PISTOLSTAR®

PortalGuard

Browser-Based Authentication and Password Security For Microsoft SharePoint, IBM WebSphere and WebSphere Portal, and IBM Lotus Domino

Tech Brief

PistolStar, Inc.
PO Box 1226
Amherst, NH 03031 USA

Phone: 603.547.1200
Fax: 603.546.2309
E-mail: salesteam@pistolstar.com
Website: www.pistolstar.com

PortalGuard

Summary

PortalGuard is an authentication and security solution that allows end-users to securely authenticate and manage their portal login credentials directly from a Web browser, while providing administrators with functionality to meet or exceed their security objectives and achieve compliance with government and industry regulations. With PortalGuard, administrators can implement best practices for ensuring stronger and consistently secure authentication.

To keep up with the speed of business, organizations must provide globally and remotely-based employees, customers, partners and vendors with streamlined access to corporate intranets, extranets and portals. Whether they are working remotely or onsite, end-users need to have the ability to manage their own passwords and perform password resets, while IT needs to ensure the authentication process remains secure. IT administrators are increasingly challenged by network intruders who attempt to gain access by guessing passwords or seizing upon weak ones, making access control and password quality a chief concern.

PortalGuard provides easy yet highly secure access to Microsoft SharePoint, IBM WebSphere, WebSphere Portal and IBM Lotus Domino. End-users can manage and reset their passwords directly from a browser, and they have the option of logging in with their Microsoft Active Directory credentials to access all SharePoint, WebSphere or Domino domains.

Providing a true upgrade to authentication functionality and password security, PortalGuard contains numerous configurable authentication and password features that allow administrators to implement processes for stronger and consistently secure authentication, thus preventing unauthorized access and ensuring that data is protected.

PortalGuard addresses the security concerns of industries that deal with highly sensitive data by providing stronger authentication features such as authentication with username, password and challenge question and answer to gain access; limiting concurrent login sessions, multiple challenge questions for self-service password reset; and configurable strike-out limits by person, group or hierarchy. By enhancing authentication security and locking down access, PortalGuard helps organizations meet the security requirements of SOX, HIPAA, GLBA and other initiatives.

Allowing end-users to perform self-service password management, whether they are working onsite or on the road, removes the downtime they experience having to call the Help Desk and frees up the time and resources used by administrators to respond to password-related requests. Best of all for administrators, PortalGuard provides them with password security functionality that helps with satisfying compliance auditors.

The following provides a general explanation of how PortalGuard works plus an explanation specific to the WebSphere Portal. The later explanation corresponds with the diagram in Figure 1 below.

How It Works – *General Explanation*

Authentication and Access Control
With PortalGuard, the end-user signs on to their SharePoint, WebSphere or Domino server as they would with native authentication. However, PortalGuard provides numerous options for defining successful authentication and controlling access. For example, authentication can be prevented if the end-user strikes out, their password has expired, or they already have an open login session on a different machine.

Active Directory/LDAP Integration
PortalGuard has the ability to integrate Active Directory or another LDAP implementation, allowing end-users to use their Active Directory credentials to log into all SharePoint, WebSphere or Domino domains. By leveraging the Active Directory password and password policies, PortalGuard reduces the number of passwords, password prompts and password stores.

PortalGuard uses the LDAP protocol to test the entered username against Active Directory/LDAP. With this capability on Domino, administrators do not need to maintain the HTTP password in the Domino Directory. They also no longer need audit exceptions for Domino passwords and can now clear the HTTP password values for all end-users.

Self- Service Password Reset Using Multiple Challenge-Response Functionality
End-users can also reset their portal password from a remote location without requiring Help Desk assistance. Responding to multiple challenge questions that the end-user pre-configures is required for resetting passwords securely. End-users can access this reset capability right from the logon screen or from a link specified by the admin in the “wrong password” message they receive from PortalGuard.

Powerful Features for Meeting Security and Compliance Objectives
Numerous features in PortalGuard enhance security by allowing administrators to establish password rules by person, group or hierarchy and enable/disable certain password behaviors. Administrators can now configure the number of password strike-outs allowed for each user and will receive an alert when a strike count is exceeded. They also have the ability to set password expiration intervals and select a grace period in which expired passwords must be changed.

PortalGuard prevents multiple users from logging in with the same credentials, validates password strength during login, locks out inactive users after *n* days, and restricts the frequency with which a previously used password can be re-used.

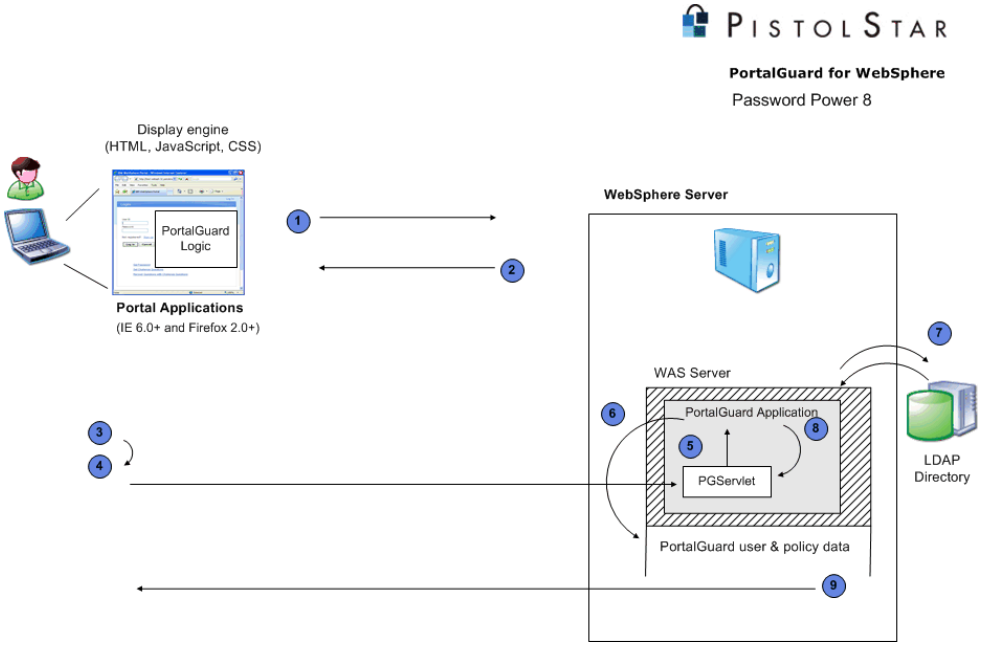
With PortalGuard’s auditing capabilities, administrators can quickly monitor and log end-user authentication activity that could potentially impact data security, such as last login date and time, invalid usernames and numerous other actions.

Improved End-User/Administrator Experience
PortalGuard offers several authentication features that are not found in the latest releases of SharePoint, WebSphere or Domino, and which improve the end-users’ and administrators’ experience. End-users have the ability to gain access from remote locations as well as self-manage their passwords. They also possess fewer passwords and can choose to employ alternate logons (e.g. Active Directory or Domino HTTP to access Domino). PortalGuard’s self-registration feature

allows end-users to create their own accounts with simple workflow and without an administrator's involvement, if desired.

PortalGuard increases productivity by allowing end-users to perform self-service password resets and eliminating the administrators' burden of resetting forgotten passwords and managing multiple user accounts and password policies.

How It Works – *For WebSphere Portal*



Steps:

1. With PortalGuard for WebSphere Portal, the unauthenticated user opens their Web browser and requests a resource from the WebSphere Portal server.

2. The server responds by sending the Portal's login page. PortalGuard's logic has been added to the source HTML of the login page.

3. The user enters their username and password in the normal fields on the login page and clicks the Login button.

4. The PortalGuard logic in the login page intercepts the login attempt and uses AJAX in the background to send the login information to the "PGServlet" on the Portal server that was created as part of the PortalGuard installation.

5. The PGServlet loads and calls into the PortalGuard application on the Portal server. PortalGuard is implemented as stand-alone DLLs that are installed on the Portal server and loaded by the PGServlet.

6. PortalGuard reads its policy and user profile data to determine how to authenticate the user. For ex., if the user account is already locked, a response structure

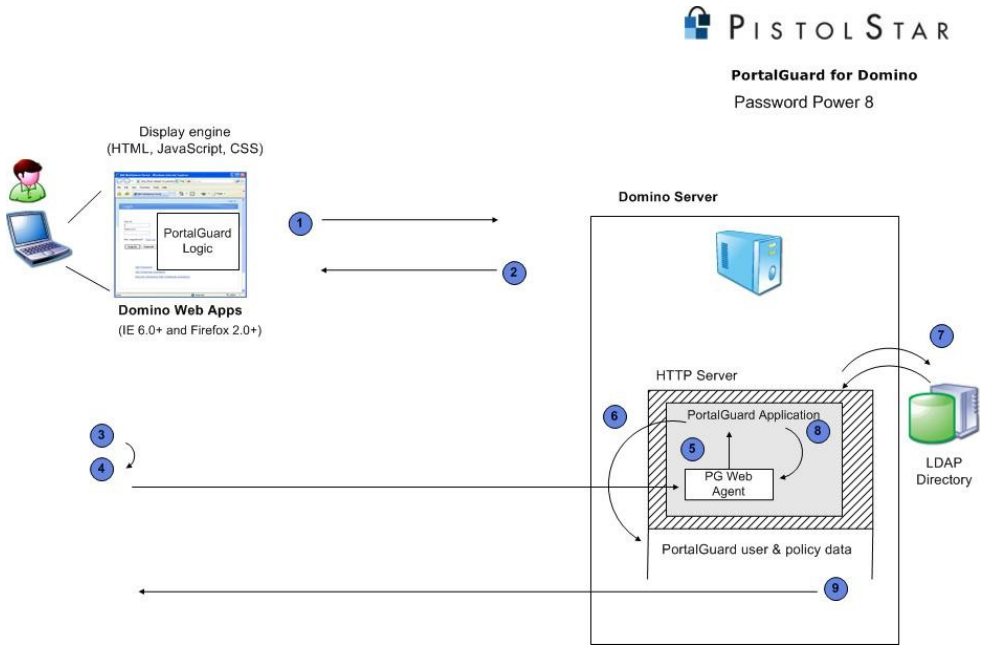
is populated with this information and the process skips to Step 8.

7. The user is looked up in the designated LDAP server and, if they exist, the validity of the provided password is checked by performing a LDAP bind operation against the server with the submitted credentials.

8. PortalGuard checks to see if there are any constraints that would prevent the user from logging in (e.g., expired password, too many open login sessions, etc.). A response structure is populated with the result of these checks and passed back to the PGServlet.

9. The PGServlet generates the response from the structure and sends the information back to the PortalGuard logic on the login page. This response is interpreted by JavaScript within the PortalGuard logic and any error conditions or messages are displayed to the user. If there are no conditions that would prevent the login, the PortalGuard logic allows the original login attempt from Step 3 to continue, passing the authentication request to the Portal server's default authentication mechanism.

How It Works – *For Domino R6/7/8/8.5*



Steps:

1. Unauthenticated user opens web browser and requests a resource from the Domino HTTP server.

2. The Domino login page is returned as the response from the server. PortalGuard's logic by default is added to the standard login page in the domcfg.nsf.

3. The user enters their username and password in the normal fields and clicks the Login button.

4. The PortalGuard logic in the login page uses AJAX to send the login information in the background to a Domino web agent. The agent resides the domcfg.nsf on the server and is created as part of the PortalGuard installation. The user's browser does not navigate away form the login page during this time.

5. The web agent is written in Lotusscript and all additional design elements (forms, script libraries, etc) are contained entirely within the domcfg.nsf on the Domino server.

6. The web agent reads in its policy and user profile data to determine how to authenticate the user. For example, if the user account is already locked, a response structure is populated with this information and the process skips to step #8.

7. The user is looked up in the Domino Directory infrastructure. If required, a designated LDAP server can also be provided. If the user exists, the validity of the provided password is checked by checking the hash in the user's Person document (or performing a LDAP bind operation) with the submitted credentials.

8. PortalGuard checks to see if there are any constraints which should prevent this user from logging in (expired password, already a current login session, password is insufficiently complex, etc). A response structure is populated with the result of these checks.

9. The web agent generates the response from the structure and sends the information back to the PortalGuard logic on the login page. The response is interpreted by Javascript within the PortalGuard logic and any error conditions or messages are intuitively displayed to the user directly on the login page clearly prompting them with any corrective actions they may need to take.

If there are no conditions that should prevent the login, the PortalGuard logic allows the original login attempt from item #3 to continue which passes the authentication request to the Portal server's default authentication mechanism. This will result in the user receiving a session token from Domino that will be used for the remainder of their browser session or until they manually log out.

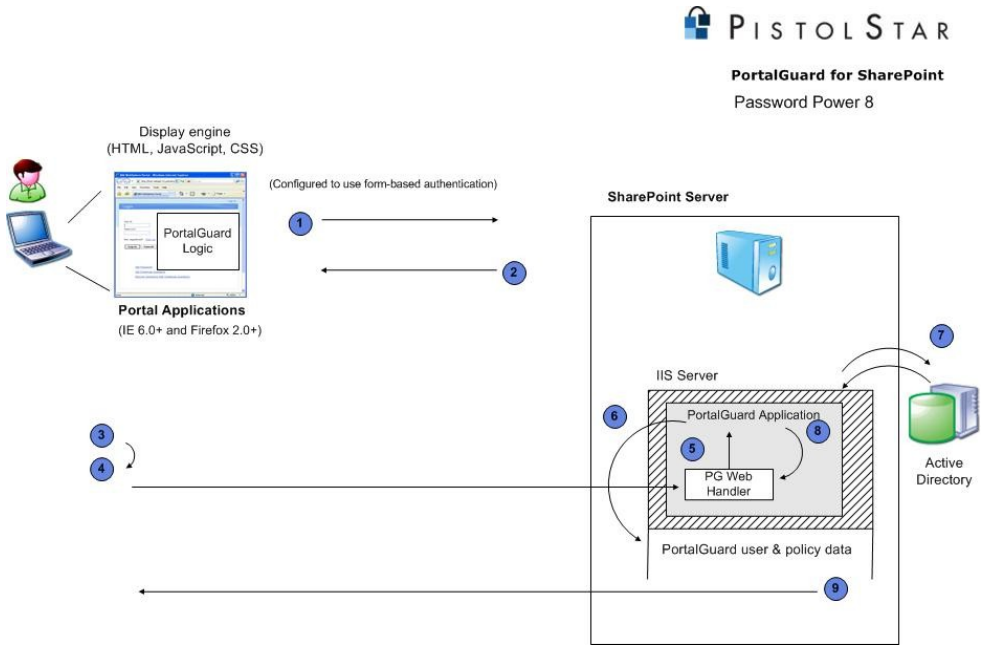
How It Works – *For Sharepoint*

Steps:

1. Unauthenticated user opens web browser and requests a resource from the Sharepoint server. The Sharepoint server has been configured to use Forms-based authentication.

2. The Sharepoint login page is returned as the response from the server. Alternatively, PortalGuard's logic can also be added to the source HTML of the standard login page.

3. The user enters their username and password in the normal fields and clicks the Login button.



4. The PortalGuard logic in the login page uses AJAX to send the login information in the background to an ASP web handler called “PG”. This server resides on the Sharepoint server and is created as part of the PortalGuard installation. The user’s browser does not navigate away from the login page during this time.

5. The PG web handler dynamically loads and calls into the PortalGuard application on the Sharepoint server. PortalGuard is implemented as stand-alone DLLs that are installed on the Sharepoint server. These DLLs are loaded by the PG web handler using native calls to unmanaged code.

6. PortalGuard reads in its policy and user profile data to determine how to authenticate the user. For example, if the user account is already locked, a response structure is populated with this information and the process skips to step #8.

7. The user is looked up in the designated LDAP server (e.g. Active Directory) and if the user exists, the validity of the provided password is checked by performing a LDAP bind operation against the server with the submitted credentials.

8. PortalGuard checks to see if there are any constraints which should prevent this user from logging in (expired password, already a current login session, password is insufficiently complex, etc). A response structure is populated with the result of these checks and passed back to the PG web handler.

9. The PG web handler generates the response from the structure and sends the information back to the PortalGuard logic on the login page. The response is interpreted by Javascript within PortalGuard logic and any error conditions or messages are intuitively displayed to the user directly on the login page clearly prompting them with any corrective actions they may need to take.

If there are any conditions that should prevent the login, the PortalGuard logic allows the original login attempt form item #3 to continue which passes the authentication request to the Sharepoint server’s default authentication mechanism. This will result in the user receiving a session token form IIS/Sharepoint that will

Deployment

The PortalGuard installation is very quick, typically less than 30 minutes. It is only installed on the servers on which you would like to enable its functionality and does not require hardware or changes to Active Directory schema. There is no end-user set-up other than to configure the challenge questions and answers.

System Requirements

PortalGuard is a server-side solution that integrates seamlessly with Microsoft SharePoint, IBM WebSphere and IBM Lotus Domino. It supports Windows SharePoint Services 3.0 and higher, WebSphere/WebSphere Portal 5.1 and higher, and Domino 5 and higher.

###