1 CheatSheet: linux-capabilities LINUX

- PDF Link: cheatsheet-linux-capabilities-A4.pdf, Category: linux
- Blog URL: `https://cheatsheet.dennyzhang.com/cheatsheet-linux-capabilities-A4`
- Related posts: CheatSheet: Shell, #denny-cheatsheets

File me Issues or star this repo.

1.1 Linux Capabilities - Frequent

Starting with kernel 2.2, Linux divides the privileges traditionally associated with superuser into distinct units, known as **linux capabilities**.

Name	Comment
`CAP_CHOWN`	Make arbitrary changes to file UIDs and GIDs
`CAP_NET_RAW`	use RAW and PACKET sockets; bind to any address for transparent proxying
`CAP_SYS_CHROOT`	Use chroot
`CAP_SETUID`	Make arbitrary manipulations of process UIDs
`CAP_SETGID`	Make arbitrary manipulations of process GIDs and supplementary GID list
`CAP_DAC_OVERRIDE`	Bypass file read, write, and execute permission checks
`CAP_MKNOD`	Create special files using mknod
`CAP_NET_BIND_SERVICE`	Bind a socket to Internet domain privileged ports (port numbers less than 1024).
`CAP_NET_RAW`	Use RAW and PACKET sockets; bind to any address for transparent proxying.
`CAP_SETFCAP`	Set file capabilities.
Reference	link: ubuntu linux capabilities help usage

1.2 Linux Capabilities - UID/GID

Name	Comment
`CAP_SETUID`	Make arbitrary manipulations of process UIDs
`CAP_SETGID`	Make arbitrary manipulations of process GIDs and supplementary GID list

1.3 Linux Capabilities - File

Name	Comment
`CAP_CHOWN`	Make arbitrary changes to file UIDs and GIDs
`CAP_MKNOD`	Create special files using mknod
`CAP_DAC_OVERRIDE`	Bypass file read, write, and execute permission checks
`Cap_dac_read_search`	Bypass file read permission checks and directory read and execute permission checks
`CAP_LEASE`	Establish leases on arbitrary files
`CAP_SETFCAP`	Set file capabilities.
`CAP_KILL`	Bypass permission checks for sending signals

1.4 Linux Capabilities - Network

Name	Comment
`CAP_MAC_OVERRIDE`	Allow MAC configuration or state changes
`CAP_NET_BIND_SERVICE`	Bind a socket to Internet domain privileged ports (port numbers less than 1024).
`CAP_NET_BROADCAST`	(Unused) Make socket broadcasts, and listen to multicasts.
`CAP_NET_RAW`	Use RAW and PACKET sockets; bind to any address for transparent proxying.

1.5 Linux Capabilities - Process

Name	Comment
`CAP_KILL`	Bypass permission checks for sending signals
`CAP_SYS_NICE`	
`CAP_SYS_CHROOT`	Use chroot
`CAP_SYS_BOOT`	Use reboot and $\text{kexec}_{\text{load}}$
`CAP_WAKE_ALARM`	Trigger something that will wake up the system (set $\text{CLOCK}_{\text{REALTIMEALARM}}$ and $\text{CLOCK}_{\text{BOOTTIMEALARM}}$

1.6 Linux Capabilities - Adhoc

Name	Comment
CAP_AUDIT_CONTROL	Enable and disable kernel auditing; change auditing filter rules; retrieve auditing status and rules.
CAP_AUDIT_WRITE	Write records to kernel auditing log.
CAP_BLOCK_SUSPEND	Employ features that can block system suspend
CAP_FOWNER	
CAP_FSETID	
CAP_IPC_LOCK	Lock memory
CAP_IPC_OWNER	Bypass permission checks for operations on System V IPC objects.
CAP_LINUX_IMMUTABLE	Set the $FS_{APPENDFL}$ and $FS_{IMMUTABLEFL}$ i-node flags (see chattr(1))
CAP_MAC_ADMIN	Override Mandatory Access Control (MAC)
CAP_NET_ADMIN	
CAP_SETPCAP	
CAP_SYS_ADMIN	
CAP_SYS_MODULE	Load and unload kernel modules
CAP_SYS_PACCT	Use acct
CAP_SYS_PTRACE	Trace arbitrary processes using ptrace; apply $get_{robustlist}$ to arbitrary processes; inspect processes using
CAP_SYS_RAWIO	
CAP_SYS_RESOURCE	
CAP_SYS_TIME	Set system clock (settimeofday, stime, adjtimex); set real-time (hardware) clock.
CAP_SYS_TTY_CONFIG	Use vhangup; employ various privileged ioctl operations on virtual terminals.
CAP_SYSLOG	

1.7 More Resources

http://manpages.ubuntu.com/manpages/trusty/man7/capabilities.7.html