Repositories and Host Allowance/Denial

$ vim /etc/yum.repos.d/rhce.repo

```
name=RHCE_RHEL7
baseurl=http://<baseurl>
enabled=1
gpgcheck=0
```

$ yum repolist

1. Allow SSH for a domain and deny SSH to all the others:

```
vim /etc/hosts.deny` -> `sshd: ALL```

2. Allow SSH for only specific IP and block all the
others:

```vim /etc/hosts.deny -> sshd: ALL EXCEPT
192.168.0.1```

3. Denies all services to all hosts unless permitted
in hosts.allow:

`vim /etc/hosts.allow` -> `ALL: .foobar.edu EXCEPT
terminalserver.foobar.edu`
`vim /etc/hosts.deny` -> `ALL`

4. Access granted by default, redundant file
hosts.allow

`vim /etc/hosts.deny` -> `some.host.name,
```
```

```
.some.domain`
`vim /etc/hosts.deny` -> `ALL EXCEPT in.fingerd:
other.host.name, .other.domain`

5. Rules can be also only in one file, for example:

`vim /etc/hosts.allow` -> `ALL: .friendly.domain:
ALLOW`
                          `ALL: ALL: DENY`
`vim /etc/hosts.allow` -> `ALL: .bad.domain: DENY`
                          `ALL: ALL: ALLOW`

# SERVICES
```systemctl --failed --type=service
systemctl show <unit>
systemctl status <-l> <unit>
systemctl stop|start|restart|reload <unit>
systemctl mask|unmask <unit>
systemctl enable|disable <unit>
systemctl list-dependencies <unit>
systemctl list-units --type=service --all
systemctl list-unit-files --type=service
systemctl get-default
systemctl set-default <graphical|multi-
user|rescue|emergency>
systemctl isolate <graphical|multi-
user|rescue|emergency>```

2               IPV4
nmcli dev status
nmcli con show <name>
nmcli con show --active
ip addr show <eth0>
ip link
nmcli con add con-name <name> type ethernet ifname
<eth0> ip4 xxx.xxx.xx.x/24 gw4 xxx.xxx.xx.x
nmcli con <up|down> <name>
nmcli dev status
nmcli dev dis <eth0>
```
```

```
nmcli con mod <name> +ipv4.dns xxx.xxx.xx.x
        vim /etc/sysconfig/network-script/ifcfg-
<name>
nmcli con reload
nmcli con del <name>
hostname
hostnamectl set-hostname <name>
        vim /etc/hostname
hostnamectl status
ip route
ss -tulpn | grep sshd

3               IPV6
nmcli con add con-name <name> type ethernet ifname
<eth0> ip6 xxxx:xxxx:xxx:x:x:x/64 gw6
xxxx:xxxx:xxx:x:x:x
ip -6 route show
ping6 xxxx:xxxx:xxx:x:x:x
ping6 xxxx:xxxx:xxx:x:x:x<%eth1> for link-local
addresses and multicast groups
tracepath6 xxxx:xxxx:xxx:x:x:x
ss -A inet -n
netstat -46n

4               TEAMING
a/ nmcli con add con-name <team0> type team ifname
<team0> config '{ "runner": { "name": "
<activebackup|broadcast|loadbalance|roundrobin|lacp>"
}}'
b/ nmcli con mod <team0> ipv4.address xxx.xxx.xx.x/24
c/ nmcli con mod <team0> ipv4.method manual
d/ nmcli con add con-name <team0-port1> type team-
slave ifname <eth0> master <team0>
e/ nmcli con add con-name <team0-port2> type team-
slave ifname <eth1> master <team0>
f/ nmcli con up <team0>
nmcli dev dis eth1
teamdctl <team0> state
teamdctl <team0> config dump
```

```
teamnl <team0> ports
teamnl <team0> options
teamnl <team0> getoption activeport
teamnl <team0> setoption activeport <2>

5               BRIDGING
a/ nmcli con add con-name <bridge0> type bridge
ifname <br0>
b/ nmcli con add con-name <bridge0-port1> type
bridge-slave ifname <eth0> master <br0>
c/ nmcli con add con-name <bridge0-port2> type
bridge-slave ifname <eth1> master <br0>
brctl show

6               FIREWALL
a/ systemctl mask <iptables|ip6tables|ebtables>
firewall-cmd --set-default zone=
<dmz|trusted|home|internal|work|public|external|block|
drop>
        trusted=all incoming traffic allowed
        home=reject incoming unless matching
outgoing, accept incoming ssh,mdns,ipp-client,samba-
client,dhcpv6-client
        internal=same as home
        work=reject incoming unless matching
outgoing, accept incoming ssh,ipp-client,dhcpv6-
client
        [DEFAULT]public=reject incoming unless
matching outgoing, accept incoming ssh,dhcpv6-client
        external=reject incoming unless matching
outgoing, accept incoming ssh, masquerading enabled
        dmz=reject incoming unless matching outgoing,
accept incoming ssh
        block=reject incoming unless matching
outgoing
        drop=reject incoming unless matching
outgoing, does not respond at all
firewall-cmd --<get-default-zone|set-default-
zone|get-zones|get-services|get-active-zones|list-
```

```
all>
firewall-cmd --permanent --zone=<name> --add-
source=xxx.xxx.xx.x/24
firewall-cmd --timeout=60 --zone=<name> --add-
service=mysql
firewall-cmd --reload
firewall-cmd --<remove-service=SERVICE|remove-
port=PORT/PROTOCOL>
firewall-cmd --permanent --zone=<name> --add-rich-
rule='rule family=ipv4 source address=xxx.xxx.xx.x/32
reject'
firewall-cmd --permanent --zone=<name> --add-rich-
rule='rule family=ipv4 source address=xxx.xxx.xx.x/24
port=xxxx-xxxx protocol tcp <accept|reject|drop>'
firewall-cmd --permanent --zone=<name> --add-
masquerade
firewall-cmd --permanent --zone=<name> --add-rich-
rule='rule family=ipv4 source address=xxx.xxx.xx.x/24
masquerade'
firewall-cmd --permanent --zone=<name> --add-forward-
port=port=<xxxx>:proto=<tcp>[:toport=<xxxx>:toaddr=
<xxx.xxx.xx.x>]
firewall-cmd --<remove-rich-rule=RULE|query-rich-
rule=RULE|list-rich-rules>
b/ SELinux
semanage port -l
semanage port -<a|d|m> -t http_port_t -p tcp <88>
yum -y install selinux-policy-devel
mandb
man -k _selinux

7               DNS
vim /etc/resolv.conf
host -v -t A example.com
host -v -t AAAA a.root-servers.net
host -v -t A ipa-ca-server0.example.com
host -v -t PTR 172.25.0.10
host -v -t PTR 2001:503:ba3e::2:30
host -v -t <NS|SOA|MX|TXT> example.com
```

```
host -v -t SRV _ldap._tcp.server0.example.com
yum -y install unbound
systemctl start unbound
systemctl enable unbound
vim /etc/unbound.conf
        interface: 0.0.0.0
        access-control: 172.25.0.0/24 allow
        forward-zone:
                name: "."
                forward-addr: 172.25.254.254
        domain-insecure: example.com
unbound-checkconf
systemctl restart unbound
firewall-cmd --permanent --add-service=dns
firewall-cmd --reload
unbound-control dump_cache > dump.out
unbound-control load_cache < dump.out
unbound-control flush_zone <example.com>
unbound-control flush <www.example.com>
getent hosts <example.com>
gethostip <example.com>
dig A <example.com>
dig @<dns.example.com> A <www.example.com>
dig +tcp A <example.com>
dig +dnssec DNSKEY <example.com>

9               POSTFIX AS NULL CLIENT
vim /etc/postfix/main.cf
        inet_interfaces = loopback-only (which NIC
Postfix listens on for incoming/outgoing messages)
        myorigin = clientX.example.com (e-mails will
appear to come from this domain)
        relayhost = [server.example.com] (forward all
messages to this mail server)
        mydestination = (which domains the mail
server is an end point for)
        local_transport = error: local delivery
disabled
        mynetworks = 127.0.0.0/8, [::1]/128 (allow
```

```
relay from these networks)
systemctl restart postfix
postconf <-e> 'VAR = VAL'
postqueue -<p|f>
mail -s "serverX null client"
student@desktopX.example.com null client test

10              iSCSI
a/ Targets - server creating
        yum -y install targetcli
                LVM: fdisk /dev/vdb => type 8e;
pvcreate /dev/vdb1; vgcreate iSCSI_vg /dev/vdb1;
lvcreate -n disk1_lv -L 100m iSCSI_vg
        targetcli
        cd /backstores
        block/ create <block1> /dev/iSCSI_vg/disk1_lv
        block/ create <block2> /dev/vdb2
        block/ create <file1> /root/disk1_file 100M
        cd /iscsi
        create iqn.2015-10.com.example:server
        cd iqn.2015-10.com.example:server/tpg1
        acls/ create iqn.2015-10.com.example:
<client.example.com>
        luns/ create /backstores/block/block1
        luns/ create /backstores/block/block2
        luns/ create /backstores/fileio/file1
        portals/ create 172.25.0.11
        exit
        firewall-cmd --permanent --add-port=3260/tcp
        firewall-cmd --reload
        systemctl enable target
b/ Targets - client accessing
        yum -y install iscsi-initiator-utils
        vim /etc/iscsi/initiatorname.iscsi
(InitiatorName=client.example.com)
        systemctl restart iscsi
        iscsiadm -m discovery -t sendtargets -p
172.25.0.11:3260
        iscsiadm -m node -T iqn.2015-
```

```
10.com.example:server -p 172.25.0.11 -l
        lsblk
        iscsiadm -m session -P 3
        cd /var/lib/iscsi/nodes; ls -lR
c/ Targets - client disconnecting
        iscsiadm -m node -T iqn.2015-
10.com.example:server -p 172.25.0.11 -u
        iscsiadm -m node -T iqn.2015-
10.com.example:server -p 172.25.0.11 -o delete
        lsblk
        systemctl restart iscsi

11              NFS
a/ Server - insecure
yum -y install nfs-utils
systemctl start nfs-server
systemctl enable nfs-server
mkdir /myshare
chown nfsnobody /myshare
vim /etc/exports
        /myshare client.example.com(rw)
        /myshare *.example.com
        /myshare server[0-20].example.com
        /myshare 172.25.0.0/16
        /myshare 172.25.11.10(rw,no_root_squash)
*.example.com(ro)
exportfs -r
firewall-cmd --permanent --add-services=nfs
firewall-cmd --reload
showmount -e
b/ Client - insecure
mount server.example.com:/myshare /mnt/nfs
c/ Server - secure
wget -O /etc/krb5.keytab http://xxxxxxxxxx
vim /etc/sysconfig/nfs (RPCNFSDARGS="-V 4.2")
systemctl restart nfs-server
systemctl restart nfs-secure-server
systemctl enable nfs-secure-server
vim /etc/exports
```

```
        /mysecureshare
client.example.com(sec=krb5p,rw)
                                sec=none: uses
nfsnobody
                                sec=sys: using linux
file permissions
                                sec=krb5: kerberos
and then linux file permissions apply
                                sec=krb5i: adds
checksums to the data transfers
                                sec=krb5p: adds
encryption
exportfs -r
firewall-cmd --permanent --add-services=nfs
firewall-cmd --reload
d/ Client - secure
wget -O /etc/krb5.keytab http://xxxxxxxxxx
yum -y install nfs-utils
systemctl start nfs-secure
systemctl enable nfs-secure
mount -o sec=krb5p,v4.2
server.example.com:/mysecureshare /mnt/nfs
        vim /etc/fstab
        serverx:/securenfs /mnt/secureshare nfs
defaults,v4.2,sec=krb5p 0 0
        mount -a
e/ SELinux
context default: nfs_t or public_content_t,
for writable, change context: public_content_rw_t +
nfsd_anon_write boolean
boolean default: nfs_export_all_ro, nfs_export_all_rw

12              SMB
a/ Server
yum -y install samba samba-client
vim /etc/samba/smb.conf
        [global]
                workgroup=WORKGROUP
                security=user (requires samba
```

```
password)
                hosts allow=172.25. .example.com
        [myshare]
                path=/sharedpath
                writable=<yes|no>
                        write list=<user>
                valid users=<blank>|
<user>|@management|+users
        [homes]
                read only=no
        [printers]
testparm
useradd -s /sbin/nologin -G <group> <user>
smbpasswd -<a|x> <user>
systemctl reload smb nmb
systemctl enable smb nmb
firewall-cmd --permanent --add-services=samba
firewall-cmd --reload
chmod 2775 /sharedpath
b/ Client - singleuser
yum -y install cifs-utils
mount -o <username=
<user>|credentials=credentials.txt>
//server.example.com/<sharename> /mnt/smb
smbclient -L server.example.com
c/ Client - multiuser
yum -y install cifs-utils
cifscreds <add|update|clear|clearall> -u <user>
<server.example.com>
mount -o multiuser,sec=ntlmssp,username=
<user>,credentials=<multiuser_file.txt>
//server.example.com/<sharename> /mnt/multiuser
        vim /root/multiuser_file.txt
                username=<user1>
                password=<password1>
smbclient -L server.example.com
d/ SELinux
context: samba_share_t, public_content_t,
public_content_rw_t + smbd_content_rw_t boolean
```

```
boolean for homes: samba_enable_home_dirs on the
server, use_samba_home_dirs on the client
        e.g. setsebool -P samba_enable_home_dirs=on

13              MARIADB
yum -y groupinstall mariadb mariadb-client
systemctl start mariadb
systemctl enable mariadb
mysql_secure_installation
vim /etc/my.cnf
        [mysqld]
                bind-address <::|0.0.0.0|blank>
                skip-networking <1=not even localhost
can connect,only socket|0>
                port
firewall-cmd --permanent --add-rule=mysql
firewall-cmd --reload
mysql -u <root> -h <hostname> -p
create database <name>;
use <name>;
a/ Managing users and access rights
create user <user>@'<%|192.168.1.%|localhost>'
identified by '<password>';
        mysql -u <user> -h <hostname> -p
grant select on <database.table> to
<user>@<hostname>;
grant select on <database.*> to <user>@<hostname>;
grant select on <*.*> to <user>@<hostname>;
grant <create,alter,drop> on <database.*> to
<user>@<hostname>;
grant all privileges on <*.*> to <user>@<hostname>;
revoke <select,update,delete,insert> on
<database.table> from <user>@<hostname>;
flush privileges;
show grants for <user>@<hostname>;
drop user <user>@<hostname>;
b/ Backup - logical
                mysqldump -u root -p <dbname> >
/tmp/dbname.dump
```

```
                mysqldump -u root -p --<all-
databases|add-drop-tables|no-data|lock-all-
tables|add-drop-databases> > /tmp/all.dump
c/ Backup - physical
                mysqladmin variables | grep datadir
                        cat /etc/my.cnf | grep -i
datadir
                df /var/lib/mysql (/dev/mapper/vg0-
mariadb shows 'vg0' is volume group and 'mariadb' is
logical volume name)
                vgdisplay vg0 | grep free
                tty0: mysql -u root -p
                tty0: flush tables with read lock;
                tty1: lvcreate -L20G -s -n mariadb-
backup /dev/vg0/mariadb
                tty0: unlock tables;
                mkdir /mnt_snapshot
                mount /dev/vg0/mariadb-backup
/mnt_snapshot
                tar cvzf mariadb_backup.tar.gz
/mnt_snapshot/var/lib/mysql
                umount /mnt_snapshot
                lvremove /dev/vg0/mariadb-backup
d/ Restore - logical
                mysql -u root -p <dbname> <
/backup/dbname.dump
e/ Restore - physical
                systemctl stop mariadb
                mysqladmin variables | grep datadir
                rm -rf /var/lib/mysql/*
                tar xvzf mariadb_backup.tar.gz
/var/lib/mysql
f/ Queries
                show databases;
                select * from product;
                show tables;
                describe <table>;
                insert into <product> (name,price)
values ('oracle',1000);
```

```
                delete from <product> where <id=1>;
                delete from <category> where name
like 'Memory';
                update <product> set <price=999>
where <id=1>;
                select name,price,stock from product;
                select * from product where price >
90;
                exit;

14              APACHE
yum -y install httpd httpd-manual
vim /etc/httpd/conf/httpd.conf
        ServerRoot "/etc/httpd" (where are config
files))
        Listen 80 (can be Listen 1.2.3.4:80)
        Include conf.modules.d/*.conf (if multiple
are present, they will be alphabetically included)
        User apache
        Group apache
        ServerAdmin root@localhost
        <Directory /> (directives specific to the dir
and all descendent dirs)
                AllowOverride none (.htaccess will
not be used)
                Require all denied (refuse to serve
conten from dir)
        </Directory>

        DocumentRoot "/var/www/html" (where apache
looks for files)
        <Directory  "/var/www/">
                AllowOverride none
                Require all granted
        </Directory>
        <Directory "/var/www/html">
                Options Indexes FollowSymLinks
                AllowOverride none
                Require all granted
```

```
        </Directory>

        <IfModule dir_module> (if this module is
loaded, what happens)
                DirectoryIndex index.html (this file
will be used when the direcory is requested)
        </IfModule>

        <Files ".ht*"> (same as directory, but for
file wildcards)
                Require all denied
        </Files>

        ErrorLog "logs/error_log" (it will go to
/etc/httpd/logs/error_log, which is symlink to
/var/log/httpd/error_log)
        LogLevel warn
        CustomLog "logs/access_log" combined
        AddDefaultCharset UTF-8 (can be disabled by
AddDefaultCharset Off)
        IncludeOptional conf.d/*.conf (same as
regular include)
systemctl enable httpd
systemctl start httpd
firewall-cmd --permanent --add-service=http --add-
service=https
firewall-cmd --reload
semanage port -l | grep '^http_'
a/ New DocumentRoot for group 'webmasters'
                mkdir -p -m 2775 /new/web
                chgrp webmasters /new/web
                chmod 2775 /new/web
                setfacl -R -m g:webmasters:rwX
/new/web
                setfacl -R -m d:g:webmasters:rwX
/new/web
                semanage fcontext -a -t
httpd_sys_content_t "/new/web(/.*)?"
                restorecon -Rv /new/web
```

```
                systemctl reload httpd
b/ Virtual hosts
vim /etc/httpd/conf.d/00-site1.conf
        <Directory /srv/site1/www> (this block
provides access to document root further down)
                Require all granted
                AllowOverride none
        </Directory>

        <VirtualHost 192.168.0.1:80> (this block must
be considered for all connections on 192.168.0.1:80,
can be _default_:80 or *:80)
                DocumentRoot /srv/site1/www (only
applies for within this virtual host)
                ServerName site1.example.com (name-
based virtual hosting, if multiple virtual hosts are
defined, the one where hostname matches this will be
used)
                (ServerAlias - if the virtual host
needs to be used for more than one domain name)
                ServerAdmin root@site1.example.com
                ErrorLog "logs/site1_error_log"
                CustomLog "logs/site1_access_log"
combined
        </VirtualHost>
        semanage fcontext -a -t httpd_sys_content_t
"/srv/site1/www(/.*)?"
        restorecon -Rv /srv/site1/www
c/ SSL/TLS
yum -y install crypto-utils mod_ssl
genkey <www.example.com>
vim /etc/httpd/conf.d/ssl.conf
        Listen 443 https
        SSLPassPhraseDialog exec:/usr/libexec/httpd-
ssl-pass-dialog (if the private key uses passphrase)
        <VirtualHost _default_:443>
                SSLEngine on
                SSLProtocol all -SSLv2
                SSLCipherSuite
```

```
HIGH:MEDIUM:!aNULL:!MD5
                (SSLHonorCipherOrder On)
                SSLCertificateFile
/etc/pki/tls/certs/www.example.com.crt (public key)
                SSLCertificateKeyFile
/etc/pki/tls/certs/www.example.com.key (private key)
                (SSLCertificateChainFile
/etc/pki/tls/certs/example-ca.crt) (copy of all CA
certificates)
        </VirtualHost>
semanage fcontext -a -t cert_t /etc/pki/tls/certs/*.*
chmod 0600 /etc/pki/tls/certs/*.key
chmod 0644 /etc/pki/tls/certs/*.crt
d/ HSTS - strict transport security
        <VirtualHost *:80>
        Header always set Strict-Transport-Security
"max_age=15768000"
        RewriteEngine on
        RewriteRule ^(/.*)$ https://%{HTTP_POST}$1
[redirect=301]
        <VirtualHost>
e/ Dynamic content
                I. CGI
                        vim
/etc/httpd/conf/httpd.conf
                        ScriptAlias /cgi-bin/
"/var/www/cgi-bin/"
                SELinux fcontext:
httpd_sys_script_exec_t
                II. PHP
                        yum -y install mod_php php
php-mysql
                        <FilesMatch \.php$>
                                SetHandler
application/x-httpd-php
                        </FilesMatch>
                        DirectoryIndex index.php
                III. Python
                        yum -y install mod_wsgi
```

```
                        vim
/etc/httpd/conf/httpd.conf
                        WSGIScriptAlias /myapp
"/srv/my.py"
                        SELinux fcontext:
httpd_sys_content_t
SEBooleans:
        I. if the database is on remote host:
httpd_can_network_connect_db on
        II. if the known port number is used for db
connection: httpd_can_network_connect on

15              SHELL ENVIRONMENT
a/ Global
        /etc/profile
        /etc/profile.d/*.sh
b/ User
        ~/.bash_profile, .bash_login, .profile
        ~/.bashrc
        /etc/bashrc
Profiles are for setting and exporting of environment
variables, as well as running commands that should
only be run upon login.
RCs are for running commands, setting aliases,
defining functions and other settings that cannot be
exported to sub-shells.
Usually, profiles are only executed in a login shell,
whereas RCs are executed every time a shell is
created, login or non-login.
export MYVAR
alias
unalias
function () {...}
set
unset
```