PANEL Proposal: **Identity, Access Control and PMI for the Enterprise**

Panel Participants:
Anil Saldhana, RedHat Inc, Oasis Technical Committee (XACML, SAML)
Sunil Madhu, Securent Inc, Oasis Technical Committee (XACML)
Stefan Guilhen, Red Hat Inc

Topic: ***Extending Identity and Access Control in a Java based Enterprise using Oasis SAML and XACML***
Speaker: Anil Saldhana
Time: 45 Minutes

About the Speaker: Anil Saldhana represents Red Hat Middleware on many Security related standards at the W3C (Security Context), Oasis (XACML, SAML, EKMI) and the Java Community Process (JCP). He has spoken at various conferences on security related topics. He is the Leader for JBoss Security and Identity Management at Red Hat Middleware.

Java based Enterprises typically run Enterprise applications that are based on the Java EE standard from the Java Community Process (JCP). The Java EE standard provides specifications for web, Enterprise Java Beans (EJB), Web Services and other components.

The Java EE specification provides coarse grained access control based on roles. Since the need of the hour in enterprises is fine grained or context based access control in addition to RBAC, there is a necessity to incorporate standards such as Oasis XACML to extend the Java EE specification.

The Java EE specifications have not really given any consideration to federated identity needs, which is increasingly becoming a necessity in the enterprise. Hence there is a need for extensions to the Java EE specification using standards such as Oasis SAML.

In this presentation, we will see how the authentication process can be made aware of a SAML based identity and then utilize XACML policies for fine-grained access control in a Java EE environment. SAML Attribute Statements can also be utilized to provide the user attributes for access control.

Topic: ***Service based Entitlements Management using Oasis XACML in an Enterprise***
Speaker: Sunil Madhu
Time: 45 Minutes

About the Speaker: Sunil Madhu is a Senior Solutions Architect at Securent Inc. Securent is a voting member in Oasis XACML Technical Committee.

The traditional approach to security has been to embed authentication and authorization logic in each and every application. Since business requirements vary from one application to the next, this trend has led to security silos that are often inconsistent, opaque and in some instances, cryptic. Several years ago, the industry began to view security as a reusable infrastructure component and this led to the abstraction of embedded authentication to a service based approach -- Single Sign-On was more of a side effect than a feature of turning authentication into a service. Identity Management and Provisioning was a logical off-shoot of this advent, since managing and distributing SSO identities and their granular attribute components throughout enterprise-wide data stores from a centralized service made sense. Since enterprise technology trends have been mimicking trends in Internet technologies, B2B and B2C collaboration across the internet made a lot of sense given that traditional mechanisms of data-exchange such as EDI were comparably more complex and expensive to implement. Thus around about the same time, the need to federate access emerged; as did standards which made federation a reality.

The last piece of the puzzle then is turning authorization into a service; both coarse and fine grained. SSO platforms have dealt with coarse-grained authorization for web-based applications to some degree of success. However, few if any of these SSO solutions have tried to tackle authorization for non-web-based, COTS and legacy applications. Identity Management systems on the other hand help to provide us a meta-view of an individual's attributes and how these are distributed across the enterprise. However that meta-view in itself is insufficient in helping to address the challenges of fine-grained authorization since business logic embedded in applications consume elements of an individual's identity along with run-time and distributed application-specific data in order to answer the basic Entitlement Management challenge; viz., what can someone do once authenticated to an application? Identity Management systems' design patterns do not have visibility into that business logic and therefore these systems are not suited to address the Entitlement Management challenge.

Along come mature standards -- XACML (Extensible Access Control Markup Language) to be more specific, and the Entitlement Management technologies that use that standard to boot. Securent with its award-winning, industry-recognized "best-of-breed" Entitlement Management Solution (EMS) has made an attempt to stand out. Securent EMS is a tool that helps an enterprise turn Entitlement Management -- and embedded authorization logic -- into a service by decoupling programmatic authorization logic for each application from the functional logic. Based on Service Architecture, applications can move authorization logic from an opaque programmatic silo paradigm to a transparent declarative paradigm and as a result gain efficiencies in managing, auditing and reporting on authorization right across the enterprise stack; from networks to applications and their data. Application owners still retain control of the authorization logic which can be changed via configuration at the speed of change in their business requirements. Developers needn't concern themselves with maintaining authorization-related code and can focus on managing functional code. C-level executives can rest assured that the applications and services within their enterprise are compliant with corporate and regulatory policies. Applications remain secure. The Entitlement Management service can be tuned to meet OLA and SLA requirements. Everyone wins.

Topic: ***Privilege Management Infrastructure (PMI) for the Enterprise***
Speaker: Stefan Guilhen, Red Hat Inc
Time: 45 Minutes

About the Speaker: Stefan works as a Senior Software Engineer at Red Hat Middleware. He has done research on PMI and has presented PMI related work at various conferences.

The Public Key Infrastructure defines the well known X509 Public Key Certificate, or PKC. The PKC is a digitally signed document that binds a public key to an identity. The main goal of the PKI is to provide a strong authentication mechanism based on the X509 PKC, and this model is widely used by security protocols designed to guarantee integrity and confidentiality in communication channels (SSL, TLS).

In practice, however, enterprise applications requirements revealed a need to store more information on the PKC besides just the holder's identity. More specifically, a set of extensions was designed to allow for the insertion of authorization information (roles, groups, credentials) in the PKC, so that target applications could retrieve a subject's authorization information from the same certificate it used to authenticate that subject. At first, this approach seems to simplify considerably the management of a subject's attributes, but it has at least two major drawbacks:

* The lifetime of the attributes is usually different from the lifetime of the public key. The set of attributes of an individual changes much more frequently than the public key. As a result, every time an attribute is added, changed, or removed, the entire PKC has to be revoked and a new one has to be issued, even though the public key wasn't compromised in any way.
* The Certificate Authority, that issues the PKCs, usually does not have enough information to determine the set of attributes a subject may have on a particular domain. As a result, it will either issue a PKC with an incomplete attribute set, or it will have to interact with other services to gather information, which increases the complexity of the authority.

These problems were recognized, and resulted in the creation of a little known extension of the X.509 standard: the Privilege Management Infrastructure. The central element of the PMI is the X.509 Attribute Certificate that binds a set of attributes to an identity, and is issued by an Attribute Authority. The PMI establishes a clear separation between the authentication and authorization processes, by leaving the management of authorization attributes to authorities that actually own this information, and by removing the authorization information from the PKCs. As a result, the X.509 ACs can be revoked independently of the associated PKC.

It is common to find enterprise applications that support the use of PKCs for client authentication. The authorization engines, however, are usually implemented using non-standard data structures, often pulling information from some storage service to decide if access should be granted or not. The use of X.509 ACs promotes interoperability, as it defines a standard data structure to store authorization and other attributes. Besides that, as the X.509 ACs are digitally signed documents, they offer integrity protection for the data it holds. This adds security to the authorization engine, since it can easily verify that the information contained in a X.509 AC was indeed issued by a trusted authority, and that this information has not been changed afterwards.

Even though handling certificate revocation can be hard sometimes, the usage of the X.509 PKCs together with the X.509 ACs allows for the establishment of a strong and flexible security infrastructure for enterprise applications.