THE SECURITY LEADER'S GUIDE TO ACCESS SECURITY

If you're like many security leaders, you're in the throes of digital transformation. As your users and apps move beyond the firewall, you're tasked with securing your digital enterprise against increasing security threats. And without impeding productivity or progress.

You could look to your legacy web access management (WAM) system, but relying on it for modern security is cumbersome at best. At worst, it's downright risky.

Unsuited to today's challenges, WAM systems don't support the migration of apps to the cloud and struggle with mobile apps and APIs. You could try to implement add-ons, but doing so is complex, requires lots of resources and creates a domino effect that leaves your enterprise vulnerable.

The Problems with Using WAM for Modern Security

But you can't put the brakes on either. You need a solution that's designed to address today's challenges. One that provides:

- Secure, centralized, policy-driven access for all users to all applications—no matter their type or where they live (public cloud, private cloud, on-premises, enterprise, third-party, mobile).
- Context-sensitive access management that adapts to and continuously verifies user, device and application data, including location, network and device status.
- Centralized control of identity data with policy-based data governance, privacy and consent management—and end-to-end identity data security.
- An extensible platform with industry leading, comprehensive support for modern IAM standards across employee, partner and customer use cases.

Said another way, to secure your modern enterprise, you need a modern access security solution.

01

MODERNIZE USING IDENTITY

When it comes to securing and granting access beyond the firewall, identity provides the gateway. By securing your web applications and APIs using mature identity standards—plus state-of-the-art authentication and authorization protocols—you're able to give the right people access to the right things, seamlessly and securely.

Modern access management requires a three-pronged approach to give users secure access to all apps and resources:

1. **Single Sign-on (SSO):** Replaces all those passwords with a single set of enterprise credentials and a consistent authentication experience.
2. **Multi-factor Authentication (MFA):** Adds another layer of security beyond passwords and is a must for the digital enterprise.
3. **Access Security:** Centralizes access control with a policy-driven security layer for all web and mobile apps, and APIs.

Modern access security allows you to form a central access point to control who has access to what and for how long. You can deploy your applications on premises, in a private cloud or in the public cloud, while maintaining the same level of identity and access management across all three. And you can do so quickly to protect new systems, accelerating digital transformation.

While it's natural to want to leverage your legacy system, WAM systems—which pre-date cloud and mobile applications—are intended to protect internal web resources that are hosted in enterprise data centers. Combining a legacy WAM with API gateways offers limited integration, not to mention complicated deployment and management. You're left with heavy, but fragile products that require intensive system integration services for simple installs and updates. Not to mention a risky security posture at best.

Bottom line: relying on a legacy system to do a job it wasn't designed for slows forward progress, and increases your costs, headaches and vulnerability to attack.

02

STANDARDIZE ON STANDARDS

Today, most organizations use password-based authentication and authorization for APIs. In the absence of standardized identity systems, this is a simple and highly insecure interaction:

USER » APP: Hi, I would like to see my data.
APP » USER: Please sign-on with userid and password.
APP » API: Hi, I need access to Service A, here is the user id and password.
API » APP: Ok, here is Service A.
APP » USER: Here is the data you requested.

The above is true whether the service being accessed is a social site or a cloud app or the user is in front of a desktop machine, laptop, tablet or phone. Secure access to APIs is critical for both web and mobile applications. Obviously, this is not a secure way to control access and provides no scope of access.

The OAuth 2.0 protocol solves this by allowing a formal way for a client to ask for access to a resource (an API) by presenting an access token previously issued by an authorization server. OAuth specifies three roles in this process:

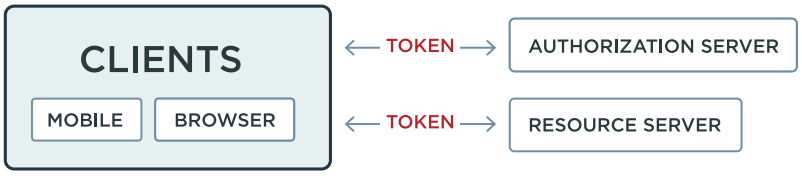

Figure 1: The OAuth 2.0 protocol provides a formal way for a client to request access to an API by presenting a token issued by an authorization server.

CLIENT: Typically an application (web or mobile) that wants access to a particular API.

AUTHORIZATION SERVER: A server that issues access tokens and refresh tokens on behalf of the resource server.

RESOURCE SERVER: A server that hosts and protects resources and makes them available to properly authenticated and authorized clients.

As shown in Figure 1, a modern access security solution that supports OAuth provides a secure way to authenticate a user, ask for consent and provide control over authorization scope—which is critical to managing secure access for today's enterprise.

Want to learn more? Read the Essential OAuth Primer.

03

DYNAMIC PAGE-LEVEL ACCESS CONTROL

Today, most organizations have built some level of authorization into applications which can make maintaining policy difficult, expensive and time consuming. Employing a single control layer that determines access policy for each application and application page is a better way to manage authorization.

Let's say you have a shopping application with a page or section for rewards members. When a user clicks to navigate to that page, you have to determine whether that user should have access or not. A modern access security solution protects all URLs with a proxy/gateway that executes policy. This policy typically looks something like:

- Check if the user has already signed on (authenticated).
- Require the user to authenticate if not previously signed on.
- Check user attributes once authenticated to determine if the user is authorized by policy to access the page.
- For sensitive pages, require the user (if authorized) to step-up authentication via multi-factor authentication or some other mechanism.
- Display the page.

Today's access security solutions allow you to apply policy centrally to multiple pages or applications. If you need to change access policy for any page, you can do so without modifying the application.

04

ACCELERATE YOUR CLOUD MIGRATION

A modern access security solution doesn't only help you ensure secure access across applications, it helps you accelerate your migration of legacy applications to cloud environments (like Amazon Web Services).

Benefitting from a lightweight proxy-based architecture and centrally managed access policies, you're able to maintain security policy to applications that were previously managed by WAM systems, like CA Siteminder or Oracle Access Manager. Once applications are migrated to the new system, you can move them to any environment, easily and at any time.

Isn't that the type of responsive, flexible and lightweight solution you want to meet today's security challenges?

Providing secure access is more challenging than ever before. You don't need to make it any harder. Identity gives you the key to enabling the right people access to the right things, seamlessly and securely.

Ready to learn more about leveraging identity to secure your enterprise? Read our Ultimate Guide to Modern Access Management.