

File System Security Checks

There are certain files whose presence in the Linux file system can present a security risk and should be remedied as soon as possible.

When the SUID (set user ID) or SGID (set group ID) bits are set on an executable, that program executes with the UID or GID of owner of the file, as opposed to the user executing it. This means that all executables with SUID bit set and are owned by root are executed with the UID of root. This situation is a security risk and should be minimized unless the program is designed for this risk.

To find all files on your file system that have the SUID or SGID bit set, execute:

```
# find / -path /proc -prune -o -type f -perm +6000 -ls
```

It is good practice to generate a list of SUID or SGID files on your server as soon as possible, and re-run the above command on a regular basis to ensure new binaries with unsafe permissions are not being added to your server.

World-writable files are a security risk as well. World-writable files and directories are dangerous since it allows anyone to modify them. World-writable directories allow anyone to add or delete files.

To find all world-writable files and directories, execute:

```
# find / -path /proc -prune -o -perm -2 ! -type 1 -ls
```

Another file permission issue are files not owned by any user or group. While this is not technically a security vulnerability, an audited system should not contain any unowned files. This is to prevent the situation where a new user is assigned a previous user's UID, so now the previous owner's files, if any, are all owned by the new user.

To find all files that are not owned by any user or group, execute:

```
# find / -path /proc -prune -o -nouser -o -nogroup
```

Network Security

To get a list of listening network ports, run the following:

```
# netstat -tulp
```

Disable any ports that are not necessary. To do so, kill the PID shown by netstat. The only port that your server must be listening on is SSH (22/tcp). Other ports that will need to be listening depend upon the specific purpose of your dedicated server. Note that by killing the PID of the process you are not preventing your server from starting the same service again on bootup. To disable services, see below.

In order to see what programs your server is launching on startup, execute the following:

```
# chkconfig --list |grep on (Red hat systems)
```

```
# ls -l /etc/rc2.d/S* | cut -d/ -f6 (Debian systems)
```

This command will show you which programs are to be executed in which run levels. In Red hat, full multiuser mode is 3. **To disable a service permanently, issue the following:**

```
# chkconfig <service_name> off
```

To disable any service in Debian, simply execute the following:

```
# rm -f /etc/rc2.d/S*<service_name>
```

Please note that the above commands do not actually disable the service, they simply prevent the service from being executed on startup.

User Security

The first thing you should take stock of are the users with unlocked accounts. Users with unlocked accounts are allowed to login if assigned a valid shell, and should be kept to a minimum.

To get a list of unlocked users, execute the following:

```
# egrep -v '.*:\*|:!' /etc/shadow|awk -F: '{print $1}'
```

If you do not recognize any user returned by the above command, check to see if that user owns any files by executing:

```
# find / -path /proc -prune -o -user <user_name> -ls
```

If the user does not own any files, or files that will not hinder the stability of your server, delete the user by executing:

```
# userdel -r <user_name>
```

TCP/IP Hardening

All of the following lines and values should be added to the file `/etc/sysctl.conf` if you want to enable or disable the feature mentioned. You will need to restart your system for these changes to take effect.

TCP SYN Cookie Protection `net.ipv4.tcp_syncookies = 1`

Disable IP Source Routing `net.ipv4.conf.all.accept_source_route = 0`

Disable ICMP Redirect Acceptance `net.ipv4.conf.all.accept_redirects = 0`

IP Spoofing Protection `net.ipv4.conf.all.rp_filter = 1`

Ignoring Broadcasts Request `net.ipv4.icmp_echo_ignore_broadcasts=1`

Bad Error Message Protection `net.ipv4.icmp_ignore_bogus_error_responses = 1`

System Security

One of the most important things you can do to protect your server is implementing very basic access control. Access control can eliminate a majority of the risk involved in running out of date services on the Internet.

In order to implement an effective access control policy on your dedicated server, you will need the following pieces of information:

- The IP address or addresses of your Internet connection. For some, this may be one static address, while for others it is a pool of addresses. If you have more than one Internet connection, please be sure to get ALL the IP addresses you could be assigned at any time. You may need to contact your Internet Service Provider for this information.

SSH

While we do not recommend anybody running outdated software, especially something as crucial as SSH, a not insignificant portion of the risks involved in running an outdated SSH server can be mitigated by only allowing certain IP networks to access your SSH server.

```
# $IPTABLES -A INPUT -p tcp -dport 22 -s X.X.X/NN -j ACCEPT
```

The above line will allow TCP packets destined for port 22 to be accepted if and only if the source of the packets are within the network denoted in X.X.X.X/NN. If you have more than one Internet connection, or have multiple networks, simply add another line, replacing X.X.X.X/NN with the proper values.

Control Panel Software

If your server is running a control panel, you can also improve your security by implementing an access control policy on the control panel administrative port.

Plesk: `$IPTABLES -A INPUT -p tcp -dport 8443 -s X.X.X.X/NN -j ACCEPT`

Ensim: `$IPTABLES -A INPUT -p tcp -dport 19638 -s X.X.X.X/NN -j ACCEPT`

Cpanel: `$IPTABLES -A INPUT -p tcp -dport 2082 -s X.X.X.X/NN -j ACCEPT`

FTP Server

Another service you may want to implement an access control policy on is FTP. If you, or a small handful of people are the only allowed users to FTP into your dedicated server, then you will certainly benefit.

```
$IPTABLES -A INPUT -p tcp -s X.X.X.X/NN -dport 20 -syn -j ACCEPT
```

```
$IPTABLES -A INPUT -p tcp -s X.X.X.X/NN -dport 21 -syn -j ACCEPT
```

Note that both of the above lines must be executed for each source network.