To install Linux with philosophy

작 성 자 : 서 진 우
소　　속 : 클루닉스 기술부 / 부장
작 성 일 : 2004년 4월 20일
Clunix Homepage : http://www.clunix.com
Private Homepage : http://www.sysmng.com

클루닉스 기술부 1차 세미나 자료

철학이 있는 리눅스 설치

작성자 : 클루닉스/기술부 서진우 (alang@clunix.com)
작성일 : 2004년 4월 20일 1차 완성

** 본 문서는 클루닉스 기술부 내부 세미나용으로 본인의 허락 없이는 외부 유출을 금합니다.

주요 내용

==========

1. 리눅스 설치
   - ① 설치 전 고려 사항
     - i. 파티션 정책
     - ii. SCSI Adapter Driver Patch
   - ② 리눅스 설치
     - i. 파티션 정책
     - ii. 패키지 정책
     - iii. 설치 완료 후 기본 확인 작업
   - ③ 설치 완료 후 부가 작업
     - i. 주요 서비스 파티션 구성 확인 및 파일 시스템
     - ii. 시스템 구성 확인

2. 패키지 업데이트
   - ① S/W 패키지 업데이트
   - ② H/W 패키지 업데이트

3. OS 관련 기본 보안 정책 및 보안 패키지 설치
   - ① OS관련 기본 보안 정책
     - i. 데몬 관리 정책
     - ii. 기본 시스템 계정 정책
     - iii. 시스템 퍼미션 정책
     - iv. 일반 사용자 관리 정책
     - v. TCP 보안 정책
   - ② 보안 패키지 설치
     - i. Fcheck 설치 후 관리 하기

4. OS 설치 후 중요 데이터 백업 정책
   - ① 백업 정책 설정 시 고려할 점
   - ② 백업의 방식
   - ③ 2차 및 3차 백업에 대해
     - i. FTP를 이용한 2차 백업
     - ii. RSYNC를 이용한 2차 백업
   - ④ 복구의 방식
   - ⑤ 백업 및 복구 시 주의점
   - ⑥ 실전 백업 예제

5. 문제 발생 시 응급 복구 방법
   - ① 긴급 부팅 시키기
   - ② 부팅 디스켓 만들기
   - ③ 파일 시스템 점검 수리
   - ④ RPM 패키지 상태 초기화 하기

==========

1. 리눅스 설치

① 설치 전 고려 사항

i. 파티션 정책

자동파티션 설치를 선택하면 기본적으로 /boot , / , swap 3개의 파티션으로 자동 활당되어진다. 하지만 정식 서비스를 위해서는 가지고 있는 하드디스크를 계산하여 성능 및 보안등을 고려하여 현 서비스에 가장 적절한 정책을 세워야 한다.

리눅스의 대표 파티션들은 모두 보안적인 측면이나 성능적인 측면에서 의미를 부여하고 있다. 리눅스의 대표적인 파티션을 살펴 보면 다음과 같다.

/boot
/
/etc
/home
/usr
/usr/local
/tmp
/var
swap

각 대표적인 모든 파티션을 다 나누어 설치를 하게 되면 가장 안정적일수 있지만 제한된 하드디스크로 모든 파티션을 나누게 되면 각 파티션에서 사용할 수 있는 용량이 고정되어 버리기 때문에 실제 서비스 도중 특정 파티션에 용량이 초과할 경우 추가 하드 디스크를 사용해야 하는 경우가 발생하기 쉽다.

뿐만 아니라 많은 파티션을 나누어 놓게 되면 성능면에서는 Disk i/o 를 분산 시킬수 있어 바람직 할수 있지만 관리 측면을 고려 해 본다면 필요에 의한 적절한 파티션 분배가 이루어 져야 할것이다.

각 파티션별로 그 의미에 대해 알아보자.

/boot :

/boot 파티션은 리눅스 부팅에 관련된 파일이나 이미지들이 모여 있는 파티션 이다. 만일 /boot 파티션을 나누어 놓은 경우는 크게 세 가지 의미가 있다.

첫째는 안정성이다. 실제 /boot 파티션을 나누지 않고 그냥 / 에 포함 시킬 경우 시스템이 과도한 사용이나 해킹등의 이유로 / 파티션이 망가질 경우 실제 / 포함 된 /boot 디렉토리의 booting image 들 역시 깨어질 위험성이 크다. /boot 가 나누어져 있을 경우 / 가 망가지더라도 새로운 / 파티션을 만들고 /boot 에 있는 부팅 이미지를 가지고 어느 정도 까지는 복구가 가능하다.

둘째는 기능성(?)이라 볼수 있다. 실제 /boot 에 들어 있는 kernal image 가 즉 리눅스 OS 라 볼수도 있다. 만일 당신이 개발자나 리눅스 전문 엔지니어라서 여러개의 배포판을 설치 할 경우 각 배포판 마다 /, swap 등의 파티션을 별도로 나누어서 설치해야 하는 어리석음을 갖지 말길 바란다. 실제 배포버전별로 /boot 파티션을 나누어 준다고 하면 각각의 다른 리눅스 배포판을 설치 하고도 실제 /etc /usr/local 등에 있는 설정이나 패키지를 통합해서 관리 할수가 있다.

셋째는 리눅스 OS의 부트로더의 기능적인 제한때문에 /boot 를 나누는 경우가 많다. 리눅스의 대표적인 부트로더로 Lilo 가 있는데 이는 실제 1024 실린더 밖에 설치 될 경우 치명적인 에러가 발생한다. 그래서 대부분이 하드디스크의 MBR 영역에 설치를 하게 되는데 만일 MBR 에 다른 부트 로더를 설치하고 Lilo 는 리눅스 파티션의 첫번째 파티션에 설치를 한다고 할때 /boot 파티션을 첫 번째 파티션으로 만들어 이곳에 Lilo를 설치 할수 있다.

/ :

루트 파티션은 실제 리눅스 파티션의 최 상위 파티션으로 실제 / 파티션과 Swap 파티션만으로도 리눅스를 설치 하여 서비스를 할 수도 있다. 하지만 이렇게 설치를 하게 되면 앞에서 언급하였고 뒤에서 설명하는 바와 같이 부적절한 상황을 초래하기 쉽다. 일단 커널 이미지에서 실제 / 파티션을 찾아서 / 파티션을 최상위로 해서 하위의 파티션들을 찾기 때문에 / 파티션이 없으면 리눅스를 부팅 시킬수 없게 된다. 아래에서 설명하는 바와 같이 여러개의 파티션을 나눌 경우 실제 별도의 파티션을 나누지 않는 상위 디렉토리들이 모두 이 / 파티션안에 포함되게 될것이다.

/etc :

/etc 는 리눅스 시스템의 모든 설정 파일이 모여 있는 디렉토리이다. 만일 /etc 를 / 파티션에 포함한 경우 변수에 의해 / 파티션이 깨어진 경우 별도의 백업이 없으면 관리자로써 상당히 난처한 상황에 빠지게 딘다.

즉 아무리 데이터를 백업을 하고 있다 하더라도 설정 파일이 없으면 서비스를 구동 할수 없게 된다. 그렇기 때문에 /etc 를 별도의 파티션으로 나누는 것도 고려해야 할것이다. 하지만 /etc 의 용량은 매우 작기 때문에 백업만 적절히 한다고 하면 굳이 나눌 필요는 없다. 파티션이 너무 많이 나누어 지면 관리측면에서 부적절 할수도 있다.

/usr :

/usr 는 실제 레드헷 리눅스의 RPM 패키지가 설치되는 파티션이다. 이 파티션을 나누는 이유는 크게 성능과 패키지 관리 측면에서 볼수 있다.
일단 레드헷 리눅스의 모든 프로그램이 이곳에 설치가 된다고 볼수 있다.
즉 서비스 구동 시에 프로그램 성능에 가장 큰 영향을 주는 파티션인 만큼 다른 Disk i/o 과 많은 다른 파티션과 동일 파티션으로 구성하면 성능에 많은 지장을 주게 된다.
관리 측면으로 보면 최근 배포판 버전으로 업그레이드를 하고자 할때 /usr 파티션이 나누어져 있을 경우 설치 시 /usr 파티션만 새로 포맷하고 새 패키지를 설치하면 실제 redhat 7.3 에서 redhat 9 로 업그레이드를 한다고 해서 서버의 모든 자료를 백업하고 새로 OS 설치 후 복구하는 수고를 들수 있다.

/usr/local :

/usr/local 는 응용프로그램을 Source 로 설치 할 경우 파일이 위치 하는 디렉토리이다. 기본적으로 배포판의 패키지는 기본적이 서버 구성에 해당하는 패키지만 설치 하고 나머지는 모두 Source 설치 하는걸 권장 한다. 리눅스 배포판의 패키지를 무분별하게 설치 할 경우에는 보안/ 관리 측면에서 상당히 부적절 할 수 있다.

그러므로 최소 설치 후 필요한 프로그램을 Source 로 설치 하는 것을 권장하는 바이다. 즉 Source 로 설치하는 프로그램을 관리하는 파티션이다. /usr 과 같은 의미를 가진다. 성능과 패키지 관리 측면이다. 만일 / , /usr 등의 문제로 인해 다시 설치 할 경우 /usr/local 파티션이 / , /usr 에 속해 있으면 재 설치 할때 마다.. 일일이 Source 설치도 다시 해야 한다. 하지만 /usr/local 이 별도로 나누어져 있으면 OS를 다시 깔아도 실제 서비스 프로그램을 다시 깔 필요는 없게 된다.

/var :

/var 는 시스템 log 파일이 저장 되는 곳이다. /var 파티션을 분리하는 이유는 실제 log 를 기록 하면 그만큼 많은 Disk i/o 병목이 시스템에 발생하게 된다. 실제 서비스가 이루어 지는 / 혹은 /usr 와 사용자 데이터가 저장되는 /home 등과 같은 파티션이 존재 하게 되면 /var 에 일어나는 병목으로 인해 시스템의 성능이 저하되게 될것이다. 이를 방지하기 위해 /var 파티션을 별도로 활당하던지 네트워크 드라이브를 이용하여 원격디스크를 활용하는 방법을 사용한다. (log 서버와 같이..)

/var 는 로그 및 메일 큐등이 쌓이는 곳이기 때문에 해커들이 고의적으로 시스템에 나쁜 코드를 심어 놓거나 mail 서버의 spam를 무수히 보냄으로 해서 /var의 디스크 용량이 무지커지는 경우가 발생한다. 이때 /var의 파티션이 나누어 있지 않는 경우 실제 / 파티션의 disk 용량이 가득차게 되고, 이로 인해 / 파티션의 중요한 파일들이 손상을 입게 된다. 이런 일들을 방지 하기 위해서도 /var 파티션을 나누는 것을 권장한다.

이밖에 해커들이 / 파티션을 공격하여 시스템을 깨어버린다 하더라도 실제 /var 파티션이 나누어져 있으면 이런 해커의 작업등이 시스템 로그에 남게 됨으로 시스템은 망가졌지만 이후 /var 의 로그를 이용하여 시스템에 무슨 문제가 있었는지를 확인할 수도 있다.

/tmp :

/tmp 는 실제 시스템 서비스가 가동될때 임시 파일이나 세션 파일들이 생기는 곳이다. 보통 nobody 권한의 파일들이 생김으로 해서 보안이 약한 곳이라 볼수 있는데 해커들이 주고 이 /tmp 를 통해 원격지에서 시스템을 공격하는 경우가 많다. 그러므로 해서 해커의 공격을 받더라도 /tmp 디렉토리만 영향을 받게 하기 위해 파티션을 분리하는것을 권장한다.

swap :

은 물리적인 메모리가 부족한 경우 하드디스크를 메모리 처럼 사용할 수 있게 하는데 이용되는 파티션이다.
보통 시스템의 물리적인 메모리의 1.5배에서 2배로 잡는것이 정석이다.
시스템의 주요 사용 용도 중에 매우 큰 용량의 파일을 다루는 작업일 경우 보통 swap 을 크게 잡아야 한다. 대표적으로 Oracle 의 경우 보통 DB file 하나가 작게는 500M 에서 크게는 2Gbyte 정도 하는 실제 DB 엔진에서 메모리에 이 큰 파일을 올려 놓고 작업을 하게 되는데 실제 물리적 메모리가 1.5 G 일 경우 swap 이 없는 경우에는 작업을 할수 없게 되거나 페이징이 일어나 성능이 급속도록 떨어지게 된다. 이를 방지하기 위해서 서비스에 이용되는 메모리 수치를 예상하여 swap 을 잡으면 된다. swap 은 상황에 따라서 여러가지 방법으로 추가할수 있기 때문에 초기에는 권장방법대로 1.5배에서 2배 정도로 잡으면 된다.

ii. SCSI Adapter Driver Patch

최신 Ultra320 SCSI Controllor 의 경우 기본 레드헷 지원 드라이버를 사용하면 i/o error 를 발생 하는 경우가 있다. 대표적인 제품으로 Adatec 의 AIC-29320 제품의 경우 그러함. 설치 시에 그냥 설치 하면 SCSI 장치 인식 중에 멈처 버리는 증세가 발생함.

boot : linux apic

와 같은 방식으로 부팅하면 장치는 인식하지만 레드헷 기본 커널에 있는 드라이버로 서비스 가동 시에 조금 과도한 Disk Access 작업을 할 경우 kernel panic 상태가 된다.

이에 설치 전에 각 장치 공급사의 홈페이지에서 최신 드라이버를 다운 받아서 설치 시에 최신 드라이버를 인식하도록 해주어야 한다.

http://www.adaptec.com/ 에서 최신 드라이버를 다운 받을 수 있음

먼저 각 하드웨어 공급사의 최신 드라이브를 다운 받은 후 설치 시 적용할 수 있는 이미지 디스켓을 만든다.

==========

fdformat /dev/fd0
dd if=aic79xx-2.0.2-i686-rh90.img of=/dev/fd0 bs=1024

==========

그런후 설치 CD 로 부팅 후 boot prompt 가 뜨면 < linux dd > 란 command 로 설치 시 추가 드라이브를 적용 시킨다.

==========

boot : linux dd

==========

그럼 설치 CD 의 커널 이미지로 부팅을 진행하면서 하드디스크 혹은 기타 장치를 인식하기 전에 추가 드라이브 디스켓을 삽입하라는 메세지가 뜬다. 이때 위에서 만든 드라이브 디스켓을 삽입한다.

그러면 설치 CD 에 있는 장치 드라이브를 이용하는것이 아니라 다운 받은 최신 드라이브로 장치를 인식하여 설치를 하게 된다.

② 레드헷 리눅스 서버 설치

서버용으로 설치를 할 경우 앞에서 설명한 바와 같이 이 서버가 서비스할 서비스 내역을 충분히 이해 한 후 반드시 필요한 패키지 만을 설치 하는 것을 권장한다.

사용하지 않는 패키지를 설치 할 경우 사용하지 않는 패키지의 보안적 문제가 생겼을때 관리자는 이를 신경쓰지 않게 되고, 이 패키지를 통한 해킹의 위험을 가

지게 된다.

기본적은 설치 과정에 대한 설명은 생략 하고 파티션과 패키지 구성에 대한 정책만을 설명하겠다. 이는 저의 개인적인 견해를 설명한것이니 이가 완변한 설치 정책이라 말하지는 않겠다.

i. 파티션 정책

실제 서비스를 대상으로 하는 서버를 설치 할 경우에는 전 기본적으로 아래와 같이 파티션을 나눈다.

/boot
/
/usr
/usr/local
/var
/tmp
/home
swap

/boot : 100M ~ 200M
/ : 1G
/usr : 3G
/usr/local : 2G
/var : 2G ~ 3G
/tmp : 500M
/home : 필요한 만큼
swap : 물리적인 메모리의 2배

여기서 /boot /tmp 는 그 용량이 작기 때문에 적절한 백업을 할 경우에는 관리상의 이유로 / 에 포함하는 경우도 많다.

/usr/local 역시 관리상의 이유로 /usr 에 포함하는 경우도 많다.
대신 이와 같이 포함하는 경우에는 반드시 포함 시키는 파티션의 크기에 신경을 써야 한다.

/var 파티션의 경우 로그의 중요성이 낮거나 메일 서비스등을 하지 않는 경우에는 관리상의 이유로 하여 / 에 포함하는 경우도 많다. 하지만 메일 서버등을 중점으로 하는 서비스에서는 반드시 /var 의 파티션을 나누어 주고 용량도 충분히 주길 바란다.

/home /usr/local 의 파티션을 별도로 나눌 경우 성능을 고려 한다면 설치 시 파티션을 지정 하지 말고 기본 OS 설치 후 fdisk 를 통해 수동으로 파티션을 나눈 후 레드헷 기본 파일 시스템인 ext3 대신 reiserfs 혹은 xfs 파일 시스템을 사용하는 것을 권장 한다. ext3 에 비해 reiserfs 나 xfs 파일 시스템이안정성에서 우수함을 인증 받은 상태이고 성능에서도 30% 정도의 성능 향상을 가져 올수 있다.

(hdperm 으로 디스크 성능을 체크 해 보면 ext3 의 경우 초당 40M 정도의 read buffer 의 성능을 나타내지만 reiserfs 의 경우 초당 55M 의 성능을 발휘한다.)

/home 이나 /usr/local 파티션의 성격 상 디스크 사용량이 많거나 실제 사용하는 프로그램이 설치 되는 곳인만큼 시스템의 Disk I/O 성능의 영향을 많이 받는 곳이기에 보다 안정스럽고 좋은 성능의 파일 시스템을 사용하는게 좋다.

swap 역시 oracle 과 같은 큰 파일을 다루는 서버라고 하면 1G 용량으로 여러개의 swap 파티션을 잡아 두는 것을 권장한다.
(그냥 큰 용량으로 swap을 잡지 않고 적절한 용량으로 나누어 여러개를 잡는 이유는 swap 파티션의 크기 역시 일반 IA32 시스템에서는 2G 제한이 있고 2G의 스왑파티션 보다는 1G의 스왑파티션 2개가 I/O 분산으로 인해 성능이 좋기 때문이다.

OS 의 설치는 초보 엔지니어나 경력 엔지니어나 다 기본으로 하기 마련이다.
하지만 시스템 엔지니어의 시작이 OS 설치 라고 하면 마지막 역시 OS 설치라고 말할수 있는 만큼 자신만의 설치 방법을 세워 두길 바란다.

ii. 패키지 정책

실제 OS 구동에 필요한 기본 프로그램 (부팅관련 프로그램, 라이브러리)은 배포판에 있는 패키지를 선택하지만 서비스에 필요한 프로그램은 대도록이면 Source 로 직접 설치 하는 것을 권장한다.

이는 패키지 관리 및 보안성격상에도 큰 의미를 가지고 있지만 시스템 성능에 미치는 영향도 크다

실제 배포판은 대표적인 아키텍처에 해당하는 시스템에서 해당 프로그램을 build 하여 패키지를 만들어 놓은 것이다. 유사한 아키텍처를 가진 시스템에서 사용하는데는 지장이 없지만 그래도 자기 시스템에 최적화된 프로그램을 사용하기 위해서는 실제 Source 로 해당 시스템에서 직접 설치 하는 것이 좋다.

(이와 같은 패키지 관리를 한다고 하면앞에서 설명한 바와 같이 /usr/local 파티션을 별도로 나누어 관리하는 것이 시스템 문제 발생하여 재 설치 시 오랜 시간이 소요되는 Source rebuiling 작업을 생략할 수 있게 된다.)

레드헷 리눅스에서의 패키지 선택은 기본적으로 패키지 선택 단계에서

/ 편집기 / DNS 서버 / 네트워크 서버 / 개발 도구 / 커널 개발 /시스템 도구

정도를 기본 설치 하고 기타 서비스 관련 프로그램 및 기타 관리 도구는 필요한 프로그램만을 수동으로 설치 하면 된다.

iii. 설치 완료 후 기본 확인 작업

기본적으로 설치가 완료 된 후 부팅이 정상적으로 되는지와 부팅 시 에러 내용이 있는지, 파티션이 정책대로 적용이 되었는지 등을 확인 합니다.

```
# dmesg  -> 부팅 메세지 확인
# df -h  -> 파티션별 용량 확인
```

부팅 시 bootloader 에서 문제가 발생하는 경우가 많다.

특히 SCSI 하드와 IDE 하드를 같이 장착하고 설치 한 경우 LILO 혹은 GRUB 등의 bootloader 에서 정상적으로 커널 이미지를 Loading 하지 못하는 경우가 생긴다.
기본적인 해결 방법을 설명하겠다.

// LILO Bootloader 로 부팅 시 문제 발생 했을때 해결 방법

==========
리눅스설치CD로 부팅해서 rescue 모드로 들어갔다.
df 쳐 보니 /dev/hda2 가 /mnt/sysimage 에 mount 되어 있다.
하드디스크에 있는 lilo.conf 는 /mnt/sysimage/etc/lilo.conf 에 잘 있다.
이제 이 파일을 잘 고쳐서 lilo 를 실행하면 된다. rescue 모드에서도 vi는 실행된다.

vi /mnt/sysimage/etc/lilo.conf
/mnt/sysimage/etc/lilo.conf 를 수정 후

/etc 에 복사한다
cp /mnt/sysimage/etc/lilo.conf /etc
여기까지만 하고 lilo 를 실행시키면 에러가 난다.

/etc/lilo.conf 에서 /boot 라고 되어있는 곳을 /mnt/sysimage/boot 로 바꾸고 저장한다.

이제 lilo 를 실행시킨다.

/mnt/sysimage/sbin/lilo

/boot/chain.b 를 못 찾는다는 메시지가 나타나면,

cp /mnt/sysimage/boot/chain.b /boot/ 한 다음에
#/mnt/sysimage/sbin/lilo

혹은

chroot /mnt/sysimage
/sbin/lilo

하면 된다.

==========

// GRUB Bootloader 로 부팅 시 문제 발생 했을때 해결 방법

==========

긴급시 grub 으로 부팅을 시키는 과정입니다.

grub >

먼저 root 지정을 해 줍니다.

grub > root (hd0,0)

-> 실제 /boot 파티션의 위치를 지정한다. 만일 별도의 /boot 파티션
이 없는 경우에는 / 위치를 지정합니다.

grub > kernel /vmlinuz-2.4.20-8smp ro root=/dev/sda2

-> 실제 / 파티션 위치를 적어 줍니다.

grub > initrd /initrd-2.4.20-8smp.img

-> SCSI 하드인 경우 initrd 이미지가 없으면 하드 인식을 못함.

grub > boot

하면 부팅 됩니다.

그런 후 /etc/grub.conf 를 수정하거나 쉽게 LILO 를 수정하여 다시 적용하고 실행
하면 된다.

참고로 LILO 로 SCSI 와 IDE 를 병행 할때 sda 가 첫번째 디스크가 아니라는 에러가
발생 하는 경우가 있다.

이는 LILO라는 부트로더는 실제 BIOS 의 장치에 관련된 정보를 참조 하지 않기 때문에
발생하는 문제이다.

디스크에 관련해서 이런 문제가 발생하면..

lilo.conf 에 다음 내용을 추가해 주면 된다.

```
disk=/dev/sda
bios=0x80
disk=/dev/hda
bios=0x81
```

위의 설정 내용은 SCSI 가 첫번째 하드라는 것을 LILO 에서 인식하도록 해주는 것이다.
이렇게 해야 부팅시 정상적으로 LILO 가 띄워 질 것이다.

③ 설치 완료 후 부가 작업

i. 주요 서비스 파티션 구성 확인 및 파일 시스템 설정

앞의 파티션 정책에서 설명 했듯이 주요 서비스 파티션 중 데이터 관련 파티션인
/home 과 서비스 프로그램 관련 파티션인 /usr/local 을 구성해야 하고 이 두개의 파티션
을 안정성 및 효율이 높은 reiserfs 혹은 xfs 로 파일 시스템을 설정 하는 하도록 한다.

fdisk /dev/sda

-> /usr/local : 3G
-> /home : 나머지 전부 활당

mkfs.reiserfs /dev/sda8 -> /dev/sda? 에는 해당 파티션의 디바이스명을 적어준다.

```
# mkfs.reiserfs   /dev/sda9

# vi /etc/fstab
================================================================================
==
LABEL=/                 /                       ext3      defaults           1 1
LABEL=/boot             /boot                   ext3      defaults           1 2
none                    /dev/pts                devpts    gid=5,mode=620   0 0
none                    /proc                   proc      defaults           0 0
none                    /dev/shm                tmpfs     defaults           0 0
LABEL=/tmp              /tmp                    ext3      defaults           1 2
LABEL=/usr              /usr                    ext3      defaults           1 2
LABEL=/var              /var                    ext3      defaults           1 2
/dev/sda6               swap                    swap      defaults           0 0
/dev/cdrom              /mnt/cdrom              udf,iso9660 noauto,owner,kudzu,ro 0 0
/dev/fd0                /mnt/floppy             auto      noauto,owner,kudzu 0 0
# 아래를 추가해줌
     /dev/sda8              /usr/local              reiserfs   defaults          1 2
     /dev/sda9              /home                   reiserfs   defaults          1 2

================================================================================
==

# tar czvf local.tgz /usr/local    ->  /usr/local 의 디렉토리 구조를 백업해 둠
# cp local.tgz /
# mount -a
# cd /
# tar xzvf local.tgz
```

실제 파티션과 파일 시스템 구성이 정상적으로 되었는지를 확인 한다.

```
# df -h
================================================================================
==
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda2             981M  121M  811M  13% /
/dev/sda1             198M   15M  173M   8% /boot
none                 1009M     0 1009M   0% /dev/shm
/dev/sda7             487M  8.1M  454M   2% /tmp
/dev/sda5             2.9G  898M  1.9G  33% /usr
/dev/sda3             2.9G   58M  2.7G   3% /var
/dev/sda8             2.9G   33M  2.8G   2% /usr/local
/dev/sda9              22G   33M   22G   1% /home
================================================================================
==
```

ii. 시스템 구성 확인

마지막으로 현재 시스템의 H/W, S/W 구성 정보를 확인 및 기록 해 두도록 합니다.
이는 현재 시스템 장치가 재대로 인식을 하는지 확인과 동시에 이후 유지 관리를 위한 시스템 정보를 기록하는 의미를 가지고 있습니다. 이는 시스템의 정보를 확인 하는 방

법으로 이후 관리 시에 큰 도움이 될것입니다.

- booting message 확인
dmesg

- CPU 정보
cat /proc/cpuinfo

- 메모리 정보
cat /proc/meminfo

- SCSI 정보 HDD 정보
cat /proc/scsi/scsi

- SCSI controllor 정보
cat /proc/scsi/aic7xxx/0

- IDE 하드 정보
cat /proc/ide/hda/model
cat /proc/ide/hda/setting

- 네트워크 정보
ifconfig
route -n

- 파티션 정보
fdisk -l /dev/sda

- 디스크 용량
df -h

이로써 대략적인 설치 과정이 완료 되었습니다.
다음 장에서는 설치 이후 설치 된 패키지의 최신 패키지 업데이트 및 기타 하드웨어의 최신 드라이브 업데이트에 대한 설명을 하겠습니다.

2. 패키지 업데이트

① S/W Package Update

리눅스 시스템은 Open 된 OS 인 만큼 보안에 대한 취약점이 있다고 생각할 수 있지만 이와 달리 Open 된 시스템인 만큼 보안의 취약점을 미리 사전에 파악하여 패치 함으로 관리자의 관심만 유지되면 보안에 더 강력하다고 말할 수 있다.

레드헷 리눅스의 경우 up2date 란 프로그램을 통해 윈도우 처럼 자동으로 패키지를 업데이트 할 수 있는데 기본적으로 배포판에 설치되는 up2date 로 업그레이드를 하면 SSL 인증 에러가 발생하게 될것이다. 이는 현재 배포 버전인 Redhat9 이 발표된 이후 레드헷사의 up2date 인증 방식에 변경이 있었기 때문에 정상적인 up2date를 이용하여 전체적인 패키지 업데이트를 하기 위해서는 먼저 up2date 를 업그레이드 해야 한다.

wget ftp://updates.redhat.com/9/en/os/i386/up2date*

위의 구문으로 바로 up2date 에 관련된 패키지를 다운 받을 수 있을 것이다.
up2date / up2date-gnome 두개의 패키지를 다운 받게 될것인데 이중 up2date-gnome
은 Xwindows 의 gnome 이 설치 된 경우에만 설치 하도록 한다.

rpm -Uvh up2date

/etc/rc.d/init.d/rhnsd restart
rhn_register

최초 rhn_register 를 실행하면 rhn_register 구성 편집 화면이 나오는데 그냥
q 를 치면 빠저 나온다.

rpm --import /usr/share/rhn/RPM-GPG-KEY
rhn_register
gpg --import /usr/share/rhn/RPM-GPG-KEY

레드헷 사에 현 시스템의 정보 및 간단한 인증 정보를 기록합니다.
정식 레드헷 배포판을 구매하면 up2date 의 지원을 계속 받을 수 있지만 공개 배포판의
경우에는 3개월 기간 제한이 있기 때문에 설치 시에 한번 지원 받는다는 의미를 가진다.

up2date -p -> 현 시스템이 RPM 패키지 정보를 정리한다.
up2date -u -d -> 업데이트를 시킬 패키지를 다운 받는다.

up2date 에서 -d 옵션을 사용하면 업데이트 대상 패키지를 다운 받아 /var/spool/up2date
안데 저장된다.

-u 옵션만 사용하면 다운 받아서 업데이트 패키지를 바로 설치 하고 설치가 완료된
rpm 을 삭제 하게 됩니다. 만일 업데이트 대상 시스템이 여러대 일 경우 계속 같은 방법
으로 업데이트 시킬 필요 없이 한곳에서 다운 받아놓고 그 패키지를 이용하여 다른 시스
템들의 업그레이드를 시키는 방법이 편할 것이다. 단 위 경우는 시스템의 패키지 구성이
유사한 경우에 의미가 있다.

② H/W Driver Update

dmesg message 를 살펴 보면 커널이 로딩한 장치의 드라이브 버전을 확인 할 수 있다.
흔히 문제가 되는 것이 SCSI Controllor 와 Intel Gigabit Card 가 대표적이다.
먼저 현재 시스템에 설치 시 인식된 장치를 확인합니다.

cat /etc/modules.conf
==========
alias eth0 e1000
alias eth1 e100
alias scsi_hostadapter aic7xxx
alias usb-controller usb-uhci
==========

dmesg | grep Driver
dmesg | grep DRIVER

dmesg | grep driver

scsi0 : Adaptec AIC7XXX EISA/VLB/PCI SCSI HBA DRIVER, Rev 6.2.8
Intel(R) PRO/1000 Network Driver - version 4.4.19-k1
Intel(R) PRO/100 Network Driver - version 2.1.29-k2

대표적인 버전은 AIC79XX -> 2.0.2 이상
e1000 -> 5.0.43 이상
e100 -> 2.2.21 이상

의 버전을 설치 하도록 한다.

rpm 으로 만들어진 드라이브 버전은 현 시스템 커널 버전에 의존함으로 드라이버 설치 후 kernal upgrade를 할 경우에는 반드시 다시 장치 드라이브 버전을 확인하셔야 합니다.

3. OS 관련 기본 보안 정책 및 보안 패키지 설치

① OS 관련 기본 보안 정책

i. 데몬 관리 정책

* 불필요한 데몬 제거

/sbin/chkconfig --del anacron
/sbin/chkconfig --del atd
/sbin/chkconfig --del autofs
/sbin/chkconfig --del echo
/sbin/chkconfig --del finger
/sbin/chkconfig --del gpm
/sbin/chkconfig --del nfs
/sbin/chkconfig --del nfslock
/sbin/chkconfig --del portmap
/sbin/chkconfig --del rlogin
/sbin/chkconfig --del rsh
/sbin/chkconfig --del rsync
/sbin/chkconfig --del lpd
/sbin/chkconfig --del xfs

* 기본 데몬

/sbin/chkconfig --add crond
/sbin/chkconfig --add network
/sbin/chkconfig --add sshd
/sbin/chkconfig --add syslog

* 서비스 데몬

/sbin/chkconfig --add named
/sbin/chkconfig --add proftpd

/sbin/chkconfig --add sendmail
/sbin/chkconfig --add nfs
/sbin/chkconfig --add portmap
/sbin/chkconfig --add rlogin
/sbin/chkconfig --add rsh
/sbin/chkconfig --add rsync

* 불필요한 xinetd 데몬 제거

rm -f chargen
rm -f chargen-udp
rm -f daytime
rm -f daytime-udp
rm -f echo
rm -f echo-udp
rm -f finger
rm -f ntalk
rm -f rexec
rm -f rlogin
rm -f rsh
rm -f rsync
rm -f servers
rm -f services
rm -f sgi_fam
rm -f talk
rm -f telnet
rm -f time
rm -f time-udp
rm -f proftpd-inetd

ii. 기본 시스템 계정 정책

* 불필요한 시스템 계정 삭제
# userdel adm
# userdel lp
# userdel sync
# userdel shutdown
# userdel halt
# userdel news
# userdel uucp
# userdel operator
# userdel games
# userdel gopher

* 불필요한 시스템 그룹 삭제

groupdel adm
groupdel lp
groupdel news
groupdel uucp

```
# groupdel games
# groupdel dip
```

iii. 시스템 퍼미션 정책

* suid sgid 퍼미션의 명령어 검색 후 퍼미션 변경

```
# find / -type f ₩( -perm -4000 -o -perm -2000 ₩)

chmod 700 /usr/bin/chage
chmod 700 /usr/bin/gpasswd
chmod 700 /usr/bin/wall
chmod 700 /usr/bin/chfn
chmod 700 /usr/bin/chsh
chmod 700 /usr/bin/newgrp
chmod 700 /usr/bin/write
chmod 700 /usr/bin/passwd
chmod 700 /usr/bin/at
chmod 700 /usr/bin/lockfile
chmod 700 /usr/bin/rcp
chmod 700 /usr/bin/rlogin
chmod 700 /usr/bin/rsh
chmod 700 /usr/bin/slocate
chmod 700 /usr/bin/crontab
chmod 700 /usr/libexec/openssh/ssh-keysign
chmod 700 /usr/sbin/ping6
chmod 700 /usr/sbin/traceroute6
chmod 700 /usr/sbin/usernetctl
chmod 700 /usr/sbin/userhelper
chmod 700 /usr/sbin/lockdev
chmod 700 /usr/sbin/userisdnctl
chmod 700 /usr/sbin/sendmail.sendmail
chmod 700 /usr/sbin/traceroute
chmod 700 /usr/sbin/utempter
chmod 700 /bin/ping
chmod 700 /bin/mount
chmod 700 /bin/umount
chmod 700 /bin/su
chmod 700 /sbin/pam_timestamp_check
chmod 700 /sbin/pwdb_chkpwd
chmod 700 /sbin/unix_chkpwd
chmod 700 /sbin/netreport

chmod 4750 /usr/bin/rcp
chmod 4750 /usr/bin/rsh
chmod 4750 /usr/bin/rlogin
chmod 4111 /usr/bin/sudo
chmod 2750 /usr/sbin/sendmail.sendmail
chmod 4750 /bin/su
```

* 중요 시스템 설정 파일 보호 정책

chattr +i /etc/fstab

* 관리자 그룹에 포함된 계정에게 시스템 관리 명령어 사용 허용 정책

chmod 750 /bin/ps
chmod 750 /bin/netstat
chmod 750 /bin/dmesg
chmod 750 /bin/df
chmod 750 /usr/bin/w
chmod 750 /usr/bin/who
chmod 750 /usr/bin/last
chmod 750 /usr/bin/top
chmod 750 /usr/sbin/lsof

chgrp wheel /bin/ps
chgrp wheel /bin/netstat
chgrp wheel /bin/dmesg
chgrp wheel /bin/df
chgrp wheel /usr/bin/w
chgrp wheel /usr/bin/who
chgrp wheel /usr/bin/last
chgrp wheel /usr/bin/top
chgrp wheel /usr/sbin/lsof
chgrp wheel /usr/bin/rcp
chgrp wheel /usr/bin/rsh
chgrp wheel /usr/bin/rlogin

vi /etc/group
==========
.
.
wheel:x:10:root,clunix,alang
.
==========

* 시스템 최 상위 디렉토리 퍼미션 정책

chmod 711 /
chmod 711 /home
chmod 711 /var
chmod 711 /var/log
chmod 711 /etc
chmod 700 /root

iv. 일반 사용자 관리 정책

* 일반 shell 사용자 작업 내용 감시 하기

/etc/profile 파일의 제일 밑 부분에 아래 내용 추가

```
==================================================================

if [ $LOGNAME != "admin" ]
then
HISTFILE=/var/log/user_history
TMOUT=200
echo -n "
==================================================================
userhistory 로그인 ID: $LOGNAME 접속시간 : `/bin/date`
==================================================================
" >> /var/log/user_history
fi

==================================================================
```

/root/.bashrc 파일 밑 부분에 다음 라인 추가

```
HISTFILE=/root/.bash_history
TMOUT=-1
```

마지막으로 /var/log/user_history 파일의 퍼미션을 662 으로 해줍니다.

v. TCP 보안 정책

* TCP Wrapper 로 TCP 서비스 접근 권한 정책

vi /etc/hosts.deny

```
sshd : ALL : twist ( /root/bin/hostchk Y Y %a %c %d %h %n %p %s %u ) &
in.proftpd : ALL : twist ( /root/bin/hostchk Y Y %a %c %d %h %n %p %s %u ) &
```

vi /etc/hosts.allow

```
sshd : localhost 127.0.0.1 211.241.202. 192.168.1.
in.proftpd : localhost 127.0.0.1 211.241.202. 192.168.1.
```

vi /root/bin/hostchk

```
```

```
#!/bin/sh

##################################  변수정의부문

# 메일 수신자
mailto=alang@clunix.com

# 화면출력 여부, 메일전송 여부
dsp=$1; msg=$2

# 접속자 정보 등
a=$3; c=$4; d=$5; h=$6; n=$7; p=$8; s=$9; u=$10

# 현재 시간
time=`date`

# 접속시도자 소속 서버의 finger 정보
finger=`/usr/bin/finger -l @$h 2> /dev/null`

##################################  화면 출력부문

if [ $dsp = Y ]
     then

/bin/echo -n "
                        ===================================
                             접속이 허용되지 않습니다.
                        ===================================

                Access Time                 : $time
                Client host address       : $a
                Client information        : $c
                Client host name(or IP) : $h
                Client host name          : $n
                Client user name          : $u
"
fi


##################################  메일 송신부문

if [ $msg = Y ]
     then

/bin/echo -n "
                          ================================
                               접속 거부자 상세정보
                          ================================

          Access Time                         : $time
          Access client host address       : $a
```

```
          Access client information        : $c
          The daemon process name          : $d
          Access client host name(or IP) : $h
          Access client host name          : $n
          The daemon process id            : $p
          Server information               : $s
          Access client user name          : $u
                    ---------------------------------------
                       Access client finger information
                    ---------------------------------------
                               $finger
                  -----------------------------------------------
 " | ₩
       /bin/mail -s "tcp_wrapper report [$d]" $mailto

fi
```

* Syn Flooding DOS 공격 방지 정책

vi /root/bin/synfl

```
sysctl -w net.ipv4.tcp_max_syn_backlog=1024
sysctl -w net.ipv4.tcp_syncookies=1
sysctl -w net.ipv4.icmp_destunreach_rate=1
sysctl -w net.ipv4.icmp_echo_ignore_broadcasts=1
sysctl -w net.ipv4.icmp_echoreply_rate=1
sysctl -w net.ipv4.icmp_ignore_bogus_error_responses=1
sysctl -w net.ipv4.icmp_paramprob_rate=1
sysctl -w net.ipv4.icmp_timeexceed_rate=1
sysctl -w net.ipv4.igmp_max_memberships=1
sysctl -w net.ipv4.ip_default_ttl=64
sysctl -w net.ipv4.ip_forward=0
sysctl -w net.ipv4.ipfrag_time=15
sysctl -w net.ipv4.tcp_syn_retries=3
sysctl -w net.ipv4.tcp_retries1=3
sysctl -w net.ipv4.tcp_retries2=7
sysctl -w net.ipv4.conf.eth0.rp_filter=2
sysctl -w net.ipv4.conf.lo.rp_filter=2
susctl -w net.ipv4.conf.default.rp_filter=2
sysctl -w net.ipv4.conf.default.rp_filter=2
sysctl -w net.ipv4.conf.all.rp_filter=2
sysctl -w net.ipv4.conf.eth0.accept_redirects=0
sysctl -w net.ipv4.conf.lo.accept_redirects=0
sysctl -w net.ipv4.conf.default.accept_redirects=0
sysctl -w net.ipv4.conf.all.accept_redirects=0
sysctl -w net.ipv4.conf.eth0.accept_source_route=0
sysctl -w net.ipv4.conf.lo.accept_source_route=0
sysctl -w net.ipv4.conf.default.accept_source_route=0
```

```
sysctl -w net.ipv4.conf.all.accept_source_route=0
sysctl -w net.ipv4.conf.eth0.bootp_relay=0
sysctl -w net.ipv4.conf.lo.bootp_relay=0
sysctl -w net.ipv4.conf.default.bootp_relay=0
sysctl -w net.ipv4.conf.all.bootp_relay=0
sysctl -w net.ipv4.conf.eth0.log_martians=1
sysctl -w net.ipv4.conf.lo.log_martians=1
sysctl -w net.ipv4.conf.default.log_martians=1
sysctl -w net.ipv4.conf.all.log_martians=1
sysctl -w net.ipv4.conf.eth0.secure_redirects=0
sysctl -w net.ipv4.conf.lo.secure_redirects=0
sysctl -w net.ipv4.conf.default.secure_redirects=0
sysctl -w net.ipv4.conf.all.secure_redirects=0
sysctl -w net.ipv4.tcp_keepalive_time=30
sysctl -w net.ipv4.tcp_fin_timeout=30
sysctl -w net.ipv4.tcp_tw_buckets=1440000
sysctl -w net.ipv4.tcp_tw_buckets=1440000
sysctl -w net.ipv4.tcp_keepalive_probes=2
sysctl -w net.ipv4.tcp_max_ka_probes=100
```

* 시스템 시작 시 자동 실행 코드 추가

vi /etc/rc.d/rc.local

```
rdate -s time.bora.net
/root/bin/synfl
```

* 최종적으로 Open port scan 확인

기본적인 OS 설정 및 보안 설정이 완료되면 최종적으로 현재 서비스가 이루어
지는 TCP/IP Port 에 대해 시스템 초기 상태에서 알아 두어야 한다.

이후 이때 파악 되지 않는 서비스 포트가 열려 있을 경우 보안에 의심을 해보
아야 할것이다.

Port Scan 도구로는 가장 많이 사용하는것이 nmap,nc,netstat,lsof 등 이다.
사용법은 다음과 같다.

nmap,nc,netstat 등을 이용해서 시스템의 열린 포트를 점검한다.

```
# nmap localhost -p 1-65535
# nc -w 3 -v -z localhost 1-65535
# netstat -ant | grep LISTEN
```

```
21/tcp      open        ftp
22/tcp      open        ssh
23/tcp      open        telnet
25/tcp      open        smtp
53/tcp      open        domain
80/tcp      open        http
3306/tcp    open        mysql
```

② 보안 패키지 설치

i. fcheck 설치 및 관리

** 파일 무결성 체크 프로그램 (Fcheck 설치..)

지금까지의 시스템 설치후 기본 보안에 신경을 써야 할부분에 대해서 알아보았고 마지막으로 현재 기본 보안 처리된 시스템의 무결성에 대해 앞으로 감시해서 유지해야 할것이다. 무결성 체크 프로그램으로는 대표적인게 Tripwire 가 있다.
하지만 여기서 소개하는 Fcheck 는 가볍고 간단하면서 Tripwire 의 역활을 충분히 할수 있다.

먼저 프로그램을 다운 받도록 하자.

http://www.geocities.com/fcheck2000/
가면 최신 프로그램이 있을것이다. 다운을 받고 /usr/local 및에 둔다.
압축을 풀고..fcheck,fcheck.cfg 파일의 설정값을 몇개 수정하자.

```
# tar xzvf FCheck_2.07.59.tar.gz
# cd fcheck
# vi fcheck
```

```
##############################################################################
###############
#                                                                            #
#                    User modifiable variable definitions:                   #
#                                                                            #
##############################################################################
###############

# This should be passed through the command line, but hard coding still works
$config="/usr/local/fcheck/fcheck.cfg";
---------------------------------------- 이부분을 시스템 환경에 맞게 수정

------------------------------------------------------------------------

# vi fcheck.cfg
-----------------------------------------------------------------------
Directory    = /etc/
Directory    = /bin/
```

```
Directory    = /usr/bin/
Directory    = /sbin/
Directory    = /usr/sbin/
Directory    = /usr/local/bin/
Directory    = /usr/local/sbin/
Directory    = /lib/
Directory    = /usr/lib/
Directory    = /usr/local/lib/

#
# Directory 설정은 fcheck 가 체크할 파일들이 들어 있는 디렉토리로 시스템에
# 중요한 명령어나 설정파일등이 있는 곳을 정해 주면 된다.
#

Exclusion       = /etc/passwd
Exclusion        = /etc/shadow

#
# Exclusion 설정은 Directory 설정에서 정한 디렉토리중 자주 변경이 되는 파일
# 이 있어서 체크때마다 걸리므로 체크 생략을 요하는 파일을 정해 주면 된다.
# 호스팅 업체나 기타 자주 사용자를 추가하는 서버라면 위와 같이 해주면 된다.
#

DataBase          = /usr/local/fcheck/data/data.dbf

#
# DataBase 설정은 리눅스 파일 정보를 DB 파일로 저장해서 다음 체크시 비교
# 분석할때 사용되어진다.
#

TimeZone          = GMT-9

#
# 한국 시간을 적용한다. GMT-9
#

#File    = /usr/local/admtools/logs/sol.dbf
```

File 설정은 필요 없으니 주석 처리 해준다.
기타 여러 설정값이 있으나 크게 작동하는데 영향을 주지 않는다.기본값을 적용한다.

이와 같이 적용후 /usr/local/fcheck/data 디렉토리를 만들어 준다.

```
# mkdir /usr/local/fcheck/data
# /usr/local/fcheck/fcheck -ac
```

이와 같이 fcheck -ac 로 파일 무결성 체크를 시작한다. 그럼 data 디렉토리 안에 data.dbf 파일이 생성되어 진다.

이제 Directory 설정에 해당하는 디렉토리 안에다가 파일을 하나 생성하고 재대
로 점검을 하는지 테스트를 하여 보자..

```
# touch /etc/test
# /usr/local/fcheck/fcheck -a
```

그럼 아마 아래와 같은 결과를 출력할것이다.

```
PROGRESS: validating integrity of /etc/
STATUS:
        ADDITION: [zzang911.net] /etc/test
        Inode    Permissons        Size      Created On
        19508    -rw-r--r--         0          Sep 19 20:13 2001
```

자 이제 이 명령어를 이용하여 주기적으로 시스템 파일 무결성을 체크하고 관리
자에게 통보하는 프로그램을 만들어 보도록 하자.

```
# vi fcheck.sh
```

```
#!/bin/sh


CHECK=`/usr/local/fcheck/fcheck -a | grep Inode`
HOSTNAME=`hostname`

if [ -n "$CHECK" ]
    then
    /usr/local/fcheck/fcheck -a > fcheck_result
    mail -s "$HOSTNAME File 무결성 체크 결과" admin@clunix.com < fcheck_result
    rm -f fcheck_result
fi
```

간단한 이정도 스크립터로 보다 효율적인 관리가 가능해 질것이다.
이제 cron 에 등록시켜 놓고..매일 파일 무결성 체크를 간단히 메일로 받아서 관리
할수 있게 된다. 만일 변화된 파일이 정당한 변화라면..

```
# /usr/local/fcheck/fcheck -ac
```

로 DBfile 를 업데이트 시켜 줘야 한다. 아님..계속 메일이 날아 오게 된다.

이로써 시스템 설치후 점검해야 할 보안 설정에 대해서 마치겠다. 시스템 보안은
초기의 보안 설정이 아주 큰 부분을 차지 하고 있다. 하지만..지속적은 감시와
관리역시 초기 보안 상태를 유지하는 가장 중요한 작업이라고도 할수 있다.

4. OS 설치 후 중요 데이터 백업 정책

이른바 “정보보호” 의 마인드가 확산되면서 해킹이나 크래킹에 대비한 정보보호의 측면으로 많은 논의가 되고 정보가 공유되고 있지만 정작 매우 중요한 백업에 대한 구체적 방식 및 주의점에 대해서는 많은 정보가 공유되고 있지 않는 실정이다. 실무적으로 시스템을 담당하는 이들은 사전에 백업의 중요성을 인식하지 못하다가 중요한 데이터를 훼손당하거나 긴급하게 데이터를 복구하여야 할 필요성이 있을 경우에서야 비로소 백업의 중요성을 인식하여 소 잃고 외양간 고치는 격으로 백업을 실시하거나, 설사 백업을 하더라도 사전 지식이나 올바른 정책 없이 실시하여 매우 비효율적인 방법으로 백업을 하는 경우가 많이 있다.

따라서 이 섹션에서는 일반적으로 폭넓게 사용하고 있는 리눅스를 중심으로 보안이라는 화두와 결코 따로 갈 수 없는 백업의 방식과 백업시 주의점에 대해 알아보도록 하겠다.

① 백업정책 설정 시 고려할 점.

백업을 실시할 경우 어떤 방법으로 어떤 매체를 이용해 어떤 주기로 할 것인지 이른바 “백업정책” 을 설정하여야 하는데. 이때 고려하여야 할 점에 대해 알아보기로 하자.
(1) 호환성 - 자신이 운영하는 시스템에서 백업의 대상이 되는 데이터가 각기 다른 운영체제와 다른 시스템과 호환이 될 필요가 있다면 백업 후 데이터의 호환성이나 이식성을 고려하여야 한다.

(2) 자동 백업 - 사실 백업이라는 절차가 만일의 사고에 대비하여 꼭 필요한 절차이기는 하지만 손이 많이 가는 작업이기는 하다. 특히 백업을 하여야 할 대상이 여러 시스템이거나 하나의 시스템내에서도 일괄 백업이 아닌 일부의 데이터에 대해서만 백업을 할 필요가 있다면 일련의 백업작업의 자동화는 필수라고 할 수 있을 것이다.
(3) 비용 - 데이터의 유실이나 사고 발생시에는 백업의 소중함을 더 없이 인식하지만 평소에는 백업이 마치 사치라고 생각하는 사람이 있을 수도 있다. 따라서 가급적이면 적은 비용으로 백업을 할 수 있다면 좋을 것이다.

(4) 작업의 편리성 - 요즘에는 편리하게 백업정책을 설정하고 복잡한 설정도 쉽게 할 수 있으며 백업 상황등을 GUI 환경으로 볼 수 있는 백업유틸리티나 상용 솔루션들이 많이 나와 있다. 그러나 이러한 X-Windows 기반의 프로그램의 경우 일반 데스크탑에서는 유용하나 보안을 중요시하는 서버 시스템의 경우 X-Windows 자체를 사용하지 않는 경우가 있으므로 이 부분은 신중하게 검토할 필요가 있다.

(5) 네트워크 백업 - 이 부분은 자동백업과 함께 가장 중요한 부분중의 하나라고 판단된다. 일반적으로 백업은 백업매체의 자체백업(또는 1차 백업이라 한다.) 뿐만 아니라 일정 주기로 2차,3차 백업을 하게 되는데, 이러한 2차,3차 백업이 자동으로 되려면 네트워크를 통한 백업이 가능하여야 한다.
(6) 저장 매체의 선택 - 백업의 매체에 따라 저장방식이 다른 경우가 있으므로 신중히 선택되어야 한다. 고전적인 방식의 테이프나 추가의 하드 디스크, 다시쓸 수 있는 CD등 다양한 매체가 있을 수 있다. 위의 여러 가지 사항 및 자신이 관리하는 시스템의 특성을 고려하여 저장매체를 선택하여야 할 것이다.

권장할 만한 방식.

위의 여러 가지 상황을 고려하여 이식 성이 높고 여러 옵션이 지원되며 백업 파일을 이용

해 복구 시에서도 강력한 기능을 제공하는 tar를 이용하여, 하드 디스크에 저장하는 방식을 권장한다. tar는 "Tape ARchiver" 에서 나온 말에서도 알 수 있듯이 초기에 주로 테이프 백업을 위해 자주 쓰였던 유틸리티인데, 이는 리눅스를 포함한 변종의 유닉스 계열에서도 모두 사용이 가능하고 Shell Script 등을 이용하면 복잡해 보이는 백업정책도 쉽고 단순하게 세울 수 있으며 백업후 복구가 필요한 시점에서도 강력한 기능을 발휘할 수 있다.

그리고, 예전에는 주로 테이프 백업을 선호하였지만 요즘에는 고속의 고용량, 저가의 하드 드라이브가 보편화되면서 테이프보다는 하드 드라이브를 추가 장착하여 백업매체로 써 널리 이용하는 추세이다.

② 백업의 방식

그럼, 이제 구체적으로 어떻게 백업할 것인지 백업의 방식에 대해 이야기 해 보도록 하자.
먼저, 어떤 데이터를 어떤 주기로 백업할 것인가를 선택하여야 할 것이다.
모든 데이터를 매일 백업하는 것은 비효율적이므로 꼭 백업이 필요한 데이터만 적당한 주기로 백업하는 것이 불필요한 백업으로 인한 디스크 공간을 절약할 수 있고 백업작업으로 이한 필요이상의 프로세스를 줄일 수 있다.
백업을 할 대상은, 다른 서버에서 복사(이식)가 가능한 바이너리나 기본설정화일들은 굳이 백업할 필요가 없으며 각 시스템마다 고유한 시스템 설정(Configuration) 과 실제 데이터는 반드시 백업하여야 할 것이다.

일반적으로 리눅스의 경우 대부분의 설정은 /etc 디렉토리에 있으며, 각종 데이터는 /home 이하에, 그리고 기타 부가적으로 설치하는 프로그램은 /usr/local 에 있으니 자신의 시스템 고유의 설정을 고려하여 각각의 디렉토리이하에 대해 적절히 백업을 하면 된다. 이를테면 메일 서버의 경우 추가적으로 /var/spool/mail 이하의 데이터가 매우 중요하므로 백업시 추가하여야 할 것이다. 백업 디렉토리 설정시 주의할 점은 backup이 되는 파일을 다시 백업하여 무한루프에 빠지는 실수를 주의하여야 하고 실시간으로 시스템내 프로세스에 대한 정보가 저장되는 /proc 역시 백업을 하는 실수를 범하지 말아야 한다.

그럼. 예제를 통해 실제로 백업을 하는 명령어를 알아보도록 하자.

```
cd /backup
tar cvpfz /backup/home.tar.gz /home --exclude=/home/test
```

[스크립트 1]

위의 명령에서 사용한 옵션에 대해 하나씩 알아보도록 하자.
먼저 백업파일은 /backup 이라는 디렉토리에서 생성하기를 원하므로 /backup 으로 이동한후 tar 명령을 이용해 백업을 시작한다.

“c" 는 Create a new archive 의 뜻으로 백업시 새로운 파일을 생성하며
”v" 는 Verbosely list files processed 의 뜻으로 백업시 백업이 진행되고 있는 상황

및 디렉토리 리스트를 보여주며
“f" 는 archive file is local 의 뜻이며 명령어 이후의 이름이
생성될 파일의 이름이라는 뜻이며
“p" 는 Preserve-Permissions 의 뜻으로 이전 데이터의 Permission 정보를
그대로 보존한다는 뜻이며
"z" 는 filter the archive through gZip의 뜻으로 gzip 으로 압축한다는 뜻이다.
일반적으로 tar로 백업시 사용되는 확장자 규칙은 압축되지 않고 단순히 하나의 파일로 패킹(packing) 한 파일의 경우에는 “tar" 를, gzip 으로 압축된 파일의 경우에는 ”tar.gz" 나 “tgz" 를 확장자로 한다.
따라서 확장자가 home.tar.gz 나 home.tgz 일 경우 tar 로packing
(즉 단순히 하나의 파일로 묶기만 함) 된후 gzip 으로 압축되었음을 뜻한다.

[스크립트 1] 과 같이 실행되었을 경우에는
/backup 디렉토리에 /home/test 이하를 제외한 /home 이하의 모든 파일들이 home.tar.gz 라는 이름으로 생성된다. 제외할 디렉토리나 파일이 더 있을 경우에는 --exclude 옵션을 계속 이어서 쓰면 된다.
그리고 백업시 ``z'' (압축) 옵션의 사용에 유의하여야 한다. 단순히 하나의 파일로 압축백업시에는 때에 따라 데이터의 일부가 손상되는 경우가 있는데, 이때에는 백업된 파일 전체가 깨어져 복구를 할 수 없게 되는 경우가 있다. 백업시 압축을 하지 않고 단순히 tar로 Packing한(묶은) 경우에는 설사 파일이 깨어지거나 손상되었다 하더라도 파일의 복구가 가능하기 때문에 백업시 압축을 할 것인지 여부를 신중히 선택하여야 한다.
아울러 현재까지 최신 안정 커널인 Kernel 2.2 리눅스의 경우, 하나의 파일은 최대 2G가 최대 사이즈인데, 특정 파티션 이하에 많은 파일이 있어 하나의 파일로 백업하였는데, 2G를 넘을 경우에는 적당히 분산하여 백업을 하여야 한다.
이때에는 아래와 같은 방법이 가능할 것이다.

```
tar cvfpz home_a_h.tar.gz /home/[a-h]*
tar cvfpz home_i_p.tar.gz /home/[i-p]*
tar cvfpz home_q_z.tar.gz /home/[q-z]*
```

/home 이하의 디렉토리에 많은 하부 디렉토리 및 파일이 있을 때에는 전체 디렉토리를 백업할 경우 2기가를 초과하여 백업파일이 제대로 생성되지 않으므로, 이를 /home 이하의 디렉토리명이나 피일 이름의 이니셜로 구분하여 3등분하면서 백업을 하는 것이다.
그럼, 실제로 백업을 하는 스크립트를 분석해 보자.

```
#!/bin/sh ..... (1)
cd /backup ..... (2)
rm -f *.tar.gz ..... (3)
tar cvfpz apache.tar.gz /usr/local/apache/* ..... (4)
tar cvfpz home.tar.gz /home/* --exclude=/home/test ..... (5)
tar cvfpz etc.tar.gz /etc/* ..... (6)
tar cvfpz mail.tar.gz /var/spool/mail/* ..... (7) ,
chown backup.backup *.tar.gz ..... (8)
chmod 700 *.tar.gz ..... (9)
ls -alh | mail -s www50_backup backup@clunix.com ..... (10)
```

```
df -sh | mail -s www50_df backup@clunix.com     ..... (11)
```

[스크립트2]

(1) 번은 시작되는 내용이 Shell Script 임을 정의한다.
(2) 백업작업이 이루어지는 파티션인 /backup 으로 이동한다.
(3) 새롭게 백업을 하여야 하므로 기존의 백업된 파일을 모두 삭제한다.
(4) - (7) 번까지는 백업을 해야 할 디렉토리 이하에 대해 데이터를 각각 백업한다.
(8) 백업된 파일의 소유권을 backup 으로 설정한다.
(9) 백업된 파일의 권한을 700 으로 설정한다.
(10) 백업이 제대로 되었는지 확인하기 위해 현재의 디렉토리내 백업 파일의 리스트등의 정보를 백업담당자인 backup@clunix.com 에게 메일로 발송한다.
(11) 혹 파티션이 가득 찰수도 있으니 현재의 파티션 사용 정보를 백업담당자에게 메일로 발송한다.

별도의 설명이 필요하지 않을 정도로 간단한 스크립트이다.
Shell 스크립트는 실행되어야 할 명령어를 순서대로 나열하기만 하면 된다.
위 예에서는 Full Backup을 보여주고 있는데, 필요에 따라 적절히 “증가분 백업” 도 이용하기 바란다. 증가분(변경분) 백업은 풀백업을 받은 후 기존의 백업 데이터와 비교하여 이미 백업된 이후 변경되거나 추가된 부분만 백업하는 방식이다.

예를 들어 설명해 보도록 하자.
아래의 예는 백업을 단순화 하기 위해 /home 에 대해서만 백업하는 예를 들겠다.
백업정책은 일주일을 단위로 풀백업을 받고 이후 매일 증가분 백업을 하기로 한다.

```
#!/bin/sh                                    .....(1)
TODAY=`date +"%d %b %Y"`          ..... (2)
PREVIOUSDAY=`cat FULL_BACKED_DAY`    ..... (3)
tar cvfpzG _modified_home.tgz -N "$PREVIOUSDAY" /home .....(4)
```

[스크립트3]

먼저 위의 스크립트를 실행하기에 앞서 [스크립트 2]를 참고하여 일주일을 단위로(정기적으로 백업을 하는 방법은 이후에 설명할 CRON을 이용하면 된다.) /home 에 대해 Full Backup을 실행하기로 한다.(이때 백업 스크립트 제일 하단에
echo `date +"%d %b %Y"` > FULL_BACKED_DAY 를 추가한다)
이후 매일 위의 [스크립트3] 스트립트를 실행하면 풀백업이후 추가 및 변경된 파일과 디렉토리만 선별하여 별도로 백업을 하게 되는 것이다.

(1) 실행되는 스크립트가 Shell 스크립트임을 정의한다.
(2) 현재의 시각을 -N 옵션이 이해할 수 있는 13 Jan 2001 형태로 TODAY 라는 변수에 대입한다.
(3) 이전에 Full 백업시의 시각인 FULL_BACKED_DAY를 읽어 PREVIOUSDAY 라는 변 수에 대입한다.
(4) /home 디렉토리 이하에 대해 Full 백업시의 시각 이후로 변경, 추가된 파일에 대해서만 modified_home.tgz 라는 파일로 저장한다.

증가분 백업 스크립트는 다소 복잡하기는 하지만 각자의 환경에 따라 [스크립트 2] 스크립트와 적절히 혼합하여 사용한다면 프로세스의 유발을 최소화하여 유용하게 사용이 가능할 것이다.
그럼, 이의 스크립트를 매주 또는 매일 일일이 실행할 필요 없이 자동으로 실행할 수 있는 방법은 무엇일까? 이 물음에 해답을 줄 수 있는 것이 바로 cron을 이용하는 것이다.

cron 은 일종의 일정관리 데몬으로서 기본 설정파일인 crontab 에서 정해진 시각에 따라 주기적으로 명령을 수행한다. crontab 은 아래와 같이 7개의 구성요소로 이루어지는데, 6번째 필드(User)는 생략되어도 무방하다.
분 | 시 | 날짜 | 달 | 요일 | 사용자 | 명령어
각 항목은 정수로 나타낼 수도 있고 또한 몇 개의 항목은 와일드카드 문자로 인식되는 ``*'' 문자로 표현이 가능한데, * 는 "매" 의 의미이다. 즉 시간 항목에 * 이 있다면 매시간이라는 뜻이고 날짜 항목에 * 이 있으면 매일이라는 뜻이 되는 것이다. 그리고 하나의 필드에 중복된 시간을 나타내고자 하면 콤마로 구분하면 되고, 연속된 시간을 나타내고자 하면 하이픈(-)을 이용하여 일정 기간을 나타낼 수 있다.
참고로 분은 0-59, 시는 0-23, 날짜는 0-31, 달은 0-12(0또는 12는 12월, 1은 1월), 요일은 0-7(0과 7은 일요일, 1은 월요일)로 나타낸다.
그럼, crontab 파일을 열어보자.

```
01 * * * * root run-parts /etc/cron.hourly
02 2 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
0 0 * * 1 root rdate -s time.bora.net && clock -w
```

위 설정에서 각각의 의미를 알아보도록 하자.
첫줄은 매시 1분에 /etc/cron.hourly 디렉토리내의 스크립트를 실행한다는 의미이고
두번째줄은 매일 02시 2분에 /etc/cron.daily 내의 스크립트를 실행한다는 의미이며
세번째줄은 매월, 매일 새벽 4시 22분에 /etc/cron.weekly 디렉토리내 스크립트를 실행하는 것처럼 보이지만 5번째 필드인 <요일> 필드값이 *이 아닌 0 이므로 1주일에 1회 일요일(0이므로) 에 /etc/cron.weekly 디렉토리내의 스크립트를 실행하라는 의미이다.
네번째줄은 같은 방식으로 직접 해석해 보기 바란다.
다섯번째줄과 같이 매일, 매주 0시0분, 매주 월요일(1) 에 rdate -s time.bora.net &&

clock -w 라는 명령어를 실행하라는 형식으로 직접 명령어를 지정해 줄 수도 있다.
각 디렉토리내 실행 스크립트는 정기적으로 시스템에 의해 실행이 되어야 하므로 실행(eXecute) 권한이 있어야 함은 당연하다. 따라서 위에서 작성한 스크립트를 700 정도의 적당한 실행권한을 부여하여 해당 디렉토리에 설정해 주면 될 것이다.
또는 직접 /etc/crontab를 편집하여 실행시간을 정의할 수도 있다.

기타 tar나 crontab 에 대한 부가적인 명령어 옵셥은 man 페이지나 HOWTO 문서를 참고하기 바란다.

③ 2차 및 3차 백업에 대해

서버 자체에서의 1차 백업은 짧은 기간에 정기적으로 이전의 백업을 삭제하고 새롭게 백업을 받으므로 데이터의 훼손이나 삭제사실을 늦게 발견하였을 경우에는 이미 삭제된 데이터로 백업을 받아 복구할 수 없는 경우가 발생한다. 또는 자체 백업이 되는 서버에 장애가 있거나 백업 데이터 자체가 훼손이 되는 경우도 있다. 이러한 경우에 대비하여 반드시 2차백업을, 가능하다면 3차백업까지 설정 하여 운영하는 것이 필요한데, 2차 백업은 1차 백업의 내용을 ftp로 전송하거나, rsync를 이용하여 특정 디렉토리 이하에 대해 동기화를 하는 방법등이 있다. 일반적으로 백업 주기를 타이트하게 설정시 1차백업은 월,수,금 새벽에 진행하고, 2차 백업은 화,목,토 새벽에, 그리고 3차백업은 2차백업서버에 접속하여 수,일 새벽정도에 하는 것이 권장할 만한 방법이다.

i. FTP 를 이용한 2차 백업

ftp를 이용한 2차 백업서버로의 전송은 expect 스크립트를 이용해 ftp 작업을 자동화 할 수 있으며 ncftpget을 이용, 스크립트를 작성하는 방법도 있다.

```
## Expect script를 이용하는 방법

#!/usr/bin/expect                              .....(1)
spawn ftp data.tt..co.kr                       .....(2)
expect "Name :"                                .....(3)
send "backup₩r"                                .....(4)
expect "Password:"                             .....(5)
send "tomorrow₩r"                              .....(6)
expect "ftp>"                                  .....(7)
sleep 1                                        ..... (8)
send "bin₩r"                                   .....(9)
expect "200 Type set to I."                    .....(10)
sleep 1                                        .....(11)
expect "ftp>"                                  .....(12)
send "get /backup/home.tar.gz /home/pr/data_home.tar.gz₩r" .....(13)
expect " Kbytes/sec)"                          .....(14)
sleep 1                                        .....(15)
expect "ftp>"                                  .....(16)
send "quit₩r"                                  .....(17)
expect "221 Goodbye."                          .....(18)
interact                                       .....(19)
```

(1)먼저 expect 스크립트라는 것을 정의한후
(2)백업을 받으려는 서버인 data.clunix.com 에 ftp로 접속한다.
(3)"Name :" 이라는 응답이 오기를 기대(expect)한다.
(4) 데이터 서버에 접속하기 위해 username 인 backup 을 입력후 엔터를 입력한다.
(5) username을 입력했으므로 Password: 라는 반응이 오기를 기다린다.
(6) username 에 해당하는 암호인 tomorrow를 입력한다.

(이 부분은 각자의 환경에 따라 다를 것이다)
(7) ftp> 라는 반응이 오기를 기다린다.
(8) 1초간 대기를 한후
(9) binary 로 전송을 받아야 하므로 bin 입력을 한다.
(10) 서버로부터 200 Type set to I 라는 반응이 오기를 기다린후
(11) 1초간 대기를 한후
(12) ftp> 라는 반응이 오기를 기대(expect) 한후
(13) 원격지 데이터 서버의 /home/pr/data_home.tar.gz를 백업서버의
/backup디렉토리에 home.tar.gz 라는 파일로 저장한다.
(14) 사버로부터 Kbytes/sec) 라는 메시지가 출력되면
(15) 1초간 대기를 (sleep)한 후
(16) ftp> 메시지가 오기를 기다린 후
(17) quit를 입력하여 접속을 끊고
(18) 221 Goodbye 메시지가 나오기를 기대한 후
(19) 이후 명령어입력 정보를 기다린다.

만약 전송받는 파일의 사이즈가 클 경우에는 sleep 1 대신에
set timeout -1 를 지정하여야 정상적으로 ftp 전송이 가능하다.

그러나 expect 보다는 ncftpget을 이용하는 방법이 더 간단하고 빠르게 전송할 수 있다.

```
#!/bin/sh

rm -f *.tar.gz              .................(1)
ncftpget -u backup -p 'root///' data.clunix.com /home/data_backup ₩
 '/backup/home.tar.gz' .........(2)
```

이 두줄로 간단하게 처리가 되는데 이는
(1) 기존의 백업받았던 데이터를 삭제후
(2) ncftpget을 이용해 data.clunix.com 에 접속하여 /backup 디렉토리의 home.tar.gz 라는 파일을 백업서버의 /home/data_backup 디렉토리이하에 home.tar.gz 라는 파일로 ncftp전송한다는 뜻이다.
expect 보다는 ncftpget 이 좀더 빠르고 간단하므로 ncftpget을 사용하기를 권장한다.

ii. RSYNC를 이용한 2차 백업

ncftpget 과 비슷한 방법으로 rsync 라는 패키지를 이용하는 방법이 있는데, 이는 rpm 으로도 기본 제공되므로 쉽게 이용이 가능하며 백업뿐만이 아니라 Clustering 으로도 적용 가능하다.

설정방법은
(1) 먼저 데이터 서버의 /etc/inetd.conf 에
rsync stream tcp nowait root /usr/bin/rsync rsyncd --daemon를 추가 설정후
killall -HUP inetd 로 INETD를 재설정한다.
rsync 는 포트 873을 이용하며 --daemon은 데몬 모드로 작동한다는 뜻이다.

(2) /etc/rsyncd.conf 에 아래와 같이 설정한다.

```
[mail]
path = /backup
comment = mail 서버
uid = root
gid = root
use chroot = yes
read only = yes
hosts allow = 211.11.22.33
max connections = 1
```

[mail] rsync의 서비스명으로 어떠한 이름을 설정하여도 된다.
단, 백업서버에서 데이터 전송시 필요한 설정이다.
path 데이터를 동기화할(즉, 백업으로 전송할) 디렉토리를 설정한다.
여기에서는 /backup 이하의 디렉토리에 대해 2차 백업을 할 것이다.
comment 서비스명에 대한 설명이다.
uid 파일을 전송하는 사용자의 uid로 기본값은 nobody이다.
gid 파일을 전송하는 사용자의 gid로 기본값은 nobody이다.
use chroot 위의 path를 최상위 디렉토리로 사용하여 이외의 디렉토리는 접근할 수 없도록
하는 설정으로 보안상 필요하므로 꼭 설정한다.
read only 읽기전용모드로 설정한다는 의미이다.
hosts allow 기본값은 모든 접속을 받아들이므로 보안을 유지하려면 반드시 백업서버의 IP 를 설정하여야 한다.
max connections 백업서버에서 전송시 동시접속자수를 뜻한다. 1정도면 적당하다.
timeout 클라이언트에서의 접근시 타임아웃시간이다.

아래는 데이터를 백업할 2차 백업서버에서의 설정 내용이다.

```
#!/bin/sh ..... (1)
cd /mail_backup ..... (2)
rm -rf * ..... (3)
/usr/bin/rsync -av data.clunix.com::mail /mail_backup ..... (4)
chown root.root * ..... (5)
chmod 700 * ..... (6)
```

(1) shell 스크립트임을 정의한다.
(2) 백업을 받을 디렉토리인 /mail_backup 으로 이동후
(3) 기존의 백업받았던 데이터를 삭제한다.
(4) rsync을 이용해 서버에 접속하여 데이터를 받아온다.
이때 -a는 아카이브 모드로서 파일의 속성이나 퍼미션 및 소유권등의 정보
를 보존하여 동기화한다는 의미이며 -v 는 verbose 의 의미로 동기화의 진행상황을

자세하게 보여준다. 데이터 서버에서 서비스명을 [mail] 로 했으므로
data.clunix.com::mail 로 설정되었으며 동기화한 데이터는 /mail_backup 이라는
디렉토리에 저장된다.
(5),(6)은 위에서 설명되었으며 보다 세부적인 rsynhc에 대한 내용은
HOWTO를 참고하기 바란다.

2차 백업을 좀더 효율적으로 하기 위해서는 데이터 서버에 이더넷 카드를 2장 설치하여 eth0 은 일반 서비스로 이용하고 eth1은 사설 IP를 할당하여 백업서버와 Direct 로 연결하여 백업시 이용하는 방법도 있다. 이 경우에는 2차 백업시 별도의 이더넷 카드로 전송이 되므로 백업 작업시 발생하는 이더넷 카드에서의 트래픽이 서비스에 영향을 주지 않아 병목현상을 피할수 있고, 사설 IP를 이용하므로 백업서버의 보안도 일정 정도 강화할 수 있다.
단, 이러할 경우 백업서버는 데이터서버와 근거리에 위치하여야 하고, 사설 IP를 할당하므로써 백업서버에 직접 접근시 사설 IP로 접근하여야 하는 단점이 있기는 하다.

④ 복구의 방식

이제 어렵게 백업한 파일을 복구할 필요가 생기게 되었다.
백업 파일로부터의 복구방법은 그리 복잡하지 않은데, 복구시에는 백업때와는 달리 c 옵션대신 eXtract 의 의미인 x를 사용하는 것만 주의하면 된다.
예를 들어보면,
tar zxvpf /backup/etc.tar.gz 와 같이 할 경우 현재의 디렉토리에
압축된 파일에 들어있는 모든 파일을 복구하며 p옵션을 주었으므로 원래 파일의 소유자와 퍼미션도 그대로 복구된다.

백업 파일에 있는 모든 파일을 복구하는 것이 아니라 특정 디렉토리나 파일에 대해서만 복구하려면 아래와 같이 하면 된다.

tar zxvpf /backup/home.tar.gz etc/passwd home/clunix/public_html

현재의 디렉토리가 /home/user 라면 위와같은 경우 현재의 디렉토리에 원래
의 /etc/passwd 파일이 /home/user/etc/passwd 파일로 복원이 되고 원래
의 /home/clunix/public_html 이하의 파일들이 /home/user/home/clunix/public_html 로 복원되는 것이다. 특정 파일을 지정할 경우에는 특정파일이 복구되고, 디렉토리를 지정할 경우에는 디렉토리 이하의 디렉토리 및 파일들이 복원된다. 이때 주의할 점은 백업이 풀릴 파일을 지정시 /etc/passwd 와 같이 원래의 절대경로를 설정하면 백업에서 풀리면서 현재의 설정 파일에 백업본이 그대로 overwrite를 되므로 주의하여야 한다는 점이다. 따라서 복구시 / 를 입력하지 않는다.

⑤ 백업 및 복구 시 주의점

(1) 데이터의 변화주기 및 경중을 고려하여 백업주기를 다르게 설정한다.

백업할 데이터가 홈페이지 관련 데이터일 경우 일반적인 HTML 이나 관련 GIF등의 이미지 파일들은 자주 변경이 되지는 않으며 주로 작업자의 PC에서 작업후 서버에 업로드를 하므로 작업자의 PC에 저장되어 있는 경우가 많다. 그러므로 이의 데이터들은 매일 또

는 수시로 백업을 받을 필요는 없으며 변경된 내용만을 백업하거나 일정주기마다 백업을 하면 된다. 그리고 백업이 동작시 일반 데이터뿐만 아니라 zip 이나 asf 또는 mpeg 등과 같이 파일자체가 압축되어 있는 형식의 경우에는 파일자체의 사이즈가 크고 백업시 많은 부하를 유발하게 되므로 이러한 파일의 경우에는 꼭 백업할 필요가 없다면 데이터의 경중을 고려하여 백업에서 제외하거나 백업일정을 적당히 조정할 필요가 있다.
요즘은 회원제 운영이 보편화되면서 대부분의 홈페이지에서 각종 데이터베이스를 연동하여 사용하는 경우가 많은데, 데이터 베이스의 경우 회원의 ID / PW 등의 각종 회원 데이터가 보관되고 정보가 자주 변경되므로 수시로 백업을 할 필요가 있다.
각 데이터베이스 프로그램의 경우 별도의 백업 명령어(msqldump., mysqldump등) 가 있으므로 tar 가 아닌 dump 명령어를 이용하는 것도 권장할 만 한다.

(2) 1차 백업시 백업 파티션은 별도의 디스크로 구성한다.

시스템 인스톨시 하나의 물리적인 디스크를 여러 파티션으로 논리 분할하여 이용하는 경우가 많은데, 이때 백업 파티션의 경우에는 반드시 별도의 물리적인 파티션 분할을 이용하여 별도로 디스크를 구성하는 것이 중요하다. 이는 만약 데이터가 보관된 하드 디스크가 물리적인 결함으로 복구가 불가능할 경우 이 디스크에 있던 데이터는 물론이고 복구시 필요한 백업까지도 함께 복구가 불가능하여 백업으로서의 의미가 상실되기 때문이다.
일반적인 데이터는 빠른 연산이 필요하므로 SCSI 를 이용하고 백업디스크는 비용부담이 적은 고용량의 IDE 방식도 무난하다.

(3) 백업된 데이터에 대한 보안설정도 중요하다.
일반적으로 백업 스크립트는 root 소유로 작동하므로 백업된 파일이 생성될 때에는 root 소유로 백업 화일이 생성된다. 그러나 일반적으로 umask 는 022 로 설정되어 있으므로 생성된 백업파일은 644로 저장된다. 644일 경우 아래와 같이 일반 사용자가 백업된 파일에 접근하여 심지어는 접근이 불가능한 shadow 파일에도 접근이 가능하게 되어 보안상 문제가 될 수 있다.
따라서 백업 파티션을 별도로 구성하여 백업이 끝난 후에는 반드시 백업 파티션을 umount하여 일반 사용자가 접근할 수 없도록 하고 백업을 시작하기 전에 mount를 하면 될 것이다. 또는 생성된 파일의 권한을 700 정도로 제한하여도 된다. 그리고 백업작업 중간에 일반 사용자가 백업 파일을 복사할 수도 있으므로 백업시작 스크립트에 umask를 066으로 설정하여 생성되는 파일의 소유권을 600 으로 하고 백업이 끝난후에 umask를 다시 022로 재설정하여 일반 사용자의 비정상적인 접근을 차단할 수 있다.

```
[user@www user]$id

uid=500(user) gid=500(user)
[user @www user]$ cat /etc/shadow
cat: /etc/shadow: 허가 거부됨
[user@www user]$ ls -la /backup/etc.tar.gz
-rw-r--r--    1 root   root     954039 12월  12 04:22 /backup/etc.tar.gz
[user@www user]$ cp /backup/etc.tar.gz .
[user@www user]$ tar zxvfp etc.tar.gz etc/shadow
etc/shadow
[user@www user]$ cat etc/shadow
root:$1$V.120Zso3$cD/aUqQr2EBY0:11295:0:99999:7:-1:-1:134540356
bin:*:11266:0:99999:7:::
daemon:*:11266:0:99999:7:::
```

(4) 2차, 3차 백업서버는 가급적 네트워크에서 격리할 필요가 있다.
만약 1차 백업이 되고 있는 데이터 서버가 해킹을 당해 데이터가 유실되었다면 2차나 3차 백업이 되고 있는 서버의 데이터로 복원을 하여야 한다. 그런데, 만약 2차,3차 백업서버도 기존의 서버와 같은 설정으로 되어 있어 같은 방식으로 해킹을 당해 데이터가 유실되었다면? 해결책이 없는 것이다. 따라서 최후의 보루라 할 수 있는 2차,3차 백업 서버는 기존의 서버와 환경설정을 다르게 하고 관리자 암호등 접근권한도 다르게 설정 할 필요가 있다.
그리고 백업서버는 일체의 제공 서비스 없이 백업만을 담당하므로 불필요한 서비스는 띄워놓을 필요가 없으며 백업이 작동할 때 이외에는 네트워크에 연결되어 있을 필요가 거의 없으므로 백업이 끝난 후에는 네트워크에서 격리할 필요가 있는데, 이는 ipchains 등을 이용하여 자체 방화벽을 구성하여 외부에서의 접근을 일체 차단할 수도 있겠지만 다음과 같이 아예 네트워크에서 격리를 할 수도 있다.

백업 시작전 : /etc/rc.d/init.d/network start 로 Netwok 연결
백업 종료후 : /etc/rc.d/init.d/network stop 로 Network 차단.

2차,3차 백업서버는 네트워크에서의 격리뿐만 아니라 물리적으로도 격리 할 필요가 있다.
화재나 천재지변등 피치 못할 사정으로 사고 발생시 백업서버가 원 DATA가 위치한 서버와 공동으로 위치할 경우 돌이킬 수 없는 문제가 벌생할 수 있으므로 가능한대로 백업 서버는 다른 건물로, 같은 건물의 경우라면 다른층에 위치하는 것이 좋다.
부득이 같은 층에 있다면 멀리 위치시켜 예기치 못한 서고의 가능성을 분산시키는 것이 필요하다.

(5) 백업 담당자를 지정, 운영한다.

보안담당자도 없는데 백업담당자가 필요하겠냐고 생각할 수 있겠지만, 전담자까지는 없더라도 담당자는 지정을 하여 정기적으로 체크를 함으로써 보다 책임성 있고 신뢰성 있는 백업체계를 유지할 필요가 있다. 백업이 되고 있는줄 알고 복구를 할 일이 있어 백업서버를 살펴 보니 corn 데몬이 죽어 있다거나 설정한 스크립트가 실행권한이 없어 실행이 되고 있지 않는 경우도 있으니 백업 담당자가 수시로 체크를 하여야 한다. 백업 체크 일지를 두거나 정기적으로 체크하는 스크립트를 만들어 자동으로 체크여부를 알려주는 것도 좋은 방법이 될 것이다.

(6) tar 이외 cp를 이용하는 방법도 고려할 만 하다.
TAR를 이용하여 Packing 이나 압축없이 cp를 이용하는 방법도 있는데,
이는 일주일에 1회정도
cp -ap /etc /backup/ 와 같이 /etc 디렉토리 전체를 풀복사한후
cp -aufp /etc/ /backup/etc/ 로 풀복사 이후 새롭게 생성이나 변경, 수정된 파일이나 디렉토리만 복사하는 방법도 있다.

(7) 라우터와 스위칭의 설정파일도 백업을 잊지 않는다.

서버뿐만이 아니라 라우터와 스위칭의 Configuration File 도 설정 변경시 백업을 할 필요가 있다. 특히 라우터나 스위칭의 경우 설정 변경시 원치 않던 설정까지 변경되는 경우가 있고 라우터의 장애시에는 라우터 밑단에 연결되어 있는 모든 네트워크가 영향을 받으므

로 반드시 백업을 받아둘 필요가 있다. 다만 설정이 자주 변경되지는 않으므로 정기적으로 백업을 할 필요는 없고, 설정이 변경될 때만 그때 그때마다 설정파일을 Copy & Paste 로 복사를 해 두면 된다.

⑥ 실전 백업 예제

이 백업 스크립터는 총 3개의 스크립터가 필요하다.
shell script 에 대한 준비가 부족하다 생각하면 스크립터 경로를 변경하지 말길 바란다.

```
/root/bin/backup-sh
/root/bin/sysbkp
/root/mesg/backupdisplay
```

이 스크립터는 매일 중요 데이터를 백업 하되 이틀간의 자료만을 보관하는 정책의 스크립터 입니다.

vi /root/bin/backup-sh

```
#!/bin/sh

rdate -s time.bora.net

bkdir_name=`date +%Y%m%d`

# /dev/hdd 하드는 백업 하드를 의미한다. 즉 백업 하드 디스크가 계속 mount 되어있다면
# 실제 해커가 들어왔을때 백업 파티션에 있는 백업 자료까지도 손상시킬수 있기 때문에
# 백업 바로 전에 mount 시켜 주고 백업 작업이 완료 된후 반드시 umount 시켜 주도록 한다.

mount -t ext3 /dev/hdd   /backup

# 기존의 백업 자료를 삭제 한다. 즉 오늘 백업을 하면 실제 2일 전 자료를 삭제 하는 것이다.

rm -rf /backup/bk_*-1

# 어제 자료를 다른 이름으로 하루 더 보관하기 위한 방법으로 Name Label 에다 -1 를 붙여
# 디렉토리 명을 변경 한다.
mv /backup/* /backup/bk_${bkdir_name}-1

# 실제 시스템 백업 리스트 입니다. 각 해당 백업이 저장될 폴더를 생성함.

mkdir /backup/bk_${bkdir_name}
```

```
mkdir /backup/bk_${bkdir_name}/home
mkdir /backup/bk_${bkdir_name}/named
mkdir /backup/bk_${bkdir_name}/named/zone
mkdir /backup/bk_${bkdir_name}/passwd
mkdir /backup/bk_${bkdir_name}/apache
mkdir /backup/bk_${bkdir_name}/sendmail
mkdir /backup/bk_${bkdir_name}/mysql
mkdir /backup/bk_${bkdir_name}/real
mkdir /backup/bk_${bkdir_name}/etc
mkdir /backup/bk_${bkdir_name}/cron
mkdir /backup/bk_${bkdir_name}/root

# 사용자의 개별 데이터 백업 ( 홈디렉토리 백업 )

tar cvfpz /backup/home_a_h.tar.gz /home/[a-h]*
tar cvfpz /backup/home_i_p.tar.gz /home/[i-p]*
tar cvfpz /backup/home_q_z.tar.gz /home/[q-z]*

# 설정 파일 및 시스템 중요 파일 백업

# DNS 설정 파일 백업

cp -a /etc/named.conf /backup/bk_${bkdir_name}/named
cp -a /etc/hosts /backup/bk_${bkdir_name}/named
cp -a /etc/resolv.conf /backup/bk_${bkdir_name}/named
cp -a /var/named/* /backup/bk_${bkdir_name}/named/zone

# 메일 설정 파일 백업

cp -a /etc/mail/* /backup/bk_${bkdir_name}/sendmail

# 시스템 계정 설정 파일 백업

cp -a /etc/passwd* /backup/bk_${bkdir_name}/passwd
cp -a /etc/shadow* /backup/bk_${bkdir_name}/passwd
cp -a /etc/group* /backup/bk_${bkdir_name}/passwd

# Apache web service 설정 파일 백업

cp -a /usr/local/apache/conf/httpd.conf /backup/bk_${bkdir_name}/apache

# real media server 설정 파일 백업

cp -a /usr/local/real/rmserver.cfg /backup/bk_${bkdir_name}/real

# 시스템 자동설정(cron)에 대한 백업

cp -a /var/spool/cron/root /backup/bk_${bkdir_name}/cron

# 사용자 디스크 쿼터에 대한 설정 백업
```

```
cp -a /home/aquota.user* /backup/bk_${bkdir_name}/home

# 시스템 전체 설정 파일 백업

tar czvfp /backup/bk_${bkdir_name}/etc/etc.tgz /etc

# Mysql 데이터 백업 ( 근래에는 /usr/local/mysql/data 디렉토리를 사용하는 경우 있음 )

tar czvfp /backup/bk_${bkdir_name}/mysql/mysqldata.tgz /usr/local/mysql/var

# root 디렉토리 백업

tar czvfp /backup/bk_${bkdir_name}/root/root.tgz /root

# 1일 동안 수정된 파일 점검 -> 보안적인 의미

find /home  ₩! -type d -atime +1 -exec ls -l {} ₩; >
 /backup/bk_${bkdir_name}/home/today_access_file

umount /backup
```

sysbkp 스크립터는 위의 backup-sh 스크립터를 실행하고 백업 중에 발생한 오류 및 백업 결과 보고를 관리자에게 메일로 해주는 스크립터이다.

vi /root/bin/sysbkp

```
#!/bin/sh

data=`date +%Y%m%d`

/root/bin/backup-sh 2> /root/mesg/err_all
cat /root/mesg/err_all | grep -v "Removing leading" > err_chk
/root/mesg/backupdisplay > /root/mesg/backupnotice
mail -s "$data 백업 결과 통보" alang@clunix.com < /root/mesg/backupnotice
```

backupdisplay 스크립터는 백업 결과 보고서의 내용을 보고서 형태로 정리해 주는 스크립터이다.

vi /root/mesg/backupdisplay

```
#!/bin/sh

bk_data=`date +%Y년%m월%d일%H시%M분`
```

```
echo -n "
------------------------------------------------------------
*                                                          *
*       WELCOM TO SYSTEM MANAGER WEB SITE !!!!!!             *
*       SECURITY !! FAST !! GOOD !! WEBSERVER !!           *
*                                                          *
*       -------  http://www.clunix.com   --------          *
*                                                          *
------------------------------------------------------------

♥♥♥ ♡♥:*:....:*:♥♡♥:*:......:*:♥♡♥
:*:....:*:♥♡ ♥:*:......:*:♥♡

////////////////////  System user notice  ////////////////////

"
if [ ! -s /root/mesg/err_chk ]
then
    echo "시스템 백업이 $bk_data 에 무사히 마쳤습니다."
else
    echo "$bk_data 이루어진 백업에 오류가 발생하였습니다."
fi

echo -n "
////////////////////////////////////////////////////////////

/////////////////  what's error  ////////////////////////////
"
echo
/bin/cat /root/mesg/err_chk

echo -n "
////////////////////////////////////////////////////////////

:*:♥♥:*:....:*:♥♡♥:*:......:*:♥♡♥:*:..
♡♥:*:......:*:♥♡♥:*:....:*:♥ ♡♥:*:....*:
```

;; 워낙 옛날에 만든 스크립터이지만 지금에 와서 보니 너무 유치하다.

마지막으로 root 계정에서 /root/bin/sysbkp 스크립터를 자동 실행 하게 cron 에 등록한다.

crontab -e

```
0 3 * * * /root/bin/sysbkp
```

/etc/rc.d/init.d/crond restart

5. 문제 발생 시 응급 복구 방법

파일 시스템이 깨졌거나 해킹등의 문제로 정상 부팅이 안되는 경우 복구 방법 입니다. 실제 리눅스 커널로 부팅을 하는 것이 가장 중요하다. 부팅을 해서 시스템 상태를 점검해야 이 다시 설치를 할지 아님 일부 중요 파일의 백업을 이용하여 복구를 할지 결정이 가능하다. 먼저 시스템 부팅이 안된다. 이제 부팅을 시켜 보자

① 긴급 부팅 시키기

1장에서 설정한 bootloader 설정 내용도 참조하세요.

i. Linux 배포 CD를 통한 부팅

단 이때는 기본적으로 시스템의 하드의 파티션 중 / 파티션의 디바이스 명을 알고 있어야 한다. 1장에서 언급한바와 같이 그래서 초기 셋팅 후 시스템 정보를 보관하는것은 상당히 중요하다.

Linux Install CD 를 넣고 부팅 한다. boot 프롬프트에 다음내용을 기재한다.

boot : linux initrd= root=/dev/hda2 -> /dev/hda2 가 실제 / 파티션의 device name

이는 Linux Install CD 에 있는 기본 커널로 시스템을 부팅하여 현 시스템의 root mount point (/) 에 접근하는 것이다.

이경우는 IDE 하드에 리눅스가 설치 된 경우 주로 사용한다. SCSI 의 경우 Install CD 커널에 SCSI 모듈 정보가 포함안되어져 있어서 root device 를 못찾는다면서 실패할 가능성이 많다.

주로 Lilo 가 깨졌을때 많이 사용한다.

ii. rescue mode booting

역시 Linux Install CD 를 이용하여 부팅할때 boot 프롬프트에서 rescue 라는 옵션과 같이 부팅

boot : linux rescue

그럼 text 설치 과정과 비슷하게 넘어가다가 bash prompt (#) 바로 떨어진다.

하드디스크에 크게 문제가 없을 경우에는 /mnt/sysimage 폴더에 system root 파티션이 mount 되어져 있을것이다.

chroot /mnt/sysimage

하면 마치 정상적으로 부팅 한것 처럼 mount 가 되어진다.

이상태에서 정검하면 됩니다. 진짜 중요한 파일에 문제가 있어 이것이 안된다고 하면

수동으로 mount 하여 중요파일의 손상 정도를 파악한다.
rescue 방식은 주로 SCSI 하드 디스크 일 경우 많이 사용함.

주로 Lilo 깨졌을때, 파일 시스템 깨졌을때, 완전히 망가 졌을 때

iii. single mode 로 부팅하기

파일 시스템에 문제가 발생하여 파일 시스템을 체크하기 위해 부팅하는 모드로 LILO Prompt 가 뜨면 ..

Lilo : linux single

위와 같이 부팅한다. 그런 후 안정스럽게 파일 시스템을 점검 및 복구 한다.

이밖에 파일 시스템 복구 목적으로 긴급 부팅 하는 방법으로는

lilo: linux init=/bin/sh

도 있다.

② 부팅 디스켓 만들기

먼저 부팅디스켓으로 사용할 커널 이미지를 선택한다.
그리고 선택한 커널 이미지가 root 장치(root mount point)가 재대로 잡혀 있는지 확인한다.

```
# rdev /boot/vmlinuz
Root device /dev/hda1
```

만일 루트 장치가 현재 잡혀 있는거랑 틀리다면 아래와 같이 수정한다.

```
# rdev /boot/vmlinuz /dev/hda3
#
```

이제 깨끗한 플로피 디스켓 하나 준비하고 포맷한다.

```
# fdformat /dev/fd0
# dd if=/boot/vmlinuz of=/dev/fd0 bs=8192
```

③ 파일시스템 점검 수리

```
# fsck -t [type] [device]
```

fsck 파일 점검은 가능한 umount 한 후에 실행한다. 하지만 / 을 mount 하지 않은 면 당근 부팅이 제대로 될수 없기에 / 를 체크할때는 플로피로 부팅을 하던지 아님

single mode 로 부팅한후 체크 한다.

체킹후 mount -t -w -o remount /

해주면 된다.

④ RPM 패키지 상태 초기화 하기

누가 해킹 혹은 실수로 RPM 패키지의퍼미션이나 권한등을 변경하여 정상적인 작동을 안할 경우 이를 설치 시 상태로 복구 하는 방법이다.

rpm --setperms -a

이로써 장작 3일간의 "철학이 있는 리눅스 설치" 의 모든 내용을 마치도록 하겠습니다.