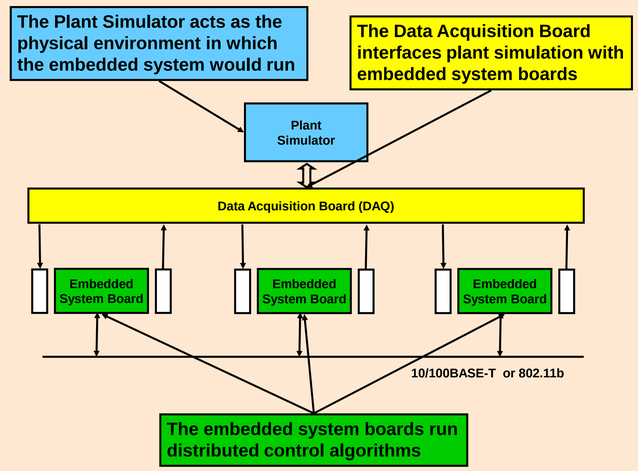
Experimental Platform for Model-Based Design of Embedded Systems

Matt Eby, Jan Werner, Janos Mathe, Gabor Karsai, Sandeep Neema, Janos Sztipanovits, Yuan Xue
Institute for Software Integrated Systems, Vanderbilt University

Experimental Platform Architecture



Specifications

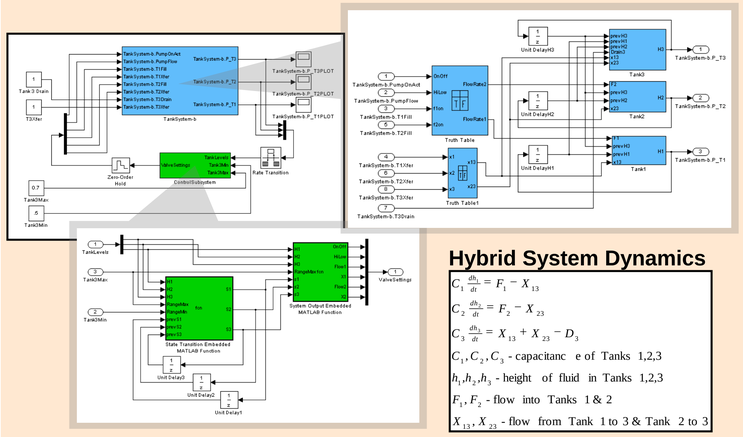
Plant Simulator
- Standard Desktop PC running Mathworks xPC
- DAQ blocks are appended to Plant Models
- xPC Code Generated with Real-Time Workshop

Data Acquisition Board
- Measurement Computing PCI-DDA08/12
- 8 analog output channels (12 bit resolution)
- 48 Digital I/O

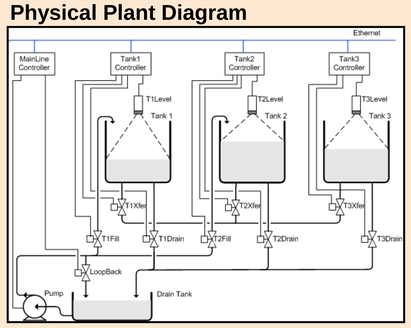
Embedded System Board
- Micro/Sys SBC4495
- Cyrix Intel 486 compatible processor
- 8 Analog Inputs & Outputs (14 bit resolution)
- 24 Digital I/O
- 10/100BASE-T Ethernet, 802.11b
- Supported OS
  - Linux, Windows CE/98, VxWorks, LynxOS, PharLap ETS, MSDOS 5.0

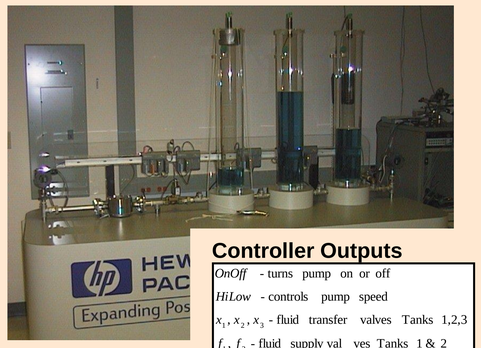
Three Tank System

- System is a test bed for the Modeling and Analysis of Complex Systems (MACS) group at Vanderbilt University
- The three tank system was chosen as an archetypical component controlled by SCADA system
- Three tank systems are common in chemical processing systems
- Tanks 1 & 2 regulate fluid levels in Tank 3 while Tank 3 supplies fluid to some process downstream
- We use this system to demonstrate and test the capabilities of security measures introduced via Model-Based Design



Physical Plant Diagram





Hybrid System Dynamics

$C_1 \frac{dh_1}{dt} = F_1 - X_{13}$

$C_2 \frac{dh_2}{dt} = F_2 - X_{23}$

$C_3 \frac{dh_3}{dt} = X_{13} + X_{23} - D_3$

C_1, C_2, C_3 - capacitanc e of Tanks 1,2,3

h_1, h_2, h_3 - height of fluid in Tanks 1,2,3

F_1, F_2 - flow into Tanks 1 & 2

X_{13}, X_{23} - flow from Tank 1 to 3 & Tank 2 to 3

Controller Outputs

OnOff - turns pump on or off

HiLow - controls pump speed

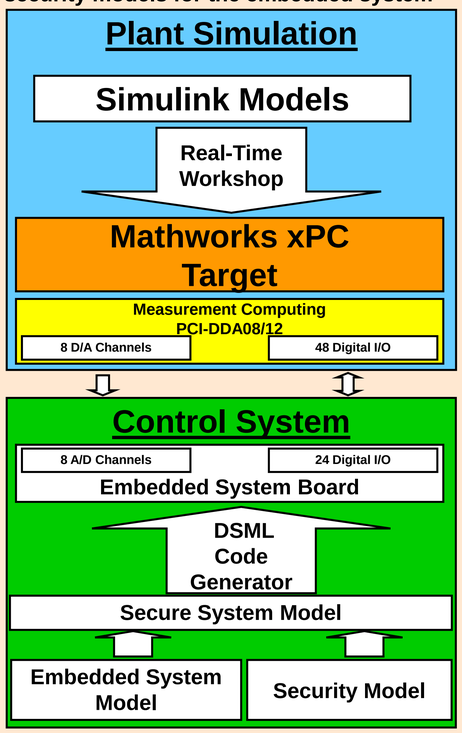
x_1, x_2, x_3 - fluid transfer valves Tanks 1,2,3

f_1, f_2 - fluid supply val ves Tanks 1 & 2

d_3 - Tank 3 drain

Testing Model-Based Security Features

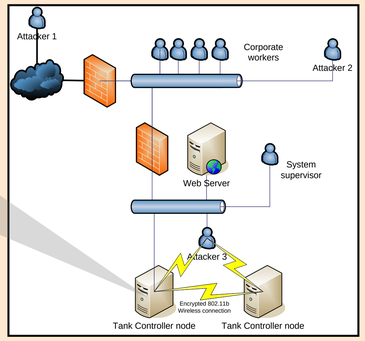
- The experimental platform facilitates "Hardware"-in-the-Loop testing of controllers.
- High fidelity plant simulations behave just as the actual physical environment would.
- Controllers can run on various operating systems with different security designs.
- Code for controllers is generated based on security models for the embedded system



- The experimental platform is configured for specific control problems such as a Three Tank System controlled by a SCADA system.
- We then test a variety of attacks against the system
- This allows us to exercise the code produced from the security models for:
  - Performance overhead
  - Strength of security for specific attacks
  - Comparison between different operating systems

Configuration of Experimental Platform for Three Tank Testing

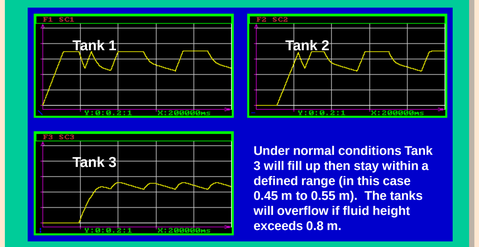
Application Code	Application Code
GRsecurity Extensions	
Gentoo Linux (kernel 2.4.32)	
Device Drivers	
Embedded System Board	



- For the tests conducted on a Three Tank Controller we are running Gentoo Linux (kernel 2.4.32) with GRsecurity extensions.
- GRsecurity adds 3.9% (33 kB) to the kernel footprint
- Performance overhead is 3.5% for non-executable memory protection
- GRsecurity extensions allow fine grained control over system resources
  - I/O registers
  - Memory Protection
  - Inter-process Communication

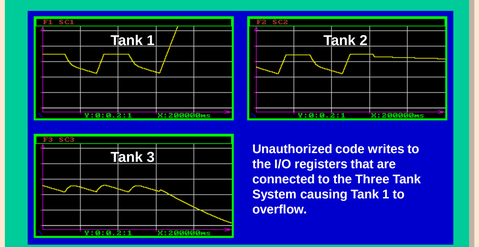
Attacks on Three Tank System

Normal Operation



Under normal conditions Tank 3 will fill up then stay within a defined range (in this case 0.45 m to 0.55 m). The tanks will overflow if fluid height exceeds 0.8 m.

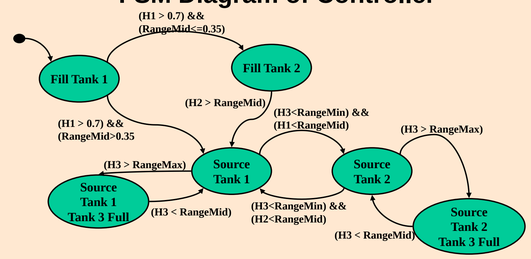
Unauthorized Access to I/O registers



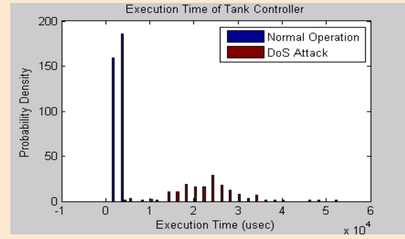
Unauthorized code writes to the I/O registers that are connected to the Three Tank System causing Tank 1 to overflow.

- With I/O register protection only the tank control process has permission to write to I/O channels
- Model-Based approach can map desired security properties to underlying platform services such as POSIX capabilities (e.g. CAP_SYS_RAWIO)

FSM Diagram of Controller



- Denial of Service attack can increase execution time of tank control process
  - Operation under normal conditions
    - Worst case execution time = 12712 µs
    - Mean execution time = 3123 µs
  - Denial of Service attack on network data access component
    - Worst case execution time = 52600 µs
    - Mean execution time = 23200 µs



- DoS attacks cannot be easily prevented without support of platform services such as packet filtering.