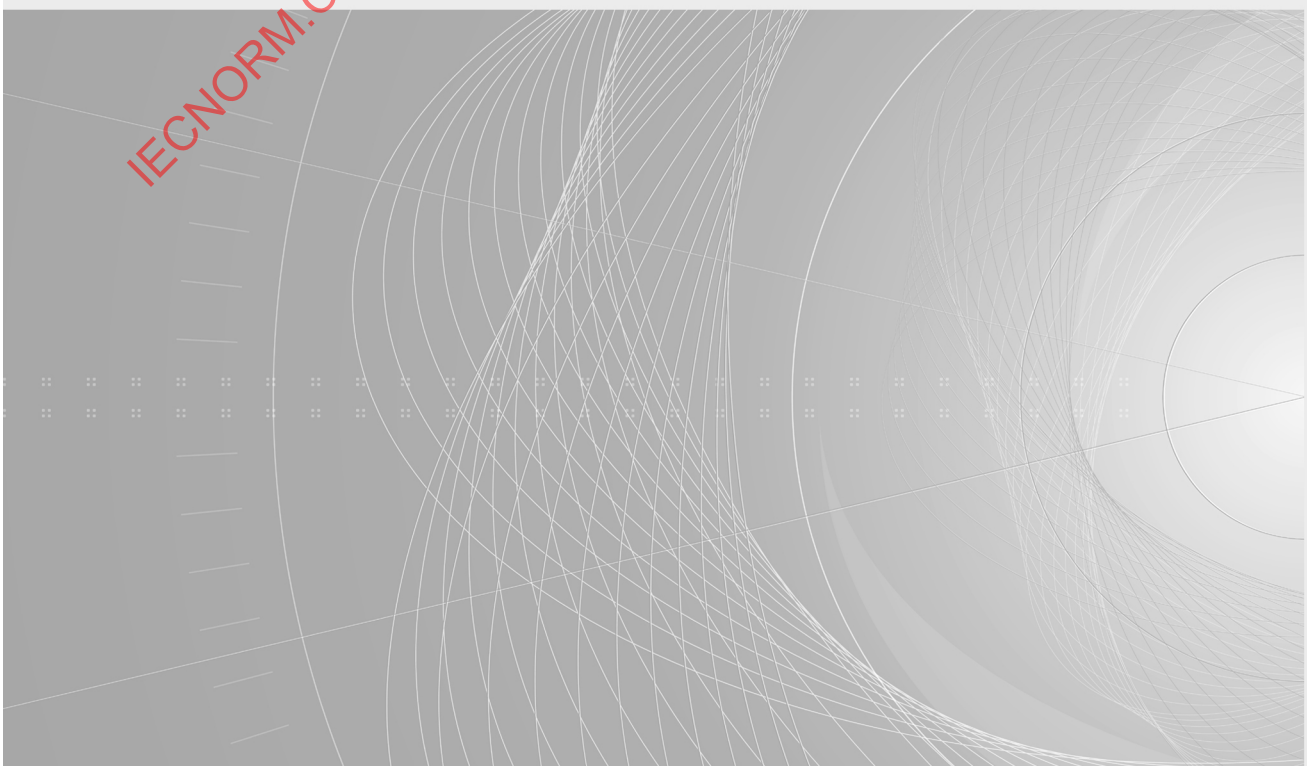
IEC TR 62443-2-3

Edition 1.0 2015-06

TECHNICAL REPORT

colour inside

Security for industrial automation and control systems – Part 2-3: Patch management in the IACS environment



IEC TR 62443-2-3:2015-06(en)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC
The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications
The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

IEC Catalogue - webstore.iec.ch/catalogue
The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub
The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,…). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished
Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and also once a month by email.

Electropedia - www.electropedia.org
The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in 15 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary
More than 60 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC Customer Service Centre - webstore.iec.ch/csc
If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

IEC TR 62443-2-3

Edition 1.0 2015-06

TECHNICAL REPORT

colour inside

Security for industrial automation and control systems – Part 2-3: Patch management in the IACS environment

INTERNATIONAL ELECTROTECHNICAL COMMISSION

ICS : 25.040.40; 35.040; 35.100

ISBN 978-2-8322-2768-8

Warning! Make sure that you obtained this publication from an authorized distributor.

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

SECURITY FOR INDUSTRIAL AUTOMATION AND CONTROL SYSTEMS –

Part 2-3: Patch management in the IACS environment

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

The main task of IEC technical committees is to prepare International Standards. However, a technical committee may propose the publication of a technical report when it has collected data of a different kind from that which is normally published as an International Standard, for example "state of the art".

Technical Report IEC 62443-2-3 has been prepared by ISA Technical Committee 99 in partnership with IEC technical committee 65: Industrial-process measurement, control and automation.

The text of this standard is based on the following documents:

Enquiry draft	Report on voting
65/554/DTR	65/564/RVC

Full information on the voting for the approval of this technical report can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts in the IEC 62443 series, published under the general title *Security for industrial automation and control systems*, can be found on the IEC website.

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC website under "http://webstore.iec.ch" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

A bilingual version of this publication may be issued at a later date.

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

Cyber security is an increasingly important topic in modern organizations. Many organizations involved in information technology (IT) and business have been concerned with cyber security for many years and have well-established information security management systems (ISMS) in place as defined by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC), in ISO/IEC 27001 and ISO/IEC 27002. These management systems provide an organization with a well-established method for protecting its assets from cyber-attacks.

Industrial Automation and Control Systems (IACS) suppliers and owners are using commercial-off-the-shelf (COTS) technology developed for business systems in their everyday processes. This provides an increased opportunity for cyber-attack against the IACS equipment, since COTS systems are more widely known and used. There has also been new interest in ICS security research that has uncovered numerous device vulnerabilities as well. Successful attacks against industrial systems may lead to health, safety and environmental (HSE) consequences.

Organizations may try to use the business cyber security strategy to address security for IACS without understanding the consequences. While many of these solutions can be applied to IACS, they need to be applied in the correct way to eliminate inadvertent consequences.

This technical report addresses the patch management aspect of IACS cyber security. Patch management is part of a comprehensive cyber security strategy that increases cyber security through the installation of patches, also called software updates, software upgrades, firmware upgrades, service packs, hotfixes, basic input output system (BIOS) updates and other digital electronic program updates that resolve bugs, operability, reliability and cyber security vulnerabilities. This technical report introduces to the reader many of the problems and industry concerns associated with IACS patch management for asset owners and IACS product suppliers. It also describes the impacts poor patch management can have on the reliability and/or operability of the IACS.

SECURITY FOR INDUSTRIAL AUTOMATION AND CONTROL SYSTEMS –

Part 2-3: Patch management in the IACS environment

1 Scope

This part of IEC 62443, which is a Technical Report, describes requirements for asset owners and industrial automation and control system (IACS) product suppliers that have established and are now maintaining an IACS patch management program.

This Technical Report recommends a defined format for the distribution of information about security patches from asset owners to IACS product suppliers, a definition of some of the activities associated with the development of the patch information by IACS product suppliers and deployment and installation of the patches by asset owners. The exchange format and activities are defined for use in security related patches; however, it may also be applicable for non-security related patches or updates.

The Technical Report does not differentiate between patches made available for the operating systems (OSs), applications or devices. It does not differentiate between the product suppliers that supply the infrastructure components or the IACS applications; it provides guidance for all patches applicable to the IACS. Additionally, the type of patch can be for the resolution of bugs, reliability issues, operability issues or security vulnerabilities.

NOTE 1 This Technical Report does not provide guidance on the ethics and approaches for the discovery and disclosure of security vulnerabilities affecting IACS. This is a general issue outside the scope of this report.

NOTE 2 This Technical Report does not provide guidance on the mitigation of vulnerabilities in the period between when the vulnerability is discovered and the date that the patch resolving the vulnerability is created. For guidance on multiple countermeasures to mitigate security risks as part of an IACS security management system (IACS-SMS), refer to, Annexes B.4.5, B.4.6 and B.8.5 in this Technical Report and other documents in the IEC 62443 series.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC TS 62443-1-1, *Industrial communication networks – Network and system security – Part 1-1: Terminology, concepts and models*

IEC 62443-2-1, *Industrial communication networks – Network and system security – Part 2-1: Establishing an industrial automation and control system security program*

3 Terms, definitions, abbreviated terms and acronyms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in the normative references specified in Clause 2, as well as the following, apply:

3.1.1
bug
flaw in the original development of software (such as a security vulnerability), which causes it to perform or behave in an unintended manner (such as cause reliability or operability issues)

3.1.2
patch
incremental software change in order to address a security vulnerability, a bug, reliability or operability issue (update) or add a new feature (upgrade)

Note 1 to entry: Patches may also be called software updates, software upgrades, firmware upgrades, service packs, hotfixes, basic input output system (BIOS) updates, security advisories and other digital electronic program updates.

3.1.3
patch lifecycle
period in time that a patch is recommended or created until the patch is installed

Note 1 to entry: In the context of this technical report, this lifecycle begins when the patch is created and made available.

Note 2 to entry: Some feel that the patching lifecycle begins when the vulnerability has been disclosed. However, it is not possible for this technical report to provide all possible guidance for the mitigation of vulnerabilities for the period between disclosure of a vulnerability, the decision to create a patch and the availability of a patch. It is also to the discretion of the software developer or product supplier to determine if they develop a patch.

3.1.4
patch management
set of processes used to monitor patch releases, decide which patches should be installed to which system under consideration (SuC), if the patch should be tested prior to installation on a production SuC, at which specified time the patch should be installed and of tracking the successful installation

3.2 Abbreviated terms and acronyms

ANSI	American National Standards Institute
BCP	Business continuity planning
BIA	Business impact assessment
BIOS	Basic input output system
CCTS	Core Components Technical Specification
CERT	Cyber Emergency Response Team, Computer Emergency Readiness Team or other regional/industry variant
CD	Compact disc
COTS	Commercial-off-the-shelf
CPNI	[UK] Centre for Protection of National Infrastructure
CPU	Central processing unit
DCS	Distributed control system
DHS	[US] Department of Homeland Security
DRP	Disaster recovery planning
DVD	Digital versatile disc
EULA	End user license agreement
FAT	Factory acceptance testing
HSE	Health, safety and environmental
HTML	Hypertext Markup Language
HTTP	Hypertext transfer protocol
ICS-CERT	[US DHS] Industrial Control Systems Cyber Emergency Response Team

IACS	Industrial automation and control system(s)
IACS-SMS	IACS security management system
IDS	Intrusion detection system
IEC	International Electro-technical Commission
IP	Internet protocol
IPS	Intrusion prevention system
ISA	International Society of Automation
ISMS	Information security management system
ISO	International Organization for Standardization
IT	Information technology
KPI	Key performance indicator
MD5	Message digest 5
MES	Manufacturing execution system
MESA	Manufacturing Enterprise Solutions Association International
MSMUG	Microsoft Manufacturing Users Group
NERC	North American Electric Reliability Corporation
NISCC	[US] National Infrastructure Security Co-ordination Centre
NSA	[US] National Security Agency
OAGIS	Open Applications Group Integration Specification
OEM	Original equipment manufacturer
OS	Operating system
PLC	Programmable logic controller
RACI	Responsible, accountable, consulted, informed
RAID	Redundant array of independent disks
RASCI	Responsible, accountable, supportive, consulted and informed
RTU	Remote terminal unit
SAT	Site acceptance testing
SHA	Secure hash algorithm
SIS	Safety instrumented system
SMTP	Simple Mail Transfer Protocol
SPX	Sequenced packet exchange
SQL	Structured query language
SuC	System under consideration
TC	Technical committee
UN	United Nations
UN/CEFACT	United Nations Centre for Trade Facilitation and Electronic Business
URI	Uniform resource identifier
USB	Universal serial bus
US-CERT	United States Computer Emergency Readiness Team
VPC	Vendor patch compatibility
WAN	Wide area network
XML	eXtensible Markup Language
XSD	XML schema definition

4 Industrial automation and control system patching

4.1 Patching problems faced in industrial automation and control systems

There are many challenges that asset owners face when attempting to implement a patch management program for their IACS. Patching an IACS means changing the IACS and changes can negatively affect its safety, operability or reliability if not performed correctly. Preparing an IACS to be patched can require a tremendous amount of work and asset owners may struggle for the necessary resources to address the added workload. For each patch and for each product they own, an asset owner will have to gather and analyze patch information for each device, install and verify on a test system, ensure backups are created before and after, ensure testing again before turning the system back over to operations and finally track all the necessary documentation of the changes.

Due to the resources and efforts recommended to patch an IACS most organizations schedule patch installations during other normal routine maintenance outages. Sometimes these outage windows are quarterly, yearly or even less frequently. Some extremely critical systems may not have outage windows available and can therefore not be patched if a system outage is required to do so.

Applying patches is a risk management decision. If the cost of applying patches is greater than the risk evaluated cost, then the patch may be delayed, especially if there are other security controls in place that mitigate the risk (such as disable or remove features).

The unintended consequences of a poor patch management program can include:

- incompatibility between patches and control system software;
- false positives due to antivirus and anti-malware; and
- degradation of system performance, reliability and operability with insufficient testing.

For additional information, see B.4.2.

4.2 Impacts of poor patch management

Adversaries (for example, malicious threat actors) will always have an advantage over their targets given the challenges product suppliers and asset owners face in keeping their systems up to date to minimize security risk caused by vulnerabilities. The moment a vulnerability is disclosed, whether by well-intentioned or malicious intent, the problem is then transferred primarily to the asset owner to apply the patch as quickly as possible. The asset owner may or may not be able to apply the patch and it becomes a risk-based decision on how to mitigate the vulnerability risk. Though it may never be possible to eliminate all software vulnerabilities, there should be no excuse for not evaluating the risk of the vulnerability and determining when and how patches should be applied.

The primary impact of poor IACS patch management is an increased risk of loss or compromise of an IACS system. Unlike for example office or enterprise systems, compromise of an IACS may have consequences beyond the loss of data or downtime of the system. A compromise of an IACS may impact system safety, the physical safety of operational personnel, the quality of produced products, the safety of produced products and the usability of produced products.

For additional information, see B.4.2.

NOTE 1 If critical documentation on the production of a product is lost, the product may have to be scrapped, even if there was no physical damage done to the product (such as pharmaceutical development, food production, etc.)

NOTE 2 Directed attacks of unpatched IACS systems may even result in the destruction of equipment. Undirected attacks of unpatched IACS systems, where the IACS system is not a primary target, may still cause the loss of control with resultant risks to safety and product quality. One example of such an attack is Structured Query Language (SQL) injection worms, which consume all central processing unit (CPU) and network resources.

4.3 Obsolete IACS patch management mitigation

Asset owners may experience the situation where products are no longer supported by their suppliers but have reported vulnerabilities. IACS systems are typically in production for decades and adversaries know these older systems are vulnerable. Asset owners need to consider other mitigations when patching is not an option.

For additional information on countermeasures to mitigate security risks as part of an IACS patch management process see Annexes B.4.5, B.4.6 and B.5.5, and other documents in the IEC 62443 series.

4.4 Patch lifecycle state

Patches have a defined lifecycle state model. They progress from available to authorized to effective and installed. Not all patches available are relevant to the IACS and not all patches are compatible with the IACS applications. It is important for an effective IACS patch management process to know the state of all available patches. Lifecycle states for patches are defined in Table 1.

Table 1 – Patch lifecycle states

Patch state	Patch state definition	Managed by
Available	The patch has been provided by a third party or an IACS supplier but has not been tested.	Asset owner Product supplier
In Test	The patch is being tested by an IACS supplier.	Product supplier
Not Approved	The patch has failed the testing of the IACS supplier and should not be used, unless and until the IACS supplier confirms that the patch has been Approved.	Product supplier
Not Applicable	The patch has been tested and is not considered relevant to IACS use.	Product supplier
Approved	The patch has passed testing by the IACS supplier.	Product supplier
Released	The patch is released for use by the IACS supplier or third party, or the patch may be directly applicable by the asset owner for their internally developed systems.	Asset owner Product supplier
In Internal Test	The patch is being tested by the asset owner testing team.	Asset owner
Not Authorized	The patch has failed internal testing, or may not be applicable.	Asset owner
Authorized	The patch is released by the asset owner and meets company standards for updatable devices, or by inspection did not need testing.	Asset owner
Effective	The patch is posted by the asset owner for use.	Asset owner
Installed	The patch is installed on the system.	Asset owner

The state model for lifecycle states is shown in Figure 1. The states maintained by the IACS product supplier are in the dark gray area in the left half of the figure. The states maintained by the IACS asset owner are in the light grey area in the right half of the figure. The transitions between states are activities of the asset owners or the product suppliers, as defined in the other parts of this report.

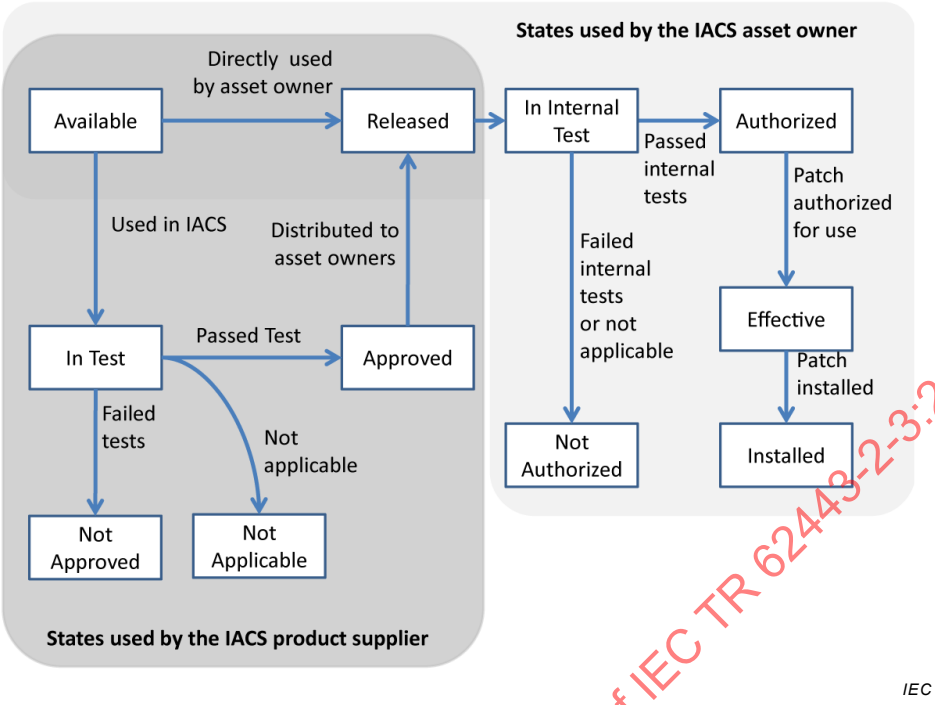


Figure 1 – Patch state model

5 Recommended requirements for asset owner

Asset owners have an implied obligation to uphold the safety, reliability, operability, security and quality of their operations. Achieving cyber security assurance, through patching IACS assets, is a critical part of that obligation.

IACS asset owners should:

a) establish and maintain an inventory of all electronic devices associated with the IACS, that may be updated by: modification of their functionality, configuration, operation, software, firmware, operating code, etc. These devices should be referred to as ‘updatable’ devices;

b) establish and maintain an accurate record of the currently installed versions for each device, called the ‘installed’ version;

c) determine on a regular schedule what upgrades and updates are available for each device, called the ‘latest’ version;

d) determine on a regular schedule the ‘released versions’ of upgrades and updates which are identified as compatible by the IACS product supplier and meet the asset owners standards for ‘updatable’ devices;

e) test the installation of IACS patches in a way that accurately reflects the production environment, so as to ensure that the reliability and operability of the IACS is not negatively affected when patches are installed on the IACS in the actual production environment. Patches which have successfully passed these tests are called the ‘authorized patches’;

f) schedule authorized, effective patches for installation at the next available opportunity within the constraints of system design (for example, redundancy, fault-tolerance, safety) and operational requirements (for example, unplanned outage, scheduled outage, on-process, etc.);

g) update records at a planned interval, at least on a quarterly basis, to include for each updateable device: installed versions, authorized versions, effective versions and released versions;

h) identify a planned interval for installation of patches, such as: when patches are available, or at least on an annual basis; and

i) install patches and/or implement compensating countermeasures to mitigate security vulnerabilities that exist in the IACS.

Additional guidance that can be used to achieve these requirements is provided in Annex B.

6 Recommended requirements for IACS product supplier

Security of the IACS on a running facility is very important and proactive measures are needed to reduce the probability of the plant being compromised, therefore determining which patches apply to, and should be tested on, a product is a critical responsibility of the IACS product supplier.

IACS product suppliers should:

a) provide documentation describing the software patching policy for the products and systems they supply;

b) qualify in terms of applicability and compatibility, all patches, by analyzing and verifying the patches, including patches that are released by the supplier of the OS that is used, and all suppliers of third-party software, that may be used by the IACS products;

c) provide a list of all patches and their approval status, 4.4, including the information and data in the format described in Clause 7 and Annex A;

d) inform the asset owners, and update the list of patches described in: clause 6. b) above, Clause 7, and Annex A periodically, and ideally within 30 days after a patch is released by the supplier of the OS or third-party software;

e) provide adequate warning (at least two years in advance) about the components reaching 'end of life,' or for which cyber security patches will no longer be made available; and

f) provide information to IACS users regarding the policy of supporting IACS products, including security updates.

Additional guidance that can be used to achieve these requirements is provided in Annex C.

7 Exchanging patch information

7.1 General

Patch information is required because a complete IACS is usually based on commercial OSs, commercial application systems, such as distributed control systems (DCSs), historian and manufacturing execution systems (MESs) and application specific software programs using commercial IT tools, such as databases and libraries. All of these software elements require periodic updates to correct newly discovered errors or to correct newly discovered security deficiencies.

Implementing a system to manage patches requires knowledge of: what patches are available, if the patches are applicable to installed systems, if the patches have been tested against the installed products, and if the IACS product supplier recommends that the patches should be installed.

Determining the compatibility of patches can be a complex task. IACS product suppliers perform tests of their products against OS and library patches in order to determine if the patch should be used with their automation products. Because failures in automation products due to incompatibility with patches may result in the loss of life, property or product, there is often a requirement that all related automation products have been tested with the patch prior to installation of the patch in a production system.

IACS users often have various IACS product supplier systems in their facilities and managing the patch compatibility information from multiple IACS product suppliers is difficult because the patch information is usually available in each IACS product supplier's specific format. This clause defines a standard format for the exchange of patch information necessary to identify a product, patch and status of the patch. The exchanged information includes:

a) an identification of the IACS product supplier providing the product;

b) an identification of the IACS supplier's product and version;

c) an identification of the product supplier providing the patch, such as the company providing the OS;

d) an identification of the patch supplier's product, such as the OS version;

e) the patch supplier's identification of the patch;

f) an indication if the patch is applicable to the IACS product;

g) an indication of status of testing of the patch against the IACS product; and

h) an indication of the results of testing of the patch against the IACS product.

This information allows end users to make informed decisions before they decide to install the patch. The exchange information defines if a specific patch from an IACS system supplier, an OS supplier or a third party software product supplier has been tested against a specific version of the IACS software, with an indication that the tested patch works with that version of the IACS software.

7.2 Patch information exchange format

The format for the minimal patch compatibility information is based on eXtensible Markup Language (XML) technology and is defined using an XML schema definition (XSD) file. The patch information file is identified as vendor patch compatibility (VPC).

7.3 Patch compatibility information filename convention

The filename of a VPC file should be defined according to the following syntax:

```
<filename> = <vendor_name> "_patch_compatibility_" <date> "_" <number> ".xml"
```

where

`<vendor_name>` the generally recognized short name of the IACS company

`<date>` the date the compatibility file was released by the IACS product supplier (formatted according to IS0 8601 [8])

`<number>` a number identifying the file if the IACS product supplier releases more than one file on a single date.

EXAMPLE 1 SomeCompany_patch_compatibility_2010-01-08_01.xml

EXAMPLE 2 OtherCompany_patch_compatibility_2010-01-08_02.xml

NOTE Since the recipient will use the contents of this file to manage whether a patch is to be applied, there is a need to ensure that its contents are authentic. The means by which source authentication and integrity protection of this file is attained is outside the scope of this Technical Report."

7.4 VPC file schema

The VPC format allows for the exchange of patch information about multiple patches and IACS product supplier products in the same VPC exchange file.

Figure 2 illustrates the VPC file schema definition.

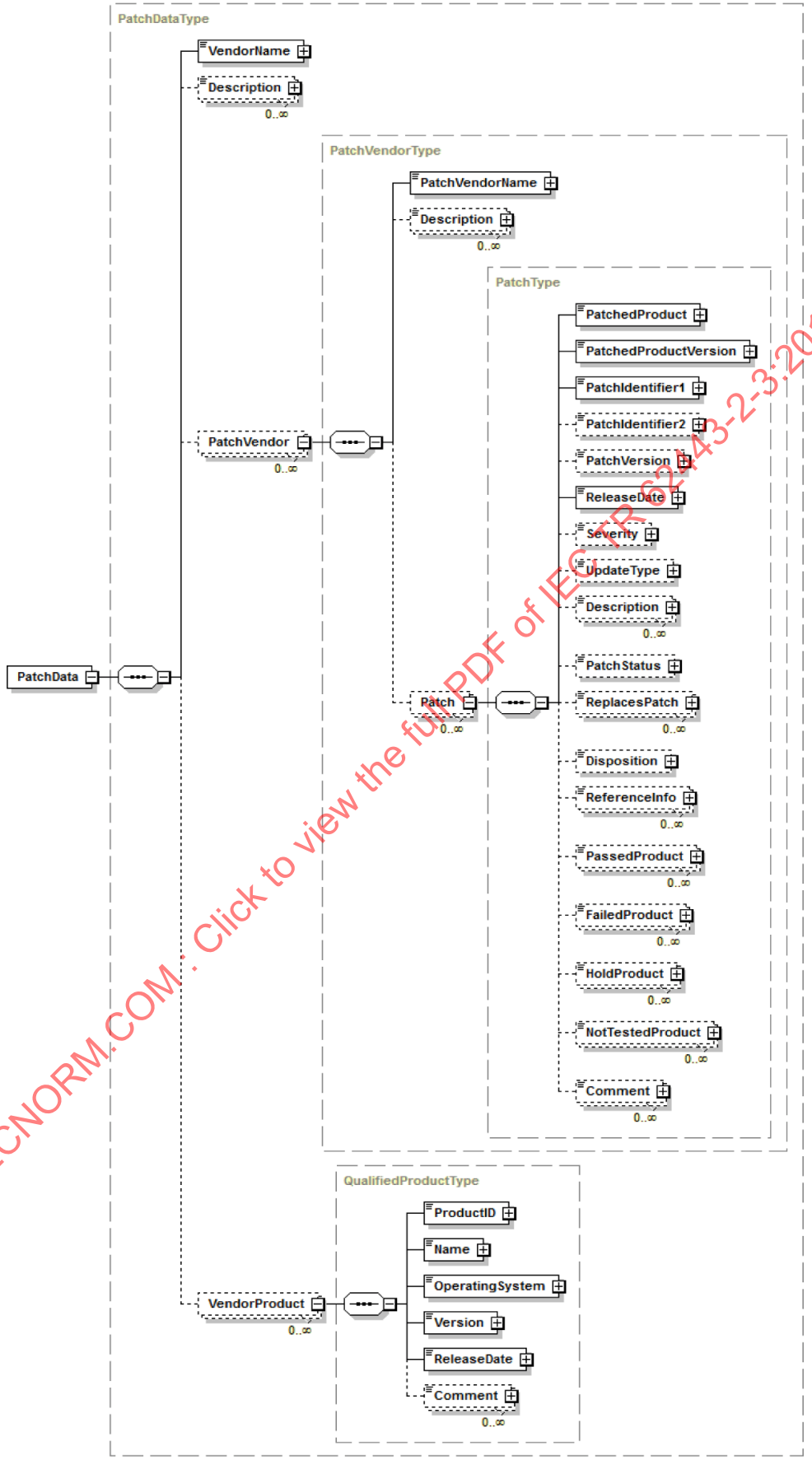


Figure 2 – VPC file schema

Figure 3 illustrates the VPC file schema diagram format.

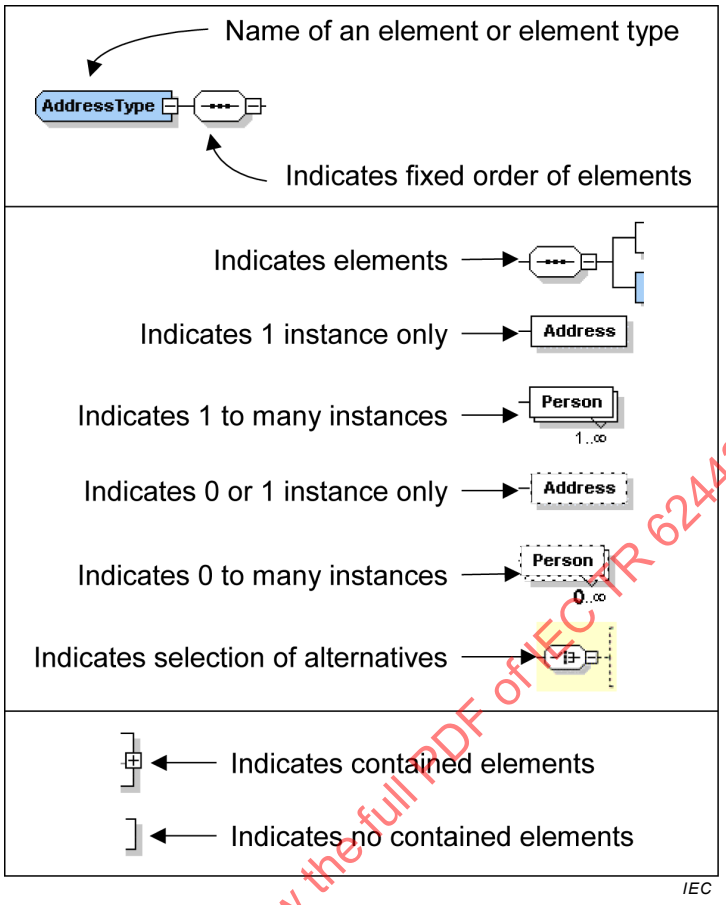


Figure 3 – VPC file schema diagram format

7.5 VPC file element definitions

Table 2 through Table 5 define each element in the VPC XSD file. Only the enumerations, defined in the "Definition" column, for the applicability, test result and test status should be used in the exchange file.

Table 2 – VPC XSD PatchData file elements

Element	Type	Definition
VendorName	IdentifierType	A required string containing the name of the vendor that is providing the patch information. EXAMPLES: "DCS vendor", "MES vendor"
Description	TextType	An optional string containing a description of the vendor information.

Table 3 – VPC XSD PatchVendor file elements

Element	Type	Definition
PatchVendorName	IdentifierType	A required string containing the name of the vendor that is providing the patch. EXAMPLES: “OS Vendor”, “DCS Vendor”, “Protocol Stack Vendor”
Description	TextType	An optional string containing a description of the vendor.

Table 4 – VPC XSD Patch file elements

Element	Type	Definition
PatchedProduct	IdentifierType	A required string containing the patch product supplier's name of the product that the patch is targeting. EXAMPLES: "SQL Server", “Microsoft Windows”
PatchedProductVersion	IdentifierType	A required string containing the version of the product the patch is for. EXAMPLES: "Service Pack 2", "7.15", "A", "R23.9"
PatchIdentifier1	IdentifierType	A required string containing the patch product supplier defined primary identification of the patch. EXAMPLE: “KB1234567”
PatchIdentifier2	IdentifierType	An optional string containing a patch product supplier defined secondary identification of the patch.
PatchVersion	IdentifierType	An optional string containing the version number of the patch. EXAMPLES: "1", "1.0", "1.2" NOTE This may be needed if multiple versions of the patch are released due to errors in the previous patch version.
ReleaseDate	DateTimeType	A required string containing the released date of the patch. EXAMPLE: "2014-01-17" NOTE Format this string according to ISO 8601.
Severity	CodeType	An optional string containing the severity of the patch. The value should be one of the items in the following enumeration: • **Critical –** The patch should be installed. It corrects a vulnerability whose exploitation could allow code execution without user interaction. These scenarios include for example: self-propagating malware (such as network worms) and unavoidable common use scenarios where code execution occurs without warnings or prompts. • **Important –** The patch should be installed. The patch corrects a vulnerability whose exploitation could result in compromise of the confidentiality, integrity or availability of data, or of the integrity or availability of processing resources, but which requires a user action. • **Optional –** The patch may be installed. The patch corrects a vulnerability that requires unique or uncommon user actions.

Element	Type	Definition
UpdateType	CodeType	A required string value containing the type of patch. The value should be one of the items in the following enumeration: • **Non_Security –** This update is related to a non-security related issue and updates a known issue that is not related to security. • **Security –** This update is related to a security issue and repairs a known security problem.
Description	TextType	An optional string value that contains a description of the update and/or patch.
PatchStatus	CodeType	A required string value containing the patch status. The value should be one of the items in the following enumeration: • **Deprecated –** This patch is no longer a required update for the system. • **Current –** This patch is up to date and should be installed on any products identified as "PassedProduct" for this patch.
ReplacesPatch	TextType	An optional string value that contains a description of the patch being replaced. EXAMPLE: "KB1234567 Windows 2008 64 bit"
Disposition	CodeType	A required string value containing the patch status. The value should be one of the items in the following enumeration: • **Primary –** This patch is a primary patch. Other patches are dependent on this patch. • **Dependent –** This patch is dependent on a primary patch and/or other patches. • **Standalone –** This is a standalone patch. It does not depend on any primary patch and no patches depend on this standalone patch.
ReferenceInfo	TextType	An optional string value that contains a URL to the update for further details.
PassedProduct	IdentifierType	An optional string value containing the <VendorProduct><ProductID> for referencing the vendor specific product. This is an item in the list of products that PASSED testing with this patch.
FailedProduct	IdentifierType	An optional string value containing the <VendorProduct><ProductID> for referencing the vendor specific product. This is an item in the list of products that FAILED testing with this patch.
HoldProduct	IdentifierType	An optional string value containing the <VendorProduct><ProductID> for referencing the vendor specific product. This is an item in the list of products that are on HOLD for testing with this patch.
NotTestedProduct	IdentifierType	An optional string value containing the <VendorProduct><ProductID> for referencing the vendor specific product. This is an item in the list of products NOT TESTED for this patch.
Comment	TextType	An optional string that contains a comment on the patch.

Table 5 – VPC XSD VendorProduct file elements

Element	Type	Definition
ProductID	IdentifierType	A required string identifier that will be used to map the name and version of a product to a test result related to a patch.
Name	TextType	A required string containing the name of the product.
OperatingSystem	TextType	A required string containing the operating systems description for the product.
Version	TextType	A required string containing the version of the product.
ReleaseDate	DateTimeType	A required string containing the released date of the product version.
Comments	TextType	An optional string that contains a comment on the product and version information.

Annex A
(informative)

VPC XSD file format

A.1 VPC XSD file format specification

The following is a code listing of the XSD file containing the XML schema validation rules.

```
<?xml version="1.0" encoding="utf-8"?>
<!-- product supplier patch-information -->
<!-- Schema for product supplier patch-compatibility -->
<xs:schema id                       = "patch-compatibility"
           xmlns                    = "http://www.isa.org/xml/VendorPatchCompatibility"
           targetNamespace          = "http://www.isa.org/xml/VendorPatchCompatibility"
           xmlns:mstns              = "VendorPatchCompatibility.xsd"
           xmlns:xs                 = "http://www.w3.org/2001/XMLSchema"
           xmlns:msdata             = "urn:schemas-microsoft-com:xml-msdata"
           attributeFormDefault     = "qualified"
           elementFormDefault       = "unqualified">
   <xs:annotation>
      <xs:documentation>
         ISA62443 Product Supplier Patch Compatibility
         Copyright 2014 ISA, All Rights Reserved. http://www.isa.org

         This ISA work (including specifications, documents, software, and related items)
         referred to as the ISA Product Supplier Patch Compatibility markup language is
         provided by the copyright holders under the following license.

         Permission to use, copy, modify, or redistribute this Work and its
         documentation, with or without modification, for any purpose and
         without fee or royalty is hereby granted, provided ISA is acknowledged
         as the originator of this work using the following statement:

         The Product Supplier Patch Compatibility Markup Language is used courtesy of ISA.

         In no event shall the ISA, its members, or any third-party be liable for
         any costs, expenses, losses, damages or injuries incurred by use of this
         ISA work or as a result of this agreement.
      </xs:documentation>
   </xs:annotation>
   <!--  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -   -->
   <!--  Top Level PatchData element - - - - - - - - - - - - - - -   -->
   <!--  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -   -->
   <xs:element name="PatchData" type="PatchDataType" />

   <!--  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -   -->
   <!--  Core Component Types    - - - - - - - - - - - - - - - - -   -->
   <!--  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -   -->

   <!--  ************************************    -->
   <!--  CodeType used for any enumerated strings    -->
   <!--  ************************************    -->
   <xs:complexType name="CodeType">
      <xs:simpleContent>
         <xs:extension base="xs:normalizedString">
            <xs:attribute name="listID"          type="xs:normalizedString" use="optional" />
            <xs:attribute name="listAgencyID"    type="xs:normalizedString" use="optional" />
            <xs:attribute name="listAgencyName"  type="xs:string"           use="optional" />
            <xs:attribute name="listName"        type="xs:string"           use="optional" />
            <xs:attribute name="listVersionID"   type="xs:normalizedString" use="optional" />
            <xs:attribute name="name"            type="xs:string"           use="optional" />
            <xs:attribute name="languageID"      type="xs:language"         use="optional" />
            <xs:attribute name="listURI"         type="xs:anyURI"           use="optional" />
            <xs:attribute name="listSchemeURI"   type="xs:anyURI"           use="optional" />
         </xs:extension>
      </xs:simpleContent>
   </xs:complexType>
   <!--  ********************************************************    -->
   <!--  DateTimeType used for any date and/or time representations    -->
   <!--  ********************************************************    -->
   <xs:complexType name = "DateTimeType">
      <xs:simpleContent>
         <xs:extension base ="xs:dateTime">
```

```
        <xs:attribute name="format"           type="xs:string"           use="optional" />
      </xs:extension>
   </xs:simpleContent>
</xs:complexType>
<!--  ************************************************************    -->
<!--  IdentifierType used for any string used to identify an element    -->
<!--  ************************************************************    -->
<xs:complexType name="IdentifierType">
   <xs:simpleContent>
      <xs:extension base="xs:normalizedString">
         <xs:attribute name="schemeID"         type="xs:normalizedString" use="optional" />
         <xs:attribute name="schemeName"       type="xs:string"           use="optional" />
         <xs:attribute name="schemeAgencyID"   type="xs:normalizedString" use="optional" />
         <xs:attribute name="schemeAgencyName" type="xs:string"           use="optional" />
         <xs:attribute name="schemeVersionID"  type="xs:normalizedString" use="optional" />
         <xs:attribute name="schemeDataURI"    type="xs:anyURI"           use="optional" />
         <xs:attribute name="schemeURI"        type="xs:anyURI"           use="optional" />
      </xs:extension>
   </xs:simpleContent>
</xs:complexType>
<!--  ********************************************************    -->
<!--  TextType used for any element that requires a string value    -->
<!--  ********************************************************    -->
<xs:complexType name="TextType">
   <xs:simpleContent>
      <xs:extension base="xs:string">
         <xs:attribute name="languageID"       type="xs:language"         use="optional" />
      </xs:extension>
   </xs:simpleContent>
</xs:complexType>
 <!--     -->
<!--  ************************************************************    -->
<!--  PatchDataType used to contain patch and vendor information      -->
<!--  ************************************************************    -->
 <xs:complexType name="PatchDataType">
   <xs:sequence>
      <xs:element name="VendorName"    type="IdentifierType" />
      <xs:element name="Description"   type="TextType" minOccurs="0" maxOccurs="unbounded"/
                  >
      <xs:element name="PatchVendor"   type="PatchVendorType" minOccurs="0" maxOccurs="unbo
                  unded"/>
      <xs:element name="VendorProduct" type="QualifiedProductType" minOccurs="0" maxOccurs=
                  "unbounded"/>
   </xs:sequence>
 </xs:complexType>
<!--  ************************************************************    -->
<!--  PatchVendorType used to contain patch compatibility information    -->
<!--  ************************************************************    -->
 <xs:complexType name="PatchVendorType">
     <xs:sequence>
       <xs:element name="PatchVendorName" type="IdentifierType" />
       <xs:element name="Description"     type="TextType"  minOccurs="0" maxOccurs="unbound
                   ed"/>
       <xs:element name="Patch"           type="PatchType" minOccurs="0" maxOccurs="unbound
                   ed"/>
     </xs:sequence>
 </xs:complexType>
<!--  ************************************************************    -->
<!--  PatchType used to contain information on a specific patch       -->
<!--  ************************************************************    -->
 <xs:complexType name="PatchType">
     <xs:sequence>
       <xs:element name="PatchedProduct"        type="IdentifierType" />
       <xs:element name="PatchedProductVersion" type="IdentifierType" />
       <xs:element name="PatchIdentifier1"      type="IdentifierType" />
       <xs:element name="PatchIdentifier2"      type="IdentifierType" minOccurs="0" />
       <xs:element name="PatchVersion"          type="IdentifierType" minOccurs="0" />
       <xs:element name="ReleaseDate"           type="DateTimeType" />
       <xs:element name="Severity"              type="CodeType"       minOccurs="0" />
       <xs:element name="UpdateType"            type="CodeType"       minOccurs="0" />
       <xs:element name="Description"           type="TextType"       minOccurs="0" maxOccu
                   rs="unbounded" />
       <xs:element name="PatchStatus"           type="CodeType"       minOccurs="0" />
       <xs:element name="ReplacesPatch"         type="TextType"       minOccurs="0" maxOccu
                   rs="unbounded" />
       <xs:element name="Disposition"           type="CodeType"       minOccurs="0" />
       <xs:element name="ReferenceInfo"         type="TextType"       minOccurs="0" maxOccu
                   rs="unbounded" />
```

```
          <!--
List of Product IDs referencing VendorQualifiedProductDetails for the PassedProductList -->
          <xs:element name="PassedProduct"         type="IdentifierType"  minOccurs="0" maxOcc
                      urs="unbounded" />
          <!--
List of Product IDs referencing VendorQualifiedProductDetails for the FailedProductList -->
          <xs:element name="FailedProduct"         type="IdentifierType"  minOccurs="0" maxOcc
                      urs="unbounded" />
          <!--
List of Product IDs referencing VendorQualifiedProductDetails for the HoldProductList -->
          <xs:element name="HoldProduct"           type="IdentifierType"  minOccurs="0" maxOcc
                      urs="unbounded" />
          <!--
List of Product IDs referencing VendorQualifiedProductDetails for the NotTestedProductList -->
          <xs:element name="NotTestedProduct"      type="IdentifierType"  minOccurs="0" maxOcc
                      urs="unbounded" />
          <xs:element name="Comment"               type="TextType"        minOccurs="0" maxOcc
                      urs="unbounded" />
        </xs:sequence>
    </xs:complexType>
   <!--  ************************************************************    -->
   <!--  PatchType used to contain information on a specific patch       -->
   <!--  ************************************************************    -->
    <xs:complexType name="QualifiedProductType">
        <xs:sequence>
          <xs:element name="ProductID"             type="IdentifierType" />
          <xs:element name="Name"                  type="TextType" />
          <xs:element name="OperatingSystem"       type="TextType" />
          <xs:element name="Version"               type="TextType" />
          <xs:element name="ReleaseDate"           type="DateTimeType" />
          <xs:element name="Comment"               type="TextType" minOccurs="0" maxOccurs="un
                      bounded" />
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

A.2 Core component types

A.2.1 Overview

The base types for most elements are derived from core component types that are compatible with the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) core component types. The UN/CEFACT core component types are a common set of types that define specific terms with semantic meaning (for example, the meaning of a quantity, currency, amount and identifier). The UN/CEFACT core components were defined in a Core Components Technical Specification (CCTS) developed by the ebXML project now organized by UN/CEFACT and ISO technical committee (TC) 154.

NOTE The core components contain optional attributes that may be used to specify the context and source of the associated element value. All attributes are optional in the VPC schema.

The core components use several international standards for the representation of semantic and standardized information:

a) language code as specified in ISO 639-1:2002 [4][1]; and

b) date and time representation as specified in ISO 8601:2004 [8].

A.2.2 CodeType

CodeType is used to define a character string that is used to represent an entry from a fixed enumeration. It is derived from the type normalizedString. All of the VPC enumerations are derived from CodeType. Table A.1 describes the optional attributes for the CodeType data type.

[1] Numbers in square brackets refer to the bibliography.

Table A.1 – CodeType optional attributes

Optional Attribute	Base XML Type	Description
listID	normalizedString	An identifier specifying a code list that this is registered with an agency. EXAMPLE: UN/EDIFACT data element 3055 code list
listAgencyID	normalizedString	An identifier specifying the agency that maintains one or more lists of codes. EXAMPLE: UN/EDIFACT
listAgencyName	string	Text containing the name of the agency that maintains the list of codes.
listName	string	Text containing the name of a code list that is registered with an agency.
listVersionID	normalizedString	An identifier specifying the version of the code list.
Name	string	Text equivalent of the code content component.
languageID	language	An identifier specifying the language used in the code name.
listURI	anyURI	The uniform resource identifier (URI) identifying where the code list is located.
listSchemaURI	anyURI	The URI identifying where the code list schema is located.

A.2.3 DateTimeType

DateTimeType is used to define a particular point in time together with the relevant supplementary information to identify the time zone information. It is derived from the type dateTime. In a VPC file this is a specific instance of time using the ISO 8601 Common Era calendar extended format and abbreviated versions.

EXAMPLE yyyy-mm-ddThh:mm:ssZ for UTC as "2002-09-22T13:15:23Z"

Table A.2 describes the optional attributes for the DateTimeType data type.

Table A.2 – DateTimeType optional attributes

Optional Attribute	Base XML Type	Description
format	string	A string specifying the format of the date time content. NOTE 1 The format of the attribute is not defined in the UN/CEFACT CCTS. NOTE 2 This attribute is not needed in a VPC file, but is maintained for compatibility with Open Applications Group Integration Specification (OAGIS) and Manufacturing Enterprise Solutions Association International (MESA) use.

A.2.4 IdentifierType

IdentifierType is used to define a character string to identify and distinguish uniquely, one instance of an object in an identification schema from all other objects in the same schema. It is derived from the type normalizedString. Table A.3 describes the optional attributes for the IdentifierType data type.

Table A.3 – IdentifierType optional attributes

Optional Attribute	Base XML Type	Description
schemaID	normalizedString	An identifier specifying the identification schema.
schemaName	string	Text containing the name of the identification schema.
schemaAgencyID	normalizedString	An identifier specifying the agency that maintains the schema.
schemaAgencyName	string	Text containing the name of the agency that maintains the schema.
schemaVersionID	normalizedString	The version (as an identifier) of the schema.
schemaDataURI	anyURI	The URI identifying where schema data is located.
schemaURI	anyURI	The URI identifying where schema is located.

A.2.5 IndicatorType

IndicatorType is used to define a list of two mutually exclusive boolean values that express the only possible states of a property. It is derived from the type string. For VPC purposes the defined values for indicator type are “True” and “False”. Table A.4 describes the optional attributes for the IndicatorType data type.

Table A.4 – IndicatorType optional attributes

Optional Attribute	Base XML Type	Description
Format	string	A string specifying whether the indicator is numeric, textual or binary. NOTE The format of the format attribute is not defined in the UN/CEFACT CCTS.

A.2.6 TextType

TextType is used to define a character string (for example, a finite set of characters) generally in the form of words of a language. It is derived from the type string. Table A.5 describes the optional attributes for the TextType data type.

Table A.5 – TextType optional attributes

Optional Attribute	Base XML Type	Description
languageID	language	An identifier specifying the language used in the content component.
languageLocaleID	normalizedString	An identifier specifying the locale of the language using the ISO 639-1 [4] code for representation of the name of the language.

Annex B
(informative)

IACS asset owner guidance on patching

B.1 Annex organization

Annex B provides guidance to IACS asset owners that are establishing and/or operating an IACS patch management program. This Technical Report is written using a 'workflow' approach, in which the sequence of activities, tasks and requirements are written in the approximate order that would be followed by individuals managing a patch management system. Asset owner staff, consultants and contractors should find this information immediately actionable and relevant to the challenges associated with patching IACS systems. The goal of this Annex B is to help asset owners establish their patch management programs more quickly, increase effectiveness, reduce vulnerabilities and increase overall IACS reliability.

This annex focuses on the following major activities for patch management:

- **Information gathering activities** – This includes creating the inventory of updateable devices, building product supplier relationships and evaluating and assessing the existing environment and its supportability requirements.
- **Project planning and implementation activities** – This includes developing the business case, defining the roles and responsibilities, establishing a patch deployment and installation infrastructure and establishing a backup and restoration infrastructure.
- **Procedures and policies for patch management** – This includes monitoring for patches, evaluating patches, testing patches, installing patches and change management.
- **Operating a patch management system** – This includes executing the patch management procedures and policies, vulnerability awareness, outage scheduling, inventory maintenance, new device additions, reporting, key performance indicators (KPIs), auditing and verification. Often called "run and maintain," operating the patch management system will be a continuously repeating maintenance process.

B.2 Overview

The purpose of this annex is to describe patch management procedures and processes, along with guidance on how those procedures could be implemented by an asset owner with one or more control system environments. The objective of these procedures and processes is to assist the asset owner with the creation of their own program. Asset owners have the option to re-use, modify or abandon the guidelines appropriate to the size and complexity of their environment. Once the procedure is documented, it can be shared with those responsible for its execution, so that those individuals can perform the tasks more quickly, with higher quality and greater consistency. Without documented procedures, it is difficult to assign, train or ensure that objectives such as effective patch management will occur.

Figure B.1 illustrates the processes and procedures required to support the patch management workflow. Note that one-time project activities are not shown in the workflow.

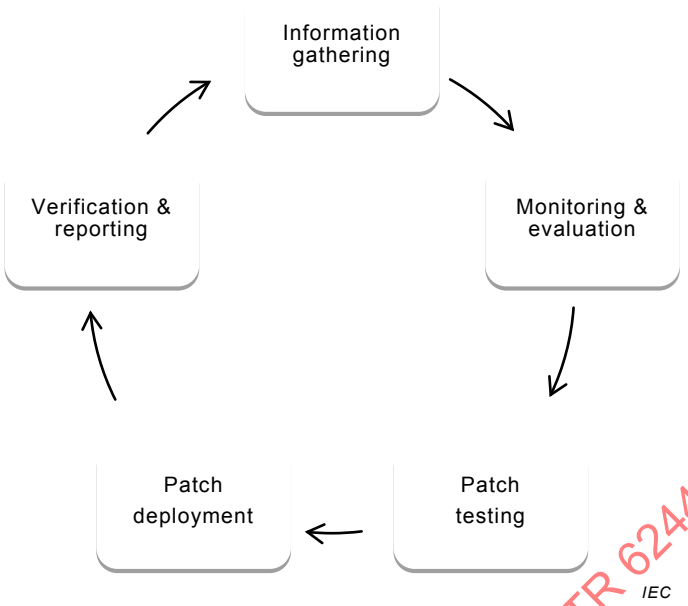


Figure B.1 – IACS patch management workflow

Each phase of the IACS patch management workflow is described later in this technical report.

A workflow-based approach is used in this Technical Report. It describes the steps involved, the activities performed and, where appropriate, how they are performed. The asset owner should document the procedure they intend to follow, so that it can be communicated to others, and implemented consistently, within their organization. Any objective that involves multiple steps will be better implemented if the entire procedure is documented.

This Annex B is written so that an asset owner will be able to establish their own patch management process by using this guidance as the starting point.

B.3 Information gathering

B.3.1 Inventory of existing environment

For IACS patch management, a large amount of information is required about the current environment, before analysis and planning can occur. This data can be very costly to gather and very revealing to potential attackers, so it should be secured appropriately.

Establishing an IACS patch management program begins with an accurate inventory assessment, to identify: the devices in scope, and the software and patch versions in use. If the asset inventory information is not accurate, neither will be the risk-based decisions based on that information.

Additionally, when a new vulnerability is discovered, an accurate inventory environment will enable the owner of the environment to determine which assets or devices in their facilities have that vulnerability. This will allow the owner of the vulnerable assets or devices to take mitigating actions to protect the vulnerable equipment. A more detailed discussion of some of the actions that can be taken, to mitigate the risk on a new vulnerability can be found in B.6.8.

The first step is to identify the components and devices that are part of the IACS. This includes all updateable device types, such as: servers, workstations, switches, routers, firewalls, printers, serial to Ethernet converters, programmable logic controllers (PLCs), remote terminal units (RTUs) and all non-updateable devices, which could be replaced by a patched or updated device. A number of resources and methods may be used to characterize the existing environment, such as:

- asset management information systems that may include: purchase records, serial numbers, asset tags and other identifiers of those electronic devices owned and maintained by the asset owner;
- IACS documentation such as: device lists, architecture drawings, design documentation original IACS product supplier documentation;
- IT documentation such as: internet protocol (IP) address lists, network drawings;
- physical inspection of the facility to identify devices and their connectivity relative to the documentation available;
- interrogation of switches/routers for MAC and IP addresses to identify connected systems;
- network attached device discovery tools, such as: slow speed ping sweep and network analyzers; and
- existing business impact assessment (BIA), business continuity planning (BCP) and disaster recovery planning (DRP) documentation, if it exists.

NOTE If BIA and BCP information is available; it provides additional insight, as it may categorize the criticality and importance of specific business processes and systems. This criticality data can be used throughout the entire IACS patch management program including: initial planning, patch evaluation and later patch installation planning.

The approaches above should all be leveraged during the inventory of the existing environment, to develop an accurate list of devices to be considered, and their criticality.

With an accurate inventory list of devices in place, the next step is to gather specific information from each individual device. The objective of this data collection is to identify any information that can be used, or is required, to establish and operate the IACS patch management program. Data collected should include:

a) **Ownership –** This identifies the asset owner or custodian personnel, and those resources capable of supporting it. This information will be used later when assigning responsibilities and critical decision making.

b) **Product supplier, make, model number –** This information will be used later when it is time to contact the product suppliers.

c) **Version –** The version associated with any hardware components, and their associated firmware, including the boot code version, firmware version and boot image versions.

d) **OS version –** The version associated with the OS environment. This includes the OS name, version, service packs, hotfixes, patches, service releases, etc. Depending on the environment, the OS may be part of a virtualization hyper-visor solution and the software of the virtualization host is also required. Alternatively, the OS may be part of embedded device firmware.

e) **Software versions –** The versions associated with software installed on top of the OS. Examples include web browsers, control system software, databases, remote access, etc. Refer to column 1 of Table B.3 for more ideas. It is important to document the product supplier for each software component, as this information will be required later, in addition to the software title.

f) **Redundancy –** This defines the fail-over and fault-tolerance capabilities of the hardware and software. This information will be used later to support evaluation, planning and installation of patches. For example, is a full outage required, or can the patch be applied to one device, part of a redundant set, and then the other, without interrupting IACS operation.

g) **Computer role –** This defines the function of individual computers and is essential in order to evaluate the impact of restarting the computer (if necessary) once a software update has been installed. For example, if the computer is a server running a business-critical application, it is advisable to schedule software update during periods when they will have minimal impact on the business. It also may be necessary to make arrangements for business continuity, so that users can continue operations while the server is being restarted.

h) **Computer group –** This defines the categorization and the grouping of devices performing a similar function (for example, domain controllers, operator workstations) that would be expected to have similar or even the same hardware, software, configuration and IACS patch management strategy.

i) **Network architecture and connectivity –** This defines the network architecture and structure. Understanding the layout of the network infrastructure, its capabilities, security level, link speed and link availability is important for effective patching. This layout should also include remote access systems such as support and management systems. Software updates can vary in size, and knowing the constraints of the network infrastructure can potentially reduce any delays in distributing software updates. It can also dictate the manner in which the software update will be deployed to, and installed on, particular client computers.

j) **Installed and not installed software updates –** This identifies which software updates have or have not been installed on computers, and is essential information.

k) **Support status –** This identifies the support status of each computer system. If software or hardware updates are not available and upgrading is not practicable for a computer system this needs to be recorded, because the system will fall outside the patch management regime and will need a separate security management regime, such as a hardened configuration or use of multiple layers of defense (defense-in-depth). A known or expected end of product supplier support date should be recorded. A periodic review of this data will allow for controlled changes in support plans.

l) **Inter-dependencies –** This describes the inter-dependencies between the different device types, categories and groups of devices. This information will support later evaluation, planning and installation of patches to ensure risks are mitigated for inter-dependent devices.

m) **Criticality –** This describes any management of change constraints, based on the criticality of the components and groups of systems. Often the critical data paths in a control environment have not been analyzed or documented with the detail required to identify critical interdependencies. The focus of patching is on applying updates to software running on these computers. The decision about when to patch will start with a thorough understanding of the critical processes and critical data flows that the computers and embedded controllers provide to the system operation. The operation of each critical process requires the interaction and interdependencies of computer systems to be understood.

n) **Vulnerability assessment tools applicability –** This describes if assessment tools can be run against the system either automatically or manually or never. The asset owner should consider vulnerability assessment tools as an additional method for identifying security vulnerabilities associated with their IACS. Vulnerability assessment tools can help identify and prioritize risks that can be mitigated through configuration changes, installing patches or other mitigating controls. Active vulnerability scanning tools can negatively impact the IACS, and should only be used after testing under controlled conditions and at specified scanning levels.

o) **Configuration Files –** Note if any configuration information will need to be captured before a modification and then have to be reapplied afterward.

To support information collection above, consider the guidance in B.3.2.

B.3.2 Tools for manual and automatic scanning

At the time of this writing, some tools are emerging under development while others are currently available to facilitate the automated data collection, identification and characterization of the control network architecture and those devices attached to the control network. The more invasive the tools are, the greater the risk they might pose to IACS. It is crucial that the user of any tool be intimately aware of the impacts imposed by applying the tool on the targeted IACS, including those industrial automation processes containing multiple systems architectures comprised of different manufacturer's products and networks. Additionally the user of the tool should interact with the full range of product suppliers that manufacture the systems and devices contained in the IACS, in order to fully understand the hardware and software compatibilities relative to applying the automated tool(s).

It is also important to know that many of the automatic scanning tools apply new “plug-ins” on a very frequent basis. This means that a scan that worked last week may experience issues the next week, due to new tests that the tool is now performing. Each asset owner must monitor and control the configuration of the tools used, and should consider testing any new “plug-ins” before they are added to an automated scanning tool.

It is important that when running any tools causes issues that feedback is provided to the tool creator as well as to the appropriate group that created the affected equipment.

Automated tools may not be able to collect the full range of asset inventory data elements described above, such as OS, patch levels, ownership, criticality, management of change constraints or vulnerabilities, so a manual data-entry component typically is also required. Some inventory collection tools are integrated into automated patch distribution tools.

The choice to use automated tools to assist with collection of the inventory may also be impacted by the important ongoing need to maintain the asset inventory data and verify the existing systems configurations and architecture from time to time. The value of having a highly accurate asset inventory is that it is the fundamental data needed for the appropriate assessment of risk when operating the IACS patch management program. For example, when a new vulnerability is discovered, an accurate inventory environment will enable the owner of the environment to determine which assets or devices in their facilities have that vulnerability. This will allow the owner of the vulnerable assets or devices to either install the appropriate patch(es), and/or take mitigating actions to protect the vulnerable equipment. A more detailed discussion of some of the actions that can be taken, to mitigate the risk on a new vulnerability can be found in B.6.8. A more detailed discussion of when to schedule an outage for patch installation may be found in B.8.4.

Also note that for some automated inventory collection tools, an agent may need to be installed on the various computers either as a client, server or as a separate monitoring system, anywhere in the IT or IACS architecture. A full configuration profile for a machine can be sizeable, including the inventory application processing demand. Attention may be required to confirm that this process and the associated tools are not detrimental to the systems that they are designed to serve. Particular attention and expert knowledge is required when applying these tools on redundant and deterministic control networks.

Another option available to asset owners to expedite first-time and ongoing data collection is to consult with their product suppliers for the suppliers recommendations on supported methods, tools and services.

B.3.3 IACS product supplier contact and relationship building

After all devices have been identified and their specific versions have been collected as per B.3.1, the next step is to identify the IACS product suppliers for those components of the IACS. The input to this step is a list of hardware product suppliers, software product suppliers and service providers that will be instrumental to the asset owner’s IACS patch management program.

The following information is required for each IACS product supplier:

a) current business name and history of acquisitions that may affect supportability of legacy products in the asset owner’s IACS;

b) contact information for groups within the product supplier’s organization, which can provide information to the asset owner; and

c) state of support contracts for asset owner products, or the costs to establish agreements to be entitled to notification of patches.

Additional information may also be suitable for the IACS. Table B.1 provides an example profile of the type of information that can be compiled for each product supplier.

Table B.1 – Sample product supplier profile

Main website:	http://www.microsoft.com
Support website:	http://support.microsoft.com http://update.microsoft.com The Microsoft support website provides links to technical support for all supported Microsoft products and OSs. This website also provides links to downloads and updates for all supported Microsoft products, and to product-specific solution centers. The Microsoft Update website is a component of the Windows Update framework, and is an automated service that, when accessed by a computer, will scan to verify the computer is running the most current versions of the installed Microsoft software, and that it has the latest updates for its Windows OS. For Microsoft updates, please refer to: http://office.microsoft.com/en-us/downloads/default.aspx
Support website user account:	Record this information if a shared account is used. However it is better if each support person should have an individual account.
Update notifications:	The Microsoft Windows Update Service can be configured to send email notifications about new available updates. The Microsoft Security Newsletter, which is sent out by email, can be subscribed to at the following website: https://profile.microsoft.com/Regsysprofilecenter/subscriptionwizard.aspx?wizid=6e2dfc95-9fea-4e12-827e-c9d2135149b9 Comprehensive Email Notification of Security Alerts can be subscribed to at the following website: https://profile.microsoft.com/RegSysProfileCenter/subscriptionwizard.aspx?wizid=5a2a311b-5189-4c9b-9f1a-d5e913a26c2e&lcid=1033 The Microsoft Security Advisories RSS feed can be subscribed to at this address: http://www.microsoft.com/technet/security/advisory/RssFeed.aspx?securityadvisory
Additional information:	The Microsoft Windows Update Services enables system administrators to deploy and install the latest Microsoft product updates to computers running a Windows OS. Windows Update Services allows administrators to manage the distribution of updates that are released through Microsoft Updates to computers on the network. Microsoft Windows Update Services also provides a management process for tracking, approving and installing applicable cyber security software patches and updates for Cyber Assets within the Electronic Security Perimeter(s).
Type of product supplied:	See Table B.3 for example categories. Browser, CD/DVD/Tape Backup, Client Tool, Document Editing, Driver, Email, Messaging, Multimedia, OS, Remote Access, Security Tool and Server App – Software and Hardware
Security point of contact	This is a contact that is named by a company service provider, where available such security contact should also be requested and recorded from vendors.

For purchased software, including the OS, the product supplier can be consulted as to the extent to which they perform pre-release testing both of their software as well as any underlying software such as OS accessories. The end user needs to also consider the effect of patching on end user license agreements (EULAs), warranties and support agreements. Consider whether the vendor support agreement addresses specifics as to the addition of or lack of patching, and who is responsible for implementing patches. Guidance from a product supplier for process critical systems should be seriously regarded.

Consideration can also be given to the potential of contracting third-parties such as the IACS product supplier or integrator to provide maintenance services that include risk management of patching. In such cases, it is still appropriate to consider that the asset owner is ultimately accountable for the patching process as severe liabilities may be incurred, even when leveraging the services of others to execute the process.

NOTE An emerging trend is to incorporate overall cyber security requirements into procurement specifications. It is important to note that the statement by any vendor of patch compliance to standards or regulations does not necessarily equate to a secure network.

In more hazardous or regulated situations change management restrictions and/or procurement specifications will be more extensive. This may mean requiring full range impact assessment and recovery processes to be established and tested, or validated to certain regulatory statutes or standards using factory acceptance testing (FAT) and site acceptance testing (SAT) practices.

B.3.4 Supportability and product supplier product lifecycle

It is important to have an accurate understanding of the suppliers' products that are in use in the IACS, but equally important is the supportability of those products.

It is critical to the cyber security of an IACS system to determine what support is offered by the IACS product supplier in terms of version upgrades, patches and services offered. This needs to be determined for every hardware and software component of the IACS. The following questions and information should be asked of each of the product suppliers:

a) Is there a support agreement in place between the asset owner and the suppliers that covers each products supplied?

b) What products still have support and release of patches?

c) Have they announced an end-of-life date for specific products? Is there an upgrade path?

d) What can be patched, legally, without voiding product supplier warranty?

e) What information, support or patches are available for the product supplier's legacy solutions?

f) What are the product supplier testing cycles and timeframes?

g) If applicable, what is the standard response time from the release of a patch by the patch supplier, until the IACS product supplier tests and confirms support of those patches?

h) Can the IACS product supplier distribute patches to the asset owner? Or must the asset owner download the patches from the patch supplier. Can the IACS product supplier also test and install the patches at the asset owner premise?

i) What security measures are used to ensure the trusted delivery and installation of the patches provided?

Asset owners should not limit themselves to this list of questions when determining the supportability of their products.

B.3.5 Evaluation and assessment of existing environment

The objective of this step is to assess all of the gathered information about the IACS, including the devices for update, hardware/software details, products in use, patches required and the supportability of those products. This information forms the basis of establishing the IACS patch management program, as it defines the scope of the IACS and the current state of its patches. This may be referred to as a scoping study.

At the start of the evaluation and assessment of the existing environment, the asset owner will be faced with the task of gathering and processing an overwhelming amount of information. They may also be considering ways to focus patch management efforts to those components of the highest criticality, highest vulnerability and highest effectiveness. The information in this section helps to prioritize what information to gather and how to evaluate the gathered information and prioritize IACS patch management efforts.

The first evaluation is to determine what will be patched. There may be legacy systems due for retirement, which will be decommissioned before the patching program is operational. There may be isolated systems or components of low importance to the organization, and would have little or no impact on the IACS security.

During this evaluation it is important to remember that any component with exploitable vulnerabilities can be used to gain a foothold on the network and thus compromise the entire system.

Asset owners should evaluate their IACS based on the following factors, to determine which software, hardware and other components are to be patched.

a) **Supportability –** Which parts of the system have patches available and which do not. For parts with unpatched vulnerabilities, mitigating controls should be put in place based on risk assessment.

b) **Criticality and business impact –** Security vulnerabilities that remain in IACS critical components can have an impact on the entire organization.

c) **Robustness and redundancy –** A good system design is engineered with sufficient redundancies to allow patching to occur with little business disruption. The amount of redundancies should correspond with the criticality and business impact of the IACS components.

d) **Communication capabilities –** This context can apply to the IACS as a whole, and the network architecture that it communicates within. It can also apply to the individual software titles installed on each device.

e) **Purpose of software component –** The IACS may include hundreds of software titles, each performing a different purpose. Depending on its purpose and/or criticality, the asset owner may choose to include, exclude or defer patching of certain software components.

The objective of reviewing and assessing the device and patch related information thus far is to determine a feasible and realistic scope to include in the first and subsequent patch management program efforts. With limited resources it may only be possible to address a limited number of devices and risks in the first phase, allowing additional patching efforts to occur in later phases or years.

B.3.6 Classification and categorization of assets/hardware/software

This section describes categories and methods for the categorization of devices, hardware and software to allow the prioritization of IACS patch management resources. The categorization proposed in this section can be useful in determining which are the highest risk devices, which have the greatest vulnerability, and which devices will receive potentially the greatest benefit by applying patch management resources, in the most effective manner possible.

One categorization that can be applied to devices, hardware and software are the attack surfaces available through their communication capabilities, such as in the list given in Table B.2:

Table B.2 – Communication capabilities

Capability	Description
Listens for and accepts connections	This includes devices, services, applications and other processes that are listening for and capable of accepting network communications and data from other Cyber Assets. These are a high security risk because they may allow an attack to remotely exploit the open incoming or inbound-port. These types of services and process will appear on network ports (such as netstat) collection results with the state of LISTENING.
Outgoing connections only	This includes devices, services, applications and other processes that are capable of establishing network communication sessions and/or sending data to remote Cyber Assets across the network but will refuse to accept connections initiated by remote devices. These services, applications and processes have medium security risks because it may possible for an attacker to compromise the service or process by hosting a malicious network service listening for connections. These types of services and processes will appear on network status collection results as outgoing or outbound-ports, but never in the state of LISTENING. This also refers to client applications making outbound-only connections to servers.
Hardwired communication only	This includes devices that have universal serial bus (USB), compact disc (CD), digital versatile disc (DVD), serial, Ethernet or other ports available for hardware exchange of information. USB devices are a known source of vulnerability due to the auto-install nature of the USB drivers. This also includes devices with programming ports, serial or networked, which are normally not connected, but can become sources of attacks. Depending on the physical security offered for the device these may be high security risks, if there is no physical security, to low risks if there is physical security and sufficient processes to ensure that no unauthorized or unchecked access is allowed.
No communication capabilities	This includes devices, services, applications and other processes that have no communication capabilities and cannot send data to remote Cyber Assets, and cannot accept or listen for connections from other Cyber Assets across the network. With the exception of a compromised supply chain, these services and processes have very low initial security risk. Services and applications with no communication capabilities that are on devices with communication capability do have security risk in post exploitation scenarios. These types of services and processes will never appear on network status collection results. Note that this category does not include devices that rely on “air gaps” since those devices may still have communication capabilities such as with USB, DVD and other methods.

The asset owner may choose to prioritize patching efforts to those devices and software components that listen for and accept network connections, followed by those that have other communication capabilities.

IACS devices may have a large number of software titles installed on them, each having their own strategy for patch management. See the table below for software rationalization, determining how to handle each software type, and possible recommendations for action.

Table B.3 – Sample software categorization

Software Type	Definition	Recommendation
Browser	Browser includes any application used for browsing Hypertext Markup Language (HTML) content and/or integrates with a web browser.	In a control system environment, any software categorized as browser should be evaluated to determine if it is necessary. If not necessary they should be considered for removal or being disabled. It is recommended that asset owner standardize the web browser, and remove all other browsers and add-ons.
CD/DVD/Tape backup	CD, DVD and Tape Backup, include any application for making backup using CD, DVD or tape.	In a control system environment, CD/DVD/Tape backup applications not used as part of a Backup and Restoration Strategy or device support strategy should be removed or disabled. The risk of CD/DVD/Tape backup applications being exploited remotely is low. If they are deemed to be necessary by asset owner, they can be given a low priority for frequent update.
Client tools	Client tools include all other desktop tools that have not been categorized yet.	The risk of client tools in an IACS being exploited remotely is low. Therefore, client tools can be given a low priority for update.
Control (IACS)	IACS includes the core software provided by each control system product supplier for plant operations.	Industrial automation and control systems are critical to the operation of asset owner. Control systems need to be updated as required and authorized by product suppliers.
Document editing	Document editing software includes applications specifically used for creating or editing or viewing documents.	In a control system environment, any software categorized as document editing may be considered unnecessary and are candidates for removal. If they are deemed to be necessary by the asset owner, they should be maintained to receive the latest security patches.
Drivers	Drivers are software that allow communication with hardware devices such as video cards, audio cards, modems, network cards, redundant array of independent disks (RAID) controllers, etc.	Device drivers are not normally remotely exploitable. Therefore, it is not usually necessary to monitor and regularly install updates for drivers.
Email and messaging	Email and messaging category includes any software for communication between computer users across the network via chat, email and video conferencing.	In a control system environment, email software may be considered unnecessary and are candidates for removal or being disabled. If they are deemed to be necessary by the asset owner, they should be standardized and maintained to receive the latest security patches. One recommendation might be to use outbound Simple Mail Transfer Protocol (SMTP) messages rather than employ a full email application if the requirement is that the IACS send notifications of a situation via an email message.

Software Type	Definition	Recommendation
Multimedia	Multimedia software includes any software for working with pictures, audio and video.	In a control system environment, multimedia software may be considered unnecessary and are candidates for removal or being disabled. If they are deemed to be necessary by the asset owner, they should be standardized and maintained to receive the latest security patches.
OS	OS software includes components and frameworks distributed from the OS supplier that are not explicitly defined in other categories.	Security patches for each OS need to be applied as released by the OS supplier, within the bounds of what the control system product suppliers support.
Remote access	Remote access applications include software that are used to access or administer remote devices or server components for interactive access.	If remote access applications are not utilized for remote support, they may be considered unnecessary and candidates for removal or being disabled. If remote support is critical for IACS operation, the remote access tool used should be standardized and maintained to receive the latest security patches.
Server applications	Server applications provide transaction based functions and data to users. They may supply information at the operator interface level in the IACS or supply data to users on network segments outside the IACS. DCS and Historian systems are typical examples of server applications.	Server applications listen for requests from users. These applications may have ports and services that can be remotely exploited and therefore patching these applications for security vulnerabilities is particularly important.
Security tools	Security tools includes purpose built software for enabling or performing security controls and functions.	Security tools should be kept up to date in order to be effective. If possible, security tools installed on assets should be standardized.
Smart device or intelligent communication finite element instrument	A process or control finite element containing its own processor, control and communication ability over wired or wireless networks, subnetworks, having firmware and application updates and patches from its manufacturer.	Evaluate on regular intervals for patches for the device vulnerabilities for both ingress and egress data.

B.4 Project planning and implementation

B.4.1 Overview

Once the initial information gathering and assessment activities have been completed, the next step is to begin the detailed project planning to establish the IACS patch management program. Planning is important, because if the project is not adequately planned, it will not fulfill the patch management objectives. The steps involved in project planning to establish a patch management program are shown in Figure B.2.

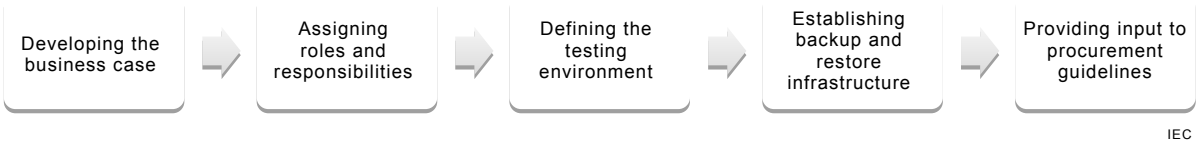


Figure B.2 – Planning an IACS patch management process

The first step in establishing an IACS patch management program is developing the business case that can be presented to senior management in order to secure the necessary funding, resources and support.

It is critical to the success of the program to have clearly defined and communicated ownership, accountability, roles and responsibilities throughout the asset owner organization to perform IACS patch management.

Additional steps may include guidelines on the implementation of a testing environment, automated patch deployment and installation infrastructure and backup/restoration infrastructure.

The last step is to provide input to the procurement requirements and legal terms and conditions to ensure that product suppliers are supporting the asset owner's IACS security objectives.

B.4.2 Developing the business case

Personnel within an asset owner organization will have to justify the costs and benefits associated with an IACS patch management program to their senior management. Many organizations have not historically performed IACS patch management, and it may be necessary to educate critical decision makers on the reasons for patching, and the risks that are reduced as a result.

NOTE IEC 62443-2-1 provides guidance on developing the Business Rationale for establishing an IACS Cyber Security Management System, as does *Cross Sector Roadmap for Cybersecurity of Control Systems* [16]. The business value and cost items discussed in those documents are directly applicable to the business case for a patching program

To appreciate the full business impact of poor or non-existent patch management, consider the following.

- **Increasing targeting of IACS –** There is now significantly more information available that the frequency of incidents and attacks on IACS assets has increased, resulting in increased likelihood of a successful cyber-attack with significant negative consequences.
- **Downtime –** Computer downtime in an organization can be costly. Business or nationally critical systems can be interrupted. The opportunity cost of lost end-user productivity, missing transactions on critical systems and lost business during an incident can have a negative financial impact to the business. Most hacking attacks result in some downtime, either as a direct result of the attack itself or as a necessary part of remediation efforts. Some attacks have left computers down for several months.
- **Quality –** The quality of the product produced by the process under control can be affected by compromised IACS devices that are not able to perform their intended function due to compromised computing capability. Security patching reduces the likelihood of compromise of the IACS and so protects product and process quality.
- **Health and safety –** The health and safety of company personnel can be jeopardized if a cyber-attack occurs.
- **Environment –** The environment may be polluted or otherwise negatively impacted if a cyber-attack occurs.
- **Remediation time –** The time required to fix a wide-ranging problem in an organization can be sizeable and long. Finding and installing clean versions is often an issue when a good backup program is not in place. (See B.4.5 and B.6.7.) The cost to rebuild a computer's software environment is not negligible, and many security breaches require a complete reinstallation to be certain that back doors (permitting future exploits) are not left by the attack.
- **Questionable data integrity –** In the event that an attack damages data integrity, the cost of recovering that data from the last known good backup or confirming data correctness with customers and partners can be high.

- **Negative public relations –** Failure to protect customers' personal information can result in a business being viewed as unreliable. A health, safety or environmental (HSE) incident resulting from a cyber attack can result in negative community, employee and shareholder reactions, including lawsuits. This, and other failures due to poor patch management, can result in negative public relations.
- **Legal defenses –** Poor patch management can leave a system vulnerable to cyber-attack or costly legal action from dependent organizations. Organizations providing important services to others have had their patch management process (or lack of one) put on trial.
- **Stolen intellectual property –** A company's intellectual property is put at risk by poorly maintained patch management. Resulting costs of intellectual property being stolen or destroyed can be sizeable.
- **Regulatory compliance –** Many asset owners have regulatory requirements they have to fulfill, that could be affected.

It is also important to consider the following when proceeding with the program.

A comprehensive patch management process is one that can have a significant cost associated with it. The up-front costs to set up the equipment resources and software licenses for a test environment, that may be needed, for patching may be very expensive, and the ongoing management can be resource intensive. The personnel resources required to sustain the patching process may be quite high if the number of devices to patch is high. However, the costs associated with the program will usually be less than compared to an incident for a poor or non-existent program. See B.8.2.

It is important for business leadership and local site leadership to understand the development, implementation and workflow cost to sustain the patching program. Failure to understand this cost up-front could result in a situation where patching is not sustained and the costs initially invested to roll-out the patching program will have yielded little security improvement. The business case should articulate the balance between the workflow cost and the value to be derived from patching IACS devices.

B.4.3 Establishing and assigning roles and responsibilities

For the IACS patch management program to be successful it requires the support and participation of stakeholder groups throughout the asset owner organization and their product suppliers. It is the objective of this section to assist asset owners in the definition of those activities required for patch management, and how to both determine and communicate those associated roles and responsibilities.

The first step to assigning roles and responsibilities is to identify what activities and business processes will be performed. This is an iterative process that is completed incrementally over time, as many business processes will be discovered as the patch management processes are defined.

To begin with, the following minimum patch management activities should be assigned:

- establish, document and maintain product supplier relationships, support contracts and communication;

  NOTE 1 Check if there are multiple people each responsible for a group of product suppliers.

- monitor for vulnerabilities and availability of patches and upgrades;
- perform patch evaluation;

  NOTE 2 This may require a multi-disciplined group of experts throughout the organization, such as security, IT, networks, controls, safety, engineering and maintenance.

- perform patch testing;
- deploy and install patches;

- track, validate and report the installation of patches; and
- audit and monitor the effectiveness of the program.

The list above is an overview of all activities associated with patch management. Additionally, there are multiple stakeholder groups that may be involved, each sharing responsibility for some or all activities.

Some of the stakeholder groups involved may include:

- technical experts;
- control system application support;
- manufacturing operations application support;
- cyber security;
- product suppliers, systems integrators;
- information technology; and
- plant management.

With a list of tasks to be performed, and a variety of stakeholder groups to perform them, it is necessary to map them together. One approach is a responsibilities matrix, also known as a responsible, accountable, consulted and informed (RACI) or responsible, accountable, supportive, consulted and informed (RASCI). The objective is to cross reference the task to the role, and map out the level of responsibility. The levels of responsibility are defined in Table B.4.

Table B.4 – Responsibility assignment definitions

Assignment	Letter	Definition
Accountable	A	Approves the completed work and is held accountable for it. There should only by one 'A' for a given task.
Responsible	R	Person who conducts the work and owns the problem or impact. All activities should have at least one 'R'.
Supportive	S	Can provide resources or can play a supporting role in implementation. Only occurs if requested by the 'R' role.
Consulted	C	Has information and/or capability necessary to complete the work. These people need to give input before the work can be performed and signed off. These people are "in the loop" and active participants.
Informed	I	Needs to be notified of results and remain aware of activities. These people do not contribute directly to the task or decision. This is typically a one-way communication from R to I.

The RASCI responsibilities chart is an effective method to identify, assign and communicate to multiple stakeholder groups their role in IACS patch management.

It is ultimately the asset owner's responsibility to identify tasks, determine stakeholder groups and assign roles appropriate to their organization. The level of capability between the IT and IACS groups, as well as the product supplier may provide unique opportunities or constraints. The asset owner may choose to centralize this function for a number of facilities, or allow each facility to handle it independently. The asset owner may outsource some responsibilities, or handle them all with company personnel. Lastly, this exercise helps identify those activities that are unassigned.

A sample RASCI chart is shown in Figure B.3. As seen in the 6th column, labeled John Doe, activities can be assigned to specific individuals or groups.

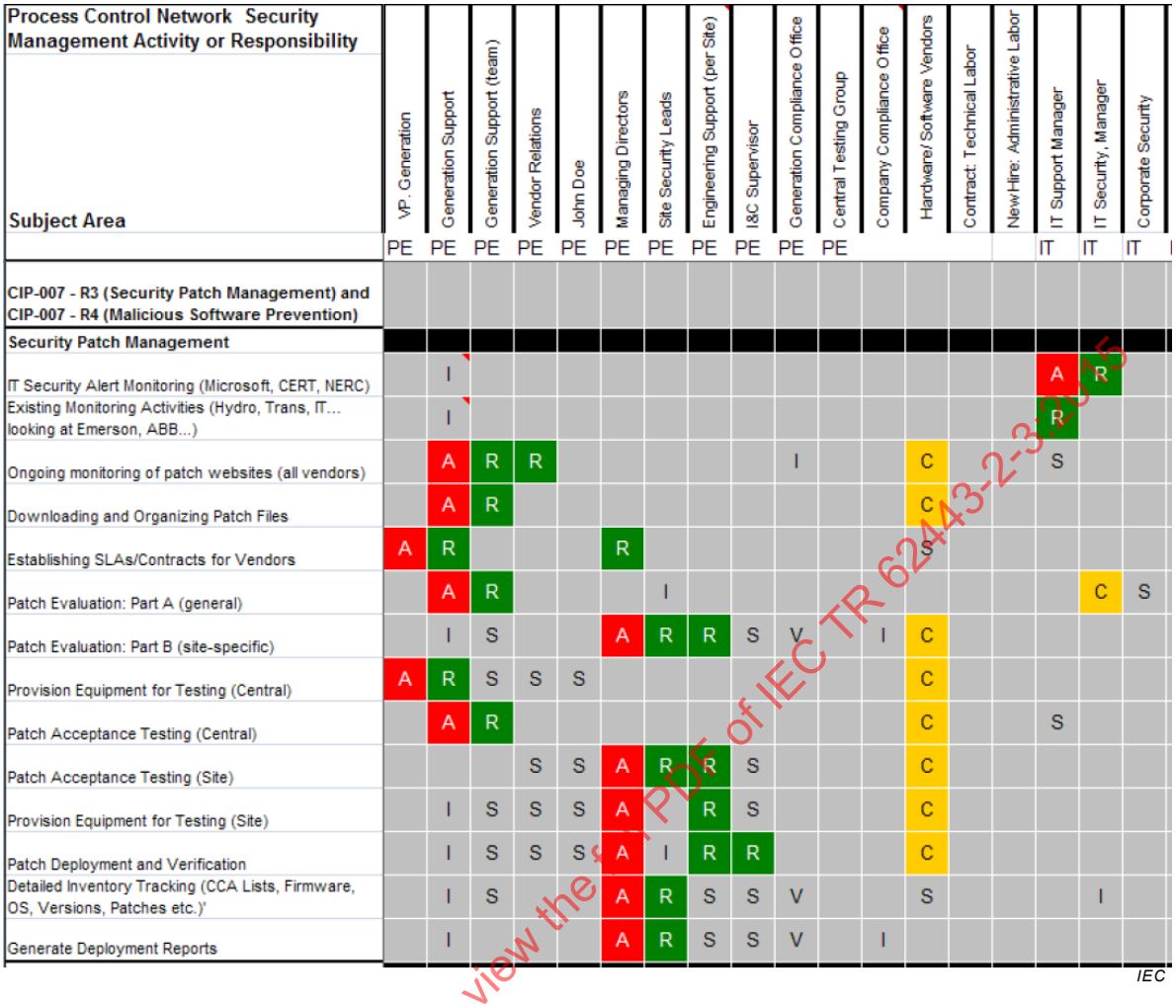


Figure B.3 – Sample responsibilities chart

B.4.4 Testing environment and infrastructure

Asset owners should ideally have a test environment that allows for functional testing of security patches prior to deployment and installation in the production environment. This would give the asset owner the confidence to Install patches to the production systems in use. Some examples of patch testing environments include:

- A permanent security test platform designated for testing of security patches, antivirus and service packs prior to installation in the production environment. A representative configuration of hardware, software and applications that allows for testing, in as close to the production environment as possible.
- A process control lab where research and development is conducted makes an excellent test environment. The process control lab is often isolated from the corporate network and usually isolated from production process control networks.
- A virtualized system is a method to test and recover easily if a patch is found not compatible with the process control systems.
- Development/engineering systems may tolerate the downtime needed for patch functional testing and validation.
- Training simulator systems by their nature may have time available to do initial patch installation, functional testing and validation.
- Less critical operations that can tolerate flexible scheduling and downtime required to do the patch testing.

While a company may employ the same IACS product and version at more than one manufacturing location, each instance may be slightly different due to the implemented design, custom application/control software and support differences from one location to another. IACS tested and approved patches have been known to create operational issues when applied because of seemingly minor differences between sites. Creating a test environment that matches the production environment can be very expensive and not obtainable. If this is the situation, the user must consider alternate approaches to reduce the likelihood of patch installation causing problems with the in-production IACS. Abandoning patching due to the cost of fully testing patches before installing them on the production equipment must be avoided. Leadership may be willing to accept the additional risk of not performing detailed offline patch testing.

Alternatives include relying on IACS product supplier testing and monitoring for reports of issues when others have installed the patches and/or following a carefully managed patch roll-out program that tests the IACS vendor approved patches directly in the production environment following a plan that minimizes the potential for common mode failures.

In many critical manufacturing situations there may be more than one IACS device of the same type that is performing the same function. For example there may be more than one operator workstation that has the same operational controls. The patch roll-out plan may install the patches on one of these operator workstations and have users carefully watch for any operational issues over one or more days. If no issues are observed, install the patch on the next IACS device. In some cases the installed architecture may employ redundant devices in an automatic fail-over scheme. Patches can be installed on the off-line redundant device and then the off-line device is placed into service as the primary device.

B.4.5 Implement backup and restoration infrastructure

All process control systems should have a disaster recovery process plan (see IEC 62443-2-1). This is critically important for when a company is patching systems. There have been instances when patches have caused disruption of the operation of the process control system. While testing of patches is intended to minimize the probability of negative side effects due to patch installation, those negative effects may still happen. The ability to restore a failed system to a known good state is an important part of a patch testing, validation and implementation program.

The effort to restore a system can be simple or complex depending on the system architecture. A system architecture that is organized to simplify the number or frequency of full images can greatly reduce the number of man hours a company invests in backups.

The factors that must be managed are the ones that increase the number of different configurations or versions on company equipment. If backup images can be shared between many devices then the time needed to make more backups can be greatly reduced.

Operations can also be simplified if a company can take advantage of technologies like terminal services or virtual machines.

The length of time between full backups should be precisely managed. The specific length of time will be dependent on how many changes are made in a defined period of time or how long it will take to reapply any changes and newer patches after a full backup is restored.

There are no fixed rules for these considerations; each support team should evaluate the options for themselves. The choices made are often also based on considerations like the criticality of the systems and how easy they are to bring back on line. A device's recovery approach should be commensurate with its criticality. For example, if an outage of 30 minutes is intolerable, then the recovery mechanism should be less than 30 minutes.

Retention of backups is another area that does not have hard rules but will be dependent on how often changes are made. The key considerations are how many and how critical are the changes currently running compared to what is in the backup. If you have more than three backup versions then you really need to justify the value of the oldest ones.

An important recommendation is that on a planned basis a backup should be restored and put into use in production or a test system. This will insure that the methods of backup and restoration are fully understood and can be implemented without error.

Critical backups should be tested off-line to verify their integrity prior to performing patch management activities, and to ensure that they can successfully restore to a working system, in the event of a failed patch, when necessary.

Every backup process shares common basic infrastructure, including:

- **Full backup image process**;
- **Incremental backup process –** a company that always does full image backups may consider this optional;
- **Management of change process –** if a company is unable to monitor what has been changed there is a need to do more frequent full backups;
- **Backup storage –** it is common to have 3 reasonably recent images located in more than one secure locations with at least one location off site;
- **Recovery method evaluation strategy –** a process for determining when a restoration from a backup is necessary, and which backup copy should be used; and
- **Restoration process –** a process to be followed after it has been determined that a recovery or restoration is necessary.

B.4.6 Establishing product supplier procurement guidelines

When negotiating contracts with IACS product suppliers, original equipment manufacturers (OEMs) and system integrators, it should be specified that their products comply with the asset owner's current security policies, including policies for deployment and installation of anti-malware and patches. FAT should include the functional validation of the process control system in the security compliant environment. Some product suppliers do, and some do not, certify their products in an anti-malware environment, or only with specific anti-malware products. Product suppliers, OEMs, system integrators and the asset owner should agree on which anti-malware product will be used, and make sure that particular anti-malware product is kept up-to-date, in order to stay certified.

Refer to Good Practice Guide: Manage Third-Party Risk [11] and Cyber Security Procurement Language for Control Systems [14].

B.5 Monitoring and evaluation

B.5.1 Overview

The purpose of this section is to provide procedural guidance on how to determine if patches are applicable, the impact upon the IACS environment, as well as risks associated with and mitigated by the patch. At the end of the patch monitoring and evaluation process, a clear decision can be made on whether to install the patch, deploy other countermeasures or do nothing. The steps in monitoring and evaluating are shown in Figure B.4.

Figure B.4 – Patch monitoring and evaluation process

B.5.2 Monitoring and identification of security related patches

The purpose of this section is to provide guidance on the regular monitoring and identification of patches considered applicable to the IACS environment.

Asset owners are encouraged to initiate contact with each of their product suppliers, and build a relationship that ensures the asset owner is entitled to receive patches and updates to their hardware and software. At a minimum, the asset owner should know methods to communicate with each of their hardware and software product suppliers for updates, and if there are updates available. On an interval, consistent with the release of updates from the product suppliers, the asset owner should contact each product supplier to identify any patches available. The interval chosen for monitoring their product suppliers for patches can be the supplier's patch schedule, the asset owner's patch schedule or as required by regulations, based on the resources available to the organization or the impact of the control system security on the business.

Tasks associated with monitoring for patches that should be assigned to roles within the asset owner organization include:

- managing inventory and creating the list of hardware and software;
- managing the relationship between asset owner and each individual hardware or software product supplier;
- maintaining the list of patch and vulnerability update sources;
- monitoring for product supplier patch releases and approvals periodically; and
- notifying and/or documenting patches to begin the patch evaluation process.

Clearly assigning the above activities to responsible individuals or groups within the asset owner's organization, will ensure that they are performed.

When establishing a patch management program for the first time, there could be several hundred or thousands of available patches that may be available. The best approach is to compile the patches available from each product supplier, and proceed to the patch evaluation process as a group. It is not uncommon that suppliers periodically publish updates which accumulate a number of patches into one update. Such updates, often called service packs or fix packs, can greatly reduce the effort of installing large numbers of patches.

There are several methods for identifying patches for installation in the IACS environment:

a) asset owner makes a request to the product supplier or OS platform manufacturer for a list of available or recently released patches;

b) product supplier or OS platform manufacturer proactively releases a notification, or posts a list of available or recently released patches; and

c) asset owner directly assesses the patches installed on their IACS devices and compares, manually or automatically, against the list of available or recently released patches.

The output objective of this step is to create a list of patches that are available for the patch evaluation process.

B.5.3 Determining patch applicability

Applicability of a patch can be determined with three questions:

a) Has the control systems product supplier qualified and approved the patch for installation?

b) Is the available patch appropriate for a device or application currently in use?

c) Is the update security-related to mitigate a vulnerability?

If the answer for all questions above is “Yes,” then the patch or update is considered applicable and should be further evaluated.

If any answer above is “No”, then the patch may be considered not applicable to the environment, recorded accordingly and no further action required.

NOTE 1 There is the scenario where a patch is applicable to the environment, but does not mitigate a security vulnerability such as a functionality or reliability improvement. This Technical Report provides no guideance on whether this type of patch should be installed or not; that is the responsibility of the asset owner to determine.

As the focus of this Technical Report is on patch management for the purposes of cyber-security, this procedure will not provide guidance on the installation of non-security updates. These other updates could provide reliability improvements, overcome bugs or add features that do not improve the security robustness of the device or application. These patches are left to the discretion of the asset owner to determine their need for these enhancements.

NOTE 2 If the patch is required, and the patch requires a previous patch that was not installed, then the previous patch’s assessment should also be redone.

There may be a significant number of patches available from product suppliers each interval. Asset owners should verify if they have the product, version or feature in use in their environment before evaluating the available patches.

The following tasks are associated with determining patch applicability and should be assigned:

- managing inventory and maintaining the list of hardware and software;
- notifying and/or documenting of patches to begin the patch evaluation process;
- investigating and researching of technical advisories and knowledge bases to determine if they are security-related; and
- verifying that the patch is applicable to the environment, by evaluating it against the hardware/software inventory.

B.5.4 Impact, criticality and risk assessment

If the patch is considered applicable to the environment, it is necessary to determine the risks of applying or disregarding the installation of the patch. This requires a consideration of the impact to the production environment, the importance of the systems affected and the criticality of the vulnerability affected.

The following questions should be considered in the asset owner risk assessment.

a) What is the release date of the patch?

b) How many devices are affected?

c) Does the patch require the device or application to be rebooted, shutdown or cause a service disruption?

d) What is the importance of the devices affected relative to the business priorities (for example, cost to operations and effect on customers)?

e) What is the duration of the outage required, if any? Can it be reduced or eliminated with a redundant system design?

f) What is the criticality or exploitability of the security vulnerability to be mitigated by installing the patch? (For example, remote exploit and escalation of privilege)

g) Is there a high likelihood of compromise? Consider factors such as: if exploit code is readily available, the difficulty of attack, knowledge required by the attacker and countermeasures already in place to impede its effectiveness.

h) What is the difficulty of the installation and success rate? Is it a routine or non-standard process used for patch installation?

i) Is the infrastructure, methods, skills and experience already in place for testing, distribution and installation of the patch?

j) Is the device in question essential to normal operations?

The answers to the questions above will require the involvement of multiple parties, and a committee or group approach is recommended. It may require system administrators, information security, IT, control system product suppliers, contractors, engineering and operations to fully understand the situation.

The outcome of the risk assessment is an understanding of the risks, benefits and challenges that will support the decision to proceed with the patch or not. The risk assessment should also determine the priority for deployment and installation of the patch based on the severity level (for example, immediate, within weeks and never).

B.5.5 Decision for installation

The purpose of this section is to provide a suggested table of reference for the deployment and installation of patches based on their risk level to the organization. If an organization has already defined a schedule for installation of patches, they can use their own, or consider these suggestions.

Most criticality ranking systems are based on 3 or 4 levels from low to high criticality or importance. The ranking system shown in Table B.5 could be used by an asset owner, or modified for their own needs.

Table B.5 – Sample severity based patch management timeframes

Priority level	Target installation timeframe after approval of the patch by the IACS vendor
High	Within 1 week
Medium (default)	Within 3 months
Low	Within 1 year, or next available outage
None	Never

The installation timeframe serves as a starting point for an asset owner to develop their own guidelines for the installation of patches of varying criticality level.

B.6 Patch testing

B.6.1 Patch testing process

Figure B.5 illustrates a patch testing process. The next subclauses define each of the steps in the process.

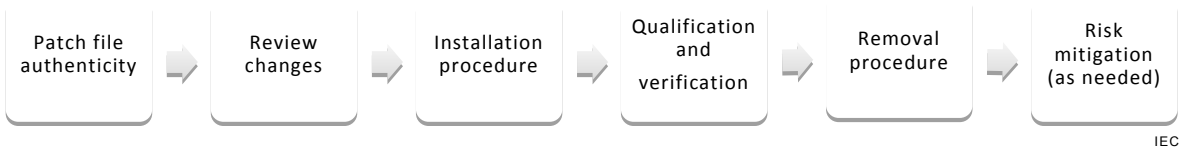


Figure B.5 – A patch testing process

B.6.2 Asset owner qualification of security patches prior to installation

Prior to installing security patches, an owner/operator should test and qualify the patches in a quality assurance, or laboratory environment, using live data feeds and interaction with other system components, operators and operating procedures if possible. This is an important step to ensure that security patches, which are intended to fix a specific problem, do not result in functional or security issues related to other system components or operating procedures.

In most cases security patches will be scheduled for a phased deployment and installation. For this reason it is important the quality assurance testing include mixed-mode testing that includes patching of those IACS devices and un-patched IACS devices in accordance with the phased patch installation plan.

B.6.3 Determining patch file authenticity

Once a patch has been evaluated and before proceeding with testing and installation, patch files should be authenticated to ensure they are from a trusted source. Although rare, there still exists the risk that a patch may be obtained from an untrusted source or may have an integrity error. Methods for checking the patch authenticity include:

- **Determining the patch source –** In most cases, patches and updates are obtained from the manufacturer or the product supplier. In those rare cases where the patch or update cannot be obtained from the manufacturer or product supplier, the source should be noted. The asset owner may choose to use a form to record the source of the patch used for testing and installation.
- **Verifying the file size –** After downloading the patch, verify its file size. Do this by comparing the downloaded patch size to the value that has been published by the same location where the patch was downloaded. The asset owner may choose to use a form to record the results of the size verification.
- **Verifying the checksum and digital signature –** After the patch is downloaded, the integrity and source of the file still has to be verified using current technology. Do this using a digital signature, a secure hash algorithm or some form of checksum, such as message digest 5 (MD5), secure hash algorithm 1 (SHA1). One or more of such integrity checks may be available from the same location where the patch was downloaded. Such verification techniques ensure the patch or update has not been modified since the signature was applied or since the checksum was calculated.

  NOTE For information on how to check the digital signature or checksum of the patch, contact the source of the patch (the product supplier or manufacturer, for example).

- **Scanning the patch for viruses –** Scan all patches using an antivirus client with the most recent virus definitions. This helps to ensure a patch file is safe and has not been compromised with malware.

B.6.4 Review functional and security changes from patches

An asset owner should consider the functionality changes to their environment as a result of installing the patch. Historically, this is the most common type of testing performed which attempts to validate that the patch does not negatively affect the functionality, operability or reliability of the IACS devices. The following should be tested:

- effects on system performance;
- effects on system reliability;
- effect on redundancy or fault-tolerance capabilities;
- ability to be installed while the affected IACS component is operational (if possible);
- removal of functionality previously relied upon;
- required staff with special skills to be present at time of installation; and
- ability to roll back the patch in the event of unforeseen effects.

An asset owner should consider the security changes as a result of any patch. This identifies where a patch may negatively affect the security controls or vulnerabilities of the IACS device. The following should be tested:

- new users created by the patch;
- access controls or privileges of existing users that the patch modifies;
- new ports, services or modified states of an existing service that the patch opens or creates;
- disabled or modified existing security controls;
- disabled or closed ports and services;
- new and previously unavailable security capabilities;
- new or exposed vulnerabilities when the IACS is scanned with a vulnerability scanner; and
- alternative methods or countermeasures to mitigate the need for patching.

The asset owner may choose to compare the device configuration before and after the patch to identify the changes above. The asset owner may also choose to use security vulnerability assessment software.

B.6.5 Installation procedure

a) Review platform instructions and technical notes:

The IACS component may be built upon commercial-off-the-shelf (COTS) hardware, software and technology. In the context of IACS patching, the platform refers to the underlying OS upon which it operates.

The platform product suppliers regularly provide updates and include instructions for their installation and other technical guidance. The asset owner should review those instructions and technical notes prepared by the platform product supplier associated with each patch to identify any special procedures, requirements, limitations, etc.

b) Review IACS product supplier installation instructions and technical notes:

The asset owner should review any installation instructions and technical notes for the patches any supplier has released.

In cases where the IACS solution is installed on top of a platform such as Microsoft Windows, the IACS product supplier may have additional instructions and technical notes. The product supplier may require a special procedure to be used for the reliable installation of the patch, or provide other technical guidance.

c) Identify prerequisites:

The asset owner should identify any prerequisites that are required before the patch can be installed from the IACS vendor supplied patch information. This may include a minimum product version, service pack, other patches, free disk space, free memory, etc. In the cases where a service pack is required, this should be handled like any other patch and subjected to the same evaluation and testing.

d) Identify target devices and appropriate test samples:

The asset owner should clearly identify those devices that require the security update. The objectives of this step are to ensure that all applicable devices are identified, an appropriate test environment and/or test plan is prepared to ensure predictable results and there is an accurate representation of the production environment for testing, in order to identify potential issues and prepare workarounds ahead of time.

If the test environment or methods used do not reflect the production environment, the results may not identify compatibility or troubleshooting issues when the patch is installed on the most critical IACS components. The end result is inconsistency and unpredictable behavior when the patch is installed on the production devices.

The asset owner should select the most appropriate testing approach above based on the infrastructure and resources available.

e) Patch installation to test environment:

The objective is to have a well-documented patch installation procedure that includes any necessary workarounds that may be required when the patch is installed in the production environment. The asset owner should install the patch in the test environment preferably in the same manner and with the same procedure that will be used in the production environment. Any instructions, methods and technical notes provided by the IACS product supplier should be followed.

Methods for patch installation include:

- executing individual patch installation files on individual devices;
- packaging two or more patches together through special utilities or scripts or service packs to increase installation automation;
- executing the IACS product supplier's update application, which may include special tools and utilities;
- using automated patch deployment and installation tools that manage multiple devices; and
- upgrading or replacing the current firmware/Operating System/runtime with a newer version.

The personnel performing the patch installation should monitor the performance impacts to the device, the changes performed, installation logs and its successful completion. All notes should be retained such that they can be used for patch installation to the production environment.

The asset owner may choose to assign responsibility for patch installation testing to an organizational group who should document the appropriate installation procedure.

B.6.6 Patch qualification and validation

The objective of patch qualification is to build confidence with technical validation that the patch will not negatively affect the performance, safety or reliability of the IACS.

The asset owner may choose to rely on iterative testing phases by various groups as part of their qualification process. This may include:

- approval from IACS product supplier after they have certified patch compatibility;
- successful installation and testing of the patch on a non-IACS environment (for example, business network) with similar hardware and software;
- successful installation and testing on a central test environment;
- successful installation and testing on a control group;
- successful installation and testing to devices of low regulatory impact, non-critical devices and standby devices; and
- successful installation and testing to devices of high regulatory or safety impact.

The asset owner has the discretion to choose the testing phases appropriate to the criticality of the IACS, safety risks and other risks to the organization before installation on the most critical of production devices.

B.6.7 Patch removal, roll back, restoration procedures

To mitigate potential reliability and operability risks that may occur after the installation of the patch, the asset owner should prepare and validate procedures for the removal of the change. There always exists the potential need to revert any changes if unwanted system behavior occurs and the prior trusted configuration is required.

Prior to any patch installation a full system backup should be performed, eliminating questions about the age of the last backup. Selectively removing newly applied patches may be the fastest way to get a process back into production. If selective rollback is not available then the full system backup can be used to restore the system to its previous state,

If time allows, patches should be installed and tested in groups. Consideration should be given to group patches that have defined rollback procedures versus those that do not. The company can then test those patches that may be easier, or use the same methods, to rollback, before committing to those patches that will take more work.

This technical report only provides the minimum guidance on restoration methods. Asset owners are advised to coordinate with the product suppliers for the best methods and procedures.

B.6.8 Risk mitigation alternatives

Risk mitigation should be performed in situations where the patch cannot be installed due to: the patch not existing yet, an appropriate time to install the patch not being available yet, incompatibility, reliability issues, performance issues, lack of product supplier support or when pre-requisites are not available, such as free memory, service pack level, proper version of the OS, etc.

Although this Technical Report is primarily concerned with the management of patches, it is important to note that when a new vulnerability is discovered, it may take weeks or months until a patch for that vulnerability is developed by the vendor, and potentially additional weeks or months until the vulnerable asset is in a state that will allow the new patch to be installed. During this “window of vulnerability,” after the vulnerability has been discovered but before the patch can be installed, it is advisable to take mitigating actions such as those described in this clause. (See also B.8.4.)

In these cases, the following options are available for mitigating the security risks if the patch is not installed and the security vulnerability remains:

- reconfiguring a product;
- removing or disabling the feature or component which is vulnerable and requires the patch;
- removing affected software;
- disabling the startup of the service which is vulnerable and requires the patch;
- implementing network filtering controls to prevent access to the vulnerable service, such as implementing a host-based or network-based firewall;
- implementing intrusion prevention system (IPS) rules and signatures to block malicious packets and network attacks to the vulnerable service;
- implementing access controls to the programs and executables to ensure only authorized personnel may use the vulnerable program;
- implementing security policies with technical supporting solutions to prevent the introduction of malicious software into the IACS, including requirements to use antivirus software, scanning portable memory, etc.;
- investigating the removal and replacement of the vulnerable device(s) based on the cost and business justification; and
- deploying secure outbound-only gateways with strong access controls that insulate vulnerable systems and devices from being accessed by devices that might contain malware.
- isolating the system by disconnecting the system from the corporate network provides a short term solution. This does not guard against all compromises, but it does provide time to implement long term solutions.
- fixing the problem by updating the IACS to a newer supported version. This is not always possible when the applications are no longer available or there is no upgrade path. When a fix is not possible, then enhancing is the next best choice.

- enhancing the IACS to new versions with enhanced user functionality. Many users of older systems can justify the expense of adding new functionality at the same time the system is updated. Enhancements are often possible when there is an application upgrade path. If no upgrade path exists, then the choices are to abandon the system or to replace it.
- abandoning or retiring the IACS. While the application was probably essential when it was initially installed, often other newer applications duplicate the older application's functionality, but do not have the functions turned on. The older applications were kept in place because it was less painful to keep them going than to eliminate them and use the alternative. Now with a possible increased risk of compromise, it is better to switch to supportable applications.
- replacing the IACS with a new supported IACS. This is usually the most time consuming alternative, but when all else fails, it may be the least risky choice. If the application is critical, then there is economic justification for replacement. If the application is not critical, then it may have to be abandoned, because of the risk and impact of a compromise.

In those industries that are required by regulations to follow cyber security standards, those asset owners are also usually required to document those mitigating controls put in place in lieu of installing patches.

In some cases, the use of mitigating controls is easier than patch installation. Some benefits of mitigating controls are:

- vulnerabilities may be mitigated before the patch is released or approved by the IACS product supplier independent of product development by the product supplier or hardware/software manufacturer;
- less impact on product functionality;
- less testing by asset owner may be required, allowing faster mitigation of the vulnerability;
- lower resistance to implementation by asset owners as it may not require devices to be rebooted or the process to be shutdown; and
- support of legacy products for which vulnerabilities may continue to be discovered, but which are no longer patched or supported by the original product supplier.

Lastly, any mitigating controls that are implemented should only be valid for a specific time frame (such as, one year) before they are re-evaluated, to ensure they are still acceptable for another term.

B.7 Patch deployment and installation

B.7.1 Patch deployment and installation process

Figure B.6 illustrates a patch deployment and installation process.

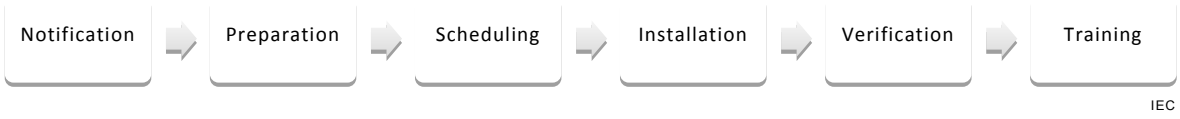


Figure B.6 – A patch deployment and installation process

B.7.2 Notification of affected parties

Depending on the size and structure of the asset owner's organization, maintenance contracts and warranties with the vendor, the installation of patches may be handled by the same group responsible for testing, performed by specific personnel at each facility or outsourced to an integrator or IACS product supplier.