链滴

centos7 升级 openssh 到最新版本（open ssh-8.0p1）

作者：expoli
原文链接：https://ld246.com/article/1568940290199
来源网站：链滴

centos7 升级openssh到最新版本（openssh-8.0p1）

0. 注意事项

整个过程不需要卸载原先的openssl包和openssh的rpm包、不影响我们的操作。

本文的环境都是系统自带的openssh，没有经历过手动编译安装方式。如果之前有手动编译安装过openssh，请参照本文自行测试是否能成功。

1. 原系统信息（未升级前）

1.1 系统版本

```
# cat /etc/redhat-release
CentOS Linux release 7.7.1908 (Core)
```

1.2 原 openssl 版本

```
# openssl version
OpenSSL 1.0.2k-fips  26 Jan 2017
```

1.3 原 openssh 版本

```
# ssh -V
OpenSSH_7.4p1, OpenSSL 1.0.2k-fips  26 Jan 2017
```

2. 配置更新环境

2.1 yum 更新 openssh

yum update openssh升级下到目前yum仓库默认的openssh7.4p1版本。（这里准备统一openssh本为7.4p1之后再统一编译安装升级到openssh8.0p1）

```
# openssl version
OpenSSL 1.0.2k-fips  26 Jan 2017
# ssh -V
OpenSSH_7.4p1, OpenSSL 1.0.2k-fips  26 Jan 2017
```

2.2 安装telnet-server以及xinetd

因为我们现在是远程更新 openssh 所以需要先使用另一种连接方式连接到服务器进行相关操作。

```
yum install xinetd telnet-server -y
已加载插件：fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirrors.tuna.tsinghua.edu.cn
 * extras: mirrors.tuna.tsinghua.edu.cn
 * nux-dextop: mirror.li.nux.ro
 * updates: mirrors.tuna.tsinghua.edu.cn
```

2.3 配置telnet

现在很多centos7版本安装telnet-server以及xinetd之后没有一个叫telnet的配置文件了。

如果下面telnet文件不存在的话，可以跳过这部分的更改（我所测试的时候并没有此项文件）。

```
# ll /etc/xinetd.d/telnet
ls: 无法访问/etc/xinetd.d/telnet: 没有那个文件或目录
```

如果下面文件存在，请更改配置telnet可以root登录，把disable = no改成disable = yes

```
# cat /etc/xinetd.d/telnet
# default: on
# description: The telnet server serves telnet sessions; it uses \
#   unencrypted username/password pairs for authentication.
service telnet
{
   disable = no
   flags       = REUSE
   socket_type = stream
   wait        = no
   user        = root
   server      = /usr/sbin/in.telnetd
   log_on_failure  += USERID
}

[root@rhel yum.repos.d]# vim /etc/xinetd.d/telnet
[root@rhel yum.repos.d]# cat /etc/xinetd.d/telnet
# default: on
# description: The telnet server serves telnet sessions; it uses \
#   unencrypted username/password pairs for authentication.
```

```
service telnet
{
   disable = yes
   flags       = REUSE
   socket_type = stream
   wait        = no
   user        = root
   server      = /usr/sbin/in.telnetd
   log_on_failure  += USERID
}
```

2.4 配置telnet登录的终端类型

在 /etc/securetty 文件末尾增加一些pts终端，如下

```
pts/0
pts/1
pts/2
pts/3
```

2.5 启动telnet服务，并设置开机自动启动

```
# systemctl start telnet.socket
# systemctl enable telnet.socket
Created symlink from /etc/systemd/system/sockets.target.wants/telnet.socket to /usr/lib/syst
md/system/telnet.socket.

# systemctl restart telnet.socket
# systemctl status  telnet.socket
● telnet.socket - Telnet Server Activation Socket
   Loaded: loaded (/usr/lib/systemd/system/telnet.socket; disabled; vendor preset: disabled)
   Active: active (listening) since 四 2019-09-19 19:33:58 CST; 14s ago
     Docs: man:telnetd(8)
   Listen: [::]:23 (Stream)
 Accepted: 0; Connected: 0

9月 19 19:33:58 sz-lab-centos7-gitlab-nginx-proxy-192.168.178.46 systemd[1]: Closed Telnet
erver Activation Socket.
9月 19 19:33:58 sz-lab-centos7-gitlab-nginx-proxy-192.168.178.46 systemd[1]: Stopping Telne
 Server Activation Socket.
9月 19 19:33:58 sz-lab-centos7-gitlab-nginx-proxy-192.168.178.46 systemd[1]: Listening on Te
net Server Activation Socket.
```

2.6 查看端口监听状态、确认 telnet 工作状态

```
# netstat -lntp|grep 23
tcp        0      0 127.0.0.1:9236          0.0.0.0:*               LISTEN      4858/gitaly
tcp        0      0 192.168.122.1:53        0.0.0.0:*               LISTEN      2334/dnsmasq
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      29723/cupsd
tcp6       0      0 :::23                   :::*                    LISTEN      1/systemd
tcp6       0      0 ::1:631                 :::*                    LISTEN      29723/cupsd
```

2.7 添加防火墙规则

```
# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s31f6
  sources:
  services: dhcpv6-client ftp ssh zabbix-agent
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:


# firewall-cmd --add-service=telnet --permanent
success
# firewall-cmd --reload
success
# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s31f6
  sources:
  services:dhcpv6-client ftp ssh telnet zabbix-agent
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

2.8 telnet 连接目标服务器

```
# telnet 192.168.1.2
# 输入用户名密码
# 回车登录
```

2.9 安装依赖包

```
# yum install  -y gcc gcc-c++ glibc make autoconf openssl openssl-devel pcre-devel  pam-de
el
Loading mirror speeds from cached hostfile
 * base: mirrors.tuna.tsinghua.edu.cn
 * extras: mirrors.tuna.tsinghua.edu.cn
 * nux-dextop: mirror.li.nux.ro
 * updates: mirrors.tuna.tsinghua.edu.cn
```

2.10 安装pam和zlib等

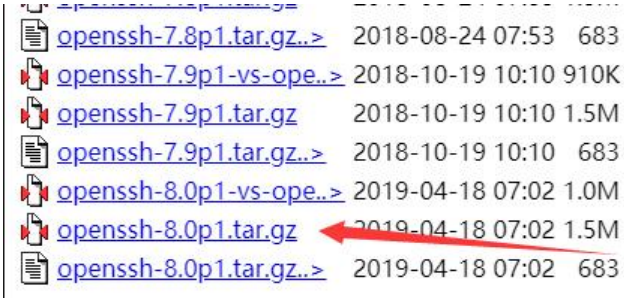
```
# yum install  -y pam* zlib*
已加载插件：fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirrors.tuna.tsinghua.edu.cn
 * extras: mirrors.tuna.tsinghua.edu.cn
 * nux-dextop: mirror.li.nux.ro
 * updates: mirrors.tuna.tsinghua.edu.cn
```

3. 下载所需文件

选择一个你所喜欢的文件夹保存下面你所下载的文件。

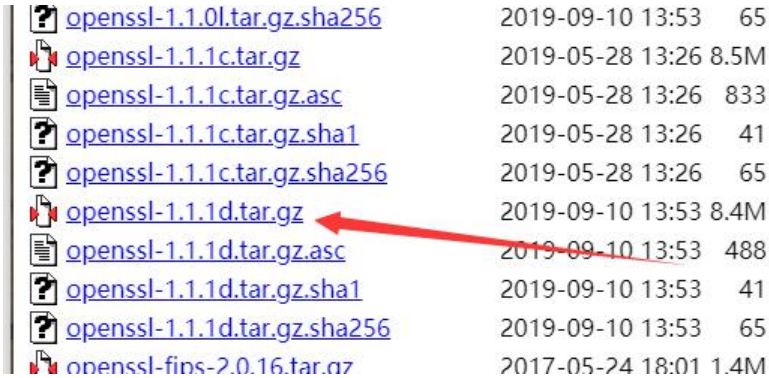
3.1 下载最新版本的 openssh

https://openbsd.hk/pub/OpenBSD/OpenSSH/portable/



3.2 下载最新版本的 openssl

https://ftp.openssl.org/source/




```
# ls
openssh-8.0p1.tar.gz  openssl-1.1.1d.tar.gz
```

4. 开始安装

4.1 安装 openssl

4.1.1 解压缩

```
# tar xfz openssh-8.0p1.tar.gz
# ls
openssh-8.0p1  openssh-8.0p1.tar.gz  openssl-1.1.1d.tar.gz
# tar xfz openssl-1.1.1d.tar.gz
# ls
openssh-8.0p1  openssh-8.0p1.tar.gz  openssl-1.1.1d  openssl-1.1.1d.tar.gz
#
```

4.1.2 备份原文件

```
# ll /usr/bin/openssl
-rwxr-xr-x 1 root root 555288 8月   9 09:38 /usr/bin/openssl
# mv /usr/bin/openssl /usr/bin/openssl_bak
# mv /usr/include/openssl /usr/include/openssl_bak
# ll /usr/include/openssl_bak/
总用量 1864
-rw-r--r-- 1 root root   6146 8月   9 09:38 aes.h
-rw-r--r-- 1 root root  63204 8月   9 09:38 asn1.h
-rw-r--r-- 1 root root  24435 8月   9 09:38 asn1_mac.h
-rw-r--r-- 1 root root  34475 8月   9 09:38 asn1t.h
-rw-r--r-- 1 root root  38742 8月   9 09:38 bio.h
...
```

4.1.3 开始编译安装

```
# cd openssl-1.1.1d/
# ./config shared && make && make install
```

4.1.4 后续配置

查看编译安装后的 openssl 的目录结构、默认安装到 /usr/local 目录下

```
# ls /usr/local/
bin/        etc/        games/      include/    lib/        lib64/      libexec/    sbin/       Server
tatus/ share/       src/        ssl/

# tree -L 2
.
├── bin
│   ├── c_rehash
│   └── openssl
├── include
│   └── openssl
│       ├── aes.h
│       ├── asn1err.h
│       ....
├── lib64
│   ├── engines-1.1
│   ├── libcrypto.a
│   ├── libcrypto.so -> libcrypto.so.1.1
│   ├── libcrypto.so.1.1
│   ├── libssl.a
```

```
│   ├── libssl.so -> libssl.so.1.1
│   ├── libssl.so.1.1
│   └── pkgconfig
└── ssl
    ├── certs
    ├── ct_log_list.cnf
    ├── ct_log_list.cnf.dist
    ├── misc
    ├── openssl.cnf
    ├── openssl.cnf.dist
    └── private
```

4.1.4 软连接 openssl 目录

```
# ln -s /usr/local/bin/openssl /usr/bin/openssl
# ln -s /usr/local/include/openssl/ /usr/include/openssl
# ll /usr/bin/openssl
lrwxrwxrwx 1 root root 22 9月  19 20:14 /usr/bin/openssl -> /usr/local/bin/openssl
#  ll /usr/include/openssl -ld
lrwxrwxrwx 1 root root 27 9月  19 20:14 /usr/include/openssl -> /usr/local/include/openssl/
```

4.1.5 加载新配置

```
echo "/usr/local/lib64"   >> /etc/ld.so.conf

/sbin/ldconfig
```

4.1.6 查看确认版本。没问题

```
# openssl version
OpenSSL 1.1.1d  10 Sep 2019
```

4.2 安装 openssh

4.2.1 解压并设置权限

```
# tar xfz openssh-8.0p1.tar.gz
# cd openssh-8.0p1
# 可能文件默认显示uid和gid数组都是1000，这里重新授权下。不授权可能也不影响安装（请自行测
)
# chown -R root.root /data/tools/openssh-8.0p1
```

4.2.2 备份原ssh的配置文件和目录

```
# mv /etc/ssh/* your_backup_dir
```

4.2.3 配置、编译、安装

1. 查看编译参数、根据需要选择

```
# ./configure -h
`configure' configures OpenSSH Portable to adapt to many kinds of systems.

Usage: ./configure [OPTION]... [VAR=VALUE]...

To assign environment variables (e.g., CC, CFLAGS...), specify them as
VAR=VALUE.  See below for descriptions of some of the useful variables.

Defaults for the options are specified in brackets.

Configuration:
  -h, --help              display this help and exit
      --help=short        display options specific to this package
      --help=recursive    display the short help of all the included packages
  -V, --version           display version information and exit
  -q, --quiet, --silent   do not print `checking ...' messages
      --cache-file=FILE   cache test results in FILE [disabled]
  -C, --config-cache      alias for `--cache-file=config.cache'
  -n, --no-create         do not create output files
      --srcdir=DIR        find the sources in DIR [configure dir or `..']

Installation directories:
  --prefix=PREFIX         install architecture-independent files in PREFIX
                          [/usr/local]
  --exec-prefix=EPREFIX   install architecture-dependent files in EPREFIX
                          [PREFIX]

By default, `make install' will install all the files in
`/usr/local/bin', `/usr/local/lib' etc.  You can specify
an installation prefix other than `/usr/local' using `--prefix',
for instance `--prefix=$HOME'.

For better control, use the options below.

Fine tuning of the installation directories:
  --bindir=DIR            user executables [EPREFIX/bin]
  --sbindir=DIR           system admin executables [EPREFIX/sbin]
  --libexecdir=DIR        program executables [EPREFIX/libexec]
  --sysconfdir=DIR        read-only single-machine data [PREFIX/etc]
  --sharedstatedir=DIR    modifiable architecture-independent data [PREFIX/com]
  --localstatedir=DIR     modifiable single-machine data [PREFIX/var]
  --libdir=DIR            object code libraries [EPREFIX/lib]
  --includedir=DIR        C header files [PREFIX/include]
  --oldincludedir=DIR     C header files for non-gcc [/usr/include]
  --datarootdir=DIR       read-only arch.-independent data root [PREFIX/share]
  --datadir=DIR           read-only architecture-independent data [DATAROOTDIR]
  --infodir=DIR           info documentation [DATAROOTDIR/info]
  --localedir=DIR         locale-dependent data [DATAROOTDIR/locale]
  --mandir=DIR            man documentation [DATAROOTDIR/man]
  --docdir=DIR            documentation root [DATAROOTDIR/doc/openssh]
  --htmldir=DIR           html documentation [DOCDIR]
  --dvidir=DIR            dvi documentation [DOCDIR]
  --pdfdir=DIR            pdf documentation [DOCDIR]
  --psdir=DIR             ps documentation [DOCDIR]
```

```
System types:
  --build=BUILD     configure for building on BUILD [guessed]
  --host=HOST       cross-compile to build programs to run on HOST [BUILD]

Optional Features:
  --disable-option-checking  ignore unrecognized --enable/--with options
  --disable-FEATURE       do not include FEATURE (same as --enable-FEATURE=no)
  --enable-FEATURE[=ARG]  include FEATURE [ARG=yes]
  --disable-largefile     omit support for large files
  --disable-pkcs11        disable PKCS#11 support code [no]
  --disable-strip         Disable calling strip(1) on install
  --disable-etc-default-login Disable using PATH from /etc/default/login no
  --disable-lastlog       disable use of lastlog even if detected no
  --disable-utmp          disable use of utmp even if detected no
  --disable-utmpx         disable use of utmpx even if detected no
  --disable-wtmp          disable use of wtmp even if detected no
  --disable-wtmpx         disable use of wtmpx even if detected no
  --disable-libutil       disable use of libutil (login() etc.) no
  --disable-pututline     disable use of pututline() etc. (uwtmp) no
  --disable-pututxline    disable use of pututxline() etc. (uwtmpx) no

Optional Packages:
  --with-PACKAGE[=ARG]    use PACKAGE [ARG=yes]
  --without-PACKAGE       do not use PACKAGE (same as --with-PACKAGE=no)
  --without-openssl       Disable use of OpenSSL; use only limited internal crypto **EXPERIME
TAL**
  --without-stackprotect  Don't use compiler's stack protection
  --without-hardening     Don't use toolchain hardening flags
  --without-rpath         Disable auto-added -R linker paths
  --with-cflags           Specify additional flags to pass to compiler
  --with-cflags-after     Specify additional flags to pass to compiler after configure
  --with-cppflags         Specify additional flags to pass to preprocessor
  --with-ldflags          Specify additional flags to pass to linker
  --with-ldflags-after    Specify additional flags to pass to linker after configure
  --with-libs             Specify additional libraries to link with
  --with-Werror           Build main code with -Werror
  --with-solaris-contracts Enable Solaris process contracts (experimental)
  --with-solaris-projects Enable Solaris projects (experimental)
  --with-solaris-privs    Enable Solaris/Illumos privileges (experimental)
  --with-osfsia           Enable Digital Unix SIA
  --with-zlib=PATH        Use zlib in PATH
  --without-zlib-version-check Disable zlib version check
  --with-ldns[=PATH]      Use ldns for DNSSEC support (optionally in PATH)
  --with-libedit[=PATH]   Enable libedit support for sftp
  --with-audit=module     Enable audit support (modules=debug,bsm,linux)
  --with-pie              Build Position Independent Executables if possible
  --with-ssl-dir=PATH     Specify path to OpenSSL installation
  --without-openssl-header-check Disable OpenSSL version consistency check
  --with-ssl-engine       Enable OpenSSL (hardware) ENGINE support
  --with-prngd-port=PORT  read entropy from PRNGD/EGD TCP localhost:PORT
  --with-prngd-socket=FILE read entropy from PRNGD/EGD socket FILE (default=/var/run/egd
pool)
  --with-pam              Enable PAM support
```

```
  --with-pam-service=name Specify PAM service name
  --with-privsep-user=user Specify non-privileged user for privilege separation
  --with-sandbox=style    Specify privilege separation sandbox (no, capsicum, darwin, rlimit, se
comp_filter, systrace, pledge)
  --with-selinux          Enable SELinux support
  --with-kerberos5=PATH   Enable Kerberos 5 support
  --with-privsep-path=xxx Path for privilege separation chroot (default=/var/empty)
  --with-xauth=PATH       Specify path to xauth program
  --with-maildir=/path/to/mail    Specify your system mail directory
  --with-mantype=man|cat|doc  Set man page type
  --with-md5-passwords    Enable use of MD5 passwords
  --without-shadow        Disable shadow password support
  --with-ipaddr-display   Use ip address instead of hostname in $DISPLAY
  --with-default-path=    Specify default $PATH environment for server
  --with-superuser-path=  Specify different path for super-user
  --with-4in6             Check for and convert IPv4 in IPv6 mapped addresses
  --with-bsd-auth         Enable BSD auth support
  --with-pid-dir=PATH     Specify location of sshd.pid file
  --with-lastlog=FILE|DIR specify lastlog location common locations

Some influential environment variables:
  CC          C compiler command
  CFLAGS      C compiler flags
  LDFLAGS     linker flags, e.g. -L<lib dir> if you have libraries in a
              nonstandard directory <lib dir>
  LIBS        libraries to pass to the linker, e.g. -l<library>
  CPPFLAGS    (Objective) C/C++ preprocessor flags, e.g. -I<include dir> if
              you have headers in a nonstandard directory <include dir>
  CPP         C preprocessor

Use these variables to override the choices made by `configure' or to help
it to find libraries and programs with nonstandard names/locations.

Report bugs to <openssh-unix-dev@mindrot.org>.
```

2. configure 参数

```
# ./configure --prefix=/usr/ --sysconfdir=/etc/ssh  --with-ssl-dir=/usr/local/lib64 --with-zlib
-with-md5-passwords   --with-pam --with-ssl-engine --with-selinux --with-ipaddr-display
```

3. 安装

```
# make && make install
```

4.3 配置验证（最后）

4.3.1 查看相应的配置文件

```
# ls /etc/ssh/
moduli  ssh_config  sshd_config  ssh_host_dsa_key  ssh_host_dsa_key.pub  ssh_host_ecdsa_key
 ssh_host_ecdsa_key.pub  ssh_host_ed25519_key  ssh_host_ed25519_key.pub  ssh_host_rsa_ke
  ssh_host_rsa_key.pub
```

4.3.2 修改 sshd 配置文件

```
# vim /etc/ssh/sshd_config
```

4.3.3 配置启动文件

从原先的解压的包中拷贝一些文件到目标位置（如果目标目录存在就覆盖）

```
# cp -a contrib/redhat/sshd.init /etc/init.d/sshd
# cp -a contrib/redhat/sshd.pam /etc/pam.d/sshd.pam
# chmod +x /etc/init.d/sshd

# chkconfig --add sshd
# systemctl enable sshd
```

把原先的systemd管理的sshd文件删除或者移走或者删除，不移走的话影响我们重启sshd服务

```
# mv  /usr/lib/systemd/system/sshd.service your_backup_dir
```

4.3.4 配置开机启动

```
# chkconfig sshd on
Note: Forwarding request to 'systemctl enable sshd.socket'.
Created symlink from /etc/systemd/system/sockets.target.wants/sshd.socket to /usr/lib/syst
md/system/sshd.socket.
```

4.3.5 接下来测试启停服务

```
# /etc/init.d/sshd restart
Restarting sshd (via systemctl):                          [  确定  ]

# netstat -lntp | grep 22
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      26069/sshd
tcp6       0      0 :::22                   :::*                    LISTEN      26069/sshd

# /etc/init.d/sshd stop
Stopping sshd (via systemctl):                            [  确定  ]
# netstat -lntp | grep 22

# /etc/init.d/sshd start
Starting sshd (via systemctl):                            [  确定  ]
# systemctl status sshd
● sshd.service - SYSV: OpenSSH server daemon
   Loaded: loaded (/etc/rc.d/init.d/sshd; bad; vendor preset: enabled)
   Active: active (running) since 四 2019-09-19 20:39:57 CST; 11s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 26229 ExecStop=/etc/rc.d/init.d/sshd stop (code=exited, status=0/SUCCESS)
  Process: 26310 ExecStart=/etc/rc.d/init.d/sshd start (code=exited, status=0/SUCCESS)
 Main PID: 26320 (sshd)
    Tasks: 1
   Memory: 608.0K
   CGroup: /system.slice/sshd.service
```

```
          └─26320 /usr/sbin/sshd

9月 19 20:39:57 sz-lab-centos7-gitlab-nginx-proxy-192.168.178.46 systemd[1]: Starting SYSV:
OpenSSH server daemon...
9月 19 20:39:57 sz-lab-centos7-gitlab-nginx-proxy-192.168.178.46 sshd[26320]: Server listeni
g on 0.0.0.0 port 22.
9月 19 20:39:57 sz-lab-centos7-gitlab-nginx-proxy-192.168.178.46 sshd[26320]: Server listeni
g on :: port 22.
9月 19 20:39:57 sz-lab-centos7-gitlab-nginx-proxy-192.168.178.46 sshd[26310]: Starting sshd:[
 确定  ]
9月 19 20:39:57 sz-lab-centos7-gitlab-nginx-proxy-192.168.178.46 systemd[1]: Started SYSV:
penSSH server daemon.
```

4.4 验证版本

```
# ssh -V
OpenSSH_8.0p1, OpenSSL 1.1.1d  10 Sep 2019
```

4.5 SSH 连接测试

```
# ssh you_username@your_server_ip
```

4.6 重启测试

```
# sync
# reboot now
```