blackbox fuzzing with QemuAFL and WinAFL.



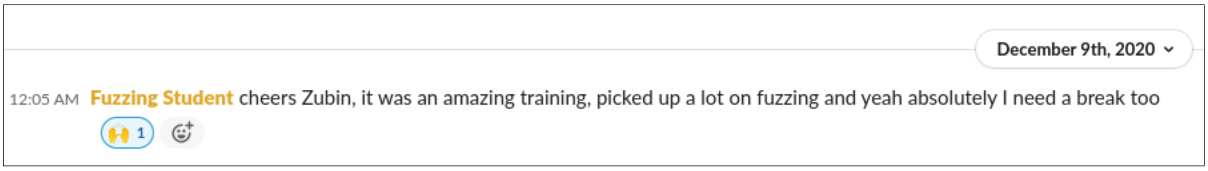
"It was a good mix of theory and hands on. The VM and the examples were well prepared. It was a good atmosphere for asking questions, probably also caused by the small group size. I really enjoyed it and looking forward to go more into fuzzing."

"And thanks again for the nice presentations! You clearly have tons of experience and thank you for sharing that with the community!"



11:36 PM **Ag Bh** Thank you very very very very much!! 🙂 It was a great training and I enjoyed it a lot

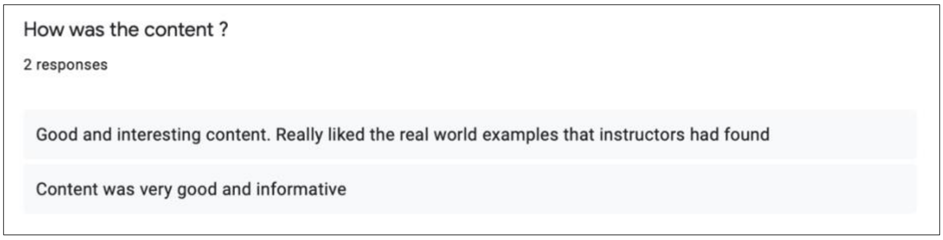
💯 1



December 9th, 2020

12:05 AM **Fuzzing Student** cheers Zubin, it was an amazing training, picked up a lot on fuzzing and yeah absolutely I need a break too

🙌 1

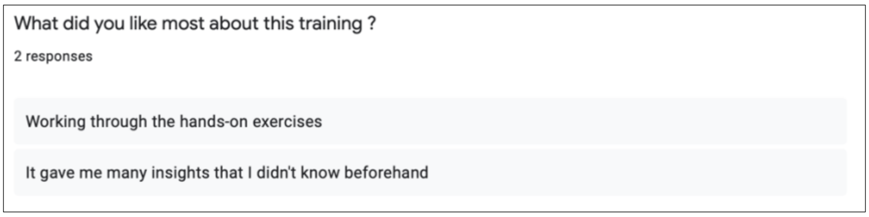


How was the content ?

2 responses

Good and interesting content. Really liked the real world examples that instructors had found

Content was very good and informative



What did you like most about this training ?

2 responses

Working through the hands-on exercises

It gave me many insights that I didn't know beforehand

“It gave me many insights that I didn't know beforehand”

“It was an amazing training, picked up a lot on fuzzing”

“A lot of hands-on exercises, making the understanding of the theory easier.”

“Content was very thorough and touched all kinds different aspects of the subject"

“Nice Presentations! You clearly have tons of experience and thank you for sharing that”

“Good and interesting content. Really liked the real world examples that instructors had found”

“The content was very well organised, and I really appreciated the runbook as well as the clarity of the presentations.”

“hands-on activies and coverage on different areas of fuzzing that have initial steep learning curve”

“It was a good mix of theory and hands on. The VM and the examples were well prepared. It was a good atmosphere for asking questions, probably also caused by the small group size. I really enjoyed it and looking forward to go more into fuzzing”

“Very informative”

“For everything that was taught, there was a hands-on exercise.”

“Really great. Started from basic & went on to advanced fuzzing techniques.”

“The content was great and I have learned lots of new things regarding fuzzing i had wonderful experience thanks to Dhiraj and Zubin for presenting a tremendous journey to fuzzing.”

“I appreciated how you guys explained the complex concepts in a simple and easy-to-understand manner. It was an awesome experience and something new for me. I would say kudos to both of you for providing this learning opportunity :)”