THE SIXTEENTH ANNUAL

HIGH CONFIDENCE SOFTWARE AND SYSTEMS CONFERENCE

Annapolis, MD | May 10-12, 2016
http://cps-hcss.org

HCSS

The Sixteenth Annual

HIGH CONFIDENCE SOFTWARE AND SYSTEMS CONFERENCE

http://cps-hcss.org

Annapolis, Maryland, USA | May 10-12, 2016

WELCOME MESSAGE

Kathleen Fisher
Tufts University

Stephen Magill
Galois, Inc.

The Co-Chairs, NITRD HCSS Coordinating Group, and steering committee are pleased to welcome you to the 16th annual High Confidence Software and Systems (HCSS) Conference being held again this year at the Historic Inns of Annapolis in Annapolis, Maryland.

This year's program continues the tradition of excellence over the sixteen-year history of the Conference. A host of world class research scientists representing academia, industry, and Government will deliver a range of experience and technical talks. These presentations will provide new scientific and technological foundations that can enable entirely new generations of engineered systems – systems that are essential for effectively operating life-, safety-, security-, and mission-critical operations. New foundations in science, technology, and advanced practice continue to be needed to build these systems, where computing, communication, information, and control are pervasively embedded at all levels. Talks will be focused on the themes of Measuring Security, Proofs that Cross IP Boundaries, Programming and Reasoning with Uncertainty, and Verification of Autonomous and Adaptive Systems. These themes and other topics will also be depicted through technical poster displays at this year's poster session.

We are pleased to host the Software Certification Consortium (SCC) meeting again this year. Formed in 2007, the SCC comprises industry researchers, government regulators, and academicians whose goal is to understand certification issues with respect to systems that contain significant software components (e.g., aerospace, automotive, medical devices, nuclear, defense, etc.), and to objectively make recommendations on processes and standards that impact the certification of such systems.

We hope that you will find the 2016 Conference as stimulating and informational as in years past. We greatly appreciate your attendance, and look forward to your continued participation and support of future conferences.

TABLE OF CONTENTS

GENERAL INFORMATION

REGISTRATION

Registration will be in the state lobby of the Governor Calvert House and will be open:

7:00 a.m. to 6:00 p.m. Monday
8:00 a.m. to 4:30 p.m. Tuesday
8:00 a.m. to 4:30 p.m. Wednesday
8:00 a.m. to 4:00 p.m. Thursday

WIRELESS INTERNET CONNECTION

A wireless internet connection will be available in the Governor Calvert Ballroom and Atrium. The network name is: Governor Calvert. The username and password are both "spring".

POSTER PRESENTATIONS

Poster sessions will be held between 2:30 p.m. and 3:30 p.m. on Tuesday, May 10 and Wednesday, May 11 in the atrium of the Governor Calvert House. Posters will be set up for display by the conference staff. Presenters can drop off their posters at the registration desk by noon on Tuesday, May 10.

CONFERENCE PRESENTATIONS

Conference presentations and posters will be available online at http://cps-hcss.org.

HOTEL PARKING

Parking at the Historic Inns of Annapolis is by valet only. A reduced parking rate has been negotiated for daily conference attendees, both with in and out privileges. This reduced rate is $12/day. Daily parking for *local government attendees* is complimentary with approved government ID. Government attendees should visit the registration table each day to have your parking validated or the full fee may apply.

SURVEY

Please take a moment to respond to our short survey at: http://cps-vo.org/group/hcss2016/survey. Your valuable feedback will help us plan future conferences.

CONFERENCE ORGANIZATION

PROGRAM CO-CHAIRS

Kathleen Fisher, Tufts University

Stephen Magill, Galois, Inc.

STEERING COMMITTEE

John Hatcliff, Kansas State University

John Launchbury, DARPA

Brad Martin, National Security Agency

Ray Richards, DARPA

Bill Scherlis, Carnegie Mellon University

Frank Seaton Taylor, National Security Agency

MEETING ORGANIZERS

Katie Dey, Vanderbilt University

Anne Dyson, Innovative Analytics

GRAPHIC DESIGN

Amy Karns, Vanderbilt University

SPONSOR AGENCY

NITRD HCSS Coordinating Group

PROGRAM AGENDA

TUESDAY, MAY 10

PROGRAM AGENDA

TIME	TITLE	SPEAKER	PAGE
0800 - 1630	**REGISTRATION**		
0800 - 0900	**BREAKFAST**		
0900 - 1000	**KEYNOTE PRESENTATION** Verification Across Intellectual Property Boundaries	*Sagar Chaki (CMU-SEI)*	11
1000 - 1030	The Science of Deep Specification	*Benjamin Pierce (University of Pennsylvania)*	12
1030 - 1100	**REFRESHMENTS**		
1100 - 1130	Interrupts in OS code: let's reason about them. Yes, this means concurrency.	*June Andronick (Data61 | CSIRO and UNSW)*	14
1130 - 1200	Android Platform Modeling and Android App Verification in the ACL2 Theorem Prover	*Eric W. Smith (Kestrel Institute)*	16
1200 - 1330	**LUNCH (ON YOUR OWN)**		
1330 - 1400	"It's QEDs All the Way Down"	*David S. Hardin (Rockwell Collins)*	18
1400 - 1430	Moving Hardware from "Security through Obscurity" to "Secure by Design"	*Ryan Kastner (UCSD)*	20
1430 - 1530	**POSTER SESSION AND REFRESHMENTS**		50
1530 - 1600	Learning State-Rich Specifications from "Big Code"	*Swarat Chaudhuri (Rice University)*	22
1600 - 1630	A String, Regular Expression, and Integer Solver for Bug-finding and Security	*Yunhui Zheng (IBM)*	24
1630	**ADJOURN FOR THE DAY**		

PROGRAM AGENDA

WEDNESDAY, MAY 11

TIME	TITLE	SPEAKER	PAGE
0800 - 1630	**REGISTRATION**		
0800 - 0900	**BREAKFAST**		
0900 - 1000	**KEYNOTE PRESENTATION** Metric or English: Characterizing and Conveying Trustworthiness	*Fred B. Schneider* *(Cornell University)*	26
1000 - 1030	Build It Break It Fix It: Measuring Secure Development	*Andrew Ruef* *(University of Maryland)*	27
1030 - 1100	**REFRESHMENTS**		
1100 - 1130	Measuring Protocol Strength with Security Goals	*Paul Rowe* *(The MITRE Corporation)*	29
1130 - 1200	Combinatorial Coverage Analysis of Subsets of the TLS Cipher Suite Registry	*Dimitris Simos* *(SBA Research)*	30
1200 - 1330	**LUNCH (ON YOUR OWN)**		
1330 - 1400	New Perspectives on Automated Vulnerability Discovery	*Artem Dinaburg* *(Trail of Bits)*	31
1400 - 1430	Gradual Information Flow Control	*Peter Thiemann* *(Universität Freiburg)*	32
1430 - 1530	**POSTER SESSION AND REFRESHMENTS**		50
1530 - 1600	TQL-1 Qualification of a Model-Based Code Generator	*S. Tucker Taft* *(AdaCore)*	34
1600 - 1630	Collaboration and Automation for Threat Assessment and Mitigation	*David Archer* *(Galois, Inc.)*	35
1630	**ADJOURN FOR THE DAY**		
1830	**CONFERENCE DINNER** | Chart House Annapolis 300 2nd Street | Annapolis, Maryland 21403		74

THURSDAY, MAY 12

PROGRAM AGENDA

TIME	TITLE	SPEAKER	PAGE
0800 - 1600	**REGISTRATION**		
0800 - 0900	**BREAKFAST**		
0900 - 1000	**KEYNOTE PRESENTATION** Programming Uncertain <T>hings	*Kathryn McKinley* *(Microsoft Research)*	37
1000 - 1030	Formal Verification of C Programs with Floating-Point Computations: Certified Error Bounds for Signal Processing	*Tahina Ramananandro* *(Reservoir Labs, Inc.)*	38
1030 - 1100	**REFRESHMENTS**		
1100 - 1130	Wigmore: A Constraint-Based Language for Reasoning About Evidence and Uncertainty	*David Burke* *(Galois, Inc.)*	41
1130 - 1200	Formal Modeling and Analysis of Hierarchical Path Planning	*Cesare Tinelli* *(The University of Iowa)*	43
1200 - 1330	**LUNCH (ON YOUR OWN)**		
1330 - 1430	**KEYNOTE PRESENTATION** Data and Decision Analytics	*Robert Bonneau* *(Office of the Secretary of Defense)*	45
1430 - 1500	**REFRESHMENTS**		
1500 - 1530	Automatic Software Verification for High-Assurance Embedded Control Systems	*Miroslav Pajic* *(Duke University)*	46
1530 - 1600	Safety-Constrained Reinforcement Learning for MDPs	*Nils Jansen* *(RWTH Aachen University)*	48
1600	**CONFERENCE ADJOURNED**		

CONFERENCE PRESENTATIONS

bold name denotes presenter

KEYNOTE PRESENTATION
VERIFICATION ACROSS INTELLECTUAL PROPERTY BOUNDARIES

Sagar Chaki*, Christian Schallhart**, Helmut Veith†

**Software Engineering Institute, Carnegie Mellon University,*
***Oxford University , †Vienna University of Technology*

Abstract:

In many industries, the importance of software components provided by third-party suppliers is steadily increasing. As the suppliers seek to secure their intellectual property (IP) rights, the customer usually has no direct access to the supplier's source code, and is able to enforce the use of verification tools only by legal requirements. In turn, the supplier has no means to convince the customer about successful verification without revealing the source code. This article presents an approach to resolve the conflict between the IP interests of the supplier and the quality interests of the customer. We introduce a protocol in which a dedicated server (called the "amanat") is controlled by both parties: the customer controls the verification task performed by the amanat, while the supplier controls the communication channels of the amanat to ensure that the amanat does not leak information about the source code. We argue that the protocol is both practically useful and mathematically sound. As the protocol is based on well-known (and relatively lightweight) cryptographic primitives, it allows a straightforward implementation on top of existing verification tool chains. To substantiate our security claims, we establish the correctness of the protocol by cryptographic reduction proofs.

This is joint work with Christian Schallhart and Helmut Veith.

Bio:

Sagar Chaki is a senior Member of Technical Staff at the Software Engineering Institute at Carnegie Mellon University. He received a B.Tech in Computer Science & Engineering from the Indian Institute of Technology, Kharagpur in 1999, and a Ph.D. in Computer Science from Carnegie Mellon University in 2005. These days, he works mainly on model checking software for real-time and cyber-physical systems, but he is generally interested in rigorous and automated approaches for improving software quality. He has developed several automated software verification tools, including two model checkers for C programs, MAGIC and Copper. He has co-authored over 50 peer reviewed publications. More details about Sagar and his current work can be found at http://www.contrib.andrew.cmu.edu/~schaki/.

THE SCIENCE OF DEEP SPECIFICATION

Benjamin C. Pierce[†], Andrew Appel**, Adam Chlipala*, Zhong Shao[‡], Stephanie Weirich[†], Steve Zdancewic[†]

**Massachusetts Institute of Technology,*
***Princeton, [†]University of Pennsylvania, [‡]Yale*

Abstract:

Abstraction and modularity underlie all successful hardware and software systems: We build complex artifacts by decomposing them into parts that can be understood separately. Modular decomposition depends crucially on the artful choice of interfaces between pieces. As these interfaces become more expressive, we think of them as specifications of components or layers. Rich specifications based on formal logic are little used in industry today, but a practical platform for working with them would significantly reduce the costs of system implementation and evolution by identifying vulnerabilities, helping programmers understand the behavior of new components, facilitating rigorous change-impact analysis, and supporting maintainable machine-checked verifications that components are correct and fit together correctly.

Recently, research in the area has begun to focus on a particularly rich class of specifications, which might be called "deep specifications". Deep specifications are *rich* (describing complex component behaviors in detail); *two-sided* (connected to both implementations and clients); *formal* (written in a mathematical notation with clear semantics to support tools such as type checkers, analysis and testing tools, automated or machine-assisted provers, and advanced IDEs); and *live* (connected via machine-checkable proofs to the implementation and client code). These requirements impose strong functional correctness conditions on individual components and permit them to be connected together with rigorous composition theorems.

This talk discusses the key features of deep specifications, surveys recent achievements and ongoing efforts in the research community (in particular, work at Penn, Princeton, Yale, and MIT on formalizing a rich interconnected collection of deep specifications for critical system software components, under the umbrella of a recently funded NSF Expedition in Computing), and argues that the time is ripe for an intensive effort in this area, involving both academia and industry and integrating research, education and community building. The ultimate goal is to provide rigorously checked proofs about much larger artifacts than are feasible today, based on decomposition of proof effort across components with deep specifications.

http://deepspec.org

Bio:

Benjamin Pierce is Henry Salvatori Professor of Computer and Information Science at the University of Pennsylvania and a Fellow of the ACM. His research interests include programming languages, type systems, language-based security, computer-assisted formal verification, differential privacy, and synchronization technologies. He is the author of the widely used graduate textbooks Types and Programming Languages and Software Foundations. He has served as co-Editor in Chief of the Journal of Functional Programming, as Managing Editor for Logical Methods in Computer Science, and as editorial board member of Mathematical Structures in Computer Science, Formal Aspects of Computing, and ACM Transactions on Programming Languages and Systems. He is also the lead designer of the popular Unison file synchronizer.

INTERRUPTS IN OS CODE: LET'S REASON ABOUT THEM. YES, THIS MEANS CONCURRENCY.

June Andronick[†*], Corey Lewis[†], Carroll Morgan[†*]

[†]*Data61 | CSIRO (formerly NICTA), * The University of New South Wales*

Abstract:

Existing modelled and verified operating systems *(OS's)* typically run on uniprocessor platforms and run with interrupts mostly disabled. This makes formal reasoning more tractable: execution is mostly *sequential.*

The *eChronos* OS is a real-time *OS* used in tightly constrained devices such as medical implants, running on (uniprocessor) embedded micro-controllers with no memory-protection support. It is used in the DARPA-funded HACMS program, where it runs the flight control software of a high-assurance quadcopter.

To provide low latency, the *eChronos* OS runs with interrupts *enabled* and provides a *preemptive* scheduler. This means that external hardware device interrupts may happen at any time, including during execution of OS code, and that application tasks may be preempted by the scheduler to run a more "urgent" task. The notion of urgency is captured by *priorities* given to tasks, and the main job of the scheduler is to guarantee that the task executing is always the highest-priority runnable task.

In terms of verification, this means that concurrency reasoning is required: at any given point in time, a hardware interrupt may happen as OS code is running (including scheduler code), leading to execution switching to interrupt handling code. Formally reasoning about concurrent execution significantly increase the complexity: application and OS instructions may be interleaved with handler instructions

In our work we explicitly model the effect of interrupts and their handling by the hardware and OS. We provide a general formal model of the interleaving between OS code, application code and interrupt handlers. We then instantiate this model to formalise the scheduling behavior of the *eChronos* OS, and prove the main scheduler property mentioned above.

Our work is all formalised and machine-checked in the Isabelle theorem prover. It adapts the Owicki-Gries methods for concurrency reasoning to model the non- deterministic occurrences of interrupts. Our model supports both direct and delayed calls to the scheduler, as well as nested interrupts.

Formal verification has increased significantly the reliability and security of software systems in recent years. Tackling the concurrency induced by inter- rupts, and more generally providing frameworks for reasoning about interleaved execution of low-level OS code, will make it possible to apply high-assurance techniques to a wider range of domains, including real-time and multicore.

Acknowledgements

Data61 (formerly NICTA) is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre-of-Excellence Program.

Bio:

June Andronick is a Senior Researcher at Data61|CSIRO (formerly NICTA). Her research focuses on increasing the reliability of critical software systems, by mathematically proving that the code behaves as expected and satisfies security and safety requirements. She contributed to the seL4 correctness proof and now focuses on concurrency reasoning for OS code. She leads the concurrency software verification research in Data61, and is deputy leader of the Trustworthy Systems group. She was recognised in 2011 by MIT's Technology Review as one of the world's top young innovators (TR35). She holds a PhD in Computer Science from the University of Paris-Sud, France.

ANDROID PLATFORM MODELING AND ANDROID APP VERIFICATION IN THE ACL2 THEOREM PROVER

Eric W. Smith

Kestrel Institute

Abstract:

We present our work in using the ACL2 theorem prover to formally model the Android platform and to formally verify Android apps. Our approach allows the verification of the full functional correctness of apps (e.g., that a calculator app computes the correct numeric result) as well as security properties (e.g., that an app only sends data to certain URLs). Verifying an app with our system provides high assurance that it satisfies its specification. A major motivation for this work is to detect or prove the absence of "functional malware", malicious app functionality that is triggered by complex conditions on state and whose malicious action is to cause the app to calculate the wrong results or otherwise behave incorrectly, unbeknown to the user.

Android is an event-driven system. Our formal model is an executable simulator of a growing subset of the Android platform, and app proofs are done by automated symbolic execution of the app's event handlers using the formal model. By induction, we prove that an app satisfies an invariant, including the correctness properties of interest, for all possible sequences of events. To our knowledge, our formal Android model is the most detailed and our Android app verification is the most thorough, compared to other approaches.

Bio:

Dr. Eric W. Smith is a Senior Computer Scientist at Kestrel Institute with 15 years of experience in formal methods. He currently leads Kestrel's DARPA MUSE effort, which uses synthesis techniques (derivations, specifications, and refinements) to construct correctness proofs of code found in large online repositories. This allows the code to be soundly used in program synthesis. He also leads Kestrel's DARPA APAC effort to find malware in Android apps or formally prove its absence. Dr. Smith has applied Kestrel's Specware system for software synthesis in various domains for DoD customers. This work included producing machine-checkable proofs of correctness in the Isabelle/HOL theorem prover. He is currently leading a project to bring Specware-like synthesis capabilities to the ACL2 theorem prover.

Before joining Kestrel, Dr. Smith completed his Ph.D. in Computer Science at Stanford University under Prof. David Dill. He wrote Axe, a theorem prover and equivalence checker capable of highly automated proofs about real-world cryptographic programs. Dr. Smith has extensive experience with the ACL2 theorem prover, using it for microprocessor verification at AMD and for processor modeling and machine code proofs at Rockwell Collins. Dr. Smith did his undergraduate work at the University of Texas at Austin, where he earned bachelor's degrees in Computer Science and Plan II Honors under thesis advisor J Moore.

"IT'S QEDs ALL THE WAY DOWN"

Experience Report on the Use of a Verified Logic-Specification-to-Machine-Code Toolchain and Verified Execution Environment

David S. Hardin and Konrad Slind

Rockwell Collins

Abstract:

Traditional Formal Verification efforts have suffered from a fundamental issue, namely that the verification artifacts are only a model of the system that is to be verified. This modeling gap can be ameliorated by the development of verified translators; often, however, such engineering-artifact-to-logic-specification translators are untrusted. And even if a verified translator is created, it generally functions at the source code level, leaving us to trust the compiler/assembler/linker. The situation can be improved somewhat through the use of a verified compiler, such as the CompCert C compiler; however, for CompCert, the backend assembler and linker tools charged with producing the final executable are still trusted. Further, there is no guarantee that CompCert's C semantics match that of one's own verified translator, unless the CompCert frontend can be re-used, or an equivalence proof is carried out.

Instead of starting with a translated model, let us imagine a scenario in which a developer crafts --- within a theorem proving environment --- an initial formal specification of the functionality that she/he intends to produce. Most major theorem provers provide some sort of source code generation path from such a starting point, but none of them provides verified translation to source code, much less to machine code. Let's assume that such a verified translation path existed, allowing a developer to create a trustworthy machine code program from a logic specification by a series of verified steps. This sort of development could be conducted according to a verified program refinement process; with sufficient automation support, such verified program refinement could be performed by developers who are not theorem proving experts. Further, by utilizing a verified execution environment, including a verified program loader as well as verified runtime components such as a garbage collector and bignum library, one could achieve both verification and validation of the machine code expression of a formal logic model, with no "trusted" elements above the CPU hardware. This may be a higher level of assurance than most developments require, but for certain high-assurance security-critical and safety-critical applications, we deem the effort worthwhile.

We have been experimenting with such a verified toolchain and verified execution environment, using technologies from the CakeML project. CakeML provides mechanisms to convert HOL4 logic formulas to a subset of Standard ML, as well as a verified compiler from that ML subset to x86 machine code. We have been able to perform a verified translation of a HOL4 specification of a Deterministic Finite-state Automaton-based regular expression matcher to x86 machine code, and successfully execute that machine code on x86 hardware using the CakeML verified read-eval-print loop, exercising a large number of validation test cases. HOL4 proofs formally connect the behavior of the DFA to the set-theoretic semantics of regular expressions. We have also experimented successfully with other classic functional programs, such as programs to manage priority queues and red/black trees.

Future work includes integrating this work with an established program refinement methodology such as Kestrel's Specware; investigating the creation of efficient machine code from functional specifications (inspired by ACL2's single-threaded object feature); and executing our verified x86 machine code programs on the detailed ACL2 formal model of the x86 currently in development at the University of Texas.

Bio:

Dr. David S. Hardin has made contributions in the areas of formal methods, computer architecture for High Assurance systems, as well as real-time and embedded Java. He is currently a Principal Research Engineer in the Advanced Technology Center at Rockwell Collins, where he has served as Principal Investigator for several U.S. DoD Research and Development programs. He is the editor of the book Design and Verification of Microprocessor Systems for High-Assurance Applications (Springer 2010), and a co-author of The Real-Time Specification for Java, as well as the Java 2 Micro Edition Connected Device Configuration/Foundation Profile standards. He is author or co-author of more than 40 peer-reviewed publications, and is a co-inventor on eleven U.S. patents. In 1999, Dr. Hardin co-founded aJile Systems, a startup company focused on real-time and embedded Java technology, and served as aJile's Chief Technical Officer from 1999 to 2003. Dr. Hardin was selected as a Rockwell Engineer of the Year for 1997 for his contributions to the development of the world's first Java microprocessor. His academic career includes BSEE and MSEE degrees from the University of Kentucky, and a Ph.D. in Electrical and Computer Engineering from Kansas State University, in 1989. Dr. Hardin is a proud native of the Commonwealth of Kentucky, and is a Kentucky Colonel.

MOVING HARDWARE FROM "SECURITY THROUGH OBSCURITY" TO "SECURE BY DESIGN"

Ryan Kastner

University of California, San Diego

Abstract:

It is a difficult, perhaps impossible, task to design modern hardware that is impervious to any and every attack. It is hard to insure that these complex, multi-billion transistor systems are functionally correct, let alone secure. Yet, for the most part, computing system designers assume that the hardware is secure and focus their security efforts at higher levels of abstraction (OS, programming language, algorithm, etc.). Recent attacks have shown this is a false premise, and building upon an insecure foundation is a recipe for disaster.

In this talk, we discuss techniques that enable the designer to reason about hardware security. These techniques are based upon information flow and information theoretic measures. They are oblivious to the types of variables under consideration. Thus, we can assess both functional security properties related to confidentiality and integrity as well as covert channels. Our techniques enable the characterization of portions of the system that are potentially vulnerable to attacks. And they determine the effectiveness of mitigation techniques on the overall security of the system. The end result is more secure hardware, which leads to safer and more secure computing systems.

Bio:

Ryan Kastner is a professor in the Department of Computer Science and Engineering at the University of California, San Diego. He received a PhD in Computer Science (2002) at UCLA,, a Masters degree in engineering (2000) and Bachelor degrees (BS) in both Electrical Engineering and Computer Engineering (1999), all from Northwestern University. He spent the first five years after his PhD as a professor in the Department of Electrical and Computer Engineering at the University of California, Santa Barbara.

Professor Kastner's current research interests fall into three areas: hardware acceleration, hardware security, and remote sensing. He is the co-director of the Wireless Embedded Systems Master of Advanced Studies Program. He also co-directs the Engineers for Exploration Program. He has published over 150 technical articles, and has authored three books, "Synthesis Techniques and Optimizations for Reconfigurable Systems", "Arithmetic Optimizations for Polynomial Expressions and Linear Systems", and "Handbook on FPGA Design Security". He has served as member of numerous conference technical committees spanning topics like reconfigurable computing (ISFPGA, FPL, FPT), hardware design (DAC, ICCAD, DATE), hardware security (HOST), and underwater networking (WUWNet).

LEARNING STATE-RICH SPECIFICATIONS FROM "BIG CODE"

Swarat Chaudhuri

Rice University

Abstract:

Automated reasoning about software, whether static or dynamic, requires well-defined notions of program correctness. However, in many real-world settings, correctness specifications are simply not available.

Consider, in particular, the problem of ensuring that a program conforms to the usage protocol for a software library that it uses. Many well-known research projects have studied this problem; the problem is also especially pertinent today with the availability of many freely available software libraries. The difficulty, however, is that real-world libraries seldom come with formally-defined usage specifications.

In this talk, we describe a method to *automatically learn* stateful, probabilistic specifications of library usage from large amounts of code. The learned specifications can be used in a variety of formal methods settings, including static analysis, runtime monitoring, program repair, and program synthesis.

The central idea in our approach is to automatically learn specifications for a library from *examples* of its real-world use in large corpora of open-source code. Through automatic analysis of such *Big Code*, we generate large volumes of data that captures all common ways in which a library is used. Machine learning techniques are then used to recognize library usage patterns in the data. Under the reasonable assumption that most real-world code uses libraries correctly, these patterns serve as normative specifications of how future developers *should* use the library. A client of the library that deviates from these norms would be using the library anomalously. Possibly, such an anomaly would be a bug, and if not, pointing out such deviations would arguably be of interest to an application developer.

The hallmark of our approach, something that sets it apart from prior statistical approaches to software analysis, is the ability to learn *state-rich* specifications that assert logical constraints between arguments and return values of different library calls. For example of such a state-rich specification is that a call to the `hasNext()` method for an iterator should return *true* before a call to `getNext()` is made. Such complex specifications assert properties of the *program state* at each event, something that existing statistical approaches do not capture. Our primary

contribution is a method to explicitly model the program state at events during the execution of a program in a way that is amenable to subsequent machine learning.

The main difficulty with state-rich specifications is a combinatorial explosion in the number of events that the model tracks, arising from the fact that each event must now include nontrivial information about program

state. This is a direct consequence of the well-known state explosion problem in program analysis. The key feature of our statistical model is that it is specifically designed to be cognizant of program state information in the data, and uses some natural structural insights about specifications, to avoid this combinatorial blowup.

Our model employs a set of finite-state machines, or *monitors*, to observe the evolution of a sequence of calls. These monitors execute synchronously with a call sequence, performing state transitions when new calls are seen. Transitions can depend on logical relationships between the arguments or state of the monitor at the current call and some previous calls, and thus capture temporal constraints over calls. The model's event at a call is a vector of monitor states, one for each monitor. We then use a Markov Random Field to learn a probabilistic explanation for how these monitors perform their state transitions. A highlight of the model is the use of independence assumptions that fit intuitions about customized for our application, and permits tractable learning and inference algorithms.

The specifications encoded by our model can be used in a variety of applications. In this talk, we describe one such application: the use of our model to detect anomalous library usage in Android applications.

Specifically, we train our model on sequences of library calls generated from a corpus of 1300 Android apps, generated via symbolic execution. Sequences to which the model assigns low probabilities often correspond to subtle and difficult-to-spot library usage violations. These violations range from GUI bugs to inadequate encryption strength; in several cases, they defy easy logical characterization.

Bio:

Swarat Chaudhuri is an associate professor of computer science at Rice University. He is an expert on methods for automated reasoning about systems, in particular abstract interpretation and model checking, and the application of such methods in computer-aided programming.

Prof. Chaudhuri received a bachelor's degree in computer science from the Indian Institute of Technology, Kharagpur, in 2001, and a doctoral degree in computer science from the University of Pennsylvania in 2007. From 2008-2011, he was an assistant professor at the Pennsylvania State University, University Park. He is a recipient of the National Science Foundation CAREER award, the ACM SIGPLAN Outstanding Doctoral Dissertation Award, and the Morris and Dorothy Rubinoff Dissertation Award from the University of Pennsylvania.

A STRING, REGULAR EXPRESSION, AND INTEGER SOLVER FOR BUG-FINDING AND SECURITY

Yunhui Zheng*, Julian Dolby*, Vijay Ganesh†, Sanu Subramanian†, Omer Tripp*, Xiangyu Zhang**

**IBM, ***Purdue University, †University of Waterloo*

Abstract:

In recent years, string solvers have become an essential component in many formal-verification, security-analysis and bug-finding tools. Such solvers typically support a theory of string equations, the length function as well as the regular-expression membership predicate. These enable considerable expressive power, which comes at the cost of slow solving time, and in some cases even nontermination. We present two techniques, designed for word-based SMT string solvers, to mitigate these problems: (i) sound and complete detection of overlapping variables, which is essential to avoiding common cases of nontermination; and (ii) pruning of the search space via bi-directional integration between the string and integer theories, enabling new cross-domain heuristics. We have implemented both techniques atop the Z3-str solver, resulting in a significantly more robust and efficient solver, dubbed Z3str2, for the quantifier-free theory of string equations, the regular-expression membership predicate and linear arithmetic over the length function. We report on a series of experiments over four sets of challenging real-world benchmarks, where we compared Z3str2 with five different string solvers: S3, CVC4, Kaluza, PISA and Stranger. Each of these tools utilizes a different solving strategy and/or string representation (based e.g. on words, bit vectors or automata). The results point to the efficacy of our proposed techniques, which yield dramatic performance improvement. We argue that the techniques presented here are of broad applicability, and can be integrated into other SMT-backed string solvers to improve their performance.

Bio:

Yunhui Zheng is a Research Staff Member at the IBM T.J. Watson Research Center. His interest lies in program analysis of the web and mobile applications for testing, debugging, verification and vulnerability detection. He is also interested in string analysis that integrates (string) constraint modeling and solving techniques into program analysis. He is the main author of the string constraint solver Z3-str/Z3str2.

He received his PhD in Computer Science from Purdue University. For his thesis, he investigated techniques for static web application analysis and string constraint solving.

KEYNOTE PRESENTATION

METRIC OR ENGLISH: CHARACTERIZING AND CONVEYING TRUSTWORTHINESS

Fred Schneider

Cornell University

Abstract:

Not only do we seek trustworthy systems, but we must be convinced that trustworthiness is being achieved, and we must convince others of it too. We similarly seek a way to convey the returns from research in this space. Yet means of quantification have eluded us. This talk will survey the landscape, characterizing what we might expect of a characterization of trustworthiness (along with inherent limitations) and what kinds of vehicles are infeasible.

Bio:

Fred B. Schneider is the chairman and Samuel B. Eckert Professor of Computer Science at Cornell, where he has been on the faculty since 1978.

Schneider's research concerns trustworthy systems, most recently focusing on computer security. He was the editor of "Trust in Cyberspace" which reports findings from the US National Research Council's study committee on information systems trustworthiness that Schneider chaired.

A fellow of the AAAS, ACM, and IEEE, Schneider was awarded a D.Sc. [honoris causa] by the University of Newcastle-upon-Tyne in 2003. His survey paper on state machine replication received a SIGOPS Hall of Fame Award in 2007. He received the 2012 IEEE Emanuel R. Piore Award for "contributions to trustworthy computing through novel approaches to security, fault-tolerance and formal methods for concurrent and distributed systems". And he was elected to membership of the US National Academy of Engineering (NAE) and to its Norwegean counterpart (NTV).

Schneider is a member of the board for the Computing Research Association, the NRC Computer Science Telecommunications Board (CSTB) and NRC Naval Studies Board (NSB), and he is the founding chair of NRC Forum on Cyber-Resilience. He has served on the Pentagon's Defense Science Board (DSB) and continues to serve on various of its task forces.

BUILD IT BREAK IT FIX IT: MEASURING SECURE DEVELOPMENT

Andrew Ruef, Michael Hicks, David Levin, Piotr Mardziel, Atif Memon, James Parker, Jandelyn Plane

University of Maryland

Abstract:

Security failures plague our software infrastructure every day. Many specialists have proposed their own fixes. The programming languages community often asserts that if software was made from stronger stuff then these failures would occur less frequently. Security practitioners insist that developers must be trained in security and that security must be built in from day one. Static analysis companies say their tools would identify the bugs before they were pushed to production. Security oriented library authors say their libraries are too simple for developers to mis-use and their use would help the security of software.

Can we measure the security impact that programming languages and developer practices have? We believe that we can, using a contest that we have developed: Build It Break It Fix It. Our hope is that this contest provides a source of data by which we can study and understand the relationship between security critical failures of software, and the manner in which that software was developed.

The format of the contest differs from past capture the flag and programming competitions. Our contest takes place over three phases. The first phase, Build It, has the contestants create software to a specification that we provide. The software may be in any programming language as long as it compiles on a specific Linux virtual machine. The specification defines correct behavior of the system as well as a basic threat model and security invariants that the specification should hold. We provide automated unit testing of the applications. We assign a score to each implementation based on performance properties of the application, for example execution time and the size of data generated.

In the second phase, Break It, contestants are given the source code to every other contestants implementation and told to find security bugs. These bugs are either correctness, confidentiality, or integrity bugs in the language of the original specification. In the final phase, Fix It, teams may respond to bug reports against their application by identifying that different reports all refer to the same bug in their system. At the end of the three phases, we have winners in two categories: building, and breaking.

We have run this contest multiple times with multiple specifications and in our talk will share our initial analysis of the data, our experiences running the experiment, and our plans for the future. Our contest runs have included both independent contests with participants from the open Internet as well as contests held as a capstone exercise as part of a Coursera MOOC on software security.

We believe that the corpus of applications and specifications would be of interest for the application security community. This corpus represents the efforts of programmers with different levels of education, experience, and exposure to security topics to create secure software in different programming languages. We can compare and contrast these software artifacts, as well as use these artifacts to test the effectiveness of bug finding systems and methodologies. We would also be interested in feedback on our experimental design and suggestions for future problem specifications to run as contests.

Bio:

Andrew Reuf is a PhD student at the University of Maryland (UMD), College Park. His research focuses on programming languages and computer security. Before starting his graduate work, Andrew worked for ten years as a security researcher and developer of low level and operating system software.

MEASURING PROTOCOL STRENGTH WITH SECURITY GOALS

Paul D. Rowe[†], Joshua D. Guttman[†*], Moses D. Leskov[†]

[†]*The MITRE Corporation, * Worcester Polytechnic Institute*

Abstract:

Flaws in published standards for security protocols are found regularly, often after systems implementing those standards have been deployed. Because of deployment constraints and disagreements among stakeholders, different fixes may be proposed and debated. In this process, security improvements must be balanced with issues of functionality and compatibility.

We provide a family of rigorous metrics for protocol security improvements. These metrics are sets of first order formulas in a goal language

$\mathcal{GL}$ (II) associated with a protocol. The semantics of $\mathcal{GL}$ (II) is compatible with many ways to analyze protocols, and some metrics in this family are supported by many protocol analysis tools. Other metrics are supported by our Cryptographic Protocol Shapes Analyzer CPSA.

This family of metrics refines several "hierarchies" of security goals in the literature. Our metrics are applicable even when, to mitigate a flaw, participants must enforce policies that constrain protocol execution. We recommend that protocols submitted to standards groups characterize their goals using formulas in $\mathcal{GL}$ (II), and that discussions comparing alternative protocol refinements measure their security in these terms.

Bio:

Dr. Paul D. Rowe is a Lead Cybersecurity Researcher at The MITRE Corporation. His research interests include cryptographic protocol analysis, Trusted Computing, cyber resiliency and formal methods for modeling and verification. He is a key contributor to MITRE's protocol analysis tool, the Cryptographic Protocol Shapes Analyzer (CPSA), with applications ranging from key management systems for small unmanned aviation systems (SUAS) to the trust infrastructure of emerging vehicle-to-vehicle communications. He received his PhD in mathematics from the University of Pennsylvania.

COMBINATORIAL COVERAGE ANALYSIS OF SUBSETS OF THE TLS CIPHER SUITE REGISTRY

Dimitris Simos**, Raghu Kacker*, Kristoffer Kleine**, Rick Kuhn***

**National Institute of Standards and Technology , **SBA Research*

Abstract:

We present a combinatorial coverage measurement for (subsets) of the TLS cipher suite registries by analyzing the specific ciphers of IANA, ENISA, BSI, Mozilla and NSA Suite B. Our findings contribute towards the design of quality measures of recommended ciphers for TLS and also lead to important questions regarding the future development of TLS.

Bio:

Dr. Dimitris E. Simos is a Key Researcher with SBA Research, Austria, working on mathematical aspects of information security. He is also an Adjunct Lecturer with Vienna University of Technology.

Dimitris has a keen interest on combinatorial designs and error-correcting codes. His research interests extend to the application of combinatorial designs to software testing, combinatorial testing in particular, error-correcting codes and their applications to post-quantum cryptography.

He holds a Ph.D. in Discrete Mathematics and Combinatorics (2011) from the National Technical University of Athens. Prior to joining SBA Research, he was within the Project Team SECRET of INRIA Paris-Rocquencourt Research Center working on the design and analysis of cryptographic algorithms. He has been awarded a Marie Curie Fellowship (2012-2015) and he is also a Fellow of the Institute of Combinatorics and its Applications (FTICA) since 2012.

NEW PERSPECTIVES ON AUTOMATED VULNERABILITY DISCOVERY

Artem Dinaburg

Trail of Bits

Abstract:

Automated vulnerability discovery systems are effective, but rarely used because they are complex and difficult to maintain and extend. Small and well-tested tools such as fuzzers are fundamentally limited in their capability, but widely deployed to secure production code. In this talk I will discuss a new model for automated vulnerability discovery that intelligently combines simple, existing tools to achieve effectiveness comparable to large integrated vulnerability discovery systems. This approach to vulnerability discovery is extendable by design and simple to parallelize and distribute.

Bio:

Artem Dinaburg was the Principal Investigator for Trail of Bits™ DARPA Cyber Grand Challenge team. He was responsible for the architecture, design, and development of the Trail of Bits™ automated vulnerability discovery system. Mr. Dinaburg has extensive software engineering experience working in application software development, low-level software development, vulnerability research, reverse engineering, malicious software analysis, and program analysis.

GRADUAL INFORMATION FLOW CONTROL

Peter Thiemann and Luminous Fennell

Universität Freiburg

Abstract:

Information-flow control (IFC) is a cornerstone of language-based security. A typical IFC policy rules out the flow of information from classified sources to public sinks. The technical property aimed for is noninterference: changes in a classified source do not influence the public sinks. Noninterference comes in different flavors depending on the observational capabilities of an attacker.

Static IFC may take the form of a security type system, which guarantees noninterference for well-typed programs. Dynamic IFC attaches run-time security labels to values, propagates them along with the values, and checks them at appropriate points during program execution, which can be expensive. Hybrid systems enhance the precision of dynamic IFC with additional static analysis to detect implicit flows.

We investigate gradual security type systems that enable mixing static and dynamic (hybrid) IFC in the same program. Our systems guarantee secure information flow for sequential programs with mutable objects and virtual method calls. A program is composed of fragments that are checked either statically or dynamically. Statically checked fragments adhere to a security type system so that they incur no run-time penalty whereas dynamically checked fragments rely on passing and processing run-time security labels. The programmer marks the boundaries between static and dynamic checking with casts so that it is always clear whether a program fragment requires run-time checks.

Our system relies on security annotations on fields and methods. A field annotation either specifies a fixed static security level or it prescribes dynamic checking. A method annotation is a constrained polymorphic security signature. The types of local variables in method bodies are analyzed flow-sensitively and require no annotation. The dynamic checking of fields can be improved by relying on an optional static pointer analysis to approximate implicit flows.

The system is sound and guarantees termination-insensitive noninterference. We sketch the design of a run-time system, the steps needed to extend to a full OO-language like Java, and a path to integrate legacy code.

Bio:

Peter Thiemann obtained his diploma in computer science in 1987 at the Technical University of Aachen, Germany. He graduated in 1991 at the University of Tübingen, Germany, where he worked as a research and teaching assistant until 1997. In 1998, he was a lecturer in Computer Science at the University of Nottingham, England. Since 1999 he teaches at the University of Freiburg, Germany. He is a full professor at the computer science department and leads the programming languages group.

His research interests comprise theory and practice of modern programming languages, in particular typing, program analysis, and program transformation. He has authored and co-authored more than 100 papers on these and related topics. The focus of his recent research is on automatic program transformation, static and dynamic program analysis for JavaScript, and gradual typing in a security context.

TQL-1 QUALIFICATION OF A MODEL-BASED CODE GENERATOR

S. Tucker Taft

AdaCore

Abstract:

Model-based development is of growing importance in the arena of high-integrity software, including software that is to be certified at level A under DO-178C. For the model-based approach to be practical, the tool that automatically generates source code from a model itself needs to be trusted. Under DO-178C, the process of achieving the highest level of trust in a code generation tool is call "tool qualification at level 1," or simply "TQL-1." This talk will present the innovative process chosen by AdaCore to accomplish TQL-1 qualification of its code generator for Simulink in a systematic yet cost-effective manner.

Bio:

S. Tucker Taft is VP and Director of Language Research at AdaCore, a company focused on open-source tools to support the development of high-integrity software. Tucker joined AdaCore in 2011 as part of a merger with SofCheck, a company he had founded in 2002 to develop advanced static analysis technology. Prior to that Tucker was a Chief Scientist at Intermetrics, Inc. and its follow-ons for 22 years, where in 1990-1995 he led the design of Ada 95. Tucker received an A.B. Summa Cum Laude degree from Harvard University, where he has more recently taught compiler construction and programming language design.

COLLABORATION AND AUTOMATION FOR THREAT ASSESSMENT AND MITIGATION

David Archer and Rogan Creswick

Galois, Inc.

Abstract:

Complex computer networks suffer from a huge number of potential attack surfaces: not just from vulnerabilities in systems, but also from social engineering attacks against the people who use them. Given the ever-changing threat landscape, large numbers of vulnerabilities, and complexity of network resources, human analysts don't have the luxury of carefully considering the severity and implication of each threat, and weighing potential mitigations against one other. The only way to keep up with the adversaries is to add automation to this analysis -- augmenting the human users with automated measurements of the system's security, in the current operational context.

This presentation covers the initial development phase of the Threat Fusion and Effective Response (TFER) project -- a reference implementation of a decision analysis system focused on such automation. The TFER system aims to make the best use of analyst's time in understanding and prioritizing potential threats, make the best use of mitigation resources to respond to those threats, and balance the work and priorities between related teams and organizations engaged in these activities. The reference implementation helps to answer the following three questions:

1) Which Threats are most dangerous to the current operation?
2) Which Assets are at greatest risk?
3) Which Mitigations provide the greatest reduction of risk?

In this presentation we will demonstrate the pre-operational TFER system to show how an assortment of algorithms can assist multiple users in triaging and relating naturally expressed Threat and Mitigation information with computing assets (servers, workstations, laptops, cell phones, and so on). The resulting system automatically draws relationships between these three types of data to provide a baseline level of autonomy that can "fill the gaps" in the (limited) user input available from expert security analysts.

The TFER system demonstrates the feasibility of our general approach: the application of limited autonomy to augment and support multiple human experts, resulting in a cohesive view of the threat landscape as it applies to an operation. Multiple users are able to use the TFER interface to influence the automated reasoning systems, and the changes from those users can be aggregated to provide more holistic cyber situational awareness.a path to integrate legacy code.

Bios:

David Archer

Dr. Dave Archer of Galois, Inc. directs research on high assurance cyber-conflict platforms, cryptographic program obfuscation, computing on encrypted data, identification of persistent threats in computer systems and networks, and assuring information privacy and integrity. Dr. Archer holds a Ph.D. in Computer Science from Portland State University (Portland, OR), and an M.S. in Electrical Engineering and BS in Computer Engineering from the University of Illinois (Urbana-Champaign, IL). Prior to Galois, Dr. Archer was Director of Engineering in the Server Chipset Division at Intel Corporation, and was instrumental in development of the communications network for the ASCI Red TeraFLOPS high performance computer at Sandia.

Rogan Creswick develops unique tools and techniques for software development and security analysis at Galois, Inc. His research interests focus on improving the state of the art in software engineering tools and user interfaces. His experience also reaches into the areas of user interface automation and customization via integrated assistants and automated documentation aides. He strives to provide intuitive tools that ease communication with complex and semi-sentient agents so people can work more efficiently while building trust in their computing systems.

KEYNOTE PRESENTATION
PROGRAMMING UNCERTAIN <T>HINGS

Kathryn McKinley

Microsoft Research

Abstract:

Innovation flourishes with good abstractions. For instance, codification of the IEEE Floating Point standard in 1985 was critical to the subsequent success of scientific computing. Programming languages currently lack appropriate abstractions for uncertain data. Applications already use estimates from sensors, machine learning, big data, humans, and approximate algorithms, but most programming languages do not help developers address correctness, programmability, and optimization problems due to estimates.

To address these problems, we propose a new programming abstraction called Uncertain<T> embedded into languages, such as C#, C++, Java, Python, and JavaScript. Applications use familiar discrete operations for estimates with Uncertain<T>. Overloaded conditional operators specify hypothesis tests and applications use them to control false positives and negatives. A simple compositional operator expresses domain knowledge. We carefully restrict expressiveness such that we can build a runtime that implements correct statistical reasoning at conditionals. Our system relieves developers of the need to implement or deeply understand statistics. We demonstrate substantial programmability, correctness, and efficiency benefits of this programming model for GPS sensor navigation, approximate computing,machine learning, and xBox.

We encourage the community to develop and use abstractions for estimates.

Bio:

Kathryn S. McKinley is a Principal Research at Microsoft. Her research interests span programming languages, compilers, runtime systems, architecture, performance, and energy with a recent focus on programming models for estimates. She and her collaborators have produced several widely used tools: the DaCapo Java Benchmarks (30,000+ downloads), TRIPS Compiler, Hoard memory manager, MMTk memory management toolkit, and Immix garbage collector. Her awards include the ACM SIGPLAN Programming Languages Software Award; ACM SIGPLAN Distinguished Service Award; and best & test of time paper awards from ASPLOS, OOPSLA, ICS, SIGMETRICS, IEEE Top Picks, SIGPLAN Research Highlights, and CACM Research Highlights. She served as program chair for ASPLOS, PACT, PLDI, ISMM, and CGO. She is currently CRA and CRA-W Board member. Dr. McKinley was honored to testify to the House Science Committee (Feb. 14, 2013). She is honored to be among the IEEE and ACM Fellows and to have graduated 22 PhD students. She and her husband have three sons.

FORMAL VERIFICATION OF C PROGRAMS WITH FLOATING-POINT COMPUTATIONS: CERTIFIED ERROR BOUNDS FOR SIGNAL PROCESSING

Tahina Ramananandro

Reservoir Labs Inc.

Abstract:

Model-based development is of growing importance in the arena of high-integrity software, In this technical talk, I will present our results related to the certification of floating-point error bounds in C implementations of signal processing algorithms. In particular, this relates to the topic of reasoning about the uncertainty caused by the noise arising from floating-point rounding errors and approximate computations in C programs.

Our work, funded by the DARPA Microsystems Technology Office (MTO) in the Power Efficiency Revolution for Embedded Computing Technologies (PERFECT) program, is directed at assuring the performance of signal processing when compromises are done to reduce precision to save power. Our work allows one to provably bound the uncertainty introduced by the additional noise due to such compromises.

Technical approach We introduce VCFloat [9], a verification framework for floating-point computations in C programs, based on the Coq proof assistant [4], the Flocq [3] formal specification of IEEE 754 floating-point arithmetic, and the CompCert Clight [1, 6] formal semantics for a realistic subset of C.

Since Flocq and Clight are formal Coq specifications, our approach solely relies on Coq and their faithfulness, thus guaranteeing an unprecedented level of trust in the proofs of floating-point computations in C programs, compared to previous work [2] based on heterogeneous combinations of verification tools.

For a C program, we use CompCert to generate its Clight abstract syntax, then our VCFloat framework transforms C floating-point expressions into their real-number semantics with the appropriate rounding error terms, by automatically generating and checking their validity conditions using the Coq-Interval [7] tactic library for interval arithmetic. Thus, VCFloat solves all floating-point rounding

issues, so that all remaining reasoning about error bounds can be done at the level of real numbers, using Coq-Interval to automatically compute error bounds and their proofs.

For instance, we can prove the correctness of a C implementation of approximate sine against the following Coq specification:

```
forall x,
    (is_finite x = true /\ Rabs (B2R x) <= 2147483647)
->  exists y,
    (eval_funcall FSIN (Vfloat x :: nil) (Vfloat y) /\
     is_finite y = true /\ Rabs (B2R y - sin (B2R x)) <= BOUND).
```

This specification states that, if the argument x is a valid floating-point number no greater than 231, then FSIN, the C implementation of approximate sine studied, does not crash, and produces a result within some absolute error bounded by BOUND of the ideal real-number sine. We compute BOUND within Coq at the same time as we build the correctness proof, using Coq-Interval and VCFloat.

Applications and Conclusion For concreteness, we demonstrate how our approach can provide certifications of realistic C implementations of Synthetic Aperture Radar (SAR) backprojection [5], particularly the safety of energy-efficient optimizations based on

approximate implementations inspired from [8]. But our work can also apply to more general settings such as simulation and numerical algorithms in High-Performance Computing.

Interestingly, our work also pertains to the issue of proofs that cross IP boundaries because VCFloat, and also our correctness proof of SAR backprojection with patent-protected energy-efficient optimizations, build on proof libraries from a variety of providers, each with their own licensing policy, leading to a complex problem in itself, to determine what can be shared, published, and how.

Acknowledgments

Joint work with Paul Mountcastle, Benoît Meister and Richard Lethin, Reservoir Labs Inc. This work is sponsored in part by DARPA MTO as part of the PERFECT program (issued by DARPA/CMO under Contract No: HR0011-12-C-0123). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressly or implied, of the DARPA or the U.S. Government.

References

[1] Sandrine Blazy and Xavier Leroy. Mechanized semantics for the Clight subset of the C language. *Journal of Automated Reasoning,* 43(3):263–288, 2009.

[2] Sylvie Boldo, François Clément, Jean-Christophe Filliâtre, Micaela Mayero, Guillaume Melquiond, and Pierre Weis. Wave equation numerical resolution: A comprehensive mechanical proof of a C program. *Journal of Automated Reasoning,* 50(4):423–456, 2013.

[3] Sylvie Boldo and Guillaume Melquiond. Flocq: A unified library for proving floating-point algorithms in Coq. In *Computer Arithmetic (ARITH), 2011 20th IEEE Symposium on* , pages 243–252, July 2011.

[4] The Coq development team. The Coq proof assistant. `http://compcert.inria.fr` 1984–2015.

[5] Mita D. Desai and W. Kenneth Jenkins. Convolution backprojection image reconstruction for spotlight mode synthetic aperture radar. *Image Processing, IEEE Transactions on*, 1(4):505–517, Oct 1992.

[6] Xavier Leroy. Compcert. `http://compcert.inria.fr`, 2005–2015.

[7] Guillaume Melquiond. Coq-interval. `http://coq-interval. gforge.inria.fr/`, 2008–2015.

[8] Jongsoo Park, Ping Tak Peter Tang, Mikhail Smelyanskiy, Daehyun Kim, and Thomas Benson. Efficient backprojection-based synthetic aperture radar computation with many-core processors. In *Proceedings of Supercomputing '12,* 2012.

[9] Tahina Ramananandro, Paul Mountcastle, Benoît Meister, and Richard Lethin. A Unified Coq Framework for Verifying C Programs with Floating-Point Computations. In 5*th ACM/SIGPLAN International Conference on Certified Programs and Proofs (CPP)*, 2016

Bio:

Dr. Tahina Oliver Ramananandro has been a Senior Engineer at Reservoir Labs, Inc. since 2014. He obtained his Ph. D at Université Denis Diderot in Paris, France in 2012. Dr. Ramananandro has an extensive expertise on advanced compilers and formal verification with proof assistants thanks to his experience in specifying, implementing and proving verified C++ compiler front-ends to the CompCert C verified compiler developed at INRIA, certified separate compilation and linking for the CertiKOS verified operating system kernel developed in collaboration with Yale University, and formally proving numerical properties about floating-point computations in C programs.

WIGMORE: A CONSTRAINT-BASED LANGUAGE FOR REASONING ABOUT EVIDENCE AND UNCERTAINTY

David Burke

Galois, Inc.

Abstract:

Historically, probability theory has proven to be very useful in dealing with uncertainty, especially when it can be quantified by statistical means. This is why the literature on the subject often distinguishes between risk, which applies to situations where uncertainty can be captured by a probability, and ambiguity, when there exists uncertainty without meaningful probabilities.

In the cybersecurity realm, we are often dealing with great amounts of uncertainty, and it is our experience that in this domain, we are dealing with events that are better characterized by ambiguity, not risk, primarily due to the fact that the adversary is not best modeled as a natural, stochastic process, but rather, as a sentient, learning entity.

We are interested in creating software tools to reason about this kind of uncertainty in order to support effective decision-making in the cyber domain, and our work is inspired by a field of research called 'Belief Functions', which is in turn based on the well-known Dempster-Shafer (D-S) theory. Roughly, the difference between D-S theory and traditional probabilistic approaches such as Bayesian networks is that D-S theory is concerned with combining strength of evidence, not about the updating of probabilities. Existing belief function methods typically consist of just a small number of evidence combination operators. While these operators are useful, our needs for adversarial reasoning include not just the aggregation of data as evidence, but also the aggregation of defender's beliefs.

In this talk, we will discuss the design of Wigmore: a language that we have designed (named John Henry Wigmore, a pioneer in the visualization of complex evidence chains) to address

these needs. We have augmented existing belief function operators and D-S theory concepts to produce an expressive, constraint-based language that allows operators to express a rich set of beliefs about the combination of ambiguous pieces of evidence. In this presentation, we will introduce the language, and work through some illustrative examples to show how it can be used to make sense of evidence when conditions of uncertainty prevail.

Bio:

David Burke is a Principal Investigator at Galois with over 20 years of experience in the application of statistical and mathematical modeling, machine learning, and AI techniques to problems in the natural and social sciences, with a specialization in generalized Bayesian techniques for reasoning under uncertainty. He received a M.S. in Computer Science from the Oregon Graduate Institute, and a B.S.M.E. from Lehigh University. Since joining Galois in 2004, his work has included conducting research into logics for reasoning about trust in the design of secure systems, techniques for ensuring robust decision-making in multi-agent systems, and the application of bio-inspired approaches to machine learning and network security. His recent experience includes a PI role on several DoD-funded projects focused on counterdeception, adversarial reasoning, and decision support systems. Mr. Burke is a U.S. citizen.

FORMAL MODELING AND ANALYSIS OF HIERARCHICAL PATH PLANNING

Cesare Tinelli

The University of Iowa

Abstract:

Hierarchical path planning is a family of (2-dimensional) path planning approaches for ground vehicle navigation in complex environments. Its main idea is to represent planning problems at several, increasingly more refined levels of abstraction and have specialized planners at each level. Rough plans are first generated by the planner at the highest level and then progressively refined, if possible, by planners at lower levels until an executable plan is generated. We report on our experience in modeling hierarchical path planners formally with the goal of analyzing and proving meta-level properties such as, for instance, the realizability of a high level plan at a lower level of detail under suitable environmental conditions. Our emphasis is on automated proofs. We have experimented with modeling a common hierarchical path planning approach using a number of modeling languages based on variants of first-order logic, and proving some of its properties using automated provers based on these logics. In this talk, we will focus on two such languages: Alloy and SMT-LIB. We will describe and discuss both successes and remaining challenges in achieving a higher degree of automation in the formal analysis of planning methods.

Joint with Baoluo Meng and Alessandro Pinto, partially supported by a grant from United Technologies Research Center.

Bio:

Cesare Tinelli is a professor of Computer Science at the University of Iowa. He received a PhD in Computer Science from the University of Illinois at Urbana-Champaign in 1999. His research interests include automated reasoning, formal methods, software verification, foundations of programming languages, and applications of logic in computer science.

He has done seminal work in automated reasoning, in particular in Satisfiability Modulo Theory (SMT), a field he helped establish through his research and service activities. He leads the development of the Kind 2 SMT-based infinite-state model checker, and co-leads the development of the award winning and widely used CVC4 SMT solver. He is also a founder and coordinator of the SMT-LIB initiative, an international effort aimed at standardizing benchmarks and I/O formats for SMT solvers. He also co-leads the development of StarExec, a cross community web-based service for the comparative evaluation of logic solvers.

His research has been funded both by governmental agencies (AFOSR, AFRL, DARPA, NASA, and NSF) and corporations (General Electric, Intel, Rockwell Collins, and United Technologies). He received an NSF CAREER award in 2003 and a Haifa Verification Conference award in 2010. He is an associate editor of the Journal of Automated Reasoning and a founder the SMT workshop series and the Midwest Verification Day series. He has served in the program committee of numerous conferences and workshops, and in the steering committee of CADE, ETAPS, FTP, FroCoS, IJCAR, and SMT. He was PC chair of FroCoS'11 and of TACAS'15.

KEYNOTE PRESENTATION
DATA AND DECISION ANALYTICS

Robert Bonneau

Office of the Secretary of Defense

Abstract:

A gap in the practice of modern information technology has been in the evaluation of uncertainty and computational latency, and measurement of information for data analytics algorithms. As more scientific and data analysis becomes automated, verification and validation of automated algorithms will become increasingly critical. We will explore evaluating risk in algorithm performance and human interaction with these processes. In addition, we will investigate strategies for mathematically representing this performance in a wide variety of tasks, from sensor information processing to infrastructure performance analysis.

Bio:

Dr. Robert Bonneau is currently Associate Director for Command, Control, Data Analytics, and Software in the Office of the Secretary of Defense, Assistant Secretary of Defense for Research and Engineering. He also is co-chair of the White House Office of Science and Technology Policy, National Information Technology Research and Development, Large Scale Networking Interagency Working Group. Dr. Bonneau was also the Chief of the Information, Decision, and Complex Networks Division at the Air Force Office of Scientific Research, where he established the Complex Networks, and Foundations of Information Systems Programs. He has held academic positions most recently in the Statistics Department at George Washington University, and engineering and computer science departments at Columbia University, Rensselaer Polytechnic Institute, and Temple University. Dr Bonneau has a Ph.D. in electrical engineering from Columbia University, and a Masters and Bachelors in electrical engineering from Cornell University. Dr. Bonneau has served as Associate Editor of the Springer Journal of Infrastructure Complexity, has over 85 journal and conference papers, has 1 book co-authorship, contributed to 2 book chapters, holds 3 patents, and is a Senior Member of IEEE.

AUTOMATIC SOFTWARE VERIFICATION FOR HIGH-ASSURANCE EMBEDDED CONTROL SYSTEMS

Miroslav Pajic

Duke University

Abstract:

Software based controllers are at the core of many safety--critical embedded and real--time systems, and thus ensuring their correctness is of paramount importance. To reduce development time and provide some degree of assurance, modern controllers are designed in a model-driven manner; from models of control components, different tools are used to automatically generate control code. On the one hand, verification of control systems and evaluation of the quality of control is typically performed at the modeling level. On the other hand, code generators often provide optimized code to improve system performance, potentially causing discrepancies with the initial model while not affecting input-output behavior of the code. Hence, correctly implemented control software may not satisfy invariants directly derived from the initial model.

To ensure that the generated software implementation of the controller is correct with respect to its model, we ideally would like to have verified code generators that would guarantee that any generated controller correctly implements its model. In practice, however, code generators for control software are complex tools that are not easily amenable to formal verification, and are typically offered as black boxes. One of the reasons is that verification would require 'transformation--capturing' annotations that specify information about employed code optimization techniques, which can cause intellectual property (IP) concerns.

Consequently, our work focuses on techniques for verification of instances of generated code against their model while requiring only input--output conformance between the code and the initial model. In this talk, we first present our efforts on automatic verification of linear controllers, the most commonly used type of controllers.

We describe techniques to automatically derive software annotations that are insensitive to optimization performed by a code generator. Furthermore, we present methods based on symbolic code execution and equivalence checking to establish proofs of the software

correctness from the input--output perspective. Although software verification using the aforementioned techniques is initially performed in the domain of real numbers, we consider imprecise implementations of the controller as a step towards numerical verification of control software.

Bio:

Miroslav Pajic is an Assistant Professor in the Department of Electrical & Computer Engineering, Duke University. He also holds a secondary appointment in the Computer Science Department at Duke University. He received his Ph.D. and M.S. degrees in Electrical Engineering from the University of Pennsylvania in 2012 and 2010, and the M.S. and Dipl. Ing. degrees from the University of Belgrade, Serbia, in 2007 and 2003, respectively. Prior to joining Duke in 2015, he was a Postdoctoral Researcher in the PRECISE Center, University of Pennsylvania.

His research interests focus on the design and analysis of cyber-physical systems and in particular real-time and embedded systems, distributed/networked control system, and high-confidence medical device systems. Dr. Pajic received various awards including the 2011 ACM SIGBED Frank Anger Memorial Award, the Joseph and Rosaline Wolf Award for Best Electrical and Systems Engineering Dissertation from Penn Engineering, the Best Paper Award at the 2014 ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS), the Best Student Paper Award at the 2012 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), and Honeywell User Group Wireless Innovation Award.

SAFETY-CONSTRAINED REINFORCEMENT LEARNING FOR MDPs

Nils Jansen**, Christian Dehnert*, Sebastian Junges*,
Joost-Pieter Katoen*, Ufuk Topcu**

**RWTH Aachen University, **University of Texas, Austin*

Abstract:

Many formal system models are inherently stochastic, consider for instance randomized distributed algorithms (where randomization breaks the symmetry between processes), security (e.g., key generation at encryption), systems biology (where species randomly react depending on their concentration), or embedded systems (interacting with unknown and varying environments).

In contrast to verification of such systems, controller synthesis is a relatively new topic in this setting. Having a formal model of a controllable entity–e. g. a robot–and an environment, the goal is to synthesize a controller that satisfies certain requirements. Again, often faithful models are stochastic, imagine, e. g., sensor imprecisions of robots, message loss, or unpredictable environment behavior. Moreover, it might be the case that certain information–such as cost caused by energy consumption–is not known prior to exploration and observation.

Here, we abstract problems as Markov decision processes in which the expected performance is measured using a cost function that is unknown prior to run-time exploration of the state space. Consider for instance a motion planning scenario placed in a grid-world, where a robot wants to move to a certain position. Acting unpredictably, a janitor moves randomly through the grid. The robot reaches its goal safely if it moves according to a strategy that avoids the janitor. Moreover, each movement of the robot occasions cost-depending on the surface while it only learns the actual cost during physically executing actions within the environment.

Standard learning approaches synthesize cost-optimal strategies without guaranteeing safety properties. To remedy this, we first compute safe, permissive strategies. In contrast to standard strategies, where for each system run the next action to take is fixed, more permissiveness is given in the sense that several actions are allowed. Then, exploration is constrained to these strategies and thereby meets the imposed safety requirements. Exploiting an iterative procedure, the resulting strategy is safety-constrained and optimal. We show correctness and completeness of the method and discuss the use of several heuristics to increase its scalability. Moreover, we demonstrate the applicability by means of a prototype implementation.

Bio:

Nils Jansen did his PhD with Erika Ábrahám at RWTH Aachen University, Germany, on the topic "Counterexamples in Probabilistic Verification". Afterwards, he started working as a postdoc with Joost-Pieter Katoen, also in Aachen. His research interests lie in the formal verification, in particular of stochastic systems. Currently, he is mostly working on parametric Markov models. He is also interested in all kinds of applications to formal verification.

CONFERENCE POSTERS

Poster sessions will be held between 2:30 p.m. and 3:30 p.m. on Tuesday, May 10 and Wednesday, May 11 in the atrium of the Governor Calvert House. Posters will be set up for display by the conference staff. Presenters can drop off their posters at the registration desk by noon on Tuesday, May 10.

bold name denotes presenter

An Assessment Methodology, Models For National Security Systems

Jennifer Guild, *US Navy*

Applying User Sessions to Detect SQL Injection Vulnerabilities in Web Applications

Sreedevi Sampath, Isaiah Yoon, Mengzi Du
University of Maryland, Baltimore County

Combinatorial Coverage as an Estimator of Residual Risk after Testing

Rick Kuhn*, Raghu Kacker*, Dimitris E. Simos** Kristoffer Kleine**
**NIST, **SBA Research*

Rule Based Systems and the Intersection of Formal Methods and Testing

Rick Kuhn*, Vincent Du*, David Ferraiolo*, Raghu Kacker*, Dylan Yaga*, Yu Lei**
**NIST , **UT Arlington*

Science of Security and Privacy

Heather Lucas, *National Security Agency*

The Bug Framework (BF): A Taxonomy For Precise and Accurate Software Bug Descriptions

Paul E. Black, Irena Bojanova, Yaacov Yesha, Yan Wu
National Institute of Standards and Technology

Towards a Process to Forecast Vulnerability in Systems of Systems

Joseph Natarian*, Marvin Worst**, Bruce Howard†
**AFRL, **National Air and Space Intelligence Center, † Wright State Research Institute*

Verification and Validation of Autonomous Systems: Verifiable Requirements for Complex Systems

Jon Hoffman* and Brian Hulbert **
**Air Force Research Laboratory, **AFRL and LinQuest Corp.*

Verification of Decision Procedures Modeled in Intelligent Agents

Siddhartha Bhattacharyya* Tom C. Eskridge*, Marco M. Carvalho*, Jennifer Davis**
**Florida Institute of Technology, **Rockwell Collins*

Verifying Programs with Complex Data Structures Using Coq

Kenneth Roe, *The Johns Hopkins University - Applied Physics Lab*

AN ASSESSMENT METHODOLOGY, MODELS FOR NATIONAL SECURITY SYSTEMS

Jennifer Guild

US Navy

Abstract:

The assurance of National Security Systems (NSS), like all computer systems, is measured, or assessed, by a variety of methodologies and assessors. Most assessors know that the level of assurance required for each system is dependent upon dynamic factors such as attack vector persistence, operational environment, and probability of a successful attack, regardless of its complexity or connectivity. This paper presents a methodology that implements mathematical models that are simple enough for non-mathematicians to use, can be integrated into existing acceptance and certification methodologies, or can be implemented standalone, and is based upon lessons learned from over a decade of direct, real world, assessment experience.

In assessments, evidence must be collected and assessed against a model. In existing assessment methodologies, that model is a complete, live implementation of a system in a single operational environment. Each operational environment may contain multiple situational instances or states of physical characterizations (such as an aircraft in flight vice parked).

This paper presents mathematical models detailing various individual factors that contribute to an aggregate measure, including the operational environment states, flaws, countermeasures, vulnerabilities, threats, probabilities, attack vectors, impacts, and risk. The models provide a basis for a new assessment methodology that can be combined with the current and future assessment methodologies, improve confidence in the system by requiring an independent assessor to be integrated into the development process to achieve greater insight, and improve cost savings by preventing duplicate assessments and reducing the time it takes to conduct assessments by allowing future assessments to build on the findings of past assessments.

This methodology provides the ability to model system states to characterize dynamic aspects of the system and environment. Computers alter states every time a decision is completed. So, computers and networks exist in fluidity, each constantly changing. The models need to represent systems in multiple states based on dynamic aspects, analogous to the modeling used in weather forecast models, nuclear explosions, and disease infection rates. This type of modeling provides objective evidence throughout the assessment.

The proposed methodology provides to the assessor mechanisms to map the evidence to mathematical models to assessor's findings. Currently, assessors must rely on documentation provided by the vendor, which can be biased. An Information Systems Security Engineer (ISSE) is key to the entire methodology as it removes any possible bias from a vendor, design team, program manager, command, etc, and the ISSE can provide mathematical foundations supporting evidence creation. The use of the models increases objectiveness, repeatability, and knowledge of system robustness from ISSE to risk acceptor, as well as ISSE to ISSE.

The methodology can be implemented at any time within the development lifecycle of a system. The earlier in the lifecycle the methodology is implemented, the greater the applicability of evidence that is available to the ISSE. In addition, the methodology strongly integrates the ISSE with system's developers and engineers. An ISSE that is involved in the system development processes starting at design conception, can increase the measure of confidence in the assurance of the system by identifying applicable supplementary artifacts, and through the use of subject matter expertise, increase the quality of all assurance evidence.

Individual models will be iteratively addressed so that the ISSE is able to represent each impression of the system's capabilities, correlate the models to the evidence, and provide a level of assessment detail that has heretofore not been provided. As the ISSE's knowledge of the system increases, the content of these models will go from generalized to specific as the assessment progresses. These individual models will build into the overall assessment model. The individual models will be iteratively developed, fulfilling the needs of the assessor to represent their initial impression of the system's capabilities, represent the system's capabilities as it is assessed, and finally, to representatively correlate or map the completed models to the empirical evidence of the assessment.

Within the proposed methodology, there are multiple stages, with each stage correlating to the progression of the assessor's exposure to the system. At each stage, the ISSE iterates the individual models to represent their impression of the system's capabilities. As each assessment is individualistic, the number of stages and the stage at which a model is created will vary wildly based upon the system functionality, and the point in the lifecycle in which the system enters the methodology, and the information available at that the time.

Bio:

Jennifer Guild is a PhD candidate at the University of Idaho who is employed as a computer scientist by the US Navy. She specializes in the assessment of complex systems, such as Cross Domain Solutions. Ms. Guild received an MS in Computer Science from the US Naval Postgraduate School.

APPLYING USER SESSIONS TO DETECT SQL INJECTION VULNERABILITIES IN WEB APPLICATIONS

Sreedevi Sampath, Mengzi Du, Isaiah Yoon

University of Maryland, Baltimore County

Abstract:

Vulnerabilities in web applications are a serious concern for companies and consumers. The large number of technologies that are involved in a web application, such as Flash, HTML, JavaScript, PHP, Ajax etc., and the underlying software, such as web servers and browsers, suggest that the vulnerability can be in any language, technology or component. One of the most common exploits that plague web applications is Code Injection attacks, such as SQL Injection and Cross Site Scripting. In 2011, SQL injection was ranked first, and Cross Site Scripting was ranked fourth on the MITRE Common Weakness Enumeration (CWE)/SANS Top 25 Most Dangerous Software Errors list[1]. Providing developers and testers with a mechanism by which they can identify the parameters that are vulnerable to Code Injection attacks, specifically SQL Injection, will help them develop secure web applications.

In particular, we propose to capitalize on user-session-based test cases to create test cases that are able to expose SQL Injection Vulnerabilities. User sessions capture all user interactions with a web system and thus are representative of actual field usage of the web application [2]. They are particularly useful for SQL Injection attacks, because the attacks are themselves caused by malicious end-users of the web application.

In our approach, we first identify malicious values that cause code injection attacks that are typically given for parameters in web applications. The malicious values we identify are ones commonly used in different types of SQL Injection attacks, such as Boolean Exploitation, Union exploitation, Stacked queries, Time-based, and Error-based exploitation. Then, we select a subset of user sessions by applying reduction algorithms and mutate the selected user sessions by replacing normal values of parameters with the afore identified malicious values. In this poster, we present our approach and report results from an experimental evaluation designed to study the effectiveness of the newly developed test cases at detecting SQL Injection vulnerabilities. In the future, we plan to implement the proposed approach in a tool that will be made available to practitioners and researchers.

References :

[1] Bob Martin, Mason Brown, Alan Paller and Kirby, D. 2011 CWE/SANS Top 25 Most Dangerous Software Errors, (2011) Retrieved, From The MITRE Corporation: http://cwe.mitre.org/top25/archive/2011/2011_cwe_sans_top25.html.

[2] Bryce, R. C., Sampath, S. and Memon, A. M. Developing a Single Model and Test Prioritization Strategies for Event-Driven Software. Software Engineering, IEEE Transactions on, 37, 1 (2011), 48-64. DOI=10.1109/tse.2010.12.

Bio:

Sreedevi Sampath is an Associate Professor in the Department of Information Systems at the University of Maryland, Baltimore County. She earned her Ph.D. and M.S. in Computer and Information Sciences from the University of Delaware in 2006 and 2002, respectively, and her B.E. degree from Osmania University in Computer Science and Engineering in 2000. Her research interests are in the areas of software testing and quality assurance, web applications, software maintenance and software security. She has served on the program committees of international conferences, such as the International Conference on Software Testing Verification and Validation (ICST), International Symposium on Software Reliability Engineering (ISSRE), and the International Conference on Empirical Software Engineering and Measurement (ESEM). She is a member of the IEEE Computer Society.

COMBINATORIAL COVERAGE AS AN ESTIMATOR OF RESIDUAL RISK AFTER TESTING

Rick Kuhn*, Raghu Kacker*, Kristoffer Kleine**, Dimitris E. Simos**

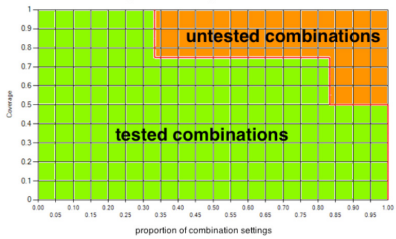
**National Institute of Standards and Technology, **SBA Research*

Abstract:

Empirical data show that a significant number of software failures are induced by the interaction of two or more factors, and interaction faults can be extremely difficult to identify. Thus it is useful to measure the proportion of 2- way, 3-way, and higher strength combinations that are covered by a test set. Any combinations that have not been tested represent a portion of the input space for which the application has not been shown to be correct. Measuring the proportion of the input space for which the system response is untested and unknown can thus provide a useful quantity in estimating residual risk after testing. This poster explains the concept of combinatorial coverage measurement, a variety of measures that are available, and theorems relating (static) combinatorial coverage to (dynamic) structural coverage. These concepts are illustrated with examples comparing measures of tests for a NASA spacecraft and open source test configurations for the TLS cipher suite.

A configuration with n variables contains $\binom{n}{t}$ t-way combinations, so a test set with many configurations will contain a large number of combinations. *Combinatorial coverage* measures the inclusions of t-way combinations in a test set. Note that this measure is different from conventional structural coverage metrics (such as statement or branch coverage) and is independent of these other measures. Because combinatorial coverage measures the input space that is tested, and consequently also the untested portion of input space, it is useful in gauging the residual risk after testing.

Fig. 1. Example test set

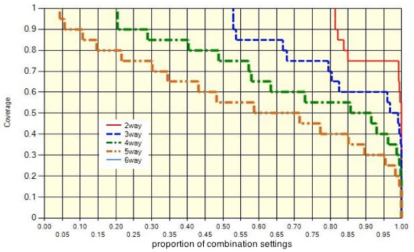


A variety of combinatorial coverage measures are available, including a fundamental measure of *total variable-value configuration coverage*: for a given combination of t variables, the proportion of all t-way value settings that are covered by at least one test case in a test set. For example, two binary variables have four possible settings. Consider four tests containing variables a, b, c, and d: $\{0000, 0110, 1001, 0111\}$.

There are $\binom{4}{2} = 6$ possible variable combinations and $2^2 \times \binom{4}{2} = 24$ possible variable-value configurations. Of these, 19 variable-value configurations are covered and the only ones missing are $ab{=}11$, $ac{=}11$, $ad{=}10$, $bc{=}01$, $bc{=}10$, so the total variable-value configuration coverage is $19/24 = 79$ These measures are shown in Figure 1, where the upper right-hand corner represents the 21% of the 2-way combinations in the input space not tested.

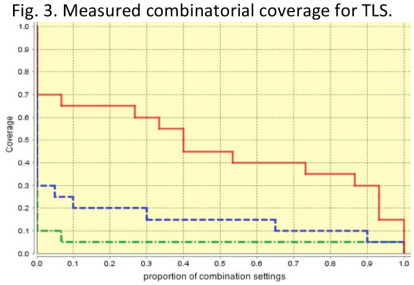
Figure 2 shows measurements for 2-way through 5-way combination coverage for 7,489 tests for a NASA spacecraft. Note that the untested portion for 2-way combinations (above red line) is only about 6% of the total, and 3-way to 5-way coverage is relatively high.

Fig 2. Measured combinatorial coverage for 7,489 tests.



Now compare the measured test configurations for open source tests of the TLS cipher suite in Fig. 3. Less than half of the 2-way combinations are tested, and virtually none for 3-way and 4-way combinations, representing areas of the input space where its configurations are uncovered and could pose significant residual risk.

Fig. 3. Measured combinatorial coverage for TLS.



References:

D. Kuhn, I. Dominguez Mendoza, R. Kacker, and Y. Lei, *"Combinatorial coverage measurement concepts and applications," in Software Testing, Verification and Validation (ICSTW), 2013 IEEE 6th Intl Conf,* pp. 352–361.

K. Kleine, D. Simos, "Coverage analysis of subsets of the TLS cipher suite registry", SBA Research, Oct. 1, 2015.

Bio:

Rick Kuhn is a computer scientist in the Computer Security Division of the National Institute of Standards and Technology. He has authored more than 100 publications on information security, empirical studies of software failure, and software assurance, and is a senior member of the IEEE. He co-developed the role based access control model (RBAC) used throughout industry and led the effort establishing RBAC as an ANSI standard. Before joining NIST, he worked as a systems analyst with NCR Corporation and the Johns Hopkins University Applied Physics Laboratory. He received an MS in computer science from the University of Maryland College Park.

RULE BASED SYSTEMS AND THE INTERSECTION OF FORMAL METHODS AND TESTING

Rick Kuhn*, Vincent Du*, David Ferraiolo*,
Raghu Kacker*, Yu Lei**, Dylan Yaga*

*National Institute of Standards and Technology,
**University of Texas, Arlington

Abstract:

Methods for generating tests from formal models often use model checkers or simulation to serve as a test oracle that determines expected results for a set of test inputs. We describe the derivation of complete test cases from formal access control rules converted to k- DNF structure, using a constraint solver and covering array generator. Two arrays are constructed such that every test in each array should produce the same result, with variations indicating an error. The method has been implemented for testing access control systems with binary *grant/deny* outputs, but may be generalized for verifying other adaptive rule-based systems with a small number of discrete outputs. An interesting aspect of the method is the manner in which the structure of the problem is used to eliminate the need for a conventional test oracle.

Attribute based access control (ABAC) is a method of controlling authorization using rules that include a subject's *attributes* and possibly changing attributes of the operating environment such as time. For example, a rule may allow access to a resource if the subject's attributes include *employee* and *US_citizen*. How should an ABAC system be tested? Confirming that access will be granted for users with any set of rule- specified attributes is easy: we can simply read off the attribute conditions for each *grant* expression and verify that the access control system returns an authorization in each case. However, with possibly hundreds of attributes, it is much more difficult to ensure that rules have not been implemented such that an unspecified combination of attributes results in authorization that should not be permitted.

To resolve this problem, we use covering arrays of attributes from ABAC policies that have been converted to k-DNF form using a constraint solver. A fixed-value covering array of m variables with v values each – denoted by CA(N, v^m, k) – is an N x m matrix of elements from a set of v symbols {0, 1, …, (v – 1)} such that every set of k columns contains each possible k-tuple of elements at least once, where the positive integer k is the *strength* of the covering array (the extension of this definition for variables with different numbers of symbols is straightforward).

A disjunctive normal form expression where no term contains more than k literals is referred to as k-DNF. Recall that a term is a conjunction of one or more literals within the disjunction. For example, $abc + de$ contains two terms, one with three literals and one with two, so the expression is in 3-DNF form. The covering array will contain all k-way combinations of variable values. Where an expression is in k-DNF, any term containing k literals that resolves to true will clearly result in the full expression being evaluated to true. Because a covering array of strength k contains every possible setting of all k-tuples and j-tuples for $j < k$, it contains every combination of values of any k attribute values. For the set of access control rules denoted by R, an array for *grant* conditions is produced such that each test will instantiate only one term to true. A Deny array is generated as a covering array of strength k, for the set of attributes included in R, with constraints specified by $\sim R$ to eliminate from the Deny array all terms that should evaluate to *grant*. This construction results in an array containing all possible conditions for which the access control system should produce an output of *deny*.

The structure of access control rule evaluation makes it possible to compress a large number of test conditions into a few tests. Rules for *grant* conditions are checked in series, then a *deny* issued only after all grant conditions have been evaluated. Each test in the Grant array contains only one term that results in a *grant* decision, ensuring that the presence of one *grant* term does not mask testing another such term in the same test. Terms that should produce a *deny*, however, can be combined in a single test. For m variables with at most k attribute values in each term, up to $\binom{m}{k}$ terms can be evaluated in each test. If any test in the Deny array produces a *grant* response, an error has been discovered, which can be repaired before running the test set again.

Because the number of rows in a covering array grows only with $\log n$ for n attributes at a given number of attributes and values, the process scales easily to systems with a large number of attributes for k-DNF rules with $k < 7$ (currently). For example, it is possible to cover all 3-way combinations of 100 boolean attributes with 45 tests, increasing only to 57 tests for 300 attributes. (Variables with more than two values may also be used.) This method has been implemented for rule sets with binary decisions, and we are extending it to cases where more than two outputs are possible.

Bio:

Rick Kuhn is a computer scientist in the Computer Security Division of the National Institute of Standards and Technology. He has authored more than 100 publications on information security, empirical studies of software failure, and software assurance, and is a senior member of the IEEE. He co-developed the role based access control model (RBAC) used throughout industry and led the effort establishing RBAC as an ANSI standard. Before joining NIST, he worked as a systems analyst with NCR Corporation and the Johns Hopkins University Applied Physics Laboratory. He received an MS in computer science from the University of Maryland College Park.

SCIENCE OF SECURITY AND PRIVACY

Heather Lucas

National Security Agency

Abstract:

The Science of Security & Privacy Initiative at the National Security Agency Research Directorate promotes foundational cybersecurity and privacy science that is needed to mature the cybersecurity discipline and to underpin advances in cyberdefense. Beginning in 2012, one part of the initiative is to fund foundational research at "Lablets." With emphasis on building a community, each lablet created partnerships with other universities called "Sub-Lablets." Science of Security researchers often freely collaborated with researchers in other institutions worldwide. In 2014, the SURE project was founded to investigate cybersecurity in the cyber-physical systems realm. Since 2012, the initiative also promotes rigorous research methods through its annual best paper competition by highlighting the best example of scientific cybersecurity research contribution.

Bio:

Heather Lucas is a program director within the Trusted Systems Research Group and is the current program lead for NSA's Science of Security Virtual Organization effort. Ms. Lucas received a BA from University of Maryland Baltimore County, where she graduated magna cum laude. She is passionate about providing a useful collaborative environment for researchers to share their work, and hopes to one day see open science as a reality. She finds the TEDxWaterloo talk by Michael Nielsen on Open Science to be truly inspirational.

Outside of work, Ms. Lucas loves to travel to warm climates to soak in the sun and play in the water. Locally she enjoys her free time rock climbing, and being creative making peep dioramas.

THE BUG FRAMEWORK (BF): A TAXONOMY FOR PRECISE AND ACCURATE SOFTWARE BUG DESCRIPTIONS

Paul E. Black, Irena Bojanova, Yan Wu, Yaacov Yesha

National Institute of Standards and Technology

Abstract:

Autonomous Intelligent Systems (AIS) are built on principles defined in cognitive architectures tOne significant contribution to measurement of the security of a system is being able to precisely and accurately characterize the vulnerabilities a system has or doesn't have. Medical doctors spend years learning a vocabulary to precisely designate muscles, bones, organs, diseases, and conditions to communicate clearly. We have some similar work, such as Software Fault Patterns (SFPs) [1], Common Attack Pattern Enumeration and Classification (CAPEC), semantic templates [2], and the Common Weakness Enumeration (CWE) [3], but none of them are complete or easy to use for many purposes. For instance, CWEs are a considerable community effort, but many of the descriptions are inaccurate, incomplete, inconsistent, or ambiguous with causes and consquences mixed in. In addition, CWEs are coarse-grained with irregular overlap of coverage and even no coverage in the areas of mobile applications and cyberphysical systems. Without a coherent definition of bugs, it is difficult to state, say, that a system is assured free from a certain class of bugs or that a new technique will absolutely detect their presence.

Just as the Global Positioning System (GPS) requires being able to measure time very precisely, allowing for the Doppler effect and general relativity, we are building the Bug Framework (BF) to precisely and accurately define software bugs. We are (1) breaking down existing CWEs, SFPs, semantic templates, etc. into simple "atoms" or components of bugs, (2) organizing them into meaningful structures and identifying assemble rules, and (3) using this to precisely define bug classes reported by assurance tools, explain known vulnerabilities, and guide development of techniques to cover gaps. The framework includes clear definitions and attributes of bug classes, along with related properties, such as sites, causes, and consequences.

This presents our latest work in three classes of software bugs: buffer overflow (BOF), injection (INJ), and interaction frequency control (IFC). For each class, we show the relation between their proximate and secondary causes, their attributes, and their consequences. For instance, BF reveals that buffer overflows have exactly two proximate causes: data exceeds array (either the programmer made the array too small or tried to use too much data) or wrong index/

pointer out of range. We also provide several examples of applying our "measurement basis" to explain public vulnerabilities, such as heartbleed.

Information structured in the fashion of the Bug Framework (BF) will enable user to more easily determine if two tools find the same sets of bugs, or if they find different, complementary sets. These definitions can serve as a coherent system of units of measurement, enabling more accurate determinations of security.

References:

[1] Nikolai Mansourov and Djenana Campara, "System Assurance: Beyond Detecting Vulnerabilities", pp 175-186, 2011, Morgan Kaufmann – Elsevier.

[2] Yan Wu, Robin A. Gandhi, and Harvey Siy, "Using semantic templates to study vulnerabilities recorded in large software repositories," in Proc. 2010 ICSE Workshop on Software Engineering for Secure Systems, (SESS '10). New York, NY, 2010, pp. 22–28. [Online]. Available: http://doi.acm.org/10.1145/1809100.1809104.

[3] The MITRE Corporation, CWE, Common Weakness Enumeration, http://cwe.mitre.org/

Bio:

Paul E. Black has nearly 20 years of industrial experience in areas such as developing software for IC design and verification, assuring software quality, and managing business data processing. He is now a Computer Scientist for the U.S. National Institute of Standards and

Technology (NIST) near Washington, D.C. The web site he began and edits, the on-line Dictionary of Algorithms and Data Structures, (http://www.nist.gov/dads/) is accessed almost 20,000 times a day from all over the world. He is a member of the Software Quality Group in the Systems and Software Division of the Information Technology Laboratory at NIST.

Dr. Black earned a B.S. in Physics and Mathematics in 1973 and an M.S. in Computer Science in 1983. He began his Ph.D. at UC Berkeley, then transferred to Brigham Young University where he graduated in 1998. Dr. Black has been active in the formal methods research community, and has served as a reviewer for DAC (Design Automation Conference) for several years. He has taught classes at Brigham Young University and Johns Hopkins University. Dr. Black has published in the areas of static analysis, software testing, software configuration control, networks and queuing analysis, formal methods, software verification, quantum computing, and computer forensics. He is a member of ACM and IEEE Computer Society and a senior member of IEEE.

TOWARDS A PROCESS TO FORECAST VULNERABILITY IN SYSTEMS OF SYSTEMS

Joseph Natarian*, Bruce Howard†, Marvin Worst**

**Air Force Research Laboratory,*
***National Air and Space Intelligence Center,*
†Wright State Research Institute

Abstract:

The desire for tomorrow's systems to rapidly collaborate and integrate information from distributed sources has increased the demand for Cyber Physical System (CPS) solutions. These solutions continue to grow in complexity, which currently correlates to larger threat from Cyber-attack. When designing these distributed collaborative CPS, a major challenge is managing the difference between the design and real implementation. These differences, or misalignments, create an aperture for access and an opportunity for Cyber vulnerabilities within underlying components to be exposed for exploitation.

This attack surface is a direct result of insufficient mechanism to identify, measure, and track cyber vulnerabilities, using only system design documents. The current state-of-design function in stovepipes with very little sharing between standalone systems design artifacts and the overall system of systems (SoS). Therefore, data files and formats have limitations in its ability to share that information. For instance, the integrated circuit design process is focused on meeting derived component requirements with little or no intent on scoping and defining requirements for undesired functionality. This is mainly due to the fact that component complexity has increased dramatically with each successive technology node, and that verification of the known good function is a major bottleneck. The inclusion of the undesired functionality would prove difficult and labor intensive because of all the undesirable states.

A solution we are presenting to this challenge is to employ a "Digital Thread" methodology to link digital design tools and representations for design, implementation, and life cycle management to create the ability to identify, measure, and track cyber vulnerabilities from systems engineering artifacts. We envision a new design-space framework that would provide system traceability (requirements, vulnerabilities, abnormalities, and ambiguities), and based on this knowledge provide forecasting of failure causalities for improved security (i.e. reduction of attack surface).

Our solution would create a new systems engineering toolchain to integrate a large corpus of digital design artifacts from system components (also called subsystems) to construct an end-to-end virtual representation of the SoS in an ultra-high fidelity modeling and simulation environment. A probabilistic framework to assess the unified model could then be used to quantify, forecast, and update system performance and capability. This process would provide designers the ability to forecast Cyber vulnerabilities at design time. The proposed solution simultaneously improves Cyber resiliency and drastically reduces costs by providing a new mechanism to assess Cyber vulnerabilities far left in the Systems Engineering 'V'.

The ability to extend the 'V' model allows for the fine tuning of information through an iterative process which leads us on a path towards a Digital Twin. One of the products of the system engineering process is the baseline architecture. The baseline architecture begins the collection of information which is referred to as the Data/Model Repository (DMR). Once all of the design information is gathered, the analyst can setup the initial Design of Experiments (DOE). The analyst runs experiments and analyzes the results in an iterative methodology. The results of analysis lead to the identification of system artifacts. These artifacts are leveraged to generate forecast models of future components within a Digital Twin and integration into the baseline architecture. This information contains the baseline architecture, design and technical documents, physics based models of the system, 3D layout models, manufacturing and tooling capabilities, design tool limitations, and any other source of information about a system or sub-system. The collection of information is critical in the success of a designer's ability to expose underlying faults which are possible for exploitation.

This new toolchain to aid in SoS systems engineering would provide system engineers a mechanism to iterate through environmental, thermal, electromagnetic and manufacturing variables in a Digital Twin of the system and assess apertures for Cyber attack. The ability to model the system leads to the forecasting of failure causalities. The reduction of faults in a system increases security and dependability, while reducing cost and uncertainty.

Bios:

Joseph Natarian (Member, IEEE) received the B.Sc. degree in electrical engineering and computer science from the Wright State University, Dayton, Ohio, in 2008. He is currently pursuing the M.Sc in electrical engineering from the University of Dayton, Dayton, Ohio. Since 2007 he has be supporting the Air Force Research Laboratory (AFRL) in Dayton, Ohio. From 2007 to 2008, he worked for General Dynamics, Advanced Information Systems, where he supported multiple research projects in the Collaborative Interfaces Branch of the Warfighter Interface Division within AFRL. In 2008 he joined the Civil Service as a member of the in-house research team in the Distributed Collaborative Sensor System Technology Branch of the Autonomic Trusted Sensing for Persistent Intelligence Technology Office within AFRL. In 2011 Mr. Natarian took a position in the Advanced Programs Division (AFRL/RYZ), where he is currently a systems engineer. As a researcher, Mr. Natarian collaborates on numerous Defense Advanced Research Project Agency (DARPA) research programs such which explore challenges with architecting and/or integrating complex systems such as trust, tools to evaluate security, and techniques for identifying and traversing threat vectors via control flow and data flow analysis.

Mr. Marvin Worst is the National Air and Space Intelligence Center (NASIC) Command Section's senior integrator for Cyber Intelligence Issues. Serving in this role since 2011, he provides executive-level and line guidance for various special topics and customers involved in the DOD, USAF, and Intelligence Community's cyber enterprise. He also supports research and development effort to establish new capabilities. Mr Worst's began his career at NASIC in 2000 and served in several analytic positions in the RADAR/MASINT and C4-IO disciplines. Prior to joining the US Government, Mr. Worst worked for multiple companies including Motoman Inc., General Electric-Aerospace Division, and Ball Aerospace. There he gained practical engineering experience while designing and developing manufacturing robotics, jet engines, and radar and space software. Mr Worst holds a variety of academic degrees. He completed an AS in Automated Software/Robotics from University of Cincinnati in 1992. He further advanced his educational pursuits by achieving a BS and MS in Computer Engineering from Wright State University in 1996 and 1999, respectively.

Dr. Bruce Howard is the Director of Research and Development at Wright State Research Institute. His current research includes embedded systems security for laboratory analytical equipment, particularly biocyberphysical systems, vulnerability forecasting from incomplete ASIC design information, and computational epigenomics. Prior to joining WSRI, Mr. Howard was the Director of the Center for Nanoscale Engineering at System Planning Corporation, where he developed, demonstrated, and transitioned high risk tailored solutions for a variety of customers. Mr. Howard holds a MS in Systems Engineering, and is currently pursuing a PhD in Computer Science and Engineering with a focus in bioinformatics.

VERIFICATION AND VALIDATION OF AUTONOMOUS SYSTEMS: VERIFIABLE REQUIREMENTS FOR COMPLEX SYSTEMS

Jon Hoffman* and Brian Hulbert**

**Air Force Research Laboratory ,*
***AFRL and LinQuest Corporation*

Abstract:

As the foundations of the 6th generation aircraft are being established, they are being designed to provide more capabilities under architectural constraints that may lead to system limitations. As the complexity grows, the traditional systems engineering methods of verification and validation (V&V) have shown deficiencies that result in cost overruns for aircraft development. In order to mitigate these V&V challenges, the Air Force Research Laboratory Verification and Validation of Complex and Autonomous Systems (VVCAS) Team has leveraged its domain expertise and input from industry, academia, and other government agencies to generate a process to more effectively design, develop, and certify complex systems. It has been observed that exhaustive test of complex and autonomous software systems is intractable and cost prohibitive; however, incorporating formal methods analysis throughout the system design process could provide a means to identify faults as they are introduced and drastically reduce the overall system development cost. In this research, formal methods, such as model checking and limited theorem proving, are applied to the requirements, architecture, and model development phases of the design process of a coupled tanks control system.

Bio:

Jon Hoffman is the Portfolio Lead for the Verification and Validation of Complex and Autonomous Systems (VVCAS) group in the Aerospace Systems Directorate at AFRL. His area of research is in formal analysis of safety critical systems as well as architectures for run-time assurance. He started as a coop student with the team in 2005 and has a BS in Computer Engineering from the University of Cincinnati. His current research interests include early analysis of system requirements, architectures, and models as well as run-time assurance of highly complex and autonomous systems. Early analysis leads to more correct, complete, and clear requirements and prevents errors from leaking to further systems engineering process steps where errors become more costly and time consuming to correct. Run-time Assurance acts as a software fault tolerance system by monitoring for bad or unwanted behavior in highly complex, adaptive, or autonomous systems and provides a simple and safe backup system to revert to when a problem has been found.

VERIFICATION OF DECISION PROCEDURES MODELED IN INTELLIGENT AGENTS

Siddhartha Bhattacharyya*, Marco M. Carvalho*,
Jennifer Davis**, Tom C. Eskridge*

**Florida Institute of Technology , **Rockwell Collins*

Abstract:

Autonomous Intelligent Systems (AIS) are built on principles defined in cognitive architectures that implement adaptive decision procedures. These procedures can be sets of rules with preconditions which, when satisfied, lead to the execution of conditionalized actions. Further, the rules themselves can be adapted based on episodic and semantic learning methods. These AIS design methods prove beneficial in structuring adaptive systems to respond to dynamic situations but fail to assure correctness of the adapted components. These methods cannot diagnose conflicts in the composition of rules during design time or during runtime after adapting to change. We propose a method of transforming a class of adaptive decision procedures into the formalism of formal methods to assure correctness of resulting composition. This research effort discusses a case study that identifies the challenges in the transformation between the cognitive and formal domains. While the presented method enables analysis of the composition of adaptive decision procedures, there are fundamental differences in the representations and constructs in the two domains that remain to be addressed.

Bios

Dr. Siddhartha Bhattacharyya is a research scientist in the application of formal analysis to the design, verification and validation of autonomous systems, smart grid and avionics. He got his Masters from Iowa State University in 2003 and PhD from University of Kentucky in 2005 in Electrical Engineering. He led and conducted research efforts with NASA Langley on Certification Considerations of Adaptive Systems, AFRL on formal verification of quasi-synchronous systems, Loyal Wingman, DARPA on System of Systems Integration Testing and Experimentation, as Sr. Research Engineer at Rockwell Collins. This led to advancing research in application of formal methods to complex systems. He conducted research at Applied Research Laboratory at Pennsylvania State University in the summer of 2004. Here he worked on the design, verification, simulation and synthesis of mission control for autonomous underwater vehicles. Additionally, he conducted research as a summer fellow in 2007 at Oak Ridge National Laboratory. Here he worked on developing methods of verification and validation

of the smart power grid. Presently, he is working on assurance for complex systems. He has served as principal investigator on projects funded through Kentucky Science and Technology Corporation to develop technologies for fault monitoring and diagnosis with wireless sensor networks. He has led and worked on projects for NASA, AFRL, DARPA and ONR in related areas. He has publications and submissions in refereed conferences and journals. He has chaired sessions and presented at American Control Conferences and other conferences. He has been a reviewer for Journals in the area of automation and control. He had been leading efforts in the area of automation and formal methods as the Interim Chair of the Division of Computer Science at Kentucky State University as a faculty from 2005 to 2012.

Marco M. Carvalho is an Associate Professor at the Florida Institute of Technology, and a Research Scientist at the Institute for Human and Machine Cognition. He graduated in Mechanical Engineering at the University Brasilia (UnB – Brazil), where he also completed his M.Sc. in Mechanical Engineering with specialization in dynamic systems and control theory. Marco Carvalho also holds a M.Sc. in Computer Science from the University of West Florida and a Ph.D. in Computer Science from Tulane University, with specialization in Machine Learning and Data Mining. Dr. Carvalho currently leads a several research efforts in the areas of cyber security, moving target defense, critical infrastructure protection, and tactical communication systems, primarily sponsored by the Department of Defense, the U.S. Army Research Laboratory, the U.S. Air Force Research Laboratory, ONR, the National Science Foundation, DoE and Industry.

Dr. Carvalho's research interests include resilient distributed systems, multi-agent systems and emergent approaches to systems optimization and security. As the IHMC Principal Investigator for the Biologically Inspired Tactical Security Infrastructure project, sponsored by ARL, and the Adaptive SCADA Technologies for Critical Infrastructure Protection project, sponsored by the DoE, Dr. Carvalho and his team have worked on the development of agent-based frameworks for adaptive defense and mission resilience.

Dr. Jennifer Davis is a Senior Applied Mathematician in the Advanced Technology Center at Rockwell Collins. Dr. Davis has been working in the field of formal methods for several years. She modeled and verified UAV mission behaviors on an internally funded project. She jointly developed a translator from LLVM to ACL2, to enable verification of system properties. She completed the correctness proofs for the DO-333 theorem proving case study with the PVS theorem prover. She made updates to the formal models and proofs of correctness with PVS for NASA-Langley's Conflict Detection and Resolution tool (a flightplan rerouting tool) called Stratway. Dr. Davis also has experience with cryptography, error-correcting codes, image processing, model-based engineering, and cybersecurity. She earned her Ph.D. in Mathematics at the University of Nebraska at Lincoln in 2007.

Dr. Thomas C. Eskridge is an Associate Professor of Information Assurance and Cybersecurity in the Harris Institute for Assured Information at the Florida Institute of Technology. Dr. Eskridge's research focuses on amplifying human performance through intelligent assistance and innovative visualizations, both of which require developing a deep understanding of operator goals and mental task models to represent, reason, and visually display. He is currently developing tools that enable software agents and human operators to collaboratively represent and reason about networks, user actions, and cyber security events. Previous projects include developing a hybrid connectionist-symbolic knowledge representation system to model human analogical reasoning, case-based reasoning systems supporting milling-machine operators, formal knowledge representation editors, distributed multi-agent systems, fixed-wing and rotary-wing cockpit displays, visualizations for cyber situation awareness, defense posture, and mission management. Dr Eskridge does not currently hold a DoD clearance.

VERIFYING PROGRAMS WITH COMPLEX DATA STRUCTURES USING COQ

Kenneth Roe

The Johns Hopkins University Applied Physics Lab

Abstract:

If one had a tool that could verify that a program correctly maintains data structure invariants, it would be very useful for nding bugs. For example, the Heart bleed bug from a couple of years ago can be traced to an invariant violation. The OpenSLL library made an assumption that packet size information stored in two di erent places was consistent. By sending packets which broke this invariant, a hacker was able to steal critical data. Had a tool existed to verify these invariants, this bug would have been caught before the software was released.

The research presented in this abstract aims at creating a tool for first documenting data structure invariants and second to verify them. We have developed a separation logic based language using the Coq theorem prover. This language is su_fficient to document most useful invariants. We are working on the verification of a simplified version of the DPLL algorithm to demonstrate the utility of the invariants. The code for this algorithm is around 200 lines of C code. The invariant describing relationship between all the data structures is around 100 lines of Coq code. This invariant describes simple relationship such as the relation between an array storing assignments for boolean variables and a linked list storing the same assignments using pairs with the variable number and value. It also describes more complex relationships such as the 2-watch variable algorithm used to quickly identify unit propagations in DPLL.

One of the keys to make completing the proof tractable is to be able to break it into smaller pieces. In order to do this, we needed to add some constructs tour separation logic framework. These constructs make it easier to represent intermediate states where for example, an intermediate state might be "all invariants hold except that one variable has been assigned a new value."

Any Coq user who as attempted a non-trivial proof has found that the process is extremely tedious. The author after analyzing some of his own workflow in developing proofs identified a number of areas in which the proof development process could be improved. One key finding is that of developing a large proof (with many lemmas) often requires many iterations of revisions on the statement of the proof. Developing the proof script often reveals errors in the statement of the proof. Changing the statement then requires the proof to be replayed which is very tedious. As part of the research, we introduce a new IDE, CoqPIE that has all the

functionality of Proof General or Coq IDE plus many new features to deal with work ow issues. For example, the IDE introduces tools to automatically replay and update proof scripts.

Bio:

Kenneth Roe is a PhD student at Johns Hopkins. He returned to graduate school in 2010 after working in the industry for many years. With this work experience, he has a good understanding of the key challenges in developing commercial quality software. This understanding guides his research in formal methods. In addition, Kenneth Roe is an active iOS developer. He has a small business selling iPhone apps. His most successful app, Smart Recorder, has over 1,000,000 device installs and has over 40,000 regular users. He also does iOS development contracts and has many clients.

CONFERENCE DINNER

The conference dinner will be held at the Chart House restaurant on Wednesday, May 6 at 6:30 p.m. Within walking distance of historic downtown Annapolis, Chart House offers fantastic waterfront views of City Dock, the state capital, and the U.S. Naval Academy. Located in the Eastport section of Annapolis, the restaurant has ample parking and is accessible by water taxi. For persons attending the dinner, tickets can be purchased (cash only) at the registration desk.

300 Second Street | Annapolis, MD 21403 | Phone: 410.268.7166

DIRECTIONS FROM THE GOVERNOR CALVERT HOUSE

Driving

Head **northwest** toward **Maryland Ave**
Exit the traffic circle onto **School St**
Turn right onto **Church Circle**
Turn right onto **Duke of Gloucester St**
Slight right onto **Compromise St**
Continue onto **6th St**
Turn left onto **Severn Ave**
Turn left onto **2nd St**
Destination will be on the left

Walking

Head **south** toward **East St**
Exit the traffic circle onto **Francis St**
Turn left onto **Main St**
At the traffic circle, continue straight to stay on **Main St**
Continue onto **Compromise St**
Continue onto **6th St**
Turn left onto **Severn Ave**
Turn left onto **2nd St**
Destination will be on the left

LOCAL RESTAURANTS

LOCAL RESTAURANTS

Armadillo's Bar & Grill – 132 Dock Street, Annapolis, MD 21401

Veteran American grill offering burgers & beers along with dock views & occasional live music.

Blackwall Hitch – 400 6th Street, Annapolis, MD 21403

Upscale-casual New American restaurant featuring outside seating, an on-site pub & stylish decor.

Cantler's Riverside Inn – 458 Forest Beach Road, Annapolis, MD 21401

The crabs are top notch, the view is without parallel.

Chick and Ruths Delly – 165 Main Street, Annapolis, MD 21401

Lively landmark diner featuring greasy-spoon breakfasts & piled-high sandwiches in kitschy environs.

Cracker Barrel – 115 Blue Jay Court, Stevensville, MD 21666

Brad's favorite. Ask for the secret wine list!

Davis' Pub – 400 Chester Avenue, Annapolis, MD 21403

Featured on Diners, Drive-Ins and Dives. Try the crab pretzel!

Dock Street Bar & Grill – 136 Dock Street, Annapolis, MD 21401

Chesapeake Bay cuisine served daily until 1 a.m.

Dry 85 – 200 Main Street, 193 B Main Street, Annapolis, MD 21401

A modern industrial take on a Prohibition-era speakeasy.

Harry Browne's – 66 State Circle, Annapolis, MD 21401

A captivating historic restaurant/lounge. Lavish lunches, divine dinners and sumptuous Sunday brunch.

Harvest – 26 Market Space, Annapolis, MD 21401

A casual dining and tap room in downtown Annapolis.

Iron Rooster – 12 Market Space, Annapolis, MD 21401

Creative all-day breakfast menu and American comfort food.

Joss Café & Sushi Bar – 1959 Main Street, Annapolis, MD 21401

Voted 'Best Sushi Restaurant' in Annapolis for 8 years running by the readers of What's Up? Magazine.

Lemongrass – 167 West St., Annapolis, MD 21401

Fresh, authentic Thai Cuisine in a warm contemporary environment.

LOCAL RESTAURANTS

Level – A Small Plates Lounge – 69 West St., Annapolis, MD 21401

Easygoing, wood-accented haunt offering eco-friendly New American small plates & creative cocktails.

Metropolitan Kitchen and Lounge – 169 West St., Annapolis, MD 21401

Casual restaurant with a broad American menu & full bar plus rooftop deck & local live music.

O'Brien's Oyster Bar – 113 Main St., Annapolis., MD 21401

Imaginative seafood dishes and nouveau American cuisine. Dancing and live entertainment nightly."

Osteria 177 – 177 Main St., Annapolis., MD 21401

One of Annapolis' premier dining spots, thriving on the taste and passion that encompasses Italian coastal cuisine.

Preserve – 164 Main Street, Annapolis, MD 21401

Preserve is a casual American restaurant using sustainable and local products.

Purple Thread Café – 137 Prince George Street, Annapolis, MD 21401

Asian fusion restaurant featuring banh mi's and bubble tea.

Pusser's Carribean Grill – 80 Compromise St., Annapolis, MD 21401

Waterfront location with a beautiful view.

Rams Head Tavern – 33 West St., Annapolis, MD 21401

An Annapolis landmark since 1989!

Sofi's Crepes – 1 Craig Street, Annapolis, MD 21401

Brunch crepes with sweet and savory fillings.

Tsunami Sushi Bar and Lounge – 51 West St., Annapolis, MD 21401

Upbeat modern Asian-fusion eatery & lounge, serving sushi, steak & seafood, with creative cocktails.

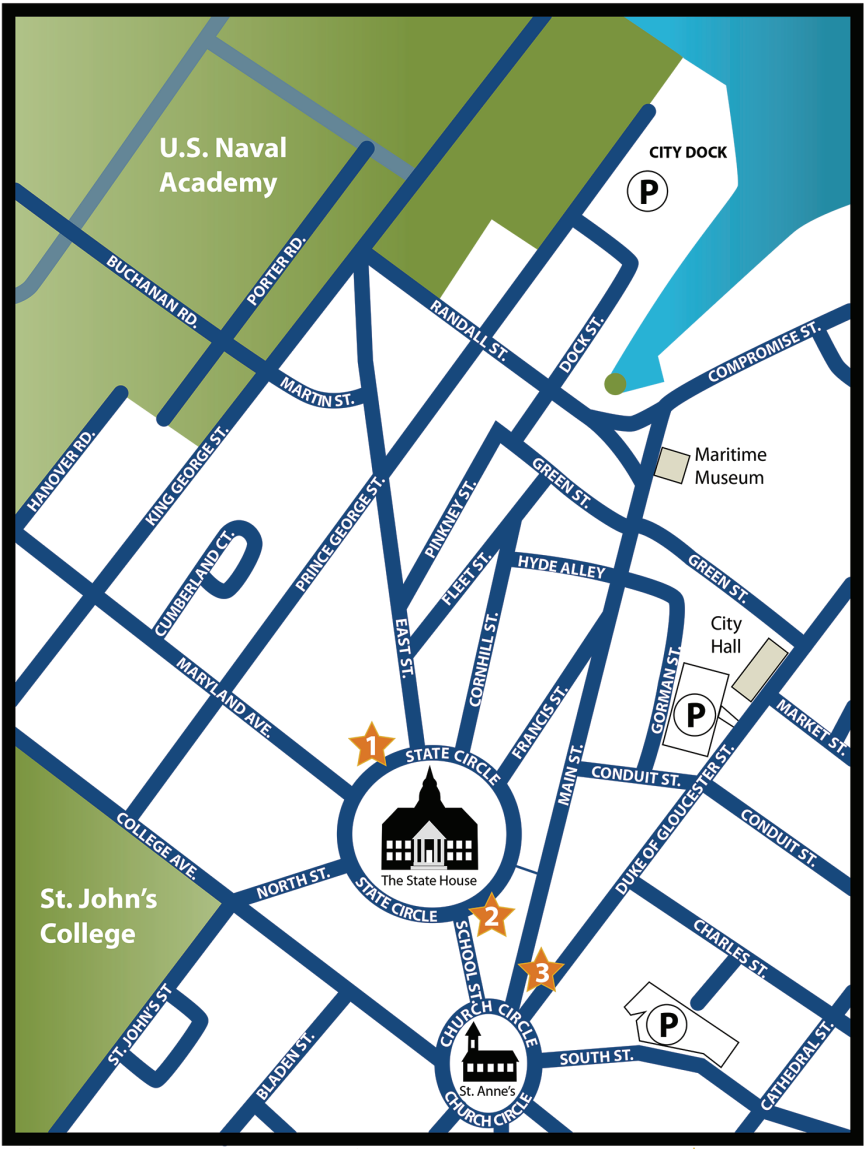
VIDA Taco Bar – 200 Main Street, Annapolis, MD 21401

VIDA Taco Bar offers the freshest, street food style tacos in a fun, hip, and energetic atmosphere.

Vin 909 – 909 Bay Ridge Ave., Annapolis, MD 21403

Great food. Great wine. Excellent service. It's worth the walk, even in the rain.

NOTES


U.S. Naval Academy
CITY DOCK
P
BUCHANAN RD.
PORTER RD.
RANDALL ST.
DOCK ST.
COMPROMISE ST.
MARTIN ST.
HANOVER RD.
KING GEORGE ST.
Maritime Museum
GREEN ST.
PRINCE GEORGE ST.
PINKNEY ST.
CUMBERLAND CT.
HYDE ALLEY
FLEET ST.
GREEN ST.
City Hall
EAST ST.
CORNHILL ST.
GORMAN ST.
MARYLAND AVE.
FRANCIS ST.
MARKET ST.
1
STATE CIRCLE
MAIN ST.
CONDUIT ST.
DUKE OF GLOUCESTER ST.
CONDUIT ST.
COLLEGE AVE.
NORTH ST.
The State House
STATE CIRCLE
St. John's College
SCHOOL ST.
2
CHARLES ST.
3
CHURCH CIRCLE
P
ST. JOHN'S ST.
BLADEN ST.
SOUTH ST.
CATHEDRAL ST.
St. Anne's
CHURCH CIRCLE
1 Governor Calvert House
2 Robert Johnson House
3 Maryland Inn