absio®
PERSISTENT INFORMATION OWNERSHIP TM

– A Comprehensive Analytical Framework for Cybersecurity –

This document contains our comments in response to NIST's "Developing a Framework to Improve Critical Infrastructure Cybersecurity" RFI.

Summary

We provide a high level overview of simple analytical framework that would work across all industry sectors for both conventional IT and real-time digital information systems and that can serve as information sharing vehicle. Additionally, we address some of the fundamental vulnerabilities of IT and real-time systems that need solutions.

Physicality

Digital information is physical. It is stored and transmitted in various forms such as alternating current, radio waves, light pulses, pits and grooves on an optical disk, or ferrous particles on disk or tape. It possesses empirically measurable physical properties. Digital information and the systems that enable its creation, processing, storage, and transport are real in the same way base chemicals, chemical plants, storage tanks, and pipelines are real. Just as base chemicals must be processed and transported to become useful, electrons, radio waves and light must be processed and transported by software and hardware to be useful.

The Technology Maturation Cycle

Digital technology and the connected world are young; TCP/IP was standardized only 31 years ago. History demonstrates that all new technological processes go through a painful but predictable maturation cycle:

1. Make it work
2. Make it scalable
3. Scale up production
4. Lower costs
5. Increase reliability
6. Increase safety

The historical and ongoing evolution of automobiles, aircraft, mining, textiles, petrochemical and all other sorts of other manufacturing and construction technologies validate this maturation cycle. Every step forward in the technical maturation cycle stems from discovery of a needed improvement followed by modification of the design and frequently, changes to engineering and administrative controls.

Safety comes late in the maturation process because it requires accumulated knowledge and enough free operating capacity to bear the overhead that it requires. Increases in safety, like each of preceding steps of the maturation cycle, are applied incrementally based on analyses to determine which improvements should be made and what order. Historically, rapid increases in safety were achieved when commonly acceptable analytical frameworks became available. The purpose of a comprehensive analytical framework is to identify vulnerabilities, assess potential solutions and determine the schedule of implementation weighing (among others) criticality of need, effectiveness, potential for disruption and cost.

Designs and Controls provide Safety

Design comprehends the materials, energy, and processes that produce the output for which the system is built. Controls tell systems and humans how to operate. Engineering controls operate in the domain of physics; administrative controls operate in the domain of interactions between humans and systems. A car's gas pedal, brakes, and speedometer are engineering controls; speed limit signs, traffic laws, and driver training are administrative controls. Design and controls properly implemented provide safety in normal operation and in, non-normal operation, that is, operation in the event of failures or attacks.

Safety produces Security

This is a critical distinction: Security can only be achieved when systems are safe. Security is the state achieved when adequate designs and controls consistently produce safety in all operating modes.

Current State

Digital information systems not only work, they work exceedingly well. Usage rates are accelerating, costs are plummeting, and the technology is reliable enough to have permeated every part of our critical infrastructure—*but digital technology has not yet been made safe*.

The current and potential costs of unsafety are rising, but a rigorous analytical framework to produce rational incremental implementation plans is absent. Other industries with complex processes use commonly accepted analytical frameworks (Failure Modes and Effects Analysis, Fault Tree Analysis, Probabilistic Risk Assessment, etc.) to guide their safety implementation plans, but the digital information industry, by and large does not.

Digital information and the systems that enable its creation, processing, storage, and transport possess physicality; we do not have to reinvent the wheel. We propose that a Level of Protection Analysis (LOPA) used for system safety analysis in the petrochemical industry, modified for digital information systems, is a suitable analytical framework to continuously assess critical infrastructure cybersecurity. LOPA provides a simple but effective framework for deciding if current protections are adequate or if more protections are needed, and it assists in choosing between alternates and implementation order.

What Are We Protecting?

Ultimately, what we are protecting is *data*. Designs and controls for cybersecurity must protect against:

1. Exfiltration of Data: Copying usable data and putting it in the possession of one or more unauthorized persons.
2. Falsification of Data: Unacceptable alteration of data in storage-based systems (conventional IT) and real-time systems (industrial controls).
3. Disrupting the Flow of Data: Denial of service, physical interruption of communication links, signal jamming, and the like.
4. Destroying Data: The unrecoverable erasure of stored data.

Limiting the definition of what needs protecting to controlling access to the boxes data is stored in and the pipes that it is transported through (perimeter security) is insufficient. While perimeter security is required it is not sufficient.

What is being attacked?

One definition of “venue” is “the place of a crime or cause of action.” We start by segmenting digital information systems into logical attack venues and their corresponding vulnerabilities. Some examples of venues and vulnerabilities are:

People

Vulnerabilities related to:

- Policies/Instructions/Training
- Social Engineering

Data
Objects

Vulnerabilities related to:

- Distribution controls
- Audit
- Provenance

Applications

Vulnerabilities related to:

- Behavior Verification
- Application Authentication
- Source Authentication
- Software patching and updating

User Identification and Authentication

Vulnerabilities related to:

- Human user identification
- System user identification
- Host hardware identification
- Ongoing authentication of human and system users and host hardware

Provisioning

Vulnerabilities related to:

- Mismatching devices to users
- Software installation
- Decommissioning (hardware, software and users)

I/O and Communications

Vulnerabilities related to:

- Unauthorized sessions
- Unauthorized endpoints
- Eavesdropping
- User I/O

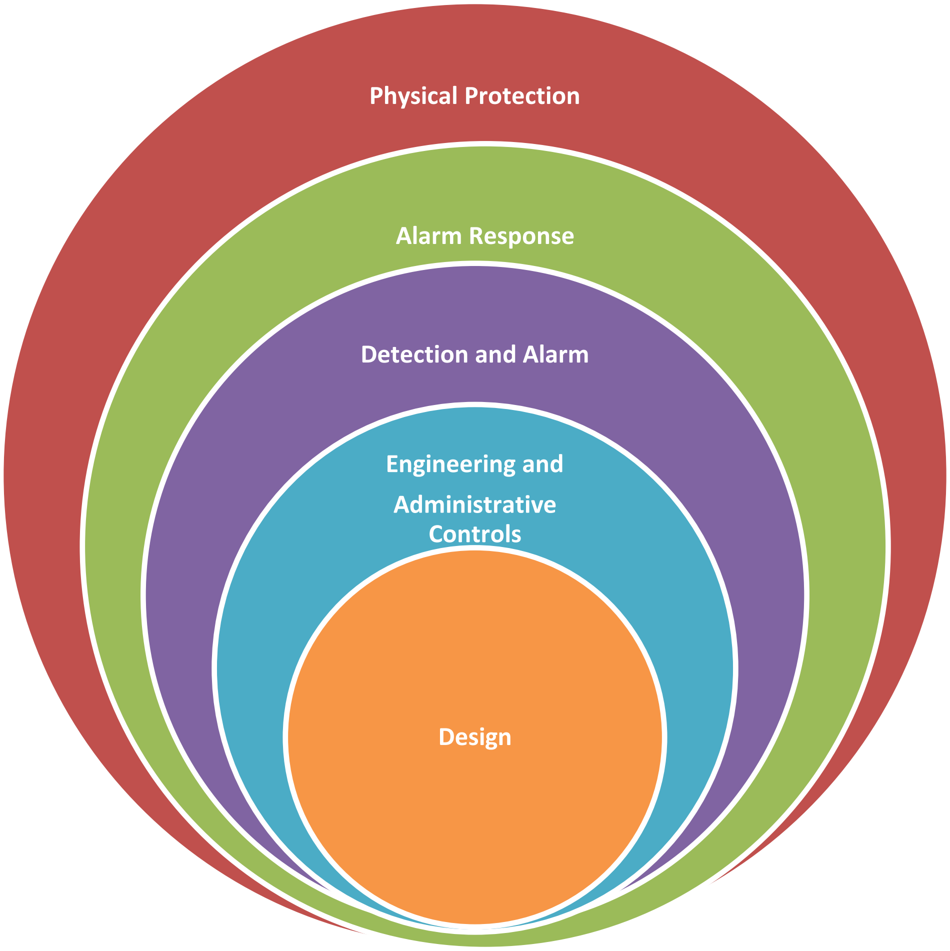
Operating System

Vulnerabilities related to:

- Unauthorized access to storage, memory, and processors
- . . . (too many others to list here)

The LOPA Process

1. A list of exploits is developed and a specific exploit is selected for examination.
2. Protections against the exploit are examined in the context of each protection layer and attack venue.
3. The results of each examination are evaluated to determine if existing protections are adequate, and if not, the efficacy, cost and time to implement modifications to existing protections, or implement new protections are determined.
4. An implementation plan consisting of protections to be modified or added, along with their costs and an implementation timeline is developed.



Physical Protection: Protections afforded by static physical safeguards and/or physical actions are examined.

Alarm Response: Protections afforded by the actions taken in response to detections and alarms are examined.

Detection and Alarm: Protections afforded by exploit detection and alarms are delivered are examined.

E&A Controls: Protections afforded by engineering and administrative controls are examined.

Design: Vulnerabilities arising from the way the system, as built, are examined at the Design level. Design level changes usually require rebuilding or replacement.

Core Attributes of the Layers of Protection

1. **Independence:** The performance of a protection layer is not affected by the failure of other protection layers.
2. **Reliability**: The probability that protection will operate as intended.
3. **Auditability:** The ability to examine computer logs in order to validate proper operation of the protection and documentation pertaining to the design, inspection, maintenance, testing, used to achieve other core attributes.
4. **Access Security**: Measures taken to reduce the potential for unintentional or unauthorized changes.
5. **Change Management:** The process used to review, document, and approve modifications to a protected system other than "replacement in kind," prior to implementation.

Example Examination

Once an exploit is selected, the process is to work within each intersection of a venue and a level of protection. If possible, all intersection should contain with current or needed protections. Some intersections will have multiple potential protections. The process is to list all current and potential protections, and begin a winnowing process based on efficacy, cost, timeline, and practicality. If no protections for a given exploit are needed, the rationale for that finding is recorded. If modified or new protections are needed, those that survive the winnowing process will be incorporated into an implementation plan.

Following are examples of an examination worksheet and a high-level list of protection needs.

Exploit: **Unauthorized export of data by an authorized user via CD/DVD**							
Venue → Level of Protection ↓	V1: Operating System	V2: Local I/O And External Communications	V3: Provisioning	V4: User Identification/ Authentication	V5: Applications	V6: Data Objects	V7: People
L1: Design	Build granularly encrypted file system				Build in data export controls to applications	Build in object level distribution control	
L2: Engineering and Administrative Controls			Issue computers without CD/DVD Drives	Log all user interactions with discrete data objects	Add data export controls to applications	Add object level distribution control	Scan personnel exiting the facility
L3: Detection and Alarms			Install real-time auditing application	Provide authenticated user data to logs	Detect unusual mass copies to CD		
L4: Alarm Response				Revoke user privileges on device	Automatically notify security officer		
L5: Physical Protection		Disable external USB to external CD/DVD drives		Revoke user facility access privileges	Remotely lock SCIF door		Pat down all personnel leaving the facility
Authors: Mitch Tanenbaum, David Kruger			Locator: 1.1.1		Rev 1.0	Updated: 2011-04-10	

The list below contains some example vulnerabilities targeted for prevention, detection, and mitigation. Attack venues are on the left and vulnerabilities with solution requirement by Level of Protection are in the right. The list illustrates a high-level list of protection needs at each Level of Protection Analysis.

Data Objects

No direct controls on data

- Design: Fine-grained controls at the data object level
- E&A Controls: Applications must use object level controls. Information Assurance ensures appropriate object level controls are always used.
- Detection/Alarm: Unapproved object level access is denied and an alarm is generated.
- Alarm Response: Determine if alarm is part of a pattern of behavior and if so escalate the alarm.
- Physical Protection: Distinct object level encryption.

Applications

Applications that bypass security controls

- Design: Applications must use an approved common security framework.
- E&A Controls: Validate that each applications uses the approved common security framework.
- Detection/Alarm: Detect or thwart attempts to bypass common security framework and alarm appropriately.
- Alarm Response: Device management terminates application
- Physical Protection: N/A

User Identification and Authentication

Application identification and authentication

- Design: Applications must authenticate before being allowed to access system resources.
- E&A Controls: Implement a robust application access control mechanism.
- Detection/Alarm: Detect or thwart unapproved access and alarm appropriately.
- Alarm Response: Deny unapproved access and notify administrator(s)
- Physical Protection: Implement software fingerprinting.

Provisioning

Unmanaged software changes

- Design: Only approved changes can be implemented.
- E&A Controls: Implement centrally controlled device management.
- Detection/Alarm: Detect or thwart unapproved changes and alarm appropriately.
- Alarm Response: Approve change or revert to approved configuration
- Physical Protection: Implement software fingerprinting

I/O and Communications

Eavesdropping

- Design: All I/O and communications are provided a secure channel.
- E&A Controls: Set software defaults to use secure channels only.
- Detection/Alarm: Detect and report attempts to use unsecure channels.
- Alarm Response: Device management system executes appropriate response.

- Physical Protection: Ensure cable plant is protected from unauthorized modification.

Operating System

Operating System Configuration

- Design: Create OS configuration(s) to match security requirements
- E&A Controls: Disable select user controls to ensure that user changes cannot subvert the security profile of the OS
- Detection/Alarm: Ensure that any changes to the security are protected and generate an alarm.
- Alarm Response: Deny network access to devices with altered operating systems.
- Physical Protection: Purchase only tamper-resistant devices.

Hardware

Supply Chain Vulnerabilities

- Design: Validate Reference Design and ensure implementation matches reference design
- E&A Controls: Ensure sufficient sampling and testing and implement robust change management process
- Detection/Alarm: Create a robust process to report sampling variance or changes that bypass the change management process
- Alarm Response: Create a robust process to respond appropriately and to escalate if needed to resolve the issue.
- Physical Protection: Ensure that controls are in place to maintain the integrity of hardware distribution (e.g. tamper-evident shipping containers)

LOPA analyses readily serves as an information sharing vehicle. The results of LOPA (with organizational identities redacted if needed) can and should be shared with other organizations operating in a similar environment. Shared LOPA can help with knowledge transfer and fine-tuning of protections as new cyber threats inevitably emerge. LOPA is designed for periodic use, that is, as condition change, i.e. changes to the IT or real-time computing environments or the emergency of new exploits, LOPA can be used to rapidly determine if additional protections are warranted.

However, it is not necessary to conduct LOPA before we can address the most pressing short term issues. In the next section we outline a few well-known fundamental vulnerabilities of IT and real-time systems; many, if not most, current exploits rely on them.

Some Things We Already Know

It is not necessary to wait for the results of LOPA or any other analytical framework to know that these fundamental vulnerabilities must be resolved in order to reduce the current threat level. We suggest to NIST that identifying and publishing other similarly fundamental vulnerabilities would be helpful.

Please note that given the time constraints for submitting comments to NIST, the remainder of our commentary is not as well-organized or complete as we would like. However, as the perfect is often the enemy of the good, we included them with the hope that they will be helpful, albeit incomplete.

Fundamental Vulnerabilities of IT Systems – Data Exfiltration

Illicitly exfiltrating data, that is, procuring copies of data such as personally identifiable financial and healthcare information, office files, copyrighted videos, music, books or games, trade secrets, confidential business and personal communications, diplomatic communiques, or military secrets is arguably the oldest and most common type of exploit. Exfiltration is common because IT systems permit anonymity and uncontrolled distribution of data.

Anonymity facilitates attackers in two ways. First, there is little chance that skilled attackers can be identified and consequently, there is little risk of reprisal. Second, permitting anonymity enables unauthenticated users, applications, or devices to exfiltrate data. Solutions must not permit anonymity; there is no logical reason to permit systems that store or transmit confidential information to allow use by an unauthenticated user, device, or application, or service.

Exfiltration capitalizes on uncontrolled distribution. Although the phrase uncontrolled distribution is not commonly used, we all know exactly what it is. Have you ever emailed a joke? If so, can you know with any certainty that the people you sent it are the only ones that received it? If those you originally sent it to subsequently distributed your joke to others, can you know who has the joke now? Can you know what devices your joke is stored on, or the last time your joke was viewed, or if your joke was altered and then sent to others? If you answered yes to the first question, you probably answered the rest with something like “I have no way of knowing. I intended, but cannot prove that the joke only went to the people I sent it to. I have no control, and therefore no visibility as to how, where, when, or by whom the joke was further distributed, altered, or where it is now stored.” Now substitute “joke” with “confidential data”—that is the uncontrolled distribution problem.

Uncontrolled distribution starts with making a perfect copy of a digital object (a discrete bundle of ones and zeros) and then transporting the copy, not the original, to other devices. Unlike a physical object, there is no “empty space on the shelf” to indicate a data object was removed. Once an uncontrolled data object is distributed, whether by email, download, or copying to removable media, whether it was distributed intentionally or illicitly exfiltrated, control is lost.

In the near term (at least the next few years) data breaches are inevitable; currently available best practices and adherence to standards, even if fully implemented, are insufficient to prevent the distribution of uncontrolled data. Until solutions are implemented to facilitate controlled distribution, there is little chance that successful exfiltration attacks by outsiders and malicious insiders will be abated.

Fundamental Vulnerabilities Real-Time Systems

Real-time critical infrastructure systems are accuracy dependent; regulating temperatures, pressures, flows, and the like requires truthful data. Attacks have a common goal: falsification of real-time data.

Current real-time communications are easy to falsify because controls systems permit *observability* and *anonymity*. Real-time communication protocols are publicly available; if attackers can eavesdrop on real-time communications, they can discern, where, when, and how to inject false signals. One obvious solution (there may be others) is to distinctly encrypt each communication so that eavesdropping or injecting false signals requires attackers to break encryption *in real-time*. While theoretically possible, it would be difficult and expensive.

Anonymity facilitates attackers in two ways. First, as with IT systems, there is little chance that skilled attackers can be identified so there is there is little risk of reprisal. Second, permitting anonymity enables unauthenticated users, applications, devices, or services to mount attacks. There is no logical reason to permit real-time systems to ever accept an input from an unauthenticated user, device, or application, or service. Because it may take only a small number of falsified communications to disrupt or destroy a system, solutions that challenge each individual communication should too prove its authenticity are needed.

Fundamental Constraints of Real-Time Systems

There are constraints particular to real-time systems that solutions need to accommodate. A list of these constraints follows.

- Infrastructure operators will vigorously and justifiably resist solutions calling for wholesale replacement of existing control systems because it would be ruinously expensive, take too long, and there is little evidence that currently available replacements would prove sufficient. Evidence from IT supports their skepticism; it suggests that determined attackers would be able to subvert replacements shortly after installation. Solutions that can be added to existing systems at acceptable cost are needed.

- Interconnected real-time control systems and business networks provide attack vectors to each other; neither can be secured without securing both. Solutions that span real-time control systems and business networks are needed. Point solutions that do not integrate with each other or across the IT/real-time domains are of little value.

- Real-time systems are a mixture of ages, brands, makes, models, communication protocols and media. Solutions must accommodate them all.

- Many real-time controllers lack sufficient computing power, communication speed, or patch tolerance to be safely or economically upgraded. Solutions must overcome these lacks.

- Many current systems do not produce sufficient audit data for real-time threat detection and behavior analysis. Solutions must produce sufficient audit data.

- Security professionals are in short supply. Solutions must reduce, not increase, overall operational complexity; solutions that require non-existent personnel to function cannot solve the problem.

Conclusions

There is a wealth of human knowledge pertaining to the safe operation of complex processes. Absio suggests that taking advantage of lessons learned making other complex processes safe will speed reduction of critical infrastructure cybersecurity risk. One example is adoption of a modified LOPA as a common analytical framework.

The fundamental vulnerabilities of IT and real-time systems should be identified and published. Attackers already know what they are; they are the basis of their exploits. Users, organizers, and solution providers should know what they are and begin addressing them in order to provide long-term stable solutions.