Critique on

Erik Buchanan, Ryan Roemer, Hovav Shacham, and Stefan Savage. 2008. When Good Instructions Go Bad: Generalizing Return Oriented Programming to RISC.

Astha Sharma
asharma6@uno.edu

SUMMARY:
The paper presented here suggests that the Return Oriented Programming (ROP) threats considered by Shacham for x86 system architecture holds true for SPARC, a fixed instruction length RISC architecture with structured control flow. In fact, the threat/vulnerability is not limited to the x86 architecture or any particular OS, but can be exploited readily and also bypasses an entire category of malware protections. The authors were successfully able to create a machine computable library of code gadget using parts of Solaris libc (a general purpose programming language) and a compiler for constructing return-oriented exploits. All in all, this approach is said to provide a simple bypass for the majority of exploitation mitigations.

STRENGTHS:

- It puts a light on the general misconception that preventing the introduction of malicious code is sufficient to prevent the introduction of malicious computation.
- The paper gives a better analysis on the scope of ROP threats.
- Simplifies Shacham's complex and laborious approach to a generic gadget exploit API, scripting language, and exploit compiler that supports simple general purpose ROP.

WEAKNESSES:

- Since the authors have only used codes from libc library, the system and functionality may not be familiar to many programmers.
- With the proposed defense mechanism, there maybe a need to introduce return oriented payload by some other means than the stack overflow, like heap corruption, format string vulnerability, etc.
- Solaris does not implement Address Space Layout Randomization on SPARC.