# Backup

김건우 하정호 홍영규

# 백업, 왜 해요?

지금 서버에 들어가서 rm -rf를 해 보면 백업을 왜 하는 지 알 수 있을 겁니다

# 백업, 왜 해요?

- 하드웨어에 물리적인 문제가 생겼을 때
- 실수로 파일을 날렸을 때
- 컴퓨터를 누가 훔쳐 갔을 때
- 랜섬웨어에 걸렸을 때

# 백업의 종류

- Full backup
- Incremental backup
- Differential backup
- Mirror backup

# 백업의 종류

| | 백업되는 데이터 | 백업 시간 | 복구 시간 | 저장 공간 |
|---|---|---|---|---|
| **Full backup** | 전부 | 매우 느림 | 빠름 | 높음 |
| **Incremental backup** | 이전 백업 이후로 추가/수정된 내용 | 빠름 | 느림 | 매우 낮음 |
| **Differential backup** | 최근 Full backup 이후로 추가/수정 내용 | 보통 | 빠름 | 보통 |
| **Mirror backup** | 최근 Full backup 이후로 추가/수정 내용 | 가장 빠름 | 가장 빠름 | 매우 높음 |

# Full backup

- 백업할 때마다 모든 내용을 백업
- 내용을 전부 압축하여 하나의 파일로 보관
- 시간 소모 많음, 복구 간단함, 해킹 당하면 모든 내용을 내어주니 암호화 필요
- 다른 종류의 백업을 위해서 주기적으로 해 줘야 함

# Differential backup

- 마지막 full backup 이후 바뀐 내용만 백업
- 백업 속도 느림, 복구 속도 빠름, 차지 용량 중간

# Incremental backup

- 마지막 백업 이후 바뀐 내용만 백업
- Apple의 Time Machine
- 적은 내용을 백업하므로 백업 시간이 적음
- 복구할 때는 여러 백업 파일을 각각 백업해야 하므로 느림

# Mirror backup

- Full backup 이후 변경 내용만을 저장
- 파일 내용을 그대로 저장 — 보안에 취약
- 압축과 암호화가 없어서 차지하는 용량이 매우 큼
- 백업 속도와 복구 속도가 가장 빠름

# 백업하기 — tar

```
$ sudo mkdir backups

$ cd backups

$ sudo tar -cvpf /backups/full-backup.tar
  —directory=/ —exclude=proc —exclude=sys —exclude=dev/pts
  —exclude=backups .
```

**-c:** tar로 묶는다
**-v:** 압축 과정을 터미널에 출력한다
**-p:** 파일 권한을 함께 저장한다
**-f:** 파일 이름을 지정한다

# 백업하기 — gzip, bzip2

$ sudo mkdir backups

$ cd backups

$ sudo tar **-zcvpf** /backups/full-backup.tar**.gz (or .bz2)**

—directory=/ —exclude=proc —exclude=sys —exclude=dev/pts

—exclude=backups **.**

bzip2가 gzip에 비해 압축 및 풀기 속도는 낮으나 더 용량이 작아짐

# 원격 백업하기

- 같은 서버에 저장된 데이터는 동시에 손상될 확률이 더 높다
- 따라서 백업을 다른 서버에 저장하는 것이 낫다
- 백업 서버를 따로 운용하여 백업 파일을 저장할 수 있다

# rsync

- 원격 파일 복사 프로그램
- 소스와 대상 파일을 비교하여, 변경된 내용만 전송하기 때문에 자료 전송량 최소화
- **빠르다!**
- 복사의 방법에 대한 다양한 옵션이 존재 (다음 슬라이드)
- rsync [옵션] [소스 (보낼 파일)] [대상 (받을 위치)]

# rsync 옵션

- **-v** or **—verbose** 자세하게 출력
- **-q** or **—quiet** 어떤 메시지도 출력하지 않음 (에러 포함)
- **-a** or **—archive** 아카이빙 (위치, 권한, 소유주 포함하여 가져옴)
- **-r** or **—recursive** 하위 구조의 디렉토리도 recursive하게
- **-z** or **—compress** 압축해서 전송

# rsync로 백업하기

$ sudo apt-get install ssh rsync

$ ssh-keygen -t rsa (RSA 타입으로 public key와 private key를 발급)

$ scp .ssh/id_rsa.pub jambo@wheels1.sparcs.org:~/.ssh/authorized_keys

$ rsync -avz —progress -e ssh /backups jambo@wheels1.sparcs.org:backups

# rsnapshot

- rsync 기반의 파일 시스템 백업 유틸리티
- Incremental backup를 채용하여 용량을 적게 차지함
- 쉽게 설정할 수 있음

# rsnapshot으로 백업하기

$ sudo apt-get install rsnapshot

$ sudo vi /etc/rsnapshot.conf

# rsnapshot config

- snapshot_root: 백업이 저장될 폴더의 절대 경로
- cmd_ssh를 uncomment하면 원격 백업이 가능
- retain: 해당 이름의 백업을 n개까지 유지
- verbose, loglevel: 출력 메시지와 로그를 어느 정도까지 상세하게 기록할 것인지

# rsnapshot 백업

$ sudo rsnapshot -v [backup name] 처럼 사용

Configuration에서 설정한 snapshot_root에 alpha.0, alpha.1, …로 저장

(작은 숫자일 수록 최신)

각 백업들이 hard link로 되어 있어 alpha.0만 복원해도 나머지를 모두 복원한다.

# rsnapshot 백업 복원

$ mkdir here

$ sudo cp -r /var/cache/rsnapshot/alpha.0 here

# 코드 치기 귀찮죠?

귀찮을 땐 뭐다?

# 코드 치기 귀찮죠?

# 쉘 스크립트를 쓰자.

실습: 실행하면 백업을 진행하고 서버로 백업 파일을 넘기는 스크립트를 짜보자.

# 정기적인 백업

1시간마다 정기적으로 백업을 하려면

휠장이 1시간마다 서버에 접속해서 쉘 스크립트를 실행하면 되겠죠? ^__^

# cron

자세한 내용은 Shell Scripts / Cron 세미나를 참고합시다!

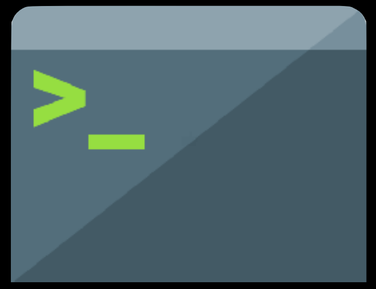
# Tips

- 오래된 백업 파일은 용량 절감을 위해 삭제 (혹은 시간 간격을 넓게 해서 남김)
- 백업 디스크를 읽기 전용 / 하드웨어로 쓰기 및 변조 방지 설정
- 백업 디스크를 만든 후 백업 전 mount, 백업 후 unmount 하는 방식으로 rm -rf 공격을 막을 수 있음!

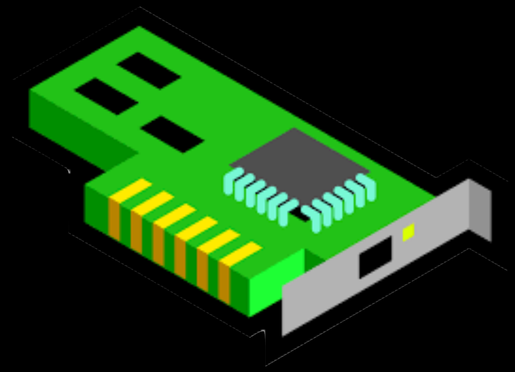
# 비상사태 관리

대비 및 대책 방법

# 세 종류의 문제들



Software



Hardware

Human

# Software Failure

- Internal
  - 파일 시스템 에러
  - 장치 설정 에러
  - 부팅 에러
  - 기타 프로그램 에러
  - 커널 패닉 (블루스크린)
  - 메모리 오버플로우
  - etc.
- External
  - 해킹
  - 악성 코드, 바이러스
  - 접속자 폭주
  - etc.

# Kernel Panic

- 운영 체제가 치명적인 내부 오류를 감지하여 복구가 불가능할 때 취하는 동작.
- 장치 드라이버 문제, 메모리 문제, 오작동의 누적 등으로 인해 생긴다.
- ex) 블루스크린

# Hardware Failure

- 깨진 파일 시스템 <- 가장 흔하다
  - e2fsck 명령어로 파일시스템을 점검 및 복구
- /etc/fstab 에서 장치명을 잘못 지정한 경우
  - Linux 설치 후 Linux Secure로 부팅한 후 수정 -> 재부팅

# 부팅이 안된다면...

- 미리 설정해둔 CD/DVD나 USB를 이용해 부팅한다.
  - 설치 CD에 Rescue 모드가 있어서 CD/DVD가 더 좋다.
  - 그 후 GRUB 등 부팅 관련 파일 복구.

# 서버에서 한글이 안나온다면...

- 일반적으로 SSH에서 나오던 한글도 콘솔에서 작업하면 안나오는 경우가 많다.
- #EXPORT LANG=C 를 통해 LANG을 ko_kr 에서 디폴트인 C로 변경하면 영어로 출력된다. (한번의 세션에서만 적용된다.)

# e2fsck

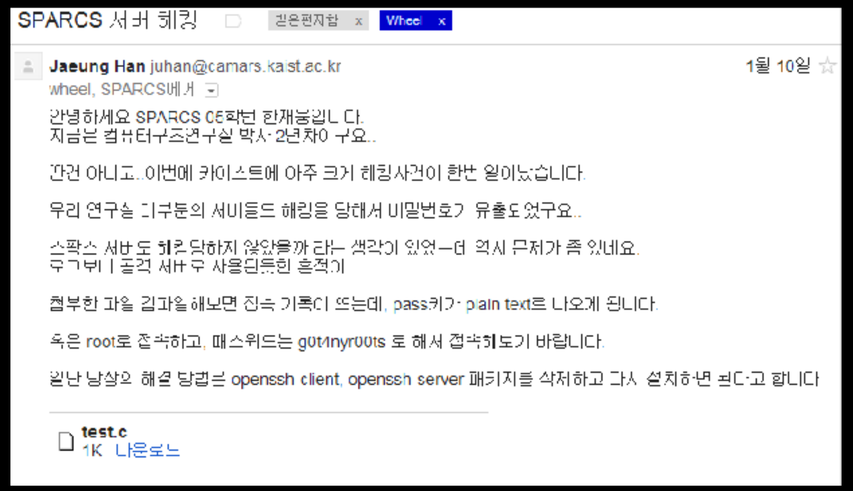
- 리눅스 파일 시스템을 점검 및 복구할 수 있는 명령어
- fsck 명령어의 확장판
- 기본적으로 점검하는 목록
  - inodes, blocks, sizes, 디렉토리 구조, 디렉토리 연결성, 파일링크, 전체 파일 개수, 전체 블록 중 사용중인 블록
  - $ e2fsck [options] [검사할 디바이스 이름]
  - 해당 파일시스템을 언마운트시킨 상태에서 수행하는 것이 좋다.

# 해킹

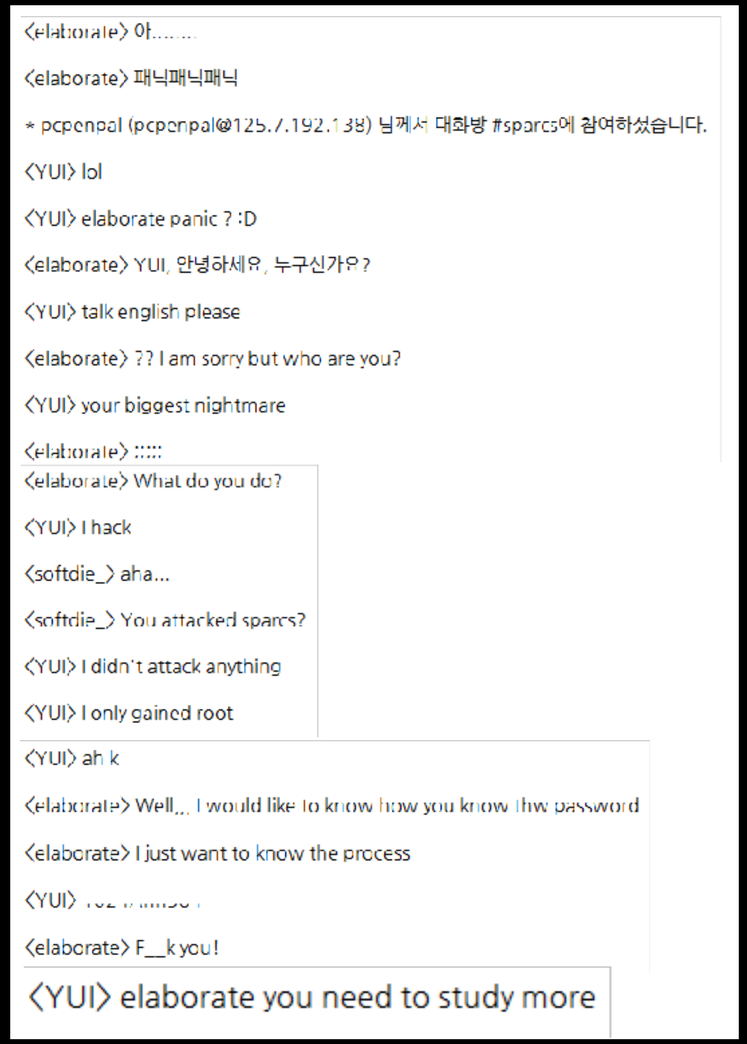
- 해킹을 당하면 대부분 명령어를 못쓰게 하거나 중요 파일 삭제, 변조를 일으킨다. 또, 비밀번호 에러가 뜨게 바꾸기도 한다.
- 미리 백업해둔 시스템 코어로 대체하여 해결하자!
- 예방: 안쓰는 Port 닫기, 의심가는 Process 죽이기, 주기적으로 프로그램 Update, 모의 해킹, Root 계정 Logout

# 2012 KAIST/SPARCS 해킹 사건

- 2012년 1월 10일.
- KAIST 연구실들이 해킹당함. 스팍스도?!
- /var/run/sshd.sync/에 사용자 비밀번호가 plain text로 저장.
- root 아이디에 g0t4nyr00ts 를 치면 접속이 된다.



# 2012 KAIST/SPARCS 해킹 사건




<YUI>란 아이디의 해커가
스팍스 IRC에 접속

# 2012 KAIST/SPARCS 해킹 사건

- 해킹 루트 분석:

  zeroboard4 취약점으로 root획득

-> 획득한 서버에 백도어 설치

-> 획득한 서버의 sshd를 변조

-> ssh 사용자의 id와 pw를 획득

-> 다른 서버에 같은 작업 반복

# Hardware Failure

- Internal
  - 랜선 고장
  - 케이블 절단
  - 전원장치 고장
  - 파워 이상
  - 냉각 이상
  - 특정 부품의 파손
  - etc.
- External
  - 먼지
  - 서버 앞에서 물총싸움 하기(?)
  - 충격
  - 화재
  - 홍수
  - 정전
  - etc.

# Hardware Failure

- 단순 접촉 불량, 랜선 고장
  - 바꿔끼우고, 접촉부위 점검
- 먼지는 주기적으로 털어주자
- 시끄러운 삐삐 소리는 대체로 냉각/전원 문제
- 안되면 현질하거나 수리기사님께 부탁하자

# 정전

- 정전 발생시 당황하지 말자.
- 서버실에는 UPS(무정전전원공급장치)가 있어서 어느정도의 시간의 서버들이 멈추지 않고 작동할수 있도록 되어 있다.
- 서버를 종료하기 전에 되도록 SPARCS로 서비스 중단 메일공고를 하고, 서버들을 차례대로 종료하도록 하자.

# 서버실 과열

- 서버실의 온도는 겨울철 26도, 여름철 33도에서 37도에 이르기까지 좀 높은 편이다. 서버실에 온도계가 설치되어 있으니 자주 확인할 수 있다.
- 서버실 온도가 높다면 일단 서버실에 들어가, 에어컨의 작동여부를 확인해보고, 에어컨이 고장난 경우, 리모컨으로 전원을 다시 켜보고 온도를 최저로 낮춰보자. 가끔 멀티탭에 문제가 있는 경우도 있다.
- 에어컨에 문제가 있다면 시설팀에 연락하고, 수리를 요청하자.
- 서버실의 문을 열고, 선풍기등을 이용해 열을 빼내도록 하자. (단 보안에 신경써야 한다)
- 비상시, 중요하지 않은 서버들 (다래, ftp2, mir 등)을 종료하여 최대한 발열량을 줄이자.

# Human Failure

- Internal
  - 관리자의 실수
  - 잘못된 입력, 오타
  - 음모와 계략
  - etc.
- External
  - 도둑
  - 해커
  - 악의적 사용자
  - 야옹이
  - etc.