

BS ISO/IEC 18180:2013



BSI Standards Publication

Information technology — Specification for the Extensible Configuration Checklist Description Format (XCCDF) Version 1.2

bsi.

...making excellence a habit.
AVA Documents™
Document Sharing Hub

National foreword

This British Standard is the UK implementation of ISO/IEC 18180:2013.

The UK participation in its preparation was entrusted to Technical Committee ICT/-/1, Information systems co-ordination.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2013. Published by BSI Standards Limited 2013

ISBN 978 0 580 78673 0

ICS 35.040

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 July 2013.

Amendments issued since publication

Date	Text affected
------	---------------

INTERNATIONAL STANDARD

BS ISO/IEC 18180:2013
ISO/IEC
18180

First edition
2013-06-15

Information technology — Specification for the Extensible Configuration Checklist Description Format (XCCDF) Version 1.2

*Technologies de l'information — Spécification de XCCDF (Extensible
Configuration Checklist Description Format) version 1.2*



Reference number
ISO/IEC 18180:2013(E)
Documents
AVA Document Sharing Hub
© ISO/IEC 2013



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2013

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 18180 was prepared by the U.S. National Institute of Standards and Technology (as NIST IR 7275, Revision 4) and was adopted, under a special “fast-track procedure”, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, in parallel with its approval by the national bodies of ISO and IEC.



U.S. Department of Commerce

NIST Interagency Report 7275
Revision 4

Specification for the Extensible Configuration Checklist Description Format (XCCDF) Version 1.2

David Waltermire
Charles Schmidt
Karen Scarfone
Neal Ziring

NIST Interagency Report 7275
Revision 4

Specification for the Extensible
Configuration Checklist Description
Format (XCCDF) Version 1.2

David Waltermire
Charles Schmidt
Karen Scarfone
Neal Ziring

C O M P U T E R S E C U R I T Y

Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8930

September 2011



U.S. Department of Commerce

Rebecca M. Blank, Acting Secretary

National Institute of Standards and Technology

**Patrick D. Gallagher, Under Secretary for
Standards and Technology and Director**

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analysis to advance the development and productive use of information technology. ITL's responsibilities include the development of technical, physical, administrative, and management standards and guidelines for the cost-effective security and privacy of sensitive unclassified information in Federal computer systems. This Interagency Report discusses ITL's research, guidance, and outreach efforts in computer security and its collaborative activities with industry, government, and academic organizations.

National Institute of Standards and Technology Interagency Report 7275 Revision 4
80 pages (Sep. 2011)

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

Acknowledgments

The authors of this report, David Waltermire of the National Institute of Standards and Technology (NIST), Charles Schmidt of The MITRE Corporation, Karen Scarfone of Scarfone Cybersecurity, and Neal Ziring of the National Security Agency (NSA), wish to thank all contributors to this revision of the publication, particularly Adam Halbardier of Booz Allen Hamilton, Vladimir Giszpenc, Kent Landfield and Richard Whitehurst of McAfee, Lisa Nordman of The MITRE Corporation, Joe Wolfkiel of DISA, and Shane Shaffer and Matt Kerr of G2, Inc.

The authors would also like to acknowledge the following individuals who contributed to the initial definition and development of the Extensible Configuration Checklist Description Format (XCCDF): David Proulx, Mike Michnikov, Andrew Buttner, Todd Wittbold, Adam Compton, George Jones, Chris Calabrese, John Banghart, Murugiah Souppaya, John Wack, Trent Pitsenbarger, and Robert Stafford. Stephen D. Quinn, Peter Mell, and Matthew Wojcik contributed to Revisions 1, 2, and 3 of this report. Ryan Wilson of Georgia Institute of Technology also made substantial contributions. Thanks also go to the Defense Information Systems Agency (DISA) Field Security Office (FSO) Vulnerability Management System (VMS)/Gold Disk team for extensive review and many suggestions.

Abstract

This report specifies the data model and Extensible Markup Language (XML) representation for the Extensible Configuration Checklist Description Format (XCCDF) Version 1.2. An XCCDF document is a structured collection of security configuration rules for some set of target systems. The XCCDF specification is designed to support information interchange, document generation, organizational and situational tailoring, automated compliance testing, and scoring. The specification also defines a data model and format for storing results of security guidance or checklist testing. The intent of XCCDF is to provide a uniform foundation for expression of security checklists and other configuration guidance, and thereby foster more widespread application of good security practices.

Audience

The primary audience of the XCCDF specification is government and industry security analysts, and security management product developers.

Trademark Information

All names are registered trademarks or trademarks of their respective companies.

Contents

1.	INTRODUCTION	1
1.1	PURPOSE AND SCOPE	1
1.2	DOCUMENT STRUCTURE	1
1.3	DOCUMENT CONVENTIONS	1
2.	NORMATIVE REFERENCES.....	2
3.	TERMS, DEFINITIONS, AND ABBREVIATIONS	3
3.1	XCCDF TERMINOLOGY	3
3.2	ACRONYMS AND ABBREVIATIONS	3
4.	CONFORMANCE	4
4.1	PRODUCT CONFORMANCE	4
4.2	BENCHMARK DOCUMENT CONFORMANCE	4
5.	XCCDF OVERVIEW.....	5
5.1	INTRODUCTION.....	5
5.2	CHECKLIST STRUCTURE AND TAILORING	6
5.3	TEST RESULTS.....	7
6.	XCCDF DATA MODEL	8
6.1	INTRODUCTION.....	8
6.2	GENERAL XML INFORMATION	9
6.2.1	<i>XCCDF Namespace and XML Schema</i>	<i>9</i>
6.2.2	<i>Element and Attribute Formatting.....</i>	<i>9</i>
6.2.3	<i>Element Identifiers</i>	<i>10</i>
6.2.4	<i><xccdf:metadata> Element.....</i>	<i>10</i>
6.2.5	<i>Platform Names</i>	<i>11</i>
6.2.6	<i><xccdf:reference> Element.....</i>	<i>12</i>
6.2.7	<i><xccdf:signature> Element.....</i>	<i>13</i>
6.2.8	<i>Status Tracking</i>	<i>13</i>
6.2.9	<i>Text Substitution</i>	<i>13</i>
6.2.10	<i>@xml:lang Attribute</i>	<i>14</i>
6.3	<i><XCCDF:BENCHMARK>.....</i>	<i>15</i>
6.3.1	<i>Basics</i>	<i>15</i>
6.3.2	<i>Properties.....</i>	<i>16</i>
6.4	<i>ITEM ELEMENTS.....</i>	<i>18</i>
6.4.1	<i>Properties.....</i>	<i>18</i>
6.4.2	<i><xccdf:warning> Element.....</i>	<i>20</i>
6.4.3	<i><xccdf:Group> Element.....</i>	<i>21</i>
6.4.4	<i><xccdf:Rule> Element.....</i>	<i>21</i>
6.4.5	<i><xccdf:Value> Element.....</i>	<i>30</i>
6.5	<i><XCCDF:PROFILE> ELEMENT.....</i>	<i>34</i>
6.5.1	<i>Basics</i>	<i>34</i>
6.5.2	<i>Properties.....</i>	<i>34</i>
6.5.3	<i>Selectors.....</i>	<i>35</i>
6.6	<i><XCCDF:TESTRESULT> ELEMENT</i>	<i>37</i>
6.6.1	<i>Basics</i>	<i>37</i>
6.6.2	<i>Properties.....</i>	<i>38</i>

6.6.3	<xccdf:fact> Element.....	41
6.6.4	<xccdf:rule-result> Element	41
6.6.5	<xccdf:tailoring-file> Element	44
6.7	<XCCDF:TAILORING> ELEMENT	45
6.7.1	Basics	45
6.7.2	Properties.....	45
6.7.3	Profile Shadowing	46
6.7.4	Tailoring Actions and Profile Selectors.....	47
7.	XCCDF PROCESSING	48
7.1	INTRODUCTION	48
7.2	LOADING AND TRAVERSAL.....	48
7.2.1	Introduction	48
7.2.2	Loading	48
7.2.3	Traversal	51
7.3	ASSESSMENT OUTPUTS	63
7.3.1	Overview	63
7.3.2	Scoring Models.....	63
APPENDIX A—	CONVERTING XCCDF 1.1.4 CONTENT TO XCCDF 1.2.....	66
A.1	CHANGES TO THE XCCDF XML NAMESPACE.....	66
A.2	CONVERSION OF IDENTIFIERS.....	66
A.3	CONVERSION OF <XCCDF:SUB> ELEMENTS	66
A.4	PROPERTIES REMOVED OR DEPRECATED SINCE XCCDF 1.1.4	67
APPENDIX B—	CHANGE LOG.....	68

Tables

TABLE 1: CONVENTIONAL XML MAPPINGS	1
TABLE 2: RECOMMENDED CLASS VALUES	9
TABLE 3: ELEMENT IDENTIFIER FORMAT CONVENTIONS	10
TABLE 4: <XCCDF:BENCHMARK> ELEMENT PROPERTIES	16
TABLE 5: ITEM ELEMENT PROPERTIES	18
TABLE 6: PROPERTIES SPECIFIC TO <XCCDF:GROUP> AND <XCCDF:RULE> ELEMENTS	20
TABLE 7: <XCCDF:WARNING> ELEMENT @CATEGORY ATTRIBUTE VALUES	21
TABLE 8: <XCCDF:GROUP> ELEMENT PROPERTIES	21
TABLE 9: <XCCDF:RULE> ELEMENT PROPERTIES	23
TABLE 10: ASSIGNED VALUES FOR THE @SYSTEM ATTRIBUTE OF AN <XCCDF:IDENT> ELEMENT	24
TABLE 11: <XCCDF:CHECK> ELEMENT PROPERTIES.....	25
TABLE 12: TRUTH TABLE FOR AND	27
TABLE 13: TRUTH TABLE FOR OR.....	27
TABLE 14: TRUTH TABLE FOR NEGATION	27
TABLE 15: POSSIBLE PROPERTIES FOR <XCCDF:FIXTEXT> ELEMENT.....	28
TABLE 16: POSSIBLE PROPERTIES FOR <XCCDF:FIX> ELEMENT	29

TABLE 17: PREDEFINED VALUES FOR @SYSTEM ATTRIBUTE OF <XCCDF:FIX> ELEMENT	30
TABLE 18: <XCCDF:VALUE> ELEMENT PROPERTIES.....	31
TABLE 19: POSSIBLE PROPERTIES FOR <XCCDF:CHOICES> ELEMENT.....	32
TABLE 20: PERMITTED OPERATORS BY VALUE TYPE.....	33
TABLE 21: <XCCDF:PROFILE> ELEMENT PROPERTIES	35
TABLE 22: SELECTORS	36
TABLE 23: <XCCDF:TESTRESULT> ELEMENT PROPERTIES	39
TABLE 24: PREDEFINED @NAME ATTRIBUTE VALUES FOR <XCCDF:FACT> ELEMENTS.....	41
TABLE 25: <XCCDF:RULE-RESULT> ELEMENT PROPERTIES	42
TABLE 26: POSSIBLE RESULTS FOR A SINGLE TEST	43
TABLE 27: <XCCDF:OVERRIDE> ELEMENT PROPERTIES	44
TABLE 28: <XCCDF:TAILORING-FILE> ELEMENT PROPERTIES	45
TABLE 29: <XCCDF:TAILORING> ELEMENT PROPERTIES.....	46
TABLE 30: PROFILE SHADOWING BEHAVIOR	47
TABLE 31: TAILORING ACTIONS AND PROFILE SELECTORS	47
TABLE 32: LOADING PROCESSING SEQUENCE SUB-STEPS.....	48
TABLE 33: INHERITANCE PROCESSING MODEL	50
TABLE 34: BENCHMARK PROCESSING ALGORITHM SUB-STEPS	51
TABLE 35: ITEM PROCESSING ALGORITHM SUB-STEPS.....	52
TABLE 36: PROFILE SELECTOR EXAMPLE: INITIAL CONFIGURATION	56
TABLE 37: PROFILE SELECTOR EXAMPLE: INITIAL BENCHMARK STATE	56
TABLE 38: PROFILE SELECTOR EXAMPLE: FINAL BENCHMARK STATE.....	58
TABLE 39: CHECK PROCESSING ALGORITHM SUB-STEPS	59
TABLE 40: DEFAULT MODEL ALGORITHM SUB-STEPS	64
TABLE 41: FLAT MODEL ALGORITHM SUB-STEPS.....	65
TABLE 42: ALTERNATIVE OPERATIONS FOR REMOVED AND DEPRECATED XCCDF 1.1.4 CONSTRUCTS	67
TABLE 43: MAPPING PREVIOUS RELEASE SECTIONS TO THIS RELEASE.....	69

Figures

FIGURE 1: TYPICAL STRUCTURE OF A BENCHMARK.....	15
FIGURE 2: CHECK PROCESSING FLOWCHART (WHEN THE CHECK'S PARENT IS AN <XCCDF:RULE>).....	60
FIGURE 3: WORKFLOW FOR ASSESSING BENCHMARK COMPLIANCE.....	63

1. Introduction

1.1 Purpose and Scope

This report defines the specification for the Extensible Configuration Checklist Description Format (XCCDF) version 1.2. The report also defines and explains the requirements that XCCDF 1.2 documents and products (i.e., software) must meet to claim conformance with the specification. This report only applies to XCCDF version 1.2. All other versions are outside the scope of this report.

1.2 Document Structure

The remainder of this report is composed of the following sections and appendices:

- Section 2 provides a list of normative references for the report.
- Section 3 defines selected terms and abbreviations used in the report.
- Section 4 provides the high-level requirements for claiming conformance with the XCCDF version 1.2 specification.
- Section 5 gives an overview of XCCDF and its capabilities.
- Section 6 provides an introduction to the XCCDF data model and details additional requirements and recommendations for XCCDF's use.
- Section 7 discusses XCCDF processing requirements and recommendations.
- Appendix A explains how to convert XCCDF 1.1.4-specific properties to their XCCDF 1.2 counterparts.
- Appendix B provides a change log that documents significant changes to released drafts of this specification. This includes a section-by-section mapping of how the document was reorganized from the previous drafts to this draft. Readers who are familiar with any previous XCCDF versions may find it helpful to review Appendix B first before the rest of the document.

1.3 Document Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in Request for Comment (RFC) 2119 [RFC2119].

Namespace prefixes used in this specification are listed in Table 1.

Table 1: Conventional XML Mappings

Prefix	Namespace	Schema
cpe2	http://cpe.mitre.org/language/2.0	Common Platform Enumeration (CPE) 2.3 Applicability Language
cpe2-dict	http://cpe.mitre.org/dictionary/2.0	CPE 2.3 Dictionary
dc	http://purl.org/dc/elements/1.1/	Simple Dublin Core elements
dsig	http://www.w3.org/2000/09/xmldsig#	Interoperable XML digital signatures
xccdf	http://checklists.nist.gov/xccdf/1.2	XCCDF policy documents
xml	http://www.w3.org/XML/1998/namespace	Common XML attributes
xsd	http://www.w3.org/2001/XMLSchema	XML Schema
xsi	http://www.w3.org/2001/XMLSchema-Instance	XML Schema Instance

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.

[DCES], DCMI (Dublin Core Metadata Initiative), *Dublin Core Metadata Element Set, Version 1.1*, October 2010, available at <<http://dublincore.org/documents/dces/>>

[DCXML], DCMI, *Guidelines for Implementing Dublin Core in XML*, April 2003, available at <<http://dublincore.org/documents/dc-xml-guidelines/>>

[ILSR], IANA, *IANA Language Subtag Registry (ILSR)*, available at <<http://www.iana.org/assignments/language-subtag-registry>>

[IR7693], NIST, NIST IR 7693, *Specification for Asset Identification 1.1*, June 2011, available at <<http://csrc.nist.gov/publications/PubsNISTIRs.html>>

[IR7695], NIST, NIST IR 7695, *Common Platform Enumeration: Naming Specification Version 2.3*, August 2011, available at <<http://csrc.nist.gov/publications/PubsNISTIRs.html>>

[IR7698], NIST, NIST IR 7698, *Common Platform Enumeration: Applicability Language Specification Version 2.3*, August 2011, available at <<http://csrc.nist.gov/publications/PubsNISTIRs.html>>

[PCRE], Perl Compatible Regular Expressions (PCRE), available at <<http://www.pcre.org>>

[RFC2119], IETF, RFC 2119, *Key words for use in RFCs to Indicate Requirement Levels*, March 1997, available at <<http://www.ietf.org/rfc/rfc2119.txt>>

[RFC5646], IETF, RFC 5646, *Tags for Identifying Languages*, September 2009, available at <<http://www.ietf.org/rfc/rfc5646.txt>>

[UNICODE], Unicode Technical Recommendation No. 18, *Unicode Regular Expressions*, version 9, January 2004, available at <<http://unicode.org/reports/tr18/>>

[XHTML], W3C (World Wide Web Consortium), *XHTML Basic*, December 2000, available at <<http://www.w3.org/TR/2000/REC-xhtml-basic-20001219/>>

[XINCLUDE], W3C, *XML Inclusions (XInclude) Version 1.0 (Second Edition)*, November 2006, available at <<http://www.w3.org/TR/xinclude/>>

[XMLDSIG], W3C, *XML Signature Syntax and Processing (Second Edition)*, June 2008, available at <<http://www.w3.org/TR/xmlsig-core/>>

[XMLNAME], W3C, *Namespaces in XML 1.0 (Third Edition)*, December 2009, available at <<http://www.w3.org/TR/REC-xml-names/>>

[XMLSCHEMA], W3C, *XML Schema Part 2: Datatypes Second Edition*, October 2004, available at <<http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>>

[XPATH], W3C, *XML Path Language (XPath) Version 1.0*, November 1999, available at <<http://www.w3.org/TR/xpath/>>

3. Terms, Definitions, and Abbreviations

For the purposes of this document, the following terms, definitions, and abbreviations apply.

3.1 XCCDF Terminology

Benchmark: The root node of an XCCDF benchmark document; may also be the root node of an XCCDF results document (the results of evaluating the XCCDF benchmark document).

Benchmark Consumer: A product that accepts an existing XCCDF benchmark document, processes it, and produces an XCCDF results document.

Benchmark Producer: A product that generates XCCDF benchmark documents.

Checklist: An organized collection of rules about a particular kind of system or platform.

Group: An item that can hold other items; allows an author to collect related items into a common structure and provide descriptive text and references about them.

Item: A named constituent of a benchmark. The three types of items are groups, rules, and values.

Profile: A named tailoring of a benchmark.

Rule: An element that holds check references and may also hold remediation information.

Tailoring: An element that specifies profiles to modify the behavior of a benchmark; the top-level element of a tailoring document.

TestResult: The container for XCCDF results. May be the root node of an XCCDF results document.

Value: A named data value that can be substituted into other items' properties or into checks.

3.2 Acronyms and Abbreviations

CCE	Common Configuration Enumeration
CPE	Common Platform Enumeration
CVE	Common Vulnerabilities and Exposures
DCMI	Dublin Core Metadata Initiative
DNS	Domain Name System
IANA	Internet Assigned Numbers Authority
IR	Interagency Report
NIST	National Institute of Standards and Technology
OCIL	Open Checklist Interactive Language
OVAL	Open Vulnerability and Assessment Language
PCRE	Perl Compatible Regular Expression
RFC	Request for Comments
SCAP	Security Content Automation Protocol
SP	Special Publication
W3C	World Wide Web Consortium
XCCDF	Extensible Configuration Checklist Description Format
XHTML	Extensible Hypertext Markup Language
XML	Extensible Markup Language

4. Conformance

Products and organizations may want to claim conformance with this specification for a variety of reasons. For example, a software vendor may want to assert that its product generates and/or processes XCCDF benchmark documents properly. Another example is a policy mandating that an organization use XCCDF for documenting and executing its security configuration checklists.

This section provides the high-level requirements that a product or benchmark document **MUST** meet for conformance with this specification. Most of the requirements listed in this section reference other sections in the report that fully define the requirements.

Other specifications that use XCCDF **MAY** define additional requirements and recommendations for XCCDF's use. Such requirements and recommendations are outside the scope of this publication.

4.1 Product Conformance

There are two types of XCCDF products: benchmark producers and benchmark consumers. *Benchmark producers* are products that generate XCCDF benchmark documents, while *benchmark consumers* are products that accept an existing XCCDF benchmark document, process it, and produce an XCCDF results document. Products claiming conformance with this specification **SHALL** adhere to the following requirements:

1. For benchmark producers, generate well-formed XCCDF benchmark documents. This includes following the benchmark document requirements specified in Section 4.2 and all of the pertinent processes defined in Sections 6 and 7.
2. For benchmark consumers, consume and process well-formed XCCDF benchmark documents, and generate well-formed XCCDF results documents. This includes following all of the pertinent processes defined in Sections 6 and 7.
3. Make an explicit claim of conformance to this specification in any documentation provided to end users.

4.2 Benchmark Document Conformance

XCCDF benchmark documents claiming conformance with this specification **SHALL** follow these requirements:

1. Adhere to the official XCCDF schema as explained in Section 6.
2. Adhere to the syntax, structural, and other XCCDF benchmark document requirements defined in Sections 6 and 7.

5. XCCDF Overview

5.1 Introduction

XCCDF was created to document technical and non-technical security checklists using a standardized format. The general objective is to allow security analysts and IT experts to create effective, interoperable automated checklists, and to support the use of these checklists with a wide variety of tools. A checklist is an organized collection of rules about a particular kind of system or platform. Automation is necessary for consistent and rapid verification of system security because of the sheer number of things to check and the number of hosts within an organization that need to be assessed (often many thousands).

XCCDF enables easier, more uniform creation of security checklists, which in turn helps to improve system security by more consistent and accurate application of sound security practices. Adoption of XCCDF lets security professionals, security tool vendors, and system auditors exchange information more quickly and precisely, and also permits greater automation of security testing and configuration assessment. Additional capabilities provided by XCCDF include the following:

- Ensure compliance to multiple policies (systems subject to the Federal Information Security Management Act [FISMA], Security Technical Implementation Guide [STIG], Health Insurance Portability and Accountability Act [HIPAA], etc.)
- Permit faster, more cooperative, and more automated definition of security rules, procedures, guidance documents, alerts, advisories, and remediation measures
- Permit fast, uniform, manageable administration of security checks and audits
- Permit composition of security rules and tests from different community groups and vendors
- Facilitate scoring, reporting, and tracking of security status and checklist conformance for systems

The XCCDF specification, which is vendor-neutral, is suited for a wide variety of checklist applications. XCCDF has an open, standardized format, amenable to generation by and editing with a variety of tools. In addition, because it is expressed using XML, an XCCDF document is embeddable inside other documents. XCCDF also includes provisions for incorporating other data formats, and it is extensible to include new functionality, features, and data stores without hindering the functionality of existing XCCDF tools.

Since XCCDF's creation, various commercial, government, and community developers have created tools that support XCCDF, allowing a single XCCDF checklist to be used by many organizations and many tools. These tools read an XCCDF checklist and follow it to perform the necessary checks and ask the necessary questions to measure conformance with the checklist and generate corresponding reports.

A common use case for an XCCDF checklist is normalizing security configuration content through automated tools. Such tools accept one or more XCCDF checklists along with supporting system test definitions, and determine whether the specified rules are satisfied by a target system. The XCCDF checklist supports generation of a report, including a weighted score. XCCDF checklists can also be used to test whether or not a system is vulnerable to a particular kind of attack. For this purpose, the XCCDF checklist plays the role of a vulnerability alert, but with the ability to describe the problem, drive automated verification of its presence, and convey recommendations for corrective actions.

The scenarios below illustrate some uses of XCCDF security checklists and tools.

- Scenario 1 – An industry consortium, in conjunction with a product vendor, wants to produce a security checklist for an application server. The core security settings are the same for all OS platforms on which the server runs, but a few settings are OS-specific. The consortium crafts one checklist for the core settings and writes several OS-specific ones that supplement the core settings. Users download the core checklist and the OS-specific checklists that apply to their installations, and then run an assessment tool to score their compliance with the checklists.
- Scenario 2 – An academic group produces a checklist for secure configuration of a particular server operating system version. A government agency issues a set of rules extending the academic checklist to meet more stringent user authorization criteria imposed by statute. A medical enterprise downloads both the academic checklist and the government extension, tailors them to fit their internal security policy, and uses them for an enterprise-wide audit using a commercial security audit tool. Reports outputted by the tool include remediation measures which the IT staff can use to bring their systems into full internal policy compliance. (Note that remediation processes should be carefully planned and implemented.)

These scenarios demonstrate some of XCCDF's range of capabilities. XCCDF can represent complex conditions and relationships about the systems to be assessed, and it can incorporate descriptive material and remediative measures. It is also designed to be modular; for example, XCCDF benchmarks acquire programmatically ascertainable information through lower-level check system languages.

5.2 Checklist Structure and Tailoring

The basic unit of structure for a checklist is a rule. A rule simply describes a state or condition which the target of the document should exhibit. A simple checklist might consist of a list of rules, but richer ones require additional structure. XCCDF allows checklist authors to impose organization within the checklist, such as putting related rules into named groups and designating the order for processing rules and groups.

Checklist users can employ tailoring tools to customize a checklist's rules for their local environment or policies. For example, an auditor might need to set the password policy requirement to be more stringent than the default recommendation. Another example is that an organization may have trouble applying particular settings because of legacy systems or conflicts with other software. In cases such as these, the checklist users may need to tailor the checklist. The following customization options are available:

- **Selectability** – A tailoring action selects or deselects a rule or group of rules. For example, an entire group of rules that relate to physical security might not apply to a network scan, so that group could be deselected. In the case of NIST Special Publication (SP) 800-53, certain rules apply according to the impact rating of the system. For example, systems that have an impact rating of *low* might not have all of the same access control requirements as a system with a *high* impact rating, so the rules that are not applicable for the *low* system can be deselected.
- **Value Modification** – A tailoring action substitutes a locally-significant value for a general value in an XCCDF variable (`<xccdf:Value>`). This locally-significant value then gets used wherever the variable is referenced. For example, at a site where all logs are sent to a single host, the address of that log server could be substituted into an audit configuration variable. Using the NIST SP 800-53 example, a system with a *moderate* impact rating might require a 12-character password, whereas a system with a *low* impact rating might only require an 8-character password.
- **Property Modification** – A tailoring action modifies a property for an element not addressed by selectability or value modification. For example, an author could alter the relative weight of particular rules or groups of rules.

XCCDF 1.2 supports the creation and use of tailoring documents, which define tailoring profiles available for use with a particular benchmark document. Having a tailoring document allows sets of checklist customizations to be recorded in a consistent manner.

XCCDF allows checklists to include descriptive and interrogative text to help checklist users make tailoring decisions, even directing users through the process. Some combinations of rules within the same checklist might conflict or be mutually exclusive. To avert problems, the checklist author can identify particular tailoring choices as incompatible. Checklist authors can also designate the modes (e.g., Gold, Platinum, High Impact Rating, Level 1) under which a rule should be processed. Checklist authors should include the appropriate text in their checklists to aid in tailoring.

Another important facet of customization is extension. XCCDF supports mechanisms for authors to extend (inherit from) existing rules, values, and groups, in addition to expressing rules, values, and groups in their entirety. For example, in XCCDF checklists, it is desirable to share descriptive material among several rules, and to allow a specialized rule to be created by extending a base rule. (Note that group extension has been deprecated in XCCDF 1.2. Use of group extension is strongly discouraged because it has known problems, such as those described in Section 7.2.2, and content that uses it is unlikely to be interoperable between XCCDF products.)

5.3 Test Results

Many organizations use several security products to determine the security of IT systems and their compliance to various policies. Unfortunately, if the outputs from these products are not standardized, costly customization and integration can be required for trending, aggregation, and reporting. Addressing this, XCCDF provides a standardized reporting format for storing the results of the rule checking subsystem. Security tools sometimes include only the results of the test or tests in the form of a pass/fail status. Other tools provide additional information (e.g., instead of simply indicating that more than one privileged account exists on a system, certain tools also provide the list of privileged accounts). The following information is basic to all XCCDF results:

- The security guidance document or checklist used, along with any adaptations via customization or tailoring applied
- Information about the target system to which the test was applied, including arbitrary identification and configuration information about the target system
- The time interval of the test, and the time instant at which each individual rule was evaluated
- One or more scores
- References to lower-level details possibly stored in other output files.

6. XCCDF Data Model

6.1 Introduction

The XCCDF data model and its XML representation are intended to be platform-independent and portable to foster broad adoption and sharing of rules. The fundamental data model for XCCDF consists of the following main element data types:

- **Benchmark.** An XCCDF benchmark document holds an `<xccdf:Benchmark>` element, which acts as a container for other elements, including `<xccdf:Group>`, `<xccdf:Rule>`, `<xccdf:Value>`, `<xccdf:Profile>`, and `<xccdf:TestResult>` elements.
- **Item.** An item is a named constituent of an `<xccdf:Benchmark>`. There are three types of items:
 - **Group.** An `<xccdf:Group>` holds other items. An `<xccdf:Group>` collects related `<xccdf:Rule>` and `<xccdf:Value>` elements into a common structure and can provide descriptive text and references about them. An `<xccdf:Group>` allows benchmark users to select and deselect related `<xccdf:Rule>` elements together; since a deselected `<xccdf:Group>` is not processed, none of its contained items are processed either. Selection of an `<xccdf:Group>` allows its children to be processed normally based on their individual selection states.
 - **Rule.** An `<xccdf:Rule>` element holds check references and can also hold remediation information.
 - **Value.** An `<xccdf:Value>` element is a named data value that can be substituted into other items' properties or into checks. It can have an associated data type and metadata that express how the value should be used and how it can be tailored.
- **Profile.** An `<xccdf:Profile>` element is a named tailoring of a benchmark using a collection of attributed references to `<xccdf:Rule>`, `<xccdf:Group>`, and `<xccdf:Value>` elements. It allows definition of named levels or baselines in a benchmark (see Section 5.2).
- **TestResult.** An `<xccdf:TestResult>` element holds the results of performing a test or check against a single target device or system. An `<xccdf:TestResult>` element references `<xccdf:Rule>` and `<xccdf:Value>` elements and may also reference an `<xccdf:Profile>` element.
- **Tailoring.** A tailoring document holds exactly one `<xccdf:Tailoring>` element, which contains `<xccdf:Profile>` elements to modify the behavior of an `<xccdf:Benchmark>`.

The rest of this section explains the relationships between these main element data types and examines each of the types in more detail. Section 6.2 discusses general information that applies to most or all of the main element data types. Sections 6.3 through 6.7 cover `<xccdf:Benchmark>`, item (`<xccdf:Group>`, `<xccdf:Rule>`, `<xccdf:Value>`), `<xccdf:Profile>`, `<xccdf:TestResult>`, and `<xccdf:Tailoring>` elements, respectively.

6.2 General XML Information

6.2.1 XCCDF Namespace and XML Schema

The namespace URI for this specification SHALL be “<http://checklists.nist.gov/xccdf/1.2>”. Applications that process XCCDF SHOULD use the namespace URI to decide whether or not they can process a given document. The XML representation of XCCDF is expressed as an XML schema at <http://scap.nist.gov/specifications/xccdf/#resource-1.2>. The XML schema implementation of XCCDF SHALL be the authoritative XML binding definition for XCCDF.

6.2.2 Element and Attribute Formatting

The structure of an XCCDF document supports transformation into HTML and various XML formats to promote portability and interoperability. The XCCDF language also allows for the inclusion of content that does not contribute directly to the technical content, such as an introduction, a rationale, warnings, and external references. XCCDF also provides mechanisms to document authors for formatting text, including images, and referencing other information resources (e.g., prose publications), but these mechanisms are separable from the text itself so they can be filtered out by applications that do not support or require them.

Throughout the rest of this section, there are tables that show the possible properties of each main element in the XCCDF data model. In the tables, properties of type “identifier” MUST be strings obeying the definition of “NCName” from [XMLSCHEMA]. Properties of type “string” and “text” MUST NOT include XHTML formatting. Properties of type “HTML-enabled text” are string data that MAY include embedded formatting, presentation, and hyperlink structure; if present, these MUST be expressed using Extensible Hypertext Markup Language (XHTML) Basic tags [XHTML]. The core modules noted in [XHTML] and the Presentation module MAY be used for this.

XHTML markup allows authors to specify how the content of certain fields SHOULD be displayed to a user. However, while this allows authors to specify every detail of a field's display, authors MAY use classifiers in these fields, using class attributes in <div> or elements. These tell products the type of content contained within a text block and allow the products to display this content according to their own conventions. Authors may wish to do this so their content fits better with other automated formatting choices made by a product or a stylesheet, and also because some products may not support the complete range of HTML tags in their displays, thus causing some explicit formatting to be ignored. A list of recommended class values appears in Table 2. Authors MAY use class values not included in this list, and it is OPTIONAL for products to have special formatting for any of the listed classifiers. However, both authors and products SHOULD use these class designators when appropriate to provide more effective display of content. Use of these class attribute values SHALL be applicable wherever HTML content can be included in XCCDF documents.

Table 2: Recommended Class Values

Class Value	Meaning
license	Licensing and use information
copyright	Copyright and ownership information
tangent	A block of text that contains tangentially related information (possibly appropriate for inclusion as a sidebar or a pop-up)
warning	Pitfalls or cautions relative to the surrounding text. High-level and general warnings SHOULD appear in designated warning fields, if available.
critical	Content of critical importance to the user

Class Value	Meaning
example	An example of some kind
instructions	Special instructions to the user
default	General information. Empty or absent class attributes also imply "default" appearance. This tag allows authors to explicitly indicate that text should appear in the default format.

6.2.3 Element Identifiers

The elements listed in Table 3 have special conventions around the format of their identifiers (*@id* attribute). Authors MUST follow these conventions because they make it easier to preserve the global uniqueness of the resulting identifiers.

Table 3: Element Identifier Format Conventions

Element	Format Convention
Benchmark	<i>xccdf_namespace_benchmark_name</i>
Profile	<i>xccdf_namespace_profile_name</i>
Group	<i>xccdf_namespace_group_name</i>
Rule	<i>xccdf_namespace_rule_name</i>
Value	<i>xccdf_namespace_value_name</i>
TestResult	<i>xccdf_namespace_testresult_name</i>
Tailoring	<i>xccdf_namespace_tailoring_name</i>

In Table 3, *namespace* contains a valid reverse-DNS style string (limited to letters, numbers, periods, and the hyphen character) that is associated with the content author. Examples include "com.acme.finance" and "gov.tla". These *namespace* strings MAY have any number of parts, and benchmark consumers processing them SHALL treat them as case-insensitive. (That is, com.ABC is considered identical to com.abc.) This association with the content author is solely to ensure that different content authors will not use the same namespace value, as this could lead to identifier collisions. Readers should not automatically infer any special authority or trust of the content based on the namespace since reuse of the element or changes to referenced content may not align with the author's original intent. The *name* component in the format conventions MAY be any NCName-compliant string [XMLSCHEMA]. Identifier fields other than those noted in Table 3 have no additional formatting conventions beyond compliance with the NCName data type.

See the example `<xccdf:Profile>` in Section 6.5.1 for several examples of the element identifier format conventions.

6.2.4 <xccdf:metadata> Element

XCCDF supports inclusion of metadata about a document, including title, name of author(s), organization providing the guidance, version number, release date, update URL, and a description. This is particularly useful for facilitating the discovery and retrieval of XCCDF checklists from public repositories.

When used, the `<xccdf:metadata>` element SHALL contain document metadata expressed in XML. The metadata element can appear one or more times as a child of the `<xccdf:Benchmark>`, `<xccdf:Rule>`, `<xccdf:Group>`, `<xccdf:Value>`, `<xccdf:Profile>`, `<xccdf:TestResult>`, `<xccdf:rule-result>`, and `<xccdf:Tailoring>` elements. The `<xccdf:Benchmark>` element SHOULD contain metadata information formatted using the Dublin

Core Metadata Initiative (DCMI) Simple DC Element specification, as described in [DCES] and [DCXML]. Benchmark consumers SHOULD be prepared to process Dublin Core metadata in the `<xccdf:metadata>` element. An example of the Dublin Core format is shown below.

```
<xccdf:metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
  <dc:title>Security Benchmark for Ethernet Hubs</dc:title>
  <dc:creator>James Smith</dc:creator>
  <dc:publisher>Center for Internet Security</dc:publisher>
  <dc:subject>network security for layer 2 devices</dc:subject>
</xccdf:metadata>
```

Elements that support the `<xccdf:metadata>` element MAY use any desired metadata format; for `<xccdf:Benchmark>`, this is in addition to the Dublin Core recommendations already mentioned. Any XML metadata structures, including ad hoc structures, MAY be included in an `<xccdf:metadata>` element. Because any structures can appear in this field, authors SHOULD tag metadata with the *xmlns* prefix [XMLNAME] and, if available, the *@xsi:schemaLocation* attribute in order to identify the metadata structure utilized. Metadata SHOULD comply with existing commercial or government metadata specifications to allow benchmarks to be discovered and indexed.

Metadata is a powerful feature for authors. However, XCCDF puts some limits on the use of metadata:

- Metadata SHOULD NOT replace the functionality of existing fields within XCCDF. If the information matches the common use of some other XCCDF field, the information MUST appear in that XCCDF field. It MAY also appear in the `<xccdf:metadata>` element.
- Metadata SHALL NOT change the processing model as outlined in Section 7. Benchmark consumer products SHALL perform XCCDF assessments in the same way regardless of the presence of metadata. Metadata MAY still be used to support assessment features that are outside the purview of XCCDF—for example, to hold post-processing instructions to be performed on the completed XCCDF results or to display additional information to the user before or during the assessment.
- Metadata SHOULD NOT alter the character content of the XCCDF properties used to generate output during document generation. Contents of the `<xccdf:metadata>` element MAY be added to the output above and beyond the XCCDF properties. Metadata MAY contain instructions that cause different stylistic conventions to be adopted in the conversion of an XCCDF document to prose.

6.2.5 Platform Names

The Common Platform Enumeration (CPE) specification allows a specific hardware or software platform to be identified by a unique name. CPE names can express only single platforms (e.g., "cpe:2.3:o:microsoft:windows_xp:*:gold:professional:*:*:*:*" for Microsoft Windows XP Professional Edition). CPE applicability language statements can express complex logical constructions of CPE names.

`<xccdf:Benchmark>`, `<xccdf:Profile>`, `<xccdf:Rule>`, and `<xccdf:Group>` elements may be qualified by applicable platform using the `<xccdf:platform>` element. `<xccdf:TestResult>` elements may also include `<xccdf:platform>` elements. The `<xccdf:platform>` element's *@idref* attribute MAY hold a reference to either a CPE name or the *@id* attribute of a CPE applicability language expression that SHALL be defined as a child of the

`<xccdf:Benchmark>` using a `<cpe2:platform-specification>` element (see Table 4). (The syntax for referring to a local CPE applicability language identifier SHOULD be a “#” character before the identifier, such as “#cpeid1”.)

Within XCCDF documents, all CPE names SHALL comply with the CPE 2.3 Naming specification [IR7695], and all CPE applicability language expressions SHALL comply with the CPE 2.3 Applicability Language specification [IR7698]. CPE 2.0 names MAY be used for backwards compatibility, but their use has been deprecated for XCCDF 1.2. All CPE 2.3 names and applicability language expressions in XCCDF documents SHOULD use formatted string bindings but MAY use URI bindings instead, both as defined in [IR7695].

Here is an example of a `<cpe2:platform-specification>` element:

```
<cpe2:platform-specification>
  <cpe2:platform id="xp_and_acrobat_9.0">
    <cpe2:logical-test operator="AND" negate="false">
      <cpe2:fact-ref
        name="cpe:2.3:o:microsoft:windows_xp:*:*:*:*:*:*:*" />
      <cpe2:fact-ref
        name="cpe:2.3:a:adobe:acrobat:9.0:*:*:*:*:*:*" />
    </cpe2:logical-test>
  </cpe2:platform>
</cpe2:platform-specification>
<xccdf:platform idref="#xp_and_acrobat_9.0" />
```

If an `<xccdf:Profile>`, `<xccdf:Rule>`, or `<xccdf:Group>` does not possess any `<xccdf:platform>` elements, then it SHALL apply to the same set of platforms as its nearest enclosing ancestor `<xccdf:Group>` or `<xccdf:Benchmark>`. If the enclosing ancestor has no `<xccdf:platform>` element, then consult the next enclosing ancestor, repeating until either an ancestor has an `<xccdf:platform>` element or the `<xccdf:Benchmark>` element is reached. If no ancestor, including the `<xccdf:Benchmark>` element, possesses any `<xccdf:platform>` elements, then the `<xccdf:Profile>`, `<xccdf:Rule>`, or `<xccdf:Group>` SHALL nominally apply to all platforms.

6.2.6 `<xccdf:reference>` Element

The `<xccdf:reference>` element provides a reference to a document or resource where the user can learn more about the subject of the parent element. It may be included within `<xccdf:Benchmark>`, `<xccdf:Rule>`, `<xccdf:Group>`, `<xccdf:Value>`, and `<xccdf:Profile>` elements. When used, it SHALL have either a simple string value or a value consisting of simple Dublin Core elements as described in [DCXML]. It MAY also have an `@href` attribute that gives a URL for the referenced resource. References SHOULD be given as Dublin Core descriptions; a bare string MAY be used for simplicity. If a bare string appears, then it is taken to be the string content for a `<dc:title>` element. For more information, consult [DCES]. Multiple `<xccdf:reference>` elements MAY appear; a benchmark consumer product MAY concatenate them or put them into a reference list, and MAY choose to number them.

6.2.7 <xccdf:signature> Element

XCCDF supports a digital signature mechanism whereby XCCDF document users can validate the integrity, origin, and authenticity of documents. XCCDF provides a means to hold such signatures and a uniform method for applying and validating them [XMLDSIG].

The `<xccdf:signature>` element MAY hold an enveloped digital signature asserting authorship and allowing verification of the integrity of associated data (e.g., its parent element, other documents, portions of other documents). The signature SHALL apply only after inclusion (i.e., XInclude) processing. The `<xccdf:signature>` element is OPTIONAL, and it MAY appear as a child of the `<xccdf:Benchmark>`, `<xccdf:Rule>`, `<xccdf:Group>`, `<xccdf:Value>`, `<xccdf:Profile>`, `<xccdf:TestResult>`, or `<xccdf:Tailoring>` elements.

Any digital signature format employed for XCCDF MUST be capable of identifying the signer, storing all information needed to verify the signature (usually a certificate or certificate chain), and detecting any change to the signed content. XCCDF products that support signatures MUST support the W3C XML-Signature standard enveloped signatures, as defined in [XMLDSIG].

If multiple signatures are needed in an XCCDF document, at most one of them MAY be enveloped; all others MUST be detached [XMLDSIG Section 2].

The single child element of the `<xccdf:signature>` element SHOULD be `<dsig:Signature>` and that `<dsig:Signature>` SHOULD contain one or more `<dsig:Reference>` elements indicating each XML element to be included in the signature. The `<dsig:Reference>` URI SHOULD be a relative URI to the `@Id` attribute value of the enclosing object (prefixed by “#”). For example, if the `Id`="abc", then the `<Reference>` URI would be “#abc”.

6.2.8 Status Tracking

XCCDF provides two elements for status tracking: `<xccdf:status>` and `<xccdf:dc-status>`. Both elements are available for `<xccdf:Benchmark>`, `<xccdf:Rule>`, `<xccdf:Group>`, `<xccdf:Value>`, `<xccdf:Profile>`, and `<xccdf:Tailoring>` elements.

The intent of the `<xccdf:status>` element is to record the maturity or consensus level for its parent element. Permissible values are “incomplete” (under initial development), “draft” (released in draft state), “interim” (revised and in the process of being finalized), “accepted” (released as final), and “deprecated” (no longer needed). If there is more than one `<xccdf:status>` element for a single parent element, then every instance of the `<xccdf:status>` element MUST have a date attribute, and the `<xccdf:status>` element with the latest date SHALL be considered the latest status.

The `<xccdf:dc-status>` element holds additional status information using the Dublin Core format, expressed as elements of the DCMI Simple DC Element specification, as described in [DCES] and [DCXML].

6.2.9 Text Substitution

XCCDF provides capabilities to perform substitution within certain properties of an `<xccdf:Benchmark>`.