Evidian

Web Access Authentication for Apps

Trusted partner for your **Digital Journey**

Authentication in Web Applications

Why Authentication and access control should not be handled by Web applications?

When developing a Web application, any design error in authentication handling may lead to bypassing the authentication mechanism.

Recently, a well-known Web application was exposed to a security flaw. This application was secured by a robust two-factor authentication. Unfortunately, it was possible to bypass this strong authentication by using web APIs exposed to the mobile Apps.

New pattern designs allow developing seamlessly for Web and mobile applications, by using web APIs exposing the same features to both worlds. Authentication and access control must be correctly adapted and managed, and no security shortcuts must be possible.

Web development process

During the development process of any Web application, new needs appear and a continuous evolution cycle is required as new features are created and exposed. Secure coding is now well- known by experimented developers. Testing security and features are always major topics even in applying Agile methods. New authentication mechanisms can be implemented at any time to ensure that the correct user has access to the appropriate features. However, one of the new features may introduce a weakness in the overall security of the application at any time. New threats appear and the time-to-market pressure is not a favorable environment in those conditions. It is necessary to keep track of any security modifications and implement the correct behavior at the application level.

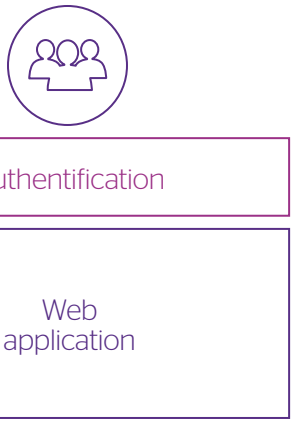
Therefore, the application developers must remember the features to develop, as well as the way the authenticated users access these features. They have to understand the impact of a new authentication policy on the developed features and on the features to develop. They have to deal with two major constraints: developing quickly new services and ensuring the security of the whole applications.

High-skilled developers are the only ones able to develop with all those constraints. However, even these developers may introduce weaknesses.

...but your application will evolve:

Authentication is the first step but authorization must also be handled.

- You will need, sooner or later, to integrate new authentication means.
- Your corporate users want to authenticate using their Windows domain authentication means, while external users need a simple strongauthentication and do not want to remember passwords.
- Is your application ready for federated users or social authentication?
- How do you manage the authentication life-cycle, new users, lost password and password lifespan?



Dynamic Authorization and Authentication

Dynamic Authorizations Management

In order to centrally manage how each feature is exposed to a particular user, the concept of an external authorization manager should be applied to externalize the logic of the authorization rules computing. As a consequence, the application development is focused on creating new features without the need to link features to the user right management. The strength of this approach is to "free" the application developer from authorization concerns.

Dynamic authorization is devoted to an external dedicated component, the authorization server. XACML is a language to interface it, but any API providing the same services may also be used. This approach keeps the conceptual architecture simple by using building blocks dedicated to each function. Applications just ask the central Authorization point whether the access is granted, based on the user's identification and environment (where from, at what time, etc.).

Nevertheless, to interact directly with the Dynamic Authorization mechanism, an application must know and understand the authentication means of users. Was the user strongly authenticated? What is the level of this authentication? In this context, adding new means of authentication or separating the user populations depending on their authentication means, must still be handled at the application level. There is still a risk that parts of the new features or old features do not take into account all the possible scenarios. To reduce this risk, authentication management must be delegated to a specified component, the Dynamic Authentication Manager.

Dynamic Authentication Management

Inside the same Web application, or a collection of Web applications and Web APIs, there may be different levels of trust requiring simple authentication or strong authentication. The way users are authenticated is associated with the authentication level from the lowest to the strongest. For example, a banking service may require a simple login and password authentication for basic operations, but they may need a two-factor authentication when users access their bank account and their transactions.

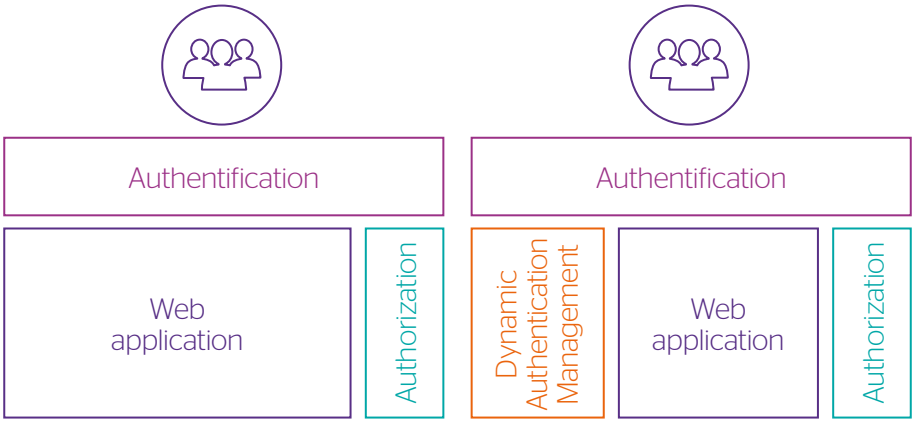
The multi-level authentication is a way of giving different meanings to how the users are authenticated; social authentication, login and password authentication, One Time Password authentication, two-factor authentication, or Kerberos authentication do not have the same meanings and levels. It may be convenient to consider OTP and Kerberos having the same level, or maybe not depending on the user's environment and point of connection. Depending on the user, his connection environment, his means of authentication and the services or Web applications he wants to access, the Dynamic Authentication Manager will provide him with a list of appropriate authentication methods. Users just choose an authentication method, depending on the means they own. Associated with this method, the level will also define the selection of the allowed resources, even for HTTP methods.

Level of Authentication

For example, a user coming from the Internet may have to choose between a social authentication and a two-factor authentication. This same user coming from the intranet may have no other choice than to authenticate using Kerberos. Two- factor authenticated users may have the same rights as Kerberos authenticated users, while the social authenticated users would have access to only a partial view of the resources. In this example, the Kerberos and 2FA authentication have the same high level, while Social Authentication has a lower level authentication.

Web applications should know the user's authentication level for each accessed resource and give this information to the Dynamic Authorization Manager building block.

The development process and the security integration are easier and independent. Nevertheless, how to integrate these two building blocks for authentication and authorization with existing applications? How to "free" the already developed applications and how to seamlessly integrate these two components? Introducing a Web Access Manager becomes the best way to separate the previous building blocks from the Web application to be developed, or already developed.



Web Access Management

Web Access Management

A Web Access Manager integrates the Dynamic Authentication management and the Dynamic Authorization management while protecting and hiding the protected Web application resources. Acting as a reverse Proxy, the WAM is able to perform the authentication and authorization phases before giving access to the resources of the Web applications.

Existing applications may be integrated using a Single-Sign-On mechanism without any modification inside the applications.

After a primary authentication, the WAM injects the secondary login and password in any existing authentication mechanism. A simplified cooperation may be deployed for applications being developed; the WAM only transmits users' identification and their Identity. Doing so, it "frees" the application from handling authentication mechanisms, the level of authentication and any re-authentication scenario. The WAM identifies the resources accessed and determines by itself which level of authentication is required before giving access.

State of the Art

The authentication method and the way users are identified will be improved and modified depending on the new needs, new requirements or new usages. The WAM may evolve or may be configured according to the new needs, without any impact on the protected applications.

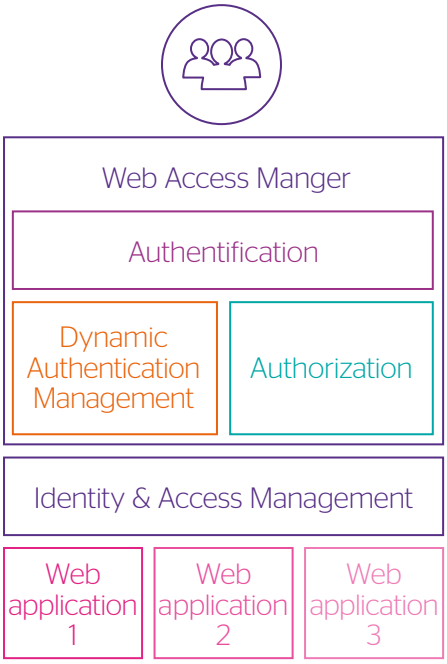
These protected applications may follow their own development life-cycle, new features will be added or improved without requiring modifications in the way users are authenticated or identified. New user populations are added, new user repositories are configured in the WAM to manage new partners, external users or social users.

The level of security of the web access will depend only on a single component; the WAM. Web applications are protected without security shortcuts, even if new services are developed and deployed.

Mandatory Features

A typical Web Access Manager building block, has to support the following features:

- Support of multiple user repositories.
- Central administration to manage authorizations and policy.
- Access policy, based on resources, location and user's group or attributes.
- Access policy to applications supporting multi-levels of authentication.
- Strong authentication with software or hardware OTP, with and without cellular network.
- A large range of authentication methods, including multi-factor and chainable authentication methods.
- The support of identity federation protocols for cloud and non-cloud applications.
- A seamless integration in the existing infrastructure, Microsoft AD, RADIUS servers, or any network load-balancer, reverse-proxies, firewall, WAF,...
- Integrated service portal and workflows for user self-registration, reset password, account provisioning, protected service accesses.
- Fully customizable and integration in the existing applications' look-and-feel.



Web Access Manager is a single point of access to protect all your Web Applications.

Web Access Manager provides security without requiring any additional software, neither on the browser side nor on the Web server side, and does not impose any specific configuration on the browser. It offers architecture for flexible Web access control.

Federation protocols are also used to provide SSO in Cloud Apps, or to use external identities from Cloud Identity Providers.

Web Access Manager allows enterprises to build a secure e-business platform, with a single point for enabling and managing secure access to their Web resources (pure Web or Web-enabled legacy), by filtering user access to the URLs according to user profiles.

Web Access Manager in action

A typical WAM scenario:

- A user tries to access a resource i.e. a protected URL.
- WAM determines the level of authentication required for this access.
- If the user is not authenticated, unknown at access time, or his/her current authentication level is too low for this access, then WAM launches the authentication process.
- The available authentication methods will depend on the connection environments, the user, and the resources to access.
- Once the user is identified and authenticated, WAM analyzed the rules set for authorization. Additionally, WAM may rely on Post authentication and Post-authorization mechanisms to determine if the resource is reachable by the user.
- Post-Authorization and Post- Authentication may be re-routed to a dedicated decision point in order to overlay the rules computed by the Web Access Manager.
- WAM requests the resources from the protected server, and possibly injects SSO or user identity information inside the HTTP query or headers.
- Before sending the resource back to the user, Web Access Manager may inject data, remove information, re-write URLs, or inject HTML/CSS/javascript, in order to give an external view of an internal resource. Indeed, WAM has the ability to parse the resource in depth, log the traffic and generate audit events about user interactions.

Introducing a new authentication method, two-factor or social authentication, is now easy and completely independent from the protected applications. Nevertheless, the use of a Web Access Manager introduces two new needs:

- An Identity and Access Management building block, in order to identify user populations, distribute means of authentication, and provision accounts inside protected applications for Single Sign-On.
- A clear identification of all visible application URLs to protect and the associated level of trust.

Identity and Access Management

The Identity and Access Manager is connected to the Web Access Manager. It will provide the necessary features to grant the user's access to protected applications, by provisioning secondary accounts inside existing applications or by setting attributes in the user's profile for the Dynamic Authorization mechanism.

The Web Access Manager will expose public parts of the Identity Manager, i.e. the Web interfaces dedicated to the users are protected by the WAM. These interfaces are used to activate the authentication means and handle users' requests to access particular services or applications.

While the Identity and Access Manager provisions the secondary accounts in existing applications, it will also provision these accounts in the Web Access Manager. The user will benefit from Single Sign-On features and only the primary authentication must be known. In the same way, when a user loses the right to access applications, the Web Access Manager will be immediately informed and there will be no more risks of a remaining unwanted account.

Impact on the development processes and responsibilities

The need to handle authentication means and mechanisms inside Web applications disappears with the adoption of a Web Access Management building block. Applications receive the users' identification and, possibly for information, the level of authentication currently in use. Based on that identification, the logic of the application remains the same. The level of authentication might not be used for security reasons, but only for displaying correct links or information about how to use the authentication mechanism, reset the password, or customize the displayed pages.

Development teams stay focused on features, roadmap and secure coding but not on how to authenticate the users anymore. They have to identify and enumerate all URLs and Web servers used to render the features.

The risks on each URL collection must be evaluated and an authentication level should be associated. Then, the Web Access Management team creates services using the URL collections and maps the authentication policies and the level of authentication with those services. Unlisted URLs are simply not reachable at all.

Web development and authentication matters are split into two separate independent responsibilities. One might evolve without any security impact on the other one.

Handling user rights, authentications and accesses, do not concern applications anymore. Identity and Access Management becomes the control tower for handling users. Introducing new authentication methods, publishing new services, deploying new authentication means or becoming social, are new tasks for the Identity & Access Management officers.

Conclusion

Independent building blocks and team responsibilities ensure that there is no workaround possible to bypass strong authentications. It simplifies the development process and removes part of the security pressure from Web development teams. At any time, the authentication strength may be increased without any impact on the already-developed features.

Dynamic Authentication Management and Dynamic Authorization Management ensure the adaptability and the evolution of the Web Access Management project for new needs and new scenarios.

What about mobile Apps?

The same Web Access Manager protects your mobile application's server backend.

Mobile Web Applications

Web applications may be used as-is on mobile device. More and more web applications respect the responsive design and are immediately useable on these devices without any modification.

All the Web interfaces of WAM are compatible with the responsive design principles. A Web application protected by WAM can therefore be used on mobile devices. Users have the ability to authenticate with any authentication method, access their account profile, reset their password, view the list of their allowed services, etc.... But there is a major drawback in using Web applications inside a Mobile Web Browser; the user's experience is not as good as with a native application.

Native Mobile Applications

When developing an application targeted to mobile devices, you should consider two different approaches:

- A native application, using the native development language and platform Software Development Kit. This kind of application will be able to use all the power-features of the devices, but you will need to develop for each supported system.
- A native-hybrid application, using both native calls to the underlying system and HTML5/Javascript for displaying screens and application logic. These kinds of applications are more powerful than pure Web applications but once again, the user's experience could be limited compared to full native applications even if there are frameworks allowing to easily integrate the platform features.

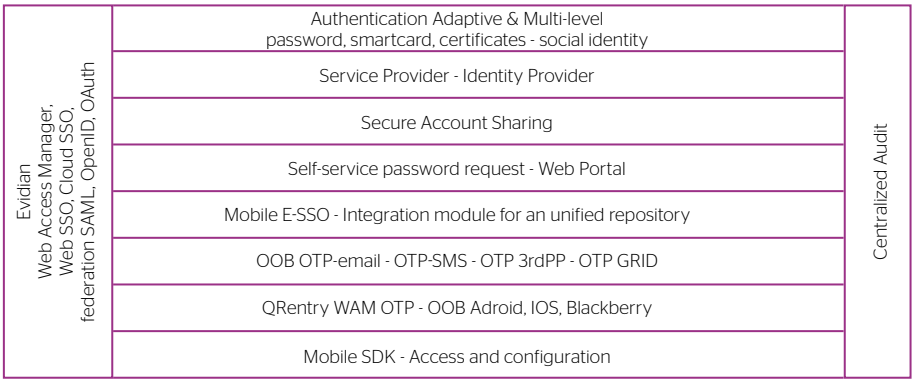
In both cases, the authentication mechanism must be developed with a security dimension in mind. Developers may introduce weaknesses by trying to shortcut the security features already in place to protect published APIs. This leads to security flaws in the whole Web Application.

So it is important to use the same security and access control in the native mobile application and the Web applications that use the same back-end APIs.

WAM Mobile SDK

The main objective of a native mobile SDK is to help developers to integrate new features without redefining the whole mechanisms.

The WAM mobile SDK brings all the necessary features to interact between a pure-native mobile application or a hybrid-native mobile application, and the protected Web APIs. These APIs are the same as the ones used by the Web Application, as described previously. APIs are mainly REST calls to a Web server backend or calls to Web Services or dedicated URLs.



The goals of the WAM SDK are:

- To provide an authentication mechanism with the same level of trust than the authentication mechanism of the Web version of an application,
- To handle authentication, expiration of authentication or multi-level authentication and to provide a seamless cinematic in the user's authentication experience.
- To provide the necessary configuration information to the mobile application, allowing to discover the WAM end-points, and the URLs of the protected services to use.
- To provide easy-to-develop secure communication mechanisms between the mobile application and the WAM protecting the back-end.

...and the API backend access

The WAM SDK helps to easily create communicating applications with backend servers protected by Web Access Manager.

The authentication methods are still defined in the Web Access Manager and are applied by the mobile application. Any change in the way users must authenticate will be reflected in the mobile application logic.
If the mobile application tries to access a forbidden URL or a restricted URL requiring a higher authentication, then the mobile application will not be allowed to retrieve the URL, or the application will start the higher authentication scheme.

Web Access Manager is a single point of access to protect your Web Applications, and your API backend for your mobile application.

Web Access Manager SDK provides required features from strong authentication to secure communication from the mobile application to the protected API. Any change in your access management policies will be reflected both in the mobile application access and in the web access to your application.

Security shortcuts are not allowed.

Identity & Access Management

A complementarity of products and features between WAM and I&AM

Identity and Access Manager

Identity & Access Manager allows managing the full lifecycle of user identities and access rights. It contains a security policy engine to define the user access rights matrix, an end-user portal to let users manage the identity and right lifecycle including a set of ready-to-use identity and access rights workflow process and a set of provisioning connectors to address applications including SaaS applications.

The Workflow Editor will let users create new workflows or modify the logics of a workflow delivered with Identity & Access Manager. Identity & Access Manager can provision the SSO products like Web Access Manager.

Evidian Identity & Access Manager easily models your security policy with roles that are understood by operational managers. Automatic and versatile rules track the changes in the users' characteristics and assign or remove access rights accordingly. A sandbox avoids the risks for such operations. Effective integration with Evidian access control means any change of access rights is immediately reflected in the Single Sign-On, and user access information makes reporting even more profitable.

I&AM simplifies the management of access rights to resources and accelerates their allocation. Users are quickly operational and naturally respect the security policy and regulatory constraints.

...combined with Web Access Manager

Web Access Manager provides all the required features to handle authentications, authorizations, end-user interface, and a set of dedicated self-service modules for authentication activation and reactivation, lost-password...

WAM provides features like logging and tracing any user request, multi-directory administration and self-registration, multiple administration user roles, secondary accounts self-provisioning or sharing...

Web Access Manager is a full-featured stand-alone product for Web Access Management, with embedded user interaction workflows, but when the population becomes more complex, when there is a mix between internal corporate users and external customer or partner users, then a powerful access governance product like Identity and Access Manager is mandatory to easily model your security policy with roles.

Web Access Manager and Identity Access Manager work together and offer you a set of processes for identity and right governance, ready-to-use and also customizable to meet your business requirements.

I&AM will drive the access policy of Web Access Manager, and it will provide unique centralized audit and reporting integration for all your accesses.

...provide access governance:

- For Web Applications.
- For native mobile applications using the Web Access manager SDK.

Identity management features to define and maintain a central identity repository.

Non – intrusive with existing identity sources.

Provisioning engine to apply access control policies on target systems whether on premises or cloud applications.

IAM password management & authentication methods.

Powerful security policy engine capable of taking into account complex and evolutive access control needs with centralized or disributed administration.

Integrated end-user self administration. Identity & entitlement management processes, for end-users, operational managers and security officers to be accountable and to collaborate, ensuring access governance.

A prerequisite for GDPR*

Identity and Access Management is one element among a range of technical counter-measures to mitigate risks related to data protection. In addition to its access control, strong authentication and identity governance, Evidian Suite takes into account the requirements for Users' Rights in all its products' roadmaps. Notifications, dedicated personal data reports and self-service functionalities allow users to exercise their rights freely and enable GDPR compliant processes.

* General Data Protection Regulation

About Evidian

Evidian is the Identity and Access Management (IAM) software suite of Eviden.

Evidian IAM is the European leader in identity and access management with a presence which is growing rapidly beyond Europe, particularly in Japan and the US.

More than 5,000,000 users in more than 900 organizations throughout the world connect to their companies every day and manage their access rights with Evidian identity and access management solutions.

CT_J1326_180626_RY_WP_WEBACCESS