Intranet Single Sign-On for Windows and Tivoli Access Manager

Using Kerberos/NTLM SPNEGO with WebSEAL

Tim Bartley
Paul Ashley

May 01, 2003

Microsoft Windows based intranets provide the ability to use desktop credentials to sign-on to intranet infrastructure based on Microsoft Internet Information Services (IIS). This is implemented using Microsoft’s SPNEGO HTTP authentication protocol to sign-on using NTLM or Kerberos credentials. Until IBM Tivoli Access Manager for e-business (TAM) 4.1 was released there was no way to achieve the same sign-on to TAM’s WebSEAL web resource authorization engine. With TAM 4.1 this sign-on can be achieved by combining the SPNEGO sign-on capability of TAM Plugin for IIS with e-Community single sign-on capabilities of WebSEAL. This article describes in detail the configuration steps required to make this work.

Introduction

Microsoft Windows based intranets provide the ability to use desktop credentials to sign-on to intranet infrastructure based on Microsoft Internet Information Services. This is implemented using Microsoft’s SPNEGO HTTP authentication protocol to sign-on using NTLM or Kerberos credentials.

Until IBM Tivoli Access Manager for e-business 4.1 was released there was no way to achieve the same sign-on to TAM’s WebSEAL web resource authorization engine.

With TAM 4.1 this sign-on can be achieved by combining the SPNEGO sign-on capability of TAM Plugin for IIS with e-Community single sign-on capabilities of WebSEAL

This article describes in detail the configuration steps required to make this work.
Figure 1: Single Sign-On From Windows Workstation to IIS

![Diagram of Single Sign-On From Windows Workstation to IIS]

Single Sign-On to Access Manager

As an extension to the single sign-on function described above, Intranet users who have an Active Directory domain login require single sign-on to Access Manager. Without single sign-on, users login to their workstation AND to Access Manager (two logins).

Figure 2: Intranet SSO Requirements

![Diagram of Intranet SSO Requirements]

Examining Figure 2, users will login to Active Directory and receive credentials at their Windows Workstation that can be used indicate they have been logged on. The user would like to be able to present these credentials to WebSEAL, and WebSEAL automatically log the user on. Hence the user desires single sign-on between the Active Directory Domain and Access Manager’s WebSEAL.
Solution

The recommended solution is to use the AM 4.1 Web Server Plug-In in an eCommunity single sign-on configuration with WebSEAL. The Web Server Plug-In is configured as the Master Authentication Server (MAS).

Figure 3: Use of Web Server Plug-In

The sequence is as follows:

1. A user logs in to their workstation authenticating to their Active Directory domain and receives the Kerberos TGT. If the user does not have a domain login then they login to their workstation and will have no Kerberos credentials.

2. The user tries to access a protected application behind WebSEAL (one requiring authentication). WebSEAL will re-direct the user to the Master Authentication Server which is the Access Manager Web Server Plug-In on IIS.

3. The Web Server Plug-In has been configured for SPNEGO authentication. If the user has the correct Kerberos credentials then they will have single sign-on to the Web Server Plug-In. In this case the Web Server Plug-In will communicate with the LDAP server to build the user’s AM credential. (Note that the user’s username needs to be the same on Active Directory and LDAP. This configuration assumes a unique username for each user). The user is then re-directed back to WebSEAL. WebSEAL will build an AM credential for the user.

4. For user’s unable to authenticate using SPNEGO the authentication at the Web Server Plug-In will fail and the user will be re-directed back to WebSEAL indicating the failure. WebSEAL will then prompt the user to authenticate. WebSEAL will build an AM credential for the user.

5. WebSEAL will now authorize access to the resources it is protecting.

Configuring SPNEGO Authentication for the TAM Web Server Plug-in

To set up the SPNEGO authentication module of the TAM Web Server Plug-in for IIS edit the pdwebpi.conf configuration file. This file is found in <PDWebPILInstallDir>\etc, where
<PDWebPIInstallDir> is the TAM Plug-ins for Web Servers installation directory which is c:\Program Files\Tivoli\PDWebPI\etc by default.

The TAM packages required on this host are:

- TAM Run-Time Environment, Version 4.1 (plus pre-reqs)
- TAM Plug-in for Web Servers, Version 4.1
- TAM Plug-in for Microsoft Internet Information Services, Version 4.1

We need to set up authentication and session handling in order to handle SPNEGO. We'll assume that we're using a dedicated IIS server for our SPNEGO authentication so this information is specified in the [common-modules] stanza of the pdwebpi.conf configuration file. If you are sharing the role of this IIS server then the stanza you need to configure is the virtual host specific stanza.

The only authentication method needed to be supported by the Web Server Plug-in is SPNEGO however only Internet Explorer supports SPNEGO authentication. If a mix of browsers is required to be supported then Basic Authentication should be specified as well. Since we want to give preference to SPNEGO we list the SPNEGO authentication module ahead of the BA module. So our authentication mechanisms will look like:

```plaintext
authentication = spnego
authentication = BA
```

We need to include SPNEGO in our list of session modules. It is used to maintain session state during the SPNEGO authentication. Once the SPNEGO authentication is complete we need to hand over session maintenance to another scheme – we'll use session cookies since this can be shared between BA and SPNEGO authenticated sessions. Our session module configuration will then look like:

```plaintext
session = spnego
session = session-cookie
```

We also require the account management post-authorization module to handle logouts and password change operations should they be necessary. So we have:

```plaintext
post-authzn = acctmgmt
```

The only configuration required for the SPNEGO module is whether or not to let IIS handle the SPNEGO authentication. When IIS handles the SPNEGO authentication itself IIS knows who the user is. This is useful if IIS is hosting an application that needs Windows user credentials to identify the user. If IIS does not handle the SPNEGO authentication itself the Web Server Plug-in will do it itself. In this case the user appears anonymous to IIS. For our purposes we don't need IIS to see the user identity so we'll specify that the SPNEGO authentication should not be delegated to IIS. We do this by setting the web-server-does-authn to no in the [spnego] stanza of the pdwebpi.conf configuration file:

```plaintext
[spnego]
web-server-does-authn = no
```
In order for the user to be able to single sign-on they will need a user in both the Windows Active Directory user registry and the TAM user registry. No password information need be maintained for the user in the TAM user registry unless they’re going to use the BA authentication method we configured because the user chooses not to use Internet Explorer. Of course if the TAM user registry is the same Active Directory being used for desktop sign-on then this replication of user information is unnecessary.

Starting (or restarting) the Access Manager Plug-ins for Web Servers service on the IIS machine will pick up the configuration changes that you’ve just made. You should now use Internet Explorer to access resources protected by the IIS without needing to specifically sign-on to TAM.

Before IE will silently sign a user on to a web site, the web site must be part of the Trusted Internet zone. Also remember that if IE 6 users uncheck the “Enable Integrated Windows Authentication” checkbox in the Security section of the Advanced Internet Options for IE then they will be prompted for login information to sign on to the MAS.

**Configuring eCSSO between the Web Server Plug-in and WebSEAL**

The e-Community Single Sign-on mechanism (eCSSO) supported by TAM allows single sign on between web servers in different domains. Domains in this context can mean both different DNS domains – that is domains that cannot share cookies to achieve single sign-on – and completely separate Access Manager domains.

Despite it's usefulness for this high level inter-domain single sign-on it is a light weight mechanism that is suitable for use in this case since it allows nomination of a Master Authentication Server (MAS) that authenticates a user and can then vouch for that users identity.

In our case, the IIS server with the TAM plug-in takes the role of the MAS. When WebSEAL sees a request from a new user it can then redirect that user to the MAS which can perform the SPNEGO based authentication. The MAS will then redirect back to the WebSEAL, vouching for the users identity. The WebSEAL can authorize the user for access to the resources it is protecting and grant or deny this access as appropriate.

The mechanism used to vouch for a user’s identity is to include a short-lived encrypted token when redirecting the client back to the WebSEAL. In order be able to encrypt and decrypt this token the two servers need to share a secret key. This secret key is created either by using the pdwpi-cdsso-key-gen program that comes with TAM Plug-ins for Web Servers or the cdsso_key_gen program that comes with WebSEAL. The two different versions of this program produce compatible key files. Once created on one system the file should be copied to the other. To create the key issue a command like:

```
pdwpi-cdsso-key-gen c:\keys\company.com.key
```

where the parameter to the program is the filename in which to store the key. We’ll refer to this location when actually configuring eCSSO for the plug-in and WebSEAL.
For the purpose of the configuration we'll assume that the host name of the IIS server is plugin.company.com and that the host name of the WebSEAL is webseal.company.com.

To configure the IIS server as the eCSSO MAS we need to register the ecsso module as a post-authorization module and then configure it. Since we are the MAS we do not need to register this module as an authentication module (though it does no harm to do so).

```
[common-modules]
pot-authzn = ecsso
```

To identify this server as the MAS we set a parameter in the [ecsso] configuration stanza:

```
[ecsso]
is-master-authn-server = yes
```

We need to pick a name for our e-community – the default name chosen by Web Server Plugins is ecsso, WebSEAL doesn't specify a default name. All participants in the e-community must have the same idea of the e-community's name - we'll pick company.com as our e-community name.

```
e-community-name = company.com
```

To specify the secret key to use when encrypting tokens for consumption by webseal.company.com we add an entry to the [ecsso-domain-keys] stanza:

```
[ecsso-domain-keys]
company.com = c:\keys\company.com.key
```

Restarting the Access Manager Plug-in for Web Servers service will now pick up the new configuration and begin acting as the Master Authentication Server.

To configure WebSEAL to use the plug-in as the Master Authentication Server we must copy the key we generated earlier. If using FTP use binary mode to transfer the file. We'll assume WebSEAL is running on a UNIX like platform and that the key file is stored in /keys/company.com.key.

Most of the configuration required for the WebSEAL is within the [e-community-sso] stanza. To enable the eCSSO functionality of WebSEAL we must specify the set of protocols that we may receive requests that trigger an eCSSO authentication. It is strongly recommended that only HTTPS support be enabled as the URL that the client is redirected to contains an encrypted token that could be replayed if captured by a malicious network user. The encrypted token's lifetime is only short – 5 minutes by default – to allow for the system clocks on the MAS and the WebSEAL to be slightly out of synchronization. The token could only be replayed for this period of time but it is best to remove any possibility by only permitting eCSSO operation over HTTPS.

```
[e-community-sso]
e-community-sso-auth = https
```
We need to indicate that this WebSEAL is not itself the master authentication server and also where the master authentication server is:

```
is-master-authn-server = no
master-authn-server = plugin.company.com
```

We specify the e-community name to match the name we specified for the MAS:

```
e-community-name = company.com
```

We specify our key in the [e-community-domain-keys] stanza:

```
[e-community-domain-keys]
company.com = /keys/company.com.key
```

WebSEAL uses CDAS’s to perform the eCSSO token decoding. The single sign-on token consumption CDAS (ssoconsume) is not enabled by default so we need to enable it. We'll assume a Solaris installation which has the “.so” library extension but the ssoconsume CDAS ships on all WebSEAL platforms so this attribute value can be easily changed according to the WebSEAL platform you are using.

```
[authentication-mechanisms]
ssoconsume = /opt/pdweb/lib/libssoconsume.so
```

In order to cater for the possibility of processing failures during the eCSSO authentication process it is recommended that a secondary authentication mechanism be enabled for the WebSEAL – for example

```
[forms]
forms-auth = https
```

This allows the WebSEAL to failback to an alternative authentication strategy when logging a user in should the eCSSO mechanism fail for any reason.

**Using Active Directory for the TAM User Registry**

If the TAM user registry is the same Active Directory that is used for authenticating desktop logins then there is no longer any need to replicate user information between Active Directory and the TAM user registry.
The only disadvantage of this is that as of TAM 4.1 this requires that all TAM infrastructure be hosted on Windows 2000 Advanced Server machines and may not be run on UNIX machines. This includes both the WebSEAL shown in the architecture above and the TAM Policy Server (not shown).

**Configuration Summary**

To summarize the configuration file changes required to enabled SPNEGO SSO to WebSEAL via the TAM Plugin for IIS:

```plaintext
[pdweb-plugins]
virtual-host = plugin.company.com

[common-modules]
authentication = spnego
authentication = BA
session = spnego
session = session-cookie
post-authzn = ecsso
post-authzn = acctmgmt

[BA]
realm = "Company.Com Intranet"

[spnego]
web-server-does-authn = no

[ecsso]
is-master-authn-server = yes
e-community-name = company.com

[ecsso-domain-keys]
company.com = c:\keys\company.com.key
```
webseald.conf

[e-community-sso]
ecsso-authn = https
is-master-authn-server = no
e-community-name = company.com
master-authn-server = plugin.company.com

[e-community-domain-keys]
company.com = /keys/company.com.key

[authentication-mechanisms]
ssoconsume = /opt/pdweb/lib/libssoconsume.so

References

For information on installing WebSEAL:
*IBM Tivoli Access Manager for e-business 4.1, WebSEAL Installation and Configuration Guide*

For information on configuring eCSSO and other aspects of WebSEAL:
*IBM Tivoli Access Manager for e-business 4.1, WebSEAL Administrators Guide*

For information on installing and configuring the Web Server Plug-ins:
*IBM Tivoli Access Manager for e-business 4.1, Plug-in for Web Servers User's Guide*

© Copyright IBM Corporation 2003

Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)