REVIRT: ENABLING INTRUSION ANALYSIS THROUGH VIRTUAL-MACHINE LOGGING AND REPLAY

GEORGE DUNLAP, SAMUEL KING, MURTAZA BASRAI, PETER CHEN OSDI’02

1. What is the motivation for this work? What two problems do current system loggers have?
2. What is the basic approach of ReVirt? How is OS-on-OS defined? Direct-on-host?
3. Is UMLinux a Type-1 or Type-2 hypervisor? What interface does UMLinux export?
4. How were privilege levels managed in Disco? How are they managed in UMLinux?
5. How were system calls managed in Disco? How are they managed in UMLinux? How are timer, interrupts, and memory exceptions handled?
6. What is the TCB in ReVirt? What would it be for a Type-1 system? Why or why not is the TCB in ReVirt smaller than the original OS? Do you agree with their argument?
7. What general information must be logged to replay a system from a checkpoint? Do all non-deterministic events need to be logged? Does all external input need to be logged? Which external input could be most problematic and how do the authors suggest handling this? What happens to output to peripherals? Do you think it could it be useful to do something different with output? What would one need to do to run ReVirt on a multiprocessor?
8. What must be logged to re-deliver signals at **same point** during replay? Why is each needed? How are hardware performance counters used? How does ReVirt deliver interrupts at exact right point? Why does this approach lead to good performance during replay?
9. Why is it difficult to handle non-deterministic instructions? What are two examples of such instructions? How does ReVirt handle each? Do you think there is a better solution?
10. How much overhead does UMLinux virtualization add? Why does the overhead vary so much across workloads? Do you agree with the statement: “an overhead of 58% is not prohibitive for sites that value security”? What was the overhead of Disco? for Vmware Workstation 3.1?
11. Is the runtime with logging reasonable? Log growth rate reasonable? Why is some log growth so much higher? Is replay runtime reasonable? Is their metric reasonable? Why can replay be faster than actual time?
12. How do you think they should show the **usefulness** of ReVirt? What do they do?
13. Do you have other questions you’d like answered in their evaluation?
14. Conclusions?