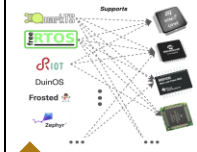
LeMix : Rehosting Embedded Applications as Linux Applications For Effective Vulnerability Detection

CERIAS
The Center for Education and Research in Information Assurance and Security

2024 - AI - PZM-I5M

MOTIVATION

MOTIVATION

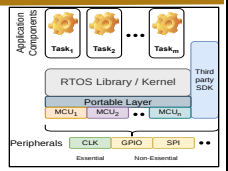


Dynamic Analysis of Embedded Systems bring in many challenges such as:

- Requires Hardware Instrumentation
- Input Injection Difficult
- The curse of extreme diversity

TYPE – II SYSTEMS & THREAT MODEL

Type – II Embedded systems utilize a RTOS, which can have a portable layer interacting with the HW peripherals.

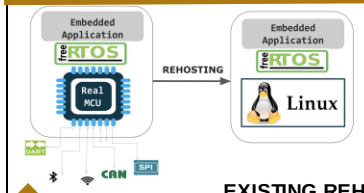


Threat Model:

- Non-essential peripherals (like GPIO) can send malicious data from the external world.
- Essential peripherals (like CLK) can send malformed data to the SW stack.

MAIN IDEA

REHOSTING AS LINUX APPLICATIONS



The main idea is to "rehost" an embedded application to be able to execute on a Linux platform.

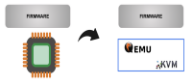
Many existing security analysis can be applied on these systems then.

EXISTING REHOSTING APPROACHES

Many existing approaches propose rehosting solutions, but they require one or more of the below:

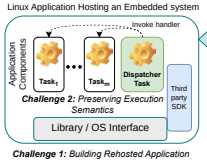
- Real hardware
- Peripheral modeling
- Base emulator support

Our approach does not require these!



CHALLENGES

REHOSTING CHALLENGES

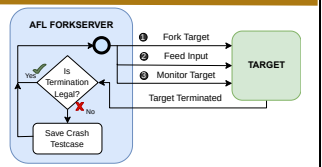


Challenges to rehosting as Linux applications are:

- Retargeting to x86-64 architecture
- Preserving Execution Semantics
- Handling Peripheral Interactions

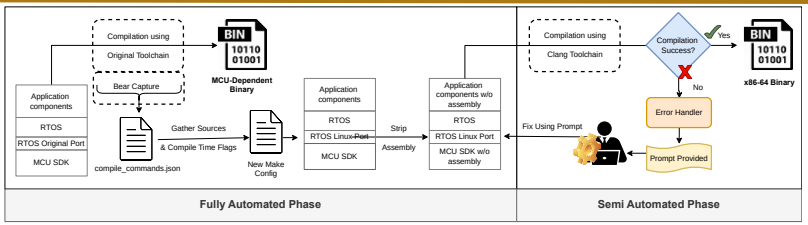
FUZZING CHALLENGES

- Fuzzing regular applications relies on program termination on every test case.
- If the termination was illegitimate, it is considered as a crash.



AUTOMATED FRAMEWORK

AUTOMATED RETARGETING



Automated phase:

- Sources and compile time flag information is gathered
- Portable layer of RTOS is changed to Linux one
- Assembly snippets are stripped.

Semi-automated phase:

- User iteratively fixes errors to get the x86-64 binary.

AUTOMATED PERIPHERAL HANDLING

MMIO in Code:

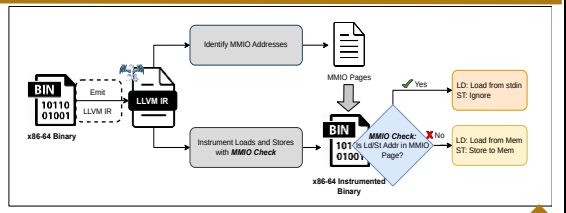
```
#define RCC ((RCC_TypeDef *)
0x40023800)
typedef struct
{
    uint32_t CR;
} RCC_TypeDef;
void register_access(){
    RCC->CR | = 0x00000001;
}
```

Static Phase:

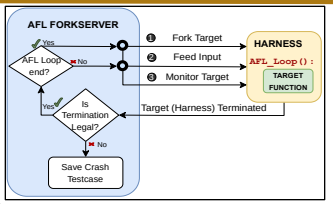
- MMIO addresses are gathered.
- Every load and store instruction is instrumented with an MMIO Check.

Dynamic Phase:

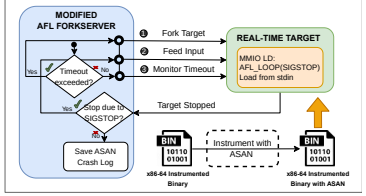
- For MMIO Loads, input is provided from standard input and MMIO Stores are ignored.



REAL-TIME FUZZING



AFL++ Persistent Mode:
Allows continuous fuzzing of a target with new inputs without having to refork the target every time.



Howwe achieve Real-time fuzzing:

- Every MMIO load stops the target using SIGSTOP.
- When the target is stopped due to this, forkserver feeds the next input and resumes target without refork.

RESULTS

PRELIMINARY RETARGETING EVALUATION

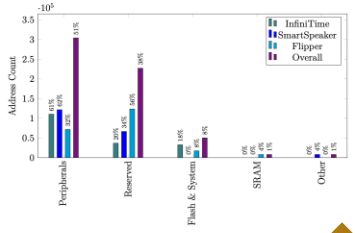
Rehosted applications compiled with ASAN were executed on Ubuntu 20.04 machine.

- The application tasks executed as expected without undefined crashes.
- Accesses to peripheral regions did not fault.
- Execution proceeded as input was provided via standard input.

PERIPHERAL HANDLING EVALUATION

Of the region identified as peripherals:

- **51%** of the addresses were correctly identified
- Considering Reserved regions, **89%** are good to be fed with standard input data



FUZZING RESULTS

```
__weak uint32_t HAL_RCC_GetSysClockFreq(void)
{
  //...
  // Reading values from hardware registers
  pllm = RCC->PLLCFGR & RCC_PLLCFGR_PLLM;
  if(__HAL_RCC_GET_PLL_OSCSOURCE()
     != RCC_PLLSOURCE_HSI)
  {
    // Usage of read value as divisor without checks
    pllvco = (uint32_t) (((uint64_t) HSE_VALUE *
    ((uint64_t) ((RCC->PLLCFGR & RCC_PLLCFGR_PLLN)
    >> RCC_PLLCFGR_PLLN_Pos))) / (uint64_t)pllm );
  } //...
}
```

Two Division By Zero Errors were found during fuzzing. Present in popularly used STM driver code, this has been fixed in their latest release.

Note: A more comprehensive fuzzing of the rehosted applications is currently underway.

PURDUE UNIVERSITY

Contact : PurS3 Lab

Read Idea paper here!

Sai Ritvik Tanksalkar, Jayashree Srinivasan, Srihari Danduri,
Paschal C. Amusuo, James C. Davis, Aravind Machiry

CERIAS