SAML SINGLE SIGN-ON
Learn how PistolStar’s PortalGuard® Reduces Multiple Login Prompts!

TABLE OF CONTENTS

Summary .................................................. 2
The Basics .............................................. 2
Usage Scenarios ................................. 4
Benefits .................................................. 4
How It Works ........................................ 5
Deployment ........................................... 8
System Requirements ....................... 8
Alternative SSO Methods .................. 9
Summary
Putting an end to users’ complaints about having to remember multiple passwords is an objective in many organizations. With multiple web applications being accessed, IT staff also struggle to manage multiple user repositories. For example, when a password changes in one repository, it may not be updated in the others. This can lead to security and support issues that make it even more difficult to implement a password security policy across multiple systems.

To solve these issues, you may look towards single sign-on (SSO) as a solution to eliminate multiple password prompts and streamline access for users. However, many SSO solutions are costly and difficult to implement to effectively handle all user access scenarios. Integration is especially difficult when attempting to allow the single sign-on experience to continue for external users, from staff to customers or partners, who all want seamless access to hosted web applications.

For example, if you logged into your locally managed application containing financial information and then decided to view emails (e.g. Office365) without SSO, you would be prompted to login manually to Office365. With an SSO solution in place, you would be able to enter Office365 in the cloud without being prompted again. This type of integration is essential for providing users with seamless application access as well as maintaining unified security and compliance standards to protect your organization’s data.

Institutions opting to use cloud infrastructures are struggling to achieve seamless access for their users. Integration is the key challenge; they are left with forcing their users and administrators to login multiple times or figure out how to federate the identity in the cloud while maintaining regulatory compliance.

The solution is a product which can create a single or federated sign-in process to handle an organization’s multiple local and cloud applications and provide a centralized point of secure access.

The Basics

Identity Federation
Identity federation is the concept of linking a user’s identity across multiple systems or servers. When two servers are federated, the authentication against one can be leveraged to prove the user’s identity to the other. Some application servers in the secondary role can allow this without requiring the user to register an account.

Identity federation typically entails some level of Single Sign-On (SSO). Once authentication has been performed against a primary server, the user’s session with that server can then be used as a launch point for SSO-based access to other federated services. This can be used to realize the common business requirement of reducing access barriers without compromising the security of the systems involved.

There are multiple protocols that can be used to apply the successful authentication against one system to another, but Security Assertion Markup Language (SAML) has emerged as a clear front-runner.
Security Assertion Markup Language (SAML)

Originally developed by the OASIS Security Services Technical Committee, Security Assertion Markup Language (SAML) is leading the way in providing seamless, web-based SSO as an open, widely implemented, industry standard protocol. SAML is an XML-based authentication protocol that passes assertions between SAML-enabled applications.

Once the end-user requests access to a resource, an online identity provider creates a SAML token containing the end-user’s identity assertions. Once those assertions are validated by the resource server, the end-user is granted access without any further password prompts.

SAML is heavily leveraged today for numerous reasons:

- It works for cloud-based services that are typically hosted offsite as well as “on premise” services.
- It is typically wrapped in the HTTP/HTTPS protocols which ensures it can be used by any client device regardless of operating system (e.g. Windows, Mac, and Linux) or architecture (PC, iPad, smart phones).
- The use of HTTP/HTTPS also allows for easier network administration since these ports are more frequently open in server or client firewalls.
- Manual user authentication for multiple services can be redirected and always be performed against a single Identity Provider (IdP). This “choke point” allows for network and access policies to be controlled at a single point, making them much easier to implement and enforce.
- Users can be authenticated against virtually any user repository using any required method(s) without impacting the downstream servers, which always receives a SAML assertion.

PortalGuard’s Identity Provider

PortalGuard’s SAML Identity Provider (IdP) acts as a SAML-based portal that uses a single set of credentials for the portal login itself and then grants access to your web-based applications. When using SAML, the end-user will no longer be prompted for multiple passwords. As a result, administrative and IT costs associated with performing password resets in several places, synchronizing numerous sets of password quality rules that may or may not overlap, and creating and disabling accounts will be greatly reduced.

The main objective with using PortalGuard’s SAML-based portal is to give a user one place to login using stronger authentication, thereby granting them access to web-based applications, partner sites, and/or to disparate sites where they are authenticated without having to sign on manually again.

PortalGuard provides seamless integration with web-based applications whether cloud-based, private, on-premise, or behind a firewall. This integration allows you to use PortalGuard not only as your single authentication point but to provide stronger authentication. This may include full two-factor or knowledge-based authentication, plus all the beneficial self-service password management functionality that helps increase usability and reduces Help Desk calls. Although many web-based applications are already SAML-enabled, alternatives such as WS-Federation, forms-based SSO, or other established authentication mechanisms, such as Kerberos, can be provided by PortalGuard to enable SSO for non-SAML applications.
Usage Scenarios
PortalGuard’s flexibility allows you to choose the appropriate authentication method for each user, group, or application by leveraging Contextual Authentication. Varying access scenarios in every organization drive the need for this type of authentication. For instance, users on your Local Area Network (LAN) may only need to provide strong passwords; whereas, a traveling salesperson or external user is presented with two-factor authentication.

CONSISTENT AUTHENTICATION INTERFACE
When the user is always forced to login to your SAML-enabled applications using PortalGuard, a consistent authentication interface and process can be enforced. This reduces end user training and frustrations associated with managing multiple accounts through multiple websites.

ENFORCING SELF-SERVICE ENROLLMENT
Using SAML SSO offers a seamless way to provide users the ability to unlock their account, enroll answers to challenge questions, reset or recover their forgotten passwords, and manage their mobile device for use in alternate or multi-factor authentication.

MULTI-FACTOR AUTHENTICATION
Because the end user communicates directly with the PortalGuard server, authentication decisions made by PortalGuard are strictly enforced. This ensures a high level of security and consistency.

Benefits

- Eliminate the need to develop and maintain your own portal
- Reduce the number of passwords users are required to remember and manage
- Implement and enforce configurable password policies
- Remove the need to manage external users’ credentials
- Optionally increase security using any combination of transparent barriers
- Add stronger authentication using two-factor and/or knowledge-based authentication for select users or groups of users (e.g. Administrators)
- Reduce password-related Help Desk calls related to password and access issues
How It Works

The following steps and screenshots show how the PortalGuard SAML IdP works using two different SSO methods, SP-Initiated and IdP-initiated.

SP-Initiated SSO

**STEP 1:** The user opens their browser and accesses the target server, e.g. http://mail.google.com/a/example.com

**STEP 2:** The target server sees the user has not yet authenticated, so it generates a SAML request and returns it and the originally requested URL (the “RelayState”) as hidden input fields in an HTML form response.

**STEP 3:** JavaScript in the response automatically submits the form to the PortalGuard Identity Provider (IdP). Note: the user can be forced to login using any of PortalGuard’s authentication methods including Knowledge-based or full two-factor authentication (2FA).

**STEP 4:** PortalGuard’s login screen is presented to the user. The user enters their username/password and clicks “Login”. Note: this login screen can be fully customized to match your organization’s branding, creating a seamless experience for the user. The user can optionally reset a forgotten password from this screen too.

**STEP 5:** PortalGuard validates the submitted username and password in real-time against the appropriate directory. If correct, the user has now established a session with the PortalGuard web server.

**STEP 6:** The original SAML request is now serviced by the PortalGuard IdP application. It generates a SAML response and sends it and the “RelayState” back to the end user’s browser wrapped in an HTML form.

**STEP 7:** JavaScript in the HTML response automatically submits the form to the target server’s Assertion Consumer Service (ACS). The “SAMLResponse” and “RelayState” are included in this form data.
SP-Initiated SSO (continued)

STEP 8: The target server parses and validates the SAML response. It uses the embedded identity claims to determine the user’s identity and gives the user access to the application.

IdP-Initiated SSO

STEP 1: The user opens their browser and accesses the PortalGuard IdP application, e.g. “portalguard.acme.com/sso.”

STEP 2: PortalGuard’s login screen is presented to the user if they do not already have an active session.

STEP 3: The user enters their username/password and clicks “Login.”

STEP 4: PortalGuard validates the submitted username and password in real-time against the appropriate directory. If correct, the user has now established a session with the PortalGuard web server.

"On August 2013, first day of classes, IT Helpdesk encountered 1,200 login issues without PortalGuard. On August 2014, first day, we encountered only about 200 login issues with PortalGuard implemented. We are very pleased and PortalGuard has been a tremendous help.

Nakitah, Robeson Community College
IdP-Initiated SSO

**STEP 5:** The user gains access to the SAML IdP jump page.

**STEP 6:** User clicks a displayed application. *Note: the applications available to the user are configurable by the admin.*

**STEP 7:** The click is serviced by the PortalGuard IdP application. It generates a SAML response and sends it back to the end user wrapped in an HTML form.

**STEP 8:** JavaScript in the response automatically submits the form to the target server’s Assertion Consumer Service (ACS).

**STEP 9:** The target server parses and validates the SAML response. It uses the embedded identity claims to determine the user’s identity and gives the user access to the application.
Deployment
Implementation of the PortalGuard platform is seamless and requires no changes to Active Directory/LDAP schema. A server-side software installation is required on at least one Windows server on the network which is running IIS.

PortalGuard is a flexible authentication platform with multiple layers of available functionality to help you achieve your authentication goals:

- Self-Service Password Reset
- Web-based Single Sign-on
- Stronger or Two-factor Authentication
- Contextual Authentication
- Password Synchronization
- Password Management

System Requirements
PortalGuard supports both direct access and authentication to cloud/web-based applications. PortalGuard can be installed on the following platforms:

The PortalGuard server also has the following requirements:
- .NET 2.0 framework or later must be installed
- (64-bit OS only) Microsoft Visual C++ 2005 SP1 Redistributable Package (x64)

PortalGuard is fully supported for installation on physical and virtual machines. PortalGuard supports running on the following platforms:
- Microsoft Windows Server 2003 (32 or 64-bit)
- Microsoft Windows Server 2008 (32 or 64-bit)
- Microsoft Windows Server 2008 R2
- Microsoft Windows Server 2012 and 2012 R2

The PortalGuard IdP requires the following:
- The target server must support SP-initiated SAML SSO using SAML v2.0 POST binding method
- The target server must be configured to not allow manual authentication. Otherwise, users could use that method and bypass the interactions with PortalGuard (typically this is implicit when enabling SAML).
- A trust must be configured between the PortalGuard Identity Provider and the target server/Service Provider by importing the PortalGuard public signing certificate.
- The end user must have network connectivity (typically HTTPS) to both the PortalGuard server and the target server.
- The PortalGuard server does not need network connectivity to the target server since the user’s browser delivers all SAML messages.
Alternative SSO Methods

This tech brief is intended to highlight SAML-based SSO; however, the following single sign-on methods are also useful:

**Cookie-based SSO:** Works by using Web based HTTP Cookies to transport user credentials from browser to server without input from the user. Existing credentials on the client machine are gathered and encrypted before being stored in the cookie and sent to the destination server. The server receives the cookie, extracts and decrypts the credentials and validates them against the internal server directory of users.

**Kerberos-based SSO:** Kerberos enables a user to log into their Windows domain account and then receive SSO to their internal applications. Kerberos requires the user to have connectivity to a central Key Distribution Center (KDC). In Windows, each Active Directory domain controller acts as a KDC. Users authenticate themselves to services (e.g. web servers) by first authenticating to the KDC, then requesting encrypted service tickets from the KDC for the specific service they wish to use which happens automatically in all major browsers using SPNEGO (see below).

**Claims-based SSO:** Claims (aka "assertions") are created by a claims issuer that is trusted by multiple parties. Claims are typically packaged into a digitally signed token that can be sent over the network using Security Assertion Markup Language (SAML).

**Form-filling SSO:** Form-filling allows for the secure storage of information that is normally filled into a form. For users that repetitively fill out forms, especially for security access, this technology will remember/store all this information and secure it with a single password. To access the information the user only has to remember one password and the Form-filling technology can take care of filling in the forms.

**NTLM-based SSO:** It is possible for a user to prove they know their password without actually providing the password itself. NTLM achieves this using a challenge and response protocol that first determines what type of NTLM and encryption mechanisms the client and server mutually support, then cryptographically hashes the user’s password and sends it to the server requiring authentication.

**SPNEGO-based SSO:** There are instances when the client application and remote server do not know what types of authentication the other one supports. This is when SPNEGO (Simple and Protected GSSAPI Negotiation Mechanism) can be used to find out what authentication mechanisms are mutually available. Some of these mechanisms can include Kerberos and NTLM authentication.

**Reduced SSO:** Reduced single sign-on is widely used for limiting the number of times a user will be required to enter in their credentials to access different applications. With critical applications, reduced SSO also offers a technique to make sure that a user is not signed on without a second factor of authentication, having been provided by the user.

**Enrollment-based SSO:** A user logging into a website may choose to have their credentials permanently remembered for that site. This is accomplished by creating an encrypted cookie on the user’s machine for that web browser that contains the user’s credentials. This cookie persists across different browser sessions and restarts of the machine, but will be set to expire after a set period. The next time the user accesses the website, the server recognizes the cookie, decrypts it to obtain the user’s credentials and completely bypasses the login screen after validating them successfully.