# Table of Contents

SSH FAQ for CERN

- Desired authentication method
- Kerberos(5), GSSAPI
- Password
- Public/Private Key

Current authentication method

Common errors/Debugging/Troubleshooting

- X11 forwarding errors
- Kerberos authentication, but no AFS access
- Public key authentication, but no AFS access
- Kerberos authentication not working
- Password authentication, but no Kerberos/AFS credentials on login

- Cannot connect, or session closes immediately
- Further troubleshooting on the client
- Further troubleshooting on server

Resources
SSH FAQ for CERN

(the following document tries to answer the most common questions around SSH at CERN. It is aimed largely at 2nd-level support, but of course individual users are encouraged to use it for self-help as far as possible – but knowledge of certain concepts is assumed. Below “foobar” is used as placeholder for a CERN AFS account, LXPLUS is used as a typical (and usually known-to-be-working) CERN Linux machine, a local "client" is the machine where ssh is run, the remote "server" is where sshd runs (and the client want to connect).

**Desired authentication method**

The "authentication method" is the way your local machine convinces the remote machine that you are who you claim to be. The remote server will also check that you are allowed to log into the machine at the same time – failure on either looks indistinguishable to the client (i.e the client will not know whether a wrong password was used, or whether the password was OK but the remote side doesn't let you log in anyway). The various methods have their advantages and drawbacks, mostly in how far the remote side will be able to act on your behalf. Please keep in mind that access to a remote machine does not automatically allow access to AFS from there – these are two separate systems that both need to be convinced to grant access.

Something else to be aware of is that several authentication methods may be tried one after the other – this occasionally leads to spurious errors if the "usual" authentication method does not work for some reason, and the next one does – but behaves differently.

**Kerberos(5), GSSAPI**

This method uses the Kerberos protocol (GSSAPI is a more generic wrapper protocol and for the purposes of this document equivalent to Kerberos) to establish your identity, and usually would be the default to be used between Linux/UNIX machines (e.g your SLC desktop to LXPLUS). It requires the local user to have a valid Kerberos credential (called "ticket"), and for the remote server to be registered in the Kerberos database (KDC) at CERN. Kerberos is sensitive to bad time settings on either client and server, so you might want to run ntpd on both. You also will need to type your password about once/day on the client (e.g. in a screensaver, or via kinit), since Kerberos credentials expire. If everything is working fine, Kerberos is transparent, fast and reliable.

Kerberos allows to transfer your "identity" (your ticket granting ticket, TGT) to the remote side, so that you can use other services from the remote machine. This is a security risk since a compromised machine will be able to impersonate you, but this risk is mitigated by the fact that these TGTs are only valid for a short period (as opposed to a password, that once captured will stay valid for several months). TGT forwarding is actually required to access AFS from the remote machine, so or SLC, this is turned on by default (but might be different for other systems). When credential forwarding (and other forwardings: ssh agent, X11, tunneling) has been turned off, it is safe to use Kerberos even against a compromised (or suspected) machine. See http://cern.ch/linux/docs/kerberos-access.shtml for generic instruction on how to get this access method working.

**Password**

This is the safe fallback authentication method – the user is simply prompted for the password, this is validated and (if valid) used to get Kerberos and AFS credentials on the remote side. Drawbacks include the need to type something (so cumbersome if moving a lot between machines), and connecting to a compromised server will expose your password (i.e fully compromise your account). At CERN, this method is used usually from a Windows client to a Linux server (since Kerberos does not work (yet) between the two operating systems.)
Public/Private Key

This method uses cryptographic keys stored in files. Usually these keys are encrypted (and so need human interaction to be used), but they can also be stored for the duration of a session in unencrypted form in memory (in the so-called ssh-agent). It is also possible to store them unencrypted on disk, at which point they can be used by anybody with read access to the key file – a strong security risk, but occasionally required (e.g. unattended sessions between service accounts). The huge drawback of this authentication method is that

- no AFS access is possible on the remote side
- setting up an AFS account for Public key access is tricky due to the per-directory access control (and the fact that the remote sshd does not have any special rights on AFS)

This authentication method should therefore be avoided at CERN, users regularly report issues. Unfortunately, it has been recommended by some services like CVS in the past (usually geared towards Windows users, where no actual write access to AFS was required - very much a special case), and is therefore configured for several accounts. For the specific case of user cron jobs, the CERN acrontab command will provide a cron job with fresh credentials on every invocation.

Current authentication method

As stated above, the authentication methods will be negotiated (or simply tried out) between client and server. As such it is required to know which method has been used for a particular connection. Luckily the ssh verbose output contains this: ssh -v -l foobar lxplus.cern.ch look for a line that says "Authentication succeeded". Typical values:

- Authentication succeeded (gssapi-with-mic).
- Kerberos v5 authentication accepted.
  - Kerberos/GSSAPI. Also check for a line that says "Delegating credentials " or Kerberos v5 TGT forwarded (foobar@CERN.CH)
- Authentication succeeded (publickey).
- RSA authentication accepted by server.
  - These two indicate the use of public keys, i.e AFS is unlikely to work
- Authentication succeeded (password).
- Doing password authentication.
  - (rather evident, since both will have prompted for your password)

To reiterate, it is utterly pointless to report an ssh issue at CERN without including the above information, or better, the full "ssh -v ..." output. Other information usually required is

- client hostname and OS: `uname -a` (on Linux)
- server hostname and OS (dito). Please note that e.g. LXPLUS is a whole set of machines, troubleshooting is made easier if the exact machine concerned can be identified (is usually included in the ssh -v output, look for "Connecting to lxplus [137.138.5.220]")
- exact timestamp of the connection attempt: `date; ntpdate -q ip-time-1.cern.ch`
- information on Kerberos credentials on the client (if any): `klist`
- (did we mention "ssh -v ...")

In the most common scenario, the user would like to use Kerberos authentication but somehow ends up with Password
Common errors/Debugging/Troubleshooting

(a collection of issues seen already at CERN). One important point is to check whether the problem is related to the individual user on the client, the individual target user on the server, all users on the client machine, or all users on the remote server – this will make troubleshooting easier.

- One user on one client machine only: other users on same client work OK against the same server, affected user on some other client is fine as well. This is typically a "local" (non-AFS) account, so look for account-specific things (~/.ssh, user/shell environment)
- One user on all clients: other users on same client(s) work OK against the same server. Typical for an AFS account (which in fact has the same setting on all clients); again look for account-specific things (~/.ssh, user/shell environment)
- All users on one client: other users on the same client see the same issue when connecting to different servers. Look for host-specific things: client OS, ssh or kerberos client software version, /etc/ssh/sshd_config, local time, local network
- All users on one server: other users from different clients see the same issue when connecting to the same server (but other servers are OK). Look for server-specific things: server OS, ssh or kerberos server versions, /etc/ssh/sshd_config, server time, server network and DNS configuration.
- None of the above: spurious issue (cannot be reproduced at will) – suspect expiring credentials, borderline time synchronization between client and server, DNS alias issues (i.e one bad LXPLUS machine among many), software bugs. Since most of the time ssh works fine for most people, you will need to narrow this down..

X11 forwarding errors

Error message

/usr/X11R6/bin/xauth: timeout in locking authority file /afs/cern.ch/user/f/foo/.Xauthority
hepix: E: /usr/bin/fs returned error, no tokens?

This is a very common symptom, and not necessarily an actual " X11 forwarding error": it just so happens that forwarding the X11 credentials is the very first thing that requires write access to the user's home directory on AFS – so usually the underlying cause is that no AFS token has been transferred or obtained, so the user will not be able to write anything into AFS (and not read anything protected either). See next two sections 1, 2.

Other than that, it could be that the AFS home directory quota has been exceeded (should say something rather clear to this effect at login), and this error message has also occurred e.g on corrupted AFS home directories - see KerberosViaSSHNoAFS for some other (rare) cases.

In general, X11 forwarding needs some ingredients:

- on the client
  - the X11 server needs to be running and be visible to the ssh client (typically via the DISPLAY environment variable)
  - the ssh client needs to be able to get at the X11 authenticator ("MIT-MAGIC-COOKIE"), via `xauth list $DISPLAY`
  - the ssh client needs to be told to try X11 forwarding (-X option on the command line, `ForwardX11 yes` in the config files. -Y might be required for some X11 programs that grab pointers and such).
- on the server
  - sshd needs to be told to accept X11 forwardings (X11Forwarding yes in `/etc/ssh/sshd_config`
  - the xauth program needs to be executable (i.e installed)
  - the user's home directory (~/.Xauthority and lock files) needs to be writable
the X11 applications on the server need to use the DISPLAY variable as provided by ssh (and
not override that variable e.g. to point directly at the client)

Kerberos authentication, but no AFS access

Typically caused by not forwarding the Kerberos credentials (TGT) to the remote side. Quick test:

```bash
kinit -f foobar@CERN.CH

ssh -2 -v -oGSSAPIAuthentication=yes -oGSSAPIDelegateCredentials=yes -l foobar
lxplus.cern.ch klist\;tokens
```

if this works (and shows that both the TGT and and AFS token are available on the remote machine), you'll
need to find the difference between this and the normal user environment. Issues to check on the client side:

- is the TGT marked as "forwardable"? see klist -f, look for Flags: ..F. If not, check
  /etc/krb5/conf, [libdefaults] section, forwardable = true
- is the TGT for the CERN realm? If not, users needs to explicitly kinit ..@CERN.CH
- is the TGT transferred automatically? If not, check /etc/ssh/ssh_config or the user's local
  ~/.ssh/config and set the two GSSAPI.. settings from above.
- If the client is a "recent" (in 2010) Linux (Fedora/Ubuntu/Debian-less-than-stable), you might be
  affected by a security feature in recent MIT Kerberos versions (RH Bugzilla). A workaround is to
  set "allow_weak_crypto = true" in the client's /etc/krb5.conf, [libdefaults] section

On the server side (assuming a SLC-lookalike, other OSes and distributions will vary)

- check that PAM is configured to invoke pam_krb5afs.so for both "auth" and "session" (typically
done in /etc/pam.d/system-auth)
- check /etc/krb5.conf, [appdefault] section, "pam" stanza has "external = true" (or "external=sshd")
- you might want to set the "debug" option there as well (and make sure that your syslog
  configuration puts LOG_DEBUG messages somewhere, instead of discarding them)

Public key authentication, but no AFS access

(as explained above, this is expected behaviour). Use Kerberos or Password authentication, either by
removing the public keys from ~/.ssh/authorized_keys on the remote machine (LXPLUS), or by explicit
configuration/command-line options:

```bash
ssh -v -2 -oPubkeyAuthentication=no -l foobar lxplus.cern.ch
```

This method occasionally is chosen when Kerberos credentials have expired (usually Kerberos will be tried
first). In these cases, please check local credentials via klist (and run kinit to refresh if expired). You also
should check that you have no unencrypted private keys (you should have been prompted for your passphrase)
- such unencrypted keys are a major security risk, since much like a stored password they stay valid for
extended periods of time.

Kerberos authentication not working

(See http://cern.ch/linux/docs/kerberos-access.shtml)

Ingredients on the client (can be checked by the user)
• valid Kerberos configuration – compare to lxplus.cern.ch:/etc/krb5.conf
• kinit foobar@CERN.CH is successful
• klist -f shows valid forwardable TGT for CERN.CH
• ssh client attempts to do gssapi-with-mic authentication (ssh -v -2 ..., look for "Next authentication method: gssapi-with-mic")
• client and server local clocks are more-or-less in sync (ideally: run ntp)
• ssh client version knows how to cope with DNS-loadbalanced machines (such as "LXPLUS" or "ISSCVS"). Might need the ssh option GSSAPITrustDNS yes if the following error is seen:

    debug1: An invalid name was supplied

and/or if "klist" after a connection attempt shows a service ticket for a different machine than the one we connected to. See above link for details.

Ingredients on the server (need help from server admin):

• valid Kerberos configuration – compare to lxplus.cern.ch:/etc/krb5.conf
   ♦ in particular, check that the 'external' flag in the applications section is set.
• Kerberos server identity exists: /etc/krb5.keytab is on server, readable by root (only)
   ♦ check via klist -k that the stored identity in that file matches your machine hostname
   ♦ also check "key version number". Run kvno host/lxplus123.cern.ch@CERN.CH from a client while holding a valid TGT, compare to above KVNO column
   ♦ if missing/incorrect/bad kvno: run cern-config-keytab -v -f on server to fix, run kinit -R on client (since existing tickets will get invalidated)
• consistent hostname:
   ♦ hostname is fully qualified (lxplus123.cern.ch instead of lxplus)
   ♦ hostname agrees with reverse DNS (both from client, and on server – check /etc/hosts)
• client and server local clocks are more-or-less in sync (ideally: run ntpd on both)
• sshd is configured to allow GSSAPIAuthentication=yes
   ♦ check /etc/sshd/sshd_conf on server, or
   ♦ check ssh -v ..., look for " Authentications that can continue: .. gssapi-with-mic"
• has one of the machines (client or server been migrated partially to CERN's new KDC (Active Directory-based), but not the other? use klist -e to show the encryption types: DES or 3DES indicate tickets from the Heimdal/AFS KDC, ArcFour indicates tickets from Active Directory. These should work transparently, but might indicate trouble with the migration - please include these details in any report.

Password authentication, but no Kerberos/AFS credentials on login

Typically means that the server hasn't been configured to use Kerberos/AFS. For SLC desktops, run lcm --configure --all, check for error messages (and fix), reboot.

Other operating systems (largely: other Linux distributions, input for Mac/Solaris would be welcome):

• check /etc/pam.d/sshd (or gdm, kdm, login ..), usually redirects to /etc/pam.d/system-auth
   ♦ needs to have something (exact syntax may vary) like the below, in addition to local account checks. Warning, syntax errors may lock you out of the machine.

        auth sufficient pam_krb5afs.so try_first_pass minimum_uid=100
        session required pam_krb5afs.so try_first_pass

• compare /etc/krb5.conf (or local equivalent) to version on lxplus.
• Troubleshooting PAM: add debug option to above lines, configure syslogd to record *.debug to some file.
• see http://cern.ch/linux/docs/kerberos-access.shtml.
Cannot connect, or session closes immediately

- Check whether server is up (ping – machine might be down/halted)
- Check whether other services on the server can be reached (machine might have crashed/deadlocked)
- Check whether the client can reach other servers (client network/firewall issue?)
- Check whether other clients can reach the same server (server network/firewall issue?)
- ssh closing the connection immediately:
  - ♦ Could be tcp_wrappers (check /etc/hosts.allow, /etc/hosts.deny) – rare.
  - ♦ ssdh has an overload protection where new connections may get refused while too many requests are being processed (i.e not authenticated or rejected yet). This can lead to the machine "ping" ing fine, and other services being reachable, but ssh connections to fail (for everybody). If overload, will clear up by itself. If not (kernel deadlock), restarting ssdh will help (if some form of console login is possible), or a reboot might be required. You may want to check for processes stuck in "D" state on the server, and/or for file system errors (dead disks typically cause this).

Further troubleshooting on the client

Unfortunately the most "interesting" messages are not being propagated to the client since they might give valuable information to an attacker. However, some of the ssh -v .. output might be useful - and as shown above, the absence of some messages indicate whether a particular method has failed.

- [FIXME: time desync client server: weirdly-numbered GSSAPI errors]
- [FIXME: client reads public keys at startup]

Further troubleshooting on server

- look at /var/log/secure, /var/log/messages
- last resort: use debug mode on the server: /usr/sbin/sshd -d -d -d -r -p 1234
  - ♦ client connects via ssh -v -p 1234 ...
  - ♦ server will need to be restarted after every attempt
  - ♦ be aware of server-local firewalls that may prevent connections to non-standard ports

Resources

- http://www.snailbook.com/faq/ - the O'Reilly "Snail book" is considered the dead-tree reference to SSH
- http://www.openssh.com/faq.html - the upstream "question and answers"
- http://cern.ch/security/ssh/ - CERN's security team recommendations (and restrictions) on SSH at CERN
- MIT Kerberos-V5-Library-Error-Codes - translate Kerberos error codes such as
  - ♦ Unknown code krb5 144 : KRB5KRB_AP_WRONG_PRINC: Wrong principal in request
  - ♦ Unknown code krb5 195 : KRB5_FCC_NOFILE: No credentials cache found
  - ♦ Unknown code krb5 7 : KRB5KDC_ERR_S_PRINCIPAL_UNKNOWN: Server not found in Kerberos database
- Kerberos MigrationFAQ for the transition from Heimdal to Active Directory
- CERN helpdesk - please include the results of your investigations so far, otherwise you'll simply be pointed back this page