WHITE PAPER

Identity-Enabled Web Services

Standards-based identity for Web 2.0—today

Overview

Web Services are emerging as the preeminent method for program-to-program communication across corporate networks as well as the Internet. Securing Web Services has been a challenge until recently, as typical Web authentication and authorization techniques employed browser-to-server architectures (not program-to-program). This resulted in user identity ending at the Web Application Server, forcing the Web Services Provider to trust blindly that the Web Services Requester had established identity and trust with the end user. Moreover, this mechanism did not provide any way for the Web Services Provider to verify the authenticity of the request. What was needed is a way to provide end-to-end federated identity that spans all the way from the Web browser to the Web Services Provider—**Identity-Enabled Web Services.**

The convergence of Internet security standards has enabled Web Services and Web 2.0 to become a reality. Security Assertion Markup Language (SAML), an OASIS standard, has emerged as the lingua franca between Web browsers, Web application servers and Web Services Providers that use Simple Object Access Protocol (SOAP). SAML provides a common mechanism that allows identity to be passed through all layers. A WS-Trust Security Token Service (STS) is a "Rosetta Stone," translating between domain-specific security tokens and SAML—the glue that makes it all happen. Deploying an STS is the quickest, simplest and most scalable route to Identity-Enabled Web Services.

Ping Identity™

Introduction

Web Services are a gateway technology that provides interoperability across a wide variety of other software technologies. Typical enterprise deployments include using Web Services to enable ("webify") legacy systems, mainframe, ERP and CRM systems. What all of these "back-end" technologies have in common is that they are vital to the enterprise's long-term interest, and they generally lack the ability to communicate with modern systems like Web and application servers. Web Services are a set of technologies that communicate using modern, open standards like HTTP and XML to enable connectivity to the enterprise's core transactional and data-rich back-end legacy systems.

The latest incarnation of distributed computing frameworks are Service Oriented Architecture (SOA), which are most often implemented with SOAP and REST (Representational State Transfer) Web Services. Some of the major architectural principles driving the adoption of Web Service-based architectures are the ability to use open standards (e.g. XML, SOAP, WS-Security, and HTTP) and a focus on interoperability, reuse and loose coupling. Fundamentally, when a Web Service Requester and a Web Service Provider communicate with one another, they should not know or be dependent upon the underlying details of each other's implementation—they are loosely coupled using a message-based integration.

Message-based integration separates Web Services architecture from its predecessors. For example, in the 1990s, J2EE application servers like WebLogic® and WebSphere® were integration workhorses. J2EE clients communicated with J2EE servers using proprietary, binary protocols and formats such as a WebLogic client talking to a WebLogic server. The advent of XML enabled replacement of proprietary interfaces with XML-based open standards.

The Web Services Identity Challenge

Until now, most Web Services deployments relied on application- or system-level authentication to establish trusted user identity. In effect, a Web Services Provider validates the identity of the Web Service Requester–the application issuing the SOAP request–requiring it to trust whatever is contained in the message body. This trust model is not fully secure, however, because this scheme lacks any way to verify the authenticity of the request, leaving the door open for potential attack.

A typical incremental improvement over blindly trusting the request is to use mutual authentication mechanisms available in transport protocols such as HTTP and TLS. This approach markedly improves the channel security so that the Web Service provider has high confidence that a trusted host sent the message. However, the contents of the message, such as data payload, are still unverifiable, which still leaves the potential for an inside-the-firewall attack. The net result is that the business logic and decisions executed by the Web Services provider are not based on verifiable information. This is a subprime situation, particularly for mission- and business-critical applications.

In addition, increasing regulatory compliance and audit requirements are forcing organizations to consider a higher assurance level for user identity in Web Service transactions. Unfortunately, those needing a greater assurance level of the user's identity were forced to implement proprietary mechanisms with questionable levels of usability, manageability, and scalability—until now. Increasing compliance requirements and the increasingly interconnected nature of computing resources dictate a higher level of user identity assurance in their SOA environment.

Fortunately, three open standards—Web Services Security (WSS), Security Assertion Markup Language (SAML) and Web Services Trust (WS-Trust)—lay the foundation for a solution that allows trusted user identity information to be included in each SOAP request. The result: Identity-Enabled Web Services.

PingIdentity™

Web Service Security Standards

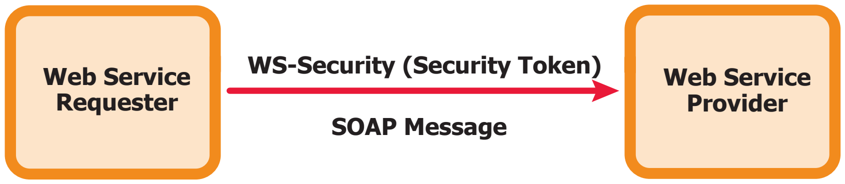
Securing Web Services involves delivering the same security services that are involved in securing any resource. Namely, based on business requirements, the architect/developer has to consider the following security functions when an organization chooses to use Web Services:

- Confidentiality
- Integrity
- Authentication
- Authorization

However, since Web Services architectures decouple the Service Requester and Service Provider, the ability to seamlessly mediate authentication and authorization is no longer possible due to the cross-domain nature of these applications. Over the last three to five years, groups like OASIS and W3C developed open standards to ensure that organizations do not have to invent proprietary mechanisms to address these security requirements for Web Services. The standards that provide these mechanisms are WS-Security, WS-Trust, and SAML.

WS-Security

WS-Security is an OASIS Web Services security standard widely implemented in all major SOAP engines such as IBM® WebSphere, Microsoft® .Net and Apache Axis. It defines mechanisms that specifically address SOAP message security. WS-Security does not define any new SOAP security mechanisms; rather it describes how to apply existing security standards to address



WS-Security Passes Identity Information via Security Token Embedded in SOAP Message

confidentiality, message integrity, authentication and authorization within a SOAP message.

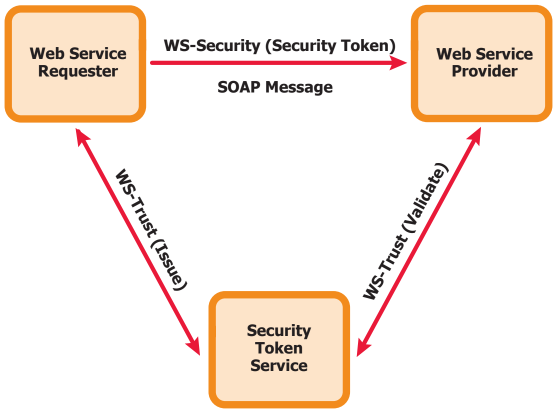
Confidentiality and message integrity are addressed via XML Encryption and XML Signature. XML Signature and XML Encryption are relatively mature standards developed within the W3C. WS-Security defines specific bindings for applying XML Signature and XML Encryption to SOAP.

Authentication and authorization are addressed via different profiles for conveying security tokens in WS-Security headers. Security tokens can contain identity information that allows the Web Service Provider to authenticate the identity of the user related to the SOAP request. A valid security token allows a Web Service to make appropriate authorization decisions based on the subject of the token. This is accomplished without requiring the user to re-authenticate directly to the Web service, in essence enabling single sign-on (SSO).

WS-Security defines profiles for securely conveying different types of security tokens. This includes profiles for Kerberos tickets, X.509 certificates, Username/Password tokens, SAML Assertions and XRML Licenses. Recall that Web Services are fundamentally integration gateway technologies, so by definition there is likely to be a number of different security tokens in the deployment mix.

WS-Trust

WS-Trust is an OASIS standard that defines a message protocol for the retrieval or validation of security tokens from a Security Token Service



WS-Security Passes Identity Information via Security Token Created by STS

(STS). Security tokens can then be conveyed in the WS-Security headers of a SOAP request via the applicable WS-Security token profile. WS-Security provides the ability to attach a security token to a message so that services make better security decisions. WS-Trust provides the architectural capability to communicate secured messages to Services across a heterogeneous environment.

SAML

The Security Assertion Markup Language (SAML) standard is also an OASIS product. SAML defines protocols and profiles for enabling identity federation within the constraints of different use cases. The most well known example of these use cases is browser-based secure Internet SSO.

The SAML specification also defines a security token called a SAML Assertion. A SAML Assertion is a secure and trusted identity statement that can be used with WS-Security (via the WS-Security SAML Token Profile) to facilitate authentication, SSO and authorization between a Web Services Requester and a Web Services Provider.

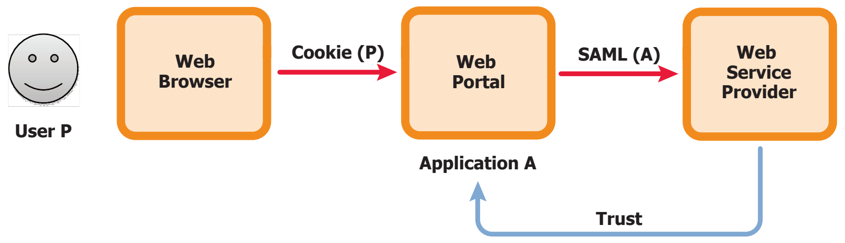
SAML is well adopted in facilitating browser-based SSO, but what happens to the security posture of the application once the user clicks "Submit" in the browser? The Web server likely communicates with a wide variety of back-end integration technologies that use Web Services. WS-Security and WS-Trust provide a trust backbone to move these tokens around the system in a policy-based way.

In summary, a Web Service Requester uses WS-Trust to communicate with a Security Token Service (STS), which issues a trusted SAML Assertion that represents the user's identity. The Web Service Requester includes the SAML Assertion in the WSS headers of each SOAP request. The STS becomes a trust aggregator and arbiter.

To validate the SAML Assertion, the Web Service Provider needs to trust the STS that issued the SAML Assertion. Alternatively, the Web Service Provider can use an STS to validate the SAML Assertion itself. The SAML Assertion contains the relevant information that allows the Web Service Provider to make appropriate authentication and authorization decisions.

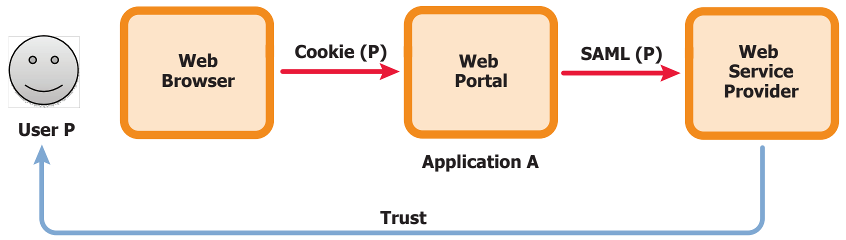
Identity-Enabled Web Services

Most Web Service Providers are a new interface into legacy systems. These systems will continue to retain authorization models that require some form of trusted user identifier or trusted user role that are used to make authorization decisions before processing the request. As such, a Web Service Provider needs to address the following question before it can process a SOAP message: "Who is this SOAP message about and how can I trust that this is the case before I allow this request to be processed?"



Web Service Provider Must Implicitly Trust the Application

This trust establishment can be accomplished in a number of ways. The Web Service Provider can choose to authenticate and trust the application that sent the SOAP message. In this case, the Web Services provider trusts that the Web Service Requester application can make requests on behalf of the user, and that the Requester has validated the identity of the user. In essence, the



Ideal Situation: Web Service Provider Trusts the User, Enabling End-to-End Security

Provider trusts all messages from the Web Service Requester implicitly. This means that any strong user identity that is established through SAML browser-based Internet SSO is lost between the Requester and the Provider.

An alternative approach is for the Web Service Provider to authenticate and trust the entity that invoked the original request, enabling end-to-end security. In this case, the Web Service Provider seeks confirmation that the actual user originated the request, implying that user identity must be passed through to the Web Service Provider.

A SAML Assertion can represent the identity of the Web Service Requester application sending the SOAP message, or it can represent the identity of the user that originated the request. In either case, Web Service Requesters must acquire a security token to access Web Services in different security domains. This is accomplished in a number of ways:

- Web Service Providers can trust and validate different security tokens from different Web Service Requesters in different security domains. This becomes expensive as the number of Web Service Requesters in disparate security domains increases. For example, a Web Service Provider trusts a user's Kerberos ticket from Web Service Requesters running on Windows desktops and a session cookie from users accessing a Web portal that initiates SOAP requests.
- Web Service Requesters can acquire different security tokens for accessing Web Service Providers in different security domains. This becomes expensive as the number of Web Service Providers in disparate security domains increases. For example, a Web Service Requester needs an X.509 certificate to access a Web Service in security domain A and a proprietary token to access a second Web Service in security domain B
- Web Service Requesters and Web Service Providers can use a common, shared security token format and perform a local translation from their local security token into the common (shared) format. This standards-based approach is more flexible, scales better, and costs less.

PingIdentity™

While all of these methods for handling security tokens are valid, the industry is converging on the SAML Assertion as the shared, common security token format for Web Services. The SAML Assertion allows Web Service Clients and Web Service Providers in different security domains to securely communicate identity information that can then be used for authentication and authorization. In addition, the flexibility of the SAML Assertion allows Web Service Providers to support different authorization models while only handling a single security token format.

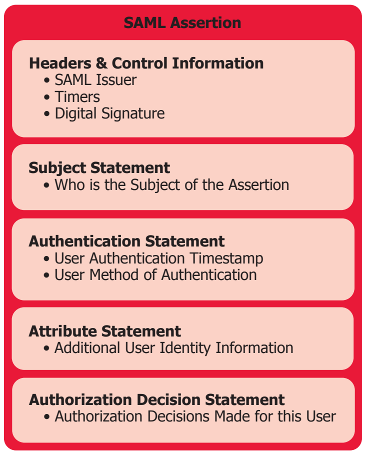
The Role of the SAML Assertion

SAML is an OASIS standard that defines XML protocols and profiles for enabling identity federation. The most well known example is secure Internet SSO between browsers and Web applications.

The SAML specification defines a security token called a SAML Assertion. The SAML Assertion is a secure and trusted identity statement that can be used with WS-Security to facilitate authentication, SSO and authorization between a Web Service Requester and a Web Service Provider. WS-Security defined a specific profile that describes how to use SAML Assertions with WS-Security. This is known as the WS-Security SAML Token Profile.

As a security token, the SAML Assertion has many unique properties that make it extremely useful for Identity-Enabled Web Services. The SAML Assertion:

- Is an open standard
- Leverages XML syntax in an XML document
- Supports heterogeneous environments
- Is flexible and extensible
- Conveys authentication, attribute and authorization information
- Is optionally self-validating


SAML Assertion
Headers & Control Information
• SAML Issuer
• Timers
• Digital Signature
Subject Statement
• Who is the Subject of the Assertion
Authentication Statement
• User Authentication Timestamp
• User Method of Authentication
Attribute Statement
• Additional User Identity Information
Authorization Decision Statement
• Authorization Decisions Made for this User
Elements of a SAML Assertion

Alternative security tokens all present significant challenges when you attempt to use them to identity-enable Web Services:

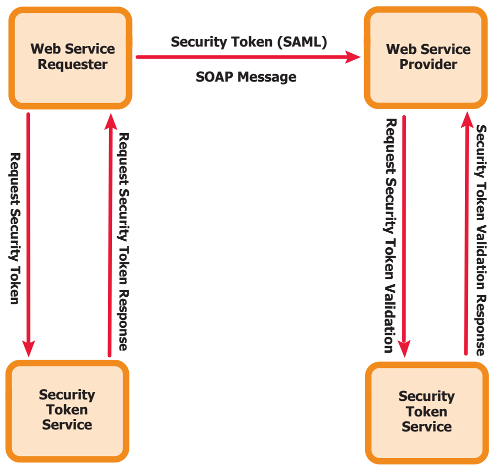
Session Cookies—all of the session and SSO tokens used by Web Access Management products are proprietary in nature. They all convey similar information but have different proprietary syntaxes and are secured in different ways. In fact, one of the fundamental drivers for the original creation of the SAML Assertion specification was that the Web Access Management vendors needed to agree on a common security token format to enable interoperability between their products.

Kerberos Tickets—while Kerberos is an open standard, it was never designed to cross domains and namespaces. It has proven difficult to establish an environment where Kerberos tickets issued in one security domain can be trusted for accessing resources in a different security domain. This has proven to be true for many reasons, including network security and firewall issues, lack of information on how to securely enable Kerberos cross-realm trust, an expectation that all security domains must understand Kerberos, and the lack of ability to simply convey additional identity information within the Kerberos ticket.

X.509 Certificates—the Public Key Infrastructure (PKI) trust model and associated X.509 certificates have proven to be problematic when used at scale to identify people. Issuing a large number of X.509 digital certificates to identify every individual in a security domain is expensive and difficult to manage and maintain. Practically speaking, PKI and X.509 lack the ability to convey directory-centric information such as dynamic attributes and group/role membership.

The SAML Assertion relies on X.509 certificates and the PKI trust model to establish trust between different security domains at an organizational rather than an individual level. As such, only a limited number of digital certificates and associated keys are required to facilitate inter-domain SSO. An X.509 certificate issued by Certificate Authority Y and states "I am Security Domain A," is used to sign SAML Assertions that convey user identity information that states: "I am User X from Security Domain A."

The SAML Assertion will become the lingua franca for Identity-Enabled Web Services. It should be considered the common security token format that bridges identity and trust between Web Service Requesters and Web Service Providers in different security domains, enabling end-to-end security.



Security Token Services Arbitrate Between Multiple Security Domains using SAML

Security Token Service – Federation for Web Services

A Security Token Service (STS) is a system role defined by the WS-Trust specification. Web Service Requesters interact with an STS to request a security token for use in SOAP messages. Web Service Providers interact with an STS to validate security tokens that arrived in a SOAP message. An STS arbitrates between different security token formats such that SOAP messages can be executed with knowledge of the complete "security context" of the request.

An STS is also an implementation of federated identity for Web Services. Identity federation processing enables trust and allows the integration of identity information between different security domains. At the core of an STS is the ability to execute identity federation processing.

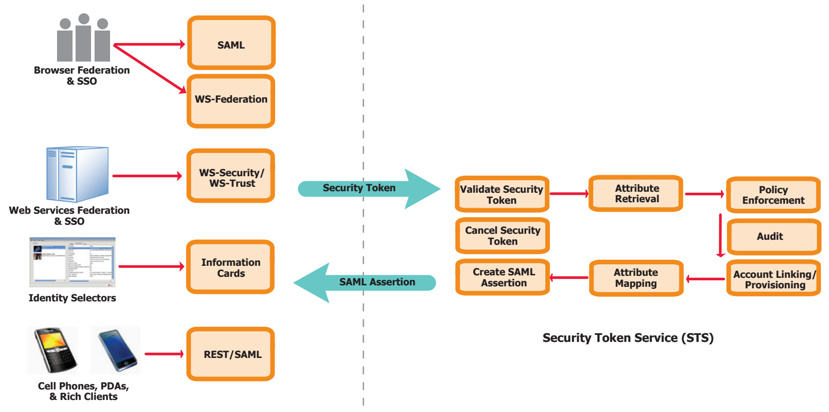
Identity federation processing is a multi-step and repeatable process that is performed regardless of the federation protocols and profiles that are used to move the resulting security tokens around the network. Identity federation processing consists of the following three fundamental tasks:

- Authentication and Trust Establishment
  Authentication is required to verify the identity (conveyed as a security token) of the user in one security domain before creating a different security token that is trusted by the partner security domain. In effect, trust is established through the exchange of security tokens. The validation and creation of security tokens will generally require cryptography to ensure the security tokens can be trusted and remain secure, creating further need for appropriate certificate and key management.

- User Identity Mapping
  A user can be known by different identifiers and different roles in different security domains. Identity mapping facilitates the ability for a security token to contain the correct user identity information for use in different security domains. This identity information can be retrieved from the

token itself or from external data sources such as an LDAP directory. Identity information can include attribute values, such as email address, role, name, address, favorite color, etc. The information can be used to personalize the user experience or to make authorization decisions within the application.

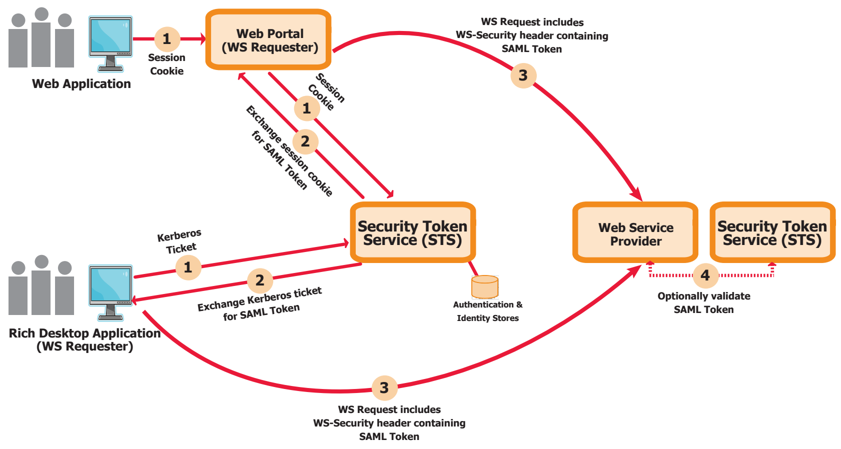
- Authorization, Auditing and Provisioning
  When enabling federation, authorization, auditing, and provisioning—each with unique security functionality—are all mechanisms that must



Security Token Services Translate Security Tokens into SAML Assertions

be considered. Auditing ensures that the appropriate user and partner information is logged and persisted for SLA and compliance requirements. Federated authorization and provisioning are additional processes that can optionally occur during federation processing. Federated authorization can be basic, ensuring that the correct role identifier exists in a security token, or more complex, facilitating the reuse of business policy information between different security domains. Federated provisioning provides for the dynamic addition, updating and deletion of user identity information in identity stores in different security domains. Lightweight, federated provisioning capabilities facilitate business functions, such as account linking and automated user account creation.

As an example, a Web Service Requester could ask the STS to issue a SAML Assertion that represents the secure identity of the user or application that refers to the body of the SOAP message. The Web Service Requester is required to establish some form of proof that it is actually authorized to request the SAML Assertion on behalf of the user. In the case of a rich desktop application, this could be a Kerberos ticket issued by Microsoft Active


Web Application
1
Session Cookie
Web Portal (WS Requester)
WS Request includes WS-Security header containing SAML Token
3
Session Cookie
1
2
Exchange session cookie for SAML Token
Security Token Service (STS)
Web Service Provider
Security Token Service (STS)
4
Optionally validate SAML Token
Authentication & Identity Stores
Kerberos Ticket
1
2
Exchange Kerberos ticket for SAML Token
Rich Desktop Application (WS Requester)
3
WS Request includes WS-Security header containing SAML Token

Security Token Services Translate Security Tokens into SAML Assertions

Directory. In the case of a Web portal, this may be a Session Cookie issued by a Web Access Management product. The Web Service Provider can process SOAP requests from arbitrary Web Service Requesters with the identity information of the user conveyed in a standard and secure manner.

Independent Security Token Service

To scale effectively while reducing administrative overhead, identity federation should be implemented as a standalone federated identity server/STS to provide an independent layer. This architecture enables:

- Agility as enterprises become more integrated and start to scale the number of users, applications and partners that need to be federated
- Centralized trust to reduce risk, centralize control and simplify compliance
- Integration of security domains that are internal and external to the organization
- Flexibility to offer a variety of hosting options for Web Services

A standalone federated identity server/STS consolidates federation processing and administration in one place, creating one doorway by which all identities in a security domain exit and all identities outside of a security domain enter. Further, the Federation Server/STS:

- Centralizes partner management and trust management
- Arbitrates between SAML Assertions and security token formats for different security domains both within and external to an organization
- Allows for linear scaling of processing capability via the addition of federation servers
- Centralizes audit information to meet compliance and SLA requirements
- Allows for additional federation protocols and profiles to be added as necessary

Conclusion

The notion of identity-enabled or federated Web Services implies that trusted user identity information is included in each SOAP request in a secure and standard way, enabling end-to-end Web Services security. As with secure Internet SSO, federated Web Services can now be completely standards-based via the combination of WS-Security, SAML and WS-Trust.

Federated Web Services will alleviate the need for point-to-point trust models and user re-authentication. The SAML Assertion should be the security token format that is used to identity-enable Web Services, enabling SSO via a standard security token trusted by all security domains. Organizations should consolidate the creation and the validation of these SAML Assertions within an independent, standalone Security Token Service. The STS should be implemented as an independent layer to scale effectively, reduce administrative overhead and aggregate trust management.

The concept of user session no longer has to end at the application. Instead, trusted user identity information can now follow transactions wherever they go throughout the Web Services environment.

About Ping Identity Corporation

Ping Identity is the market leader in Internet Identity Security, delivering on-premise software and on-demand services to more than 300 customers worldwide. For more information, dial U.S. toll-free 877.898.2905 or +1.303.468.2882, email sales@pingidentity.com or visit www.pingidentity.com.

PingIdentity™