Fedora first fifteen (IP-172.20.241.40)

v3u4 – Andrew

ccdc.cit.morainevalley.edu

<https://github.com/Lovatoty/Linux_Scripts>

Fedora scored services: **Email , \_\_­\_\_\_\_\_\_\_\_ , \_\_­\_­­­­­\_\_\_\_\_\_\_ , \_\_\_\_\_\_\_\_\_\_**

yum install vim

yum install aide tmux iptables-services

1. Login as Root: with given password or use sudo passwd to make a new one
2. Change root password: !Password123 passwd

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. Log in as root to email and database to change passwords
2. Set passwords for mysql (make sure to get password information)
   1. **vim /var/www/html/config/db.inc.php**
   2. $rcmail\_config[‘db\_dsnw’] = ‘mysql://round:Password01@172.20.240.23/roundcube’; [something like this]
3. ADMINISTRATOR is NOT is sudoers to add administrator type $visudo
   1. Then add “administrator ALL=(ALL) ALL”
4. Deal with Password.txt
5. Stop firewalld: service firewalld stop; service firewalld disable; iptables -nL
6. If iptables is not installed: yum install iptables-services;
7. Enable iptables: service iptables start; service iptables enable; service ip6tables stop,

**iptables -A INPUT -i lo -j ACCEPT**

587: SMTP  
995: pop3s  
993: imaps

110: pop (unsecure)  
143: imap (unsecure)

service iptables save, service ip6tables save

1. Set up firewall rules (allow only needed conections) attached are some basic iptables commands
2. Stop ssh: sudo systemctl stop sshd.service, sudo systemctl disable sshd.service;
3. Remove bash from users that shouldn’t have it: usermod -s /sbin/nologin ‘user\_name’
4. Configure AIDE- yum install aide
5. Initialize AIDE aide –init, mv /var/lib/aide/aide.db.new.gz /var/lib/aide/aide.db.gz
6. Shutdown cockpit
7. Get rid of password.txt files and other random scripts!
8. Set important files to immutable
9. Disable ssh, “service sshd stop” and then “chkconfig sshd off” + “chkconfig sshd –del
10. In /etc/ssh/sshd\_config: disable empty passwords, change to protocol 2, add limited login grace time, ignore Rhosts, allow only previously made user to ssh into server (add to bottom: AllowUsers “name”)
11. Look at /var/www/html/config/db.inc.php for mysql password connections
12. Get updates: yum update
13. Make sure SElinux is running: sestatus; and if it is enforcing: getenforce;
14. “Chattr +i” all important files that shouldn’t change **chattr +i /etc/passwd /etc/shadow**
15. Remove admin panel for webserver in centos,

“ rm -rf var/www/html/admin(I think)”

1. Make backups of important files
   1. tar cvzpf etcback.tar.gz /etc
   2. tar cvzpf varwwwbackup.tar.gz  /var/www
2. Make sql ssl encrypted
3. Redownload PAM from yum repos
4. Check remote access by systemd

**Additional stuff listening**

**Dovecot admin manual of r info on users and db access and password db access**

Check which ports are listening with lsof –i or ss -tulpn

Monitor connected users using “who;”

Add limited password attempts [[1]](#footnote-1)

Check for immutable files (to set immutable: chattr +i “Filename”; lsattr to list file attrib; to remove: chattr -I “filename”)

Use “dhclient” or “yum clean all” to fix yum if broken

Df -ha to see disk space left

Create a sudo user useradd “name”; passwd “name” \_\_\_\_\_\_\_ ; usermod -aG wheel “name”; (test with: groups “name”;)

Team for detailed splunk forward for the **linux** boys

1. apt-get install --only-upgrade openssl libssl0.0 wget
2. wget <https://splk.it/2Snuxcs> **Change this url to fit your linux distro**
3. dpkg -I **package name**
4. cd /opt/forwarders/bin or /opt/splunkforwarder/bin
5. ./splunk start
   1. Press q  and y
   2. Enter username: admin
   3. Enter password: \_\_\_\_\_\_\_\_\_\_\_\_
6. ./splunk enable boot-start
7. ./splunk add forward-server 172.20.241.20.:9997
8. ./splunk add monitor /var/log
   1. /var/kern.log
   2. /var/off.log
   3. /var/boot.log
   4. /var/sys.log
   5. /var/wtmp

splunk forwarders:

* Rpm <https://splk.it/2GZyDlN>
* Deb <https://splk.it/2vO6RWP>
* Windows <https://splk.it/2UsyrDr>

<https://teams.microsoft.com/l/message/19:30439ce2b054473c874d9993a39fe7d4@thread.skype/1605929073787?tenantId=5011c7c6-0ab4-46ab-9ef4-fae74a921a7f&amp;groupId=c2f1812e-d046-40fd-a729-2ba3117c4e45&amp;parentMessageId=1605929073787&amp;teamName=SCSU-CCDC-Team&amp;channelName=General&amp;createdTime=1605929073787>

Iptables input -I lo -j accept

1. **iptables -P INPUT ACCEPT** If connecting remotely we must first temporarily set the default policy on the INPUT chain to ACCEPT otherwise once we flush the current rules we will be locked out of our server.
2. **iptables -F** We used the -F switch to flush all existing rules so we start with a clean state from which to add new rules.
3. **iptables -A INPUT -i lo -j ACCEPT** Now it's time to start adding some rules. We use the -A switch to append (or add) a rule to a specific chain, the INPUT chain in this instance. Then we use the -i switch (for interface) to specify packets matching or destined for the lo (localhost, 127.0.0.1) interface and finally -j (jump) to the target action for packets matching the rule - in this case ACCEPT. So this rule will allow all incoming packets destined for the localhost interface to be accepted. This is generally required as many software applications expect to be able to communicate with the localhost adaptor.
4. **iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT** This is the rule that does most of the work, and again we are adding (-A) it to the INPUT chain. Here we're using the -m switch to load a module (state). The state module is able to examine the state of a packet and determine if it is NEW, ESTABLISHED or RELATED. NEW refers to incoming packets that are new incoming connections that weren't initiated by the host system. ESTABLISHED and RELATED refers to incoming packets that are part of an already established connection or related to and already established connection.
5. **iptables -A INPUT -p tcp --dport 80 -j ACCEPT** Here we add a rule allowing SSH connections over tcp port 22. This is to prevent accidental lockouts when working on remote systems over an SSH connection. We will explain this rule in more detail later.
6. **iptables -P INPUT DROP** The -P switch sets the default policy on the specified chain. So now we can set the default policy on the INPUT chain to DROP. This means that if an incoming packet does not match one of the following rules it will be dropped. If we were connecting remotely via SSH and had not added the rule above, we would have just locked ourself out of the system at this point.
7. **iptables -P FORWARD DROP** Similarly, here we've set the default policy on the FORWARD chain to DROP as we're not using our computer as a router so there should not be any packets passing through our computer.
8. **iptables -P OUTPUT ACCEPT** and finally, we've set the default policy on the OUTPUT chain to ACCEPT as we want to allow all outgoing traffic (as we trust our users).
9. **iptables -L -v** Finally, we can list (-L) the rules we've just added to check they've been loaded correctly.

Github scripts

MAIN:

|  |
| --- |
| sudo yum install vim aide tmux iptables ntpdate -y |
|  |

|  |
| --- |
| sudo yum install iptables-services -y |
|  |

|  |
| --- |
| aide --init |
|  |

|  |
| --- |
| mv /var/lib/aide/aide.db.new.gz /var/lib/aide/aide.db.gz |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| touch /var/fedoscriptlog.txt |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| #setup firewall |
|  |

|  |
| --- |
| systemctl stop firewalld |
|  |

|  |
| --- |
| systemctl disable firewalld |
|  |

|  |
| --- |
| systemctl start iptables |
|  |

|  |
| --- |
| systemctl enable iptables |
|  |

|  |
| --- |
| systemctl stop ip6tables |
|  |

|  |
| --- |
| systemctl disable ip6tables |
|  |

|  |
| --- |
| **#firewall rule chain** |
|  |

|  |
| --- |
| **iptables -P INPUT ACCEPT** |
|  |

|  |
| --- |
| **iptables -F** |
|  |

|  |
| --- |
| **iptables -A INPUT -i lo -j ACCEPT** |
|  |

|  |
| --- |
| **iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT** |
|  |

|  |
| --- |
| **iptables -A INPUT -p tcp --dport 110 -j ACCEPT** |
|  |

|  |
| --- |
| **iptables -A INPUT -p tcp --dport 143 -j ACCEPT** |
|  |

|  |
| --- |
| **iptables -A INPUT -p udp --dport 123 -j ACCEPT** |
|  |

|  |
| --- |
| **iptables -A INPUT -p tcp --dport 123 -j ACCEPT** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **iptables -P INPUT DROP** |
|  |

|  |
| --- |
| **iptables -P FORWARD DROP** |
|  |

|  |
| --- |
| **iptables -P OUTPUT ACCEPT** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| #stop ssh |
|  |

|  |
| --- |
| systemctl stop sshd.service |
|  |

|  |
| --- |
| systemctl disable sshd.service |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| rm /etc/issue |
|  |

|  |
| --- |
| echo -e "WARNING! Access to this device is restricted to those individuals with specific |
|  |

|  |
| --- |
| Permissions. If you are not an authorized user, disconnect now. |
|  |

|  |
| --- |
| Any attempts to gain unauthorized access will be prosecuted to |
|  |

the fullest extent of the law" >> /etc/issue

/sbin/iptables-save > /etc/sysconfig/iptables

Sestatus

Getenforce

CLEAR IP tables

|  |
| --- |
| iptables -F |
|  |

|  |
| --- |
| iptables -P INPUT ACCEPT |
|  |

|  |
| --- |
| iptables -P OUTPUT ACCEPT |
|  |

iptables -P FORWARD ACCEPT

<https://wiki.centos.org/HowTos/Network/IPTables>

1. <https://serverfault.com/questions/275669/ssh-sshd-how-do-i-set-max-login-attempts>

   **Another good source** à https://firewallingit.blogspot.com/2015/04/ccdc-debian-hardening-guide.html [↑](#footnote-ref-1)