실습공격 정리

2024년 5월 9일 목요일 오후 5:20

1.HTTP Get Flooding 공격 : 서버에 전달되는 HTTP Get 패킷이 너무 많아서 다른 클라이언트 시스템이 해당 서버에 접속할 수 없게 하는 공격.

-정상적인 과정 : 사용자가 특정 URL에 접속하려고 하면 해당 URL을 가진 웹 서버 웹 서버에 TCP 연결 후 HTTP Get 패킷을 전송한다.
이를 전달 받은 웹 서버는 해당 URL에 대한 자료(HTML 문서 등)를 패킷에 넣어서 응답한다.

-공격 원리

*(주의) 웹 서버는 매 순간 처리 가능한 HTTP 패킷의 양이 정해져 있다.
*양을 초과하는 대량의 패킷이 웹 서버에 유입되면 정상적인 서비스가 어려워진다.
*공격자는 웹 서버의 처리 용량을 초과하여 HTTP get 패킷을 전송한다.
*웹 서버는 각 HTTP Get 패킷에 대해 응답하기 위해 시스템 자원을 모두 소비하여 다른 패킷을 처리하지 못하는 상태가 된다.
*사용자는 웹 서버에 웹 페이지의 정보를 정상적으로 요청할 수 없게 된다.

-대응 방안

*방화벽을 이용하여 HTTP Get 패킷에 대해 임계값을 설정함으로써, 일정량 이상의 HTTP Get 패킷이 수신되면 이를 차단할 수 있다.

2.HTTP CC 공격 : 서버에 전달되는 HTTP Get 패킷에 캐싱 장비가 응답하지 않도록 설정하여 웹 서버의 부하를 증가시켜서 다른 클라이언트 시스템이 해당 서버에 접속할 수 없게 하는 공격

-정상적인 과정

*캐싱 장비는 사용자가 요청한 정보가 자신의 장비에 미리 저장되어 있지 않는 경우에는 웹 서버로 요청을 전달하여 응답을 받아온다.
*그러나 캐싱된 자료에 대해서는 해당 패킷을 웹 서버에 전달하지 않고, 자신의 장비에 저장된 자료를 이용하여 응답함으로써 웹 서버의 부하를 감소시킨다.
*그러나 HTTP 패킷의 헤더 내의 Cache-Control 값을 no-store(캐시 저장 금지) 또는 must-revalidate (캐시 검증 요구)로 설정하면 캐싱 장비가 응답하지 않고 웹 서버가 응답하도록 할 수 있다.

-공격 원리

* 공격자가 Cache-Control로 인해 캐싱 장비가 응답하지 않도록 설정된 다수의 HTTP Get 패킷을 서버로 보낸다.
* 이 경우, 캐싱 장비가 해당 자료를 저장하고 있더라도 응답하지 않고 웹 서버에 전달한다.
* 웹 서버는 각 HTTP Get 패킷에 대해 응답하기 위해 시스템 자원을 모두 소비하여 다른 패킷을 처리하지 못하는 상태가 된다.
*사용자는 웹 서버에 원하는 웹 페이지의 정보를 정상적으로 요청할 수 없게 된다.

-대응 방안

*캐싱 장비 또는 그 앞 단에 방화벽을 설치하고, 캐싱 장비를 이용할 수 없게 하는 문자열이 포함된 패킷에 대해 임계값을 설정하여, 일정량 이상의 공격 패킷을 차단토록 설정한다.

3.DNS 질의 및 응답 과정

- 로컬 DNS cache memory
- /etc/host/(Linux)
- 로컬 영역에 있는 DNS 서버
- 로컬 영역 밖에 있는 DNS 서버(가입된 ISP업체)

4.DNS Amplification 공격 : 공격자가 다수의 DNS 서버에 질의 패킷을 보내면서 특정 시스템에 응답 패킷이 보내지도록 조작함으로써 해당 시스템이 다른 패킷을 처리하지 못하게 하는 공격

-정상적인 과정 : DNS 서버는 DNS 클라이언트의 질의 요청을 받아 해당 서버의 도메인 이름을 네트워크 주소로 바꾸거나 그 반대의 변환을 수행한다.

-공격 원리

*공격자는 다수의 DNS 서버(네트워크 성능이 좋고, 데이터가 많을 수록 좋음)에 DNS 질의 패킷을 전송한다. 단, 출발지 IP를 공격 대상인 서버의 IP로 수정한다.

*이 질의 패킷을 수신산 각 DNS 서버는 출발지 IP로 응답 패킷을 전송한다. 출발지 IP는 서버의 IP로 수정되어 있으므로 다수의 DNS 서버로부터의 대량의 응답 패킷이 서버로 전송된다.

*모든 네트워크는 매 순간 최대로 전송할 수 있는 패킷의 양(대역폭)이 정해져 있으므로 대량의 응답 패킷으로 인해 네트워크의 대역폭이 부족해져서 다른 패킷의 전달이 지연되거나 누락될 수 있다.

*그 결과, 시스템 A는 서버와 통신할 수 없게 된다.

-공격 과정

*공격자는 다수의 DNS 서버(네트워크 성능이 좋고, 데이터가 많을수록 좋음)에 DNS 질의 패킷을 전송한다. 단, 출발지 IP를 공격 대상인 서버의 IP로 수정한다.

*이 질의 패킷을 수신한 각 DNS 서버는 출발지 IP로 응답패킷을 전송한다. 출발지 IP는 서버의 IP로 수정되어 있으므로 다수의 DNS 서버로부터의 다량의 응답 패킷이 서버로 전송된다.

*모든 네트워크는 매 순간 최대로 전송할 수 있는 패킷의 양(대역폭)이 정해져 있으므로 대량의 응답패킷으로 인해 네트워크의 대역폭이 부족해져서 다른 패킷의 전달이 지연되거나 누락될 수 있다.

*그 결과 시스템 A는 서버와 통신할 수 없게 된다.

-대응 방안

*출발지 라우터에서 IP Spoffing을 검사함으로써 출발지 IP가 변조된 공격 패킷이 전송되는 것을 차단하거나, 내부 네트워크에 유입되는 DNS 응답 패킷에 대해 임계값을 설정하여, 일정량 이상의 공격 패킷을 차단하도록 설정한다.

5.SIP Flooding 공격

*정의 : SIP * PROXY 서버에 전달되는 SIP INVITE request 패킷이 너무 많아서 해당 서버에 속한 다른 시스템들과 통신할 수 없게 하는 공격

*정상적인 과정 : 시스템 A는 시스템 B에 SIP INVITE request 패킷을 보내서 시스템 B 와의 통신을 요청한다.

*SIP proxy 서버는 자신에게 연결된 시스템으로부터 수신된 요청을 알맞은 Proxy 서버에 전달해준다.

*요청을 받은 시스템 B는 해당 연결 요청에 대한 응답으로 SIP INVITE response 패킷을 보낸다.

*이후 시스템 A와 시스템 B는 서로 통신이 가능하다.

-공격 원리
*모든 네트워크는 매 순간 최대로 전송할 수 있는 패킷의 양이 정해져 있다.
*공격자가 다수의 SIP INVITE request 패킷을 발생하여 B로 보낸다.
*시스템 B는 각 SIP패킷을 처리해야 하며, SIP 패킷이 아주 많으면 시스템 자원을 모두 소비하여 다른 패킷을 처리하지 못 하는 상태가 될 수 있다.
*또한, SIP Proxy 서버 간의 네트워크의 대역폭 제한을 초과함으로 인해 다른 패킷을 전달하지 못하는 상태가 될 수 있다.
* 그 결과, 시스템 A는 시스템 B에 패킷을 정상적으로 전송할 수 없게 된다.

-대응 방안
*인증된 사용자만 SIP 연결을 수립할 수 있도록 하거나, 네트워크에 유입되는 SIP 패킷에 대해 임계값을 설정하여, 일정량 이상의 공격 패킷을 차단하도록 설정한다.

6.UDPFlooding공격:서버에 전달되는 UDP 패킷이 너무 많아서 다른 클라이언트 시스템이 해당 서버에 접속할 수 없게 하는 공격

-정상적인 과정 :
*UDP는 TCP와 달리 연결을 설정하지 않고 바로 데이터를 전송한다.
따라서 전송 속도가 빠르며, 오류가 있으면 ICMP 등으로 보완한다.
*다음 그림은 UDP를 통한 데이터 전송이 실패한 경우(해당 UDP 패킷을 처리할 적절한 응용 프로그램이 서버에서 동작하지 않음)을 보여준다.

-공격 원리
*공격자가 다수의 UDP 패킷을 보내면, 서버는 각 UDP 패킷을 처리할 응용 프로그램을 검색한다.
*UDP 패킷이 아주 많으면 이 단계에서 이미 시스템 자원을 모두 소비하거나, 네트워크의 대역폭 제한으로 인해 다른 패킷을 처리하지 못하는 상태가 될 수 있다.
*공격자가 보내는 UDP 패킷은 적절한 응용 프로그램이 없는 등 오류가 발생할 수 있는 UDP 패킷이기 때문에 서버는 ICMP Unreachable 패킷을 응답하게 되고, 이로 인해 네트워크 부하는 더 커지게 된다.
*그 결과 시스템 A는 서버에 패킷을 정상적으로 전송할 수 없게 된다.

-대응 방안 : Snort나 방화벽을 이용하여 임계값 이상의 UDP 패킷을 차단할 수 있다.

7.ICMP Flooding 공격: 시스템에 전달되는 ICMP* 패킷이 너무 많아서 이를 처리하기 위해 사용 가능한 자원을 모두 소비함으로써 다른 시스템이 해당 시스템에 패킷을 전송할 수 없게 되는 공격
대표적으로 Ping of Death 와 Smurf 공격이 있다.

-Ping of Death(공격 과정)
*정상적인 과정 : Ping을 사용하여 특정 호스트를 향해 ICMP echo request 패킷을 전달한다.
*이를 전달 받은 시스템은 ICMP echo reply 패킷으로 응답한다.

-환경 구성 및 실습 개요
*Ping of Death 공격을 직접 실험해 보기 위해서는 3대 이상의 컴퓨터가 포함된 환경이 필요하다.

이 환경을 구축하기 위해서는 아래와 같이 2가지 방법이 있다.

1)3대 이상의 PC를 공유기에 연결시킨다.

2)Vmware에서 3개 이상의 가상 머신을 만들고, 각 가상 머신들을 Host-only로 연결한다.

*최근의 운영체제는 다수의 ICMP 패킷을 무시하도록 설정되어 있기 때문에, 윈도우 95나 98 또는 리눅스 6.0 이하의 버전을 사용해야 한다.

*공격자가 특정 호스트에 대한 다수의 ICMP echo request 패킷을 생성하기 위해 Ping 명령을 사용할 때, -1 옵션을 사용하여 크기를 크게 하거나, Hping3등의 공격 도구를 이용할 수 있다.

-Ping of Death(공격원리)

*공격자가 Ping을 이용하여 매우 큰 ICMP echo request 패킷을 대량으로 B에게 보낸다.

*이 큰 패킷은 네트워크에서 다수의 작은 패킷들로 나누어져서 B에게 전달된다.

*B는 작게 나누어진 모든 ICMP echo request 패킷을 받아서 재조립해야 하며, 이에 대한 ICMP echo reply 패킷을 전송해야 한다.

*하지만, 공격자가 큰 ICMP echo request 패킷을 대량으로 보냄으로 인해, 시스템 B는 이 패킷들을 처리하기 위해 시스템 자원을 모두 소진한 상태이기 때문에, 다른 패킷을 처리하지 못하는 상태가 된다.

*그 결과, 시스템 A는 시스템 B에 패킷을 정상적으로 전송할 수 없게 된다.

-Ping of Death(공격 과정)

*icmp 프로토콜을 사용하여 출발지 ip주소를 랜덤하게 생성한 후 특정 주소로 65000바이트의 패킷을 flooding 방식으로 공격

*패킷 분석 결과 다수의 65000 바이트의 echo ping request 발견됨

-Smurf공격 원리)

*공격자가 ICMP echo request의 출발지 IP를 공격 대상 시스템인 시스템 B의 IP로 수정한 뒤, 브로드 캐스트 방식을 통해 다수의 에이전트에 보낸다.

*각 에이전트는 ICMP echo request 패킷을 받고, 그에 대한 응답인 ICMP echo reply 패킷을 공격자가 아니라 시스템 B에 전송한다.

*그 결과, 시스템 B는 다수의 ICMP echo reply 패킷을 처리해야 하며, 이 패킷들을 처리하기 위해 시스템 자원을 모두 소진한 상태이기 때문에 다른 패킷을 처리하지 못하는 상태가 된다.

*시스템 A는 시스템 B에 패킷을 정상적으로 전송할 수 있게 된다.

-대응 방안

*Snort나 방화벽을 이용하여 임계값 이상의 ICMP echo reply 패킷을 차단할 수 있다.

명령어 복붙

2024년 5월 15일 수요일 오후 12:51

```
root@SVVP:~# cd ./
root@SVVP:~# cd /home/samadal
root@SVVP:/home/samadal# pwd
/home/samadal
root@SVVP:/home/samadal# mkdir
mkdir: 피연산자 빠짐
자세한 정보는 'mkdir --help'를 입력하십시오.
root@SVVP:/home/samadal# mkdir ./a/
root@SVVP:/home/samadal# mkdir ./a/b/
root@SVVP:/home/samadal# mkdir -p ../test/c/d/
root@SVVP:/home/samadal# cd /home/test/c/d/
root@SVVP:/home/test/c/d# pwd
/home/test/c/d
root@SVVP:/home/test/c/d#
```

cd->pwd->ls

리눅스 기본

2024년 9월 5일 목요일 오후 10:33

서버
-> 기업용(서비스 제공)su

클라이언트
-> 일반용(서비스 이용[요청])

Os (os 종류)->윈도우
리눅스 - Debian -> Ubuntu(무료)
리눅스 - Redhat -> Roky(무료)

수단(장비)	키보드	마우스Gui(그래픽모드)
	TUI or CLI(명령문모드) Text user Interface Command Line Interface	Graphical user interface

/: 최상위 디렉터리 및 리눅스 경로 표시

I.P : 인터넷 프로토콜 -> 인터넷 통신수단.
원격 서버가 구축이 되어있어야 한다-> ssh 원격서버.

포트포워딩? 접속을 하기 위한 포트번호
Tool(도구)=> Xshell

Sudo : super user(root) 권한으로 명령어를 실행(do)해라!

관리자 권한 위임 변환
1)먼저 $sudo passwd root: 관리자 비번 설정
$su - 명령: 관리자 권한 위임
암호 1
관리자 접속됨

명령 프롬프트(prompt) : 시스템 안에 어떤 계정 권한으로 "명령어를 사용하기 위한 표시 기호"
$-> 사용자 프롬프트

#-> 관리자(root) 프롬프트

디렉토리 구조

$ sudo ls -1 /

L(경로:/) 디렉터리		W(경로:₩) 폴더	

/: 최상위 디렉터리 C:₩

/home/ : 사용자들의 홈 디렉터리의 기본경로 C:₩Users₩

/root/ : 관리자의 홈 디렉터리(관리자 집)

/etc/ : 시스템과 관련된 것이 들어있는 디렉터리 C:₩Window₩System32₩*

(OS, 설치한 프로그램들 생성.)

/bin -> /usr/bin/ : 기본 명령어가 들어있는 디렉터리(명령어 실행파일들)

/sbin -> /usr/sbin/ : 시스템 명령어가 들어있는 디렉터리(관리자만 사용가능)

기본 명령어 1

-pwd : 현재 위치(경로)를 확인한다.

-cd(Change Directory): 폴더의 위치를 변경할 때 사용한다.

: 문법 cd <경로대상> cd는 sudo 와 상관 x

절대 경로 /(최상위)부터 시작한다

.=>현재 경로

.이라는 경로: 현재 경로.

cd ./../ : 현재 위치에서 상위 디렉터리로 이동. ..이라는 경로->상위경로

cd ./samadal guswo 위치에서 samadal/로 이동.

2) 상대 경로 : 현재위치에서부터 시작한다

-ls : 디렉터리 안에 있는 내용물 나열을 확인

(단, 파일일 경우에는 파일 자체만 확인하면 된다)

: 문법 ls[-옵션] <대상1> <대상2>

-ls -a : -a는 . 이란 이름으로 시작하는 데이터들(숨겨진 데이터)도 포함해서 전체를 나열한다.

-ls -l : l은 속성 정보를 자세히 확인하는 것.

D 형태

rwxr-xr-x 권한 허가권
3 링크
root root 권한(소유권)
4096 메모리 크기
9월 8 20:35 날짜

[맨 왼쪽 글자 --> d: 디렉터리, -: 파일]

#ls -R /home/test/ : -R은 경로 안 나열시, 하위에 있는 세부디렉터디들도 자동 나열

-d는 디렉터리 자체가 나타난다.(경로를 나열하지 않게 하기위한 옵션)
-ld 로 조합해야 의미가 있다(디렉터리 속성자체 확인용)

리눅스 CP(복사)

2024년 9월 10일 화요일 오후 7:13

Cp 명령어 작업할때는 실제 원본들은 안 건들인다.
Cp : 파일 또는 디렉터리 데이터를 복사해서 붙여넣기(기본 파일만)

1) 하나씩 복.붙 : 문법

->원본 이름을 생략해서 복사 붙여넣기하기.

Cp [-옵션] <원본 > <목적지>
cp /backup/grub.cfg ./

-> 원본 이름을 변경해서 복사 붙여넣기
cp /backup/debconf.conf ./debconf

2) 여러개 복사 붙여넣기하기.
->cp [-옵션] <원본1> <원본2> … <목적지>

여러 개를 한꺼번에 복사붙여넣기 할때는 이름을 변경할 수 없다.

*: 모든 글자들을 대체한다(와일드카드)
? : 글자 '하나만' 반드시 대체한다.

*cfg는 cfg로 끝나는 모든 것
Login*은 login으로 시작하는 모든 것
conf conf란 단어가 포함된, 섞인 모든 것들

Cp -r/backup/ .../test/a/ : -r은 디렉터리와 그 안의 내용물까지 포함한다.

----디렉터리를 복사 붙여넣기 하는법------
cp -r /backup/ ../test/a/ : -r은 디렉터리와 그 안의 내용물 까지 포함한다.
Test란 디렉터리안에 a 라는 디렉터리가 없으면 a 라는 이름을 유의할것.

Ls -l ../test/a/ ../test/a/backup/

리눅스 mv(move)

2024년 9월 11일 수요일 오후 9:40

-mv(move): 파일 또는 디렉터리 이동 [사용법, 문법, 특징, 형식 등 'cp'와 동일]
[cp가 하는 모든 기능 mv가 가능은 하다]
:문법 -> mv <잘라내기 : 원본1>...<붙여넣기:목적기>

스페이스 바 하나로 대상이 바뀌기 때문에 잘 확인하기.

<디렉터리 옮기기>
mv ./c/d* ./c/g* ./c/l* ./c/backup
ls -l ./a/ ./a/backup/
mv ./a/backup/ /
ls -l ./a/ /backup/

<이름 바꾸기>
ls -l /backup/
mv/backup/grub.cfg /backup/grub.conf : grub.cfg라는 파일을 grub.conf라는 파일로
복붙을 한다는 개념.

---
mkdir -p /home/samadal/a/b/e/f/g/
ls -lR /home/samadal/a/b/e
-p (parents) : 필요할 경우 부모 디렉터리도 같이 생성한다.

Linux_rm(삭제)

2024년 9월 11일 수요일 오후 9:46

rm -rf : 데이터를 삭제하는 명령어

: 문법

rm [옵션] <삭제할 대상들>...

-r : [cp의 -r과 동일하다]

-f : 메시지가 나오지 X

ex) cd/home/samadal/

rm -rf grub.cfg

ls -lR ./a/

rm -rf ./a/

rm -rf/home/test/ [주의 : 1) 한번 강제 시 복구 불가능으로 신중히

2) 삭제 했는지 반드시 확인!]

ls -l /home/

rm -rf/backup/* : 백업 안의 내용물 대상

rm -rf/backup/ : 백업 디렉터리 삭제

리눅스_기본 명령어_

2024년 9월 11일 수요일 오후 9:58

touch : 없으면 빈 문서 파일 생성
: 문법
touch<생성한 파일명>

touch debconf.conf : 같은 이름 중복시 시간만 갱신(원본엔 영향 X)
---
cat(파일 안에 적힌 내용을 모두 출력)
:문법
cat <출력한 파일 대상>
cat/etc/passwd

head/tail (파일의 내용 일부를 출력) : 문법
->head/tail -n(number:숫자) <출력할 대상(파일) 이름>
head/etc/passwd :위 -> 아래로 기본값:10줄 출력
tail/etc/passwd : 아래 -> 위로 기본값:10줄

head -2/etc/passwd : 2줄만큼 출력
tail -15 /etc/passwd : 15줄만큼 출력

-nl : 줄 번호 붙여서 출력
#nl/ et c/ passwd
-grep: 내용 출력 중 찾을 문자열(줄)만 포함해서 출력
grep root/etc/passwd

|(파이프라인) : 명령어를 동시에 사용(조합)

A | B : B라는 명령어를 보조기능으로 사용해서, A라는 출석 명령어 합쳐져서 결과 실행. ex)
| 오른쪽 : head/tail/nl/grep 등

예시)
ls -l/etc/|nl (줄번호 매기기)
ls -l/etc/|head -3
ls -l/etc/|tail -5|nl

응용) ifconfig | head -2
가장 많이 쓰게 될 | 2가지 형태

cat /etc/passwd|grep samadal|nl :grep은 찾을 문자열 포함
:nl은 줄번호 (갯수파악용)

추가 연산자: >,>>

---

1) >은 왼쪽 명령문 결과 출력문을 오른쪽에 쓴 경로에 이름으로 강제
덮어씌워서 문서 생성

cat>file

---

cat > file 1 : 입력 후 문서 생성됨
[ctrl+d을 눌러서 저장,오작동시 터미널 다시키기]

>>은 왼쪽 명령문 결과출력물을 오른쪽에 쓴 경로에 이름으로 문서 있으면 내용
추가 되어서 저장된다.
<응용>
ls -l/etc/|head -5|nl >>file2
cat file1 file2 >>file3

-find(찾기 : 검색 기능)
[주어진 경로 조건부터 검색하여 이름을 찾는다]
:문법
find <경로(경로부터)> -name<"찾을이름">
예
1)samadal이란 이름 검색
find /-name"samadal" :/부터 "samadal"이라는 이름 검색

2)samadal로 끝나는 모든 것들을 검색(개수)
find /-name"*samadal"|nl

find/-name "*samadal"-type d :/부터 samadal로 끝나는 디렉터리들을 검색
-type d:dir
-type f:file
-type[bcdpflsD]=>b,c : 블록(디스크파일),캐릭터, l(링크파일),d(디렉터리),f(파일)

-alias(별칭)
-긴 내용을 짧게 바꾸어서 별도로 칭할 때 사용
-alias 환경변수명='명령문'(단, 일시적)
-존재하는 이름으로는 절대 안하기
-명령문에 숫자가 들어간 것 잘 없기 때문에 숫자 잘 활용하기
ex)alias c='clear'

시스템 종료
1)init 0(숫자)
2)poweroff

시스템 재부팅
1)init 6
2)reboot
ll="ls -1F ./"

Linux_vi

2024년 9월 13일 금요일 오후 1:48

Vi Editor : vi 에디터 프로그램 사용법

sudo vi: 빈 문서 실행(즉, 문서 파일 작업)
sudo vi <편집할 파일명>: 지정한 파일 열고 편집
ex: sudo vi /home/samadal/login.defs

*명령 모드 / 실행(EX) 모드/ 입력 모드:
-> 명령모드 : ESC 키 커맨드 누르는 기본 모습
Vi의 기본모드
Vi가 처음 실행되거나 입력 모드에서 Esc키 누른경우.
커서이동, 문자열 수정, Cp등
명령에 따른 버튼 누르면 바로바로 실행된다.
h,j,k,l : 마우스커서
0(행의 처음으로이동)
$(행의 마지막으로 이동)
G(문서의 마지막으로 이동)
gg(문서의 처음으로 이동)
x : 커서가 있는 문자를 삭제
X: 커서가 있는 앞의 문자를 삭제
dd: 현재 커서의 행을 삭제
숫자 + dd : 현재 커서부터 숫자만큼 행 삭제
yy : 현재 커서가 있는 라인을 복사
숫자 + yy : 현재 커서부터 숫자만큼의 행을 복사
p: 복사한 내용을 현재 라인 이후에 붙여넣기
P: 복사한 내용을 현재 라인 이전에 붙여넣기
u : 되돌리기 Ctrl+r: 앞으로 r: 한 글자 치환

-> 입력모드 : 들어가는법->a,i,o
타이핑을 입력하는 모습
버퍼에 내용을 입력할 수 있는 모드
명령 상태에서 a,i,o,O 등의 키를 누르면 진입
왼쪽 하단에 -INSERT- 라고 표시됨
i : 현재 위치에서 입력 모드로 변경
a: 현재 위치에서 우측으로 한칸 이동 후
입력 모드로 변경

o : 커서 아래에 새로운 행을 추가하고 입력모드 변경

-> EX(실행) : 들어가는법 -> ,/,?
커맨드 입력시 실행되는 모습

/Pattern : Pattern 을 검색
패턴이 검색 된후 n 키를 통해 아래 방향으로 계속 찾기
패턴이 검색 된 후 N키를 통해 위 방향으로 계속 찾기

?Pattern : Pattern 을 검색
패턴이 검색 된후 N 키를 통해 아래 방향으로 계속 찾기
패턴이 검색 된 후 n 키를 통해 위 방향으로 계속 찾기

**치환(중요)
:[범위]s/[Old][New]/옵션 => Old를 New 로 치환

*범위는 n 혹은 n,n 혹은 %를 넣을 수 있다
*g 옵션을 주면 적용되는 라인의 모든 부분을 변경
*g 옵션을 주지 않으면 처음 찾은 부분만 변경.

치환법
1) 범위 생략
:s/[바꿀대상]/[바꾸고싶은이름]/g
g 없을 시: 현재 커서에 있는 줄에서 맨 처음 찾은 것만 치환
g 있을 시 : 현재 커서에 있는 줄에 있는 모든 것 치환
2)n
:10s/sam/madal/g
10번째 줄에 있는 "sam"을 "madal"로 치환
3)n,m
:3, 10s/sam/madal/g
3번째 줄 ~ 10번째까지 "sam"->"madal"로 치환
4)%(문서전체)
:%s/sam/mada
문서 전체에 "sam" -> " madal"로 전부 치환

**Shell 명령어
:![command] vi를 잠시 중단하고 명령어 수행

파일관련 명령어

:e [filename] 파일열기
$:enew 현재 창을 닫고 빈 문서 열기
:q 종료
:q! 강제 종료
:w 파일 저장
:wq 파일 저장 후 종료한다

파일 및 실행 관련
:f => 현재 작업 중인 파일의 이름과 라인 수
:[n]r[filename] : filename 파일의 내용을 현재 편집 중인 파일의 n라인부터 삽입
:[n]r![command] : Command 실행결과를 파일의 n라인부터 삽입

Liunx_user

2024년 9월 19일 목요일 오후 5:25

사용자 관련 파일
/etc/passwd : (사용자 계정 정보가 담긴 파일)
/etc/shadow : (사용자계정정보번호와 관련된 파일)
/etc/group : (그룹계정정보가 담긴 파일: 계정접속과 관계 X)

sha-256/M05

Ex)
#cat /etc/passwd | grep root
root:x:0:0:root:/root:/bin/ksh93

<사용자 정보 유형 (구조 :/etc/passwd)>
samadal : 사용자계정(UID)
x : 비밀변호표시 (비밀번호 설정확인 -> /etc/shadow에서 가능)
*:비번 설정이 불가능한 계정
! : 비번 설정 안된 계정
특수문자들 : 비번설정 암호화된것

1000 : 1000 : UID(User ID), GID(Group ID)
1) 숫자 : 컴퓨터가 보는 ID 식별값
2) 문자 : 사용자가 보는 관점(즉, 계정ID)

samadal,,, : Comment(닉네임, 부연설명)
/home/samadal : 사용자 계정 홈 디렉터리 정보(매우중요!)
/bin/bash : 쉘, 리눅스 명령어 해석기

#cat /etc/shells : 현재 설치된 쉘 종류 확인

==========

1) 사용자 생성
-문법
useradd -m [옵션1] [알맞은 값1] [옵션2] [값]...<계정명UID>
<계정작업의 순서>
useradd user1 : 문제가 있다.
(로그인해도 본인 홈디렉터리로 접근 할 수 없다. 사용자 계정 홈 디렉터리가 생성되지 못한다.)
1_1 : 계정확인 #tail -4/etc/passwd
1_2 : 사용자 홈디렉터리정보와 실제 홈디렉터리 일치 확인 :#ls -l /home/
1_3 : 비밀번호 부여 #passwd user1

==========
useradd -m user2 : -m이 있어야 자동 홈디렉터리로 지정되어 생성됨(기본 : /home/)
#tail -5 /etc/passwd
#ls -l/home/ : 사용자 계정 홈 디렉터리가 자동생성
#passwd user2
==========
useradd -c ubuntuuser -s bin/bash -m user3

tail -6 /etc/passwd
ls -l/home/ : 사용자 계정 홈 디렉터리가 자동 생성
passwd user3

-c : 닉네임, -s : 쉘 변경
==========
3) 사용자 계정 홈 디렉터리 변경해서 생성
#mkdir /cloud/ => 없는 경로는 반드시 생성해주도록 하자.
#useradd -m -d /cloud/user4 user4 =>user4 계정 생생 시 /cloud/ 하위에 user4란 이름으로 홈디렉터리 생성
#tail -7 /etc/passwd
#ls -l /home/ /cloud/
-d : 사용자의 기본 홈 디렉터리의 정보를 변경 [홈 디렉터리이름과 계정 이름은 항상 일치시키자!
==========
<유의사항>
useradd -m -d /cloud user44 -> 홈디렉터리가 생성되지 X 계정은 생성된다
==========
4) skel[스켈]=>/etc/skel/ 안의 파일과 디렉터리를 생성하면
사용자를 생성 시, 사용자의 홈 디렉터리에 자동으로 스켈 안의 내용물이 복사.
따라서 스켈안에 폴더나 문서를 만들고 계정만들면 그대로복사

useradd -m testuser -c usertest -s /bin/ksh93 -d /export/home/

==========
2. 사용자 수정

-문법
#usermod [-옵션 1] [값 1] [-옵션2] [값2] ... <계정명UID>
tail -[생성한 개수] /etc/passwd
#usermod -g user1 -s /bin/bash -c testuser user2
#cat /etc/passwd | grep user2

-g : 그룹변경
-s : 쉘 변경
-c : 닉네임 변경
==========
<usermod의 사용자 홈디렉터리 변경법>

mkdir -p /export/home/ ->없는 경로 반드시 생성!

1) 변경 전 확인

#cat/etc/passwd | grep user2

2) 홈디렉터리 정보 변경

#usermod -d /export/home/user2 user2

/etc/passwd 확인!

ls -l /home/ /export/home/

1) 실제 홈디렉터리 옮기기

#mv /home/user2/ /export/home/ -> 수동 mv로 옮길 것!

#ls -l /export/home/

-md : 한번에 깔끔하게 옮기는데 여기서 m은 경로지정의 m이다.

주의점 : md를 같이 쓸때에는 전제가 계정의 정보와 홈 디렉터리가 정확히 일치할 때 가능하다.

즉 정상인 계정에만 가능하다. 아닌 계정에 사용 시 에러가 생김. md로만 사용

---

usermod 주의사항!

< 사용자 계정 접속해서 사용중엔 홈디렉터리 변경이 되지 않는다.>

#usermod -d /export/home/user3 user3

usermod: user user3 is currently used by process 5137

[원인 ? user3 을 로그인해서 사용중엔 프로세스가 나옴 해결? 접속해서 홈디렉터리 들어간 계정을 끊으면 됨.]

<sudo 사용가능한 samadal은 현재 불가능>

why? samadal로 계정 사용중이기 때문에..

==========

3) 사용자 삭제

-문법

userdel -r <UID 계정명>

' 계정과 관련된 것을 모두 삭제>(옵션과 함께 사용한다)

Linux_Group

2024년 9월 22일 일요일 오후 10:18

그룹 생성, 수정, 삭제
/etc/group : 그룹 계정 정보
samadal : x : 1000 : [그룹에 소속된 사용자 UID]

그룹 생성
groupadd g
tail -5 /etc/group
그룹 변경
#groupmod -g 2000g
그룹 삭제
#groupdel g
#groupdel user1
#groupdel user2
#tail 3 /etc/group

작업
1.user1 계정 생성
2.그룹 (g1) 계정 생성

<계정, uid, gid 확인 명령어>
id <사용자 계정 UID>
uid=1001(user1) gid=1001(user1) groups=1001(user1)

그룹 추가 : usermod -G [추가할 그룹] <UID계정명>
#usermod -G g1 user1
#id user1
uid = 1001(user1) gid = 1001(user1) groups=1001(user1), 1002(g1)

<sudo 그룹권한 추가>
#usermod -G sudo user1
#id user1

테스트 : user1 계정 로그인 후, sudo 이용가능 확인!

Linux_허가

2024년 9월 23일 월요일 오후 9:43

권한 : U/G/O : 사용자/그룹자/ 다른 사용자
d: 디렉터리
-: 파일

r:4 w:2 x:1 2진수 000~777
허가권은 8진수 0~8까지
유저 U/G

read(읽기) : 파일 -> 안의 내용물이 보이냐 제일 중요한 건 파일에서 무조건 r의 권한
디렉터리 -> 안의 내용물이 보이냐

write(쓰기) : 변화(권한이 없으면 변화를 일으킬 수 없다)
ex) vi편집권한, mkdir, touch,>,>>,vi ~w저장하기

x(처리) 파일 -> x가 없으면 문서파일. x가 있는 사용자만이 실행가능하다.(x가 하나라도 있으면.)
디렉터리 -> 디렉터리 안으로 들어갈 수(접근이) 있는 권한. 반드시 있어야 한다. dir은 무용지물 x권한 이 없다면.

허가권 변경(Change Modification)

-권한 : permisstion, 허가:permit, 거부 : deny/ denied(차단되어있다)

문법 - # chmod[변경할 권한 값] <파일또는 디렉터리>
1] 숫자 -> chmod 623 login.defs
2] 문자 -> chmod u+r,u+w,u-x,g-r,g+w,g-x,o-r,o+w,o+x login.defs

내용물이 없을 경우 : # cp -p /backup/* /export/home/
-p : 속성값 그대로 유지해서 복사 붙여넣기

소유권(Change OWNership)
문법 : chown<uid:gid> <파일 혹은 디렉터리들>

pwd

/export/home
chown samadal grub.cfg :uid만 변경
chown :samadal debconf.conf :gid만 변경
chown root:samadal grub.cfg :uid,gid 변경
chown samadal: login.defs :uid와 gid 동일하게 변경

권한 변경 명령문은 root(관리자)로만 할것.

Linux_압축

2024년 9월 28일 토요일 오후 5:42

압축(Compress) : 여러 개의 데이터들을 하나로 묶어서 용량을 줄인다
W : 알집, 반디집, 윈도우 마법사, 7zip...
L: zip,gzip,bzip2, xz

1)gzip
-기본 압축 명령
-압축 속도율 높음

2)bzip2 (bzip2라는 패키지가 설치되어있어야 이용가능)
-gzip,bzip2 보다 압축률이 더 높다
-압축 속도율이 많이 느리다(대용량 압축)

3)xz
-gzip,bzip2보다 압축률이 더 높다
-압축 속도율이 많이 느리다(대용량 압축)

4)zip(각 OS의 통합 압축 기능)
-압축 풀기: unzip[*.zip] (현재 위치에 풀린다)

명령어
1)gzip
압축 : gzip [파일 이름]=> *.gz
해체 : gunzip[압축된 파일이름.gz]

2)bzip2
압축 : bzip2 [파일 이름] =>*.bz2
해체 : bunzip2 [압축된 파일이름.bz2]

3)xz
압축 : xz [파일 이름] => *.xz
해체 : unxz [압축된 파일 이름.xz]

예제)
cd /export/home/
cp -p /backup/* ./

gzip grub.cfg
ls -l
gunzip grub.cfg.gz
ls -l

[압축 대상은 '1개만' 가능 , 대상 형식은 '파일만' 가능
결론 : 묶는 것은 불가능 하다. tar을 이용

-tar(Tape Archive) -> 저장 보관소
: 시스템 안에 있는 데이터들을 묶음 형태로 저장한다.
즉, 데이터들을 묶어서 보관(파일형식)
형태 : *.tar => *.tar.gz *.tar.bz2 *.tar.xz
tar를 하려면 아카이브 생성을 해야한다.
Archive(저장소)
tar 문법
1) 아카이브 파일 생성
tar <options> <(Archiving File Name).tar> <Source File(s)>
cvf
2) 아카이브 파일 풀기
tar <options> <(Archiving File Name).tar> : 현재 위치에 풀림
xvf
option들(-기호를 쓰지 않는다)'

c(create):데이터들을 묶어서 생성(압축 X)
x(extract): 아카이브파일을 풀어줌
v(visual): 아카이브파일을 묶거나 풀 때, 과정을 상세히 보여줌
(verbose : 장황한 상세한)

f(file) : 아카이브 파일의 이름을 지정(정의)

z : tar+gzip 및 gunzip -> *.tar.gz 옵션(cvfz, xvfz)
j : tar + bzip2 및 bunzip2 -> *.tar.bz2 옵션(cvfj, xvfj)
J : tar + xz 및 unxz -> *.tar.xz 옵션 (cvfJ,xvfJ)

3개 묶기(dgl.tar)
#tar cvf dgl.tar grub.cfg debconf.conf login.defs

1)#gzip dgl.tar : 압축

2)#rm -rf grub.cfg debconf.conf login.defs 후
   압축 해제 후, 아카이빙 파일 풀기
gunzip dgl.tar.gz
tar xvf dgl.tar

경로 다른 대상에서 tar + 압축 및 풀기
cd /backup/

tar cvfz /export/home/dglg.tar.gz *
ls -l /backup/ /export/home/

dglg.tar.gz 은 백업에 풀기
사전작업 (grub, debconf.conf,login.defs들만 제거후)

linux_package

2024년 9월 29일 일요일 오전 12:35

<우분투의 패키지 프로그램 관리>

-리눅스 시스템에 프로그램 도구 설치,삭제 관리하는 패키지 관리 프로그램 명령어

-패키지 프로그램 관리 명령어
; dpkg(Debian PacKaGe) : *.deb 도구를 다운 받아서 수동적으로 작업
; apt(Adavanced Packaging tool) : 자동 다운 받아서 자동적으로 작업
(apt는 "인터넷을 통해서 패키지 관리함으로 " 반드시, 인터넷 되야한다!")

---

0. 리눅스 (커널) 업데이트 후, 업그레이드 (윈도우 업데이트 생각)
#apt update : 업데이트 확인 중
#apt upgrade : 업데이트 작업

---

1. 설치된 패키지 정보확인
-dpkg -l 을 활용한 확인

#dpkg -l : 현재 설치된 패키지 종류 전체 나열
방향키 및 스페이스바로 움직임, 출력 중단은 q

- ii : 설치 완료된 상태 확인
rc : 제거된 상태

#dpkg -l <패키지이름> : 설치된 패키지 정보확인
ex) #dpkg -l net -tools
#dpkg -l ssh

#dpkg -l | grep <패키지이름> : 설치된 패키지 부분 나열
ex) #dpkg -l | grep net-tools

2.패키지 삭제
#apt remove 패키지도구명 : 설치된 패키지를 삭제
ex) apt remove net-tools
#apt autoremove 패키지도구명 : 사용하지 않는 패키지도 포함해서 패키지 삭제
ex) apt autoremove ssh

---

3. 패키지 설치

#apt install <패키지이름> --> 설치하시겠습니까? y/n

또는

#apt -y install <패키지이름> --> 자동설치

ex) #apt install net - tools

#dpkg -l | grep net-tools

#ifconfig | head -2

#apt install ssh

#dpkg -l | grep ssh

원격 확인(실제 원격 서비스 제공 패키지명: openssh-server)

HDD

2024년 10월 1일 화요일 오후 7:45

1. Hard Disk 만들기 -> SCSI ->용량

2. HDD 이름 확인
   fdisk -l <hdd 장치명>
3. fdisk /dev/sdb => 파티션 작업
   파티션 모드
   d: 한 개의 파티션 삭제
   n: 파티션 추가
   p: 파티션 테이블 확인
   q: 저장 없이 종료
   w: 저장 후 종료

   4.포맷 : 새로운 파티션 생성
   (make file system(mkfs))

   sudo mkfs.ext4 /dev/sdb1
   sudo mkfs -t ext4 /dev/sdb1

   5.마운트(mount): 포맷된 장치와 디렉터리를 연동
   마운트 연동 : mount <포맷장치명> <M.P>
   ex) $ sudo mount /dev/sdb1 /mp/
   마운트 풀기:
   unmount < 포맷 장치명 > 또는 <M.P> : 둘 중 하나
   sudo umount /dev/sdb 1
   sudo umount /mp/ 중 선택

   M.P = 마우스 포인터
   M.P => 사용되지 않는 디렉터리 사용 해서 HDD 저장공간 장치와의 연동하는 디렉터리
   sudo df -h: 마운트 연동 확인

   <Mount 유의 사항들>
1. M.P는 임의의 디렉터리여야 한다. (사용되지 않는다.)
2. mount와 umount 명령 시 현재 경로가 마운트 된 디렉터리 (M.P) 여서는 안된다. (사

용 중엔 X)

3. 한 개의 파티션은 반드시 한 개의 m.p만 사용한다 (1대1)
4. 포맷 및 마운트 작업 대상은 P,L만 대상이 된다. 확장(E)는 될 수 없다.
5. 파티션 작업 시반드시 마운트 연동은 풀고 해야한다.