# Real-Time Streaming Data Processing Engine for Embedded Systems

## Overview and Problem Statement

The objective of this activity is to develop a low-cost and low-overhead hardware security engine to achieve secure and reliable execution of applications that compute critical data, in spite of potential hardware and software vulnerabilities. To achieve that goal, the barriers to application of the security engine need to be eliminated. Specifically, the security engine needs to achieve low runtime overhead, low hardware resource overhead, high source compatibility, and high binary compatibility.

## Research Objectives

- Prevent attacks from different entry points (outsiders, normal users, or insiders).
- Enforce both *spatial memory safety* and *temporal memory safety* at the same time to provide high detection coverage of memory corruption attacks.
- Efficiently transmit monitoring data collected from the main processor to the security engine so that transmission overhead is low.
- *Asynchronously* check the memory safety without interrupting the normal execution of a program.
- Apply the protection technique on existing applications with minimum involvement on the developer's side to convert unprotected programs into protected programs.
- **Smart Grid Application Area:** Apply AHEMS on the data concentrator or security gateway to protect the integrity of critical data, such as password, private key, or power grid data.

## Technical Description and Solution Approach

- The AHEMS (Asynchronously Hardware-Enforced Memory Safety) Framework (See Figure 1) is proposed to protect applications from memory corruption attacks by enforcing spatial and temporal memory safety. AHEMS has two major parts:
  - **Source Code Instrumentation:** The source code of a program is instrumented with `alloc` and `dealloc` instructions to establish the interface between the program and the hardware security engine.
  - **Hardware Security Engine:** The security engine receives the memory events from the runtime monitor, checks the memory safety *asynchronously* (i.e., does not stop the main processor) using the metadata stored on the security engine, and raises exceptions if those memory events violate the memory safety according to the metadata.

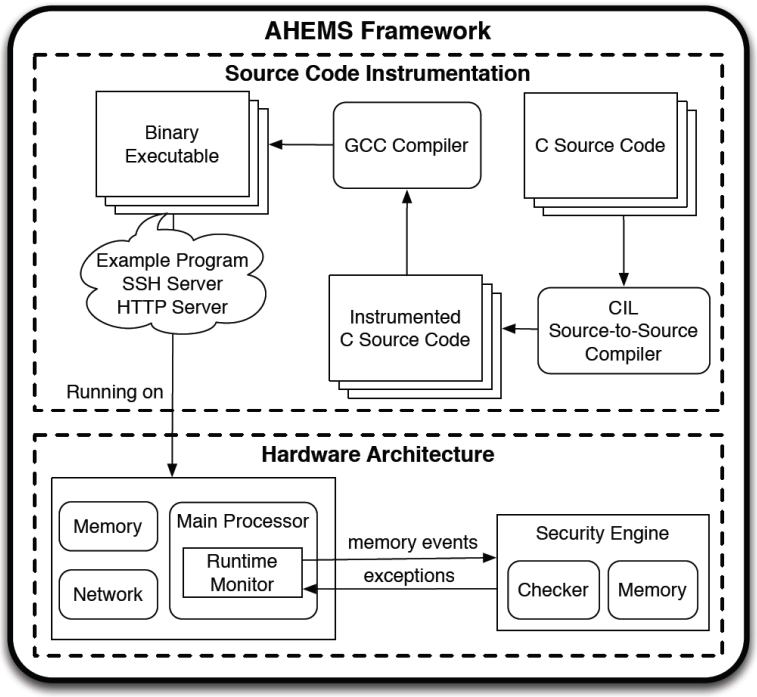


**Figure 1**: The Architecture of the AHEMS Framework

## Results and Benefits

- Our prototype of AHEMS achieves high coverage for memory corruption attacks (detecting 676 out of 677 test cases; see Table 1) with runtime overhead as low as 10.6%. It also outperforms four other state-of-the-art approaches in terms of runtime overhead (See Table 2).

**Table 1:** Detection Coverage of AHEMS on Juliet Test Suite

| **Spatial Memory Errors** | | | |
|---|---|---|---|
| **CWE No.** | **Description** | **Tested** | **Detected** |
| CWE121 | Stack-based Buffer Overflow | 209 | 208 |
| CWE122 | Heap-based Buffer Overflow | 18 | 18 |
| CWE124 | Buffer Underwrite | 102 | 102 |
| CWE126 | Buffer Overread | 145 | 145 |
| CWE127 | Buffer Underread | 33 | 33 |
| CWE588 | Attempt to Access Child of Non-structure Pointer | 34 | 34 |
| CWE680 | Integer Overflow to Buffer Overflow | 38 | 38 |
| CWE761 | Free Pointer Not at Start of Buffer | 38 | 38 |
| **Subtotal** | | **617** | **616** |
| **Temporal Memory Errors** | | | |
| **CWE No.** | **Description** | **Tested** | **Detected** |
| CWE415 | Double-free | 38 | 38 |
| CWE416 | Use-after-free | 20 | 20 |
| CWE562 | Return of Stack Variable Address | 2 | 2 |
| **Subtotal** | | **60** | **60** |
| **Total** | | **677** | **676** |

**Table 2:** Runtime Overhead of AHEMS against Four Other Approaches

| **Programs** | **AHEMS** | **Mudflap** | **Softbound+CETS** | **SAFECode** | **AddressSanitizer** |
|---|---|---|---|---|---|
| Bh | 0.2% | 13655.2% | Compiler error | Runtime error | 41.9% |
| bisort | 38.4% | 3114% | 341.1% | 154.3% | 74.7% |
| em3d | 10.5% | 705.0% | 473.0% | 192.1% | 89.5% |
| health | 17% | 13343.9% | 737.8% | 943.2% | 361.5% |
| Mst | 4.3% | 1169.2% | 395.1% | 644.1% | 92.2% |
| perimeter | 22.2% | 32317.8% | 443.1% | 212.7% | 142.8% |
| power | 0.0% | 263.9% | 1.2% | 1.0% | 2.7% |
| treeadd | 8.6% | 106008.1% | 448.1% | 518.8% | 398.7% |
| tsp | 0.0% | 1404.9% | 206.9% | 57.5% | 76.8% |
| voronoi | 4.6% | 2224.2% | False alarm | Runtime error | 156.5% |
| **Average** | **10.6%** | **17420.6%** | **380.8%** | **340.5%** | **143.7%** |

- **Technology Readiness Level:** We have implemented a prototype of AHEMS that can work on medium-size applications such as Olden Benchmarks.

## Researchers


- Kuan-Yu Tseng, mycallmax@gmail.com
- Zbigniew Kalbarczyk, kalbarcz@illinois.edu
- Ravishankar Iyer, rkiyer@illinois.edu


## Industry Collaboration

- Dennis Gammel, Schweitzer Engineering Laboratories Inc.


funded by the U.S. Department of Energy and the U.S. Department of Homeland Security