CCDC DOC

LATECH CSDC

Breaking in - Windows

- Sticky Keys
  - Shift 5 times: an administrative command prompt should appear
  - Enter the command and reset the password
    - “net user Administrator *”
- Logged in not admin
  - Rest computer
  - Pressing F8 during boot and selecting command prompt
  - Enter the commands
    - **copy c:\windows\system32\cmd.exe**
    - **copy /y c:\windows\system32\cmd.exe c:\windows\system32\sethc.exe**
  - Reboot and follow the sticky key instructions
- Change password
  - **net user <username> ***
    - Requires admin privileges

Breaking in - Linux

- Single User Mode
  - BSD
    - TODO
    - Might need to remount root partition with command
      - **mount -a o rw**
  - Debian
    - Press shift to enter GRUB menu
    - Find the line that begins with “linux#” (either 16 or 32)
    - Replace **ro** with **rw**
    - Finally add **init=/bin/bash** to get a shell on boot
  - Redhat
    - Instead of **init=/bin/bash** use **init=/sysroot/bin/sh**
- Change Password
  - **passwd** username

Breaking in - Mac

- Single User Mode
    - Hold **command-S** on startup
    - Run
        - **Mount -uw /**
    - For 10.7 and later
        - **launchctl load /System/Library/LaunchDaemons/com.apple.opendirectoryd.plist**
    - For 10.6 and before
        - **launchctl load /System/Library/LaunchDaemons/com.apple.DirectoryServices.plist**
- First time setup
    - Hold **command-S** on startup
    - Run
        - **Mount -uw /**
    - Run
        - **Vm /var/db/.AppleSetupDone**
    - Reboot and create a new administrator account

Breaking in - Palo Alto

- Always
  - Reboot the router
  - Press **m** during boot to load boot menu (called **maint**)
  - Factory reset router
- Reset IP command
  - **configure**
  - **set deviceconfig system ip-address <IP> netmask <netmask> default-gateway <gateway-IP> dns-setting servers primary <DNS IP>**
  - **commit**

SERVICE	HARD	IP ADDR	VULNS

SERVICE LOOKUP

LINUX CHECKLIST

CHECK	TODO	COMMENT
	Update sudo rules: **/etc/sudoers**	
	Update SSH config: **/etc/ssh/ssh_config**	
	Use **chattr +i** to lock sudo, ssh	
	Run **chmod -Rv go-rwx /root**	
	Run **chmod -Rv go-w /home/***	
	Add **sysadmin** user	
	Run **systemctl -l** and look for service running on host	
	Check to see if it is running as root!	
	Google service to get *configs*, *exploits,* and *default creds*	
	Backup **configs**, replacing *default creds*	
	Check for quick exploit fixes	
	Backup everything with **tar zcv - <dir> | gpg -c –cipher-algo aes256 -o <name>.tgz.gpg**	
	Flex bash history by appending this to **.bashrc** **shopt -s histappend** **PROMPT_COMMAND="history -a;$PROMPT_COMMAND"**	
	Change database default passwords	
	Setup *fail2ban*, *mod_security* (OPTIONAL)	

	1	2	3	4	5
A					
B					
C					
D					
E					
F					
G					
H					
I					
J					

PASSWORD 0

	6	7	8	9	10
A					
B					
C					
D					
E					
F					
G					
H					
I					
J					

PASSWORD 0

HOW TO USE TCPDUMP

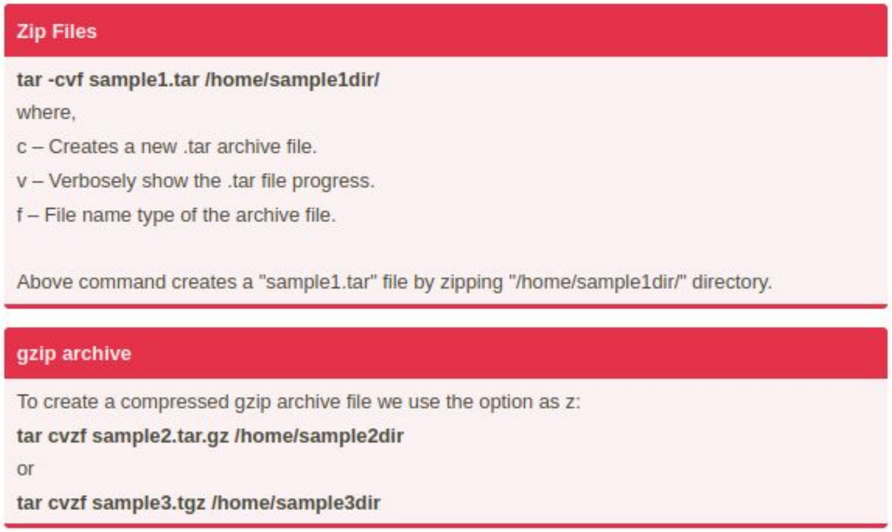
TCPDUMP

```
sudo tcpdump --interface any

# listens on all interfaces. Have fun
reading the output
```

- Dumps *all* network traffic
- Can be used to see how the scoring engine works
- And then maximize security by complying with scoring engine

HOW TO USE TAR

TAR / GUNZIP



Zip Files

tar -cvf sample1.tar /home/sample1dir/

where,

c – Creates a new .tar archive file.

v – Verbosely show the .tar file progress.

f – File name type of the archive file.

Above command creates a "sample1.tar" file by zipping "/home/sample1dir/" directory.

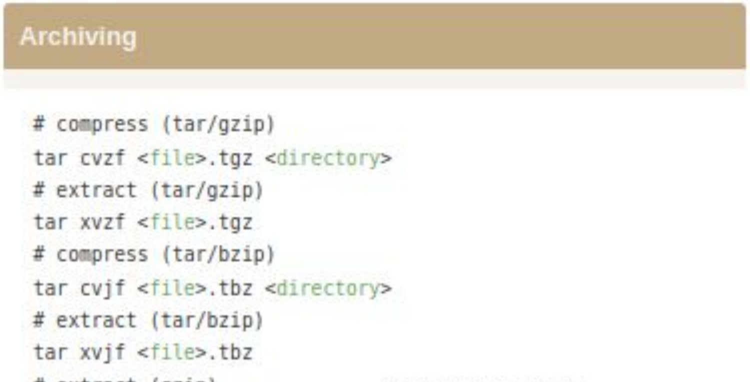
gzip archive

To create a compressed gzip archive file we use the option as z:

tar cvzf sample2.tar.gz /home/sample2dir

or

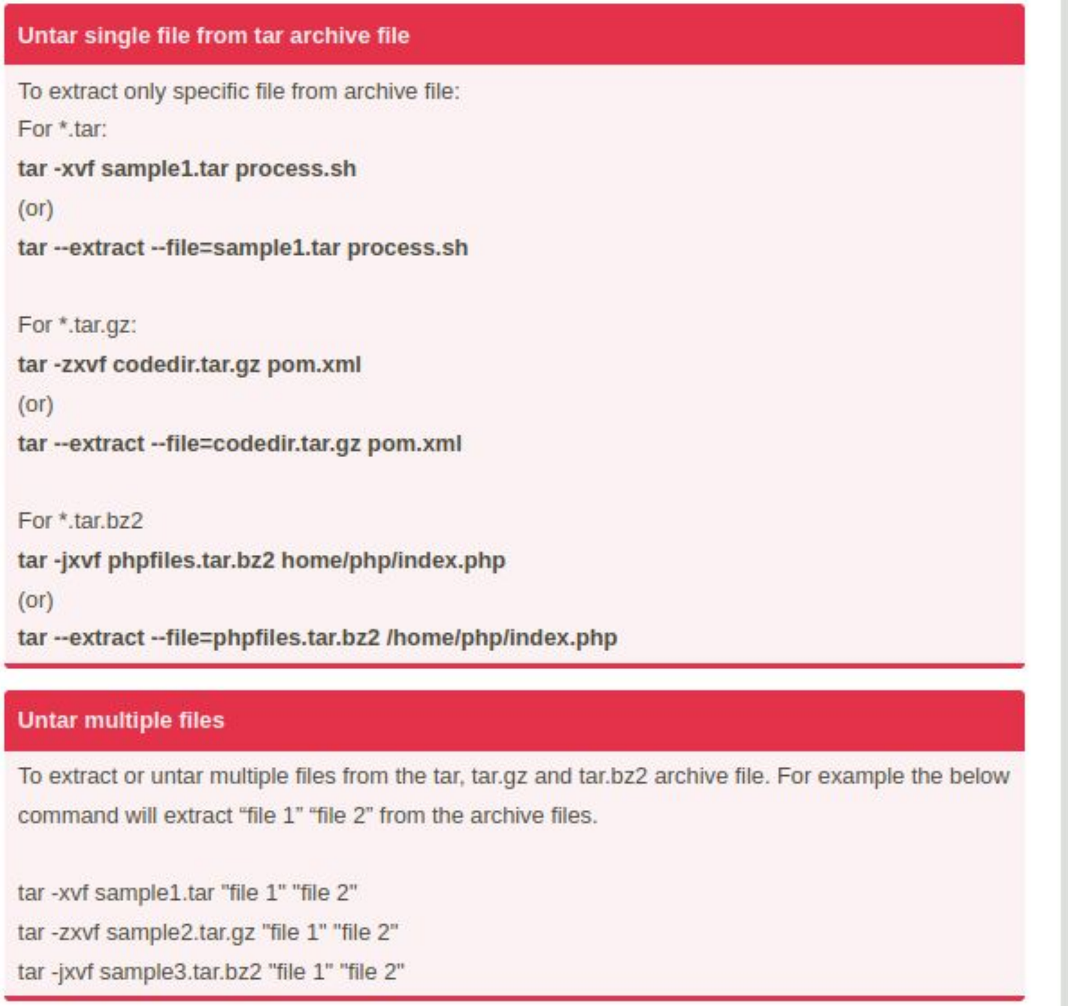
tar cvzf sample3.tgz /home/sample3dir



Archiving

```
# compress (tar/gzip)
tar cvzf <file>.tgz <directory>
# extract (tar/gzip)
tar xvzf <file>.tgz
# compress (tar/bzip)
tar cvjf <file>.tbz <directory>
# extract (tar/bzip)
tar xvjf <file>.tbz
# extract (gzip)
gunzip <file>.gzip
```

GYAHHHh.tar.gz
-XtractZeFile!!! GYAHHHh.tar.gz
-VerboseMeNow!!!
-CompressZeFilez!!! GYAHHHh.tar.gz [files]
.tar.bz2? -XtractFilezJustDoItNOW!!! GYAHHHh.tar.bz2



Untar single file from tar archive file

To extract only specific file from archive file:

For *.tar:

tar -xvf sample1.tar process.sh

(or)

tar --extract --file=sample1.tar process.sh

For *.tar.gz:

tar -zxvf codedir.tar.gz pom.xml

(or)

tar --extract --file=codedir.tar.gz pom.xml

For *.tar.bz2

tar -jxvf phpfiles.tar.bz2 home/php/index.php

(or)

tar --extract --file=phpfiles.tar.bz2 /home/php/index.php

Untar multiple files

To extract or untar multiple files from the tar, tar.gz and tar.bz2 archive file. For example the below command will extract “file 1” “file 2” from the archive files.

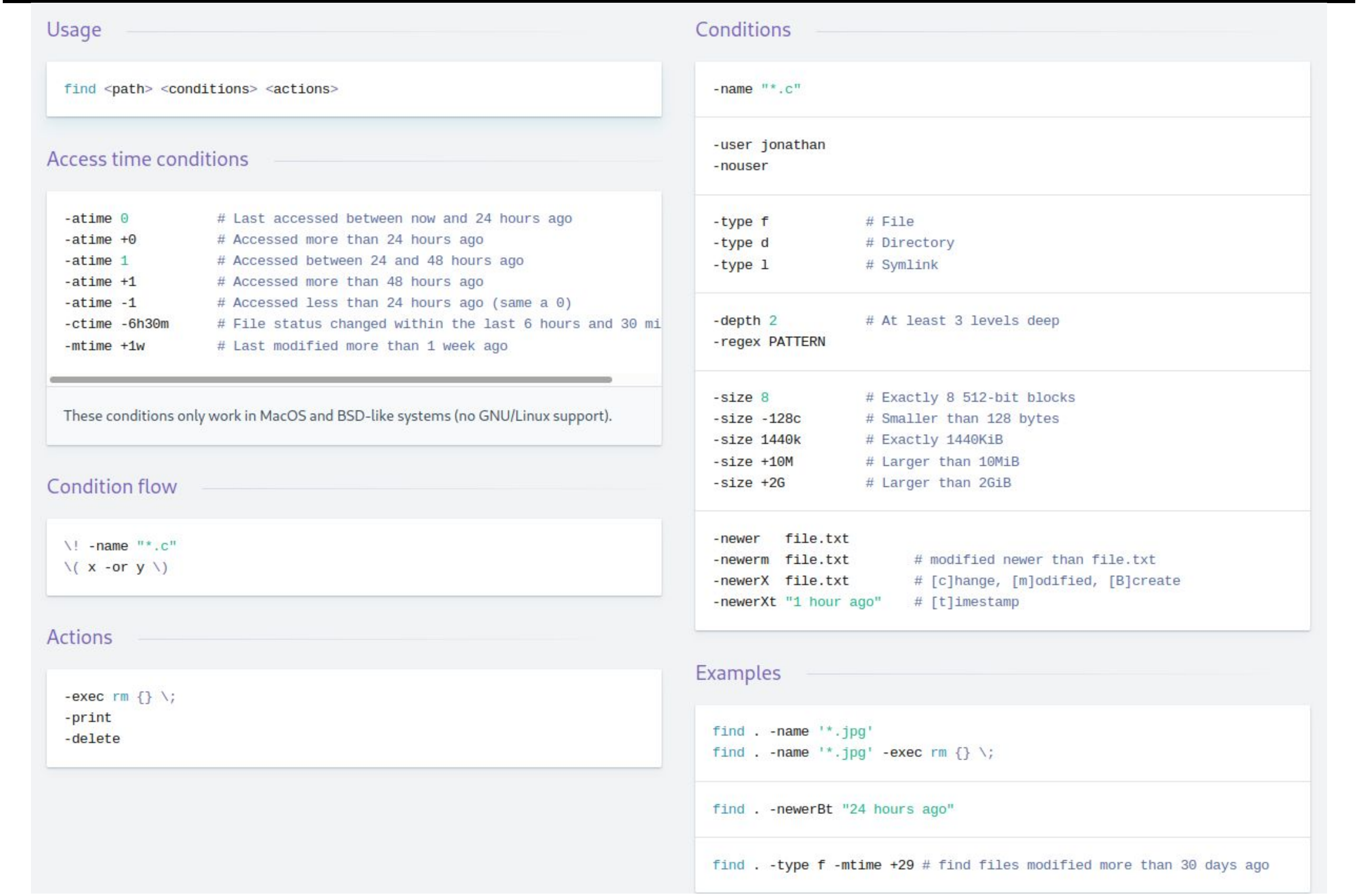
tar -xvf sample1.tar "file 1" "file 2"

tar -zxvf sample2.tar.gz "file 1" "file 2"

tar -jxvf sample3.tar.bz2 "file 1" "file 2"

HOW TO USE FIND

FIND



Usage

```
find <path> <conditions> <actions>
```

Access time conditions

```
-atime 0           # Last accessed between now and 24 hours ago
-atime +0          # Accessed more than 24 hours ago
-atime 1           # Accessed between 24 and 48 hours ago
-atime +1          # Accessed more than 48 hours ago
-atime -1          # Accessed less than 24 hours ago (same a 0)
-ctime -6h30m      # File status changed within the last 6 hours and 30 mi
-mtime +1w         # Last modified more than 1 week ago
```

These conditions only work in MacOS and BSD-like systems (no GNU/Linux support).

Condition flow

```
\! -name "*.c"
\( x -or y \)
```

Actions

```
-exec rm {} \;
-print
-delete
```

Conditions

```
-name "*.c"
```

```
-user jonathan
-nouser
```

```
-type f            # File
-type d            # Directory
-type l            # Symlink
```

```
-depth 2           # At least 3 levels deep
-regex PATTERN
```

```
-size 8            # Exactly 8 512-bit blocks
-size -128c        # Smaller than 128 bytes
-size 1440k        # Exactly 1440KiB
-size +10M         # Larger than 10MiB
-size +2G          # Larger than 2GiB
```

```
-newer   file.txt
-newerm  file.txt        # modified newer than file.txt
-newerX  file.txt        # [c]hange, [m]odified, [B]create
-newerXt "1 hour ago"    # [t]imestamp
```

Examples

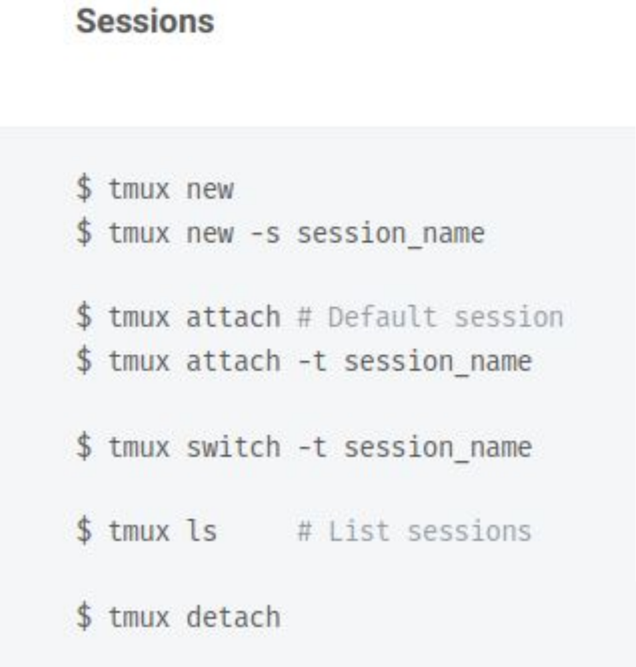
```
find . -name '*.jpg'
find . -name '*.jpg' -exec rm {} \;
```

```
find . -newerBt "24 hours ago"
```

```
find . -type f -mtime +29 # find files modified more than 30 days ago
```

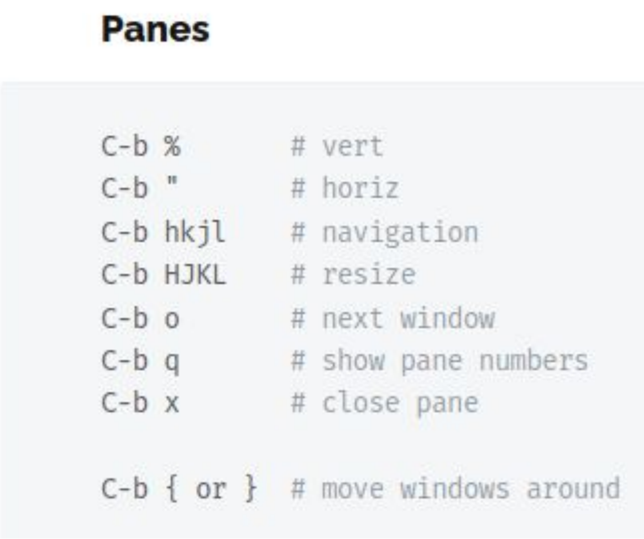
HOW TO USE TMUX

TMUX



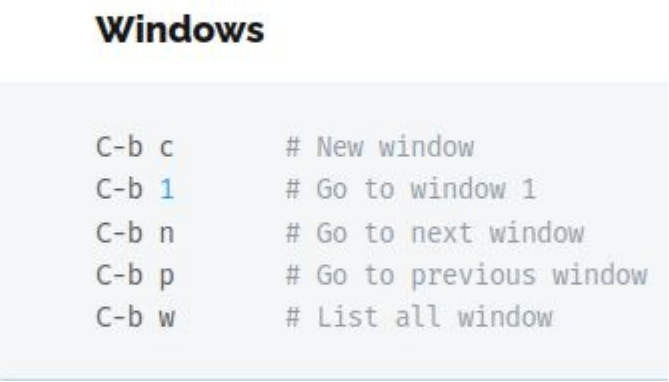
Sessions

```
$ tmux new
$ tmux new -s session_name

$ tmux attach # Default session
$ tmux attach -t session_name

$ tmux switch -t session_name

$ tmux ls     # List sessions

$ tmux detach
```



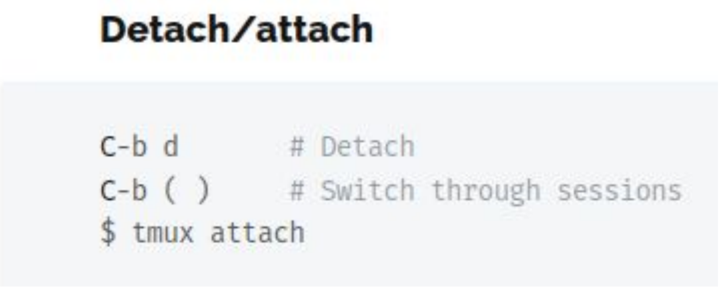
Panes

```
C-b %        # vert
C-b "        # horiz
C-b hkjl     # navigation
C-b HJKL     # resize
C-b o        # next window
C-b q        # show pane numbers
C-b x        # close pane

C-b { or }   # move windows around
```



Windows

```
C-b c        # New window
C-b 1        # Go to window 1
C-b n        # Go to next window
C-b p        # Go to previous window
C-b w        # List all window
```

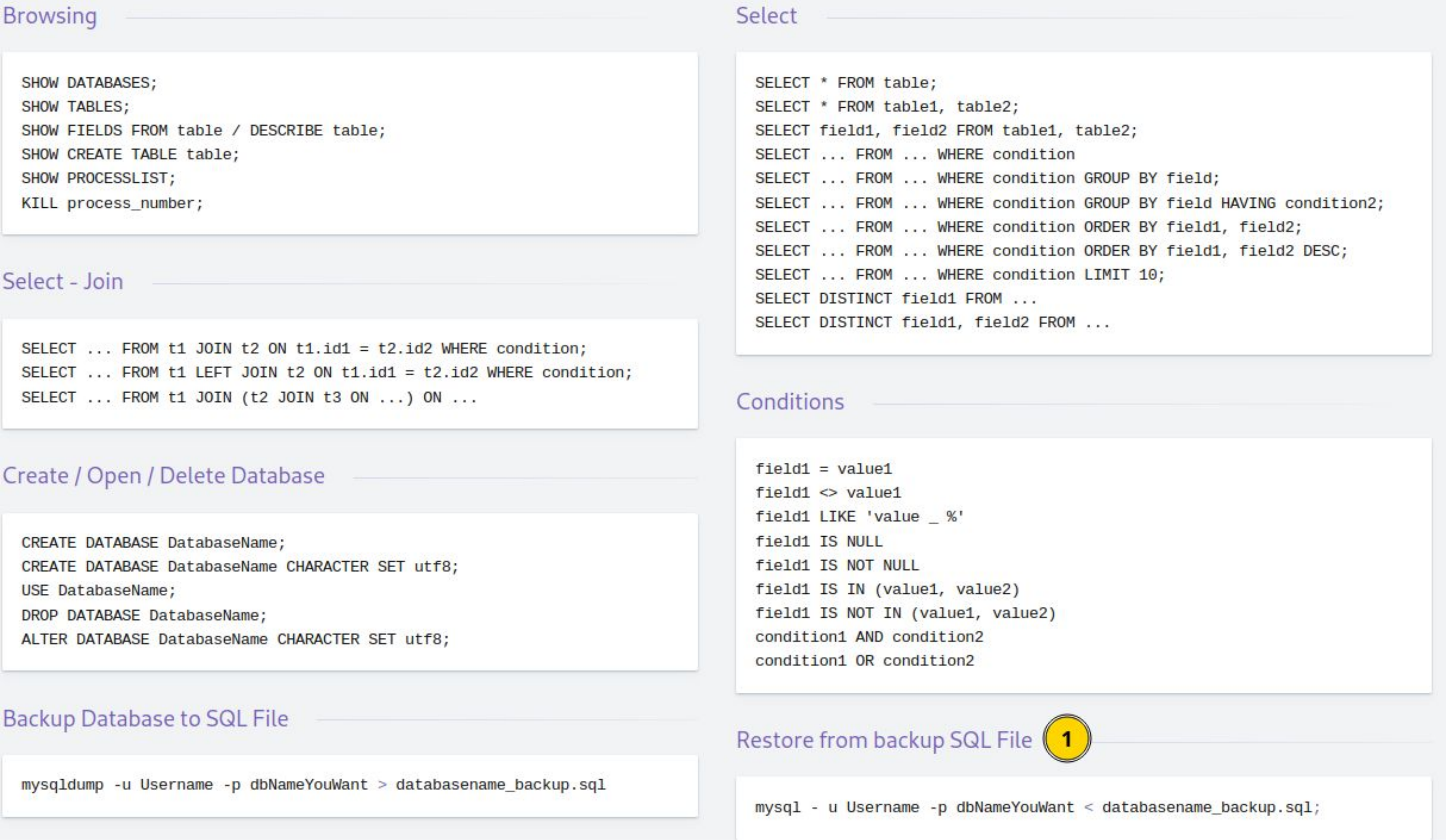


Detach/attach

```
C-b d        # Detach
C-b ( )      # Switch through sessions
$ tmux attach
```

HOW TO USE MYSQL

MYSQL



Browsing

```
SHOW DATABASES;
SHOW TABLES;
SHOW FIELDS FROM table / DESCRIBE table;
SHOW CREATE TABLE table;
SHOW PROCESSLIST;
KILL process_number;
```

Select - Join

```
SELECT ... FROM t1 JOIN t2 ON t1.id1 = t2.id2 WHERE condition;
SELECT ... FROM t1 LEFT JOIN t2 ON t1.id1 = t2.id2 WHERE condition;
SELECT ... FROM t1 JOIN (t2 JOIN t3 ON ...) ON ...
```

Create / Open / Delete Database

```
CREATE DATABASE DatabaseName;
CREATE DATABASE DatabaseName CHARACTER SET utf8;
USE DatabaseName;
DROP DATABASE DatabaseName;
ALTER DATABASE DatabaseName CHARACTER SET utf8;
```

Backup Database to SQL File

```
mysqldump -u Username -p dbNameYouWant > databasename_backup.sql
```

Select

```
SELECT * FROM table;
SELECT * FROM table1, table2;
SELECT field1, field2 FROM table1, table2;
SELECT ... FROM ... WHERE condition
SELECT ... FROM ... WHERE condition GROUP BY field;
SELECT ... FROM ... WHERE condition GROUP BY field HAVING condition2;
SELECT ... FROM ... WHERE condition ORDER BY field1, field2;
SELECT ... FROM ... WHERE condition ORDER BY field1, field2 DESC;
SELECT ... FROM ... WHERE condition LIMIT 10;
SELECT DISTINCT field1 FROM ...
SELECT DISTINCT field1, field2 FROM ...
```

Conditions

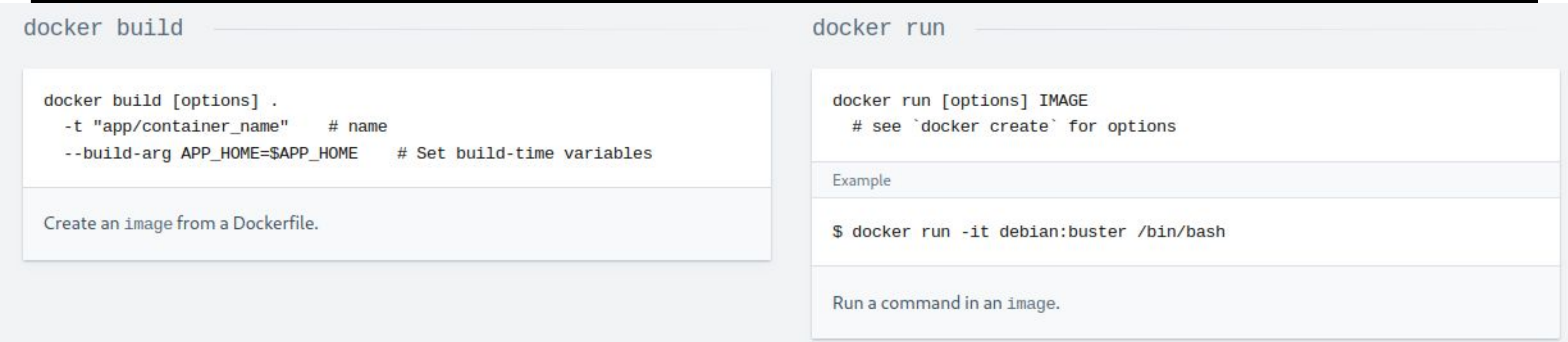
```
field1 = value1
field1 <> value1
field1 LIKE 'value _ %'
field1 IS NULL
field1 IS NOT NULL
field1 IS IN (value1, value2)
field1 IS NOT IN (value1, value2)
condition1 AND condition2
condition1 OR condition2
```

Restore from backup SQL File

```
mysql - u Username -p dbNameYouWant < databasename_backup.sql;
```

HOW TO USE DOCKER

DOCKER



docker build

```
docker build [options] .
  -t "app/container_name"    # name
  --build-arg APP_HOME=$APP_HOME    # Set build-time variables
```

Create an image from a Dockerfile.

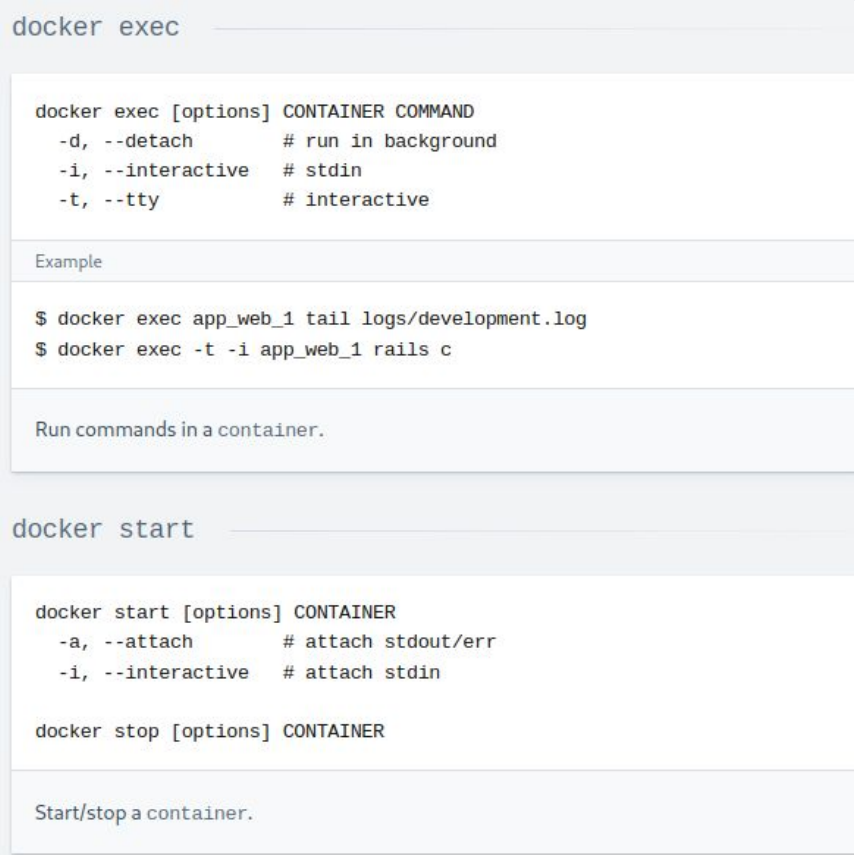
docker run

```
docker run [options] IMAGE
  # see `docker create` for options
```

Example

```
$ docker run -it debian:buster /bin/bash
```

Run a command in an image.



docker exec

```
docker exec [options] CONTAINER COMMAND
  -d, --detach        # run in background
  -i, --interactive   # stdin
  -t, --tty           # interactive
```

Example

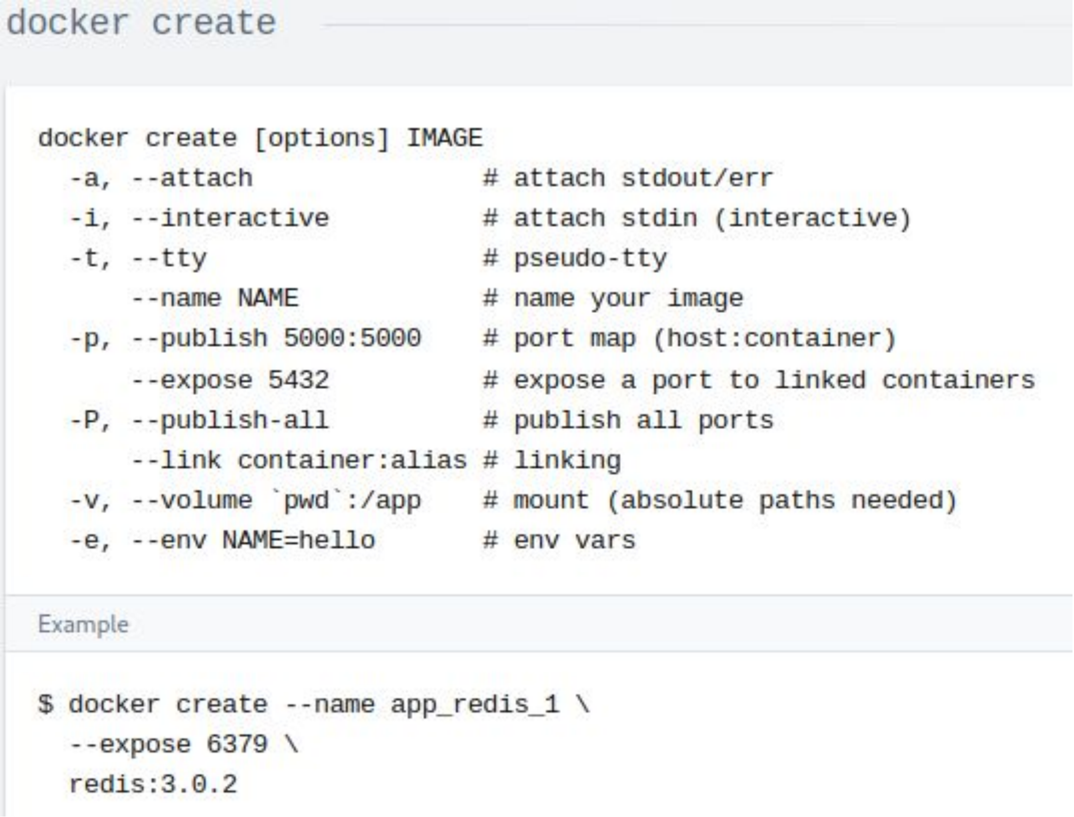
```
$ docker exec app_web_1 tail logs/development.log
$ docker exec -t -i app_web_1 rails c
```

Run commands in a container.

docker start

```
docker start [options] CONTAINER
  -a, --attach        # attach stdout/err
  -i, --interactive   # attach stdin

docker stop [options] CONTAINER
```

Start/stop a container.



docker create

```
docker create [options] IMAGE
  -a, --attach               # attach stdout/err
  -i, --interactive          # attach stdin (interactive)
  -t, --tty                  # pseudo-tty
      --name NAME            # name your image
  -p, --publish 5000:5000    # port map (host:container)
      --expose 5432          # expose a port to linked containers
  -P, --publish-all          # publish all ports
      --link container:alias # linking
  -v, --volume `pwd`:/app    # mount (absolute paths needed)
  -e, --env NAME=hello       # env vars
```

Example

```
$ docker create --name app_redis_1 \
  --expose 6379 \
  redis:3.0.2
```

HOW TO USE NMAP

NMAP

- Ping Scan - use to discover assets on network
  - **nmap -sp 192.100.1.1/24**
- Service harvesting
  - **nmap -sC -sV -oA nmap/scan.nmap 192.100.1.1/24**

HOW TO USE SYSTEMCTL

SYSTEMCTL

- Get enabled services
  - **sudo systemctl list-unit-files | grep enabled**
- Scroll through enabled services
  - **sudo systemctl -l**
- Common commands
  - **sudo systemctl start <service>**
  - **sudo systemctl status <service>**
  - **sudo systemctl stop <service>**
  - **sudo systemctl restart <service>**
- If they disabled it (or re-enable)
  - **sudo systemctl enable <service>**
  - **sudo systemctl disable <service>**

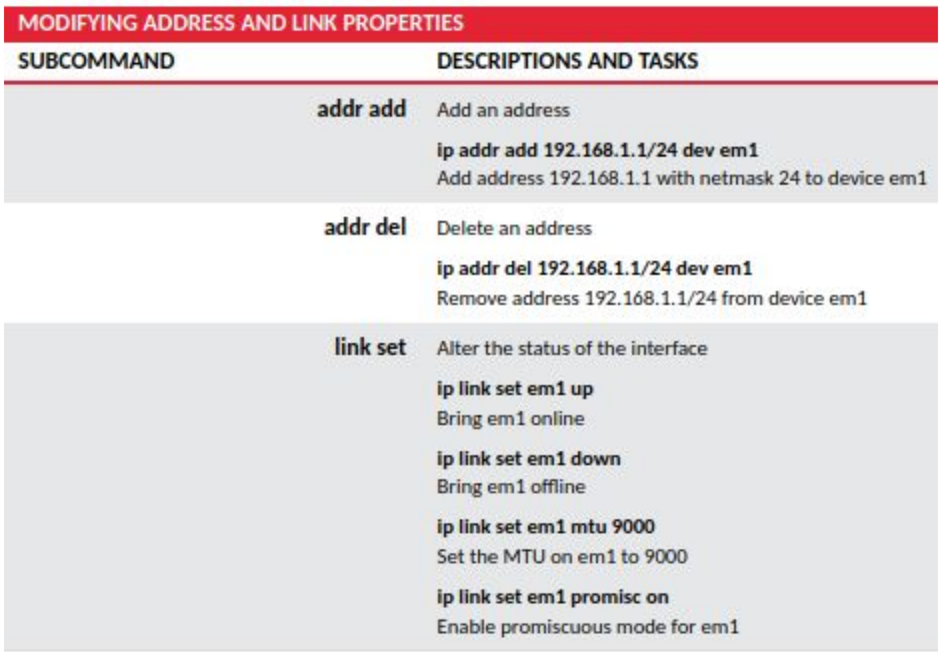
ACCOUNT MANAGEMENT

ACCOUNT MANAGEMENT

- Lock / unlock user
  - **passwd -l <username>**
  - **passwd -u <username>**
- Add sysadmin user (run as root)
  - **useradd -p password -m -d /var/$ADMINUSER $ADMINUSER**
    - **-m** create home **-d** where is home
  - **usermod -aG sudo $ADMINUSER**
  - **chmod -R 750 /var/$ADMINUSER**
  - **chown -R $ADMINUSER /var/$ADMINUSER**
- Deleting user
  - **sudo userdel -r <username>**
- Get user in group
  - **sudo getent group <group_name>**

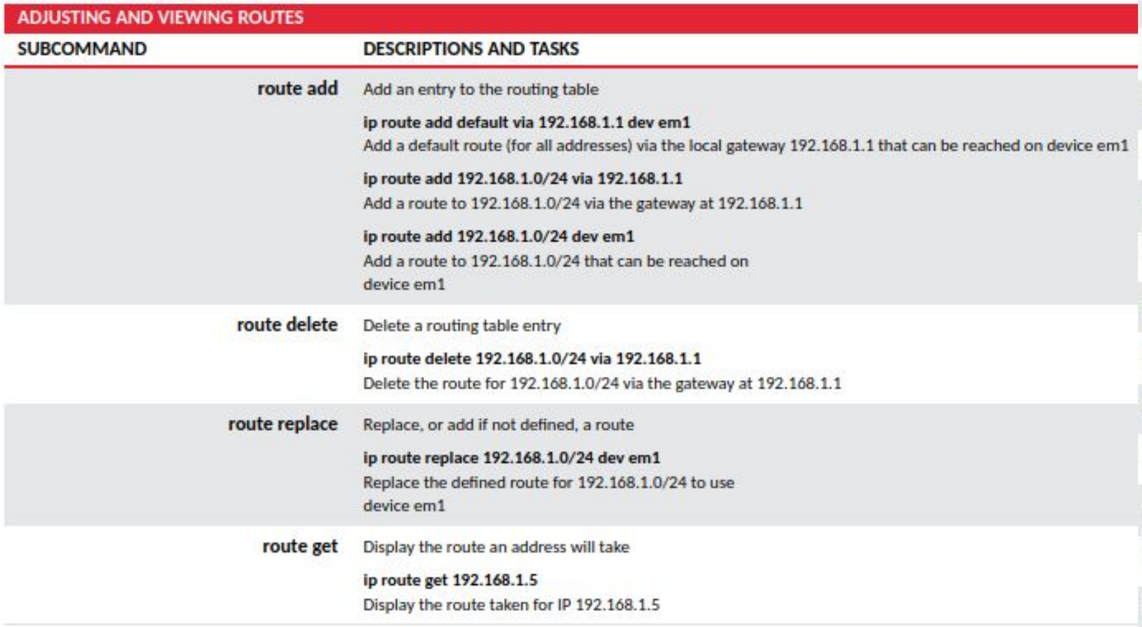
IP TOOLS

IP TOOLS



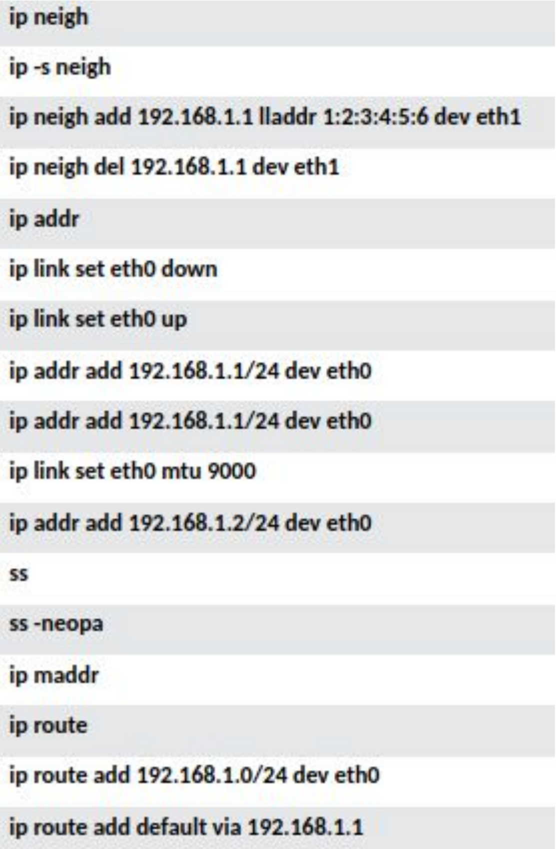
MODIFYING ADDRESS AND LINK PROPERTIES

SUBCOMMAND	DESCRIPTIONS AND TASKS
addr add	Add an address **ip addr add 192.168.1.1/24 dev em1** Add address 192.168.1.1 with netmask 24 to device em1
addr del	Delete an address **ip addr del 192.168.1.1/24 dev em1** Remove address 192.168.1.1/24 from device em1
link set	Alter the status of the interface **ip link set em1 up** Bring em1 online **ip link set em1 down** Bring em1 offline **ip link set em1 mtu 9000** Set the MTU on em1 to 9000 **ip link set em1 promisc on** Enable promiscuous mode for em1



ADJUSTING AND VIEWING ROUTES

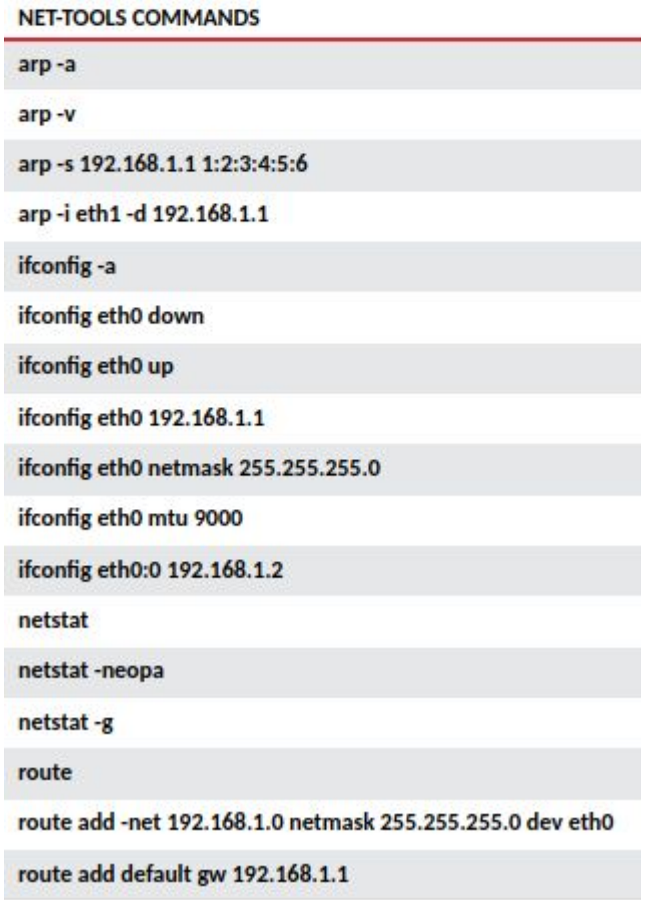
SUBCOMMAND	DESCRIPTIONS AND TASKS
route add	Add an entry to the routing table **ip route add default via 192.168.1.1 dev em1** Add a default route (for all addresses) via the local gateway 192.168.1.1 that can be reached on device em1 **ip route add 192.168.1.0/24 via 192.168.1.1** Add a route to 192.168.1.0/24 via the gateway at 192.168.1.1 **ip route add 192.168.1.0/24 dev em1** Add a route to 192.168.1.0/24 that can be reached on device em1
route delete	Delete a routing table entry **ip route delete 192.168.1.0/24 via 192.168.1.1** Delete the route for 192.168.1.0/24 via the gateway at 192.168.1.1
route replace	Replace, or add if not defined, a route **ip route replace 192.168.1.0/24 dev em1** Replace the defined route for 192.168.1.0/24 to use device em1
route get	Display the route an address will take **ip route get 192.168.1.5** Display the route taken for IP 192.168.1.5



```
ip neigh
ip -s neigh
ip neigh add 192.168.1.1 lladdr 1:2:3:4:5:6 dev eth1
ip neigh del 192.168.1.1 dev eth1
ip addr
ip link set eth0 down
ip link set eth0 up
ip addr add 192.168.1.1/24 dev eth0
ip addr add 192.168.1.1/24 dev eth0
ip link set eth0 mtu 9000
ip addr add 192.168.1.2/24 dev eth0
ss
ss -neopa
ip maddr
ip route
ip route add 192.168.1.0/24 dev eth0
ip route add default via 192.168.1.1
```

NET-TOOLS

IP TOOLS



NET-TOOLS COMMANDS
arp -a
arp -v
arp -s 192.168.1.1 1:2:3:4:5:6
arp -i eth1 -d 192.168.1.1
ifconfig -a
ifconfig eth0 down
ifconfig eth0 up
ifconfig eth0 192.168.1.1
ifconfig eth0 netmask 255.255.255.0
ifconfig eth0 mtu 9000
ifconfig eth0:0 192.168.1.2
netstat
netstat -neopa
netstat -g
route
route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0
route add default gw 192.168.1.1

UPDATE LINUX KERNEL

KERNEL

- If we want we can update kernel
- Ubuntu
  - **sudo apt-get update; sudo apt-get install linux-virtual**
- Debian
  - **sudo apt-cache search linux-image** # find appropriate version
  - **sudo apt install linux-image-<flavour>** # install
- Redhat
  - **sudo yum update kernel**

Securing SSH

/etc/ssh/ssh_config

- First *BACK IT UP*
  - **sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.bak**
- Must add rules to add
  - **PermitEmptyPasswords no**
  - **PermitRootLogin no**
  - **IgnoreRhosts yes**
  - **Protocol 2**
  - **ClientAliveInterval 180** # if inactive for 3 minutes kill session
  - **AllowUsers <username>** # if they do not use ssh for scoring engine…
  - **MaxAuthTries 3**
  - **X11Forwarding no**
  - **AllowAgentForwarding no**
  - **AllowTcpForwarding no**
  - **PermitTunnel no**
- To confirm rules
  - **sudo sshd -T**
- To apply changes
  - **sudo systemctl restart sshd**
- Cheeky shenanigans
  - **AllowUsers *@203.0.113.1 sammy@203.0.113.2** # allows only certain IP to ssh

Securing Sudo

/etc/sudoers

- First *BACK IT UP*
  - **sudo cp /etc/sudoers /etc/sudoers.bak**
- Remove
  - **<username> ALL=(ALL:ALL) ALL** # tis a bad idea
  - TODO confirm
- Must add rules to add
  - **Defaults requiretty**
  - **Defaults logfile=/var/log/sudo.log**
  - **Defaults use_pty**
- Cheeky rules to add
  - **Defaults timestamp_timeout=0** # require password for each command
  - **Defaults insults** # insults you when wrong

Setting up fail2ban

/etc/fail2ban/jail.local

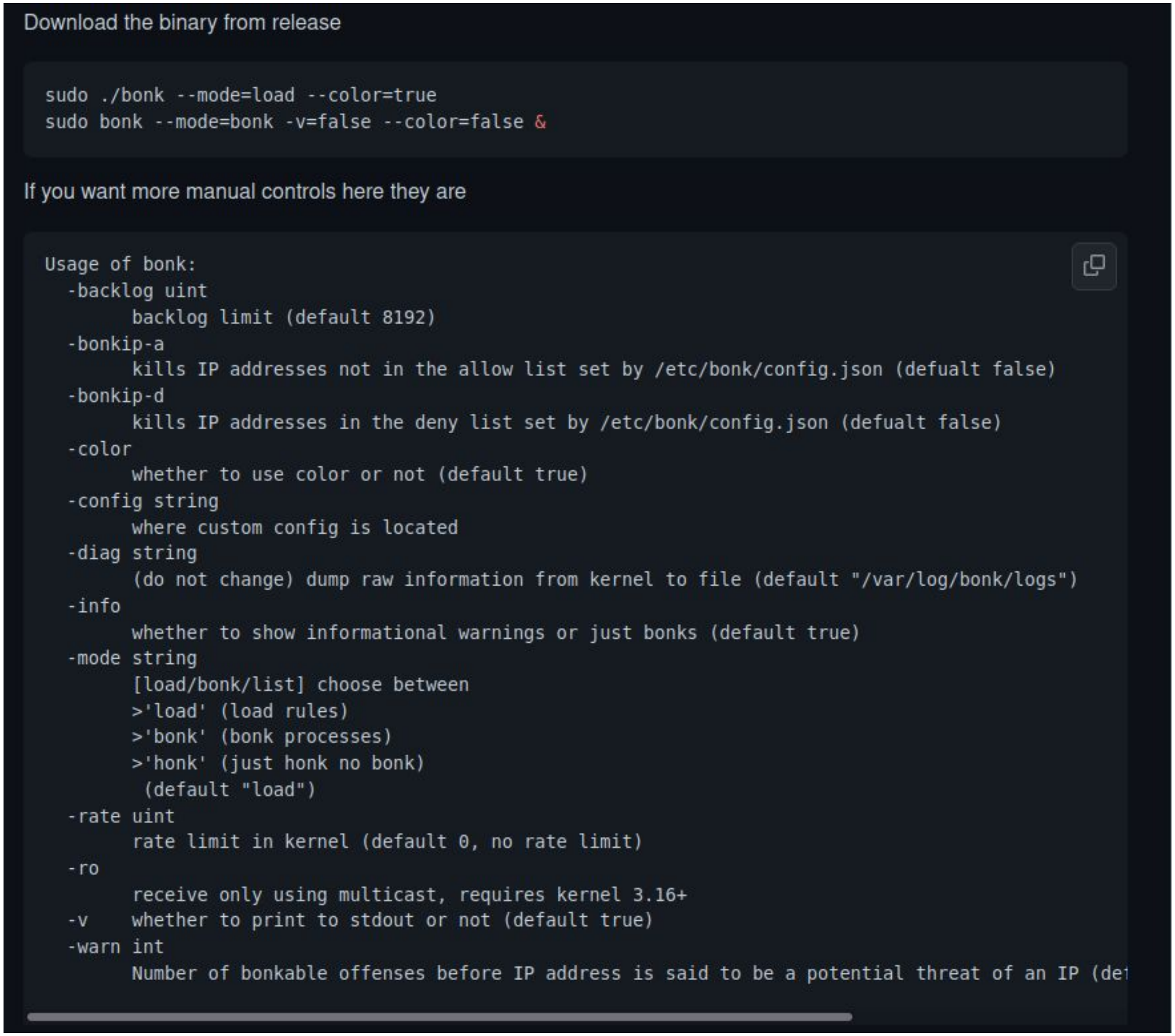
- Install it
  - **<pkg manager> install fail2ban**
- Add these rules
  - **[sshd]**
  - **enabled = true**
  - **bantime = 5m**
  - **maxretry = 3**
- Start the service
  - **systemctl start fail2ban**

Setting up mod_security

- APACHE
  - UBUNTU
    - **apt install libapache2-mod-security2**
    - **sudo a2enmod security2**
    - **sudo systemctl restart apache2**
  - FEDORA / REDHAT
    - **sudo yum install mod_security**
    - **sudo systemctl restart httpd**
- RULES
  - **sudo cp /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/modsecurity.conf**
  - **sudo nano /etc/modsecurity/modsecurity.conf**
  - Near the top of the file, you'll see **SecRuleEngine DetectionOnly**. Change **DetectionOnly** to **On**.
  - **systemctl restart apache2 #** or httpd
- OWASP CRS
  - **wget https://github.com/coreruleset/coreruleset/archive/v3.3.0.zip**
  - **unzip FileName.zip**
  - **mv coreruleset-3.3.0/crs-setup.conf.example /etc/modsecurity/crs-setup.conf**
  - **mv coreruleset-3.3.0/rules/ /etc/modsecurity/**
  - **nano /etc/apache2/mods-enabled/security2.conf**
    - **IncludeOptional /etc/modsecurity/*.conf**
    - **Include /etc/modsecurity/rules/*.conf**
  - **systemctl restart apache2**

Setting up Bonk

/var/log/bonk/bonk.log, /etc/bonk/config.json



Download the binary from release

```
sudo ./bonk --mode=load --color=true
sudo bonk --mode=bonk -v=false --color=false &
```

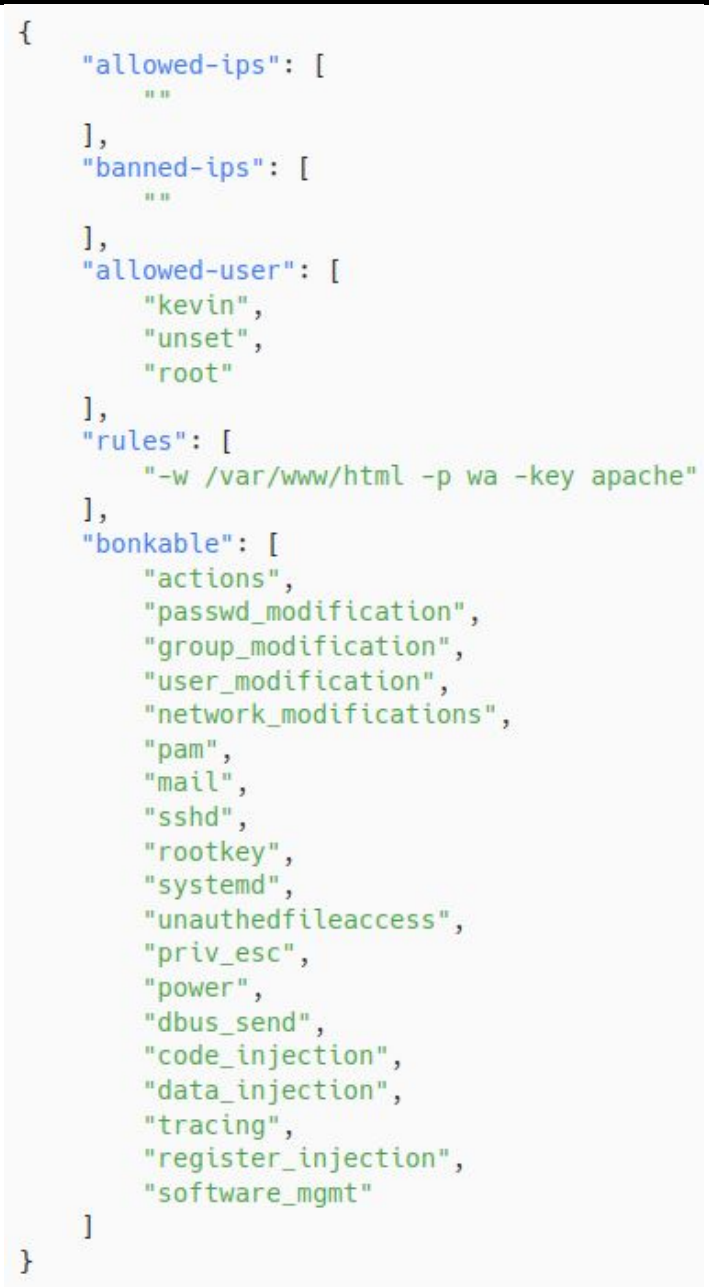
If you want more manual controls here they are

```
Usage of bonk:
  -backlog uint
        backlog limit (default 8192)
  -bonkip-a
        kills IP addresses not in the allow list set by /etc/bonk/config.json (defualt false)
  -bonkip-d
        kills IP addresses in the deny list set by /etc/bonk/config.json (defualt false)
  -color
        whether to use color or not (default true)
  -config string
        where custom config is located
  -diag string
        (do not change) dump raw information from kernel to file (default "/var/log/bonk/logs")
  -info
        whether to show informational warnings or just bonks (default true)
  -mode string
        [load/bonk/list] choose between
        >'load' (load rules)
        >'bonk' (bonk processes)
        >'honk' (just honk no bonk)
         (default "load")
  -rate uint
        rate limit in kernel (default 0, no rate limit)
  -ro
        receive only using multicast, requires kernel 3.16+
  -v    whether to print to stdout or not (default true)
  -warn int
        Number of bonkable offenses before IP address is said to be a potential threat of an IP (def
```

- **Modes**
- Load
  - yells at kernel to add rule, exits.
- Bonk
  - listens at the kernel yelling. Checks against the config.json file to see allowed users (if you remove root / current user / unset bonk will just kill itself). If the syscall is naughty, bonk)
- Honk
  - does not bonk just says hey I would nuke this if you want me to!
- bonkip-a / bonkip-b
  - looks at the process table to get the IP address and compares it against that of the config. If it violates it either tells you or bonks it.

Setting up Bonk

/var/log/bonk/bonk.log, /etc/bonk/config.json

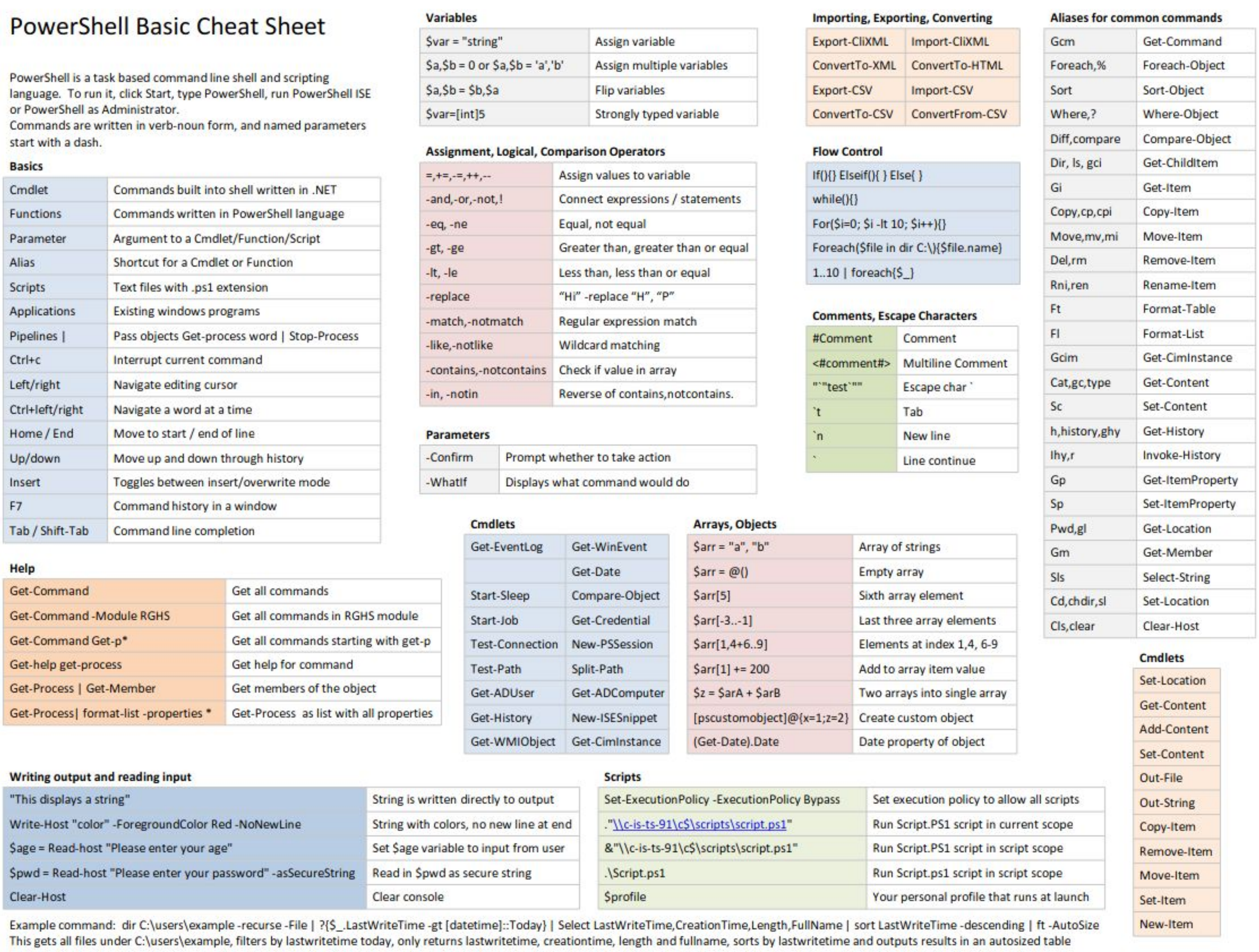


```
{
    "allowed-ips": [
        ""
    ],
    "banned-ips": [
        ""
    ],
    "allowed-user": [
        "kevin",
        "unset",
        "root"
    ],
    "rules": [
        "-w /var/www/html -p wa -key apache"
    ],
    "bonkable": [
        "actions",
        "passwd_modification",
        "group_modification",
        "user_modification",
        "network_modifications",
        "pam",
        "mail",
        "sshd",
        "rootkey",
        "systemd",
        "unauthedfileaccess",
        "priv_esc",
        "power",
        "dbus_send",
        "code_injection",
        "data_injection",
        "tracing",
        "register_injection",
        "software_mgmt"
    ]
}
```

- How to Configure
- Allowed IPs are IPs that should never be bonked with the flag **bonkip-a**
- Banned IPs are IPs that can be bonked with the flag **bonkip-b**
- Rules are audit based rules for the kernel to watch
- Bonkable are the rules to bonk

POWERSHELL

POWERSHELL



PowerShell Basic Cheat Sheet

PowerShell is a task based command line shell and scripting language. To run it, click Start, type PowerShell, run PowerShell ISE or PowerShell as Administrator.
Commands are written in verb-noun form, and named parameters start with a dash.

Basics

Cmdlet	Commands built into shell written in .NET
Functions	Commands written in PowerShell language
Parameter	Argument to a Cmdlet/Function/Script
Alias	Shortcut for a Cmdlet or Function
Scripts	Text files with .ps1 extension
Applications	Existing windows programs
Pipelines |	Pass objects Get-process word | Stop-Process
Ctrl+c	Interrupt current command
Left/right	Navigate editing cursor
Ctrl+left/right	Navigate a word at a time
Home / End	Move to start / end of line
Up/down	Move up and down through history
Insert	Toggles between insert/overwrite mode
F7	Command history in a window
Tab / Shift-Tab	Command line completion

Help

Get-Command	Get all commands
Get-Command -Module RGHS	Get all commands in RGHS module
Get-Command Get-p*	Get all commands starting with get-p
Get-help get-process	Get help for command
Get-Process | Get-Member	Get members of the object
Get-Process| format-list -properties *	Get-Process as list with all properties

Variables

$var = "string"	Assign variable
$a,$b = 0 or $a,$b = 'a','b'	Assign multiple variables
$a,$b = $b,$a	Flip variables
$var=[int]5	Strongly typed variable

Assignment, Logical, Comparison Operators

=,+=,-=,++,--	Assign values to variable
-and,-or,-not,!	Connect expressions / statements
-eq, -ne	Equal, not equal
-gt, -ge	Greater than, greater than or equal
-lt, -le	Less than, less than or equal
-replace	"Hi" -replace "H", "P"
-match,-notmatch	Regular expression match
-like,-notlike	Wildcard matching
-contains,-notcontains	Check if value in array
-in, -notin	Reverse of contains,notcontains.

Parameters

-Confirm	Prompt whether to take action
-WhatIf	Displays what command would do

Importing, Exporting, Converting

Export-CliXML	Import-CliXML
ConvertTo-XML	ConvertTo-HTML
Export-CSV	Import-CSV
ConvertTo-CSV	ConvertFrom-CSV

Flow Control

If(){} Elseif(){ } Else{ }
while(){}
For($i=0; $i -lt 10; $i++){}
Foreach($file in dir C:\){$file.name}
1..10 | foreach{$_}

Comments, Escape Characters

#Comment	Comment
<#comment#>	Multiline Comment
"`"test`""	Escape char `
`t	Tab
`n	New line
`	Line continue

Aliases for common commands

Gcm	Get-Command
Foreach,%	Foreach-Object
Sort	Sort-Object
Where,?	Where-Object
Diff,compare	Compare-Object
Dir, ls, gci	Get-ChildItem
Gi	Get-Item
Copy,cp,cpi	Copy-Item
Move,mv,mi	Move-Item
Del,rm	Remove-Item
Rni,ren	Rename-Item
Ft	Format-Table
Fl	Format-List
Gcim	Get-CimInstance
Cat,gc,type	Get-Content
Sc	Set-Content
h,history,ghy	Get-History
Ihy,r	Invoke-History
Gp	Get-ItemProperty
Sp	Set-ItemProperty
Pwd,gl	Get-Location
Gm	Get-Member
Sls	Select-String
Cd,chdir,sl	Set-Location
Cls,clear	Clear-Host

Cmdlets

Get-EventLog	Get-WinEvent
	Get-Date
Start-Sleep	Compare-Object
Start-Job	Get-Credential
Test-Connection	New-PSSession
Test-Path	Split-Path
Get-ADUser	Get-ADComputer
Get-History	New-ISESnippet
Get-WMIObject	Get-CimInstance

Arrays, Objects

$arr = "a", "b"	Array of strings
$arr = @()	Empty array
$arr[5]	Sixth array element
$arr[-3..-1]	Last three array elements
$arr[1,4+6..9]	Elements at index 1,4, 6-9
$arr[1] += 200	Add to array item value
$z = $arA + $arB	Two arrays into single array
[pscustomobject]@{x=1;z=2}	Create custom object
(Get-Date).Date	Date property of object

Cmdlets

Set-Location
Get-Content
Add-Content
Set-Content
Out-File
Out-String
Copy-Item
Remove-Item
Move-Item
Set-Item
New-Item

Writing output and reading input

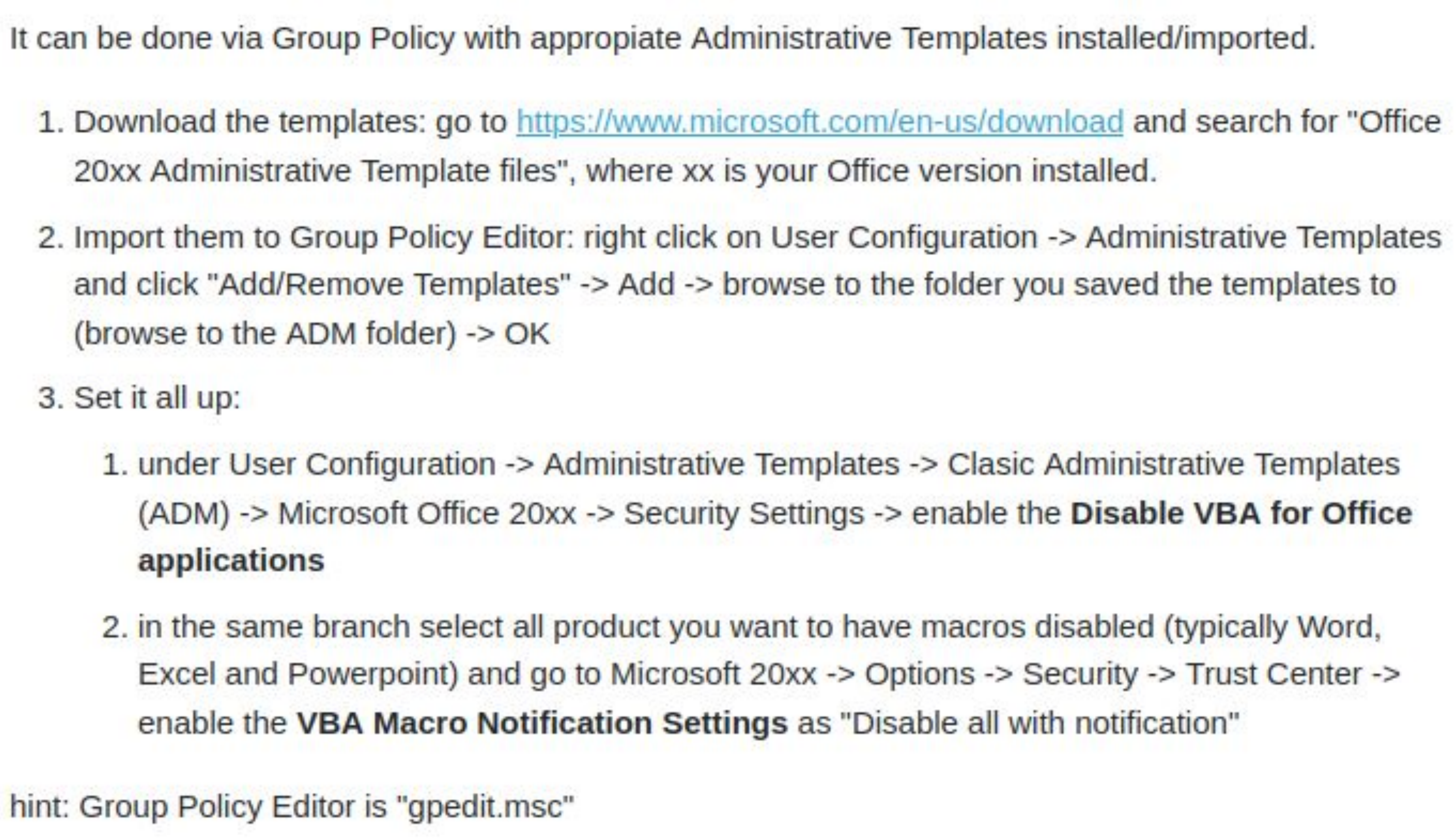
"This displays a string"	String is written directly to output
Write-Host "color" -ForegroundColor Red -NoNewLine	String with colors, no new line at end
$age = Read-host "Please enter your age"	Set $age variable to input from user
$pwd = Read-host "Please enter your password" -asSecureString	Read in $pwd as secure string
Clear-Host	Clear console

Scripts

Set-ExecutionPolicy -ExecutionPolicy Bypass	Set execution policy to allow all scripts
."\\c-is-ts-91\c$\scripts\script.ps1"	Run Script.PS1 script in current scope
&"\\c-is-ts-91\c$\scripts\script.ps1"	Run Script.PS1 script in script scope
.\Script.ps1	Run Script.ps1 script in script scope
$profile	Your personal profile that runs at launch

Example command: dir C:\users\example -recurse -File | ?{$_.LastWriteTime -gt [datetime]::Today} | Select LastWriteTime,CreationTime,Length,FullName | sort LastWriteTime -descending | ft -AutoSize
This gets all files under C:\users\example, filters by lastwritetime today, only returns lastwritetime, creationtime, length and fullname, sorts by lastwritetime and outputs results in an autosized table

Disabling Macros (Office)



It can be done via Group Policy with appropiate Administrative Templates installed/imported.

1. Download the templates: go to https://www.microsoft.com/en-us/download and search for "Office 20xx Administrative Template files", where xx is your Office version installed.
2. Import them to Group Policy Editor: right click on User Configuration -> Administrative Templates and click "Add/Remove Templates" -> Add -> browse to the folder you saved the templates to (browse to the ADM folder) -> OK
3. Set it all up:
    1. under User Configuration -> Administrative Templates -> Clasic Administrative Templates (ADM) -> Microsoft Office 20xx -> Security Settings -> enable the **Disable VBA for Office applications**
    2. in the same branch select all product you want to have macros disabled (typically Word, Excel and Powerpoint) and go to Microsoft 20xx -> Options -> Security -> Trust Center -> enable the **VBA Macro Notification Settings** as "Disable all with notification"

hint: Group Policy Editor is "gpedit.msc"