SINGAPORE CYBERSECURITY CONSORTIUM (SGCSC)
PRESENTS

CYBERSECURITY CAMP@SINGAPORE 2017

In conjunction with SG-CRC 2017

19-20 FEBRUARY 2017

SCHOOL OF COMPUTING,
NATIONAL UNIVERSITY OF SINGAPORE (NUS)

This 2-day workshop will cover cutting-edge technologies in cybersecurity on selected topics. Students will learn both the theoretical foundation as well as practical techniques and solutions to solve real-world problems. Students will also have an opportunity to gain hands-on experience through lab sessions. Through this workshop, the students will acquire advance knowledge of cybersecurity which will help to guide them in their research and professional life towards cybersecurity.

Topic 1: Deep Learning and Cyber-security

Chair: Prof. Dawn Song, U C Berkeley

Topic 2: Fuzz Testing for Finding Vulnerabilities

Chairs: Assoc. Prof. Zhenkai Liang and Prof. Abhik Roychoudhury, NUS

TECHNICAL LEADS

Prof. Dawn Song, U C Berkeley
Prof. Abhik Roychoudhury, NUS

IMPORTANT DATES

Application Deadline: **25 November 2016**
Acceptance Notification: **15 December 2016**

CAMP FEES

	Students	**Non-Students**
Full Camp: Lessons + Hackathon	S$ 150	S$ 350
Lessons Only (19 Feb 2017)	S$ 80	S$ 150

Fees are payable only upon acceptance of application.

Travel support grant is available to selected international students attending the Full Camp. Details will be provided to accepted applicants.

APPLICATION

Interested candidates may apply online at: https://mysoc.nus.edu.sg/~appln

CONTACT

Dr. Vivy Suhendra, Executive Director, Singapore Cybersecurity Consortium
vivy@comp.nus.edu.sg

DEEP LEARNING AND CYBER SECURITY

MOTIVATION AND GOALS

Deep learning has made huge advances and impact in many areas of computer science such as vision, speech, NLP, and Robotics. Many exciting research questions lie in the intersection of security and deep learning.

FIRST, how will these deep learning systems behave in the presence of adversaries? Research has shown that many of the state-of-the-art deep learning systems can be easily fooled by adversarial examples. We will explore fundamental questions in this area including what types of attacks are possible on deep learning systems, why they exist, and how we can defend against them.

SECOND, how can deep learning techniques help security applications? We will explore this area and study example security applications using deep learning techniques including program binary analysis, password security analysis, malware detection and fraud detection

FUZZING SOFTWARE VULNERABILITIES – CATCH ME IF YOU CAN!

BACKGROUND

Fuzz testing is a fully automated software testing technique where randomly generated inputs are fed to a program with the explicit goal of crashing the program. Fuzz testing can be employed on program binaries, and can benefit from an input format specification, or from the presence of sample seed program inputs. Application of fuzzing in vulnerability detection is common, and it constitutes an important technique to enhance software security.

PLAN FOR THE TOPICS

In this tutorial, we will first distinguish between **fuzzing** and **usual program testing** - by clarifying the weak oracles (or expected behavior) needed in fuzzing. We will then distinguish between **generation based fuzzing** which use input format specifications and **mutation based fuzzing** which modifies input seeds.

We will also clearly show the differences between blackbox, greybox and whitebox fuzzing. **Blackbox fuzzing** does not assume any view of the program, while **greybox fuzzing** only distinguishes different program paths executed by different inputs. The main advantage of these techniques is the ability to avoid extracting control flow from program binaries, which can be notoriously difficult. In comparison, **whitebox fuzzing** assumes knowledge of the program control flow (even if shown at binary level). On the other hand, it can achieve potentially better coverage of the program behavior by using a program execution technique called symbolic execution. We will also cover the foundations of **symbolic execution** and its use in whitebox fuzzing in this tutorial.

HANDS-ON COMPONENTS

All of the concepts covered in the tutorial will be demonstrated via hands-on usage of blackbox, greybox and whitebox fuzzing tools. The tool understanding and usage will culminate in an exciting **HACKATHON** which will challenge the students to hunt seeded as well as real vulnerabilities in binaries of tools which they would have widely used daily, either as file processing programs or as command line utilities.