

Troubleshooting and Configuring Kerberos V5 Client Support

Document ID: 13805

Introduction

Prerequisites

- Requirements
- Components Used
- Conventions

Introduction to Kerberos

- Definitions

Gotcha

Cisco IOS Router Configuration

Kerberos KDC Configuration

- Set Up Ports for inetd
- Set Up Kerberos Configuration Files
- Set Up the Database for the KDC Server

Sample Debug Output

Troubleshoot

- Wrong Realm Name
- DNS Does Not Work
- Router Clock Not Correct
- Client Not In Kerberos Database
- Client Is In Database but uses Wrong Password
- SRVTAB Entry Not Correct on Router

References

Related Information

Introduction

This document provides an example configuration, as well as some solutions to common problems. Techniques that help you to troubleshoot any issues are also provided in this document. This document does not address kerberized Telnet support.

Most of this material in this article came from the freely available documentation that comes with Kerberos and from various available frequently asked questions (FAQs) on the package. The configurations came from a functional router and Kerberos KDC server.

This document assumes that you have correctly compiled and installed a current release of Version 5 of the Kerberos package from MIT. Refer to the references at the end of this article for information on how to obtain, compile, and install Kerberos V5.

Also note that Cisco IOS[®] Software Release 11.2 or later is required for Kerberos V5 support. This provides full support of Kerberos V client authentication, which includes credential forwarding. Systems that have Kerberos V infrastructures can use their Key Distribution Centers (KDCs) in order to authenticate end-users for network or router access. This is a client implementation and not a Kerberos KDC implementation.

Kerberos is considered a legacy security service and is most beneficial in networks that already use Kerberos.

Refer to the Cisco IOS Software Release 11.2 release notes for more detailed information of which versions

include this support.

For Kerberos support in subsequent Cisco IOS Software releases, refer to the Software Advisor (registered customers only) .

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is based on these software and hardware versions:

- Cisco IOS Software Release 11.2 and later

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

For more information on document conventions, refer to the Cisco Technical Tips Conventions.

Introduction to Kerberos

Kerberos is a network authentication protocol for use on physically insecure networks. Kerberos is based on the key distribution model presented by Needham and Schroeder. (See Number 9 in the References section of this document. It is designed to provide strong authentication for client/server applications by the use of secret-key cryptography. It allows entities that communicate over networks to prove their identity to each other while it prevents eavesdropping or replay attacks. It also provides for data stream integrity (such as detection of modification) and secrecy (such as prevention of unauthorized reading) with the help of cryptography systems such as DES.

Many of the protocols used in the Internet do not provide any security. Tools used to "sniff" passwords off of the network are in common use by systems crackers. Thus, applications which send a password over the network unencrypted are vulnerable. Also, other client/server applications rely on the client program to be "honest" about the identity of the user who uses it. Other applications rely on the client to restrict its activities to those which it is allowed to do, with no other enforcement by the server.

Some sites attempt to use firewalls in order to solve their network security problems. Firewalls assume that "the bad guys" are on the outside, which is often an invalid assumption. However, the majority of the computer crime incidents that cause more damage have been carried out by insiders. Firewalls also have a significant disadvantage in that they restrict how your users are able to use the Internet.

Kerberos was created by MIT as a solution to these network security problems. The Kerberos protocol uses strong cryptography, so that a client can prove its identity to a server (and vice versa) across an insecure network connection. After a client and server has used Kerberos in order to prove their identity, they can also encrypt all of their communications in order to assure privacy and data integrity as they go about their business.

Kerberos is freely available from MIT, under a copyright permission notice that is similar to the one used for the BSD operating and X11 Windowing system. MIT provides Kerberos in source form. This is done so that anyone who wishes to use it can look over the code for themselves and assure themselves that the code is trustworthy. In addition, for those who prefer to rely on a professionally supported product, Kerberos is available as a product from many different vendors.

Kerberos V5 Client Support is based on the Kerberos authentication system developed at MIT. Under Kerberos, a client (generally either a user or a service) sends a request for a ticket to the Key Distribution Center (KDC). The KDC creates a ticket-granting ticket (TGT) for the client, encrypts it with the help of the password of the client as the key, and sends the encrypted TGT back to the client. The client then attempts to decrypt the TGT, with the help of its password. If the client successfully decrypts the TGT for example, if the client gives the correct password), it keeps the decrypted TGT. This indicates proof of the identity of the client.

The TGT, which expires at a specified time, permits the client to obtain additional tickets, which give permission for specific services. The requests and grants of these additional tickets is user-transparent.

Since Kerberos negotiates authenticated, is optionally encrypted, and communicates between any two points on the Internet, it provides a layer of security that is not dependent upon which side of a firewall either client is located. Kerberos is primarily used in application-level protocols (ISO model Level 7), such as Telnet or FTP, in order to provide user to host security. It is also used, though less frequently, as the implicit authentication system of data stream (such as **SOCK_STREAM**) or RPC mechanisms (ISO model Level 6). It can also be used at a lower level for host-to-host security, in protocols such as IP, UDP, or TCP (ISO model Levels 3 and 4). Although, such implementations are rare, if they exist at all.

It provides for mutual authentication and secure communication between principals on an open network by the manufacture of secret keys for any requester. A mechanism for these secret keys to be safely propagated through the network is also provided. Kerberos does not provide for authorization or accounting. However, applications that wish to can use their secret keys in order to perform those functions securely.

Definitions

- **Authentication** Ensure that you are who you say you are, and that we know who you are.
- **Client** An entity that can obtain a ticket. This entity is usually either a user or a host.
- **Credentials** The same as tickets.
- **Daemon** A program, usually one that runs on a UNIX host, that services network requests for authentication.
- **Host** A computer that can be accessed over a network.
- **Instance** The second part of a Kerberos principal. It gives information that qualifies the primary. The instance can be null. In the case of a user, the instance is often used in order to describe the intended use of the corresponding credentials. In the case of a host, the instance is the fully qualified hostname.
- **Kerberos** In Greek mythology, the three-headed dog that guards the entrance to the underworld. In the world of computers, Kerberos is a network security package that was developed at MIT.
- **KDC** Key Distribution Center. A machine that issues Kerberos tickets.
- **Keytab** A key table file that contains one or more keys. A host or service uses a keytab file in much the same way as a user uses their password.
- **NAS** A Network Access Server (a Cisco box) or anything else which makes TACACS+ authentication and authorization requests, or sends accounting packets.
- **Principal** A string that names a specific entity to which a set of credentials can be assigned. It generally has three parts named Primary, Instance, and REALM.

The typical format of a typical Kerberos principal is **primary/instanceREALM**.

- **Primary** The first part of a Kerberos principal. In the case of a user, it is the username. In the case of a service, it is the name of the service.

- **REALM** The logical network served by a single Kerberos database and a set of Key Distribution Centers. By convention, realm names are generally all uppercase letters, to differentiate the realm from the Internet domain.
- **Service** Any program or computer you access over a network. Examples of services include:
 - ◆ "host" a host, (for example, when you use Telnet and rsh)
 - ◆ "ftp" FTP
 - ◆ "krbtgt" authentication; such as ticket-granting ticket
 - ◆ "pop" E-mail
- **Ticket** A temporary set of electronic credentials that verify the identity of a client for a particular service.
- **TGT** Ticket-Granting Ticket. A special Kerberos ticket that permits the client to obtain additional Kerberos tickets within the same Kerberos realm.

A good analogy for the ticket-granting ticket is a three-day ski pass that is good at four different resorts. You show the pass at whichever resort you decide to go to (until it expires), and you receive a lift ticket for that resort. Once you have the lift ticket, you can ski all you want at that resort. If you go to another resort the next day, you once again show your pass, and you get an additional lift ticket for the new resort. The difference is that the Kerberos V5 programs notice that you have the weekend ski pass, and get the lift ticket for you, so you do not have to perform the transactions yourself.

Gotcha

This section lists several items that you need to be aware of:

- Make sure you remove all trailing spaces in the configuration files. Trailing spaces can cause problems with the krb5kdc server. Otherwise, you can get a message that says, "krb5kdc cannot start the database for the realm."
- Make sure the clock on the router is set to the same time as the UNIX host that runs the KDC server. In order to prevent intruders from resetting their system clocks in order to continue to use expired tickets, Kerberos V5 is set up to reject ticket requests from any host whose clock is not within the specified maximum clock skew of the KDC (as specified in the kdc.conf file). Similarly, hosts are configured to reject responses from any KDC whose clock is not within the specified maximum clock skew of the host (as specified in the krb5.conf file). The default value for maximum clock skew is 300 seconds (five minutes).
- Make sure DNS works properly. Several aspects of Kerberos rely on name service. In order for Kerberos to provide its high level of security, it is more sensitive to name service problems than some other parts of your network. It is important that your Domain Name System (DNS) entries and your hosts have the correct information. Each canonical of the host name must be the fully-qualified host name (that includes the domain), and each IP address of the host must reverse-resolve to the canonical name.
- Cisco IOS Kerberos V5 support does not allow the use of lowercase realm names and the Kerberos code in the Cisco IOS does not authenticate users if the realm is in lowercase. This was fixed in Cisco IOS Software Release 11.2(7).

Refer to Cisco bug ID CSCdj10598 (registered customers only) .

The only workaround is to use uppercase REALM names (which is conventional).

The lowercase realms work in order to retrieve a TGT, but not a service credential. Since Cisco uses their new TGT in order to retrieve a service credential (used to prevent the KDC spoofing attack) during logging authentication, Kerberos authentication that uses lowercase realms always fails.

- Kerberos V5 for PPP PAP and CHAP can crash the router. This was fixed in Cisco IOS Software

Release 11.2(6).

Refer to Cisco bug ID CSCdj08828 (registered customers only) .

The workaround for this is to force exec login into the router via **async mode interactive** without **autoselect during-login** and then have the user start PPP manually:

```
aaa authentication ppp default if-needed krb5 local
```

- Kerberos V5 does not do authorization or accounting. You need some other code in order to do this.

Cisco IOS Router Configuration

The configuration in this section depicts a fully configured AS5200 router that does Kerberos V5. The router in this configuration uses the Kerberos server in order to authenticate both VTY sessions and users that dial in to do PPP with PAP authentication.

AS5200 Config with Kerberos V5

```
version 11.2
service timestamps debug datetime msec
!
hostname cisco5200
!
aaa new-model
aaa authentication login cisco2 krb5 local
aaa authentication ppp cisco krb5 local
enable secret
enable password
!
username cisco password cisco
ip host-routing
ip domain-name cisco.edu
ip name-server 10.10.1.25
ip name-server 10.10.20.3
kerberos local-realm CISCO.EDU
kerberos srvtab entry host/cisco5200.cisco.edu@CISCO.EDU 0 861289666 2
1 80:>:11338>531159=
!
!--- You do not actually enter the previous line.
!--- Enter "kerberos srvtab remote 10.10.1.8 /ts/krb5.keytab" and the
!--- the router TFTP's the key entry on its own.

kerberos server CISCO.EDU 10.10.1.8
kerberos credentials forward
isdn switch-type primary-5ess
clock timezone GMT -6
clock summer-time CDT recurring
!
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 pri-group timeslots 1-24
!
controller T1 1
 framing esf
 clock source line secondary
 linecode b8zs
 pri-group timeslots 1-24
!
interface Ethernet0
```

```
ip address 10.10.110.245 255.255.255.0
no ip mroute-cache
!
interface Serial0
no ip address
no ip mroute-cache
shutdown
!
interface Serial1
no ip address
no ip mroute-cache
shutdown
!
interface Serial0:23
ip unnumbered Ethernet0
no ip mroute-cache
encapsulation ppp
isdn incoming-voice modem
no cdp enable
!
interface Serial1:23
ip unnumbered Ethernet0
no ip mroute-cache
encapsulation ppp
isdn incoming-voice modem
no cdp enable
!
interface Group-Async1
ip unnumbered Ethernet0
no ip mroute-cache
encapsulation ppp
async mode interactive
peer default ip address pool mypool
dialer in-band
dialer idle-timeout 9999
dialer-group 1
no cdp enable
ppp authentication pap cisco
group-range 1 48
!
ip local pool mypool 10.10.110.97 10.10.110.144
no ip classless
ip route 0.0.0.0 0.0.0.0 10.10.110.254
!
dialer-list 1 protocol ip permit
!
line con 0
login authentication test
line 1 48
autoselect ppp
login authentication cisco2
modem InOut
transport input all
line aux 0
modem InOut
transport input all
flowcontrol hardware
line vty 0 10
exec-timeout 0 0
login authentication cisco2
!
end
```

Kerberos KDC Configuration

Make sure you have the proper ports set up for **inetd**.

Note: This example uses wrappers. If you want encrypted Telnet, you need to replace the normal Telnet with the kerberized Telnet, so these files have a different appearance.

Set Up Ports for inetd

```
# cat /etc/services
-----
#
# Syntax:  ServiceName PortNumber/ProtocolName [alias\_1,...,alias\_n] [#comments]
#
# ServiceName          official Internet service name
# PortNumber           the socket port number used for the service
# ProtocolName         the transport protocol used for the service
# alias                unofficial service names
# #comments            text following the comment character (#) is ignored
#
tftp                  69/udp

kerberos              88/udp          kdc
kerberos              88/tcp          kdc

kxct                  549/tcp

klogin                543/tcp          # Kerberos authenticated rlogin
kshell                544/tcp          cmd      # and remote shell
kerberos-admin       749/tcp          # Kerberos 5 admin/changepw
kerberos-admin       749/udp          # Kerberos 5 admin/changepw
kerberos-sec         750/udp          kdc      # Kerberos authentication--udp
kerberos-sec         750/tcp          kdc      # Kerberos authentication--tcp
krb5\_prop            754/tcp          # Kerberos slave propagation
eklogin               2105/tcp         # Kerberos auth. & encrypted rlogin
krb524                4444/tcp         # Kerberos 5 to 4 ticket translator
-----

#cat /etc/inetd.conf

ident  stream  tcp    nowait  root    /usr/local/etc/in.identd in.identd
ftp    stream  tcp    nowait  root    /usr/sbin/tcpd          ftpd
telnet stream  tcp    nowait  root    /usr/sbin/tcpd          telnetd
#shell stream  tcp    nowait  root    /usr/sbin/tcpd          rshd
shell  stream  tcp    nowait  root    /usr/sbin/rshd          rshd
#login stream  tcp    nowait  root    /usr/sbin/tcpd          rlogind
login  stream  tcp    nowait  root    /usr/sbin/rlogind       rlogind
exec   stream  tcp    nowait  root    /usr/sbin/rexecd        rexecd
# Run as user "uucp" if you don't want uucpd's wtmp entries.
#uucp  stream  tcp    nowait  root    /usr/sbin/uucpd         uucpd
#finger stream  tcp    nowait  root    /usr/sbin/tcpd          fingerd
# tftp was /tmp and is now /ts for terminal server macros
tftp   dgram   udp    wait    nobody  /usr/sbin/tcpd          tftpd /ts
comsat dgram   udp    wait    root    /usr/sbin/comsat        comsat
-----
```

Set Up Kerberos Configuration Files

Next, you need to set up a few Kerberos configuration files that the KDC server reads. For more information on what these parameters mean, refer to the Kerberos Install Guide or the System Admin Guide .

```
# cat /etc/krb5.conf
```

```

[libdefaults]
    default_realm = CISCO.EDU
    ticket_lifetime = 600
    default_tgs_enctypes = des-cbc-crc
    default_tkt_enctypes = des-cbc-crc

[realms]
    CISCO.EDU = {
        kdc = ciscoaxa.cisco.edu:88
        admin_server = ciscoaxa.cisco.edu
        default_domain = CISCO.EDU
    }

[domain_realm]
    .cisco.edu = CISCO.EDU
    cisco.edu = CISCO.EDU

[logging]
    kdc = FILE:/var/log/krb5kdc.log
    admin_server = FILE:/var/log/kadmin.log
    default = FILE:/var/log/krb5lib.log

# cat /usr/local/var/krb5kdc/kdc.conf

[kdcdefaults]
    kdc_ports = 88,750

[realms]
    CISCO.EDU = {
        database_name = /usr/local/var/krb5kdc/principal
        admin_keytab = FILE:/usr/local/var/krb5kdc/kadm5.keytab
        acl_file = /usr/local/var/krb5kdc/kadm5.acl
        acl_file = /usr/local/var/krb5kdc/kadm5.dict
        key_stash_file = /usr/local/var/krb5kdc/.k5.CISCO.EDU
        kadmind_port = 749
        max_life = 10h 0m 0s
        max_renewable_life = 7d 0h 0m 0s
        master_key_type = des-cbc-crc
        supported_enctypes = des-cbc-crc:normal des:normal des:v4
        des:norealm des:onlyrealm des:afs3
    }

```

Set Up the Database for the KDC Server

Next, you need to create the database that the KDC server uses.

1. Enter the command **kdb5_util**:

```

# kadmin/dbutil/kdb5_util
Usage: kdb5_util cmd [-r realm] [-d dbname] [-k mkeytype] [-M mkeyname]
        [-m] [cmd options]
        create [-s]
        destroy [-f]
        stash [-f keyfile]
        dump [-old] [-ov] [-b6] [-verbose] [filename [princs...]]
        load [-old] [-ov] [-b6] [-verbose] [-update] filename
        dump_v4 [filename]
        load_v4 [-t] [-n] [-v] [-K] [-s stashfile] inputfile
-----

# kadmin/dbutil/kdb5_util destroy -r cisco.edu
kdb5_util: No such file or directory while setting active database to

```

```
"/usr/local/var/krb5kdc/principal"
```

```
# kadmin/dbutil/kdb5_util create -r CISCO.EDU -s
Initializing database '/usr/local/var/krb5kdc/principal'
for realm 'CISCO.EDU',
master key name 'K/M@CISCO.EDU'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key:
Re-enter KDC database master key to verify:
```

This is needed in order to retrieve the **srvtab** password from the router via TFTP with the **kerberos srvtab remote** command.

```
# kadmin/dbutil/kdb5_util stash -r CISCO.EDU
Enter KDC database master key:
```

2. In order to add principals and users to the database, use the **kadmin.local** command:

```
# kadmin/cli/kadmin.local

kadmin.local: listprincs
kadmin/admin@CISCO.EDU
kadmin/changepw@CISCO.EDU
K/M@CISCO.EDU
krbtgt/CISCO.EDU@CISCO.EDU
kadmin/history@CISCO.EDU
kadmin.local:
kadmin.local: ?
Available kadmin.local requests:

add_principal, addprinc, ank
                                Add principal
delete_principal, delprinc
                                Delete principal
modify_principal, modprinc
                                Modify principal
change_password, cpw           Change password
get_principal, getprinc        Get principal
list_principals, listprincs, get_principals, getprincs
                                List principals
add_policy, addpol            Add policy
modify_policy, modpol         Modify policy
delete_policy, delpol         Delete policy
get_policy, getpol            Get policy
list_policies, listpols, get_policies, getpols
                                List policies
get_privs, getprivs           Get privileges
ktadd, xst                    Add entry(s) to a keytab
ktremove, ktrem               Remove entry(s) from a keytab
list_requests, lr, ?          List available requests.
quit, exit, q                 Exit program.
-----
```

3. Add a user:

```
kadmin.local: ank ciscol@CISCO.EDU
Enter password for principal "ciscol@CISCO.EDU":
Re-enter password for principal "ciscol@CISCO.EDU":
Principal "ciscol@CISCO.EDU" created.
```

4. Get a list of the current database:

```
kadmin.local: listprincs
kadmin/admin@CISCO.EDU
kadmin/changepw@CISCO.EDU
```

```
cisc01@CISCO.EDU
K/M@CISCO.EDU
krbtgt/CISCO.EDU@CISCO.EDU
kadmin/history@CISCO.EDU
```

5. Add the entry for the Cisco router:

```
kadmin.local: ank host/cisco5200.cisco.edu@CISCO.EDU
Enter password for principal "host/cisco5200.cisco.edu@CISCO.EDU":
```

```
Re-enter password for principal "host/cisco5200.cisco.edu@CISCO.EDU":
Principal "host/cisco5200.cisco.edu@CISCO.EDU" created.
```

6. Extract a key to the table for the Cisco router:

```
kadmin.local: ktadd host/cisco5200.cisco.edu@CISCO.EDU
Entry for principal host/cisco5200.cisco.edu@CISCO.EDU with kvno 2,
encryption type DES-CBC-CRC added to keytab WRFILE:/etc/krb5.keytab.
```

7. Take another look at the database:

```
kadmin.local: listprincs
kadmin/admin@CISCO.EDU
kadmin/changepw@CISCO.EDU
cisc01@CISCO.EDU
K/M@CISCO.EDU
krbtgt/CISCO.EDU@CISCO.EDU
kadmin/history@CISCO.EDU
host/cisco5200.cisco.edu@CISCO.EDU
```

```
kadmin.local: quit
```

8. Move the keytab file to a place where the router is able to get to it:

```
# cp /etc/krb5.keytab /ts/
# chmod 777 /ts/krb5.keytab
```

9. Start the KDC server:

```
# kdc/krb5kdc
#
```

10. Check to make sure it actually runs:

```
# ps -A | grep 'krb5'
6043 ?? I 0:00.01 kdc/krb5kdc
23427 ttypf S + 0:00.05 grep krb5
```

11. Force the router to read its key table entry:

```
cisco5200(config)#kerberos srvtab remote 10.10.1.8 /ts/krb5.keytab
Loading /ts/krb5.keytab from 10.10.1.8 (via Ethernet0): !
[OK - 229/1000 bytes]
```

12. Check the router to make sure everything is ready:

```
cisco5200#write terminal

aaa new-model
aaa authentication login cisco2 krb5 local
aaa authentication ppp cisco krb5 local
kerberos local-realm CISCO.EDU
kerberos srvtab entry host/cisco5200.cisco.edu@CISCO.EDU 0 861289666
2 1 8 0:>:11338>531159=
kerberos server CISCO.EDU 10.10.1.8
kerberos credentials forward
```

13. Turn on debugging and try to log into the router:

```
cisco5200#terminal monitor
cisco5200#debug kerberos
```

```

Kerberos debugging is on
cisco5200#debug aaa authen
AAA Authentication debugging is on
cisco5200#show clock
10:16:41.797 CDT Thu Apr 17 1997
cisco5200#
Apr 17 15:16:58.965: AAA/AUTHEN: create_user user='' ruser='' port='tty51'
rem_addr='12.12.109.64'
          authen_TYPE=ASCII service=LOGIN priv=1
Apr 17 15:16:58.969: AAA/AUTHEN/START (0): port='tty51' list='cisco2'
ACTION=LOGIN service=LOGIN
Apr 17 15:16:58.969: AAA/AUTHEN/START (1957396): found list
Apr 17 15:16:58.973: AAA/AUTHEN/START (1667706374): METHOD=KRB5
Apr 17 15:16:58.973: AAA/AUTHEN (1667706374): status = GETUSER
Apr 17 15:17:02.493: AAA/AUTHEN/CONT (1667706374): continue_login
Apr 17 15:17:02.493: AAA/AUTHEN (1667706374): status = GETUSER
Apr 17 15:17:02.497: AAA/AUTHEN (1667706374): METHOD=KRB5
Apr 17 15:17:02.497: AAA/AUTHEN (1667706374): status = GETPASS
Apr 17 15:17:05.401: AAA/AUTHEN/CONT (1667706374): continue_login
Apr 17 15:17:05.405: AAA/AUTHEN (1667706374): status = GETPASS
Apr 17 15:17:05.405: AAA/AUTHEN (1667706374): METHOD=KRB5
Apr 17 15:17:05.413: Kerberos: Requesting TGT with expiration
date of 861319025
Apr 17 15:17:05.417: Kerberos: Sending TGT request with no
pre-authorization data.
Apr 17 15:17:05.441: Kerberos: Sent TGT request to KDC
Apr 17 15:17:06.405: Kerberos: Received TGT reply from KDC
Apr 17 15:17:06.465: Domain: query for 245.110.10.10.in-addr.arpa
to 10.10.1.25 Reply received ok
Apr 17 15:17:06.569: Kerberos: Sent TGT request to KDC
Apr 17 15:17:06.769: Kerberos: Received TGT reply from KDC
Apr 17 15:17:06.881: Kerberos: Received valid credential with
endtime of 861232625
Apr 17 15:17:06.897: AAA/AUTHEN (1667706374): status = PASS

```

Sample Debug Output

Here is a PPP user that successfully authenticates.

```

cisco5200#debug ppp auth
Apr 17 15:47:15.285: Async6: Dialer received incoming call from <unknown>
%LINK-3-UPDOWN: Interface Async6, changed state to up
Apr 17 15:47:17.293: Async6: Dialer received incoming call from <unknown>
Apr 17 15:47:17.909: PPP Async6: PAP receive authenticate request cisco1
Apr 17 15:47:17.913: PPP Async6: PAP authenticating peer cisco1
Apr 17 15:47:17.917: AAA/AUTHEN: create_user user='cisco1' ruser='' port='Async6'
rem_addr='async/6151010'
          authen_TYPE=PAP service=PPP priv=1
Apr 17 15:47:17.917: AAA/AUTHEN/START (0): port='Async6' list='cisco'
ACTION=LOGIN service=PPP
Apr 17 15:47:17.921: AAA/AUTHEN/START (4706358): found list
Apr 17 15:47:17.921: AAA/AUTHEN/START (712179591): METHOD=KRB5
Apr 17 15:47:17.929: Kerberos: Requesting TGT with expiration date of 861320837
Apr 17 15:47:17.933: Kerberos: Sending TGT request with no pre-authorization data.
Apr 17 15:47:17.957: Kerberos: Sent TGT request to KDC
Apr 17 15:47:18.765: Kerberos: Received TGT reply from KDC
Apr 17 15:47:18.893: Kerberos: Sent TGT request to KDC
Apr 17 15:47:19.097: Kerberos: Received TGT reply from KDC
Apr 17 15:47:19.205: Kerberos: Received valid credential with endtime of 861234437
Apr 17 15:47:19.221: AAA/AUTHEN (712179591): status = PASS
Apr 17 15:47:19.225: PPP Async6: Remote passed PAP authentication sending Auth-Ack.
Apr 17 15:47:19.225: Async6: authenticated host cisco1 with no matching dialer map
%LINEPROTO-5-UPDOWN: Line protocol on Interface Async6, changed state to up

```

Troubleshoot

This section contains various scenarios for potential problems. These debugs help you to quickly see a problem.

Wrong Realm Name

```
cisco5200#
cisco5200#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
cisco5200(config)#kerberos local-realm junk.COM
cisco5200#
Apr 17 15:19:16.089: AAA/AUTHEN: create_user user='' ruser=''
      port='tty51' rem_addr='12.12.109.64' authen_TYPE=ASCII
      service=LOGIN priv=1
Apr 17 15:19:16.093: AAA/AUTHEN/START (0): port='tty51' list='cisco2'
      ACTION=LOGIN service=LOGIN
Apr 17 15:19:16.097: AAA/AUTHEN/START (1957396): found list
Apr 17 15:19:16.129: AAA/AUTHEN/START (56280416): METHOD=KRB5
Apr 17 15:19:16.129: AAA/AUTHEN (56280416): status = GETUSER
Apr 17 15:19:21.721: AAA/AUTHEN/CONT (56280416): continue_login
Apr 17 15:19:21.721: AAA/AUTHEN (56280416): status = GETUSER
Apr 17 15:19:21.725: AAA/AUTHEN (56280416): METHOD=KRB5
Apr 17 15:19:21.725: AAA/AUTHEN (56280416): status = GETPASS
Apr 17 15:19:26.057: AAA/AUTHEN/CONT (56280416): continue_login
Apr 17 15:19:26.057: AAA/AUTHEN (56280416): status = GETPASS
Apr 17 15:19:26.061: AAA/AUTHEN (56280416): METHOD=KRB5
Apr 17 15:19:26.065: Kerberos: Requesting TGT with expiration date
      of 861319166
Apr 17 15:19:26.069: Kerberos: Sending TGT request with no
      pre-authorization data.
Apr 17 15:19:26.089: Kerberos: Received invalid credential.
      ~~~~~
Apr 17 15:19:26.093: AAA/AUTHEN (56280416): password incorrect
Apr 17 15:19:26.097: AAA/AUTHEN (56280416): status = FAIL
Apr 17 15:19:28.169: AAA/AUTHEN: free user cisco1 tty51 12.12.109.64
      authen_TYPE=ASCII service=LOGIN priv=1
Apr 17 15:19:28.173: AAA/AUTHEN: create_user user='' ruser=''
      port='tty51' rem_addr='12.12.109.64' authen_TYPE=ASCII
      service=LOGIN priv=1
Apr 17 15:19:28.177: AAA/AUTHEN/START (0): port='tty51' list='cisco2'
      ACTION=LOGIN service=LOGIN
Apr 17 15:19:28.177: AAA/AUTHEN/START (1957396): found list
Apr 17 15:19:28.181: AAA/AUTHEN/START (126312328): METHOD=KRB5
Apr 17 15:19:28.181: AAA/AUTHEN (126312328): status = GETUSER
```

DNS Does Not Work

```
Apr 10 17:22:15.370: Kerberos: Requesting TGT with expiration date
      of 860721735
Apr 10 17:22:15.374: Kerberos: Sending TGT request with no
      pre-authorization data.
Apr 10 17:22:15.398: Kerberos: Sent TGT request to KDC
Apr 10 17:22:16.034: Kerberos: Received TGT reply from KDC
Apr 10 17:22:16.090: Domain: query for 245.110.10.10.in-addr.arpa
      to 255.255.255.255 Reply received empty
      ~~~~~
```

Router Clock Not Correct

```
pppcisco1#
Apr 18 20:41:41.011: AAA/AUTHEN: create_user user='' ruser=''
```

```

    port='tty51' rem_addr='171.68.109.64' authen_TYPE=ASCII
    service=LOGIN priv=1
Apr 18 20:41:41.011: AAA/AUTHEN/START (0): port='tty51' list='cisco2'
    ACTION=LOGIN service=LOGIN
Apr 18 20:41:41.015: AAA/AUTHEN/START (1957396): found list
Apr 18 20:41:41.015: AAA/AUTHEN/START (4036314657): METHOD=KRB5
Apr 18 20:41:41.019: AAA/AUTHEN (4036314657): status = GETUSER
Apr 18 20:41:43.835: AAA/AUTHEN/CONT (4036314657): continue_login
Apr 18 20:41:43.839: AAA/AUTHEN (4036314657): status = GETUSER
Apr 18 20:41:43.839: AAA/AUTHEN (4036314657): METHOD=KRB5
Apr 18 20:41:43.843: AAA/AUTHEN (4036314657): status = GETPASS
Apr 18 20:41:48.835: AAA/AUTHEN/CONT (4036314657): continue_login
Apr 18 20:41:48.839: AAA/AUTHEN (4036314657): status = GETPASS
Apr 18 20:41:48.839: AAA/AUTHEN (4036314657): METHOD=KRB5
Apr 18 20:41:48.847: Kerberos: Requesting TGT with expiration date
    of 861424908
Apr 18 20:41:48.851: Kerberos: Sending TGT request with no
    pre-authorization data.
Apr 18 20:41:48.875: Kerberos: Sent TGT request to KDC
Apr 18 20:41:49.675: Kerberos: Received TGT reply from KDC
Apr 18 20:41:49.795: Kerberos: Sent TGT request to KDC
Apr 18 20:41:50.119: Kerberos: Received TGT reply from KDC
Apr 18 20:41:50.155: AAA/AUTHEN (4036314657): password incorrect
Apr 18 20:41:50.159: AAA/AUTHEN (4036314657): status = FAIL
Apr 18 20:41:52.235: AAA/AUTHEN: free user cisco1 tty51 171.68.109.64
    authen_TYPE=ASCII service=LOGIN priv=1
Apr 18 20:41:52.239: AAA/AUTHEN: create_user user='' ruser=''
    port='tty51' rem_addr='171.68.109.64' authen_TYPE=ASCII
    service=LOGIN priv=1
Apr 18 20:41:52.243: AAA/AUTHEN/START (0): port='tty51' list='cisco2' A
    CTION=LOGIN service=LOGIN
Apr 18 20:41:52.243: AAA/AUTHEN/START (1957396): found list
Apr 18 20:41:52.247: AAA/AUTHEN/START (1817975874): METHOD=KRB5
Apr 18 20:41:52.247: AAA/AUTHEN (1817975874): status = GETUSER
Apr 18 20:42:08.143: AAA/AUTHEN/ABORT: (1817975874) because
    Carrier dropped.
Apr 18 20:42:08.147: AAA/AUTHEN: free user tty51 171.68.109.64
    authen_TYPE=ASCII service=LOGIN priv=1
-----

```

Here is what the user sees:

```

$telnet 10.10.110.245
Trying 10.10.110.245 ...
Connected to 10.10.110.245.
Escape character is '^]'.

```

User Access Verification

```

Username: cisco1
Password:
Kerberos: Failed to retrieve temporary service credentials!
Kerberos: Failed to validate TGT!
% Access denied

```

Username:

Client Not In Kerberos Database

```

Apr 18 19:04:49.983: AAA/AUTHEN: create_user user=''
    ruser='' port='tty51' rem_addr='171.68.109.64' authen_TYPE=ASCII
    service=LOGIN priv=1
Apr 18 19:04:49.987: AAA/AUTHEN/START (0): port='tty51' list='cisco2'
    ACTION=LOGIN service=LOGIN

```

```

Apr 18 19:04:49.987: AAA/AUTHEN/START (1957396): found list
Apr 18 19:04:49.991: AAA/AUTHEN/START (3962282505): METHOD=KRB5
Apr 18 19:04:49.995: AAA/AUTHEN (3962282505): status = GETUSER
Apr 18 19:04:53.475: AAA/AUTHEN/CONT (3962282505): continue_login
Apr 18 19:04:53.479: AAA/AUTHEN (3962282505): status = GETUSER
Apr 18 19:04:53.479: AAA/AUTHEN (3962282505): METHOD=KRB5
Apr 18 19:04:53.483: AAA/AUTHEN (3962282505): status = GETPASS
Apr 18 19:04:56.283: AAA/AUTHEN/CONT (3962282505): continue_login
Apr 18 19:04:56.283: AAA/AUTHEN (3962282505): status = GETPASS
Apr 18 19:04:56.287: AAA/AUTHEN (3962282505): METHOD=KRB5
Apr 18 19:04:56.291: Kerberos: Requesting TGT with expiration date
    of 861419096
Apr 18 19:04:56.295: Kerberos: Sending TGT request with no
    pre-authorization data.
Apr 18 19:04:56.323: Kerberos: Sent TGT request to KDC
Apr 18 19:04:56.355: Kerberos: Received TGT reply from KDC
Apr 18 19:04:56.363: Kerberos: Client not found in Kerberos database
    ~~~~~
Apr 18 19:04:56.371: Kerberos: Received invalid credential.
Apr 18 19:04:56.375: AAA/AUTHEN (3962282505): password incorrect
Apr 18 19:04:56.379: AAA/AUTHEN (3962282505): status = FAIL
Apr 18 19:04:58.679: AAA/AUTHEN: free user cisco3 tty51 171.68.109.64
    authen_TYPE=ASCII service=LOGIN priv=1
Apr 18 19:04:58.687: AAA/AUTHEN: create_user user='' ruser=''
    port='tty51' rem_addr='171.68.109.64' authen_TYPE=ASCII
    service=LOGIN priv=1
Apr 18 19:04:58.687: AAA/AUTHEN/START (0): port='tty51' list='cisco2'
    ACTION=LOGIN service=LOGIN
Apr 18 19:04:58.691: AAA/AUTHEN/START (1957396): found list
Apr 18 19:04:58.743: AAA/AUTHEN/START (1209738018): METHOD=KRB5
Apr 18 19:04:58.747: AAA/AUTHEN (1209738018): status = GETUSER
Apr 18 19:05:04.863: AAA/AUTHEN/ABORT: (1209738018) because
    Carrier dropped.
Apr 18 19:05:04.863: AAA/AUTHEN: free user tty51 171.68.109.64
    authen_TYPE=ASCII service=LOGIN priv=1

```

Client Is In Database but uses Wrong Password

```

Apr 18 19:06:05.427: AAA/AUTHEN: create_user user='' ruser=''
    port='tty51' rem_addr='171.68.109.64' authen_TYPE=ASCII
    service=LOGIN priv=1
Apr 18 19:06:05.427: AAA/AUTHEN/START (0): port='tty51' list='cisco2'
    ACTION=LOGIN service=LOGIN
Apr 18 19:06:05.431: AAA/AUTHEN/START (1957396): found list
Apr 18 19:06:05.431: AAA/AUTHEN/START (3693437965): METHOD=KRB5
Apr 18 19:06:05.435: AAA/AUTHEN (3693437965): status = GETUSER
Apr 18 19:06:07.763: AAA/AUTHEN/CONT (3693437965): continue_login
Apr 18 19:06:07.763: AAA/AUTHEN (3693437965): status = GETUSER
Apr 18 19:06:07.767: AAA/AUTHEN (3693437965): METHOD=KRB5
Apr 18 19:06:07.767: AAA/AUTHEN (3693437965): status = GETPASS
Apr 18 19:06:14.895: AAA/AUTHEN/CONT (3693437965): continue_login
Apr 18 19:06:14.899: AAA/AUTHEN (3693437965): status = GETPASS
Apr 18 19:06:14.899: AAA/AUTHEN (3693437965): METHOD=KRB5
Apr 18 19:06:14.907: Kerberos: Requesting TGT with expiration date
    of 861419174
Apr 18 19:06:14.907: Kerberos: Sending TGT request with no
    pre-authorization data.
Apr 18 19:06:14.935: Kerberos: Sent TGT request to KDC
Apr 18 19:06:15.643: Kerberos: Received TGT reply from KDC
Apr 18 19:06:15.683: Kerberos: Received invalid credential.
Apr 18 19:06:15.687: AAA/AUTHEN (3693437965): password incorrect
    ~~~~~
Apr 18 19:06:15.691: AAA/AUTHEN (3693437965): status = FAIL
Apr 18 19:06:17.695: AAA/AUTHEN: free user cisco1 tty51 171.68.109.64
    authen_TYPE=ASCII service=LOGIN priv=1

```

```
Apr 18 19:06:17.699: AAA/AUTHEN: create_user user='' ruser=''
    port='tty51' rem_addr='171.68.109.64' authen_TYPE=ASCII
    service=LOGIN priv=1
Apr 18 19:06:17.703: AAA/AUTHEN/START (0): port='tty51' list='cisco2'
    ACTION=LOGIN service=LOGIN
Apr 18 19:06:17.703: AAA/AUTHEN/START (1957396): found list
Apr 18 19:06:17.707: AAA/AUTHEN/START (1568599595): METHOD=KRB5
Apr 18 19:06:17.707: AAA/AUTHEN (1568599595): status = GETUSER
Apr 18 19:06:22.751: AAA/AUTHEN/ABORT: (1568599595) because
    Carrier dropped.
Apr 18 19:06:22.755: AAA/AUTHEN: free user    tty51 171.68.109.64
    authen_TYPE=ASCII service=LOGIN priv=1
```

The user sees this output:

```
Trying 10.10.110.245 ...
Connected to 10.10.110.245.
Escape character is '^]'.


```

User Access Verification

```
Username: cisco1
Password:
% Access denied
```

Username:

SRVTAB Entry Not Correct on Router

```
pppcisco1#
%SYS-5-CONFIG_I: Configured from console by vty0 (171.68.109.64)
Apr 18 19:08:55.799: AAA/AUTHEN: create_user user='' ruser=''
    port='tty51' rem_addr='171.68.109.64' authen_TYPE=ASCII
    service=LOGIN priv=1
Apr 18 19:08:55.803: AAA/AUTHEN/START (0): port='tty51' list='cisco2'
    ACTION=LOGIN service=LOGIN
Apr 18 19:08:55.807: AAA/AUTHEN/START (1957396): found list
Apr 18 19:08:55.807: AAA/AUTHEN/START (3369934519): METHOD=KRB5
Apr 18 19:08:55.811: AAA/AUTHEN (3369934519): status = GETUSER
Apr 18 19:08:59.011: AAA/AUTHEN/CONT (3369934519): continue_login
Apr 18 19:08:59.011: AAA/AUTHEN (3369934519): status = GETUSER
Apr 18 19:08:59.015: AAA/AUTHEN (3369934519): METHOD=KRB5
Apr 18 19:08:59.015: AAA/AUTHEN (3369934519): status = GETPASS
Apr 18 19:09:02.219: AAA/AUTHEN/CONT (3369934519): continue_login
Apr 18 19:09:02.219: AAA/AUTHEN (3369934519): status = GETPASS
Apr 18 19:09:02.223: AAA/AUTHEN (3369934519): METHOD=KRB5
Apr 18 19:09:02.231: Kerberos: Requesting TGT with expiration date
    of 861419342
Apr 18 19:09:02.231: Kerberos: Sending TGT request with no
    pre-authorization data.
Apr 18 19:09:02.259: Kerberos: Sent TGT request to KDC
Apr 18 19:09:02.311: Kerberos: Received TGT reply from KDC
Apr 18 19:09:02.435: Kerberos: Sent TGT request to KDC
Apr 18 19:09:02.555: Kerberos: Received TGT reply from KDC
Apr 18 19:09:02.643: AAA/AUTHEN (3369934519): password incorrect
Apr 18 19:09:02.643: AAA/AUTHEN (3369934519): status = FAIL
Apr 18 19:09:04.779: AAA/AUTHEN: free user cisco1 tty51 171.68.109.64
    authen_TYPE=ASCII service=LOGIN priv=1
Apr 18 19:09:04.783: AAA/AUTHEN: create_user user='' ruser=''
    port='tty51' rem_addr='171.68.109.64' authen_TYPE=ASCII
    service=LOGIN priv=1
Apr 18 19:09:04.787: AAA/AUTHEN/START (0): port='tty51' list='cisco2'
    ACTION=LOGIN service=LOGIN
Apr 18 19:09:04.791: AAA/AUTHEN/START (1957396): found list
```

```
Apr 18 19:09:04.843: AAA/AUTHEN/START (2592922252): METHOD=KRB5
Apr 18 19:09:04.843: AAA/AUTHEN (2592922252): status = GETUSER
Apr 18 19:09:11.751: AAA/AUTHEN/ABORT: (2592922252) because
Carrier dropped.
Apr 18 19:09:11.755: AAA/AUTHEN: free user   tty51 171.68.109.64
authen_TYPE=ASCII service=LOGIN priv=1
```

Here is what the user sees:

```
Trying 10.10.110.245 ...
Connected to 10.10.110.245.
Escape character is '^]'.


```

```
User Access Verification


```

```
Username: cisco1
Password:
Failed to retrieve SRVTAB key!
Kerberos:      Failed to validate TGT!
% Access denied


```

```
Username:
```

References

1. Kerberos V5 *System Administrator's Guide* (comes in a tarred, g-zipped file)
2. Kerberos V5 *Installation Guide*
3. Kerberos V5 *UNIX User's Guide*
4. Kerberos: The Network Authentication Protocol
5. The Kerberos Network Authentication Service (USC/ISI's GOST Group)
6. Jennifer G. Steiner, Clifford Neuman, Jeffrey I. Schiller. "Kerberos: An Authentication Service for Open Network Systems ", USENIX Mar 1988
7. S. P. Miller, B. C. Neuman, J. I. Schiller, and J. H. Saltzer, "Kerberos Authentication and Authorization System," 12/21/87
8. R. M. Needham and M. D. Schroeder, "Using Encryption for Authentication in Large Networks of Computers," Communications of the ACM, Vol. 21(12), pp. 993-999 (December, 1978)
9. V. L. Voydock and S. T. Kent, "Security Mechanisms in High-Level Network Protocols," *Computing Surveys*, Vol. 15(2), ACM (June 1983)
10. Li Gong, "A Security Risk of Depending on Synchronized Clocks", *Operating Systems Review*, Vol 26, #1, pp 49-53
11. C. Neuman and J. Kohl, "The Kerberos Network Authentication Service (V5)," RFC 1510, September 1993
12. B. Clifford Neuman and Theodore Ts'o, "Kerberos: An Authentication Service for Computer Networks," IEEE Communications, 32(9), September 1994

Note: Many of these documents, that includes the one by Neuman, Schiller, and Steiner (#9) are also available via FTP from MIT Athena System --- Kerberos Documentation . In order to obtain copies of RFCs, refer to the Obtaining RFCs and Standards Documents.

Related Information

- [Kerberos Support Page](#)
 - [Kerberos in IOS Documentation](#)
 - [Technical Support – Cisco Systems](#)
-

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2009 – 2010 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

Updated: Jan 19, 2006

Document ID: 13805
