Speakers presentation info

When Are Opaque Predicates Useful?

Speaker: Dr. Gio Russello
University of Auckland

Opaque predicates are a commonly used technique in program obfuscation, intended to add complexity to control flow and to insert dummy code or watermarks. However, there are many attacks known to detect opaque predicates and remove dummy code. We survey these attacks and argue that many types of programs cannot be securely obfuscated using opaque predicates. In particular we explain that most previous works on control flow obfuscation have introduced predicates that are easily distinguished from naturally occurring predicates in code, and hence easily removed by an attacker. We state two conditions that are necessary for a program to be suitable for control flow obfuscation. We give an integrated approach to control flow obfuscation that simultaneously obfuscates real predicates and introduces opaque predicates. The opaque predicates are indistinguishable from the obfuscated real predicates in the program. If an attacker applies the usual approaches (both static and dynamic) to identify and remove opaque predicates then they are likely to remove critical functionality and introduce errors. We have implemented our obfuscator in LLVM. We provide an analysis of the performance of the resulting obfuscated code.

Speaker Bio:
Gio Russello is an Associate Professor with the Computer Science Department, at the University of Auckland. His research focuses on Android security, confidentiality and privacy solutions for the cloud, and access control models in general. He leads the SECRET Lab, where together with his group, they work on several security projects including the MBIE STRATUS project. In the past, he was founding CEO of a start-up developing secure solutions for Android.

Virtual reality art-making for stroke rehabilitation: Field study and technology probe

Speaker: Dr. Danny Lottridge
University of Auckland

How can we better understand the process of therapeutic art-making for stroke rehabilitation, and what are design opportunities for virtual reality art-making for people with stroke-related impairments? We investigated this question in a two-part study with 14 amateur artists with disabilities resulting from stroke: a three-week field study and a technology probe consisting of experiential virtual reality interviews. We uncovered what participants made, the aesthetics of the materials and the process of making. The field study revealed inspirations around identity, situatedness of choices for tools in the social and physical environment, and a breadth of application techniques (e.g., dripping paint or use of tape) that varied in need for fine motor control. The experiential virtual reality interviews highlighted the need for control, the affordances of the medium, and the challenges in viewing and reflecting on work. Emergent art reflected qualities of the 3D paint and free-form gesture. Virtual reality and traditional art-making contrasted in the speed and finality of application, opportunities for iteration and reflection, and in the need for dexterity. We discuss strengths, weaknesses and implications for design of virtual reality art-making for those with stroke-related impairments.

Speaker Bio:
Danny is a Senior Lecturer in the Department of Computer Science at the University of Auckland, New Zealand. The senior lecturer title is equivalent to associate professor in North American universities. Her research areas are in human computer interaction and human factors engineering, with interests in multitasking, user behavior and the design of mass-market and health technologies. She is an official academic collaborator with Yahoo/Oath.

Before starting at the University of Auckland, Danny was a Postdoctoral Fellow at Stanford University. She obtained a PhD in Human Factors Engineering from University of Toronto.

Detecting Concept Drift In Medical Triage

Speaker: AProf. Yun Sing Koh
University of Auckland

in their accompanying referral documents, which contain a mix of free text and structured data. By training a model to predict triage decisions from these referral documents, we can partially automate the triage process, resulting in more efficient and systematic triage decisions. One of the difficulties of this task is maintaining robustness against changes in triage priorities due to changes in policy, funding, staff, or other factors. This is reflected as changes in relationship between document features and triage labels, also known as concept drift. These changes must be detected so that the model can be retrained to reflect the new environment. We introduce a new concept drift detection algorithm for this domain called calibrated drift detection method (CDDM). We evaluated CDDM on benchmark and synthetic medical triage datasets, and find it competitive with state-of-the-art detectors, while also being less prone to false positives from feature drift.

Speaker Bio:
Yun Sing Koh is an Associate Professor at the School of Computer Science, The University of Auckland, New Zealand. Her research is in the area of machine learning. Within the broad research realm, she is currently focusing on three strands of research: data stream mining, lifelong and transfer learning, and pattern mining. She has published more than 100 research papers in this field at top venues.

She has been active in the research community including serving as the General Chair at the IEEE International Conference on Data Mining 2021, Workshop Chair at the ECML 2021, Program Co-Chair of the Australasian Data Mining Conference 2018 and as the Workshop Chair for the 15th Pacific-Asia Conference on Knowledge Discovery and Data Mining.

Agile software development: practices, self-organization, and satisfaction

Speaker: Prof. Robert Biddle
University of Auckland

This chapter deals with the Manifesto's principle of self-organizing teams. In recent work, the authors examine the state of practice using data from a study of software professionals in Switzerland, especially addressing the issue of overall satisfaction. The chapter reveals that the most striking correlation to satisfaction is the level of adoption of self-managing teams, whereas the strongest hindrances to satisfaction are a lack of ability to change the organizational culture and lack of management support. The analysis shows that technical and collaborative practices were related to self-organization and satisfaction, but were not able to explain satisfaction by themselves. Even with strong technical and collaborative practices, however, satisfaction is not assured, demonstrating that goals of creating timely and successful products and services matter.

Speaker Bio:
Robert Biddle is Professor of Computer Science and Cognitive Science at Carleton University in Ottawa, Canada. He has degrees in Mathematics, Computer Science, and Education, and has worked in industry, education, and research. He was born in the UK, educated in Canada, and lived for many years in New Zealand, working at Victoria University of Wellington. His research has always concentrated on human aspects of software, including programming language design and software development process, and more recently on computer security. He has awards for research, teaching, and graduate mentorship. Robert is a Fellow of the New Zealand Computer Society, and a Commonwealth Scholar.

Conceptual complexity of neural networks

Speaker: Prof. Brand McCane
University of Otago

We propose a complexity measure of a neural network mapping function based on the order and diversity of the set of tangent spaces from different inputs. Treating each tangent space as a linear PAC concept we use an entropy-based measure of the bundle of concepts to estimate the conceptual capacity of the network. The theoretical maximal capacity of a ReLU network is equivalent to the number of its neurons. In practice, however, due to correlations between neuron activities within the network, the actual capacity can be remarkably small, even for very big networks. We formulate a new measure of conceptual complexity by normalising the capacity of the network by the degree of separation of concepts related to different classes. Empirical evaluations show that this new measure is correlated with the generalisation capabilities of the corresponding network. It captures the effective, as opposed to the theoretical, complexity of the network function. We also showcase some uses of the proposed measures for analysis and comparison of trained neural network models.

Speaker Bio:
My research interests include computer vision, pattern recognition, machine learning, biomedical imaging, and robotics. My current research focuses on theoretical understanding of the effectiveness of deep networks, and self-learning for robots.

I also have an interest in computer graphics and participate in the computer graphics group here at the University of Otago.

My background is in Computer Science and I completed my undergraduate studies and PhD at James Cook University of North Queensland in Australia. My PhD was entitled "Learning to Recognise 3D Objects from 2D Intensity Images", which I completed in February 1996. I then held a temporary position as a lecturer at James Cook University, before taking up a position in February 1997 as a lecturer with the Computer Science Department here at Otago University.

Sizing domestic batteries for load smoothing and peak shaving based on real-world demand data

Speaker: AProf. Davis Eyers
University of Otago

Distributed battery energy storage provides a potential system-wide solution to issues of increasing variability in electricity supply and demand. In this research, we take a demand-driven approach to determining residential battery capacity based on a detailed analysis of measured time series (with per-minute resolution) of individual household demand. We consider two modes of battery operation: (i) load smoothing around the average and (ii) peak shaving, where the battery ensures grid power demand does not exceed a set threshold. We determine the battery capacity (in kWh) required for each mode of operation based on an individual household's demand patterns—independent of specific battery characteristics. In addition, we also compare the battery capacity required for the individual houses with that required for the aggregated demand of a collection of households. Our results show that the battery capacity requirements for these modes of operation can vary by more than 10 kWh per house and have a large seasonal variation. They also show that aggregation reduces the per-house battery requirements by 50% for load smoothing and 90% for peak shaving. These results have important implications for battery deployment strategies. In particular, they show that coordinated battery deployment at the street or building complex level is likely to have significant economic benefits.

Speaker Bio:
Before joining the University of Otago, I worked as a senior research associate at the University of Cambridge, from where I was awarded my PhD. My undergraduate degrees (Computer Engineering and Pure Maths) are from UNSW in Sydney, Australia.

My recent research has examined security enforcement and data dissemination mechanisms within wide-area distributed systems. In particular, I have worked with event-based middleware, role-based access control, decentralised information flow control, and try to develop connections between these technologies. My research is of growing importance to cloud and grid computing: large-scale public services, such as electronic health record repositories, must manage sensitive data in a secure manner.

A Three-Year Study on Peer Evaluation in a Software Engineering Project Course

Speaker: Dr. Moffat Matt
University of Canterbury

Background: Peer evaluation in software engineering (SE) project courses enhances the learning experience of students. It also helps instructors monitor and assess both teams and individual students. Peer evaluations might influence the way individual students and teams work; therefore, the quality of the peer evaluations should be tracked through the project course. Contribution: In this article, we analyzed the quality and scoring behavior of students in peer evaluation in an undergraduate SE project course over three years. Research Questions: RQ1: What is the quality of peer evaluation of undergraduate students in a SE project course? RQ2: How do undergraduate students in an SE project course score each other? Methodology: The quality of peer evaluation (length, level of detail, etc.) and scoring of peers based on various aspects of peer evaluations of third-year students in a year-long SE project course were studied. Taking into account the grade students received at the end of the course (A, B, C, and F-calibers), peer evaluations were categorized, analyzed over time, and compared between students calibers. Findings: After analyzing 6854 peer evaluations from 193 students, it was found that the quality of peer evaluations across students was mostly consistent throughout the course. Also, it was observed that quantitative aspects of the peer evaluation were scored similarly across student calibers. However, the qualitative aspects of the peer evaluation were impacted by the caliber of students. These findings suggest that weaker students (i.e., C-caliber students) generally receive better quality peer evaluations than stronger students (i.e., A-caliber students). Finally, a preliminary analysis showed a positive connotation of emotions and sentiments found in the textual feedback delivered by students.

Speaker Bio:
I am working in the area of Artificial Intelligence in Education (AIED). This means that I am interested in AI, Education (especially teaching strategies), Psychology (particularly to do with learning and behaviour), and of course Computer Science & Software Engineering.

Our research group focuses on building intelligent and adaptive educational systems – systems that allow the student to explore and learn while receiving customised guidance. These systems are intelligent enough to

- know about a particular domain semantically and syntactically (domain modelling),
- know and reason about each student (student modelling) using their current knowledge on each of the domain concepts and the history of their performance in the system,
- provide adaptive (customised) guidance to each student,
- allow the student to see a visualisation of their student model, and
- optionally allow the student to negotiate their student model (argue for a change in their student model).

A Lightweight Formalism for Reference Lifetimes and Borrowing in Rust

Speaker: Dr. David Pearce
Victoria University of Wellington

Rust is a relatively new programming language that has gained significant traction since its v1.0 release in 2015. Rust aims to be a systems language that competes with C/C++. A claimed advantage of Rust is a strong focus on memory safety without garbage collection. This is primarily achieved through two concepts, namely, reference lifetimes and borrowing. Both of these are well-known ideas stemming from the literature on region-based memory management and linearity/uniqueness. Rust brings both of these ideas together to form a coherent programming model. Furthermore, Rust has a strong focus on stack-allocated data and, like C/C++ but unlike Java, permits references to local variables.
Type checking in Rust can be viewed as a two-phase process: First, a traditional type checker operates in a flow-insensitive fashion; second, a borrow checker enforces an ownership invariant using a flow-sensitive analysis. In this article, we present a lightweight formalism that captures these two phases using a flow-sensitive type system that enforces "type and borrow safety." In particular, programs that are type and borrow safe will not attempt to dereference dangling pointers. Our calculus core captures many aspects of Rust, including copy- and move-semantics, mutable borrowing, reborrowing, partial moves, and lifetimes. In particular, it remains sufficiently lightweight to be easily digested and understood and, we argue, still captures the salient aspects of reference lifetimes and borrowing. Furthermore, extensions to the core can easily add more complex features (e.g., control-flow, tuples, method invocation). We provide a soundness proof to verify our key claims of the calculus. We also provide a reference implementation in Java with which we have model checked our calculus using over 500B input programs. We have also fuzz tested the Rust compiler using our calculus against 2B programs and, to date, found one confirmed compiler bug and several other possible issues.

Speaker Bio:
David graduated with a PhD from Imperial College London in 2005, and currently holds a full-time position at Victoria University of Wellington, NZ. During his PhD thesis, David developed several algorithms for static analysis of C programs which were later incorporated into the GNU C Compiler (GCC). Since then, David has worked on a wide range of topics in programming languages, compilers and static analysis. For example, an algorithm of his is used in the TensorFlow library for machine learning developed by Google. Another is part of the widely-used Mathematica application, and yet another is part of the SciPy library for scientific computing in Python.

Since 2009, David has been the technical lead on a large open source project to develop a verifying compiler. The system, called Whiley (see http://whiley.org), is designed specifically to simplify the process of verifying that programs are free from bugs. The system has currently been used for teaching in New Zealand, Australia, China and South Africa. It has also been used for verifying algorithms at Amazon and has formed the basis for more than twenty undergraduate research projects, and one MSc and one PhD thesis.