Systems, Networking, and Cybersecurity
Qualifying Exam
Spring, 2016

Distributed: January 11, 2016 (12:00PM)
Due: January 25, 2016 (11:59PM)

Honor Code. This examination is conducted under the University's Graduate Honor System Code. Students are encouraged to draw from other papers than those listed in the exam to the extent that this strengthens their arguments. However, the answers submitted must represent the sole and complete work of the student submitting the answers. Material substantially derived from other works, whether published in print or found on the web, must be explicitly and fully cited. Note that your grade will be more strongly influenced by arguments you make rather than arguments you quote or cite.

Written answers. The answers to the questions on this exam must be submitted no later than the due date listed above. Answers must be submitted in a single PDF document emailed to the exam coordinator (Dongyoon Lee, dongyoon@vt.edu).

Oral Exam. The written exam will be followed by an oral exam, where the student is expected to defend his/her solutions. Unless specifically requested, the student is not expected to make a formal presentation. In the oral exams, faculty may ask questions about any paper in the reading list to assess the student's understanding of the subject. Oral exams will be scheduled individually for each student.

Assessment. After the oral examination, the examining faculty will determine the student's score for the examination process. The score is between 0 – 3 points, depending on the student's performance on both the written and oral components. These points may be applied toward the total score of 6 points necessary to qualify for the Ph.D. The assessment criteria, as defined by GPC, are as follows:

- 3: Excellent performance, beyond that normally expected or required for a PhD student.
- 2: Performance appropriate for PhD-level work. Prime factors for assessment include being able to distinguish good work from poor work, and explain why; being able to synthesize the body of work into an assessment of the state-of-the-art on a problem (as indicated by the collection of papers); being able to identify open problems and suggest future work.
- 1: While the student adequately understands the content of the work, the student is deficient in one or more of the factors listed for assessment under score value of 2. A score of 1 is the minimum necessary for an MS-level pass.
- 0: Student's performance is such that the committee considers the student unable to do PhD-level work in Computer Science.

Systems, Networking, and Cybersecurity
Qualifying Exam
Spring, 2016

Questions on the paper "Failure sketching: a technique for automated root cause diagnosis of in-production failures"

1. For static slice computation, Gist uses the thread interprocedural control flow graph (TICFG) [75] of the program. Explain what could be go wrong if Gist does not track control flow that is implicitly created via thread creation and join operations.

2. To trace the control flow, Gist leverages Intel Processor Trace equipped with the latest 6^{th} generation processors (Broadwell/Skylake processors). Is this hardware support necessary? Can we use the order form of hardware branch trace facilities such as Last Branch Recorder (LBR) or Branch Trace Store (BTS) that are supported by older $4^{th}/5^{th}$ generation processors?

3. Gist's failure sketch engine gathers execution information from failing and successful runs, and then it determines the difference between failing and successful run. Discuss what could be go wrong if a program failure happens so rarely and thus the system were not able to gather enough number of failure runs.

4. For root cause diagnosis, Gist follows a similar approach to previous cooperative bug isolation works [4, 25, 38], which use statistical methods to correlate failure predictors to failures in programs. Describe example cases in which such statistical methods do not work well.

Systems, Networking, and Cybersecurity
Qualifying Exam
Spring, 2016

Questions on the paper "Detecting Covert Timing Channels with Time-Deterministic Replay"

1. Discuss why Sanity uses deterministic multithreading (Section 3.2)? Discussion should include what could go wrong if deterministic multithreading is not used.

2. Discuss why Sanity flushes the TLB (Section 3.6)? Discussion should include what could go wrong if Sanity does not flush the TLB.

3. Explain why and how Sanity ensures the same virtual memory layout during play and replay.

4. Sanity prototype implements a single-core JVM. Discuss what additional supports are needed to extend it to support multi-core JVM in which multithreaded Java programs can take advantage of multicores.

Systems, Networking, and Cybersecurity
Qualifying Exam
Spring, 2016

Questions on the paper "Invyswell: A Hybrid Transactional Memory for Haswell's Restricted Transactional Memory"

1. The current generation of processors including Intel's Haswell processor only support "best-effort" hardware transactional memory, providing no progress guarantee. This limitation motivates Hybrid TM approach. Discuss challenges in supporting "ideal" HTM in which transactions get aborted only if there is a data conflict, and transactions provide progress guarantee.

2. Using *hw_post_commit* counter, Invyswell prevents SpecSW from reading inconsistent values written by the BFHW, ensuring opacity [12]. Explain why transactional memory system should provide opacity.

3. Suppose a SpecSW transaction and a BFHW transaction are running concurrently, and there is a data conflict between those two transactions. Explain cases in which 1) SpecSW gets aborted, and 2) BFHW gets aborted.

4. Because LiteHWs do not maintain read or write set metadata, if a software transaction is in-flight when a LiteHW enters its commit phase, Invyswell must assume a conflict exists between the LiteHW and the software transaction and, therefore, must abort the LiteHW. Does this mean that LiteHW does not allow concurrent software transactions? Is it possible for concurrent software transactions to commit while LiteHW is running?

Systems, Networking, and Cybersecurity
Qualifying Exam
Spring, 2016

Questions on the paper "Improving Reliability with Dynamic Syndrome Allocation in Intelligent Software Defined Data Centers"

1. The paper argues that "existing data centers can dramatically improve their reliability simply by implementing novel middle-ware processes." Is this claim justified by the paper or bogus hand waving? How will you expect data center engineers to interpret the nine observations offered by the paper to design better solutions? Comment on the practicality of such solutions.

Systems, Networking, and Cybersecurity
Qualifying Exam
Spring, 2016

Questions on the paper "The Scalable Commutativity Rule: Designing Scalable Software for Multicore Processors"

1. What are the essential features of an application that enable better scaling at large scale compared to the regular APIs? Discuss how the approach can be applied to a stream processing system such as SPARK? Will the new setup scale?

Systems, Networking, and Cybersecurity
Qualifying Exam
Spring, 2016

Questions on the paper "Chaos: scale-out graph processing from secondary storage"

1. Comment on the experimental methodology employed and compare and contrast with the validation approaches adopted in previous two papers ("Improving Reliability with Dynamic Syndrome Allocation in Intelligent Software Defined Data Centers" and "The Scalable Commutativity Rule: Designing Scalable Software for Multicore Processors").

Systems, Networking, and Cybersecurity
Qualifying Exam
Spring, 2016

Questions on the paper "Free Launch: Optimizing GPU Dynamic Kernel Launches through Thread Reuse"

1. The way GPU achieves its high performance is different from the way CPU does. Please first briefly explain how GPU's execution model hides the delay of long latency operations, and then discuss why such an execution model makes the current hardware-based dynamic parallelism heavyweight in terms of the runtime overhead.

2. For the sake of the argument, let's assume your application has no use of GPU shared memory, i.e., your GPU lacks shared memory. Do you think that the assumption can make the hardware-based dynamic parallelism a lightweight approach? Please justify your answer.

3. The compiler transformation guarantees that the total number of persistent thread blocks is no larger than the maximum number of thread blocks that can concurrently run on a GPU. Why is this necessary? Please explain how the lack of such guarantee makes it impossible to synchronize across all thread blocks.

Systems, Networking, and Cybersecurity
Qualifying Exam
Spring, 2016

Questions on the paper "Exploring and Enforcing Security Guarantees via Program Dependence Graphs"

1. For the code snippet in Figure 2 (a), answer the following two PIDGINQL expressions:

    a. pgm.findPCNodes(isPassRet, TRUE)

    b. pgm.findPCNodes(isAdRet, TRUE)

where isPassRet and isAdRet are set as pgm.returnsOf("checkPassword") and pgm.returnsOf("isAdmin"), respectively.

2. As shown in Figure 2 (a), the code has three statements, i.e., 1, 2, 3. Which other statement is the "statement 3" control-dependent on? Please justify your answer.

3. Please compare the PDG based security check approach to dynamic tainting based approach, and discuss what are the pros and cons of each approach.

Systems, Networking, and Cybersecurity
Qualifying Exam
Spring, 2016

Questions on the paper "Per-Input Control-Flow Integrity"

1. What is the main difficulty of building precise CFG on the presence of many function pointers? Explain how that impacts the analysis precision in the detection of security attacks.

2. It turns out that even a fine-grained CFI with unlimited number of tags and a shadow stack is ineffective in protecting against malicious attacks, mainly due to the imprecision of the pointer analysis used. This means that the underlying SCFG used in the paper can be exploited to launch such malicious attacks. Please explain how the attacks can be launched from the attacker's perspective to thwart the proposed PICFG, and discuss how easy (difficult) the exploit would be with at least two examples, i.e., easy-to-exploit application and hard-to-exploit one.

3. Please explain how the paper leverages idempotence to update the ECFG in a safe and efficient manner. You're supposed to define the meaning of the idempotence used in the paper and discuss how it is achieved.