Lab Security in Distributed Systems

MA-INF 3320

Application

In order to get a topic, you will need to submit your expose until 08.10.2019. Please send it to the respective supervisors. You will be informed whether you were chosen/not chosen for your favourite lab topic until 15.10.2019. Multiple applications are allowed for up to three topics.

Deadline for expose: 08.10.2019

Deadline for registration in BASIS: 31.10.2019

Getting the most out of fuzzing

Supervisor

Klaus Tulbure
tulburek@iai.uni-bonn.de

Getting the most out of fuzzing

Coverage-based fuzzing is a state-of-the-art analysis for finding bugs and vulnerabilities in software. It can be described as an "intelligent" brute-force analysis that penetrates a program with input to, hopefully, cause a crash. The inputs are not random but are generated using an evolutionary algorithm with the code coverage of each run as the fitness function.

However, code coverage alone has its limits, e.g., it cannot fully describe the inputs' effects on the program flow. Hence, gathering additional information from the fuzzing might lead to further insights in the code and thereby to new bugs.

Getting the most out of fuzzing

Your task will be to gather and evaluate the interesting information from the fuzzing process, such as the program and data flow, coverage, crash information, memory usage, multi-threading, etc.

Afterwards, the collected data needs to be processed and stored. For that, suitable techniques from Algorithmics, Machine Learning and Artificial Intelligence can be used.

Requirements are programming skills in

- C/C++ to collect the data from the fuzzer
- a common language of your choice for the processing

Getting the most out of fuzzing

Literature

Klees et al. understandably explained the overall concept of coverage-based fuzzing. They evaluated fuzzers from recent years, compared the techniques used and also mentioned several weakness of the coverage-based approaches.

Klees, George, et al. "Evaluating fuzz testing." Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2018.

Usability of fuzzers

Supervisor

Stephan Plöger
ploegers@cs.uni-bonn.de

Usability of fuzzers

The idea is to conduct a study where participants solve different fuzzing tasks with different fuzzers to determine the usability of those fuzzers.

Tasks to solve are and are not limited to: Which fuzzers to use - which programs/tasks to use - type of participants - type of study.

A key problem is to find appropriate tasks and participants.

Requirements are programming skills in

- C/C++ to be able to work with the most common fuzzers
- Knowledge about study design (USECAP lecture)
- Knowing what a fuzzer is is helpful

Usability study about fixing bugs (fuzzing)

Supervisor

Stephan Plöger
ploegers@cs.uni-bonn.de

Usability study about fixing bugs (fuzzing)

The idea is to conduct a study where participants are confronted with the output of different fuzzers and asked to find and fix the corresponding bug to determine the usability of the output.

Tasks to solve are and are not limited to: Which fuzzers to use - which programs/tasks to use - type of participants - type of study.

A key problem is to find appropriate tasks and participants.

Requirements are programming skills in

- C/C++ to be able to work with the most common fuzzers
- Knowledge about study design (USECAP lecture)
- Knowing what a fuzzer is is helpful

CTF Tasks

Supervisor

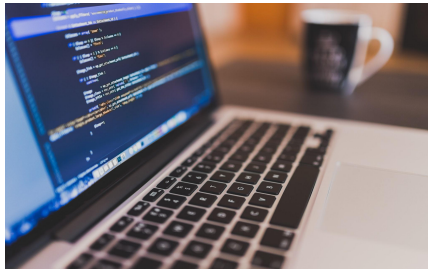
Mischa Meier
meierm@cs.uni-bonn.de

CTF Tasks

The idea is to create tasks which can be used in a Capture the Flag(CTF) event. CTFs are information security competitions which can also be used in education.

Tasks could be and are not limited to: Web-security, binary exploitation, cryptography, reversing

The key problem is to find appropriate tasks, implement them and verify the attacks.



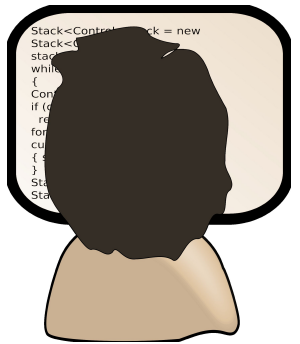
Requirements:

- Participated in CTFs before
- Knowledge about security

Usability Investigation of IDS Systems

Supervisor

Anastasia Danilova
danilova@cs.uni-bonn.de



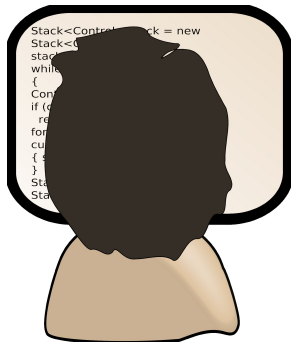
IDS Systems - Usability?

The idea is to test current intrusion detection systems (IDS) for usability.

Your task will be to test IDS's that you are familiar with for usability. First, with a heuristic evaluation, then perhaps with a small usability study. Alternatively you can try to improve the system and evaluate your prototype.

Requirements:

- Past experience with at least one IDS system
- Good technical skills and the skill to work autonomously
- Willingness to work yourself into the systems, set them up, and test them for usability



Own ideas?

Supervisors

all of us

send your ideas to
Klaus Tulbure
tulburek@iai.uni-bonn.de

Own ideas?

You've got some **own ideas** for a project in Usable Security and Privacy?

GREAT!

We'd love to work with you on that!