CHAPTER 15

Devising and Maintaining System Inventories

NIST Special Publication 800.82 Revision 2, on cybersecurity for IACS [7], says that "*[t]o properly secure an ICS, there should be an accurate listing of the assets in the system and their current configurations*". That is often called an inventory, and NIST appropriately suggests that the combination of assets may vary in ways determined by their so-called configuration. What does all this mean in precise terms? This chapter attempts to answer this question.

After the answer has been suggested, a comparison of applicable standards show that they do not say precisely what is required in a system inventory. The suggestion here appears therefore to be new, although the author would hope implicitly routine amongst those engaging in best practice.

One of the main reasons for maintaining a system inventory is to maintain the integrity of the system. If assets are modified in an unauthorised manner, the integrity of the system has been violated. One can detect such integrity violations in principle by comparing the asset in use with some parameter encoding an unviolated version of the asset ("digital signing"). Another control mechanism is to control the channels by which asset modifications could be promulgated from outside the system, both legitimately and illegitimately, which requires an inventory of external channels and requirements on their configuration, authorisation and use.

15.1 Terminology

We shall be considering the construction of an inventory of assets. So we nee a a definition of "asset":

asset: (definition 3.2.6 of [5]): physical or logical object owned by or under the custodial duties of an organization, having either a perceived or actual value to the organization

We are talking here about "computational systems", which in the case of process control systems are often distributed, which means networked to some extent, and increasingly digital-technology-based. I propose the following definitions.

digital unit: physical object with digital-computational behaviour which contains a microprocessor, FPGA or other digital-electronic elements.

software: a computer program. There are many different types of software, usually denoted as <type-of-software> code.

object code: a digital-computer program in the machine language of a digital unit contained in the memory of the unit, which implements the functionality of the unit in the specific application.

source code: a digital-computer program written in a so-called higher-level language, which is transformed through a compiler and linker into object code for a digital unit.

firmware: digital-computer program used by the supplier of a digital unit to implement the functionality of the unit independent of its application, contained in non-volatile memory (ROM, EPROM, flash memory).

Note that firmware is a particular type of software. It is so handled in NIST 800-82R2 [**?**], ISA TR84.00.09-2017 [6] and HSE-OG86 [1] with respect to its vulnerability properties (the term does not appear in IEC 62443-1-1:2009 [5]). There are other assets in a distributed system which are not included above, namely

network cable: a physical cable designed for carrying digital signals between digital units according to a defined protocol.

We can now define

physical digital asset: asset which is either a digital unit or a network cable. digital asset: physical digital asset or software.

digital asset type (datype): digital assets can be clustered into types depending on their general system functionality. For example, sensor, actuator, controller, router, switch, server, operator station/terminal, workstation, laptop computer, tablet computer (tablet), firewall, I/O (input-output device), cable, Ethernet cable, Profibus cable. This category is malleable. Digital asset version

logical connection: a logical connection exists between two digital units if there is a sequence of cables connecting the two units, perhaps via intermediary units.

architecture: a table or diagram showing the physical digital assets and their logical connections

reference architecture: a diagram showing the physical digital assets in terms of datype and their logical connections. Multiple instances of assets and their logical connections may be elided, providing this is evident in the diagram. An abstraction of the architecture. Reference architectures are a concept used widely in IEC 62443-1-1:2009, in particular for defining zones and conduits.

(security) zone: (definition 3.2.117 of [5], including notes) grouping of logical or physical assets that share common security requirements. A zone has a clear border with other zones. The security policy of a zone is typically enforced by a combination of mechanisms both at the zone edge and within the zone. Zones can be hierarchical in the sense that they can be comprised of a collection of sub-zones.

conduit: (definition 3.2.27 of [5], including notes): logical grouping of communication assets that protects the security of the channels it contains. This is analogous to the way that a physical conduit protects cables from physical damage.

An architecture gives us some information, but not enough, about what system we are running. It lists the physical assets and interconnections, but not software (including firmware). Digital assets usually have an identifier and a version. Physical assets usually have a name and a model number, for example I have a LinkSys Wireless-G Broadband Router (name, also indicating what it does) Model Number WRT54GL. It is over a decade old. If vulnerabilities or logical faults have been found in this device, it is likely that updated firmware has been issued. Firmware, and other software, is issued with a Model Number, usually called Version (Number). The functional behaviour of this device, including vulnerabilities and failure behaviour, is dependent upon the device, its model number and its firmware version. Thus we

have

digital unit inventory specification: the name of the digital unit, along with its model number and the version(s) of firmware running on the unit. A digital unit inventory specification can be represented as:

$$< \text{digital unit name}, \text{model number}, \text{firmware version number} >$$

If other software than firmware runs on the unit, then the functional behaviour of the unit is dependent not only upon the parameters above, but on the software Name and Version. The software will often be nested: there will often be an Operating System which extends the physical machine with its processor, registers and specific memory addresses to a "virtual machine" with extended functionality and a relative lack of dependence upon physical assets – for example, memory is not specifically addressed, but a chunk of memory is made available to a program called Application Software and its physical behaviour is managed by the Operating System software, not the Application Software. The virtual functionality made available is often called the Application Programming Interface (API). Software thus may be a collection of Application Software using the API of an Operating System. We can call this a

software bundle: an Operating System along with a collection of software using the API of an Operating System, running as one computer program on one or many physical digital assets.

software bundle inventory specification: an unordered list of Operating System with Version and Application Software with Version running in a software bundle. A software bundle inventory specification has the form

$$< \text{operating system}, \text{version} >$$

$$< \text{application software 1 }, \text{version} >$$

$$< \text{application software 2}, \text{version} >$$

$$\ldots$$

It is important to understand that a software bundle may run distributed across many physical digital assets. If this is so, then there is a requirement associated with the construal of the system architecture:

unit zone: a zone in which a single software bundle runs on physical digital assets, or one physical digital asset with firmware but no additional software. Note that it is not necessary that a unit zone contain more than one physical digital unit. Unit zones contain elementary unit zones and atomic unit zones, as follows.

elementary unit zone: a single physical digital asset running software other than firmware, which software is specific to the unit and not distributed across multiple units.

atomic unit zone: a single physical digital asset running firmware but no additional software.

Along with this vocabulary come two requirements.

Requirement 1: if a software bundle runs on a collection C of digital physical assets, then this collection forms a unit zone of the architecture.

Requirement 2: a partition of an architecture into zones and conduits shall not split any unit zone across other zones. Note that this requirement is implicitly fulfilled by the definition of zone given in IEC 62443-1-1:2009, since a zone has uniform security requirements, and security requirements on software are usually requirements on a software bundle.

unit zone inventory specification: an unordered list of the digital unit inventory specifications of all digital units in the unit zone, along with the software bundle inventory specification of the software bundle running in the unit zone.

The digital units are connected to each other by means of network cables. In addition to the digital units, the system also contains these connections, and they need to be part of the system inventory also. The specification of a network connection includes the network protocol (the version of the protocol is implicit in a protocol name, and is different from the Version Number of any software or hardware which implements an end of the network in a digital unit).

network inventory specification: an unordered list of the digital units participating in the network communication, along with the network protocol name. It may be represented in the following form with the protocol appearing first:

$$< \text{protocol}, \text{unit1}, \text{unit2}, \ldots >$$

Given these concepts, we can now say what it takes to formulate an inventory of the system.

unit zone architecture: an architecture of the system indicating all unit zones as well as physical digital assets

system inventory: an unordered list of the unit zone inventory specifications of all the unit zones in the unit zone architecture of the system along with the network inventory specifications of networks connected to system assets.

15.2 Use of This Vocabulary, And Extensions

We have answered the first of two pertinent questions:

- What is a system inventory?

We have seen that an inventory contains version information of software. Some software is regularly updated in order to maintain cybersecurity, to implement modifications which eliminate vulnerabilities which have been discovered and might be exploited by unauthorised agents to violate the system integrity. Such updates must be installed from outside the system, for that is where they originate. There must therefore be a way that updates migrate from outside the system, where they originate, to the inside of the system. Such a pathway we can call an external channel, defined more precisely below. The necessary existence of at least one external channel raises the issue that, while required for diligent maintenance, they can also be exploited by malfeasants looking to violate the integrity of the system. This in turn raises the following question:

- What are the external connections to the system?

The importance of this question is to list the channels along which unauthorised interference with veridical system operation might be conducted. A malfeasant could replace a sensor with another device which does not report the sensed values veridically, or might interfere with the sensor so that it produces non-veridical values, but I consider this a separate issue from that of malfeasants and malware crossing the system boundary through channels from outside the system boundary. Any system has a boundary, which ideally should be precisely defined, as follows. The system consists of its physical digital assets. The system environment consists of objects not in the system with which the system interacts (engages in joint behaviour), or has the possibility to interact.

(system) environmental channel: a communication channel or physical port by

means of which the system and its environment may engage in joint behaviour. Note that an environmental channel may be a unit zone, part of a unit zone, elementary unit zone or part of one, or an atomic zone or part of one.

Environmental channels include

- USB and other physical ports, which may communicate with non-system digital devices which are attached to the port;
- Network protocols running over network cables which are attached to system digital units and environment digital units;
- Network protocols running on wireless networks whose content may be monitored and provided by environmental digital units.

Environmental channels form logical connections to the system environment. It is theoretically possible for an agent in the environment to alter system behaviour by providing input to the system over an environmental channel. This input may take different forms.

- Input could be in the form of values for system input parameters. System input parameters can be partitioned into command parameters and data.
- Input could be in the form of material which constitutes a software update to items in the system inventory. (Note "software" includes firmware.)

Such input may be authorised/intended/benign/necessary for appropriate continued system operation, or it also could be malign, intended to disrupt the function of the system, to violate the system functional integrity. The use of environmental channels must be controlled. To do that, you first have to know what all the channels are.

environmental channel inventory specification: a unit zone inventory specification of the unit zone in the system which constitutes the part of the system which interacts with the environment over the environmental channel

environmental channels inventory: a list of all environmental channel inventory specifications in the system inventory

Given each environmental channel in the inventory, the operator can specify what interactions may take place over the channel, and this will be part of the security policy for the unit zone in which the system takes part in channel behaviour. Ideally, this should exclude any interactions which result in loss of system integrity. I don't

pursue the question of specifying the permissible environmental- channel interactions further here. Thus the question concerning external connections divides into two:

- Is there an environmental channels inventory?
- Is there an interaction specification for each environmental channel in the environmental channel inventory?

15.3 Summary

We have derived three cybersecurity requirements connected with the inventory of IACS assets.

1. Maintain a system inventory.
2. Maintain an environmental channels inventory.
3. Devise an interaction specification for each environment channel in the environment channels inventory.

15.4 Comparison With Standards and Guidance

It surprised me to discover how much of inventory management is implicit in standards.

IEC 62443-1-1:2009 [5] says in subclause 5.6.2.1 that it is necessary to have an inventory of assets requiring protection. Subclause 6.5.4.1 says that an asset inventory is an attribute of a zone, and subclause 6.5.4.3 requires such an inventory (helpfully listing all asset types that should be listed). Subclause 6.5.6.1 says that an asset inventory is also an attribute of a conduit, and subclause 6.5.6.3 requires such an inventory for each conduit. However, the standard does not define what should be in an inventory entry.

NIST SP 800-80 Revision 2 (May 2015) [**?**] lists in Table C-4, Configuration and Maintenance Vulnerabilities and Predisposing Conditions, as first vulnerability: *Hardware, firmware and software not under configuration management*. In the commentary to this vulnerability, it elaborates that

> *An organisation may not know what it has, what versions it has, where they are, and what the patch status is, resulting in an inconsistent, and ineffective defense posture. A process for controlling modificiation to hardware, firmware, software and documentation should be implemented . . . A lack of configuration change management procedures can lead to . . . risks. To properly secure an ICS, there should be an accurate listing of the assets in the system and their current configurations . . .*

It is clear from this elaboration that knowing versions of assets, "configuration", is regarded as essential inventory information. But the guidance does not say explicitly what information constitutes a configuration.

ISA TR84.00.09-2017 [6] says in Section 5.1 that a "*complete inventory of cyber assets . . . needs to be developed and documented . . .* ". In Figure 3, showing the Cybersecurity Lifecycle Assessment Phase, *Initial System Architectural Diagrams and Inventory* is a documentational input, resulting in a documentational output *Updated System Architectural Diagrams and Inventory* with *IACS external services/support identified*. There is also a documentational output *Inventory Criticality Ranking/Initial Security Level (SL) Targets*, to which inputs are *Corporate Risk Criteria* and *Preliminary Hazard Review Info*, e.g., major hazards of the process, but, surprisingly, not a system inventory. It is hard to see how to obtain an inventory criticality ranking without inputting an inventory.

In Figure 5, *Cybersecurity Lifecycle Design and Implementation Phase*, there is documentational input, inter alia *Current Inventory*, to the *Cybersecurity Site Acceptance Test*, to which the output is *Updated SuC Inventory* ("*SuC*" means "*System under Consideration*"). This is reiterated in textual form in Section 11.2 on the *Acceptance Test*. There appears to be no task or documentational input/output associated with any Lifecycle Phase later than the Acceptance Test.

A system will have to be maintained with regard to cybersecurity, including patches, after it is taken into service; there appears to be no suggestion that the inventory also needs to be maintained (namely, updated along with the asset modifications undertaken in patching). Section 6, *Hazard and Risk Analysis*, says than an inventory of cyber assets is "*equivalent base line information*" for a cyber risk assessment, but it does not appear to say at what phase of the lifecycle such a cyber risk assessment should occur.

HSE operational guidance 86 on IACS Cybersecurity (march 2017) [1] is a guide for system assessors working for the HSE. It does not mention inventory as such. It does say that "*[p]rocedures and/or work instructions should be in place for the following*" and lists

- configuration management, for example recording and retention of firewall and encryption key distributions, rulesets and definitions;
- management of changes to the systems such as additions, changes to settings, record/maintain secure configuration checksums / versions etc. Relevant IACS change control records should be retained;

So it does require some sort of inventory of versions. It is implicit that there should be an inventory of initial versions.

Bibliography

[1] UK Health and Safety Executive, *Cyber Security for Industrial Automation and Control Systems (IACS)*, March 2017. Available from http://www.hse.gov.uk/foi/internalops/og/og-0086.pdf , accessed 2017-05-23.

[2] International Electrotechnical Commission, *IEC 61508, Functional safety of electrical/electronic/programmable electronic safety-related systems*, 2nd Edition, 7 parts, 2010.

[3] International Standards Organisation/International Electrotechnical Commission, *ISO/IEC Guide 51 Edition 3, Safety aspects – Guidelines for their inclusion in standards*, 2014.

[4] International Electrotechnical Commission, *IEC 61511, Functional safety – Safety instrumented systems for the process industry sector*, 2nd Edition, 3 parts plus AMD1:2017, 2016/2017.

[5] International Electrotechnical Commission, *IEC TS 62443-1-1:2009 Industrial communication networks - Network and system security - Part 1-1: Terminology, concepts and models*, 2009.

[6] The International Society of Automation, *ISA-TR84.00.09-2013 Cybersecurity Related to the Functional Safety Lifecycle*, 2nd Edition 2017.

[7] National Institute of Standards and Technology, Keith Stouffe,r Victoria Pillitteri, Suzanne Lightman, Marshall Abrams and Adam Hahn, *NIST Special Publication 800-82 Revision 2, Guide to Industrial Control Systems (ICS) Security*, May 2015. Available from https://csrc.nist.gov/publications/detail/sp/800-82/rev-2/final , accessed 2017-11-16.

[8] VDE Verband der Elektrotechnik Elektronik Informationstechnik e.V., *VDE-AR-E 2802-10-1: Zusammenhang zwischen funktionaler Sicherheit und Information-*

ssicherheit am Beispiel der Industrieautomation – Teil 1: Grundlagen; Relation between functional safety and IT security on the example of industrial automation – Part 1: Basic principles, 2017.