Pulse Secure Access
Kerberos Constrained Delegation
Contents

1. BACKGROUND 3

2. SETTING UP CONSTRAINED DELEGATION 5
   2.1 Active Directory Configuration 5
       2.1.1 Create a Kerberos Constrained Delegation (KCD) User Account 5
       2.1.2 Enable the Delegation tab for the created user 6
       2.1.3 Add the Services. 9
   2.2 Web Application server configuration 12
       2.2.1 Configure the IIS Server. 12
       2.2.2 Web server - protocol negotiation. 14
   2.3 Setting up Pulse Secure Access for Constrained Delegation 15
       2.3.1 Web SSO General Configuration. 15
       2.3.2 Set up Constrained Delegation 17
       2.3.3 Setting up SSO Policies 19
       2.3.4 Constrained Delegation fallback flow 20

3. USER ACCESS WITH ONE-TIME-PASSWORD (OTP) 21

4. USER ACCESS WITH CERTIFICATE AUTHENTICATION 23

5. TROUBLESHOOTING 27
1. BACKGROUND

This section is based on information found at the following links.


If you want more detailed information on this topic please use them as a starting point.

What Does Kerberos Do?

The Kerberos security system protects electronic transmissions that get sent across the network. It does this by scrambling the information so that only the computer that's supposed to receive the information can unscramble it. In addition, it makes sure that your password never gets sent across the network, only a scrambled “key” to your password. Kerberos is necessary because there are people who know how to tap into the lines between the computers and listen out for passwords.

Why Kerberos is a Good Thing

Why did Microsoft bother with it? In previous versions of Windows NT, network authentication was handled by NTLM. Why make a change? How is Kerberos better than NTLM?

There are several answers. First, Kerberos provides several features that aren't available in NTLM. Delegation and mutual authentication are both available with Kerberos, but neither is possible with NTLM today. Also, Kerberos is typically faster than NTLM, since each NTLM client authentication requires a server to contact a domain controller. In Kerberos, by contrast, a client can supply the same ticket over and over, and the server can use just that ticket to authenticate the user. There's no need for the server to contact a domain controller each time a user needs to be authenticated. And finally, Kerberos is a multivendor standard, so it allows secure interoperability and the potential for single signon between the Microsoft world and other vendor environments.

Any way you look it, Kerberos qualifies as progress. It's nice to see this powerful, secure, but long-neglected protocol move into the limelight. After years of languishing in relative obscurity, Kerberos is about to go mainstream.


Constrained Delegation and Protocol Transition

There are two new extensions in the implementation of the Kerberos protocol in Windows Server 2003:

Protocol transition: The protocol transition extension allows a service that uses Kerberos to obtain a Kerberos service ticket on behalf of a Kerberos principal to the service without requiring the principal to initially authenticate to the Kerberos Key Distribution Center (KDC) with a credential.

Constrained delegation: The constrained delegation extension allows a service to obtain service tickets (under the delegated users identity) to a subset of other services after it has been presented with a service ticket that is obtained either through the TGS_REQ protocol, as defined in IETF RFC 1510, or in the protocol transition extension.

For more information about RFC 1510, see the IETF Web site (http://www.ietf.org).
The Protocol Transition Extension

The protocol transition extension allows a service to obtain a Kerberos service ticket to the service on behalf of a Kerberos security principal. No user credential is required for the transition. Applications may transition into Kerberos even though the actual authentication is done via another authentication system such as Radius, RSA SecureID, PKI/Certificates and other OTP systems.

The Constrained Delegation Extension

The reason why the constrained delegation extension is introduced in Windows Server 2003 can be best explained by describing the limitations in the Windows 2000 implementation of Kerberos delegation. In the Windows 2000 Kerberos delegation model, the Kerberos Key Distribution Center (KDC) does not limit the scope of services to which a Kerberos principal’s identity can be delegated. In other words, after a service account is trusted for delegation, it can request service tickets on behalf of an authenticated user to any other service accounts.

This delegation method does not provide precise mechanisms for an application to specify a subset of service accounts that it determines to be trustworthy for delegation. Essentially, applications are exposed to broader impersonation risks that may span across resource domains that have different levels of security policy requirements; some of the security policies may not be as strict as the applications security requirements. From the domain administrators point of view, it is too risky to enable unconstrained Kerberos delegation in the enterprise because there is no way to exclude untrusted servers from participating in delegation.

With constrained delegation, domain administrators can configure service accounts so that they delegate only to specific sets of service accounts.
2. SETTING UP CONSTRAINED DELEGATION

This section outlines how to set up Kerberos Constrained Delegation with the Pulse Secure Access product. This involves setting up an account in the Active Directory, setting up the Server hosting the services and finally configuring the Pulse Secure Access appliance.

2.1 Active Directory Configuration

This part explains the configuration steps needed on the Active Directory to enable Constrained Delegation with Pulse Secure Access.

2.1.1 Create a Kerberos Constrained Delegation (KCD) User Account

In order to get Constrained Delegation to work there a User account has to be created. This account must have the rights to do the Protocol Transition and Delegation. Essentially this is the account that has the rights to request a Kerberos Ticket on behalf of a user signing in to the Pulse Secure Access appliance.

Start by creating a new user in the Active Directory.

In this example the owacd is created as the account to provide Constrained Delegation Access to Outlook Web Access (OWA)
2.1.2 Enable the Delegation tab for the created user

Delegation is not enabled by default for a User account and need to be enabled. This involves the use of the SETSPN command-line tool that isn't included in any standard Windows 2003 installation.

Install the Windows Server 2003 Support Tools from the product CD or from the Microsoft Download Center (http://go.microsoft.com/fwlink/?LinkId=100114).

For more information about how to install Windows Support Tools from the product CD, see Install Windows Support Tools (http://go.microsoft.com/fwlink/?LinkId=62270).

If this is installed in your Windows 2003 server it can be found in C:\Program Files\Support Tools
Use the command: setspn -A HTTP/owacd junos\owacd

**NOTE:** in this example JUNOS is the Domain and owacd is the user account we just created.

This will enable the Delegation tab in the owacd properties.
If the Delegation Tab does not appear the Active Directory probably is running in mixed or native mode and need to be raised to Windows 2003 functional level.

**NOTE** that the following steps will change your Active Directory behavior and support for older Windows clients. If you are uncertain you should not raise the Domain Functional Level without checking if this has any impact to your environment since this step can not be reversed.

Once the Active Directory is at Windows 2003 functional level you can continue configuration

The Delegation tab will now be visible. Make sure to enable the “Trust this user for delegation to specified services only” and “Use any Authentication protocol”. Even though other selections might seem more accurate the “Kerberos only” options will not work since they do not enable Protocol Transition and Constrained Delegation.
2.1.3 Add the Services.

Since this is “constrained” delegation there is a need to specify the “Services” this applies to. Select “Add”.

![Screenshot of the Add Services window]

To allow services to be delegated for a user or computer, select the appropriate users or computers, and then click the services.

To select one or more user or computer names, click Users or Computers.

Available services:

<table>
<thead>
<tr>
<th>Service Type</th>
<th>User or Computer</th>
<th>Port</th>
<th>Service Name</th>
<th>Domain</th>
</tr>
</thead>
</table>

Select All

OK  Cancel
Use the Users or Computers button to select the Computer hosting these services.

In this example the Outlook Web Access service is hosted on the same server as the AD, so ADSERVER1 is selected. This could have been any other Server in the Domain though.

Note that Constrained Delegation does not support Services hosted in other Domains even though there is a trust relationship to those Domains.

Now add the “Services” on the selected Server.
Now review the settings and Apply / OK these settings.

You are now finished setting up the Active Directory part of the configuration.
2.2 Web Application server configuration

This section describes the configuration required on the server hosting the web application. In this example the application is Outlook Web Access (OWA) and it is hosted on the same server as the Active Directory. This could however be any web application hosted on any server within the domain.

2.2.1 Configure the IIS Server.

Open the Internet Information Services Manager and select the Web site you want to enable Constrained Delegation to. For OWA the Exchange sub-service is selected. NOTE the Authentication settings might be setup at the top level (Default Web Site) for all sub-services or it could be setup for each service like Exchange, Exadmin, ExchWeb, and Public in order to meet the needs to login properly to required applications.
Open the Properties and select the Directory Security tab. Select to Edit the Authentication and access control.

![Exchange Properties dialog box](image)

Make sure to disable anonymous access and enable Integrated Windows authentication (only).
2.2.2 Web server - protocol negotiation.

By selecting “Integrated Windows authentication” for Exchange (OWA) the Web Server protocol negotiation should be set to “Negotiate,NTLM” allowing for Kerberos authentication with potential fallback to NTLM for non-Kerberos capable devices. There might be situations where this has been changed and Kerberos Authentication is disabled or fallback to NTLM does not work. If this is unknown this can be checked by the following command in C:\inetpub\AdminScripts:

cscript adsutil.vbs get w3svc/WebSite/root/NTAuthenticationProviders

In this command, WebSite is a placeholder for the ID number of the Web site. The ID number of the default Web site is 1.
As you can see above the Authentication Protocol Negotiation is set to the required “Negotiate,NTLM”. If the returned string would have been something else, such as: (STRING) “NTLM” then only NTLM is enabled.

To set this back to its default, use the following command to set the string:

cscript adsutil.vbs set w3svc/1/root/NTAuthenticationProviders "Negotiate,NTLM"

2.3 Setting up Pulse Secure Access for Constrained Delegation

This section covers the steps required to enable Constrained Delegation to the previous defined application, OWA for any user connecting via the Pulse Secure Access SSL VPN appliance.

2.3.1 Web SSO General Configuration.

Start by setting up the Users > Resource Policies > Web > General.

Enable Kerberos SSO and add a Realm Definition. The Realm referred to is the Kerberos Realm. This is normally the same as the DNS Domain. In this example JUNOS.LOCAL
The Site Name field only applies, and can only be used, if your Active Directory is set up with Sites.

An Active Directory site object represents a collection of Internet Protocol (IP) subnets, usually constituting a physical Local Area Network (LAN). Multiple sites are connected for replication by site link objects.

Sites are used in Active Directory to enable clients to discover network resources (printers, published shares, domain controllers) that are close to the physical location of the client, reducing network traffic over Wide Area Network (WAN) links as well as to optimize replication between domain controllers. This is also true for Kerberos so this filed would allow you to define the specific Site Name you wish to discover the KDC in. For each Kerberos realm, there can be only one site defined in the Pulse SA. In other words, it is not possible to have two entries of the same Kerberos realm but different site names. The site name should be the site that this Pulse SA resides in. If the box is deployed in Paris, the site name should be the site name of Paris, etc.

The purpose of Kerberos pattern list is to match hosts with realms when they are in “disjoint namespaces”. What it means is that the DNS domain name of a host is not a Kerberos realm. For example, the host is application.lab.junos.net, but the Kerberos realm for this server is JUNOS.LOCAL. In such cases there is no way for IVE to figure out the Kerberos realm without the help of pattern matching.

Finally the KDC filed. In here you can define the KDC, normally the same as the Active Directory, but this is optional, since the SA will lookup the service and find the KDC for the Realm and Site, if defined by using LDAP to the Active Directory. 2.3.2 Set
2.3.2 Set up Constrained Delegation

The next step is to set up the Constrained Delegation. The first thing needed is to create the Service List. This is done by uploading a text file with the servers listed.

Open up Notepad or similar program and create a file with the server name(s)

Select Edit

Select New Service List.

Select the text file you just created. After the file is uploaded you can select OK and close the Services List dialogue.
Now the Constrained Delegation can be completed.

Start by setting a Label. This is for internal use in the Pulse SA.

Next pick the Realm in the drop down box. This will be the Realms defined in the previous step so in this example JUNOS.LOCAL

Define the Principal Account and Password. This is the account you created for Constrained Delegation in the Active Directory earlier in this guide. Make sure you type the password correctly as defined in the AD

Finally select the Service List defined previously.
2.3.3 Setting up SSO Policies

Next step is to set up SSO policies to define for what roles and resources Constrained Delegation will be performed.

Go to Users > Resource Policies > Web > Kerberos/NTLM/Basic Auth and select New Policy.

Define a Policy name and define the Resources for which this policy applies. Select the Roles this SSO policy applies to and finally what SSO to perform.

Select Constrained Delegation and use the drop-down list to select the definition you want to use. Depending on the required behavior Fallback to Kerberos can be selected or not.
2.3.4 Constrained Delegation fallback flow

In the case of Constrained Delegation failure the Kerberos Fallback will allow the SA to prompt the user for Credentials.
3. USER ACCESS WITH ONE-TIME-PASSWORD (OTP)

Now let's test the configuration. In this example the Authentication server used to authenticate the user to the Pulse SA is a Radius based SMS One-Time-Password solution from Mideye. The user supplies the username and a PIN. As a result a SMS is sent to the user mobile phone with an OTP.
The Pulse SA presents the Challenge/Response dialogue to the user that enters the OTP and submits. If the OTP is correct the authentication is successful. As you can note the User has not presented any AD credentials in this process. The username has to match a user in the AD though in order for the Pulse SA to know, on-behalf of what user, it should fetch Kerberos Tickets for via Constrained Delegation.

User is logged in to Pulse SA and uses the link to OWA. Pulse SA performs Constrained Delegation by fetching the Kerberos TGT and the S4U2Self ticket via the owacd user account using Constrained Delegation. Thereby offering SSO to OWA for the user.
4. USER ACCESS WITH CERTIFICATE AUTHENTICATION

In this example the user holds a Certificate that is used for authentication to the Pulse SA. The Certificate is issued by an internal Microsoft CA and includes the following information.

Since the Pulse SA using Constrained Delegation need to know for which user the Kerberos tickets should be fetched. There must be information in the Certificate that matches the user in the AD. In a normal Certificate the SamAccountname is normally not included, but the UserPrincipalName (UPN) is there and can be used for this purpose.
When connecting to the Pulse SA with a browser the user is prompted to select what certificate to use for authentication. This might look different if TPM, smartcards or USB certificate stores are used, since they normally have their own corresponding software.

There might be a requirement for the user to provide a pin or password to be granted access to the certificate store.

Once a valid certificate is presented to the Pulse SA the session is established and the main portal page is visible.

Since Constrained Delegation use the <USER> attribute when performing SSO it is important to select the correct Certificate attribute to be mapped as this variable.

This is described later in this document covering the Certificate Authentication server in Pulse SA.
In the above Portal Page a custom notification message is configured under the Role UI options. This simplifies the trouble shooting when setting this up, making sure the correct value is used as the `<USER>` attribute when testing constrained delegation.

If this matches the user in the AD, SSO to OWA is performed.
As mentioned above the Certificate Server created need to set the correct Certificate attribute as the Username in the variable <USER>. This is controlled by the User Name Template when creating the Certificate authentication server in the Pulse SA.
5. TROUBLESHOOTING

If you experience problems with Constrained Delegation there are a few things you can check/verify before opening a case with PSGSC.

Synchronizing system times

Kerberos authentication requires that system time is synchronized. Kerberos rejects any authentication requests from a system or client whose time is not within the specified maximum clock skew of the Kerberos server. Because each ticket is embedded with the time it was sent to a principal, hackers cannot resend the same ticket at a later time to attempt to be authenticated to the network. The client also rejects tickets from a Kerberos server if its clock is not within the maximum clock skew set during network authentication service configuration. The default value is 300 seconds (five minutes) for the maximum clock skew.

Verify the time on the AD, Server and Pulse SA to make sure the skew is less than 5 minutes. A strong suggestion is to use NTP to avoid this issue.

Check the User and Password

Another common mistake is that the username/password for the Constrained Delegation account in the AD does not match the configuration in the Pulse SA Constrained Delegation settings. Verify and re-enter the password to make sure.
Check the Server name

Verify that the server you have defined in the Service List, the SSO Resource Policy and the AD user Delegation settings is the correct one and that it can be resolved via DNS.

Test to resolve the server name from the Pulse SA by using Maintenance > Troubleshooting > Tools > Commands > NSLookup tool

Check the User Access log

Successful Constrained Delegation

2009-02-10 11:18:52 - ive - [195.67.156.182] jgyllenhammar(KCD)[Mideye] - Primary authentication successful for jgyllenhammar/Mideye from 195.67.156.182


2009-02-10 11:19:00 - ive - [195.67.156.182] jgyllenhammar(KCD)[CD-Demo] - Web SSO: Fetched Kerberos S4U2Self Ticket Client: jgyllenhammar@JUNOS.LOCAL, Server: owacd@JUNOS.LOCAL, authtime: Tue Feb 10 11:18:54 2009, startime: Tue Feb 10 11:18:54 2009, endtime: Tue Feb 10 21:18:54 2009, endtime sec: 1234297134, current sec: 1234261140, Flags reserved: 0, forwardable: 1, forwarded: 0, proxiable: 0, proxy: 0, may_postdate: 0, postdated: 0, invalid: 0, renewable: 0, initial: 0, pre_authent: 1, hw_authent: 0, transited_policy_checked: 0, ok_as_delegate: 0, anonymous: 0


2009-02-10 11:19:01 - ive - [195.67.156.182] jgyllenhammar(KCD)[CD-Demo] - WebRequest ok : Host: adserver1.junos.local, Request: GET /exchange/ HTTP/1.1


Unsuccessful Constrained Delegation

2009-02-10 11:32:05 - ive - [195.67.156.182] jgyllenhammar@junos.local(Cert- CD)[] - Primary authentication successful for jgyllenhammar@junos.local/CertSrv from 195.67.156.182

2009-02-10 11:32:05 - ive - [195.67.156.182] jgyllenhammar@junos.local(Cert- CD)[CD-Demo] - Login succeeded for jgyllenhammar@junos.local/Cert-CD from 195.67.156.182.

2009-02-10 11:32:13 - ive - [195.67.156.182] jgyllenhammar@junos.local(Cert- CD)[CD-Demo] - WebRequest ok: Host: adserver1.junos.local, Request: GET /exchange HTTP/1.1


TicketGranting Ticket is fetched, but no Kerberos S4U2Self Ticket is fetched for jgyllenhammar@JUNOS.LOCAL


That means the OWA server response is 401 and the user is prompted for Authentication as the defined fallback option.


Since the user now has supplied Credentials a Kerberos TGT is fetched for the actual user rather than the Constrained Delegation account (owacd)

2009-02-10 11:32:54 - ive - [195.67.156.182] jgyllenhammar@junos.local(Cert- CD)[CD-Demo] - WebRequest ok: Host: adserver1.junos.local, Request: GET /exchange HTTP/1.1


Kerberos Service Ticket is retrieved as access is allowed.

2009-02-10 11:32:54 - ive - [195.67.156.182] jgyllenhammar@junos.local(Cert- CD)[CD-Demo] - WebRequest completed, GET to http://adserver1.junos.local:80//exchange from 10.200.220.32 result=302 sent=22 received=161 in 1 seconds
2009-02-10 11:32:54 - ive - [195.67.156.182] jgyllehammar@junos.local(Cert- CD)[CD-Demo] - WebRequest ok : Host: adserver1.junos.local, Request: GET /exchange/ HTTP/1.1