DRAFT

IT341 Introduction to System Administration
Project VIII – Backing Up Files with **`rsync`**

Backup is one of the most important things a system administrator does. It is important to decide what data on your network is important, and to back that data up on a regular basis. Preferably, the backup process is automated. It's not a matter of *if* you will have to go to a backed-up archive for restoring a corrupted directory or file system, but *when*. As the bumper sticker says: *Things (sic) happen*.

A nice list of ten open-source backup programs may be found at http://www.techrepublic.com/blog/10-things/10-outstanding-linux-backup-utilities/. One of these is **`rsync`**. What's nice about **`rsync`** is that it is a command-line program and so can be invoked from with other scripts, providing for automation. **`rsync`** can be invoked directly or it can be set up as a daemon so as to keep a backed-up archive in sync with the working copy. **`rsync`** is one of the most widely used Linux backup tools.

Most of the work we have been doing involves **`/etc`** and perhaps **`/home.itvm2x-yz`** on each host; **`/etc`** seems to be the most important. Once we have decided what data is important to us, we must decide *where* to back it up to. There are several possibilities:

1. Somewhere else on the same host (like another disk), under the assumption that it's unlikely that two disks will fail at once.
2. On a disk on another host, under the assumption that it is even less likely that two disks on two hosts will fail at the same time.
3. On a DVD or a tape; these can be stored elsewhere, off site (protecting us from fire). The downside is that we require additional human intervention.
4. (More recently) In the cloud, where they will likely be further backed up ***and*** accessible from different locations. This, however, requires that you be able to trust your cloud provider and their security measures.

For our little network, we will go for the ***second*** option. It is unlikely that two disks on two hosts will fail at the same time. If there is a major fire, data backup is the least of our problems in this short semester. Also, the backup process can be automated (say, using a **`cron`** process for scheduling backups).

Invoking **`rsync`** Directly

1. For the server and each client, we must choose a location to use for the backup files. We will use the **`/tmp/backup`** directory tree on **`it20`** for a location. *Each team* will create a subdirectory with the name of the virtual machine and store the backup in a subdirectory named for the date of the backup. For example on April 17, 2012 we will name the subdirectory 04172012. The fully qualified directory would be ***`/tmp/backup/itvm2x-yz/04172012`***

2. For example, say we are backing **itvm28-2b** to **it20**; again, this is just an example. We must first create a location on **it20**. We log onto **it20** as ourselves and change our working directory to **/tmp/backup/itvm28-2b** (See me so that I can make this directory for you and/or give you the necessary permissions.) Execute a **pwd** command, to ensure you are in the correct location, and make sure you are able to create files/subdirectories inside there.

3. So we exit, returning to **itvm28-2b**, and execute a **rsync** command. Pay particular attention to who you are logged *into your VM* as versus who you are logging *into it20* as! (Note that some commands may be too long for this page and wrap to the next line):

```
sysadmin@itvm28-2b:~$ sudo rsync -azvv -e ssh /etc
  abird@it20.it.cs.umb.edu:/tmp/backup/itvm28-2b/04172012

opening connection using: ssh -l abird it20.it.cs.umb.edu
rsync --server -vvlogDtprze.iLsf . /tmp/backup/itvm28-
2b/04172012
abird@it20.it.cs.umb.edu's password:
sending incremental file list
created directory itvm28-2b/04172012
delta-transmission enabled
etc/
etc/.pwd.lock
etc/adduser.conf
etc/at.deny
etc/auto.home
etc/auto.home.bak

…
<lots more files>
…
etc/xml/
etc/xml/catalog
etc/xml/catalog.old
etc/xml/xml-core.xml
etc/xml/xml-core.xml.old
total: matches=0 hash_hits=0 false_alarms=0 data=1840375

sent 742037 bytes received 17334 bytes 35319.58 bytes/sec
total size is 1860318 speedup is 2.45
sysadmin@itvm28-2b:~$
```

Who are you logged into your VM as? And…what username did you use to log into it20? What is the reason for this? Write about this in your admin log.

This invocation of **rsync** requires a little explanation:

a. The options **-azv**,
   i. the **a** stands for archive mode and is equivalent to **-rlptgoD**; this calls for
      - a recursive copy
      - copying links as links
      - creating parent directories as necessary
      - preserving times, groups and owners
      - preserving devices.
   ii. the **z** says data should be compressed to save space, and
   iii. the **vv** stands for *very verbose* output; there's **v** for verbose, **vv** for very verbose, and **vvv** for even more verbose.

b. The option, **-e ssh** specifies the remote shell to be used for doing the transfer; **ssh** is safest.
c. The first path is **/etc**, the directory on the local machine ( ) that we want to back up. If it were **/etc/**, then the *contents* of **/etc** would be archived. **What is the difference? Write about this in your admin log?**
d. The second path, **sysadmin@it20.it.cs.umb.edu:/tmp/backup/itvm28-2b/04172012**, is the archive location – where we want to store the backup. We log in as `sysadmin`, so we will be challenged for `sysadmin`'s password. You could log in as yourself, or whatever.

Restoring Files From an Archive

To restore a file system from an archive, we simply run **rsync** backwards, for example (**But don't do this!!!**):

```
sudo rsync -azvv -e ssh abird@it20.it.cs.umb.edu:/tmp/backup/itvm28-2b/04172012 /tmp/etc
```

Were we to run this on **itvm28-2b**, we copy the contents of the archive back into the **/etc** directory on **itvm28-2b**. (Actually, we'd be copying it into **/tmp/etc** – this is safer; we can then locally copy the files into **/etc**.) **Again, DO NOT DO THIS!**

Of course, when acting as a system administrator, we might run such commands as **root** (on both systems), *but we don't want to risk hurting ourselves here*!

When To Do Backups?

The question arises: when should we back up our file systems? It depends. For this course, it might be good to back up each host *once a week* – once each new service (e.g. *NIS*, *NFS*, and *DNS*) has been successfully installed. We can make the process easier (and so more likely that we will do it) if we write a simple ***script***.

For example, we might edit a simple file, **/usr/local/bin/backup**, to contain:

```
#!/bin/bash
#
# backup dir -- takes one argument. dir which names a repository.
rsync -azvv -e ssh /etc abird@it20.it.cs.umb.edu:/tmp/backup/`hostname`/$1
```

Notice that **/usr/local/bin** is in $PATH:

```
abird@itvm28-2b:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
abird@itvm28-2b:~$
```

and so once we make backup executable,

```
abird@itvm28-2b:~$ sudo chmod +x /usr/local/bin/backup
```

we can execute it with an argument to name the archive:

```
abird@itvm28-2b:~$ sudo backup 04172012
[sudo] password for abird:
opening connection using: ssh -l abird it29.it.cs.umb.edu rsync
--server -vvlogDtprze.iLsf . /tmp/backup/itvm28-2b/02142012
abird@it20.it.cs.umb.edu's password:
sending incremental file list
created directory /tmp/backup/itvm28-2b/02142012
delta-transmission enabled
etc/
etc/.pwd.lock
etc/adduser.conf
etc/at.deny
etc/auto.home
etc/auto.home.bak

…
```

Now, if we log on to it20 as abird, we can do an **ls** to see the archive we have just created:

```
abird@it20:~$ ls -l /tmp/backup/itvm28-2b/04172012
total 4
drwxr-xr-x 3 abird abird 4096 2012-04-17 01:02 04172012
abird@it20:~$
```

Backing up file systems manually requires that we *remember* to do so. A better solution might be to *automate* the backup process…