Quick Reference Card 6 Course UNIX/Linux CLE9 3058 add-on, Services and Security

Your? host logo

PAM
#/etc/pam.d/file with 4 *module-types*:
auth #Ask for password and grant group mem
account #Restrict or permit access (time, location,)
session #Pre, post service logging, data exchange,
password #Update authentication token
4 *control-flags*:
required #All keywords are required
requisite #Required till failure
sufficient #Sufficient
optional #Not critical to success or failure
remember=08 #/etc/security/pam_pwcheck.conf
#/etc/sysconfig/security #YaST, Sec. and Users, Sec. Settings
#/opt/kde3/share/config/kdm/kdmc #Who can shutdown
#/etc/login.defs
#/etc/permissions.*easy* #.secure, .paranoid, .local
#/etc/permissions

Host Security #Test security vulnerabilities with perm.:
netstat #e.g. -antp or -anup
ethereal #e.g. filter dst port 80
nmap -sU -sT host #Udp and Tcp scan, www.insecure.org
nessus-mkcert #www.nessus.org (*saint* alt.), 4 steps
nessus-adduser
/etc/init.d/nessusd start
nessus

Logging
sysstat #sysstat package /var/log/sa.*date*:
#sadc, sar, isag, mpstat -P 0, iostat
last, ac #Read /var/log/wtmp
accton #acct package
lastcomm
sa -um #summarize accounting
logcheck
logsurfer
seccheck

Cryptography #Base directories in CA-dir (man x509):
mkdir certs, crl, newcerts, private, crl;chmod 700 private;touch index.txt; \
  echo 01 > serial #www.openssl.org
vi /etc/ssl/openssl.cnf #Defaults. Create root-CA certificate, e.g.:
openssl req -newkey rsa:2048 -x509 -days 3650 -keyout \
  private/cakey.pem -out cacert.pem
openssl x509 -in cacert.pem -text#View cert. Create key pair (man req):
openssl req -new -keyout private/srvprvkey.pem -out certs/srvreq.pem \
  -days 730 #Sign certificate (man ca):
openssl ca -policy policy_anything -notext -out certs/srvcert.pem \
  -infiles certs/srvreq.pem #Revoke certificate and create crl:
openssl ca -revoke certs/srvcert.pem
openssl ca -gencrl -out crl/srvcrl.pem
gpg --genkey #GNU Privacy Guard (GPG versus PGP)
gpg -a --export "realname">file #Export public key. Copy to partner
gpg --import partnerfile #Import partnerkey
gpg -ea file #Encrypt file
gpg file #Decrypt file, or: gpg -o - file
gpg --clearsign file #Sign file
gpg file #Verify signature

Network Security #See also '*TCP Wrapper*' on QuickRef 3
#Tunnel anything using port 22, (See also 'Application-level Gateways'), e.g:
ssh -L 8080:blockedsite.com:80 athome.net
http://localhost:8080 #Bring up blockedsite.com.
#Package 'stunnel':
openssl rsa < private/srvprvkey.pem > private/srvprvkeyunenc.pem
cp cacert.pem /tmp #Root CA certificate
cat certs/srvcert.pem private/srvprvkeyunenc.pem >> \
  /etc/stunnel/stunnel.pem #or aft unencrypted export via YaST:
cp srvcert.pem /etc/stunnel/stunnel.pem ; chmod 600 stunnel.pem
vi /etc/stunnel/stunnel.conf #Example 'qpopper' package tunneling
  [pop3s] #Comment out: chroot, setuid, setgid
  accept = 995 #Port 995
  exec = /usr/sbin/popper #Restart stunnel and,
  execargs = popper -s #import /tmp/cacert.pem in application

Firewall #See also QuickRef 3. Enable routing:
echo 1 > /proc/sys/net/ipv4/ip_forward
vi /etc/sysconfig/sysctl #Survive a reboot
  IP_FORWARD="yes"
cp /etc/init.d/skeleton /etc/init.d/fw-script ; chmod 744 /etc/init.d/fw-script
vi /etc/init.d/fw-script #Edit to start 'firewall.sh'

NAT #Source NAT, Masquerading, Dest. NAT
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to-source \
  192.168.199.200
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
iptab.es -t nat -A PREROUTING -p tcp -i eth1 --dport 80 -j DNAT \
  --to-destination 172.17.0.112
iptables -t nat -A PREROUTING -p tcp -i eth0 --dport 80 \
  -j REDIRECT --to-ports 3128

Application-level Gateways #Squid(http), Dante(socks), rinetd(redir.)
vi /etc/squid/squid.conf #Proxy server, e.g.:
  http_port 3128 #Set port. Digest auth. (7 lines, see .conf):
  auth_param digest program /usr/sbin/digest_pw_auth \
    /etc/squid/passwords
  auth_param digest children 5
  auth_param digest realm Squid proxy-caching web server
  auth_param digest nonce_garbage_interval 5 minutes
  auth_param digest nonce_max_duration 30 minutes
  auth_param digest nonce_max_count 50
  acl allowed_users proxy_auth REQUIRED #See also '8th line'
  acl all src 0.0.0.0/0.0.0.0
  acl hostsgrp src 10.0.0.0/24
  acl hostsgrp1 src 10.10.10.0/24
  acl lunchtime time MTWHF 12:30-13:30 #A=Saturday, S=Sunday
  acl no-internet src 10.0.2.0/24
  #See http://squid.visolve.com/squid/squid24s1/access_controls.htm
  #One aclname can be a combination of more acl lines (logical ORs):
  acl blocked url_regex -i shole.com #
  acl blocked url_regex -i ^http://fann #^ is 'starts with'
  acl blocked url_regex -i "/etc/squid/blacklist" #
  acl binary url_regex -i zip$ #$ is 'ends with'
  acl binary url_regex -i exe$
  redirect_program /usr/sbin/squidGuard #Package squidGuard
  acl whitelist url_regex -i teletekst #Quotes not required
  acl whitelist url_regex -i "/etc/squid/whitelist" #Ignore -i case sensitiv.
  #First http_access deny/allow match is implemented, rest is skipped!
  #Http_access statements combine acl's on one line using logical AND.
  acl ssl-ports port 443 563 #SSL Tunneling (3 lines)
  acl connect method CONNECT
  http_access deny connect !ssl-ports #! is logical NOT
  http_access deny blocked #or
  http_access deny blocked !lunchtime
  http_access deny no-internet
  http_access deny binary
  http_access allow allowed_users hostsgrp #'8th line' of Digest auth.
  http_access allow hostsgrp1 lunchtime #Beware of line order!
  http_access allow hostsgrp
  http_access allow whitelist
  http_access deny all #Required to prevent 'reverse default'
vi /etc/squid/blacklist #Or download www.squidblock.com
  hotmail.com #Or www.squidguard.org/blacklist
  mail.com
vi /etc/squid/whitelist
  suse
  wikipedia
  hp.com
vi /etc/squid/passwords
  kbailey:password
  mperez:password
chown squid /etc/squid/passwords
chmod 600 /etc/squid/passwords
#Package 'transconnect' enables Squid tunneling:
#(Have sshd athome.net listen on port 443)
cp /usr/share/doc/packages/transconnect/tconn.conf ~/.tconn/tconn.conf
vi ~/.tconn/tconn.conf
  LD_PRELOAD=/usr/lib/tconn.so ssh athome.net
vi /etc/squidguard #www.squidguard.org/config
  logdir /var/log/squidGuard
  dnhome /var/lib/squidGuard/db
  dest blacklist {
  domainlist blacklist/domains
  urllist blacklist/urls
  }
  acl {
  default {
  pass !blacklist all
  redirect 302:http://www.novell.com/linux
  }
  }
echo "whitehouse.com" >> /var/lib/squidGuard/db/blacklist/domains
#Transparant proxy (no https, ftp and gopher):
iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT \
  --to-port 3128
calamaris -d 10 /var/log/squid/access.log #Extra package

VPN #Three example tools (as root):
fdisk -l #List all

Intrusion Detection #Three example tools (as root):
fdisk -l #List all

AutoYaST #Share with: smb, nfs, ftp, or http
#Install. src created with: YaST, Misc, Installation Server and Autoinstallation
:autoyast=http://ip/suse/sles9/yast/file.xml