Systems, Networking, and Cybersecurity
Qualifying Exam
Spring, 2015

Distributed: January 12, 2015 (12:00PM)
Due: January 26, 2015 (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 (Dennis Kafura, kafura@cs.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, 2015

Questions on the paper "Eidetic Systems"

1. Arnold uses deterministic record and replay to reproduce the architectural state (register and address space) of user-level processes. When deterministic replay is supported by operating system as in Arnold, explain what kind of non-determinism needs to be recorded to enable deterministic replay of user-level processes in the following cases. Describe minimal OS support to reduce performance overhead.

   a. To replay a single threaded process running on uniprocessor

   b. To replay a multithreaded process running on uniprocessor (assuming no simultaneous multithreading (SMT))

   c. To replay a data-race-free multithreaded process running on shared-memory multiprocessor

   d. To replay an arbitrary multithreaded process (that may have data race) running on shared-memory multiprocessor

2. Deterministic replay also allows Arnold to defer the work needed to track lineage from the time of execution to the time of querying. Explain (a) why users want to do this (i.e., performing offline analysis); and (b) how this is possible using deterministic replay.

3. Arnold applied an optimization technique in recording messages sent from the X server in order to reduce the size of its replay log (Section 4.2). The optimization is based on the observation that with the exception of actual user input, the behavior of the X server is mostly deterministic. Explain what may go wrong if the X server is not deterministic (if an assumption is not true).

4. In the backward query case study (Section 6.4.1), the paper mentioned that the query reports four false positives. Explain (a) what could cause false positives in Arnold; and (b) how Arnold tried to provide a precise answer (less false positives). Moreover, describe (c) whether Arnold may report wrong answer (false negatives). Justify your answer clearly.

5. Arnold is claimed to target personal computers and workstations. Can the proposed system be applied to large distributed systems such as MPI-based high performance parallel computing, distributed large-scale graph processing, etc. Pick one example of large distributed system and describe what could be the problem when the unmodified Arnold is applied.

Systems, Networking, and Cybersecurity
Qualifying Exam
Spring, 2015

Questions on the paper "SKI: Exposing Kernel Concurrency Bugs through Systematic Schedule Exploration"

1. SKI pins threads to virtual CPU and suspends/resumes the corresponding virtual CPU's execution of machine instructions in order to control the progress of threads (Section 3.2). However, there are kernel threads that cannot be pinned to CPUs for OS-specific reasons. Describe how this limitation would affect the SKI's systematic schedule exploration.

2. To explore the interleaving space, SKI requires information about whether threads are blocked or not (Section 3.3). Explain (a) why SKI needs to infer thread liveness; (b) why it is hard to infer threads liveness in SKI; and (c) how (imprecise) heuristic based approach would affect the SKI's systematic schedule exploration.

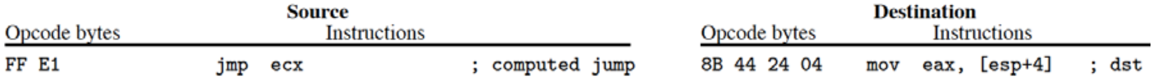
3. One naïve scheduling algorithm for systematic schedule exploration is to run one context (thread) for the longest possible period until it is no longer able to run, and then choose next context to run based on the priority of context (Section 3.4). (a) Explain the case in which a context (thread) becomes no longer able to run. (b) Illustrate the limitation of this naïve approach with an example in terms of exposing concurrency errors.

4. In Section 6.2.1, the paper mentioned that it is hard to separate reported data races into benign and harmful one. (a) Provide an example of benign data race; and (b) explain why it is hard to separate them.

5. In Section 7, the paper discusses some of the implications of VMM-based scheduling. Explain why VMM-based approach may not expose observable behaviors of hardware with weaker memory consistency models.

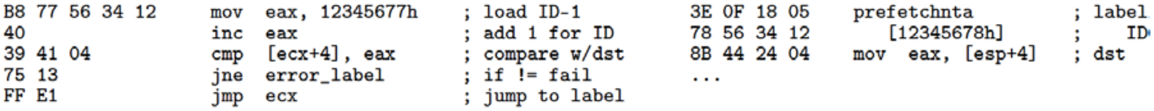
Systems, Networking, and Cybersecurity
Qualifying Exam
Spring, 2015

Questions on the paper "Control-flow integrity principles, implementations, and applications"



```
                      Source                                         Destination
Opcode bytes              Instructions                   Opcode bytes          Instructions
FF E1            jmp  ecx             ; computed jump    8B 44 24 04    mov  eax, [esp+4]   ; dst
```

1. The code above is instrumented like below to enforce CFI. Explain and justify the way the instrumentation code works.



```
B8 77 56 34 12     mov  eax, 12345677h    ; load ID-1       3E 0F 18 05     prefetchnta          ; label
40                 inc  eax               ; add 1 for ID    78 56 34 12        [12345678h]       ;    ID
39 41 04           cmp  [ecx+4], eax      ; compare w/dst   8B 44 24 04     mov  eax, [esp+4]    ; dst
75 13              jne  error_label       ; if != fail      ...
FF E1              jmp  ecx               ; jump to label
```

2. CFI enforcement can be realized with or without source code. What would be the pros and cons of using source code?

3. In Fig 1 of the paper, ‘ret 55’ has two outgoing edges and their destinations have the same label ID. Explain the issue here, its cause, and the way the paper leverages to get around it. Assuming source code availability, what static analysis would you leverage to resolve the issue? Justify your answer with 5-6 sentences.

4. 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.

5. Explain the main idea of THEOREM in the formal study section of the paper (Section 6.1), and how that affects the trustworthiness of CFI enforcement.

Systems, Networking, and Cybersecurity
Qualifying Exam
Spring, 2015

Questions on the paper "Towards optimization-safe systems: Analyzing the impact of undefined behavior"

1. Provide the definition of ‘unstable code’ and explain why it is valid to optimize away such code from the compiler’s point of view.

2. The paper says “any fragment that is reachable **only** by inputs that trigger undefined behavior is unstable code”. Explain why ‘only’ matters here to identify unstable code. (Hint: understanding “Theorem 1” should help)

3. The last sentence of the first paragraph in Section 3.1 seems to mismatch “Definition 1”. How do you think about it? If you think the sentence is not correct, please fix it. Otherwise, please explain what the sentence really means.

4. What are the sources of false positives/negatives in the identification of unstable code? For each source, explain how it affects the analysis precision.

5. Explain the role of the SMT solver over the course of unstable code identification as well as pros and cons of using the solver.

6. Briefly explain the idea behind the dominator based approximation of solver queries, and why such approximation still correctly identifies unstable code.

Questions on the paper “Arrakis: The Operating System is the Control Plane”

1. On page 2 of the paper the claim is made that "Arrakis eliminates scheduling and kernel crossing overhead entirely, because packets are delivered directly to user space." Why is the scheduling cost eliminated? Some network stack processing still has to be done (albeit in user space). But doesn't the user space process need to be running for this to occur?

2. In Figure 3 the VNIC and VSIC buffers are in kernel space while the libos is in user space. Movement of data between user and kernel space implies some form of kernel crossing overhead. However, Arrakis claims to eliminate kernel crossing overhead. Explain.

3. Suppose that you wanted to implement using Arrakis a one-way interprocess communication channel where a process could send message to another process on the same machine(though possibly on a different core). Describe the key strategies that you would use to implement this feature using Arrakis.

Questions on the paper “Pebbles: Fine-Grained Data Management Abstractions for Modern Operating Systems”

1. The authors write at the beginning of Section 3: "Specifically, we hypothesize that the inclusion of high-level storage abstractions, such as the SQLite database in Android or the CoreData abstraction in iOS, has created a new “narrow waist” for storage abstractions that largely hides the traditional hierarchical file system abstraction." Explain what is meant by the idea of a "narrow waist" and its relevance to this work. What specific assumption is made in the paper on the basis of the "narrow waist" belief.

2. Suppose an application uses only a key-value store for its data. The keys are held in a persisted hash table and the values are separate files. Considering, the algorithm in Figure 5, describe the implications this storage structure has on the ability of Pebbles to identify the LDOs for this application.

3. The HideIt application developed with Pebbles demonstrates the feasibility of hiding and later restoring a user selected object. Can this be extended so that users could request the hiding of all objects with a certain property (e.g., hide all emails from a given sender or hide all photos taken on a certain date)? Elaborate the reasons for your answer.