Backup and Change Management

Security Design

System Examination

System Configuration

Firewalls and Filters

Hardening Software

Backups and Change Management

Access Control and Authentication

Virtual Private Networking

Logging and Monitoring

Security Policy and Management Support

SELinux

Linux Systems Security

Backup

- Security breaches can cast doubt on entire installations or render them corrupt
- Files or entire systems may have to be recovered from backup
- Many tools are available to help with this task in Linux
- Two of the more commonly used ones are rsync and duplicity

https://en.wikipedia.org/wiki/Rsync

http://duplicity.nongnu.org/features.html

Legacy Tools

https://en.wikipedia.org/wiki/Manuscript_culture

- cp is the original way to make a copy of files, but assumptions people make cause problems in using it
- GUI-based drag and drop tools make the assumptions problem worse
- cpio, tar are archival tools created to copy files to backup media (tape by default) - satisfactory for years but they make it cumbersome to manage backup media
- Various software packages provide frontends to these tools in order to make backup/restore easier to manage and more robust

Rsync

- Rsync has a command line interface which resembles a smart cp, and provides a base for many backup software packages with GUIs
- Can preserve special files such as links and devices
- Can copy to or from local or remote destinations
- Provides standby device capability by synchronizing data stores

Rsync

- Can compress data in transit and only sends changed data
- Can use ssh to encrypt remote backup transfers without requiring encryption of backup storage
- Light on bandwidth, heavier on cpu and memory
- Requires rsync be available at both ends of the connection if used over the network
- Can be used with network shares to avoid requirement for rsync at both ends of connections at a price in performance and network utilization

Rsync Examples

- Simple local archive of user data

```
rsync -ahv --delete ~/* /backup/username
rsync -ahv --delete --exclude=~/myBackup ~/* ~/myBackup
```

- Local archive with history

```
date=$(date "+%F-%H-%M")
rsync -ahv --link-dest=/backup/latest --delete --exclude={/proc/*,/tmp/*,/run/*,/dev/*,/sys/*,/mnt/
*,/lost+found,/media/*,/backup} /* /backup/$date &&
ln -nsf /backup/$date /backup/latest
```

- Remote archive with history

```
date=$(date "+%F-%H-%M")
rsync -ahv --link-dest=/backup/latest --delete --exclude={/proc/*,/tmp/*,/run/*,/dev/*,/sys/*,/mnt/
*,/lost+found,/media/*,/backup}/* rsync-user@host:/backup/$date &&
ssh rsync-user@host ln -nsf /backup/$date /backup/latest
```

Rsync Restore

- Restoration of user data can be done using rsync command to copy data from backup location
- Can be copied to user staging or live data location
- Since the backup filesystem is a synchronized duplicate of the original filesystem, normal tools can be used for examining what is available in a backup
- System recovery requires additional tasks to be performed to deal with the boot block
  - Rsync the backup to a new drive
  - Install the boot block using GRUB on the new drive
  - Install the new drive and boot from it

Duplicity

- Like rsync but does not require duplicity on the receiving backup host, stores metadata with backup
- Can use many different types of backend storage, including Amazon S3, although they require additional backend packages and configuration effort
- Encrypts tarballs as a storage mechanism, stores deltas in separate tarballs, may have more steps involved in recovery
- Written in python, heavy on bandwidth, lighter on cpu and memory, reimplements ssh in python
- duplicity /what/to/backup sftp://user@host/backup/directory

Other Backup Solutions

- Many GUI frontends to the rsync, duplicity, and tar cli tools
- Backintime, DejaDup, CronoPete, Timeshift, Duplicati
- Bacula, Amanda
- Mondo Rescue for disaster recovery

Change Management

Change Management

- Implementing and maintaining security is a continuous process, these activities cause changes to your systems
- It has been said that problems created when implementing changes cause approximately 3/4 of all outages
- ITIL (Information Technology Information Library) includes change management as part of its service management best practices, ISO 20000 also covers change management
- It involves setting up advisory boards and creating formal processes for implementing changes

ITIL

- Created by the UK government in the 1980s
- A set of documents describing best practices for service management in the digital world
- Originally 30+ reference books, now 5
- Most recently updated in 2019
- Not the only way to manage change but has ties to ISO 20000 which makes it a good candidate for larger organizations concerned about standards for management

ITIL

- Information Technology Information Library

ITIL Vol. 3, Service Transition

- Volume 3 of ITIL includes change management
- Goals include minimal service disruption, back-out activity reduction, and economical resource usage to accomplish change
- 3 types of change might include rollout, normal change, and urgent/emergency change - organization dependent
- https://en.wikipedia.org/wiki/Change_management_(ITSM)

Change Process

- Change requests can be used to document change purpose, justification, impacts, activities, risks, not all parts may be present at the start of the process
- Change advisory board evaluates requests and provides management approval/denial, including funding approval - may also be where activities and risks get defined
- Change activities include tasks to execute, but also describe testing and validation tasks

Testing and Validation

- Ideally changes are validated and tested before deployment in production
- Testing may be done in-house, or delegated
- Security patches are often considered tested when released by vendors
- Testing includes verifying the change achieved the stated goals without unexpected impacts - virtual machines are good ways to do testing and validation