Network Working Group
Internet-Draft
Intended status: Informational
Expires: November 17, 2012

D. Waltermire, Ed.
NIST

May 16, 2012

Automated XML Content Data Exchange and Management
draft-waltermire-content-repository-00

Abstract

TBD...

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 17, 2012.

Copyright Notice

Table of Contents

1. Introduction

Data-driven programming is a common paradigm in software engineering. When using this approach, a program is developed to process a series of data statements that describe the sequence of actions to be taken. These data statements, often referred to as content, provide the user with a dynamic degree of control over the function of the software. In many cases, this approach can lead to a proliferation of content. Without adequate content management and distribution capabilities, use of content can become impractical.

It is common practice today to format content using the Extensible Markup Language XML . While many content management solutions exist today, few are designed to support the management and distribution of XML-based content. Current solutions largely focus on exploiting the raw XML syntax or a specific data model. Some solutions, such as XML databases, expose the raw syntax of XML for querying using techniques like XQuery. Other solutions utilize specialized database schema designed to support one or more specific data models represented in XML using XML Schema . These solutions are often brittle, inflexible to revisions of the underlying data models and do not adequately represent the logical information components used within data-driven programs.

XML-based data-driven content is produced by many organizations in a range of formats, covering many different information domains. Where content repositories exist to support this content, they often operate independently and vary in the data models and capabilities they support. Rarely do these repositories interact and if they do it is through proprietary interfaces. Content consumers often have to manually download the content they want to use with their tools. In many cases they may want to customize this content for local use and must contend with managing updates to the content manually.

One example of where data-driven programming is used is in the IT Security Automation community. Standardized security automation content is used to provide the instructions necessary for security tools to examine a computer’s state to evaluate and report on the degree of compliance to configuration policies, to detect the presence of vulnerabilities, and to verify the installation state of patches. Other tools use data-driven content to collect and correlate digital events or to aggregate security information. Much of the focus in the security automation community has been on defining the standards and schemas for expressing security-related data in XML. Standardizing the methods for retrieval and exchange of security automation content has not been a primary area of focus.

The content management challenges introduced by diverse data models,

decentralized production and use of content, and the proprietary nature of content repositories today create a need to define common content exchange requirements and mechanisms that will complement the content specifications and XML schemas.

The following challenges are addressed by this specification:

Distribution - In the absence of a standardized, automated distribution mechanism, content producers have no way to notify content consumers when new or updated content is available. Content consumers must manually import content at the point of use. This specification defines an automated notification mechanism that can be used to indicate to content consumers when new or updated content is available. The specification also defines the technical mechanisms used to exchange content between repositories, providing a standardized delivery mechanism to make remotely published content available at the point of use.

Reuse

Without a standardized method to search, retrieve and utilize existing content, both content consumers and producers have a tendency to recreate content. This duplication often causes content to become static or stale, introduces errors, and reduces the efficiency for developing content. In support of making content more reusable, this specification provides mechanisms for querying content so that it can be searched and gathered from many content providers. This allows organizations that are developing content to leverage, extend, and customize existing content from a variety of sources. This specification also defines a stable method of identifying blocks of externally provided content enabling content to be remotely referenced. This approach supports reuse and reduces the need for manual duplication across repositories.

Interoperability

Content repositories may require proprietary clients or tools to access their content. This hampers the ability for a content consumer to retrieve content from a variety of content sources using a single tool implementation. This specification standardizes the methods used to publish to and retrieve content from a content repository enabling standardized clients to be developed.

Access to content repositories may be restricted or require the use of various standard or proprietary communication protocols (e.g. HTTP, FTP). Content is often packaged using various

file formats and compression algorithms, such as Zip, CAB or GZIP. Variation in these approaches hampers interoperability. This specification standardizes the communication protocol and distribution formats used promoting interoperability.

Content packaging

XML-based content is exchanged as XML documents, also called instances. This document centric view of information does not align well with how humans use information. Humans are more comfortable working with logical objects that represent a concept (e.g. rule, assessment check, logical construct) verses XML syntax. While XML Schema enables these concepts to be modeled, XML is still represented as a collection of elements and attributes. This specification defines a metamodel that identifies the logical objects that are represented in XML-based content and their boundaries within the XML model enabling content repositories to use the conceptual view of the content.

This technique enables XML instances to be treated as containers of conceptual constructs. These conceptual constructs can be exchanged individually and can be composed into new documents dynamically based on metadata rules. This specification will provide a methodology for gathering and packaging content based on the needs or interest of the content consumer using a metadata approach.

Integrity

Content consumers and need assurances that the content that has been received has not been modified during the exchange process. This specification defines the use of automated mechanisms for verifying the integrity of exchanged content.

Confidentiality

In some scenarios, it is necessary to secure the exchange of content or restrict access to specific content. This specification will detail mechanisms for securing repository-to-repository and client-to-repository communications. Additionally this specification will specify authorization mechanisms that enable restricted access to content if needed.

Content Version Management

The content managed by content repositories may often undergo revision. When revisions occur, it is important to be able to

query specific revision to maintain the integrity of content bundles. This specification provides a query method that enables either a specific revision or the latest revision to be retrieved. This approach also enables remote references to include a content identifier and a specific revision.

Model Revision Management

Content repositories are often based on a specific data specification revision. When using this approach, updating content repository software to support specification revisions may require costly, time-consuming effort. Organizations maintaining content repositories may be reluctant to adopt new revisions or support old revisions due to this burden. This makes it difficult for a tool to use content based on an older or newer model revision. This specification defines properties within the metadata model to indicate where content is backwards and forwards compatible. These properties are then used to enable content to be provided based on the required model revision or to drive proper error handling where content is incompatible.

For example the Open Vulnerability and Assessment Language (OVAL) versions content based on the major and minor revision of the OVAL XML schema. A repository containing OVAL content may have content ranging from OVAL 5.3 to 5.10. The difference in model version, while minor, could negatively impact a security tool’s ability to properly process content that is outside of its expected range . This could cause tool errors or unexpected results to be produced. By using the model revision properties in the metamodel, the effective model revision of content returned from a content repository may be calculated based on the maximum schema revision used. Alternately, substitute content may be provided that supports a specific maximum schema revision provided in the query.

By addressing these challenges, content producers will be able to effectively manage and share content they produce, and content consumers will be able to effectively use content provided by many different providers. By defining communication interfaces that can leverage existing communication protocols, we can begin to automate content distribution among disparate systems and make content more readily available. By defining a federated data model, we can establish rules and relationships of data types which allow for flexible content management with support for dynamic methods for collecting and bundling content for consumers.

Sections [...] of this document focus on:

TBD

1.1. Requirements Language

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 RFC 2119 [RFC2119].

1.2. Terms

1.2.1. Content

1.2.2. Security Automation Content

1.2.3. Content Producer

1.2.4. Content Consumer

1.2.5. Content Bundle

2. Key Concepts

This section provides a high-level overview of key concepts introduced in this specification. The first concept subsection describes a content metamodel that provides a needed level of abstraction over XML-based data models. The second subsection describes the federated content architectural approach defined within this specification. Through the use of these concepts, a robust, general purpose, distributed content management system is possible that supports automated content exchange between content consumers and producers.

2.1. The Content Metadata Model

In order to create a generalized approach to XML-based content management it is necessary to generalize how XML-based data is processed by the content system. A variety of XML schema languages are used to define the syntax used to express a data model in XML. While these languages provide rules to constrain XML instance data, they do not adequately describe the information objects that exist within the model or the relationships between information objects. An information object is a block of XML data that represents a specific concept such as policy definition, a configuration setting or a scanning rule. Relationships represent cross references or links between information objects. Information objects and relationships are concepts that humans use to conceptualize the data model primitives that exist within content. In order for a content

management approach to be successful, a mechanism is needed that bridges the gap between the XML syntax understood by machines and the conceptual primitives that humans understand. The content metadata model provides this bridge.

Within the content metamodel, an information object is represented as an entity definition.

Complete this section...

2.2. Content federation

Complete this section...

Discuss

Use of namespaces within content identifiers for repository lookup using DNS SRV records. Discuss using external namespaces for other cases.

Discuss authoritative content repositories vs. caching repository content.

Discuss using an architectural model similar to DNS for content repositories (e.g. local, forwarding, caching).

3. IANA Considerations

This memo includes no request to IANA.

4. Security Considerations

All drafts are required to have a security considerations section. See RFC 3552 [RFC3552] for a guide.

5. References

5.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

5.2. Informative References

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.

Appendix A. Additional Stuff

This becomes an Appendix if needed.

Author's Address

David Waltermire (editor)
National Institute of Standards and Technology
100 Bureau Drive
Gaithersburg, Maryland 20877
USA

Phone:
Email: david.waltermire@nist.gov