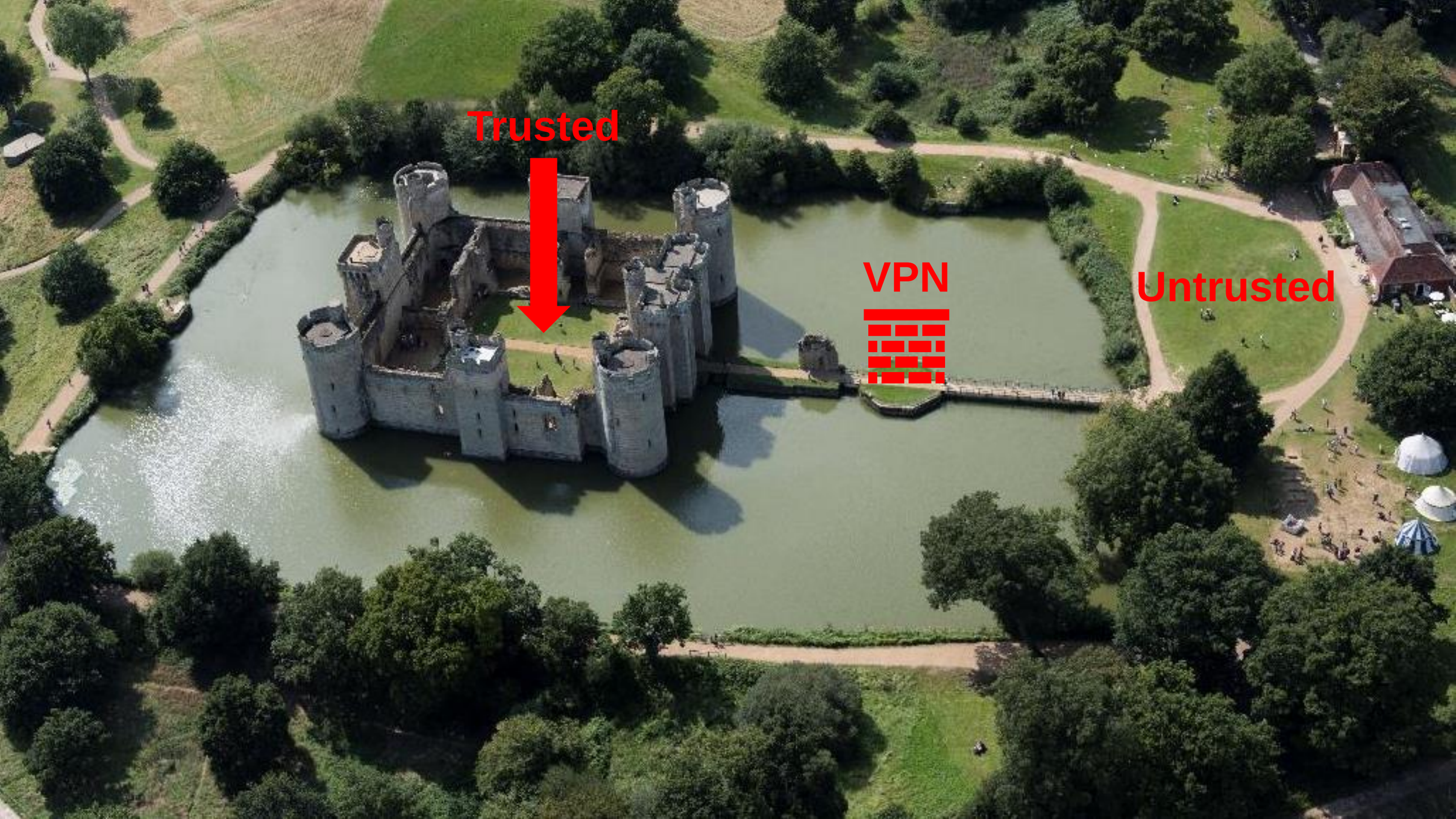
Deploying Real-Time Systems into Secure Environments

Deploying Real-Time Systems into Secure Environments
Wednesday, May 22 | 1:30 PM - 2:30 PM

Main Track: Aerospace & Defense
Cross Track: Test Development & Management

Description:
Deploying a real-time system (PXI or CompactRIO) into a secure environment requires controls to protect the network and data. Explore how to implement these controls and satisfy your security team.

CONNECT

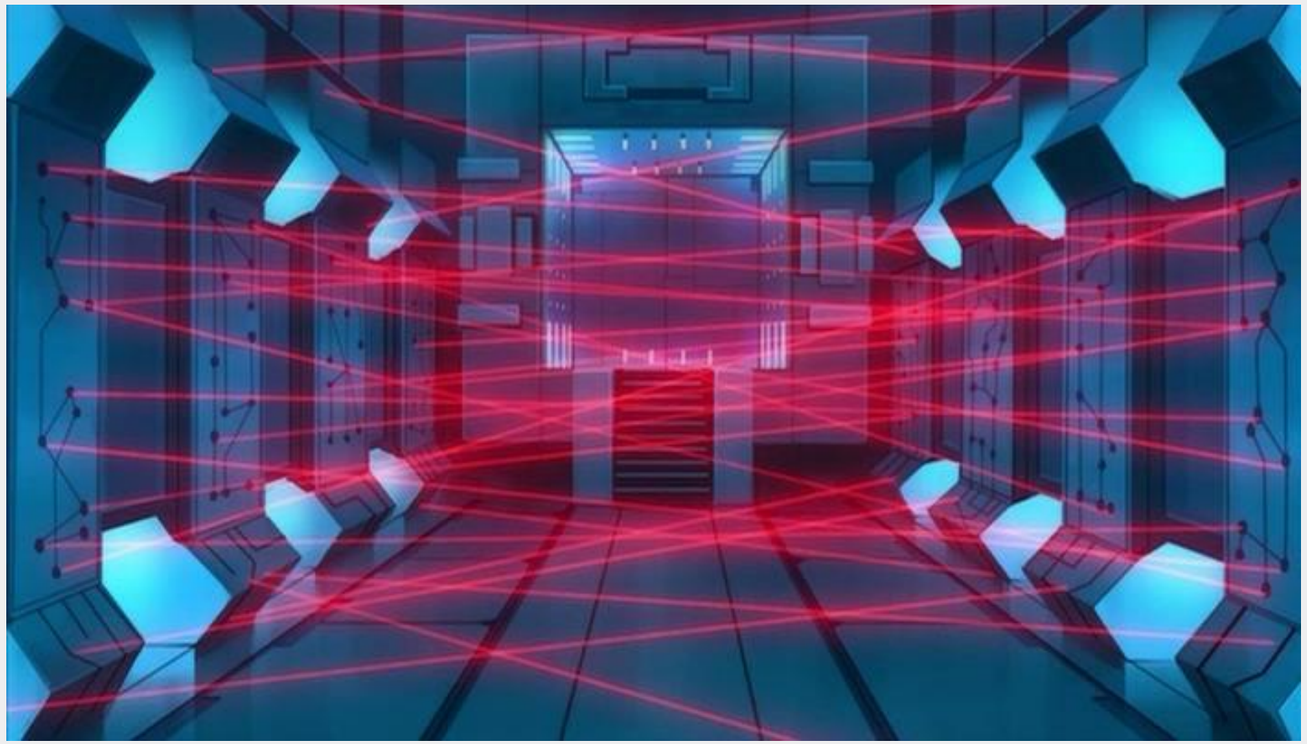

Trusted
VPN
Untrusted

Walls can be breached.

Shift to zero-trust

In a zero-trust world:

- Assume a breach can happen
- Trust no one
- Each component must provide its own security
- Air-gaps are no longer valid



System Security Approval

If you test products you may need:

- Company security approval
- Customer security approval
- Government security approval

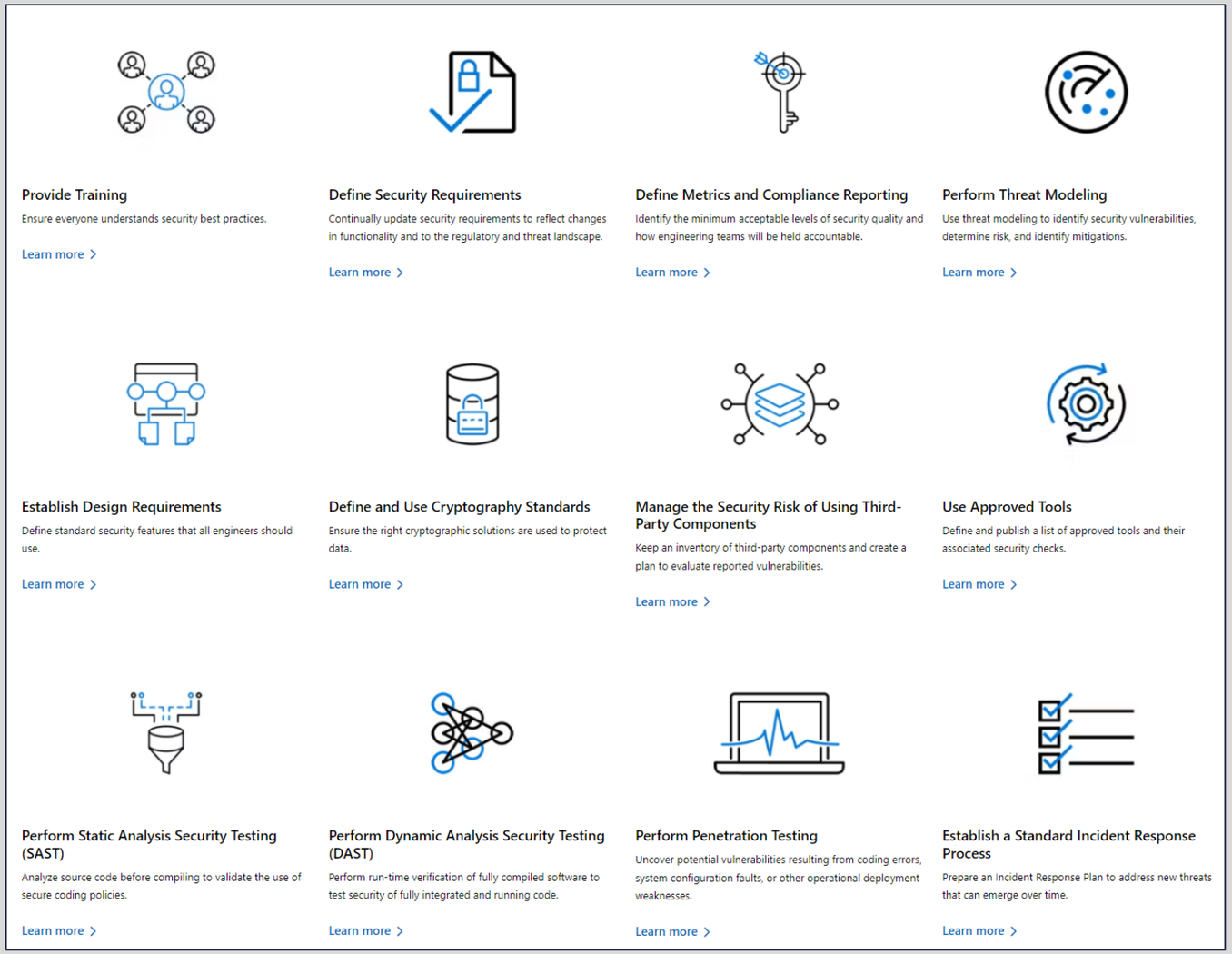
If you deliver a system you may need:

- Customer security approval
- Government security review (RMF)
- Contract flow-down requirements
- Security documentation

Security Documentation

- **Does the system meet security requirements?**
  - NIST 800-171
  - User system secure development process
- **Do the components meet security requirements?**
  - NIST 800-171
  - Self-attestation form
  - Vendor Software secure development process
  - FEDRamp
- **Do you know how to configure the system securely?**
  - Secure configuration guide
  - Letters of Volatility
- **Is the system configured securely?**
  - STIG

System Security Integrator Responsibility



Adopt a security development lifecycle

- DevOps -> DevSecOps
- Microsoft Secure Development Lifecycle
- https://www.microsoft.com/en-us/securityengineering/sdl/practices

Understand your requirements

- NIST 800-171

Develop good LabVIEW code

- NI development guidelines
- VI Analyzer

Test your code

- Dynamic analysis
- Static analysis
- Document Security

Compliance documents

- STIGs
- LOVs
- Security Training

Software for Professional Test Workflows

Electronics Validation Test

Characterizing electronic prototypes to ensure quality and performance

Set-up & Configure → Measure & Automate → Analyze & Share

Electronics Production Test

Functional test ensuring manufactured products meet specifications

Set-up & Configure → Measure & Automate → Deploy & Maintain

Electromechanical Validation Test

Characterizing physical prototypes to ensure quality and performance

Build & Customize → Configure → Analyze & Share

Embedded Software Test

Testing deployed software for defects across wide parameter variations

Configure & Map → Test & Bring Up → Automate & Execute

Security is a significant part of each of these workflows

EMERSON | ni

Validation Systems

Electronics Test | Areas of Investment

Accessible Automation

Streamline toolchain for validation use case to speed use and lower barrier to entry

Example features
- Automate interactive measurements
- Connect sequencing inside InstrumentStudio

Open Architectures

Compatibility with 3rd party hardware and software tools

Example features
- Panels and pin maps for 3rd party instruments
- Ease use of .NET, Python, MATLAB, C#

Modern Dev Practices

Improve collaborative tools in LabVIEW+ to ease large, complex application development

Example features
- git integration for LabVIEW and TestStand
- Improve diff and merge to support CI/CD

System Security

Meet regulatory requirements for security and share details of exposures

Example features
- SBOMs and CVEs
- IPv6 support

Measurement Transition

Software tool interoperability to quickly share specifications, measurements, data and results

Example features
- Measurement repository
- Generate compliance data from InstrumentStudio

Production Systems

Electronics Test | Areas of Investment

Modernize UI Building

Update UI controls to provide engaging experience for custom interfaces

Example features
- Multilanguage character support
- Web controls

Open Architectures

Compatibility with 3rd party hardware and software tools

Example features
- Step Types for 3rd party instruments
- Ease use of .NET, Python, MATLAB, C#

Modern Dev Practices

Improve collaborative tools in LabVIEW+ to ease large, complex application development

Example features
- git integration for LabVIEW and TestStand
- Improve diff and merge of LabVIEW code to support CI/CD

System Security

Meet regulatory requirements for security, especially when maintaining long term system deployments

Example features
- SBOMs and CVEs
- Linux deployments

Measurement Transition

Software tool interoperability to quickly share specifications, measurements, data and results

Example features
- Use measurements from a library
- Integration with SystemLink

Validation Systems

Electromechanical Test | Areas of Investment

Accessible Automation

Speed system development with connected applications and easy-to-use sequencing.

Example features
- Connect FlexLogger measurements with automation in LabVIEW or TestStand
- Automate durability tests without programming in FlexLogger

Open Integration

Simplify integration of 3rd party hardware, custom algorithms, and control logic.

Example features
- Improve development and debugging of FlexLogger plugins
- Develop custom measurements in any language

Modern Dev Practices

Improve collaborative tools in LabVIEW+ to ease large, complex application development.

Example features
- git integration for LabVIEW and TestStand
- Improve diff and merge to support CI/CD

System Security

Meet regulatory requirements for security, especially when maintaining long term system deployments.

Example features
- SBOMs and CVEs
- Linux RT Identity and Access Management

Out-of-the-box Measurements

Hardware and software built together to deliver measurements in minutes.

Example features
- FlexLogger Lite included with every DAQ device
- Guided setup, reference material and pin layout accessible directly from the hardware

Validation Systems

Embedded Software Test | Areas of Investment

Bus Configuration

Communicate to your DUT using required communication protocols

Example features
- VCOM Custom device
- Communication bus template
- Custom device scripting APIs

Simulink and Model Integration

Integrate models to provide simulated data to your DUT

Example features
- Expanded Simulink™ HDL Coder support for NI Hardware
- FMI 3.0 Support

Debugging

Improve tools to quickly identify & resolve errors when building your HIL system

Example features
- Improved custom device debugging
- Error handling, diagnostics, and debugging with in VeriStand

System Security

Meet regulatory requirements for security, especially when maintaining long term system deployments

Example features
- SBOMs and CVEs
- Linux deployments

Accessible Automation

Increase test throughput by automating the HIL test system

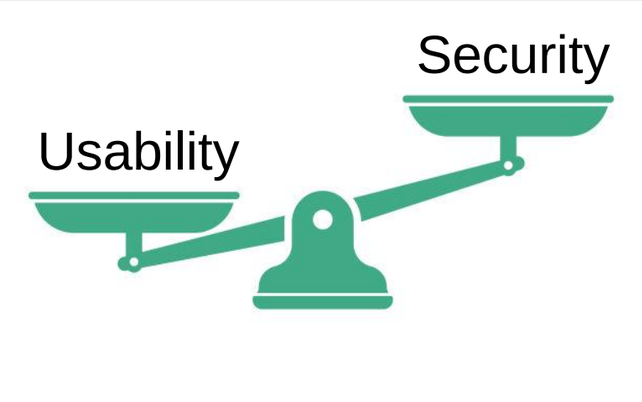
Example features
- VeriStand steps for TestStand
- In-product sequencing

Security in an NI Linux RT System

Balancing ease of use with security

Usability:

- Web access
- VI Server for remote access
- Simplified deployment of LabVIEW code
- Real-time access to running VIs
- Removeable media



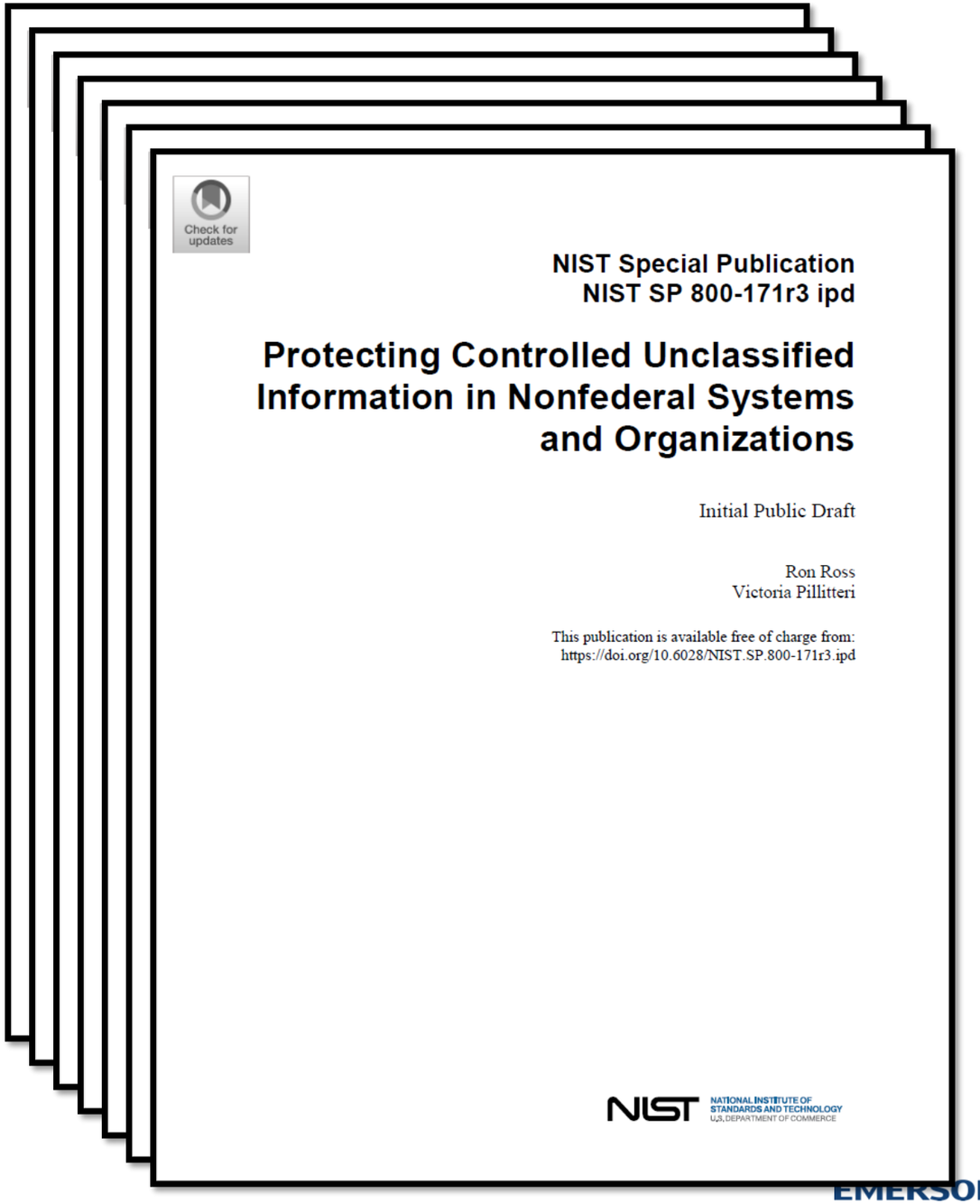
Security:

- Encrypted transfer
- Encrypted storage
- Account management
- Data segmentation

What is “Security?”

NIST SP 800-171 Rev 3

- Complete
- Well-documented
- Compatible with NIST 800-53
- Accepted by most US Government Agencies
- Control document for CMMC



Check for updates

NIST Special Publication
NIST SP 800-171r3 ipd

Protecting Controlled Unclassified Information in Nonfederal Systems and Organizations

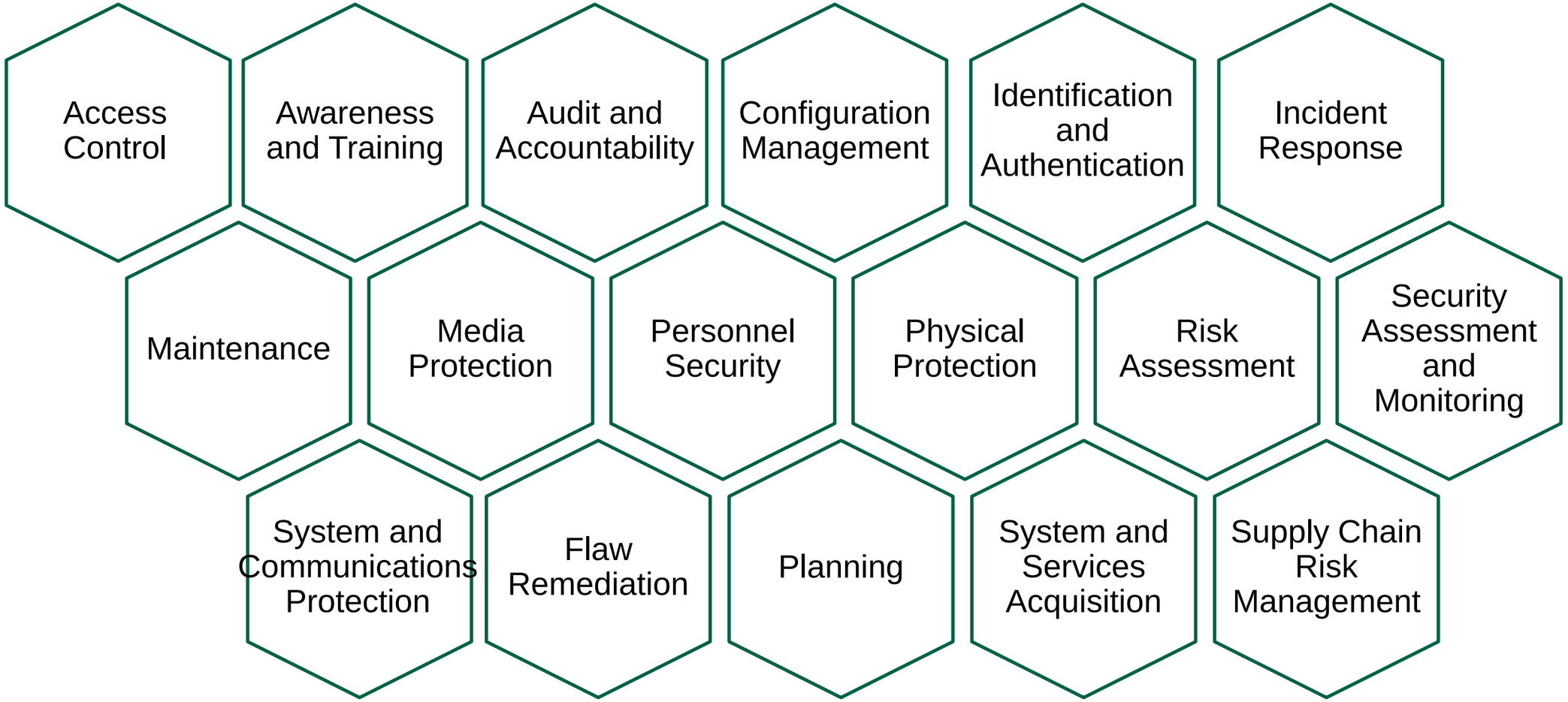
Initial Public Draft

Ron Ross
Victoria Pillitteri

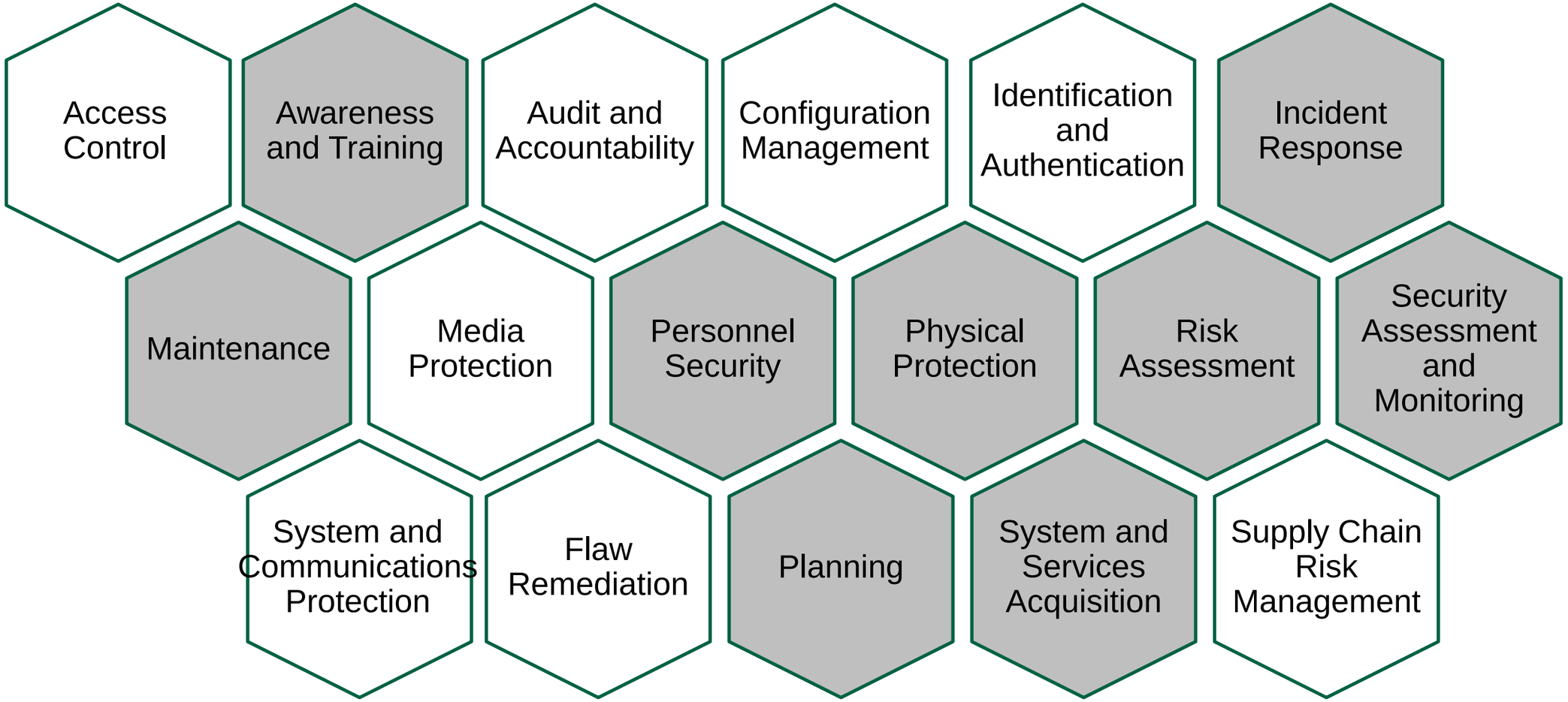
This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-171r3.ipd

NIST NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY U.S. DEPARTMENT OF COMMERCE

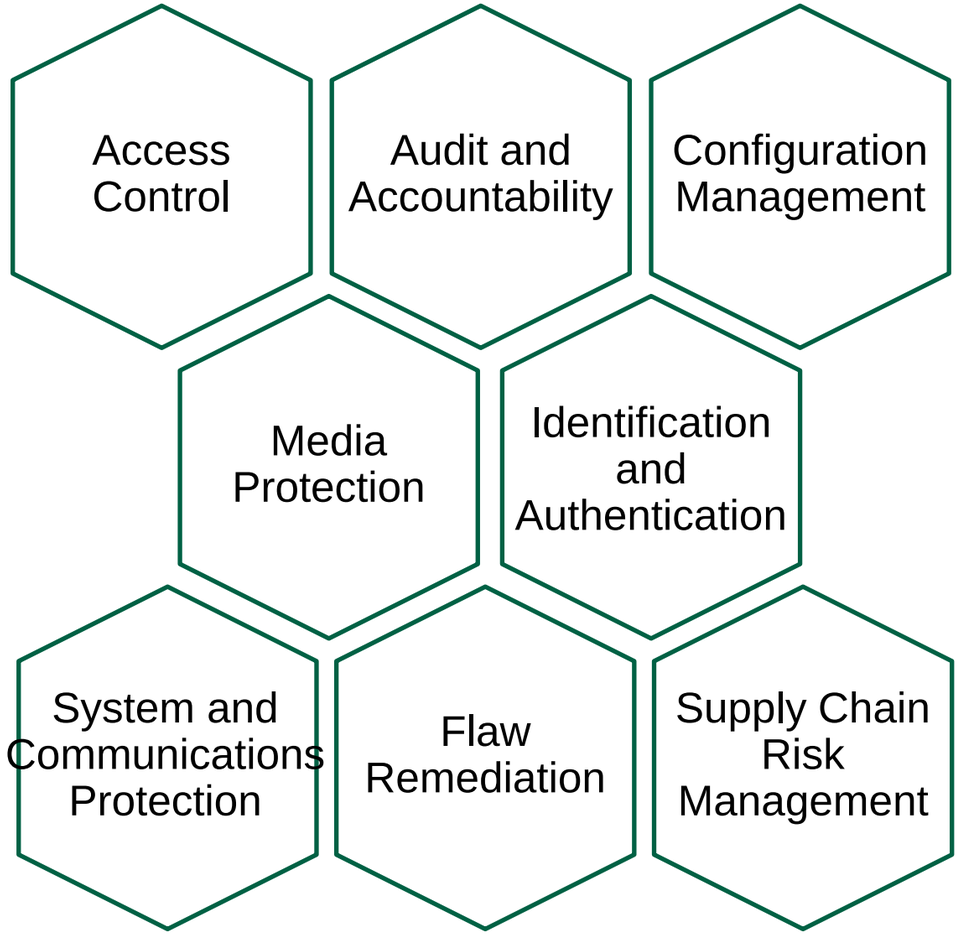
NI 800-171 Control Families



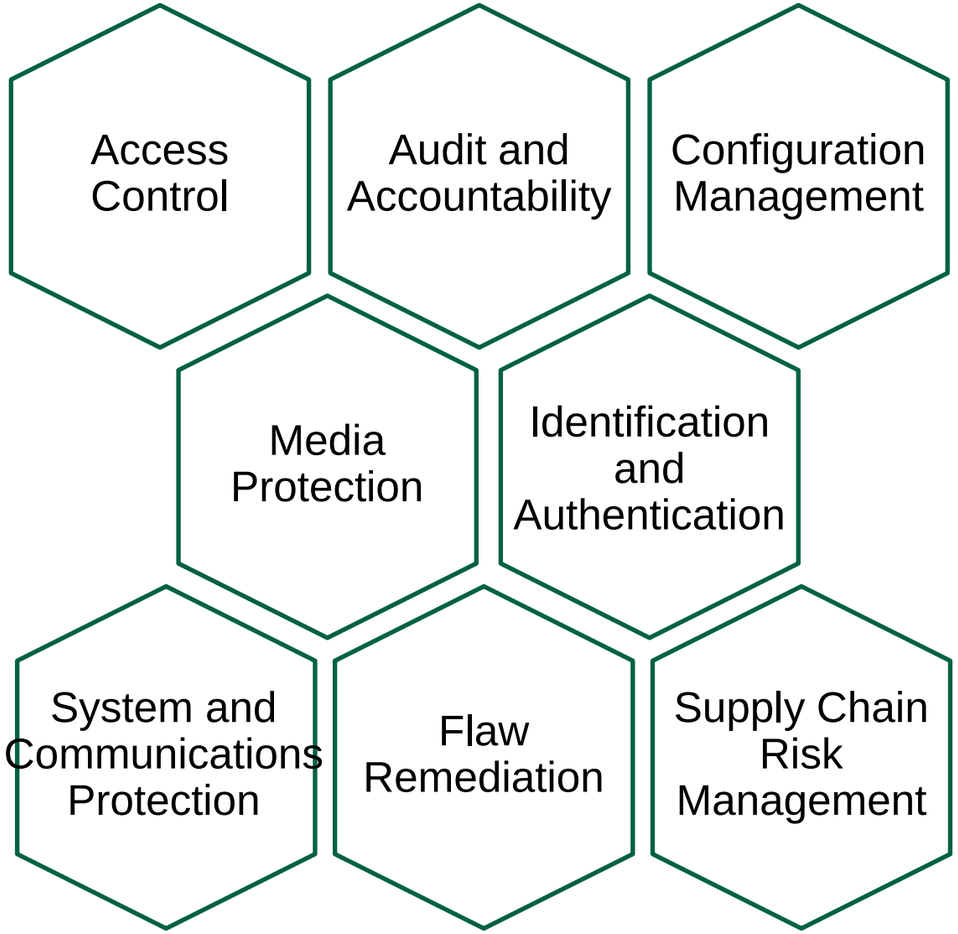
NI 800-171 Control Families – NI RT Devices



NI 800-171 Control Families – NI RT Devices



NI 800-171 Control Families – NI RT Devices



Know who is using the system

Control what users do on the system, by role

Record user actions on the system

Protect activity logs

Control configuration of the system

Control connections to external systems and devices

Scan for and fix vulnerabilities

Protect data in storage

Protect data in transit

Maintain Software Bill of Materials

Manage suppliers for security

Dispose of systems properly

Letters of Volatility

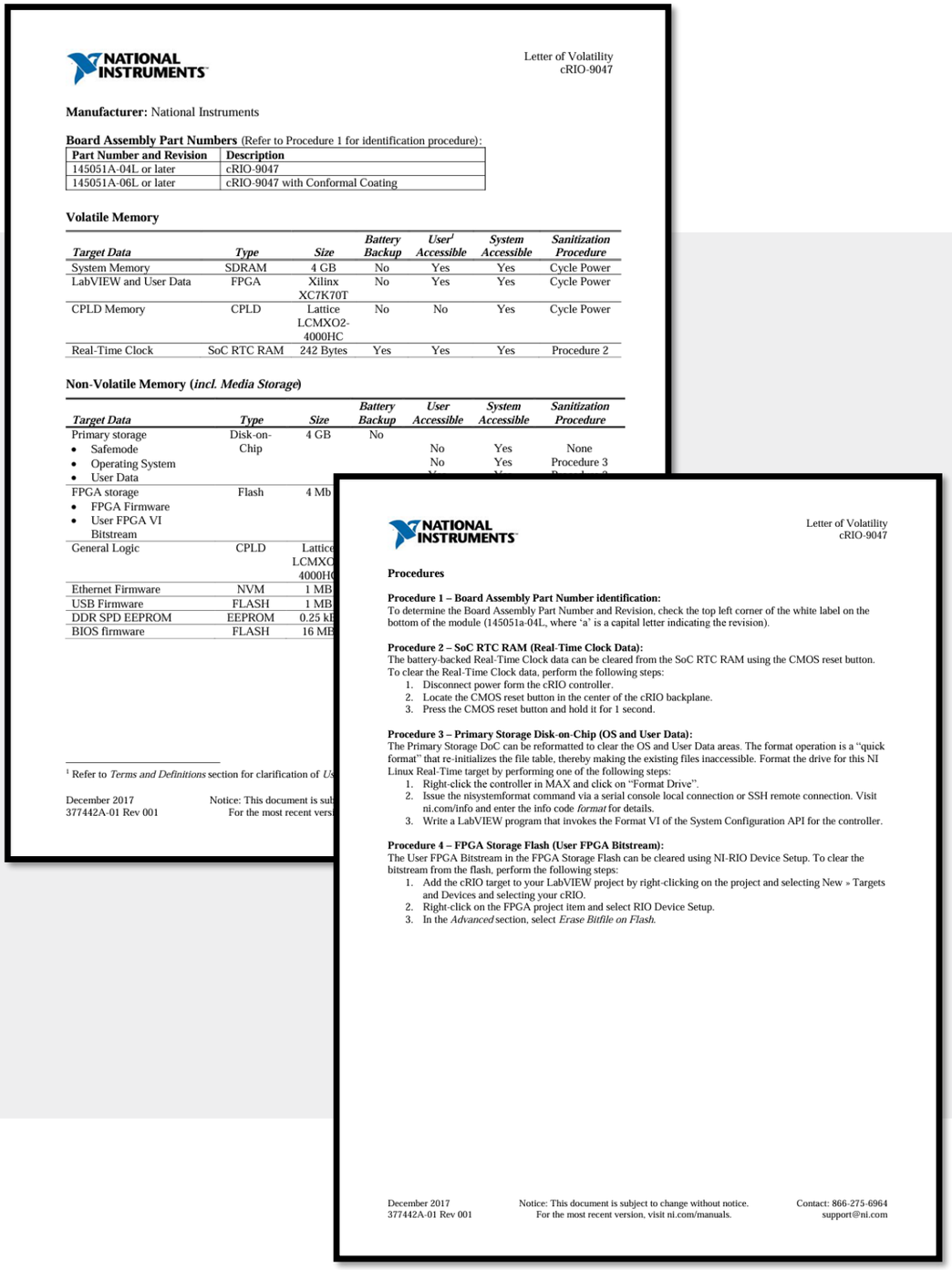
Supply Chain Risk Management

LOVs available for all NI hardware

- ni.com/letters-of-volatility
- Found in user manuals
- List of volatile, non-volatile memory locations
- Clearing instructions for non-volatile memory

Anti-Counterfeit Measures

- NI has a robust supply chain management process
- Documentation available on request



NATIONAL INSTRUMENTS

Letter of Volatility
cRIO-9047

Manufacturer: National Instruments

Board Assembly Part Numbers (Refer to Procedure 1 for identification procedure):

Part Number and Revision	Description
145051A-04L or later	cRIO-9047
145051A-06L or later	cRIO-9047 with Conformal Coating

Volatile Memory

Target Data	*Type*	*Size*	*Battery Backup*	*User[1] Accessible*	*System Accessible*	*Sanitization Procedure*
System Memory	SDRAM	4 GB	No	Yes	Yes	Cycle Power
LabVIEW and User Data	FPGA	Xilinx XC7K70T	No	Yes	Yes	Cycle Power
CPLD Memory	CPLD	Lattice LCMXO2-4000HC	No	No	Yes	Cycle Power
Real-Time Clock	SoC RTC RAM	242 Bytes	Yes	Yes	Yes	Procedure 2

Non-Volatile Memory (*incl. Media Storage*)

Target Data	*Type*	*Size*	*Battery Backup*	*User Accessible*	*System Accessible*	*Sanitization Procedure*
Primary storage	Disk-on-Chip	4 GB	No			
• Safemode				No	Yes	None
• Operating System				No	Yes	Procedure 3
• User Data						
FPGA storage	Flash	4 Mb				
• FPGA Firmware						
• User FPGA VI Bitstream						
General Logic	CPLD	Lattice LCMXO 4000H				
Ethernet Firmware	NVM	1 MB				
USB Firmware	FLASH	1 MB				
DDR SPD EEPROM	EEPROM	0.25 kB				
BIOS firmware	FLASH	16 MB				

[1] Refer to *Terms and Definitions* section for clarification of *Us*

December 2017
377442A-01 Rev 001

Notice: This document is sub
For the most recent vers

NATIONAL INSTRUMENTS

Letter of Volatility
cRIO-9047

Procedures

Procedure 1 – Board Assembly Part Number identification:
To determine the Board Assembly Part Number and Revision, check the top left corner of the white label on the bottom of the module (145051a-04L, where 'a' is a capital letter indicating the revision).

Procedure 2 – SoC RTC RAM (Real-Time Clock Data):
The battery-backed Real-Time Clock data can be cleared from the SoC RTC RAM using the CMOS reset button. To clear the Real-Time Clock data, perform the following steps:

1. Disconnect power form the cRIO controller.
2. Locate the CMOS reset button in the center of the cRIO backplane.
3. Press the CMOS reset button and hold it for 1 second.

Procedure 3 – Primary Storage Disk-on-Chip (OS and User Data):
The Primary Storage DoC can be reformatted to clear the OS and User Data areas. The format operation is a "quick format" that re-initializes the file table, thereby making the existing files inaccessible. Format the drive for this NI Linux Real-Time target by performing one of the following steps:

1. Right-click the controller in MAX and click on "Format Drive".
2. Issue the nisystemformat command via a serial console local connection or SSH remote connection. Visit ni.com/info and enter the info code *format* for details.
3. Write a LabVIEW program that invokes the Format VI of the System Configuration API for the controller.

Procedure 4 – FPGA Storage Flash (User FPGA Bitstream):
The User FPGA Bitstream in the FPGA Storage Flash can be cleared using NI-RIO Device Setup. To clear the bitstream from the flash, perform the following steps:

1. Add the cRIO target to your LabVIEW project by right-clicking on the project and selecting New » Targets and Devices and selecting your cRIO.
2. Right-click on the FPGA project item and select RIO Device Setup.
3. In the *Advanced* section, select *Erase Bitfile on Flash.*

December 2017
377442A-01 Rev 001

Notice: This document is subject to change without notice.
For the most recent version, visit ni.com/manuals.

Contact: 866-275-6964
support@ni.com

Security Technical Implementation Guide

Configuration Management

Supply Chain Risk Management

Flaw Remediation

STIGs

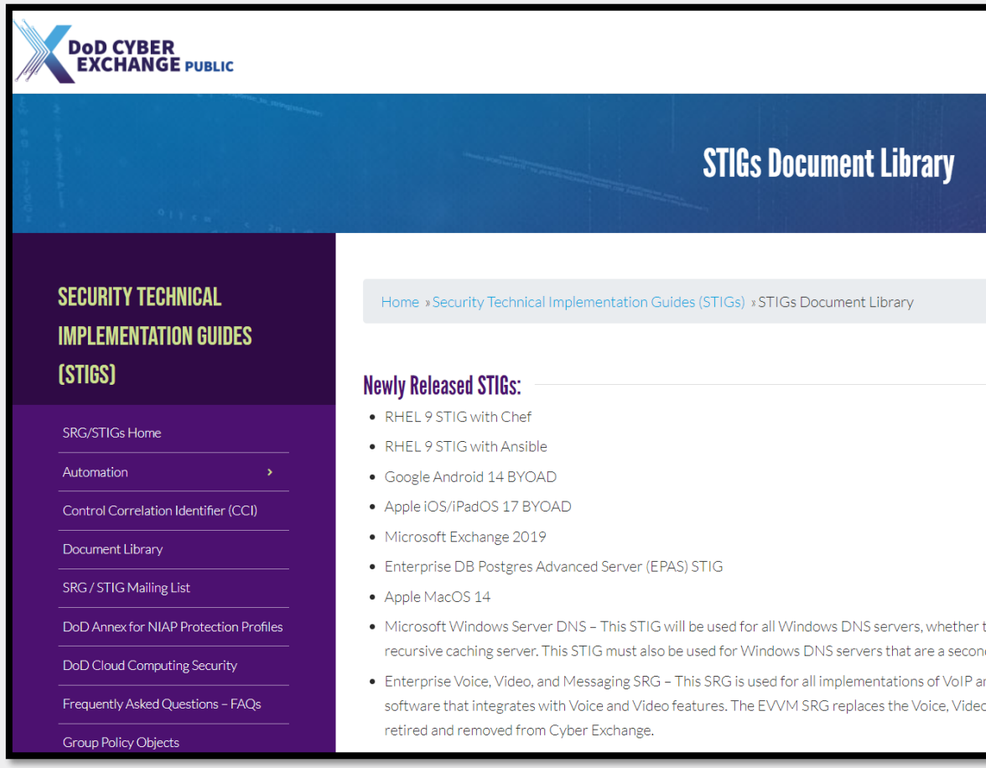
- https://public.cyber.mil/stigs/downloads/
- Approved, tested by DISA
- Instructions to test that system is in maximum secure configuration
- XML for automated scans

LabVIEW Run-Time Engine:

- DISA: Too few configurations, STIG not required.

Other Products:

- In work now – NILRT, SystemLink
- Next: TestStand



EMERSON | ni

Software Bill of Materials

Flaw Remediation

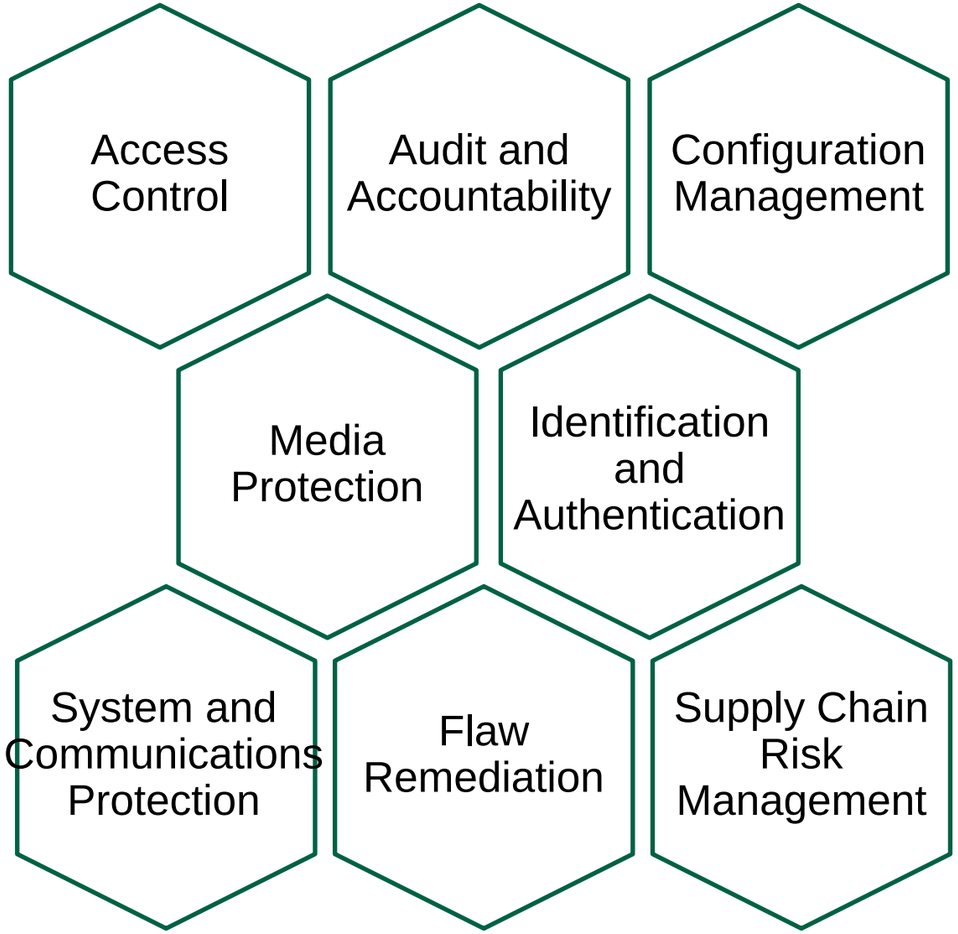
Supply Chain Risk Management

SBOMs for NI Products

- NILRT: SBOM available on request.
- SystemLink: SBOM available on request.

- LabVIEW: SBOM in final work. Scheduled for 2024 Q3 release.

NI 800-171 Control Families – NI RT Devices



Know who is using the system

Control what users do on the system, by role

Record user actions on the system

Protect activity logs

Control configuration of the system

Control connections to external systems and devices

Scan for and fix vulnerabilities

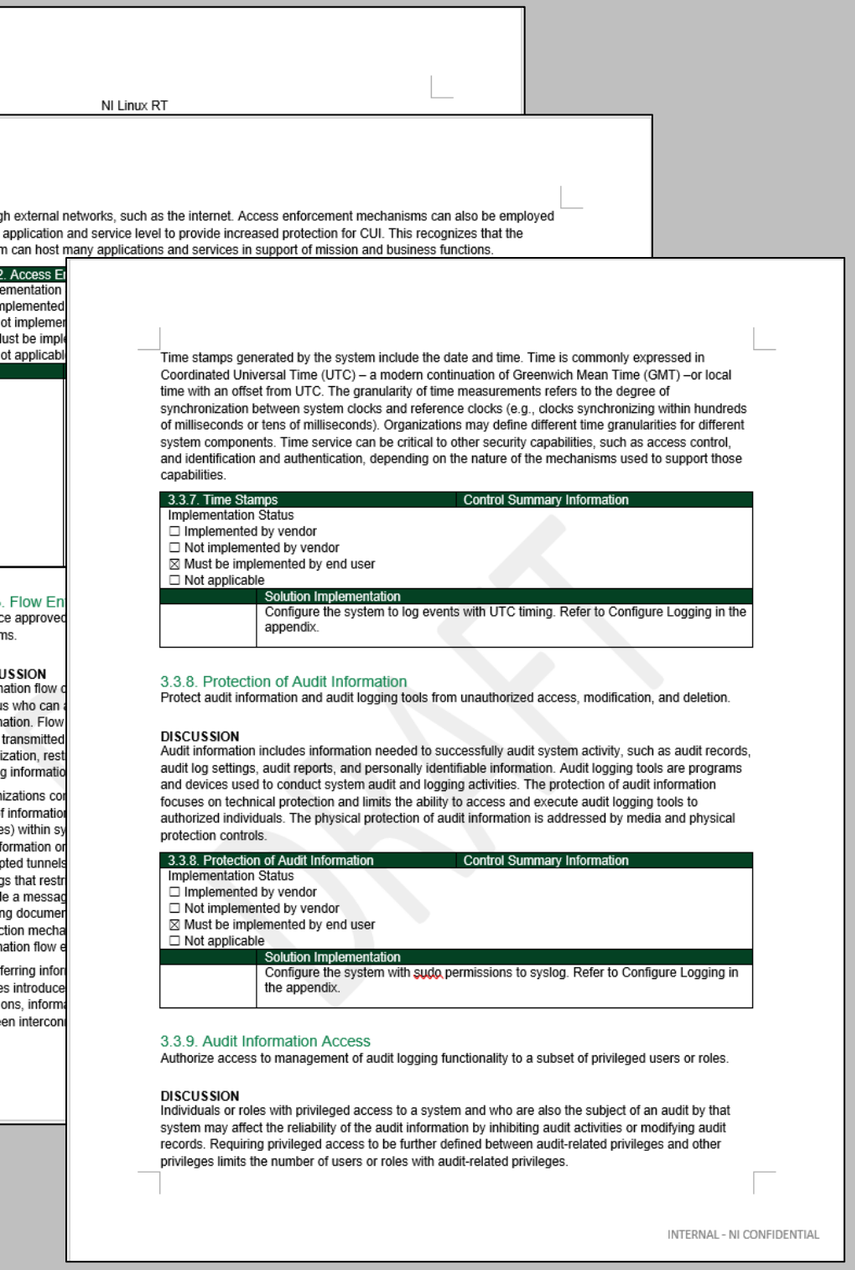
Protect data in storage

Protect data in transit

Maintain Software Bill of Materials

Manage suppliers for security

Dispose of systems properly



NI Linux RT

gh external networks, such as the internet. Access enforcement mechanisms can also be employed application and service level to provide increased protection for CUI. This recognizes that the m can host many applications and services in support of mission and business functions.

Time stamps generated by the system include the date and time. Time is commonly expressed in Coordinated Universal Time (UTC) – a modern continuation of Greenwich Mean Time (GMT) –or local time with an offset from UTC. The granularity of time measurements refers to the degree of synchronization between system clocks and reference clocks (e.g., clocks synchronizing within hundreds of milliseconds or tens of milliseconds). Organizations may define different time granularities for different system components. Time service can be critical to other security capabilities, such as access control, and identification and authentication, depending on the nature of the mechanisms used to support those capabilities.

3.3.7. Time Stamps	Control Summary Information
Implementation Status ☐ Implemented by vendor ☐ Not implemented by vendor ☒ Must be implemented by end user ☐ Not applicable	
	Solution Implementation
	Configure the system to log events with UTC timing. Refer to Configure Logging in the appendix.

3.3.8. Protection of Audit Information

Protect audit information and audit logging tools from unauthorized access, modification, and deletion.

DISCUSSION

Audit information includes information needed to successfully audit system activity, such as audit records, audit log settings, audit reports, and personally identifiable information. Audit logging tools are programs and devices used to conduct system audit and logging activities. The protection of audit information focuses on technical protection and limits the ability to access and execute audit logging tools to authorized individuals. The physical protection of audit information is addressed by media and physical protection controls.

3.3.8. Protection of Audit Information	Control Summary Information
Implementation Status ☐ Implemented by vendor ☐ Not implemented by vendor ☒ Must be implemented by end user ☐ Not applicable	
	Solution Implementation
	Configure the system with sudo permissions to syslog. Refer to Configure Logging in the appendix.

3.3.9. Audit Information Access

Authorize access to management of audit logging functionality to a subset of privileged users or roles.

DISCUSSION

Individuals or roles with privileged access to a system and who are also the subject of an audit by that system may affect the reliability of the audit information by inhibiting audit activities or modifying audit records. Requiring privileged access to be further defined between audit-related privileges and other privileges limits the number of users or roles with audit-related privileges.

INTERNAL - NI CONFIDENTIAL

NIST 800-171 Compliance Document

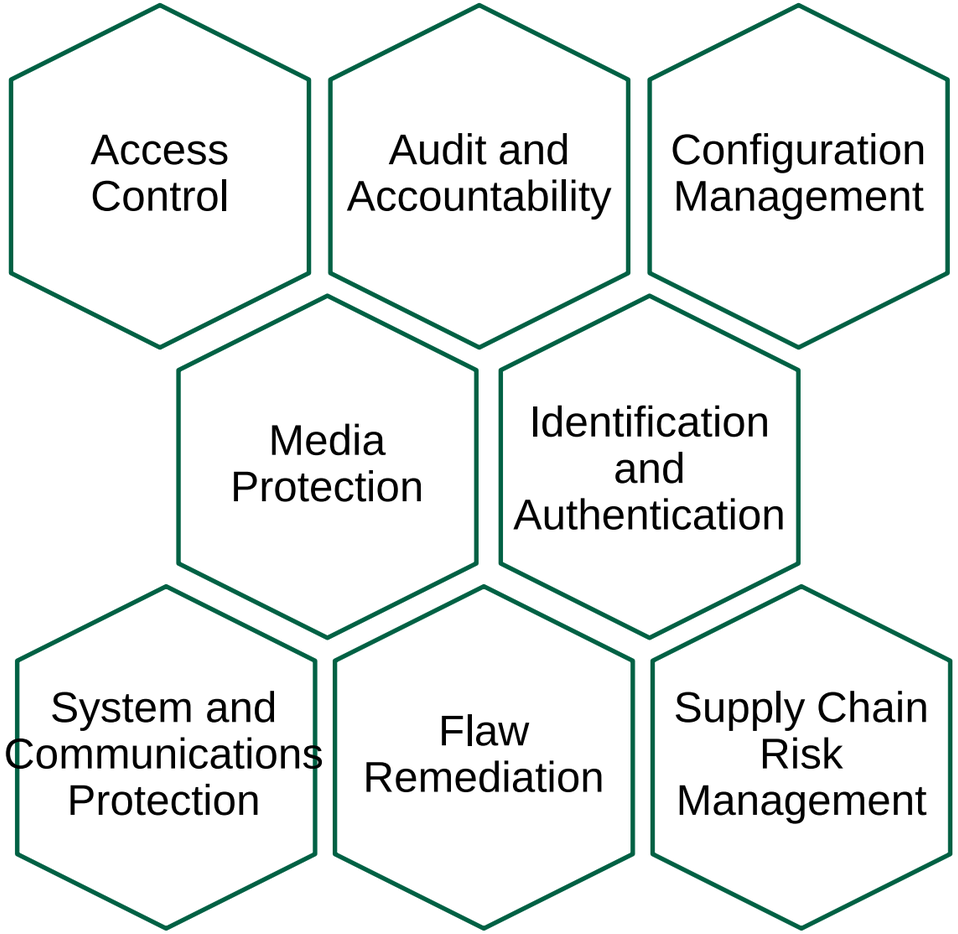
NILRT responses to each of the 110 controls in NIST 800-171

In final DRAFT now

BUT…

Requires a specific configuration of the NILRT device

NI 800-171 Control Families – NI RT Devices



Know who is using the system

Control what users do on the system, by role

Record user actions on the system

Protect activity logs

Control configuration of the system

Control connections to external systems and devices

Scan for and fix vulnerabilities

Protect data in storage

Protect data in transit

Maintain Software Bill of Materials

Manage suppliers for security

Dispose of systems properly

Secure Configuration

- **Install Runmode**
- **Remove/Disable NIAUTH**
- **Configure Opkg**
- **Install Wireguard**
- **Block Encryption (data at-rest security) with cryptsetup**
- **Disable console output**
- **Set up network firewall rules with iptables**
- **Set up an application firewall**
- **Restrict USB (Peripheral) Access / USBGuard**
- **Set up NTP**
- **Install PAM Tools**
- **Install SSH**
- **Configure Logging**

Configuration Impact

This configuration will change the way that users interact with NILRT, in the following ways:

- User Management. Users will be managed by the Linux PAM system, not NIAUTH.
- Without NIAUTH, LabVIEW Real-Time will not have access to deploy code to the system. This breaks the standard workflow of programming the real-time system directly in LabVIEW Real-Time.
- LabVIEW Real-Time programs will need to be programmed on a separate device, and then the file moved to the deployed NILRT device using SSH.
- Individuals will not be able to launch a local terminal using a mouse and keyboard attached to the NILRT device.

Configuring NILRT device for security

NILRT Configuration

- Requires familiarity with Linux
- Restricts access to device from LabVIEW RT

Neosoft Technologies

Founded in 2000, NI partner since 2008

System integration and retrofit – Software, mechanic and electric assemblies

Automated test systems

Acquisition and control systems

Embedded RT and FPGA systems

Hardware In the Loop systems

Automated inspection systems

Configuration Topics

- VPN: using WireGuard on LinuxRT
- Firewall: simplified setup !
- Syslog: get traces !
- NeoRTC: disconnect your system !

VPN: WireGuard on NI LinuxRT

CONNECT

Virtual Private Network

What is a VPN ?

CREATE A SECURE CONNECTION TO ANOTHER NETWORK

ENCRYPTS YOUR NETWORK TRAFFIC

ROUTE YOUR TRAFFIC

Solutions on NI targets

VPN solutions are available on LinuxRT !

Existing and (documented) solution for a VPN : OpenVPN

OpenVPN is heavy, not the most secure

Undocumented / hidden solution : Wireguard !

WireGuard is lightweight and efficient !

OPENVPN®

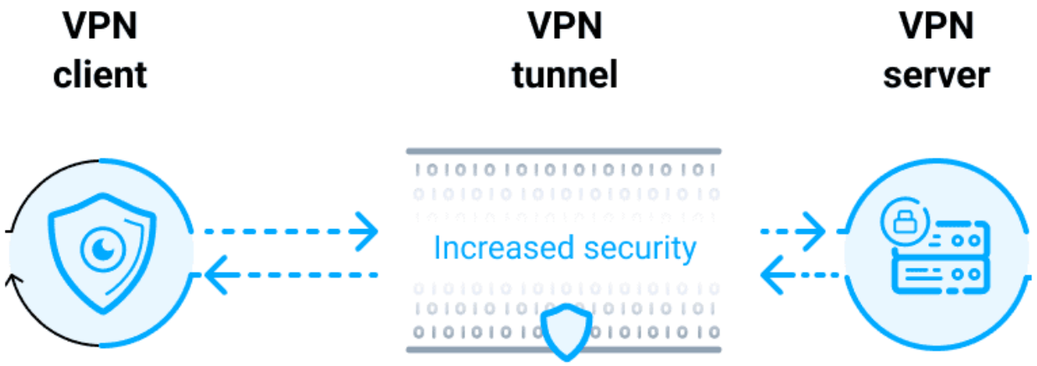
WIREGUARD
FAST, MODERN, SECURE VPN TUNNEL

What is WireGuard ?

WIREGUARD
FAST, MODERN, SECURE VPN TUNNEL

Included in Linux kernel >= v5.5

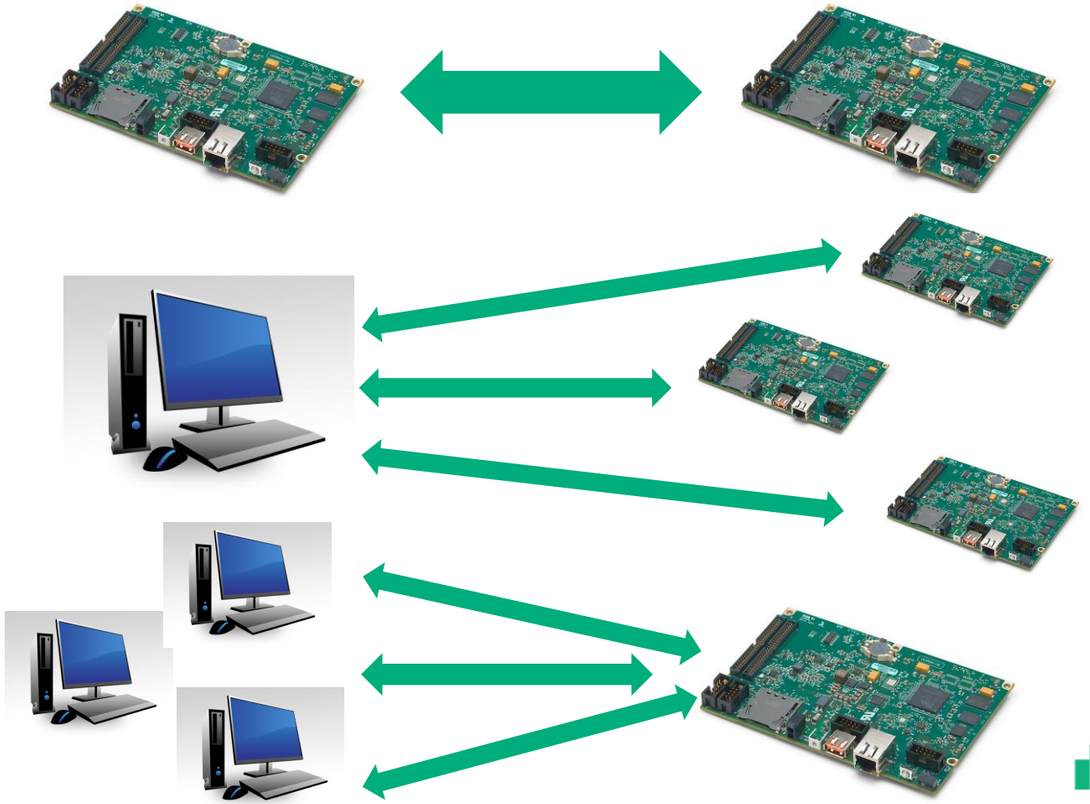
- So efficient and lightweight that it’s part of the Linux kernel
- *WireGuard is designed as a general-purpose VPN for running on embedded interfaces and super computers alike. Initially released for the Linux kernel, it is now cross-platform (Windows, macOS, BSD, iOS, Android) and widely deployable.*
- It is shown as a simple network Interface



Using WireGuard

Does not work by default after installation => Neosoft Technologies scripts to solve this

- Many things are done by the OS at startup to activate Ethernet connectivity
- Nothing done to mount, configure, start / stop the service
- Simple scripts to allow different topologies :
  - Target to target
  - Server
  - Client



WireGuard: benefits for encryption

Very simple to secure a network connection

- Identify communicating systems
- Setup the service using the scripts
- Change destination IP addresses
- That’s it !

- Not necessary to handle certificates
- Nothing to change in your LV code (no TLS primitives, no specific encryption for UDP)
- No external libraries to install

Capturing from Ethernet0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.src == 10.137.102.197

No.	Time	Source	Destination	Protocol	Length	Info
216	2.908627	10.137.102.197	10.137.10.132	TCP	102	3079 → 50835 [ACK] Seq=1 Ack=101 Win=5353 Len=48
217	2.909586	10.137.102.197	10.137.10.132	TCP	102	3079 → 50835 [ACK] Seq=49 Ack=101 Win=5353 Len=48
218	2.909586	10.137.102.197	10.137.10.132	TCP	60	3079 → 50835 [PSH, ACK] Seq=97 Ack=101 Win=5353 Len=4
609	7.933123	10.137.102.197	10.137.10.132	TCP	154	3079 → 50835 [PSH, ACK] Seq=101 Ack=201 Win=5353 Len=100
835	12.960234	10.137.102.197	10.137.10.132	TCP	154	3079 → 50835 [PSH, ACK] Seq=201 Ack=301 Win=5353 Len=100
848	13.013906	10.137.102.197	10.137.10.132	SSH	102	Server: Encrypted packet (len=48)
849	13.013906	10.137.102.197	10.137.10.132	SSH	86	Server: Encrypted packet (len=32)

```
> Frame 216: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on interface \Dev
> Ethernet II, Src: Routerboardc_44:23:18 (08:55:31:44:23:18), Dst: VMware_8a:4b:ed (00:50
> Internet Protocol Version 4, Src: 10.137.102.197, Dst: 10.137.10.132
> Transmission Control Protocol, Src Port: 3079, Dst Port: 50835, Seq: 1, Ack: 101, Len: 4
v Data (48 bytes)
    Data: f30300004f040000000000000000000b80300000000000000000000000000000000000000000
    [Length: 48]

0000  00 50 56 8a 4b ed 08 55  31 44 23 18 08 00 45 00   ·PV·K··U 1D#···E·
0010  00 58 88 ef 40 00 3e 06  2d 56 0a 89 66 c5 0a 89   ·X··@·>· -V··f···
0020  0a 84 0c 07 c6 93 63 4f  c7 8d e8 2f e1 2e 50 10   ·····cO ···/··P·
0030  14 e9 53 7e 00 00 f3 03  00 00 4f 04 00 00 00 00   ··S~···· ··O·····
0040  00 00 00 00 00 00 b8 03  00 00 00 00 00 00 00 00   ........ ........
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0060  00 00 00 00 00 00                                  ......
```

Ethernet0: <live capture in progress> | Packets: 1015 · Displayed: 7 (0.7%) | Profile: Default

UDP Stream Reader.vi Front Panel on WG_UDP_Tester.lvproj/My Computer

File Edit View Project Operate Tools Window Help

15pt Application Font

STOP

UDP Streamer must be running first!

Direct IP Address: 10.137.102.197

Wireguard interface IP: 192.168.213.2

Enable WireGuard

Total Heartbeats: 18

Data Length: 2000

UDP Chunks Dropped

Data Rate (bytes/second): 350000, 300000, 250000, 200000, 150000, 100000, 50000

52552 — Time — 52653

Average Rate: 105.0kbytes/second

WG_UDP_Tester.lvproj/My Computer

UDP Streamer.vi on WG_UDP_Tester.lvpr...

File Edit View Project Operate Tools Window Help

#Clients: 0

Stop

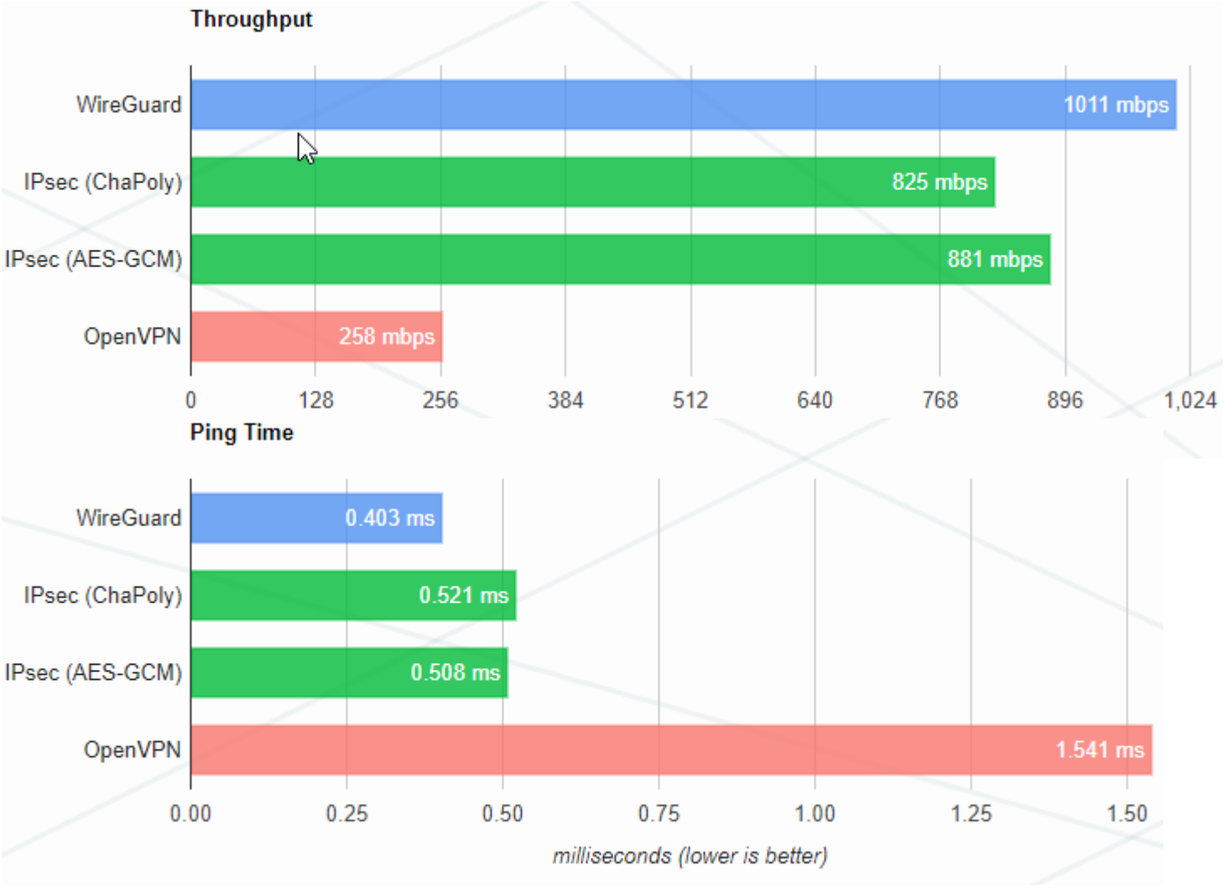
Send Packet Every (ms): 20

UDP Data Packet Size (bytes): 2000

ENG CMS 4:15 PM 2024-03-12

WireGuard: performance

In theory



In practice

Windows PowerShell

```
PS C:\Users\Neo\Desktop\iperf-3.1.3-win64> ./iperf3 -R -c 10.137.2.144
Connecting to host 10.137.2.144, port 5201
Reverse mode, remote host 10.137.2.144 is sending
[  4] local 10.137.10.132 port 58136 connected to 10.137.2.144 port 5201
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-1.00   sec   109 MBytes   915 Mbits/sec
[  4]   1.00-2.00   sec   112 MBytes   938 Mbits/sec
[  4]   2.00-3.00   sec   112 MBytes   940 Mbits/sec
[  4]   3.00-4.00   sec   111 MBytes   932 Mbits/sec
[  4]   4.00-5.00   sec   112 MBytes   940 Mbits/sec
[  4]   5.00-6.00   sec   111 MBytes   933 Mbits/sec
[  4]   6.00-7.00   sec   113 MBytes   944 Mbits/sec
[  4]   7.00-8.00   sec   111 MBytes   931 Mbits/sec
[  4]   8.00-9.00   sec   112 MBytes   937 Mbits/sec
[  4]   9.00-10.00  sec   111 MBytes   934 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec  1.09 GBytes   935 Mbits/sec    2             sender
[  4]   0.00-10.00  sec  1.09 GBytes   935 Mbits/sec                  receiver

iperf Done.
PS C:\Users\Neo\Desktop\iperf-3.1.3-win64> ./iperf3 -c 192.168.213.2 -R
Connecting to host 192.168.213.2, port 5201
Reverse mode, remote host 192.168.213.2 is sending
[  4] local 192.168.213.1 port 58138 connected to 192.168.213.2 port 5201
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-1.00   sec  84.2 MBytes   706 Mbits/sec
[  4]   1.00-2.00   sec  85.7 MBytes   719 Mbits/sec
[  4]   2.00-3.00   sec  84.6 MBytes   710 Mbits/sec
[  4]   3.00-4.00   sec  86.8 MBytes   728 Mbits/sec
[  4]   4.00-5.00   sec  85.3 MBytes   715 Mbits/sec
[  4]   5.00-6.00   sec  84.2 MBytes   707 Mbits/sec
[  4]   6.00-7.00   sec  84.4 MBytes   708 Mbits/sec
[  4]   7.00-8.00   sec  83.9 MBytes   704 Mbits/sec
[  4]   8.00-9.00   sec  84.6 MBytes   710 Mbits/sec
[  4]   9.00-10.00  sec  82.4 MBytes   691 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec   847 MBytes   710 Mbits/sec    0             sender
[  4]   0.00-10.00  sec   846 MBytes   710 Mbits/sec                  receiver

iperf Done.
PS C:\Users\Neo\Desktop\iperf-3.1.3-win64>
```

Firewall: Simplified setup

CONNECT

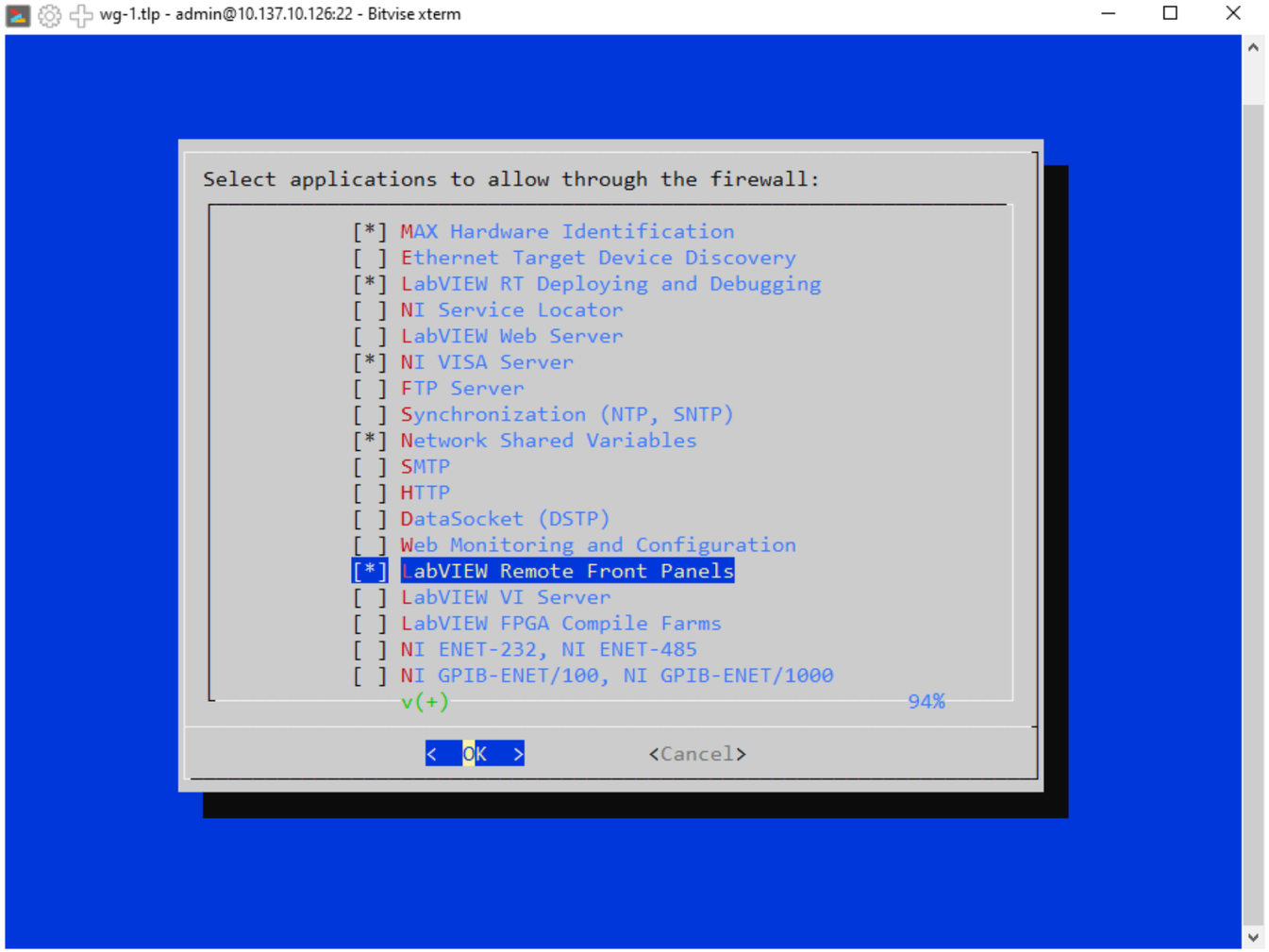
Firewall: a solution exists!

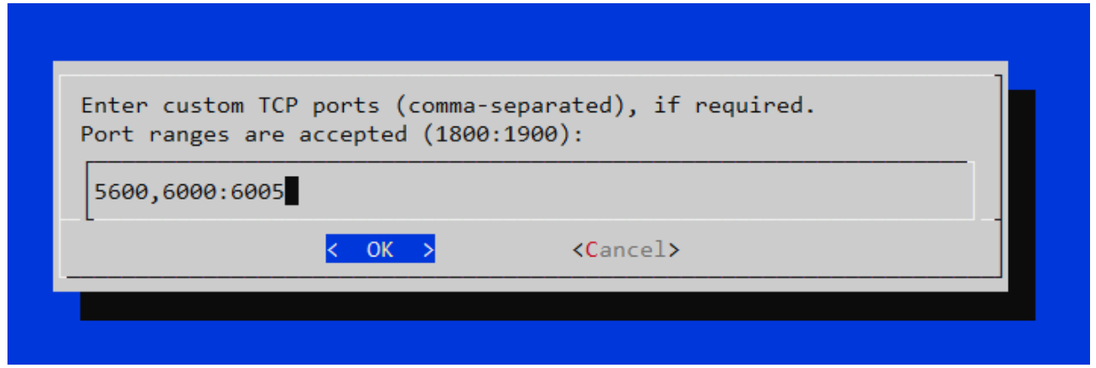
Did you know ? A firewall is installed and functional on NI LinuxRT targets !

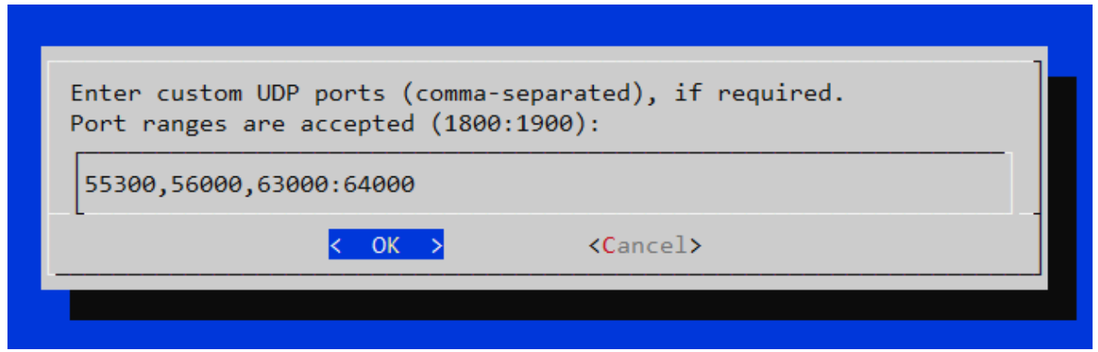
- Low-level firewall : IPTABLES
- Very powerful and performant
- Active by default … authorize all connections !
- Not user-friendly
- Can block completely your target access

Firewall: interactive setup !

Neosoft’s wizard with dialog boxes through SSH will guide you !







Firewall: interactive setup !

IPTABLES rules generated for you !

wg-1.tlp - admin@10.137.10.126:22 - Bitvise xterm

```
# Generated by iptables-save v1.8.7 on Thu Apr 11 12:19:22 2024
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state INVALID -j DROP
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p tcp -m tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -m state --state NEW,ESTABLISHED -j ACCEPT
-A INPUT -p udp -m udp --dport 44515 -j ACCEPT
-A INPUT -p udp -m udp --dport 44525 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 44516 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 3079 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 3537 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 2343 -j ACCEPT
-A INPUT -p udp -m multiport --dports 6000:6010 -j ACCEPT
-A INPUT -p tcp -m multiport --dports 59110:60000 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 8000 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 433 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 5600 -j ACCEPT
-A INPUT -p tcp -m multiport --dports 6000:6005 -j ACCEPT
-A INPUT -p udp -m udp --dport 55300 -j ACCEPT
-A INPUT -p udp -m udp --dport 56000 -j ACCEPT
-A INPUT -p udp -m multiport --dports 63000:64000 -j ACCEPT
-A FORWARD -m state --state INVALID -j DROP
-A OUTPUT -o lo -j ACCEPT
-A OUTPUT -m state --state INVALID -j DROP
-A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A OUTPUT -p tcp -m tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
COMMIT
# Completed on Thu Apr 11 12:19:22 2024
admin@LinuxRT-2987e98c:~#
```

Syslog: get traces !

What is Syslog (RFC 5424) ?

Standard Network protocol to get messages from embedded devices

- Syslog is a standard for message logging over the network (RFC 5424)
- Allows for centralized collection, filtering, and analysis of log messages from multiple sources.
- Widely used for system management and troubleshooting in IT.
- UDP port 514
- TCP possible

How to use Syslog ?

Toolkits and daemon available on NI LinuxRT

- Publish messages from LabVIEW with client toolkits
- Syslog-ng daemon is running on NI Linux RT targets
- Syslog server to receive messages from 1-N targets
- Syslog-ng publishes log file content over Syslog
- Read dmesg, auth.log, boot.log, lastlog, …
- Customizable by creating .conf file in /etc/syslog-ng.d

Collect Syslog Messages

Syslog Server

- Clients connect to an identified Server
- A server can monitor several targets
- Displays date-time, criticality, origin and content of messages
- Cloud services available (Loki, Graylog, LogTail)

NeoSyslog Collector

Fonctionnalities	Free	Paid
Live visualization	✓	✓
Configurable	✓	✓
Visualization by message type(last value)	✗	✓
Recording to internal database	✗	✓
Internal database query for post analysis	✗	✓
Import / Export recorded data	✗	✓
Print selected messages	✗	✓

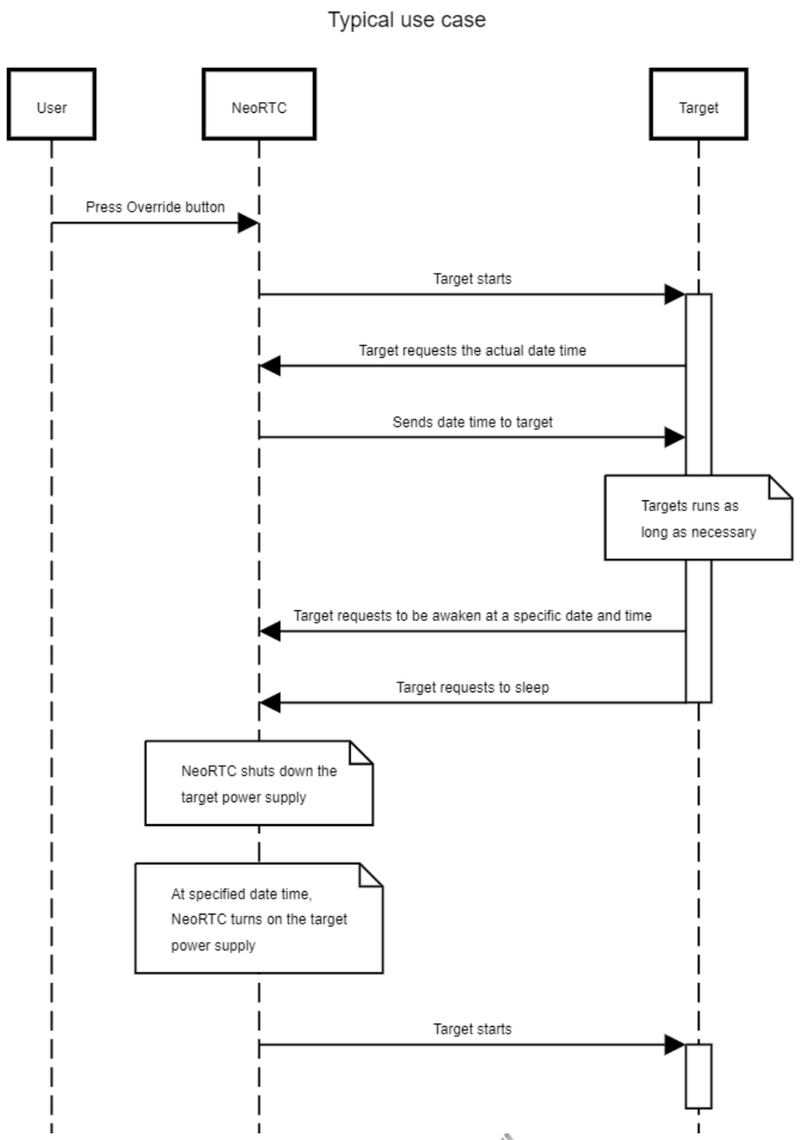
NeoRTC:
not connected, not hacked !

CONNECT

NeoRTC: Turn off your targets !

Programmatically turn OFF <=> ON your targets

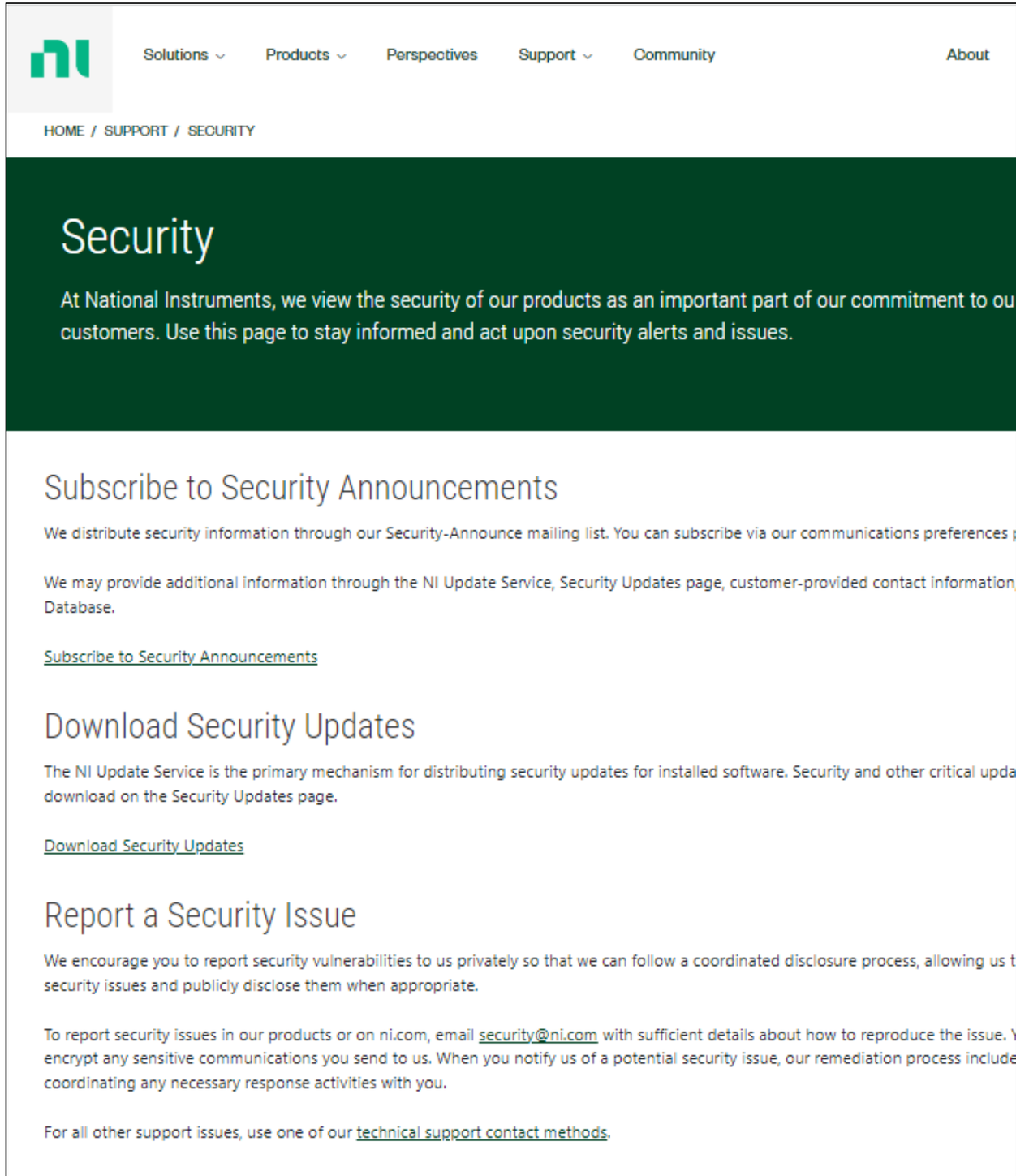
- Schedule your system to power on/off at specific times
- Integrated real-time clock to keep your target's time
- Configurable via RS232
- Internal backup battery (long life)




This is a journey

NI Resources for security

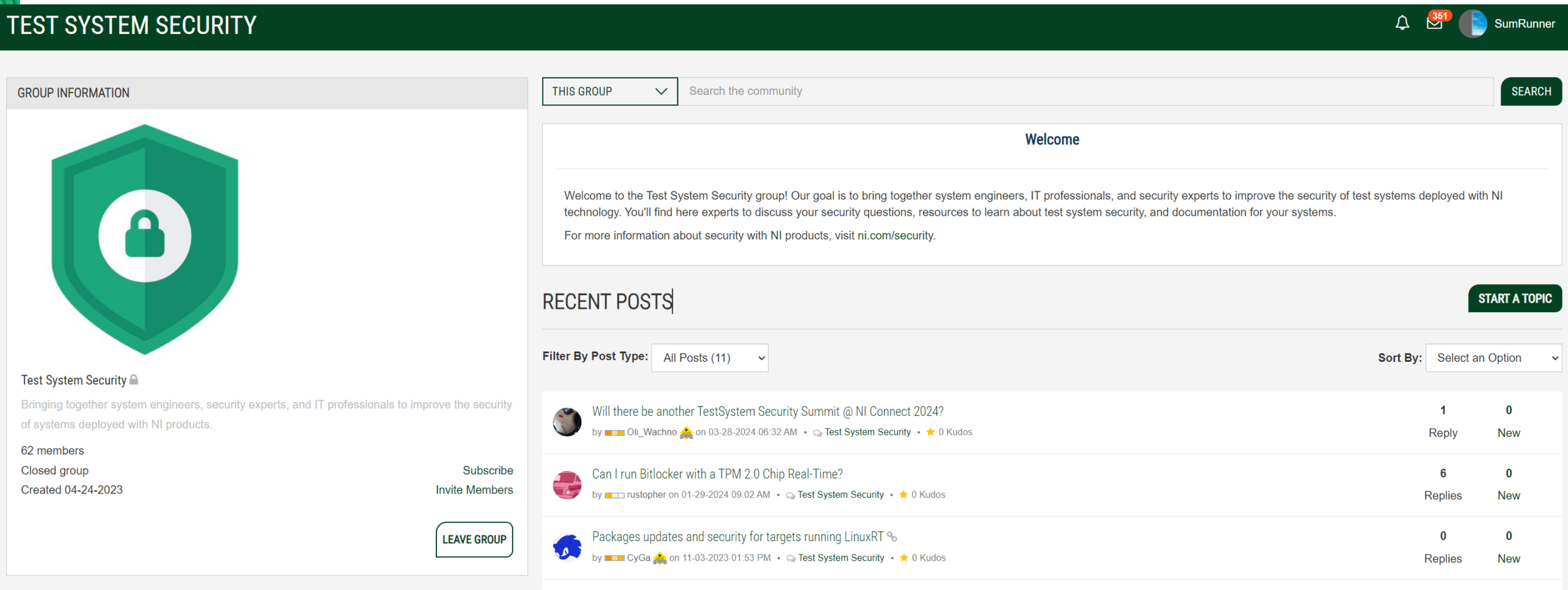
- **ni.com/security** – first stop for security information
- **security@ni.com** – report issues, request information
- **Letters of volatility** – with product manuals or at ni.com/letters-of-volatility
- **Secure development guides** – at ni.com/security
- **Additional security documentation** – available on request



Test System Security Forum

- Online Forum – Join for access
- Next meeting (virtual) – June 18 2024

  Join forum to receive invites



NI Test System Security Summit

Semi-annual meeting for test engineers, security teams, and IT professionals

Online forum to access discussions, presentations

Next meeting: June 18 2024

To be invited:

Email steve.summers@ni.com

nI CONNECT

2024 AUSTIN

EMERSON