
NATIONAL
INSTRUMENTS™

Addressing Embedded Security in LabVIEW RIO Systems

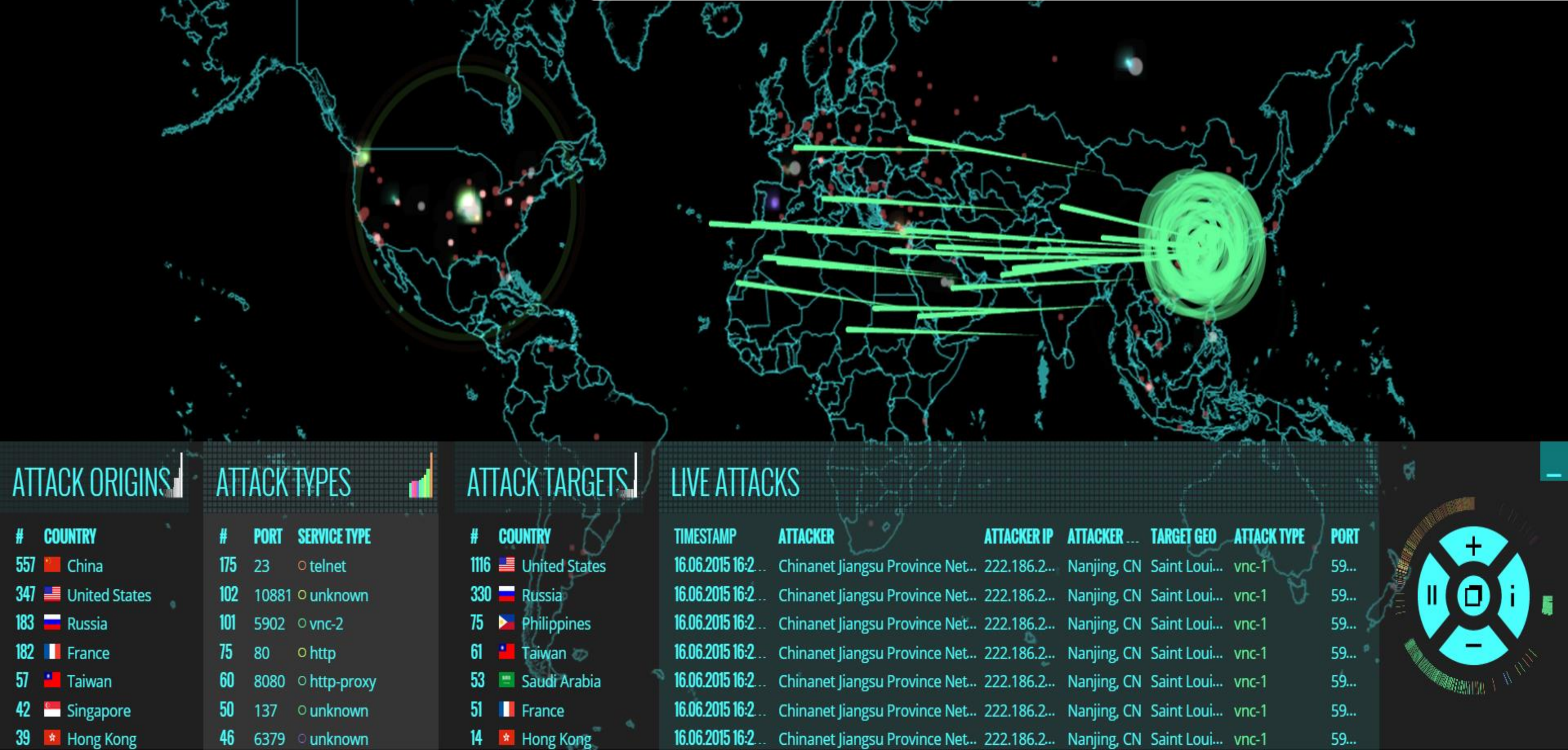
Carlos Pazos

Product Marketing Manager

Embedded Software

NATIONAL INSTRUMENTS

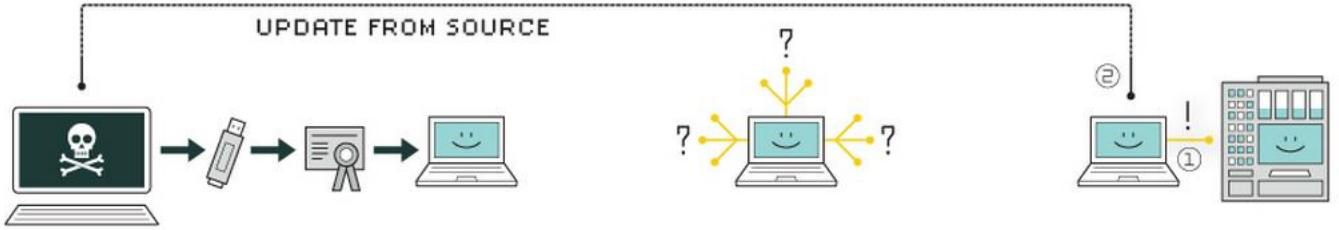
Why Care About Security?



Why Care About Industrial Security?

IEEE Spectrum, Feb 26 2013

Stuxnet Explained



1. infection
Stuxnet enters a system via a USB stick and proceeds to infect all machines running Microsoft Windows. By brandishing a digital certificate that seems to show that it comes from a reliable company, the worm is able to evade automated-detection systems.

2. search
Stuxnet then checks whether a given machine is part of the targeted industrial control system made by Siemens. Such systems are deployed in Iran to run high-speed centrifuges that help to enrich nuclear fuel.

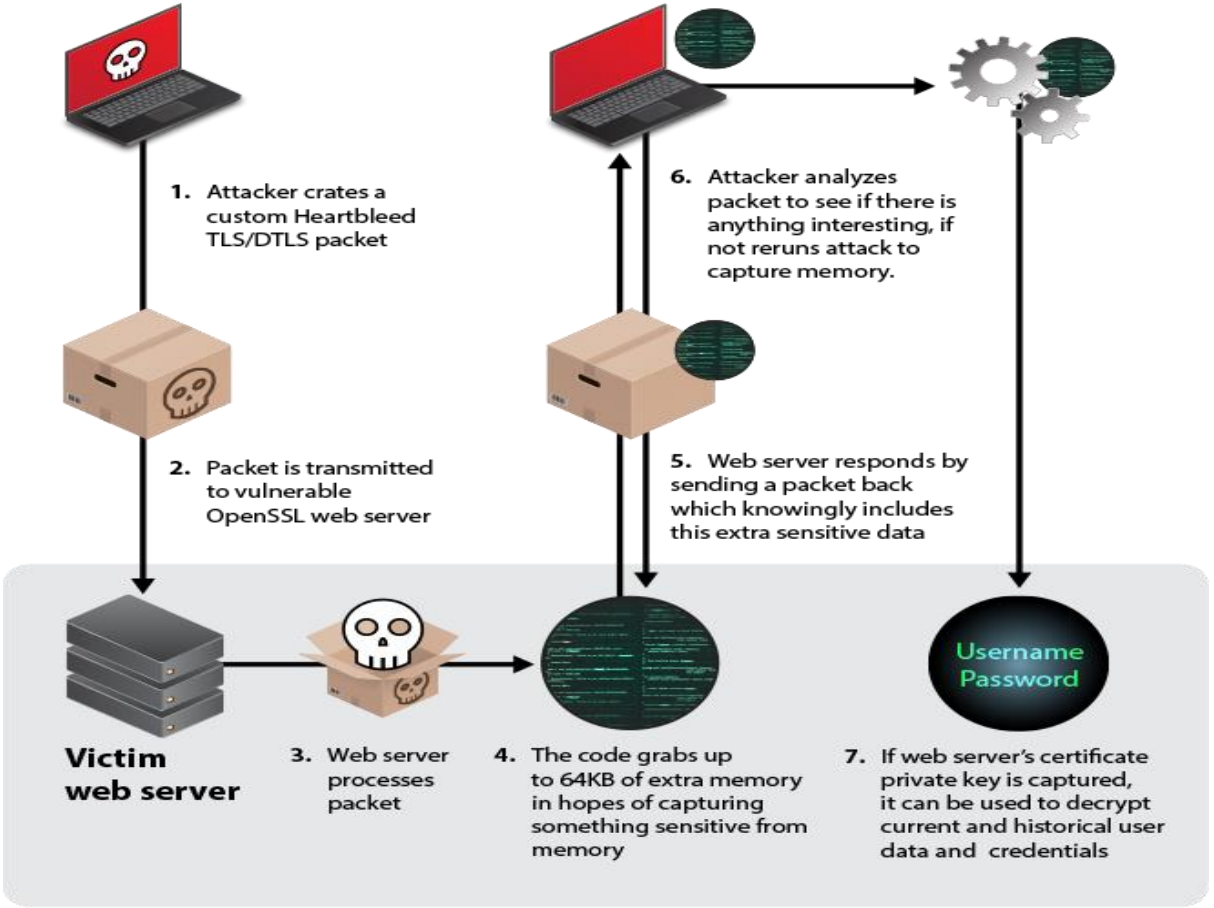
3. update
If the system isn't a target, Stuxnet does nothing; if it is, the worm attempts to access the Internet and download a more recent version of itself.

4. compromise
The worm then compromises the target system's logic controllers, exploiting "zero day" vulnerabilities-software weaknesses that haven't been identified by security experts.

5. control
In the beginning, Stuxnet spies on the operations of the targeted system. Then it uses the information it has gathered to take control of the centrifuges, making them spin themselves to failure.

6. deceive and destroy
Meanwhile, it provides false feedback to outside controllers, ensuring that they won't know what's going wrong until it's too late to do anything about it.

OpenSSL Heartbleed Vulnerability



Security is a Shared Responsibility

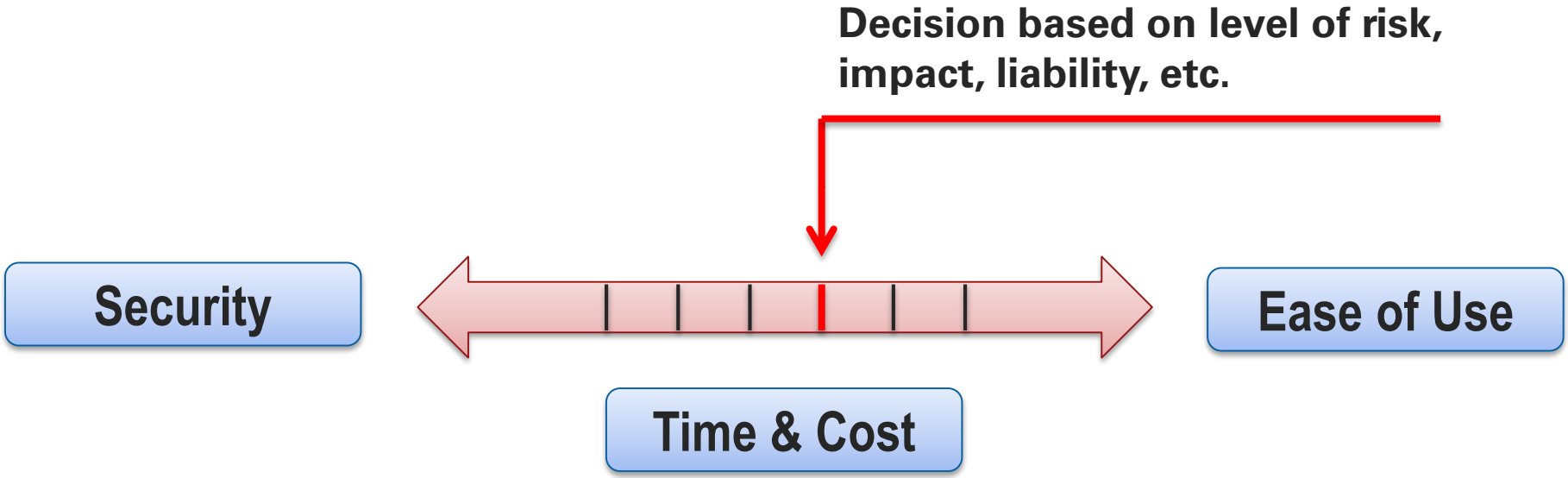
- Customers: Define security objectives, assess application-specific risks, evaluate and implement security steps

- NI: Provide best practices, provide security features that enable deployment in your threat environment, listen to customer's feedback for the future

We want to work with our customers to tackle this complex challenge.

How much time should be invested in security?

Balancing Security with Other Constraints



The Challenge

Development is hard:

Build a system that functions as designed

Security is <u>really</u> hard:

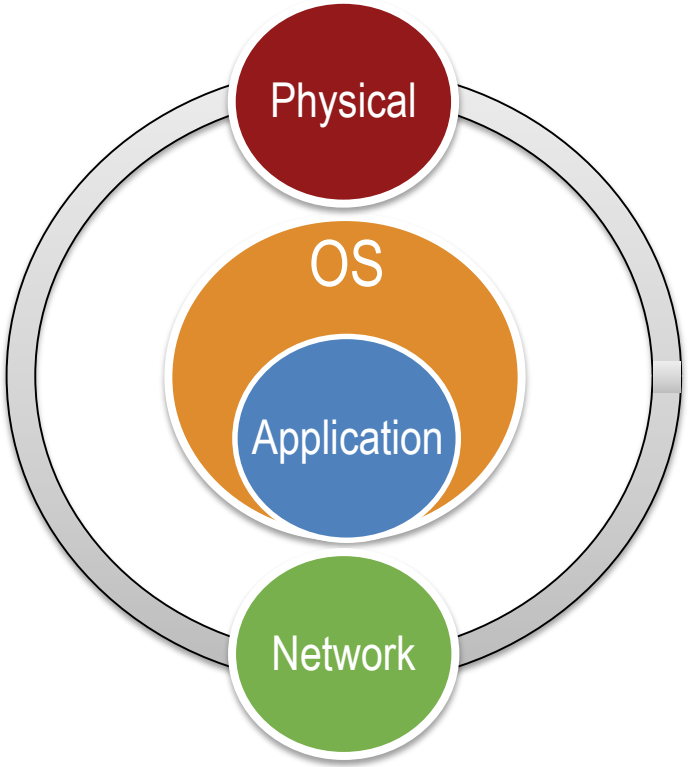
Build a system that *can't be used any other way*

Examples: Surveillance, theft, impersonation, base of operations

"A system's security is a function of its weakest link."

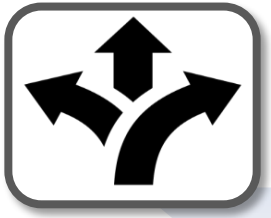
Layered Model of Security

- Security can be defined at many layers
- A breach at any layer can compromise other layers
- “Don’t invest in a retinal scanner for your house if you are going to leave your window open”
- “Defense in Depth”



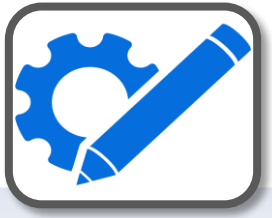
CompactRIO Development Stages

Platform Selection



System Configuration

Application Development



Application Deployment

Application Maintenance

NATIONAL INSTRUMENTS

CompactRIO Development Stages

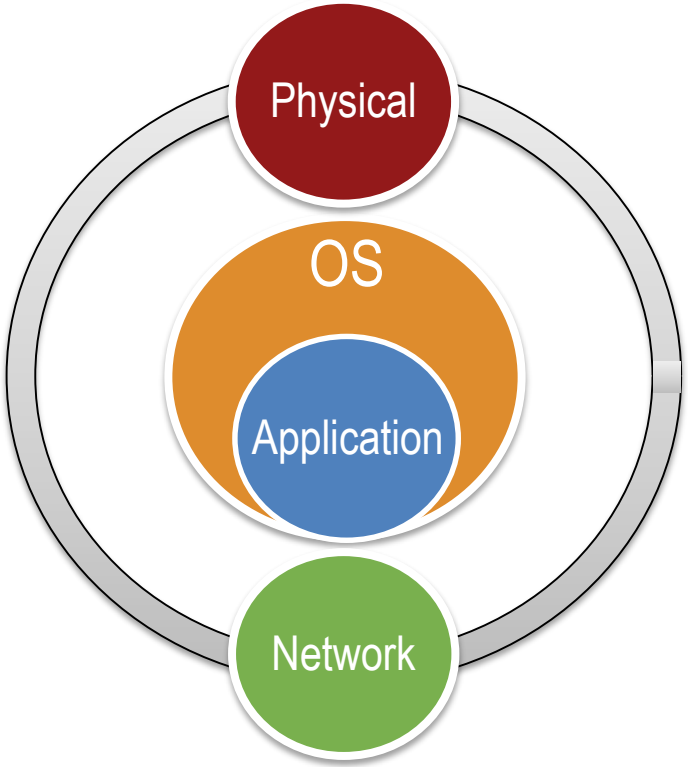
Ethernet

Development and/or Deployment PC

CompactRIO Embedded System

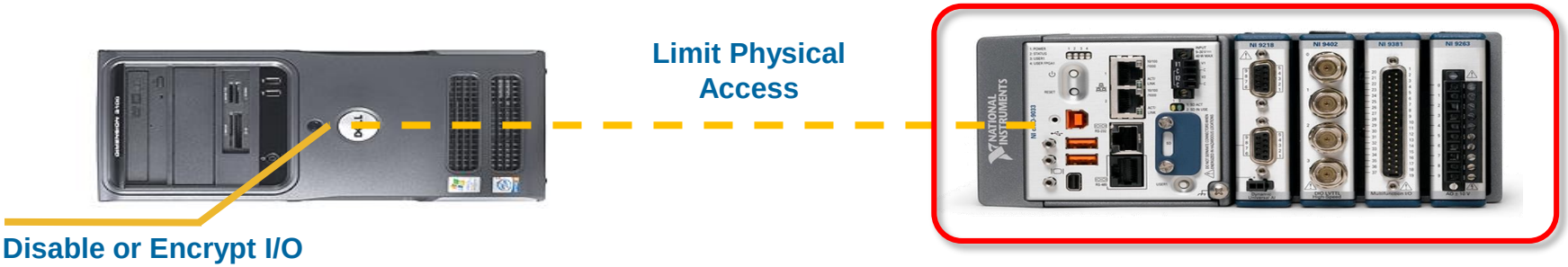
Agenda: Security Best Practices

1. Physical Security
2. Network Security
3. Application & OS Security
   - NI Linux Real-Time



Physical Security

Physical Security



- Limit environmental/physical access to Host PC and physically enclose the real-time target
- Disable or require encryption on I/O (USB, CD Drive, etc.)
- Implement hardware checking
  - USB dongle for IP Protection
  - Digital I/O to validate that enclosure is properly secured

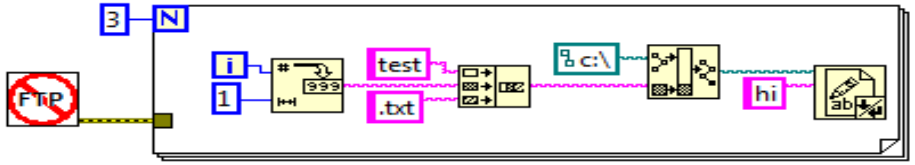
Network Security

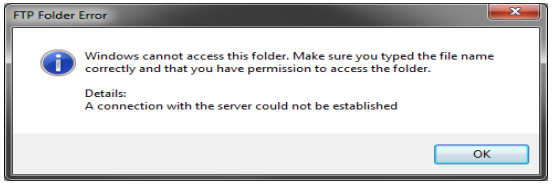
General Network Security

- **Secure network**
  - Isolated network if possible; firewall if you connect to public infrastructure
- **Use web services for network communication**
  - HTTPS provides integrity and confidentiality
  - SSL certificate-based authentication between devices
  - Password-based user authentication between services
- **Understand the security risk of network protocols.**
  No authentication nor confidentiality for:
  - Remote Front Panel, Remote VI Server, Network shared variables
  - Mitigation:
    - **Good:** Use IP address access control, blacklist Exported VIs
    - **Better:** Use lower-layer tunnel (e.g. VPN or SSL)

Disabling the FTP Server

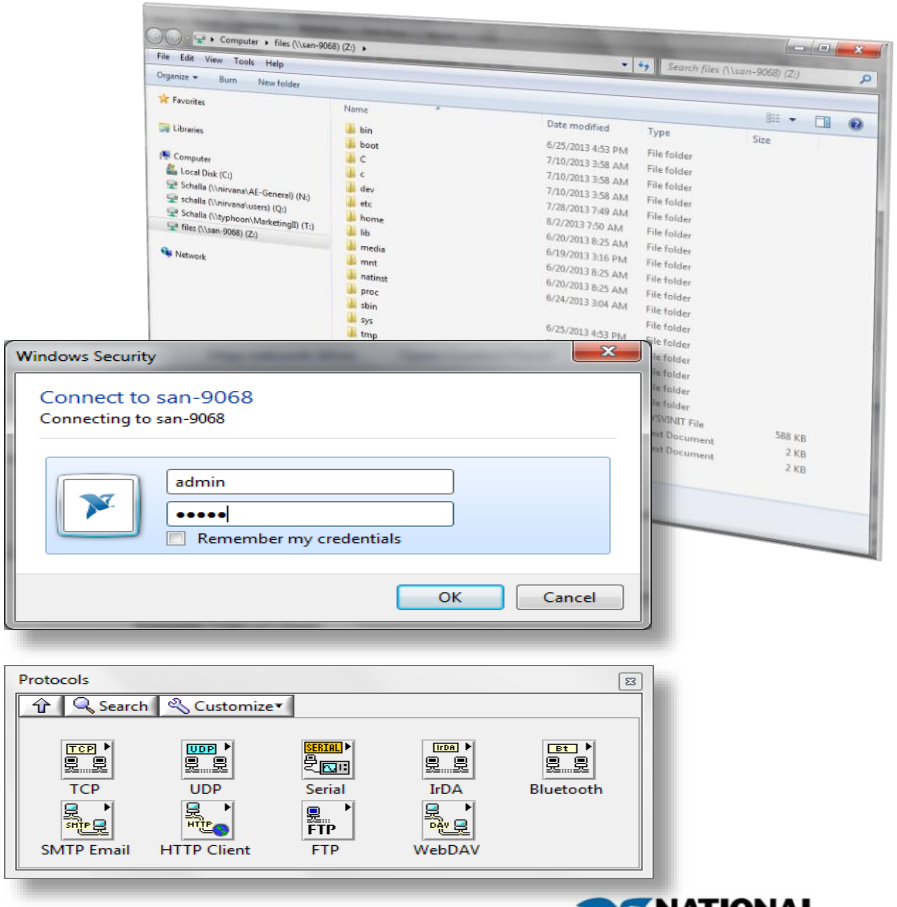
- Only on **VxWorks** & **Phar Lap**
  - There is no unsecured FTP by default on NI Linux RT
- Use Shutdown RT FTP Server VI in your RT application
- Must reboot to enable FTP server
  - Disable RT Startup App to enable FTP





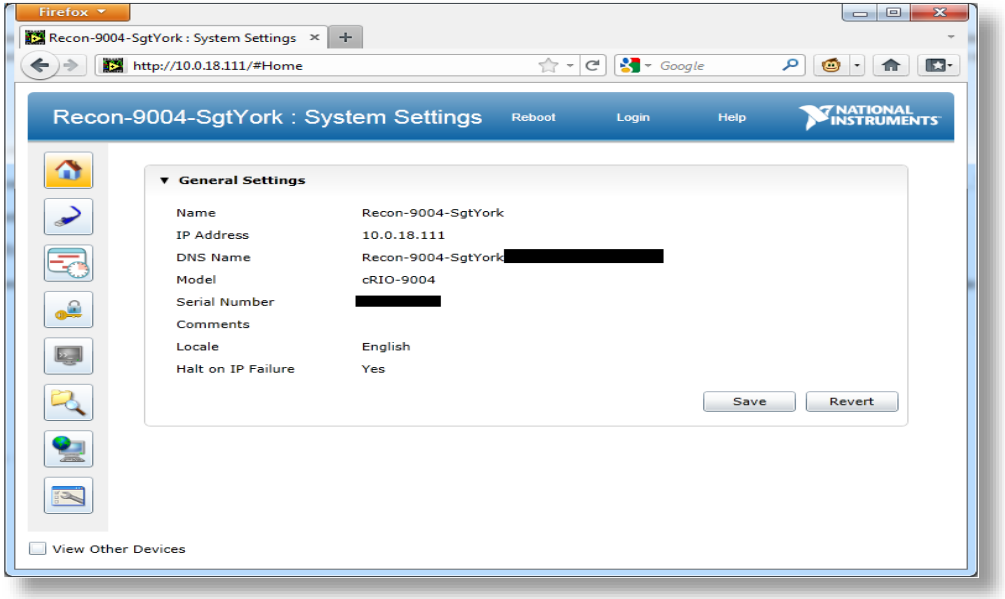
File Transfer: WebDAV

- Industry Standard Protocol
- Manage files on targets remotely over HTTP(S)
- Secure File Access
  - Authentication & Encryption
- Supported by all modern OSes and Web Browsers
- LabVIEW API for programmatic access
- WebDAV File Browser

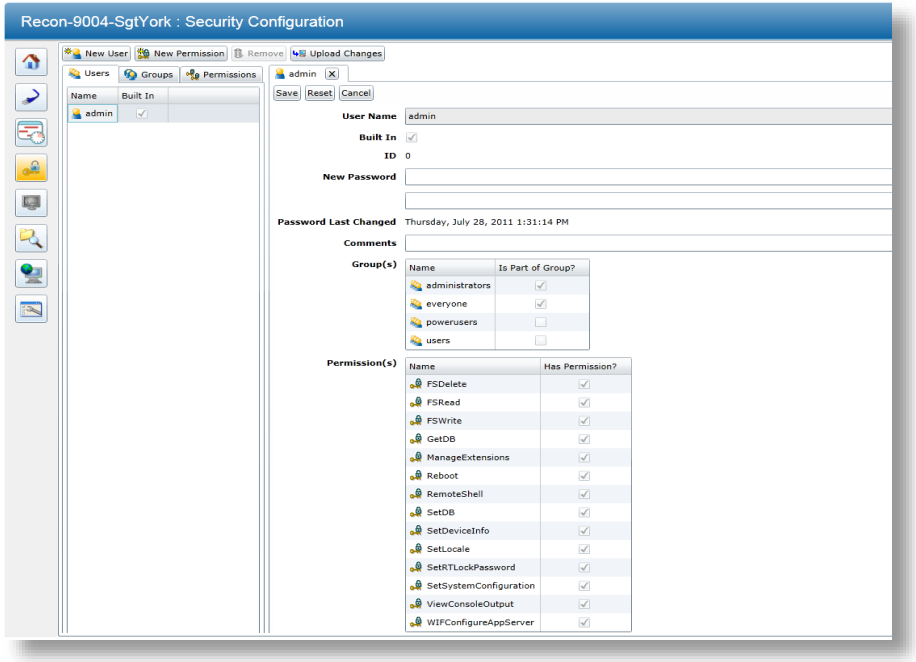


Web Based Configuration and Monitoring

- Use to access NI Auth settings
- cRIO:
  - http://<cRIO_ip.addr>
- Host PC
  - http://<host_ip.addr>:3582

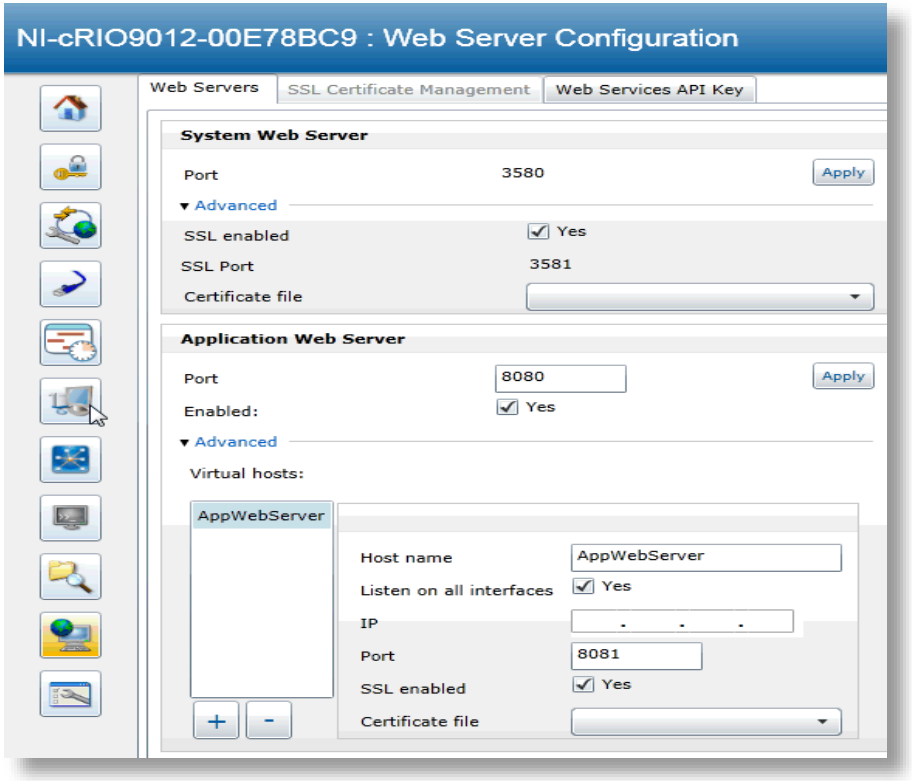


NI-Auth Users & Permissions



- **Change password on ‘admin’ account**
- Set permissions on users and groups to restrict users & groups to only activities they are responsible for
- Currently applicable only to web services and web management

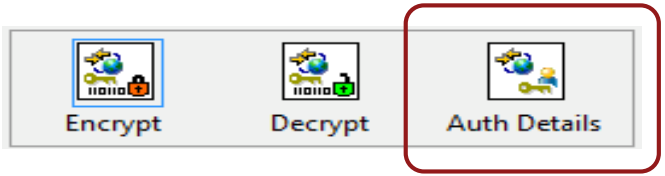
Enable SSL



- Enable SSL for both System and Application Web Servers
- Turn off HTTP version and rely only on the HTTPS version
- Select and setup self signed certificates or go through a CA
- System Web Server available at
  - https://<cRIO_ip.addr>
  - https://<host_ip.addr>:3581

Web Services Security palette

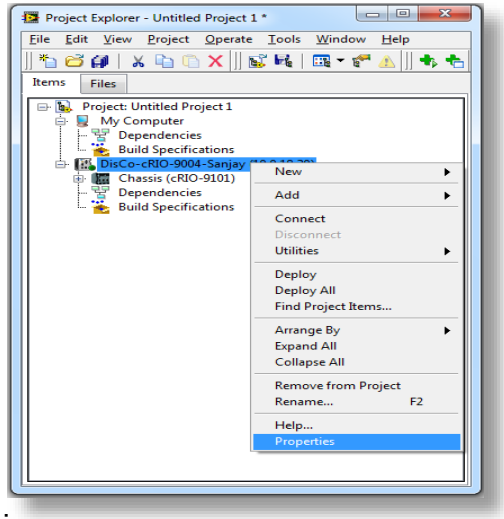
- Allows web service to retrieve NI-Auth attributes
  - User name
  - Group
  - Permissions
  - Session key (320 bits)

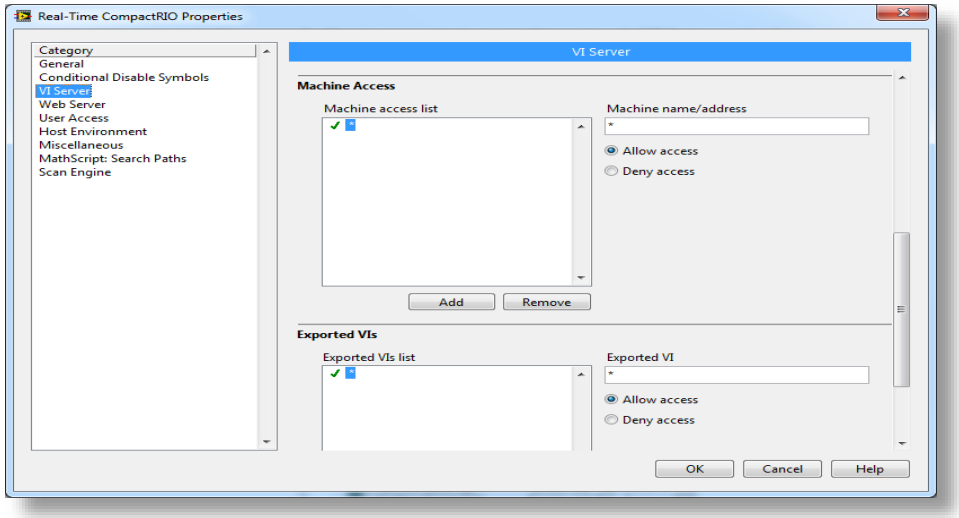


- Encryption functions that use the session key
  - Secure Remote Password (RFC 2945) using SRP_SHA1
  - Message confidentiality
  - Message authentication/integrity using third-party HMAC

Managing VI Server Access

- Manage VI Server (TCP) Access to prevent ‘LabVIEW’ viruses
  - Prevent remote access & execution of code on your PC or RT target
- For Host PC: Tools » Options » VI Server
- For cRIO: use Project Explorer as shown here

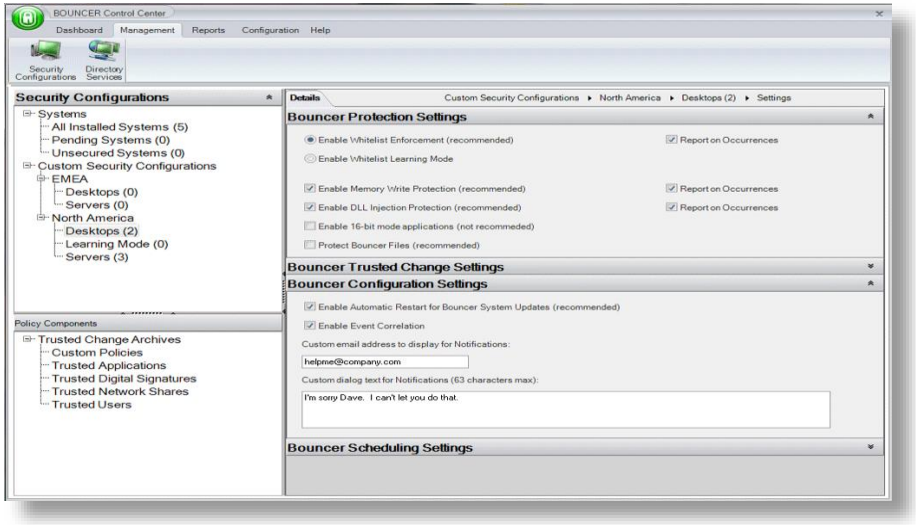




Application & OS Security

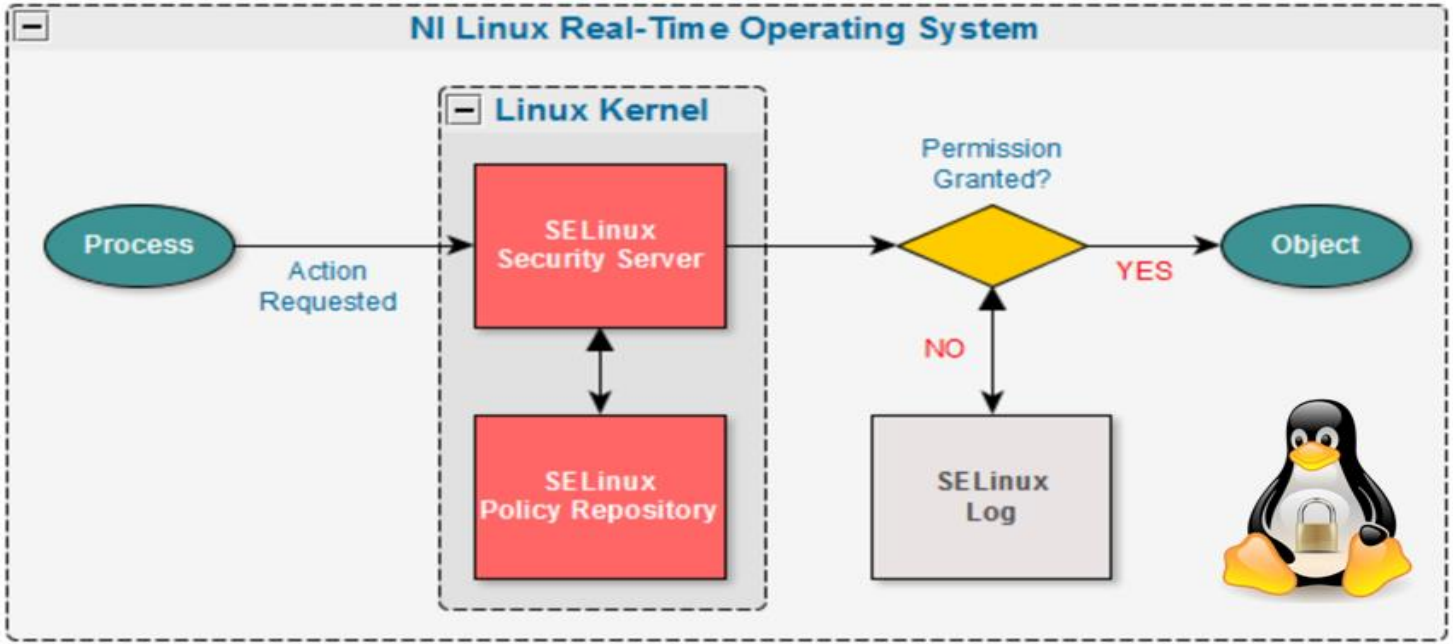
Application Whitelisting

- Use to regulate code execution on host computer
- Compute and store checksum in a known good state
- Compare checksum against table
- Solutions:
  - CoreTrace Bouncer
  - McAfee Application Control
  - Bit9 Parity Suite
  - Microsoft AppLocker

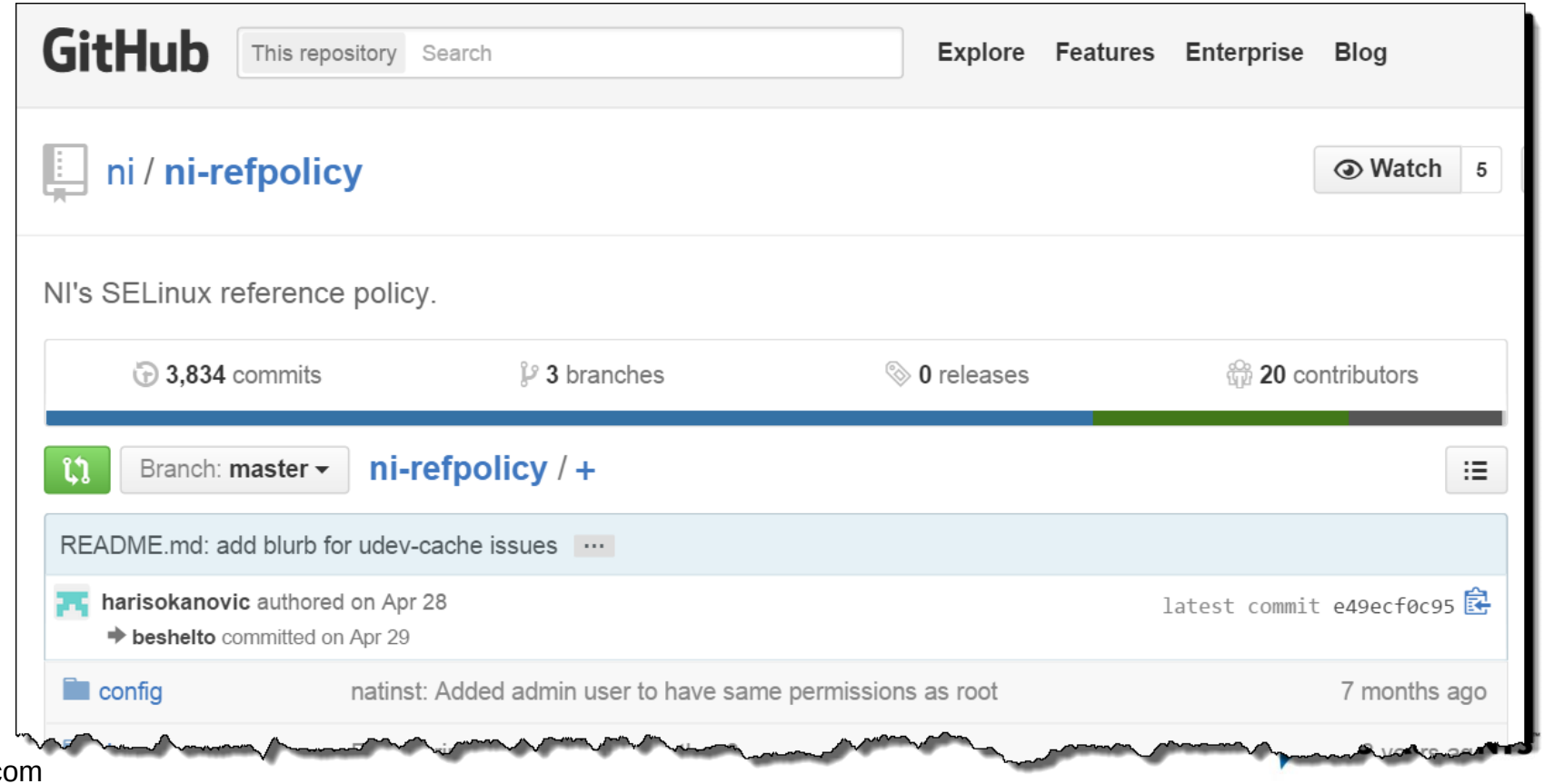


Malicious Code Prevention

Security Enhanced Linux (SELinux)



GitHub Reference Policy

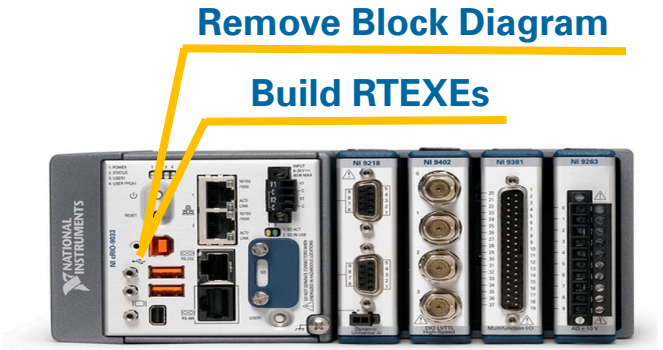


Sensitive Data Protection

- Options for encrypting data in storage
  1. Several individuals offer LabVIEW VIs
     - Caution: Much slower than C/C++ implementations
  2. Use .NET library in LabVIEW
     - system.security.cryptography library
     - Add reference to mscorlib.dll
  3. Use DLL calls to the host OpenSSL/CAPI library
- General principle: *" Ensure the confidentiality of sensitive data through its entire lifecycle."*

LabVIEW and LabVIEW Real-Time

Application Security



- Remove the block diagram: LabVIEW Help: Removing Block Diagrams from VIs
- Use Build Specifications such as EXEs & RTEXEs
- Remove the source code and the development environment from host computers on the deployed network

LabVIEW FPGA
Application Security

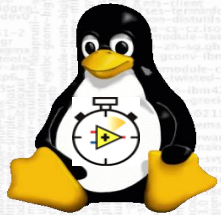
- Implement bounds checking on FPGA I/O to prevent damage
- Use an FPGA Watchdog over RT and default to FPGA ‘safe states’ if something is erroneous on RT

NI Linux Real-Time

NI Linux Real-Time

- Owned and maintained by NI
  - Custom built and optimized for NI embedded hardware
    - Supports ARM and x64, with cross-compilers provided
  - Easier integration of third-party peripherals and applications as most have Linux drivers or packages available
  - NI Package Repository: download.ni.com/ni-linux-rt/
    - Over 3,000 packages
  - OS source: github.com/ni
- PREEMPT_RT
  - Enables real-time reliability through pre-emption, priority inheritance, and scheduling
  - Standard approach to real-time performance on Linux

LINUX FOUNDATION
SILVER MEMBER



NATIONAL INSTRUMENTS

Linux Ecosystem

Database

Raima
MySQL
SQLite
MongoDB
CouchDB

Security

SELinux
OpenVPN
IP Tables
System Logging
fail2ban
denyhost

Code
Re-use

C/C++
Shell scripting
Python
Ruby
Perl

Connectivity

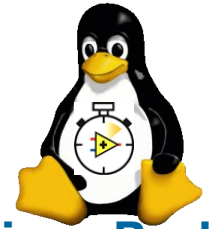
Isshd
IPv6
SNMP
NTP
netstat

Integration with the Latest Hardware Products

CompactDAQ Controller

Zynq Single-Board RIO

Controller for FlexRIO



NI Linux Real-Time

System on Module

Compact Vision System

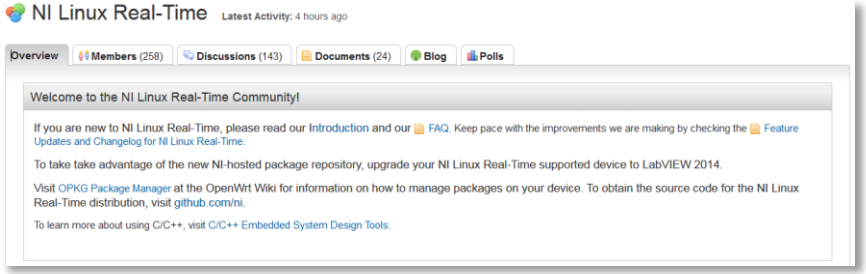
Quad-core Performance CompactRIO

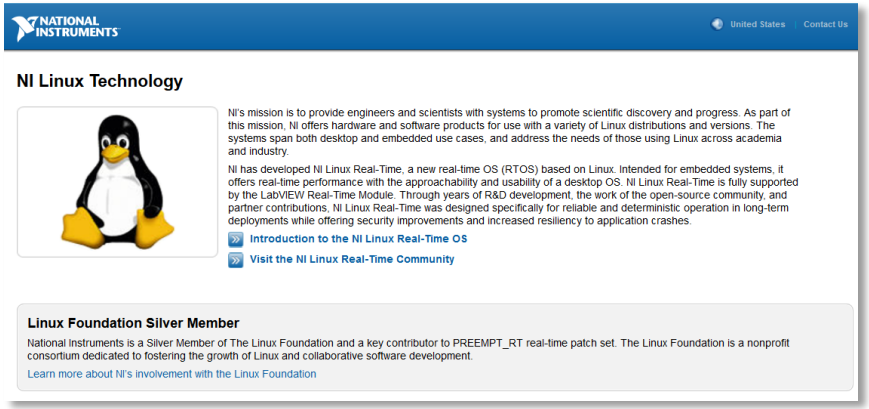
myRIO

NATIONAL INSTRUMENTS

Key Resources

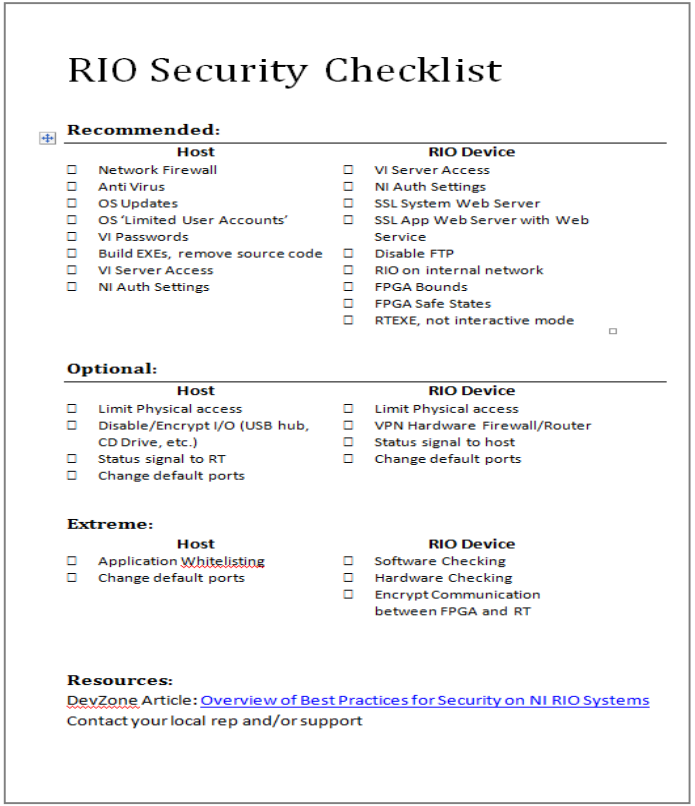
- ni.com/linuxrtforum
  - Tutorials
  - Documentation
  - Forum for discussions
- ni.com/linux
  - Links to whitepapers
  - Embedded and Desktop uses
- download.ni.com/ni-linux-rt/
  - Package Repository
- github.com/ni
  - OS Source





Next Steps

- Sign up at ni.com/security
- Leverage online resources: Overview of Best Practices for Security on NI RIO Systems
- Engage NI with security needs such as security certifications (NERC CIP, IEC 62351, etc.)
- Use the Best Practices Checklist to evaluate and secure your system



RIO Security Checklist

Recommended:

Host	RIO Device
☐ Network Firewall	☐ VI Server Access
☐ Anti Virus	☐ NI Auth Settings
☐ OS Updates	☐ SSL System Web Server
☐ OS 'Limited User Accounts'	☐ SSL App Web Server with Web Service
☐ VI Passwords	☐ Disable FTP
☐ Build EXEs, remove source code	☐ RIO on internal network
☐ VI Server Access	☐ FPGA Bounds
☐ NI Auth Settings	☐ FPGA Safe States
	☐ RTEXE, not interactive mode

Optional:

Host	RIO Device
☐ Limit Physical access	☐ Limit Physical access
☐ Disable/Encrypt I/O (USB hub, CD Drive, etc.)	☐ VPN Hardware Firewall/Router
☐ Status signal to RT	☐ Status signal to host
☐ Change default ports	☐ Change default ports

Extreme:

Host	RIO Device
☐ Application Whitelisting	☐ Software Checking
☐ Change default ports	☐ Hardware Checking
	☐ Encrypt Communication between FPGA and RT

Resources:

DevZone Article: Overview of Best Practices for Security on NI RIO Systems

Contact your local rep and/or support

Questions?

Carlos Pazos
Product Marketing Manager
Embedded Software
carlos.pazos@ni.com

NATIONAL INSTRUMENTS