CyberPatriot Linux Checklist

[Redacted] Team ID [Redacted]

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Comments |
| 1.0 | 1/16/2019 | [Redacted] | Kill me! |
| 1.0.1 | 8/9/2019 | [Redacted] | Added source list |

# Getting Started

* Bump up RAM to 4gb (Maximum)
* Increase CPU cores from 1 to 2 (or maybe even 3 if you’re feeling wild)
* Main Team ID: [Redacted]

# Readme

* Take
* Damn
* Notes
* Take a photo of passwords!

# Forensics

* Use Google

# Rest of forensics

* Hashing
  + Generally **[hashtype]sum [filename]** will work
  + **md5sum [filename]**
  + **sha256sum [filename]**
* Media files
  + **ls -al** shows hidden files & folders. (Run this in the specified folder)
* Hostname
  + **hostname**
* Logs
  + **cd /var/log**
  + **sudo grep -r [search term]**

# Updates

* Starting with updates
* Things in **BOLD** are run in terminal (ctrl-alt-t to open terminal)
* Open terminal
  + **sudo apt-get update**
  + **sudo apt-get upgrade**
  + **sudo apt-get dist-upgrade**
  + This will take awhile (You will want to run this multiple times throughout the comp)
* Search “Software Updater” and follow the prompts

## Automatic Updates

* Go to “System Settings” -> “Software and Updates”
* Click “Updates”
* Check “Important Security Updates”
* Check “Recommended Updates”
* Uncheck “Unsupported Updates”
* Automatically check for updates: “Daily”
* When there are security updates: “Download and Install Immediately”
  + If this is greyed out run this command
  + **sudo rm /etc/apt/apt.conf.d/20auto-upgrades**

**Debian**

* **sudo aptitude update -y && sudo aptitude install unattended-upgrades apt-listchanges -y**
* **sudo gedit /etc/apt/apt.conf.d/50unattended-upgrades**
  + Unattended-Upgrade::Mail "root";
* **sudo dpkg-reconfigure -plow unattended-upgrades**
* **sudo gedit /etc/apt/apt.conf.d/20auto-upgrades**
  + Make sure the following lines where added
  + APT::Periodic::Update-Package-Lists "1";  
    APT::Periodic::Unattended-Upgrade "1";
  + And add this line
  + APT::Periodic::Verbose "2";

# Verify Sources.list

* **sudo mv /etc/apt/sources.list ~/** #Backs up source
* **sudo gedit /etc/apt/sources.list**
* [https://repogen.simplylinux.ch/#](https://repogen.simplylinux.ch/)
* Paste in the new sources and save

# Script

* MAKE SURE THAT THE SCRIPT IS INSIDE THE MLGA FOLDER ON THE DESKTOP!

# Users

* “System Settings” -> “User Accounts”
  + Click the lock in the top right of the screen
  + Check for unauthorized users
  + Check for users that are Admins who aren’t supposed to be
  + Check for users that are supposed to be admins
  + Make sure only YOUR account has auto login enabled
  + Passwords
    - **DO NOT CHANGE YOUR PASSWORD, IF YOU DO THE IMAGE HAS A HIGHER CHANCE OF BREAKING**
    - Change all of the passwords to IH@t3Pupp135 (Except your account)
  + /etc/passwd
    - **sudo EDITOR=gedit vipw**
    - If there is a malicious user inside the block of users, do the following
      * The Nobody user is supposed to be there
      * Make sure it’s not supposed be there (readme)
      * If the user meets the following requirements, copy in line into a notepad (or some other notetaking thing) and the delete the line and save the file.
    - If a user has a UID set to another users UID, change it to an unused UID above 1000
    - If a user UID is set to 0, set it to an unused UID above 1000, or delet if bad
    - Every system account (UID 0-1000) should have its shell (last field) set to /usr/sbin/nologin or /bin/false
      * Exceptions:
      * ‘sync’ – ‘/bin/sync’
      * ‘gnats’ – maybe? Switch it back and forth.
      * ‘libuuid’ – no shell.

# Groups

* **sudo EDITOR=gedit vigr**
* The format for groups is [Group name]:x:[group ID]:[group member],[group member]
* So an example would be [Redacted]
* Adm
  + This is the default admin group
  + Make sure only verified admins are on there
  + syslog is a system account and should remain
* Admin
  + This sometimes has similar perms to Adm
  + Make sure only verified admins are on there
  + syslog is a system account and should remain
* Users
  + Default users group
  + Verify that the correct members are there
* Sudo
  + This group has the perms to use Sudo
  + The members should typically be the same as adm/admin
* Other groups
  + Most of the time there will be groups named after users
    - The only member (if any) should the user that the group is named after
  + Sambashare
    - No users unless specified
  + Nopasswdlogin
    - No members (Do not mess with yourself or CP)
* SAVE!

# Locking the Root Account

* **sudo EDITOR=gedit vipw**
* Search for the line that starts with “root”
* Replace “/bin/bash” with “/usr/sbin/nologin”
* SAVE
* **sudo usermod -L root**

# Configuring Password Expiry and login.defs

* This is really tedious but it’s really important
* **sudo apt-get install libpam-cracklib -y**
* **sudo EDITOR=gedit vipw -s**
  + Go to the users of the system and scroll to the very edge
  + Change the “0:99999:7” to “1:60:7”
  + Go to the first colon and change the stuff in between the dollar signs to a 6
    - E.G. [Redacted]:$6$sdfgsdfgs...
* SAVE and close
* **sudo gedit /etc/login.defs**
  + Ctrl+f and search for “PASS\_MAX\_DAYS”
    - For “PASS\_MAX\_DAYS”, change it to 60
    - For “PASS\_MIN\_DAYS”, change it to 1
    - For “PASS\_WARN\_DAYS”, change it to 7
  + Ctrl+f and search for “ENCRYPT\_METHOD”
    - Make sure that it is set to SHA512
  + Ctrl+f and search for “rounds”
    - Make sure all of the lines are hashed out mentioning rounds
  + SAVE
* **sudo chmod 640 /etc/shadow**

# Firewall

* **sudo ufw enable**
* Add firewall rules for critical services
  + E.G. SSH: **sudo ufw allow ssh**
  + E.G. Web server: **sudo ufw allow 80**
* **sudo apt-get install gufw -y**
* **sudo ufw logging on**
* **sudo gufw**
  + Deny incoming connections
  + Allow outgoing connections
  + Edit -> Preferences
  + Logging: Full
  + Check “Logging Gufw activity”
  + Check “Show confirm dialog for deleting rules”

# Networking

* Current network status is **netstat -tulpen**
* To view listening connections **netstat -I** (Uppercase i)
  + Look for
    - Listening connections
    - Open connects (be careful of CP and browser)
    - A PID of 825 is a system process and is gucci
* If you think it is malicious use htop to link its PID to a process, From there find what’s starting the program and kill it.
* Open ports
  + **sudo apt-get install nmap -y**
  + **sudo nmap -sT -O localhost**
  + Make sure all of these are legit
  + If you see something you don’t know run **netstat -anp | grep [port]**
* Services called “unknown” mean that they are unknown to the kernel and likely illegitimate.
  + Link its PID and burn it with HTOP
* **sudo gedit /etc/host.conf**
  + Add nospoof on
  + Add order bind,hosts

# Scheduled Tasks

* Handled through cron on linux
* You shouldn’t have to check this for every user on the system, just root and the user you are using
* To check crontabs: **sudo crontab -l -u [username]**
* **sudo nano /etc/cron.d/anacron**
* Crontabs also hide in other places
  + /etc/crontab
  + /etc/cron.d
  + /etc/crond.hourly, daily, weekly, monthly
  + /var/spool/cron
  + /var/spool/cron/crontabs
* Check /root/ for crontabs.

# Antivirus & Rootkit

* **sudo apt-get install clamav -y**
* **sudo apt-get install clamtk -y**
* **sudo clamtk** (Scan home directory)

# /etc/pam.d/

* PAM is a program that helps you secure Linux through config files
* Run this command
* **sudo apt-get install libpam-cracklib -y**

## su

* **sudo gedit /etc/pam.d/su**
* “Su” can start a shell as root.
* Uncomment “auth required pam\_wheel.so” This will force users to be in the group root before using su.
* SAVE

## Common-auth

* **sudo gedit /etc/pam.d/common-auth**
* Add this to the end of the file
  + auth optional pam\_tally.so deny=5 onerr=fail unlock\_time=900 audit even\_deny\_root\_account silent
* This locks out the user for 15m after 5 failed attempts
* SAVE

## Common-password

* **sudo gedit /etc/pam.d/common-password**
* Add this to the end of the file.
  + password requisite pam\_cracklib.so retry=3 minlen=8 difok=3 ucredit=1 lcredit=1 dcredit=1 ocredit=1
  + password requisite pam\_pwhistory.so use\_authtok remember=24 enforce\_for\_root
* This enforces password complexity and a password history
* SAVE
* Make sure this doesn’t screw with your passwords

# /etc/sysctl.conf

* Networking config for the system
* **sudo gedit /etc/sysctl.conf**
* **echo 2 | sudo tee /proc/sys/kernel/randomize\_va\_space**
* IPv6 (Disable unless needed)
  + Add the following lines to the bottom of the file
  + net.ipv6.conf.all.disable\_ipv6 = 1
  + net.ipv6.conf.default.disable\_ipv6 = 1
  + net.ipv6.conf.lo.disable\_ipv6 = 1
  + **sudo sysctl -p** (Reloads config)
* IPv4 security
  + kernel.dmesg\_restrict = 1
  + kernel.exec-shield = 1
  + kernel.sysrq = 0
  + kernel.randomize\_va\_space = 1
  + net.ipv6.conf.lo.disable\_ipv6 = 1
  + net.ipv6.conf.default.disable\_ipv6 = 1
  + net.ipv6.conf.all.disable\_ipv6 = 1
  + net.ipv4.ip\_forward = 0
  + net.ipv4.tcp\_rfc1337 = 1
  + kernel.unprivileged\_userns\_clone = 0
  + net.ipv4.conf.default.accept\_source\_route = 0
  + net.ipv4.tcp\_syncookies = 1
  + net.ipv4.conf.all.send\_redirects = 0
  + net.ipv4.conf.default.send\_redirects = 0
  + net.ipv4.conf.all.accept\_redirects = 0
  + net.ipv4.conf.all.secure\_redirects = 0
  + net.ipv4.conf.all.accept\_source\_route = 0
  + net.ipv4.conf.all.log\_martians = 1
  + net.ipv4.conf.default.accept\_source\_route = 0
  + net.ipv4.conf.default.accept\_redirects = 0
  + net.ipv4.conf.default.secure\_redirects = 0
  + net.ipv4.icmp\_echo\_ignore\_broadcasts = 1
  + net.ipv4.conf.all.rp\_filter = 1
  + net.ipv4.conf.default.rp\_filter = 1
  + net.ipv4.tcp\_max\_syn\_backlog = 2048
  + net.ipv4.tcp\_synack\_retries = 2
  + net.ipv4.tcp\_syn\_retries = 5
  + net.ipv4.icmp\_echo\_ignore\_broadcasts = 1
  + net.ipv4.conf.eth0.rp\_filter = 1
  + net.ipv4.conf.lo.accept\_redirects = 0
  + net.ipv4.conf.eth0.accept\_redirects = 0
  + net.ipv4.conf.lo.log\_martians = 1
  + net.ipv4.conf.eth0.log\_martians = 1
* **sudo sysctl -p**

Easy copy & paste

kernel.dmesg\_restrict = 1

kernel.exec-shield = 1

kernel.sysrq = 0

kernel.randomize\_va\_space = 1

net.ipv6.conf.lo.disable\_ipv6 = 1

net.ipv6.conf.default.disable\_ipv6 = 1

net.ipv6.conf.all.disable\_ipv6 = 1

net.ipv4.ip\_forward = 0

net.ipv4.tcp\_rfc1337 = 1

kernel.unprivileged\_userns\_clone = 0

net.ipv4.conf.default.accept\_source\_route = 0

net.ipv4.tcp\_syncookies = 1

net.ipv4.conf.all.send\_redirects = 0

net.ipv4.conf.default.send\_redirects = 0

net.ipv4.conf.all.accept\_redirects = 0

net.ipv4.conf.all.secure\_redirects = 0

net.ipv4.conf.all.accept\_source\_route = 0

net.ipv4.conf.all.log\_martians = 1

net.ipv4.conf.default.accept\_source\_route = 0

net.ipv4.conf.default.accept\_redirects = 0

net.ipv4.conf.default.secure\_redirects = 0

net.ipv4.icmp\_echo\_ignore\_broadcasts = 1

net.ipv4.conf.all.rp\_filter = 1

net.ipv4.conf.default.rp\_filter = 1

net.ipv4.tcp\_max\_syn\_backlog = 2048

net.ipv4.tcp\_synack\_retries = 2

net.ipv4.tcp\_syn\_retries = 5

net.ipv4.icmp\_echo\_ignore\_broadcasts = 1

net.ipv4.conf.eth0.rp\_filter = 1

net.ipv4.conf.lo.accept\_redirects = 0

net.ipv4.conf.eth0.accept\_redirects = 0

net.ipv4.conf.lo.log\_martians = 1

net.ipv4.conf.eth0.log\_martians = 1

# /etc/sudoers and /etc/sudoers.d

* These files set the rules required to use sudo
* This is hard to understand to buckle up
* **sudo gedit /etc/sudoers**
  + So ALL means that the specified user/group can use ALL of the commands
  + User specifies a username
  + # targets specific UIDs
  + % targets GROUPS
  + %# targets GIDs
  + (ALL) lets the target run commands as any USER
  + (ALL:ALL) lets the target run commands as any user or group
  + Overall, make sure no unauthorized users have any special perms
  + E.G. [bad user] ALL=(ALL) ALL
  + E.G. [bad group] ALL=(root:admin) ALL
  + E.G. CaliPatriot ALL=(root:admin) ALL

# Audit Policies

* **sudo apt-get install auditd -y**
* **sudo auditctl -e 1**
* **sudo gedit /etc/audit/auditd.conf**
  + Just check for anything that looks wrong

# Rootkits

* RKhunter
  + **sudo apt-get install rkhunter -y** (No configuration)
  + **sudo rkhunter --checkall**
  + To check warnings run **sudo cat /var/log/rkhunter.log**
* CHKRootkit
  + **sudo apt-get install chkrootkit**
  + **sudo chkrootkit -q**
  + Google anything you don’t know

# Malicious Packages

* Run this command for all of the following terms
* **sudo dpkg -l | grep [package]**
  + John, server, sql, apache, web, netcat, nc, php, cain, lightweight, password, cracker, game, ssh, vsftpd, port, nmap, metasploit, xinetd, irc (telepathy is fine), dns (purge bind9 not dnsmasq).
  + Look at each item that shows up closely and if you’re not sure what it is, Google it.
  + If it is bad, run **sudo apt-get purge [package]**
* Also look through the sidebar and delete non essential programs.

# Startup Files

* Ubuntu has a number of startup files, so check for bad code
* If you think you spot some bad code, copy and paste it into a text file, gedit and delete the line.
* ~ Means user specific. Check root
* /etc/profile
  + **sudo cat /etc/profile**
  + Ideally there should be some code, but nothing malicious.
* /etc/profile.d
  + **cd /etc/profile.d**
  + **sudo cat [filename]**
  + Check for bad code
* /etc/bash.bashrc
  + **sudo cat /etc/bash.bashrc**
  + Once again, there should be some code, but nothing malicious.
* ~/.profile
  + If you want to check if this is identical to default, just run **sudo cmp --silent /home/[username]/.profile /etc/skel/.profile && echo “Identical” || echo “Not Identical”**
  + If it’s not identical, gedit, note, and delete.
* ~/.bashrc
  + If you want to check if this is identical to default, just run **sudo cmp --silent /home/[Username]/.bashrc /etc/skel/.bashrc && echo “Identical” || echo “Not Identical”**
  + Same process to delet
* ~/.bash\_logout
  + If you want to check if this is identical to default, just run **sudo cmp --silent /home/[Username]/.bash\_logout /etc/skel/.bash\_logout && echo “Identical” || echo “Not Identical”**
  + Same process to delet
* /etc/rc.local
  + **sudo cat /etc/rc.local**
  + This should be EMPTY (besides the hashed out things.)
* /etc/rc[0-6].d
  + Go through each one and clean them out. There are 6.
  + E.G. **cd /etc/rc1.d**
  + You are mainly looking for bash and python scripts
  + **ls | grep -r $.sh’, ‘ls | grep -v ^l** This looks for bash files
  + **ls | grep -r $.py’, ‘ls | grep -v ^l** This looks for Python files

# Secure Ports

* **sudo ss -ln**
* If a port has 127.0.0.1:$port in its line, that means it's connected to loopback and isn't exposed. Otherwise, there should only be ports which are specified in the readme open (but there probably will be tons more).
* sudo lsof -i -P -n | grep **LISTEN**
* For each open port which should be closed:
  + **sudo lsof -i [port]**
  + Copy the program which is listening on the port. **whereis [program]**
  + Copy where the program is (if there is more than one location, just copy the first one). **dpkg -S [location]**
  + This shows which package provides the file (If there is no package, that means you can probably delete it with **rm [program]**; **killall -9 [program]**). **sudo apt-get purge [package]**
  + sudo **ss -l** to make sure the port actually closed.

# Services

* **sudo apt-get install bum -y**
* **sudo bum**
* Use common sense.

# FireFox

* Hamburger -> Preferences
* Check “Always check if Firefox is your default browser”
* Check “Automatically update search engines”
* Uncheck “Recommend extensions as you browse
* Click “Settings” under “Network Settings”
  + Click “Auto-detect proxy settings for this network”
  + If you lose points try the other one.
* Click the padlock of the side
* Check “Trackers”
  + Check “Always”
* Check “Third-Party Cookies”
  + Check “All third-party cookies”
* “Send websites a “Do Not Track “ signal…
  + Check “Always”
* Check “Block cookies and site data”
  + Select “All cookies”
* Uncheck “Ask to save logins and passwords for websites"
* “Firefox will Never remember history”
* Uncheck “Browsing history”
* Uncheck “Bookmarks”
* Uncheck “Open tabs”
* Check “Block pop-up windows”
* Check “Warn you when website try to install add-ons”
* Check “Prevent accessibility services from accessing your browser”
* Uncheck “Allow Firefox to send technical …”
* Uncheck “Allow Firefox to send backlogged crash …”
* Check all three under security
* Check “Ask you every time” under certificates
* Check “Query OSCP responder server ..”

# Lightdm.conf

* **sudo gedit /etc/lightdm/lightdm.conf**
* Add the following lines
  + allow-guest=false
  + greeter-hide-users=true
  + greeter-show-manual-login=true
  + autologin-user=none
  + xserver-allow-tcp=false
* **sudo restart lightdm**

# Hosts File

* **sudo cat /etc/hosts | grep -v ^# | grep -v -e ‘^[[:space:]]\*$’**
* Should only be 7 lines

# Grub

* Run the following commands
* **sudo apt-get install grub-common -y**
* **sudo chmod 777 /etc/grub.d/40\_custom**
* **sudo echo "set superusers=\"root\"" >> /etc/grub.d/40\_custom**
* **sudo echo "password\_pbkdf2 root grub.pbkdf2.sha512.10000.0873A9B662EA5F41DFBA2C22736E6F304608B260D86A2C16C4CA80E37EE015AE1DAE7A6C8CB7BAE313D029A85B013397DA9ADF9975A9B3B4B145389E498828F1.C43AFA155C7B24A846EABF640D25A644BF046E9E81D8F73BE0336AD10A93AB0DAC0C94F4A94A31AD024BE99B764128622E3E2E20E7CA633DF2816D38334145F6" >> /etc/grub.d/40\_custom**
* **sudo chmod 640 /etc/grub.d/40\_custom**
* **sudo update-grub**
* After you do this, when you restart you use root and IH@t3Pupp135

# $PATH

* The ‘PATH’ variable contains the locations that the shell checks for programs.
* If a directory is in there that shouldn’t be, it could mean that there’s a malicious exec on your system that you’re not seeing.
* To get the variable, run **echo $PATH**
* Defaults are ‘/usr/local/sbin’,’/usr/local/bin’,’/usr/sbin’, ‘/sbin’ and ‘/bin’.
* If there is something appended to the variable, run ‘grep -r [thing]’ to find where it’s located. Purging is up to discretion. If in doubt, comment, don’t delete.

# Shared Memory

* **sudo gedit /etc/fstab**
* Add this to the bottom of the file
  + tmpfs /run/shm tmpfs defaults,noexec,nosuid 0 0

# PHP Backdoors

* On any environment where there is a PHP server running OR you purged an inappropriate PHP server, it is possible that there are PHP backdoors.
* These will always be in a subdirectory under your webserver – ‘/var/www/’.
* If not found there, run ‘**sudo grep -r / php**’, then run ‘**sudo find / -iname [filename]**’.
* To get a potential list of backdoors, run:
* **grep '((eval.\*(base64\_decode|gzinflate|\$\_))|\$[0O]{4,}|FilesMan|JGF1dGhfc|IIIl|die\(PHP\_OS|posix\_getpwuid|Array\(base64\_decode|document\.write\("\\u00|sh(3(ll|11)))' . -lroE --include=\*.php\***
* Make sure to be in the directory where all the files are.
* If you find one, copy it: ‘cp [filename] /home/[your username]/[filename]’.
* Then delete. If you lose points, just run the command in reverse.
* Great post on the topic: https://djlab.com/2010/09/finding-php-shell-scripts-and-php-exploits/

# Samba

* **sudo gedit /etc/samba/smb.conf**
* Remove any bad shares
* Add the following to [Global]
  + lanman auth = no
  + ntlm auth = yes
  + client lanman auth = no
  + smb encrypt = mandatory
  + min protocol = SMB2
* In the shares make sure that guest is disabled and is not world writable

# Vsftpd (FTP)

* FTP is used to transfer files between computers
* **sudo gedit /etc/vsftpd.conf**
  + Set/add anonymous\_enable no
  + Uncomment line starting with chroot\_local\_user
    - Change line from no to yes if necessary
  + Set chroot\_list\_enable to yes
  + Uncomment line starting with write\_enable
    - Change line from yes to no if necessary
  + Add line max\_clients 30
  + Set local\_enable=YES
  + Set write\_enable=YES
  + Set chroot\_local\_user=YES
  + **sudo ufw allow 21**
  + **sudo restart vsftpd**
* Put a list of users in "/etc/vsftpd.chroot\_list", one per line, that should be jailed to their home directories. So put down users, not admins.
* Disable root login
  + **sudo echo root >> /etc/ftpusers**

# SSH (OpenSSH)

* **sudo gedit /etc/ssh/sshd\_config**
  + Set/add UsePAM yes
  + Set/add AllowTcpForwarding no
  + Set/add X11Forwarding no
  + Set/add LoginGraceTime 30
  + Add line ClientAliveInterval 300
  + Add line ClientAliveCountMax 0
  + Make sure that the "Protocol" line is set to "Protocol 2".
  + Set/add HostBasedAuthentication no
  + Set/add IgnoreRhosts yes
  + Set/add PermitEmptyPasswords no
  + Set/add StrictModes yes
  + Set/add UsePrivilegeSeparation yes
  + Set/add PermitRootLogin no
  + Set/add PrintLastLog no
  + Set/add DebianBanner no
  + Set/add MaxAuthTries 3
  + Set/add PasswordAuthentication no
  + Set/Add AllowGroups $SSH\_GRPS
  + Set/Add LogLevel VERBOSE
  + Uncomment the line that starts with "Banner".
    - **sudo echo “Sugma Secure Server” > /etc/issue.net**
  + Create a list of allowed users (Depends on readme): add line "AllowUsers [User1], [User 2] …".
  + Save config and exit nano.
  + **sudo /etc/init.d/ssh restart**

# PHP

* **sudo gedit /etc/php5/apache2/php.ini**
  + Add the following lines
  + disable\_functions = exec,system,shell\_exec,passthru
  + register\_globals = Off
  + expose\_php = Off
  + display\_errors = Off
  + track\_errors = Off
  + html\_errors = Off
  + magic\_quotes\_gpc = Off
  + mail.add\_x\_header = Off
* **sudo service apache2 restart**

disable\_functions = exec,system,shell\_exec,passthru

register\_globals = Off

expose\_php = Off

display\_errors = Off

track\_errors = Off

html\_errors = Off

magic\_quotes\_gpc = Off

mail.add\_x\_header = Off

# Apache2

* The document root (where the .html files or others are stored) is located at "/var/www/html" or "/var/www".
* Access log at "/var/log/httpd/access\_log".
* Configuration options:
  + **sudo gedit /etc/apache2/conf-enabled/security.conf**
    - Add/edit line ServerSignature Off
    - Add/edit line ServerTokens Prod
    - Add/edit line TraceEnable Off
    - Add/edit line Header unset ETag
    - Add/edit line Header always unset X-Powered-By
    - Add/edit line FileETag None
  + **sudo gedit /etc/apache2/apache2.conf**
    - Add/edit line Options -FollowSymLinks
    - Add/edit line AllowOverride none
    - Add/edit line Require all granted
    - Add/edit line Options -FollowSymLinks -Includess -ExecCGI
    - Add/edit line LimitRequestBody 204800
  + **sudo gedit /etc/apache2/mods-available/ssl.conf**
    - Edit SSLProtocol all -SSLv3” to “SSLProtocol all -SSLv2 -SSLv3
  + **sudo service apache2 restart**

# Bind9 (Open DNS Recursion)

* **sudo gedit /etc/bind/named.conf.options**
  + Add recursion no;
  + Add version "Not Disclosed";
* **sudo chown -R root:bind /etc/bind**
* **sudo chown root:bind /etc/bind/named.conf\***
* **sudo chmod 640 /etc/bind/named.conf\***
* **sudo gedit /etc/bind/named.conf.options**
  + Add version "Secured DNS server";
  + Add allow-recursion { localhost; 192.168.0.0/24; };
  + Add allow-query { localhost; 192.168.0.0/24; };
  + Add server-id none;
  + Add allow-transfer { 192.168.1.1; };
  + Add listen-on port 53 { 127.0.0.1; 192.168.1.1; };
* **sudo service bind9 restart**

# Low-Graphics Mode Boot

* **sudo rm /etc/lightdm/lightdm.conf  
  sudo ucf -p /etc/lightdm/lightdm.conf  
  sudo apt-get install --reinstall lightdm**

**DO MySQL https://www.digitalocean.com/community/tutorials/how-to-secure-mysql-and-mariadb-databases-in-a-linux-vps**