Oracle Database Insider

All things database: the latest news, best practices, code examples, cloud, and more

Security

Make Someone Else do the Work - Managing Oracle Database 19c Users in Active Directory (part 1 - Kerberos)

October 1, 2020 | 6 minute read

Russ Lowenthal
Product Manager, Database Security

One of my least favorite database administration activities is managing users – creating users, changing passwords, granting roles – zero fun. Even further down on my scale of fun ways to spend an afternoon is the governance that goes with user management – which users have left the organization? Do they still need the privileges they have? It's important work, but it is a constant never-ending drumbeat of repetitive tasks that seems to always interrupt more interesting activities.

Fortunately, with a few built-in features of the database (in other words, not extra-cost options, not extra-cost software that we have to maintain) we can make someone else do that tedious work and free up our time for higher value tasks.

The solution? Microsoft Active Directory. I haven't worked with many organizations in the past 15 years that do NOT have Active Directory running. And fortunately, the Oracle Database plays well with Active Directory and has ever since Oracle 9i. Even better, starting with Oracle Database 18c and improving in Oracle Database 19c that integration with Active Directory has become easier to set up and requires even less maintenance.

To integrate with Active Directory we are going to use two database features – Kerberos authentication, and Centrally Managed Users *(note: Centrally Managed Users is an Enterprise Edition feature)*. When I started to write this I realized I was going WAY beyond a reasonable length for a blog, so I'm splitting this up across two different entries, one for Kerberos, the other for Centrally Managed Users. If you'd prefer to see the details on Kerberos in video form, please take a look at this YouTube video.

At the end of this post I've got a link to another video that goes into more depth on implementation and troubleshooting.

We will use Kerberos to authenticate database users. Each Active Directory domain controller is also a Kerberos Distribution Center. Kerberos can be used standalone to authenticate database users (in place of a password). A lot of times I'll advise my clients to just use Kerberos because that does the work of placing password management on the Active Directory team's plate, and also gets me out of having to worry about immediately deprovisioning a database user when someone leaves the company. Once the account is deleted from Active Directory, that account can no longer login to the database. We can clean it up when we get around to it. Even if we are going to configure Centrally Managed Users (CMU) I'll usually configure Kerberos first because once that's done, adding on CMU is just a few more minutes worth of work.

To setup Kerberos we will need to make changes in three places:

Database Server

Client Workstation

Active Directory

On the Database Server we configure a new network file called a krb5.conf. This file tells the Kerberos libraries where the Kerberos Distribution Center (from here on out I'm just going to call it the domain controller) is located, what port it is listening on, and which alias to send to that server. A typical krb5.conf file looks like this:

```
[libdefaults]
    default_realm = DBSECLABS.COM
    clockskew = 6000
    passwd_check_s_address = false
    noaddresses = true
    forwardable = yes
[realms]
    DBSECLABS.COM = {
        kdc = DBSECLABS.COM:88
    }
[domain_realm]
    DBSECLABS.COM = DBSECLABS.COM
    .DBSECLABS.COM = DBSECLABS.COM
    dbseclabs.com = DBSECLABS.COM
    .dbseclabs.com = DBSECLABS.COM
```

The file has three sections:

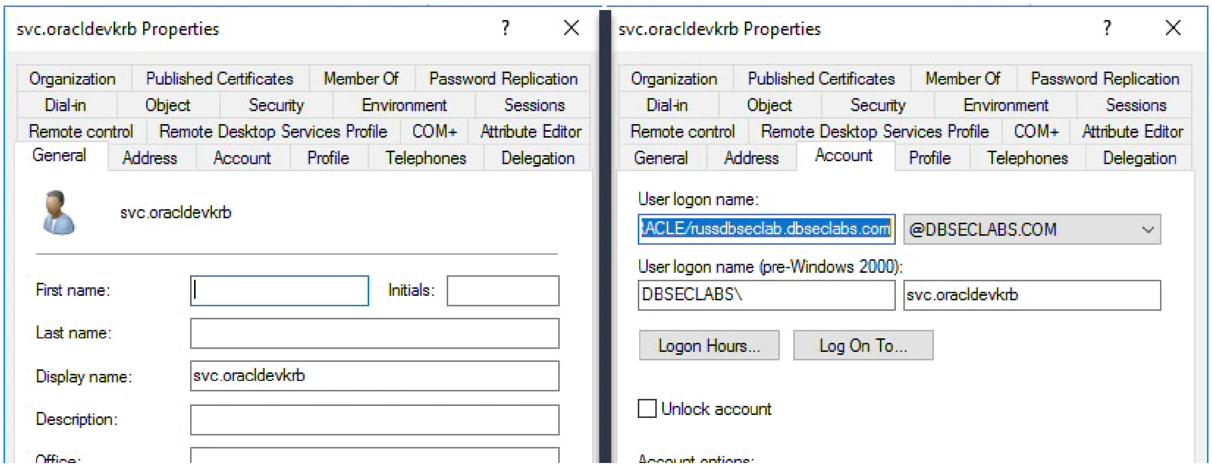
libdefaults – parameters that control how Kerberos is going to behave

realms – for each Kerberos domain, where are the domain controllers? What port do they listen for Kerberos requests on (almost always port 88) *pro tip: Notice I don't actually have a server name or IP address, instead I use the Active Directory domain name, and DNS sends me to whichever domain controller is available. This builds in fault tolerance and lets me survive the constant maintenance of domain controllers*

domain_realm – these are *aliases* for the domains listed in the realms section. The value on the left side of the equal sign is the alias, the value on the right is the entry in the realms section that this alias should direct requests to. Kerberos is case sensitive, so I list my aliases in both upper and lower case because I never can be sure what format a client will use.

The next change we need to make on the database server is to install a Kerberos keytab file. The AD administrator needs to:

1. Create a service account for our database server – this is just a regular Active Directory user account nothing special. Because it's a service account, I usually set "Password never expires" but follow your organizations standards


svc.oracldevkrb Properties

svc.oracldevkrb

First name:

Initials:

Last name:

Display name: svc.oracldevkrb

Description:

Office:

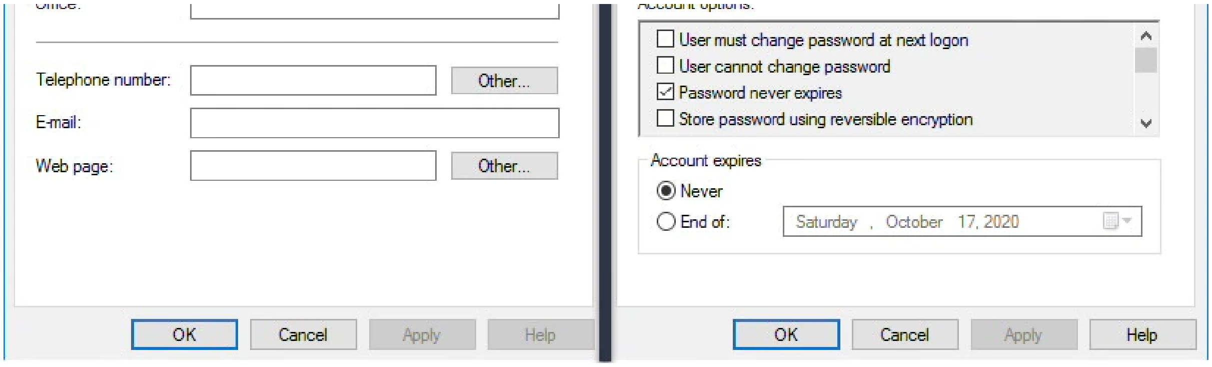
svc.oracldevkrb Properties

User logon name: ACLE/russdbseclab.dbseclabs.com @DBSECLABS.COM

User logon name (pre-Windows 2000): DBSECLABS\ svc.oracldevkrb

□ Unlock account

Account options:



Telephone number:

E-mail:

Web page:

Account options:

□ User must change password at next logon
□ User cannot change password
☑ Password never expires
□ Store password using reversible encryption

Account expires

● Never
○ End of: Saturday , October 17, 2020

2. Have your Active Directory administrator create a keytab for you. The command they will use to do this will look like this:

```
ktpass -princ ORACLE/<DATABASE_SERVER_HOST_NAME>.
<DATABASE_SERVER_HOST_DOMAIN>@<ACTIVE DIRECTORY DEFAULT DOMAIN> -pass
<ACTIVE DIRECTORY USERS PASSWORD> -mapuser <ACTIVE_DIRECTORY_USER_NAME> -
crypto ALL -ptype KRB5_NT_PRINCIPAL -out database.keytab
```

Copy the keytab to our database server. I usually put the keytab in $ORACLE_HOME/network/admin directory. If I'm using the new read-only $ORACLE_HOME feature I put it in $ORACLE_BASE_HOME/network/admin instead.

The last file we work with is our sqlnet.ora file. In this file we will add seven new parameters:

```
#Kerberos Parameters
SQLNET.AUTHENTICATION_SERVICES=(beq,kerberos5pre,kerberos5)
SQLNET.FALLBACK_AUTHENTICATION=TRUE
SQLNET.AUTHENTICATION_KERBEROS5_SERVICE=ORACLE
SQLNET.KERBEROS5_CONF=/oracle/19/dbhome_1/network/admin/krb5.conf
SQLNET.KERBEROS5_CLOCKSKEW=6000
SQLNET.KERBEROS5_CONF_MIT=TRUE
#Following parameter is server-side only
SQLNET.KERBEROS5_KEYTAB=/oracle/19/dbhome_1/network/admin/database.keytab
```

These parameters are described in the *Database Net Services Reference* guide so I'll save space by not defining them here. One of the entries, sqlnet.kerberos5_conf, points to the krb5.conf file we discussed earlier. Another, sqlnet.kerberos5_keytab, points to the keytab file generated above.

Copy the krb5.conf file to the client workstation and update the client's sqlnet.ora with the relevant parameters:

```
#Kerberos Parameters
SQLNET.AUTHENTICATION_SERVICES=(kerberos5)
SQLNET.FALLBACK_AUTHENTICATION=TRUE
SQLNET.AUTHENTICATION_KERBEROS5_SERVICE=ORACLE
SQLNET.KERBEROS5_CONF=/oracle/19/dbhome_1/network/admin/krb5.conf
SQLNET.KERBEROS5_CLOCKSKEW=6000
SQLNET.KERBEROS5_CONF_MIT=TRUE
```

That’s it – now we are ready to begin using Kerberos. First, we create an externally authenticated database user (identified by the Kerberos principle name) in the database and grant that user the CREATE SESSION privilege.

```
SQL>  create user RUSS identified externally as
'rlowenth@DBSECLABS.COM';
User created.

SQL> grant create session to RUSS;
Grant succeeded.
```

Now we get a Kerberos ticket from Active Directory and use it to login to the database. The "rlowenth" you see in the Kerberos principal name above, and the okinit command below, is my Active Directory username.

```
[oracle@russ-test ~]$ okinit rlowenth
Kerberos Utilities for Linux: Version 19.0.0.0.0 - Production on 18-
SEP-2020 22:41:01
Copyright (c) 1996, 2019 Oracle.  All rights reserved.
Configuration file :
/opt/oracle/product/19c/dbhome_1/network/admin/krb5.conf.
Password for rlowenth@DBSECLABS.COM:

[oracle@russ-test ~]$ oklist
Kerberos Utilities for Linux: Version 19.0.0.0.0 - Production on 18-
SEP-2020 22:41:07
Copyright (c) 1996, 2019 Oracle.  All rights reserved.
Configuration file :
/opt/oracle/product/19c/dbhome_1/network/admin/krb5.conf.
Ticket cache: FILE:/tmp/krb5cc_1001
Default principal: rlowenth@DBSECLABS.COM

Valid starting     Expires            Service principal
09/18/20 22:41:05  09/19/20 08:41:05
krbtgt/DBSECLABS.COM@DBSECLABS.COM
        renew until 09/19/20 22:41:01

[oracle@russ-test ~]$ sqlplus /@pdb1

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Sep 18 22:41:20 2020
Version 19.8.0.0.0
Copyright (c) 1982, 2020, Oracle.  All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.8.0.0.0
SQL> select sys_context('userenv','authentication_method') from dual;

SYS CONTEXT('USERENV','AUTHENTICATION METHOD')
```

```
--------------------------------------------------------------------
KERBEROS
```

As you can see, we can now login to the database using Kerberos. If our client is on a Windows desktop we could just use the in-memory ticket created when we logged in and skip the okinit step. For more information, this YouTube video

from the monthly Database Security community calls talks more more about Kerberos implementation and troubleshooting – if you are not already subscribed to the monthly Database Security community calls you might want to do that now.

In my next post, I'll take this integration with Active Directory a step farther, and enable Centrally Managed Users

Russ Lowenthal
Product Manager, Database Security

Russ Lowenthal is the Senior Director of Product Management for Database Security, focused on database encryption, access control, audit, and monitoring.
Russ is based in Orlando, Florida, USA and has been with Oracle for over twenty years. Leveraging

Show more

Previous Post

Next Post

Resources for

- About
- Careers
- Developers
- Investors
- Partners
- Startups

Why Oracle

- Analyst Reports
- Best CRM
- Cloud Economics
- Corporate Responsibility
- Diversity and Inclusion
- Security Practices

Learn

- What is Customer Service?
- What is ERP?
- What is Marketing Automation?
- What is Procurement?
- What is Talent Management?
- What is VM?

What's New

- Try Oracle Cloud Free Tier
- Oracle Sustainability
- Oracle COVID-19 Response
- Oracle and SailGP
- Oracle and Premier League
- Oracle and Red Bull Racing Honda

Contact Us

- US Sales 1.800.633.0738
- How can we help?
- Subscribe to Oracle Content
- Try Oracle Cloud Free Tier
- Events
- News

Site Map   Privacy / Do Not Sell My Info   Настройки cookie   Ad Choices   Careers