

Vulnerability Discovery

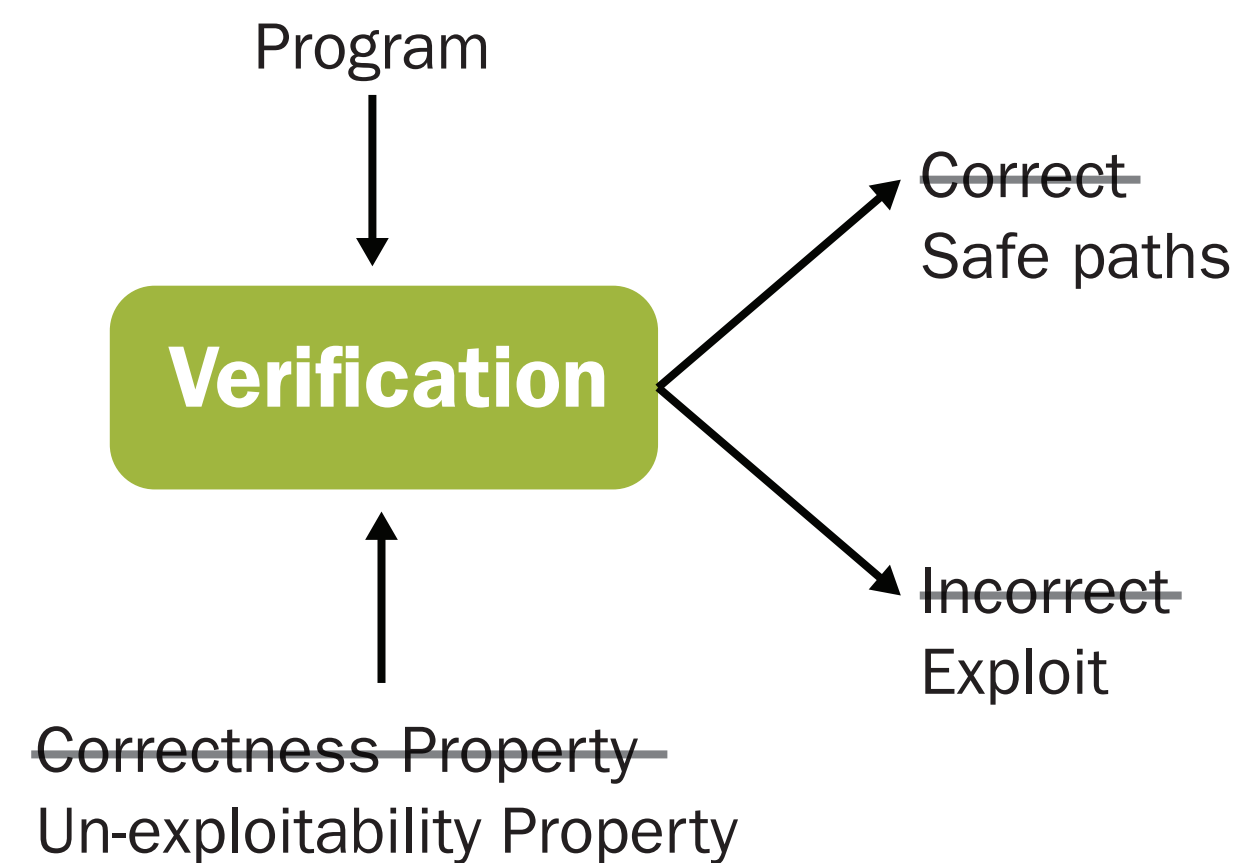
Task 1: Automated and Sound Vulnerability Discovery

Vision: Automatically check DoD software systems for exploitable bugs

Discover vulnerabilities automatically in compiled x86 applications

- “Zero false positives”
- Automatically generate an exploit for each vulnerability; no source code necessary

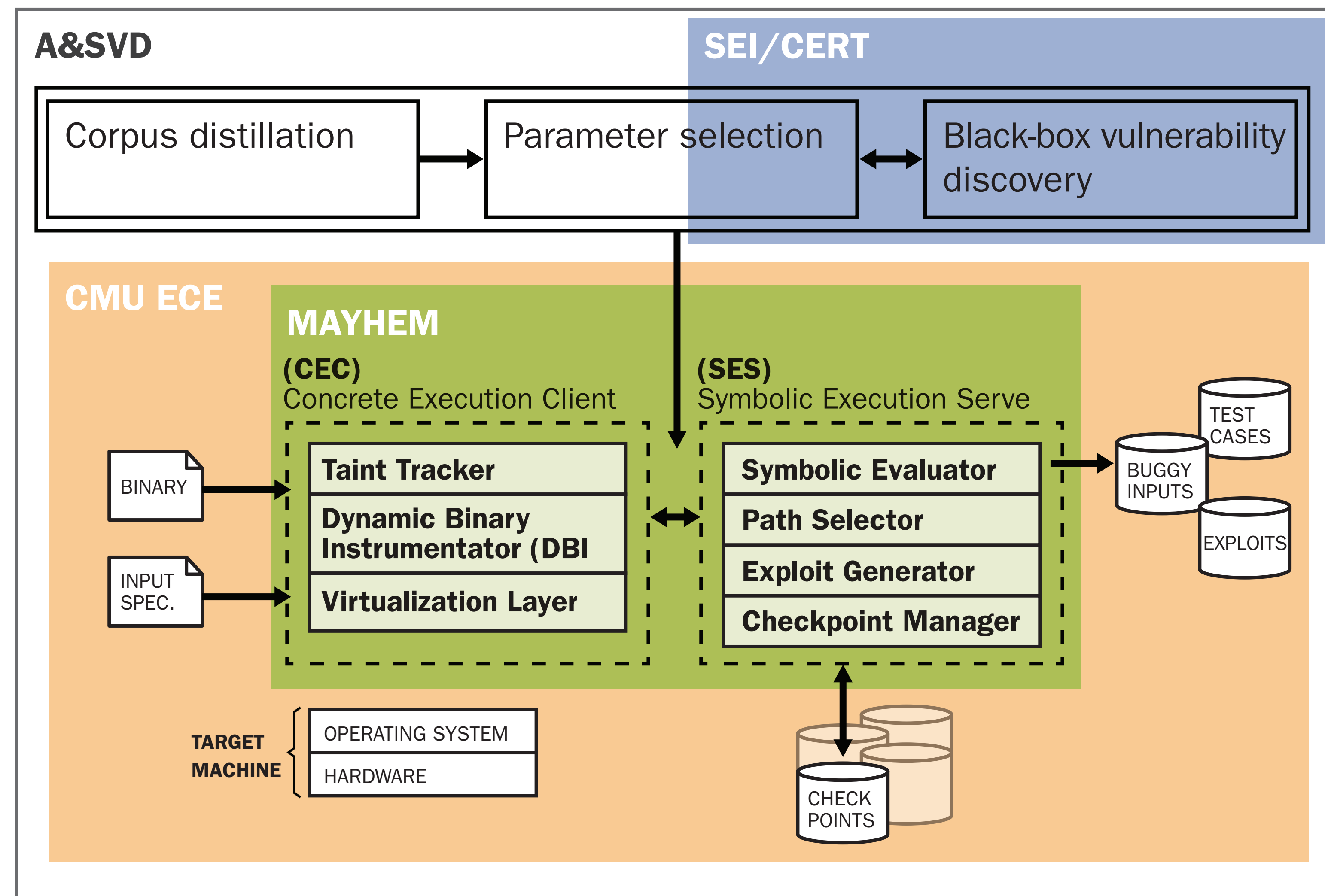
This project combines expertise from both SEI/CERT and CMU



Task 2: Low-Power Low-Bandwidth System Vulnerabilities

Focuses on vulnerability discovery in Things that have enough compute power to pose a threat to a network they are attached to, but not enough for somebody to think of them as a computer.

1. How do vulnerability discovery techniques for LPLB systems differ from those for traditional computing systems? Are there techniques that do not transfer well? Are there techniques that have shown efficacy in traditional computing but have not showed up on the LPLB side? Why?
2. What processes, tools, and techniques are most effective at improving the security of LPLB systems? For developers and creators of such systems For acquirers, deployers, and operators of such systems
3. What metrics can be applied to assess the efficacy and/or efficiency of those processes?



Copyright 2014 Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

This material has been approved for public release and unlimited distribution except as restricted below.

Internal use:* Permission to reproduce this material and to prepare derivative works from this material for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use:* This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other external and/or commercial use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

* These restrictions do not apply to U.S. government entities.

Carnegie Mellon[®], CERT[®] and CERT Coordination Center[®] are registered marks of Carnegie Mellon University.

DM-0001812