Protect Enterprise Digital Assets with Built-in Security

A Nuxeo Tech Brief
Introduction

Security technologies are stronger than ever, and yet, security breaches can still happen; absolute security does not exist. Security is more than just systems; it’s also about the human users who constantly access, share and act upon digital assets and enterprise content. An effective strategic plan is essential to not only successfully beat back security attacks, but also minimize security vulnerabilities from happening in the first place.

This tech brief, targeted to application architects and developers, will review the key elements digital asset management (DAM) or enterprise content management (ECM) systems must have to protect your most valuable digital assets and intellectual property. You will also discover the set of strategic principles built into the Nuxeo Platform that enable a high level of flexible security tactics right out of the box.
It will be helpful to begin by organizing the foundations of a solid application security strategy around a typical user session, from start to finish. Without the four key elements below, we cannot establish a viable security strategy.

This paper will address how Nuxeo provides developers with these four key security elements so developers do not have to worry about building them from scratch. We will also identify key strategic principles built into the Nuxeo Platform, which align all four of the above security elements.

**Security Integration Foundation**

To achieve security at the business application level, all strategic security elements must be coordinated and aligned as a whole.
Authentication

We must reliably identify the user and ensure her/his identity is propagated at all times.

Perhaps the single most vital strategic principle built into the Nuxeo Platform is that security is always on. It is not possible to even develop on the Nuxeo Platform without at least a default security model.

Security, of course, starts with user authentication.

Each user request goes through a central authentication filter. Any user request must be authenticated before the Nuxeo Platform will perform anything. Even at the Java level, inside the platform code, access to the Nuxeo repository requires a valid security context. This means that even if someone could find a way to bypass the http-level filter, they would still be blocked and would need to provide credentials to be able to get any data from the Nuxeo repository.

Nuxeo also supports a wide array of authentication protocols and providers, including SAML2, OpenID, login/password, oAuth, LDAP/AD and advanced two-factor authentication (2FA). You can see the entire list of protocols and advanced identify management schemes supported by Nuxeo in our documentation center.

Additionally, virtually the entire Nuxeo Platform is pluggable, making the Nuxeo Platform future-proof. For example, it is possible - and easy - to write a custom plugin to implement an authentication protocol if necessary.

Access Controls

We must ensure each user can access all permissible data and content, and perform all permissible actions - but nothing else.

Nuxeo provides two concurrent layers of security that users cannot bypass:

- Access control lists (ACLs), used by Nuxeo to manage security at the data level. Nuxeo provides the ability to allow administrators to define ACLs via the Nuxeo UI. You can also use code to apply ACLs programmatically. Because ACLs can be erroneously set by an administrator, organizations can also choose to use some automatically set ACLs, or enforce company security rules by leveraging custom security policies (see next bullet).
• **Custom security policies** consist of dynamic code that will be executed each time the Nuxeo repository needs to determine whether or not to allow a user access request or other user action. Custom security policies enable Nuxeo to enforce mandatory access controls (MACs) that may override or supplement any applicable ACLs - a common requirement for military systems.

Of course, security models are of little use if they do not scale well and impair application performance. For example, post-query filtering ("late binding" security) often causes a slow presentation of results. Nuxeo avoids this issue and provides excellent performance at enterprise scale, by enabling the indexing of security permissions, pushing filters at a low level.

For example, if the Nuxeo backend is a relational database, query filters can be processed using SQL inner joins. Alternatively, because Elasticsearch is tightly embedded within the Nuxeo Platform, a query can be processed through the Elasticsearch index, applying a filter on ACLs. Nuxeo also enables the pushing of custom security policies at a low level as well.

Additionally, the Nuxeo Platform includes a comprehensive unit testing framework that deploys a live, “real” repository with configurable sets of test features spanning all Nuxeo functionality. Common security model tests include checks to verify a given ACL or security policy works as expected, or identify which users have access to a given document.

### Data Protection & Integrity

**We must ensure all content is secure, whether at rest or in transit. This includes ensuring our content has not been falsely altered in any manner.**

Nuxeo Platform traffic is encryptable with SSL and is fully configurable by developers for optimum performance. Nuxeo also supports AES encryption of content at rest (in storage), including safe storage of keys within a hardware security module (HSM) connected to your Java virtual machine (JVM). It is also possible to encrypt the backend database and search indexes at a system level.

The Nuxeo repository stores each binary file separately from its metadata, which is retained in a database. By default, the metadata for each binary file includes its digest (also called checksum) - the unique compact digital signature. If a binary file was to be altered in some way,
it will no longer match the checksum in the database and it would be detected. The digest/checksum is also used to automatically manage de-duplication of binary files.

Additionally, it is possible to detect if any data has been altered via direct database access, outside of the normal flow of the application by computing and storing a digest/checksum of the metadata - a keyed-hash message authentication code (HMAC).

**Activity History**

**While we want to take a proactive stance against security breaches, we must ensure an audit trail exists that enables a review and analysis of any historical user or system activity which may be questionable in hindsight.**

Another core strategic principle built into the Nuxeo Platform: the ability to log any / every event.

Nuxeo provides a comprehensive audit log module to capture and record any system or application event. You can configure custom events to be logged and store anything you need alongside each audit log entry. For example, the Nuxeo log audit module can serve as a write-only change log for the whole system, storing all changes done on document properties, even including each iteration of the full document with every change!

Such in-depth logging and information retention are a common requirement for government/military entities, highly regulated industries and ISO compliance. Organizations can accumulate “Big data”-sized logs quickly.

Nuxeo provides the massive scalability necessary to manage the most demanding audit logging requirements, with support for NoSQL (MongoDB) as a content backend, as well as Elasticsearch for storing and searching of audit logs. The core Nuxeo repository itself has been benchmarked at over one billion documents processed using commodity hardware.
With the strategic security foundation already in place within the Nuxeo Platform, enterprise application developers and administrators are now equipped to tactically defend against security breaches, whether caused by deliberate attacks from would-be fraudsters, or by honest user mistakes.

Let’s explore some of the most common hacker “attack vectors” and the defensive tactics available in the Nuxeo Platform to defeat them.

<table>
<thead>
<tr>
<th>Attack Vector</th>
<th>Defensive Tactic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password theft.</td>
<td>As previously noted, Nuxeo supports a wide variety of authentication providers, including 2FA authorization, rendering a stolen login/password useless.</td>
</tr>
<tr>
<td>Administrator backlog delays and/or errors.</td>
<td>Nuxeo provides features designed to make admin work easier, including: Visually manage when a user’s permissions should be removed from the system, including automatically adding new users. Enable extra temporary user permissions that will automatically expire on a given date.</td>
</tr>
<tr>
<td>Disclosure via search results.</td>
<td>Search results are always filtered at the index level revealing only the content users are allowed to access. Additionally, Nuxeo’s “military-grade,” rules-based security provides the ability to quickly add a custom layer of permissions that can supplement or override ACLs that might otherwise allow access to new, highly confidential information.</td>
</tr>
</tbody>
</table>

A serious company crisis can occur simply due to end user search results revealing the names of confidential projects, clients or other terms; for example:

- A local advertising agency finds and uses a non-localized version of a marketing video
- An internal user finds search results matching “layoff plan”
- Search results reveal a well-known celebrity is a client, violating their confidentiality agreement
<table>
<thead>
<tr>
<th>Attack Vector</th>
<th>Defensive Tactic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Metadata fraud.</strong>&lt;br&gt;A user attempts to alter metadata outside of the business process. For example, a user tries to downgrade the confidentiality level property for a digital asset so unauthorized users can now freely access it.</td>
<td>Audit logging can store every action taken by any user or administrator, including property changes. If the property change is not allowed under access controls or security rules, the property cannot be changed. Security is applied to permitted user actions as well; for example, the user in the above example may not have access to the edit form or even an edit button!</td>
</tr>
<tr>
<td><strong>Theft of files “at rest.”</strong>&lt;br&gt;Someone attempts to copy content from “behind the screen;” directly from storage backends, whether in local or cloud-based storage, or a cloud delivery network (CDN).</td>
<td>Nuxeo supports AES encryption of all content at rest (in storage), with keys safely stored within a hardware security module (HSM) connected to your Java virtual machine (JVM). If content delivery via a CDN is enabled, Nuxeo provides users with a short lived URL for file access directly from the CDN without requiring access to go through the Nuxeo Platform. The result is faster performance without compromising security.</td>
</tr>
<tr>
<td><strong>Recording network traffic.</strong>&lt;br&gt;Someone uses a packet sniffer to deduce confidential information by identifying details of others’ searches, data entry, file names imported and downloaded, etc.</td>
<td>Nuxeo traffic is encryptable with properly configured SSL.</td>
</tr>
<tr>
<td><strong>Flawed security model logic.</strong>&lt;br&gt;For example, ACLs and/or custom rules-based security policies do not limit content access and/or user actions as intended.</td>
<td>The Nuxeo Platform includes a comprehensive unit testing framework that deploys a live, “real” repository with configurable sets of test features spanning all Nuxeo functionality, including, of course, the security model. Common security-related tests include checks to verify a given ACL or Security Policy work as expected, or identify which users have access to a given document. Because the Nuxeo test framework is based on an actual repository, and security is always on, no specific action is required to invoke your security model for testing.</td>
</tr>
<tr>
<td>Attack Vector</td>
<td>Defensive Tactic</td>
</tr>
<tr>
<td>------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Administrator-level errors or intentional fraud.</td>
<td>Nuxeo supports external identity systems, which limit possible changes an admin can make on their own.</td>
</tr>
<tr>
<td>For example, an admin person agrees to fraudulently modify a user profile to a higher level that enables access to sensitive, confidential content.</td>
<td>Policy-based security can also be defined to override certain admin-level modifications deemed to go beyond usual and customary work requirements.</td>
</tr>
<tr>
<td></td>
<td>Finally, custom workflows can be configured to require potentially sensitive admin changes to be submitted for advance approval. Every step of the approval process, as well as all actions taken by admins and users alike, are recorded and retained by the Nuxeo audit log service.</td>
</tr>
</tbody>
</table>

**Conclusion**

Security challenges can be nerve-wracking to say the least. To quote Nuxeo's own security-related documentation, “Absolute security does not exist. It is only a question of inventiveness, complexity and time for the attacker, against our ability to guess, prevent, detect, mitigate and respond to threats and vulnerabilities.” Organizations must vigilantly work to minimize threats and vulnerabilities from even existing in the first place.

Security protection for enterprise business applications cannot be an after the fact exercise, separate from the application development process itself. Nuxeo effectively fulfills this vital strategic requirement.

The Nuxeo Platform has been built from the ground up with foundational strategic principles that enact security as a permanent, pervasive process throughout the entire software development life cycle. Content-driven applications developed on the Nuxeo Platform are highly secure by design.

Global organizations rely on the Nuxeo Platform for highly secure, flexible and scalable application development. We invite you to join them and make the move to Nuxeo now.
Why Nuxeo?