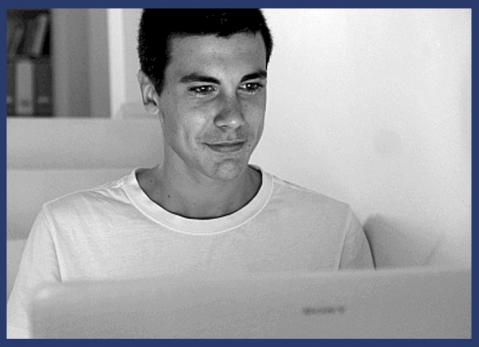
Hacker Highschool

SECURITY AWARENESS FOR TEENS



LESSON 2
ESSENTIAL COMMANDS

HACKING IS LEARNING
www.hackerhighschool.org

ISECOM

Warning – WORLD

Hacker Highschool Project(HHS)는 학습 도구로 만들어졌지만, 일부 내용의 경우 오용될 시 물리적 피해가 발생할 수 있습니다. 또한 특정 기술들로 인한 파급력과 영향력을 충분히 숙지하고 있지 못할 시 추가적인 문제가 발생할 수 있습니다. 학생들이 배운 내용들을 활용하는데 있어 주의가 필요하지만 적극적으로 배우고 연습할 수 있도록 지도해 주시기 바랍니다. 하지만 배운 내용들을 악의적으로 사용해 발생한 문제들에 대해 ISECOM은 책임을 지지 않습니다.

해당 교제에서 배울 내용들과 연습문제들은 공개되어 있으며 ISECOM에서 제시하는 다음의 조건들을 따를 시 누구나 사용할 수 있습니다:

HHS의 자료들은 초등학생, 중학생, 고등학생들에게 무료로 제공됩니다. 해당 자료들은 판매를 목적으로 생산될 수 없습니다. 대학, 직업학교, 단기과정, 방학특강 등에서 실시하는 어떤 학습 과정에서도 허가 없이 학생들에게 해당 자료들로 인한 비용을 부과해서는 안 됩니다. 허가증을 받기 위해서는 HHS 홈페이지의 http://www.hackerhighschool.org/licensing.html를 방문하여 라이센스 섹션을 참조하시기 바랍니다.

HHS는 공개 참여를 통한 오랜 노력과 여러 사람들의 지원으로 만들어졌습니다. 해당 프로젝트에 도움을 주고 싶으시다면 허가증 구매, 기부, 후원을 통해 HHS에 일조 하실 수 있습니다.

Warning – KOREA

HHS 는 ISECOM의 프로젝트입니다.
HHS 외 ISECOM의 모든 프로젝트의 한국 내에서의 관리 및 감독 등의 모든 권한은 ISECOM Korea 에 있습니다.
본 문서의 용도는 개인 학습용이며 무료 공개자료입니다.
해당 용도 외 사용 시 법적인 처벌을 받으실 수 있습니다.
해당 용도 외 문의는 ISECOM Korea 에 문의해 주시기 바랍니다.

목차

도움을 주신 분들

Pete Herzog, ISECOM
Marta Barceló, ISECOM
Bob Monroe, ISECOM
Marco Ivaldi, ISECOM
Greg Playle, ISECOM
Simone Onofri, ISECOM
Kim Truett, ISECOM
Jaume Abella, ISECOM
Tom Thomas, ISECOM
Jairo Hernández
Aneesh Dogra

ISECOM Korea

왕응석 EungSeok Wang
정진우 JinWoo Jung
박영후 YongHoo Park
최홍선 HongSeon Choe

Introduction and Objective

영화 스워드피쉬(Swordfish)의 배우 휴 잭맨을 상상하거나 영화 매트릭스(Matrix Reloaded)에서 나오는 해킹 장면을 기억한다면 해커들이 여러 가지 다양한 명령어들을 컴퓨터에 입력하는 모습이 해킹과 관련 있다고 여길 것이다.

여러분은 명령어 인터페이스(**CLI:command line interface**)를 이용해서 굉장히 많은 일들을 할 수 있다. 모든 명령어를 다 알아둘 있을 필요는 없지만 명령어들에 익숙해질 필요가 있다. 일단 기본적인 명령어들을 익힌다면 텍스트파일(script)에서도 해당 명령어들을 사용할 수 있다. 이게 가장 기본적인 프로그래밍이다.

만약 여러분이 명령어 인터페이스(CLI)의 기초를 터득하였다면, 이제 텍스트 파일(스크립트:script 라고 부름)에서 이러한 명령어들을 사용할 수 있다. 이것들은 가장 간단한 프로그래밍일 것이다.

이제부터 우리는 각종 명령어들과 윈도우, 리눅스, 그리고 OSX에서 사용되는 도구들에 대해 살펴볼 것이다. 각 레슨에서 나오는 연습문제들을 통해 충분히 공부해야 한다. 이번 장을 끝마칠 때 즈음에는 아래와 같은 기본적인 명령어들에 익숙해 질 것이다.

- 윈도우, OXS, 리눅스의 기본 명령어들
- 기본적인 네트워크 명령어들과 도구들

```
ping
tracert / traceroute
netstat
ipconfig / ifconfig
route
```

준비사항과 환경설정

요구 사항

여러분이 준비해야 할 것들은 다음과 같다.

- 윈도우가 설치되어 있는 컴퓨터
- 리눅스가 설치되어 있는 컴퓨터
- OSX가 설치되어 있는 맥(자유선택)
- 인터넷 접속

환경 설정

Figure 2.1: 기본적인 네트워크 설정

위의 그림이 바로 우리들이 앞으로 실습할 네트워크 환경이다. 컴퓨터, 인터넷, ISECOM Hacker Highschool 테스트 네트워크로 구성되어 있다.

IESECOM의 테스트 네트워크에 접속하는 것은 제한되어 있다. 접속하기 위해서는 시스템 관리자와 연락을 취해야 한다. 자세한 사항은 http://www.hackerhighschool.org에서 알 수 있다.

하지만 ISECOM이 아닌 다른 테스트 네트워크를 사용해도 된다. **주의할 점**은 다른 컴퓨터에 무단으로 접속할 수 있는 테스트 네트워크를 사용하는 것은 범죄에 해당한다는 것이다.

자신만의 테스트 네트워크를 만들고 싶다면 친구들이나 자신의 집에 있는 다른 컴퓨터를 사용하면 된다. 과정은 아주 간단하다. 물론 더욱 활동적인 작업을 하거나 실제적인 문제들과 오류들을 경험하기 위해서는 인터넷 기반의 테스트 네트워크 환경이 필요하다. 이 또한 다른 단체나 사람들과의 협력을 통해서 서로 간의 컴퓨터에 원격접속이 가능한 환경을 만들 수 있다. 하지만 네트워크 환경을 설정할 때 다른 외부의 사람들이 마음대로 침입해서 피해를 줄 수 없도록 방비를 해야 한다. 이러한 피해의 책임은 자신에게 있다는 것을 명심하자.

운영체제: 윈도우

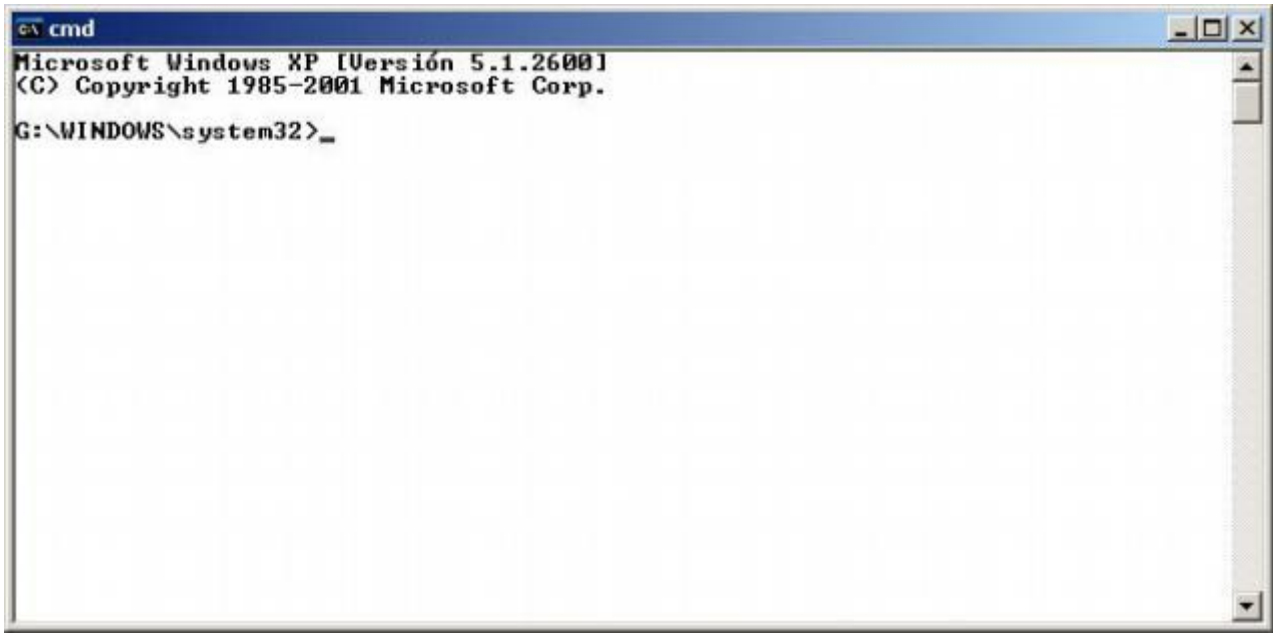
우리가 만약 유닉스를 사용하지 않았다면 DOS를 계속 사용했을 것이다. 그때는 컴퓨터 자체가 도스 체제였기 때문에 CLI를 킬 필요도 없었다. 결국 유닉스는 "윈도우"인터페이스를 개발했고 마이크로소프트 윈도우의 기본적인 환경이 되었다.

이제 윈도우에서 사용하는 도스창을 **명령어 프롬프트(command prompt)**라고 부른다. 도스 기반 컴퓨터 환경에서 벗어 난지 오래 되었지만 많은 사람들이 여전히 명령어 프롬프트를 **도스박스(DOS box)**라고 부르고 있다. DOS라고 부르기에는 많은 변화가 있지만 우리들에겐 별로 중요한 문제가 아니다. 어떻게 명령어 프롬프트를 여는지 알아보자.

CLI 사용법

모든 윈도우 버전에서 동일한 방식으로 구동시킬 수 있다.

1. 시작버튼을 누른다.
2. 실행버튼을 누른다.(윈도우 비스타에서는 이 과정을 생략한다.)
3. **cmd**라는 명령어를 입력하고 확인을 누른다.
4. 다음과 같은 창이 나타날 것이다.



5. 이제 앞으로 배울 명령어와 도구들을 사용할 수 있다.

명령어와 도구들(윈도우/도스)

명령어를 통해 윈도우에 탑재되어 있는 기능들을 사용할 수 있다. 도구들은 더 많은 기능을 가지고 있다. 예를 들면, 네트워크 탐색, 호스트 검색, 라우팅 정보 열람하거나 설정을 할 수 있다.

명령어

이텔릭체로 표기되어 있는 것은 상황에 맞게 원하는 것을 입력하면 된다.

몇 가지 명령어는 축약형 명령어를 가지고 있다.

명령어	기능
date	현재 날짜를 확인 또는 날짜 설정
time	현재 시간을 확인 또는 시간 설정
ver	MS-DOS 또는 윈도우 버전 확인
dir	하위 디렉터리와 파일들을 확인
cls	명령 프롬프트 화면을 정리
mkdir *directory* *or* **md *directory***	자신이 적은 디렉터리를 생성 예시: `md tools`
chdir *directory* *or* **cd *directory***	현재 설정되어 있는 디렉터리를 다른 디렉터리로 교체 예시: `cd tools`
rmdir *directory* *or* **rd *directory***	해당 디렉터리를 삭제 예시: `rd tools`
tree *directory*	해당 디렉터리의 폴더와 파일들을 확인 예시: `tree c:\tools`
chkdsk	디스크 검사를 실행하고 결과를 확인
mem	사용 중인 메모리와 여유 메모리를 확인
rename *source dest* *or* **ren *source dest***	대상의 이름을 변경 예시: `ren pictures MyPics`
copy *source dest*	하나 이상의 대상을 해당 위치로 복사 예시: `copy c:\tools\myfile.txt c:\tmp\`
move *source dest*	하나 이상의 대상을 해당 위치로 이동 예시: `move c:\tools c:\tmp`
type *file*	하나 이상의 파일 내용을 출력 예시: `type c:\tools\myfile.txt`
more *file*	하나 이상의 파일 내용을 한 스크린씩 출력 예시: `more c:\tools\myfile.txt`
delete *file* *or* **del *file***	하나 이상의 파일을 삭제 예시: `del c:\tools\myfile.txt`

도구

이텔릭체로 표기되어 있는 상황에 맞게 원하는 것을 입력하면 된다.

도구	기능
ping *host*	해당 호스트와의 연결 상태를 확인한다. ping 명령어를 통해 ICMP(Internet Control Message Protocol) 핑 패킷을 다른 컴퓨터에 전송해서 해당 호스트가 반응하는지와 반응하는 시간을 측정한다. 호스트명이나 IP주소를 사용할 수 있다. 예시: `ping hackerhighschool.org` `ping 216.92.116.13` 추가기능: 100개의 핑 패킷을 전송한다. 예시: `ping -n 100 hackerhighschool.org` CTRL+C를 입력해 중지할 때까지 계속해서 핑패킷을 전송한다. 예시: `ping -t 216.92.116.13` 더 많은 추가기능을 확인 하려면 `ping /h`를 입력하면 된다.
tracert *host*	패킷이 해당 호스트까지 도달하기 위한 경로를 보여준다. **tracert** 명령어는 유닉스의 **traceroute** 명령어의 수정 버전이다. (과거 DOS에서 사용할 수 있는 명령어의 문자 수는 최대 8자리였다.) 두 명령어를 사용하면 자신의 컴퓨터에서 목적지까지의 경로를 추적할 수 있다. tracert는 얼마나 많은 경로를 거쳐야 하는지 최대 30홉까지 탐색한다. 패킷이 전송되는 것을 통해 해당 컴퓨터의 호스트명을 알 수도 있다. 예시: `tracert hackerhighschool.org` `tracert 216.92.116.13` 추가기능: 대상 검색을 위한 최대 홉수를 25개로 설정한다. 예시: `tracert -h 25 hackerhighschool.org` 호스트명을 탐색하지 않는다. 예시: `tracert -d 216.92.116.13` 더 많은 추가기능을 확인 하려면 `tracert /?`를 입력하면 된다.

도구	기능
ipconfig	단일명령어로 작동하며 현제 활성화되어 있는 네트워크 인터페이스 정보를 보여준다(이더넷, PPP 등). 리눅스의 **ifconfig** 명령어와 유사하다. 추가기능: 보다 상세한 정보를 보여준다. 예시: `ipconfig /all` DHCP에서 자동으로 IP를 할당 받을 경우 네트워크 접속을 새롭게 갱신할 수 있다. 예시: `ipconfig /renew` DHCP가 사용되고 있을 때 네트워크와의 접속을 해제한다. 예시: `ipconfig /release` 더 많은 추가기능을 확인 하려면 `ipconfig /?`를 입력하면 된다.
route *print*	라우팅 테이블 목록을 보여주고 목록에 새로운 경로를 추가하거나 기존의 경로를 삭제할 수 있다. 추가기능: 라우팅 테이블 목록을 보여준다. 예시: `route print` 라우팅 테이블에서 선택한 경로를 삭제한다. 예시: `route delete` 라우팅 테이블에서 새로운 경로를 추가한다. 예시: `route add` 더 많은 추가기능을 확인 하려면 `route /?`를 입력하면 된다.

도구	기능
netstat	네트워크의 상태정보와 접속한 외부 네트워크의 정보를 보여준다. 추가기능: 모든 네트워크 연결과 대기 중인 포트를 확인한다. 예시: `netstat -a` IP주소와 포트번호를 번호 순으로 나열한다. 예시: `netstat -n` 이더넷의 통계치를 보여준다. 예시: `netstat -e` 일부 추가 기능들은 같이 사용할 수 있다. 예시: `netstat -an` 더 많은 추가기능을 확인 하려면 `netstat /?`를 입력하면 된다.

명령어와 도구들에 관한 추가 정보를 알고 싶다면 다음의 명령어를 사용하면 된다.

```
command /h
command /?
help command
```

마찬가지로 netstat 도구에 관한 추가 정보를 알고 싶다면 세 가지 방법이 있다.

```
netstat /h
netstat /?
help netstat
```

Excercise

2.1 CLI를 열어보자.

2.2 DOS와 윈도우 버전을 확인해 보자.

2.3 날짜와 시간을 확인해 보자. 정확하지 않다면 수정해 보자.

2.4 C:\ 안에 있는 모든 디렉터리와 파일들을 확인해 보자.

2.5 C:\hhs\lesson2 디렉터리를 만들고 c:\안에 있는 extension.sys 파일들을 복사해 보자. 어떤 파일들을 찾았는가?

2.6 컴퓨터의 IP 주소를 확인해 보자.

2.7 www.hackerhighschool.org까지의 경로를 확인하고 라우터들의 IP주소를 알아보자.

Game On: Taking Command

과학 선생님은 입가에 음식찌꺼기를 묻혀놓고 말했다.
“마이크로소프트 페네스트라(Macrosoft Fenestra)는 OS시스템과 인터페이스 그 어디에도 포함되지 않지. 그저 솔리테어(Solitaire)에 짜여져 있는 그래픽 시스템이란다.”

트리 선생님은 우리가 이해했다고 여겼는지 다음으로 넘어갔다.
“페네스트라는 명령어 인터페이스를 가지고 있는데 이를 이용해서 모니터에게 말을 하면 원하는 것을 얻을 수 있어. 만약 커피 한잔을 마시고 싶다면 모니터에 말하면 돼. 그러면 맛있는 커피가 짜잔 하고 나오게 된단다.”

제이스는 이 선생님을 혼내주고 싶은 마음이 들었다. 만약 이 선생을 곤경에 처하게 한다 해도 경찰과 판사가 과학 선생이 망쳐버리고 있는 컴퓨터 수업에 대해 알게 된다면 자신에게 공감하지 않을까 생각했다.
“잠깐, 잠깐만요 선생님.”

제이스는 10여분 동안 숨도 제대로 못 쉴 정도로 화가나 있어서 얼굴이 시뻘겠다.
“선생님, 페네스트라는 그래픽 유저 인터페이스로 GUI로 불립니다. 선생님이 씹다가 통에 보관해 놓은 껌 같은 거죠.”

다른 아이들은 조용히 낄낄 거렸다.
제이스는 일어나서 과학 선생님을 향해 재빠르게 나아가더니 마치 디펜스를 뚫는 프로 농구선수처럼 부드럽게 선생님을 제치고 키보드가 있는 곳에 자리를 잡았다.
“윈도우 실행창을 클릭하고 CMD를 입력하면 CLI가 나오죠. 깜박이는 칸이 보이시죠? 여기에 타이핑을 할 수 있고 지금 어떤 폴더에 위치해 있는지 알 수 있죠.”

제이스는 포뮬러1 출전 드라이버처럼 뒤를 돌아보지도 않고 속도를 높여 나갔다.
“이제 CD C:를 타이핑해서 시스템 루트에 들어왔죠.”

제이스는 좀 더 속도를 높였다.
“새로운 시스템을 통해 컴퓨터 환경설정이 어떤지 최대한 많이 알아낼 수 있어요. version의 축양형인 VER을 타이핑해서 OS의 버전을 알 수 있죠. 보이시죠?”

학생들은 집중하고 있었고 트리 선생은 마비된 듯 서있었다.
제이스는 자신이 컴퓨터와 연결되어 있다고 느끼며 점점 더 여유를 가지며 타이핑 속도를 높이기 시작했다.

"컴퓨터가 가진 모든 정보를 토해내게 해서 지금 컴퓨터에서 어떤 일이 벌어지고 있는지 알아 낼 수도 있죠."

제이스는 크게 중얼거렸다. 제이스의 화려한 손놀림에 의해 키보드의 키 하나가 날아가서 트리 선생이 씹던 껌을 보관하는 통에 쏙 들어갔다. 그 앞에 있던 여학생 3명은 놀라서 씹고 있던 껌을 삼켜 버렸다.

제이스는 그걸 마무리 사인으로 여기며 바로 일어나서 키보드를 트리 선생에게 돌려주었다. 선생의 얼굴은 창백했고 입술에 약간의 침이 묻어 있었다. 제이스는 자신의 재킷 안주머니에서 마치 총을 꺼내듯 레이저 포인터를 꺼내 선생의 이마에 레이저를 쏘았다. 이를 본 뒷자리의 학생이 폭발하듯 웃어댔다. 제이스는 뒤로 돌아 프레젠테이션 스크린을 가리키며 조용히 말했다.

"이 슬라이드들은 쓸모가 없으니 빼버리는게 좋겠어요."
"너를 빼는게 나을 것 같은데."
트리 선생은 제이스에게 교감선생님과의 방과 후 상담을 벌로 내렸다. 벌써 이번 주에만 세 번째 상담이다. 제이스의 방과 후 자유 시간은 사라지게 되었다.

Game Over

운영체제: 리눅스

윈도우에서처럼 리눅스도 CLI를 제공하며 **콘솔(consoles)**, **터미널(terminals)**, **쉘(Shells)**로 불린다.

Feed Your Head: Console, Terminal or Shell?

- **콘솔(console)**은 원격으로 컴퓨터에 접속하기 위해 사람들이 **덤 터미널(dumb terminal)**을 사용하던 시절에 컴퓨터에 직접 연결되어 있던 스크린과 키보드를 뜻한다.
- 리눅스에서 **쉘(shell)**을 사용하고 싶다면 **bash**, **tcsh**, **zsh**등의 다양한 선택지가 있다. 어떤 쉘을 사용하는지에 따라 활용할 수 있는 기능들이 다르다. 대부분의 경우 bash를 사용한다. Hacker Highschool의 테스트 네트워크에 접근할 때 **비어있는 쉘(empty shell)**을 볼 수 있을 것이다.
- **콘솔 윈도우(console window)**를 실행하는 것은 기술적인 관점으로 보자면 **터미널 애뮬레이터(terminal emulator)**나 **터미널 윈도우(terminal window)**를 열었다고 볼 수 있다. 즉 덤 터미널의 현대판 모조품인 것이다.

리눅스에서 커맨드 라인을 사용하면 무엇을 할 수 있을까? 어떤 GUI 툴을 이용하던 다양한 것을 할 수 있다. 윈도우를 사용하는 친구들에게 IP 주소를 한번 설정해 보라고 해보자. 이들은 다양한 종류의 인터페이스를 지나서야 접속하지만 리눅스에서는 다음의 명령어로 간단하게 IP 주소를 설정할 수 있다.

```
ifconfig eth0 192.168.1.205
```

클릭보다 이 명령어를 사용하는 것이 분명 더 빠를 것이다.

터미널 사용법

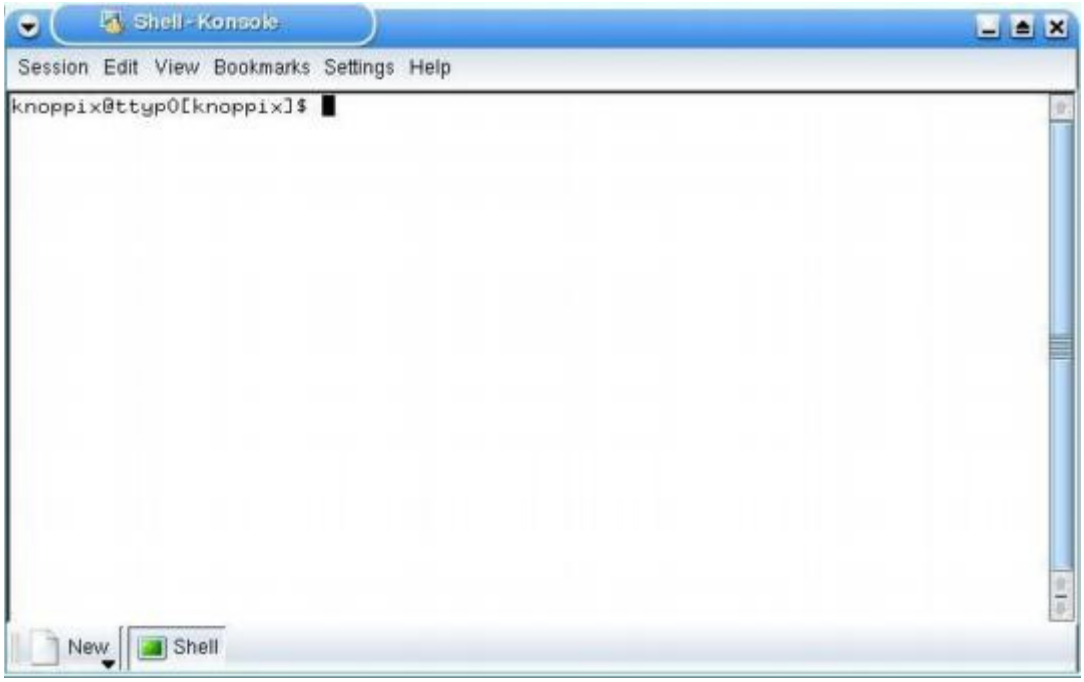
리눅스는 다양한 버전으로 존재하기 때문에 콘솔 윈도우를 시작하기 위한 방법이 다수 존재한다.

1. Start Application 버튼을 누른다.
2. ‘Run Command’를 클릭하고 “konsole"이라고 입력한다.
3. 또는 Accessories를 찾아서 Terminal을 선택하면 된다.
4. 단축키로는 Ctrl, Alt키와 t를 동시에 누르면 된다.
5. 아래의 사진과 유사한 창이 나타날 것이다.
6. 이제 앞으로 배울 명령어와 도구들을 사용할 수 있다.

리눅스 명령어와 도구들

명령어

이텔릭체로 표기되어 있는 상황에 맞게 원하는 것을 입력하면 된다.



명령	도구
date	현재 날짜를 확인 또는 날짜 설정
time	현재 시간을 확인 또는 시간 설정
fsck	파일시스템을 검사하고 결과를 확인
cat *file*	하나 이상의 텍스트 파일의 내용들을 확인 예시: `cat /etc/passwd`
pwd	현재 설정되어 있는 디렉터리를 확인

명령	도구
hostname	현재 사용하고 있는 컴퓨터의 호스트명을 확인
finger *user*	사용자의 정보를 확인 예시: `finger root`
ls	현재 설정되어 있는 디렉터리 내의 파일과 폴더들을 확인 예시: `ls -la` 추가기능: 다른 디렉터리의 파일과 폴더들을 확인 예시: `ls -la /etc`
cd *directory*	현제 설정되어 있는 디렉터리를 다른 디렉터리로 교체(디렉터리명이 정확하지 않을 시 홈 디렉터리로 교체) "fred" 이름으로 로그인을 한 후 다음과 같이 입력한다. 예시: `$ cd` /home/fred 디렉터리로 변경된다. 또한, 예시: `$ cd -` 마지막에 방문한 디렉터리로 변경된다. 예시: `$ cd /tmp` `tmp` 디렉터리로 변경
cp *source dest*	하나 이상의 대상을 해당 위치로 복사 예시: `cp /etc/passwd /tmp/bunnies`
rm *file*	해당 파일을 삭제(접속 권한을 가지거나 루트 접속만 가능) 예시: `rm letter.txt`
mv *source dest*	하나 이상의 대상을 해당 위치로 이동 예시: `mv secrets.zip innocent.zip`
mkdir *directory*	자신이 적은 디렉터리를 생성 예시: `mkdir tools`
rmdir *directory*	해당 디렉터리를 삭제(디렉터리 안이 비어있을 경우만 가능) 예시: `rmdir tools` 보너스 문제 : 해당 디렉터리가 비어있지 않을 경우 어떻게 삭제하는가?
find / -name *file*	해당 파일을 검색 예시: `find / -name myfile`

명령	기능
echo *string*	해당 문자를 스크린에 출력 예시: `echo hello`
command > *file*	해당 명령어의 결과를 파일 형식으로 생성(같은 이름의 파일이 있을 경우 원본을 지워버린다.) 예시: `ls > listing.txt`
command >> *file*	해당 명령어의 결과를 파일 형식으로 생성(같은 이름의 파일이 있을 경우 그 아래에 내용을 추가한다.) 예시: `ls >> listing.txt`
man command	해당 명령어에 관한 온라인 매뉴얼을 생성 예시: `man ls`

명령어와 도구들에 관한 추가 정보를 알고 싶다면 다음의 명령어를 사용하면 된다.

```
command –h
command –help
man command
help command
info command
```

예를 들어 명령어 ls에 관한 정보를 알고 싶다면 다음의 두 가지 선택지가 있다.

```
ls --help
man ls
```

도구

이텔릭체로 표기되어 있는 것은 상황에 맞게 원하는 것을 입력하면 된다.

도구	기능
ping *host*	해당 호스트와의 연결 상태를 확인한다. 예시: `ping www.google.com`
traceroute *host*	패킷이 해당 호스트까지 도달하기 위한 경로를 보여준다. 예시: `traceroute www.google.com`
ifconfig	현재 활성화되어 있는 네트워크 인터페이스 정보를 보여준다(이더넷, PPP 등)
route	라우팅 테이블 목록을 보여준다.
netstat	네트워크의 접속 상태를 보여준다. 예시: `netstat -an`

Excercise

2.8 **passwd** 파일의 주인이 누구인지 알아보자.(우선 파일의 위치를 알아내야 한다.)

2.9 **work** 디렉터리를 홈 디렉터리에 만들고 passwd파일을 work 디렉터리로 복사한다. (예를 들어 **fred** 라는 이름으로 로그인을 했다면, /home/fred 디렉토리가 만들어질 것이다) 복사된 파일의 주인을 알아보자.

2.10 work 디렉터리 안에 **.hide** 디렉터리를 만들자(hide 앞에 .이 붙는 것을 유의하자). .hide 디렉터리의 내용을 확인하려면 어떻게 해야 하는가?

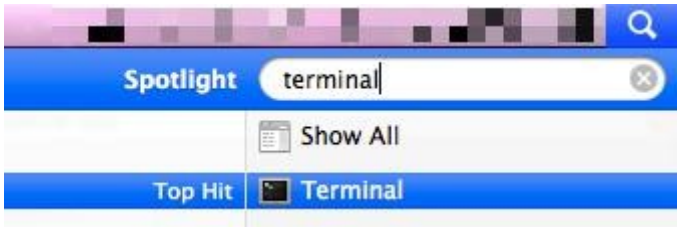
2.11 "This is the content of the file test1"이라는 내용이 들어있는 **test1**파일과 “This is the content of the file test2"의 내용이 들어있는 test2파일을 work 디렉터리 안에 만들자. **test**파일을 만들고 test1과 test2의 내용을 복사해보자.

운영체제: OSX

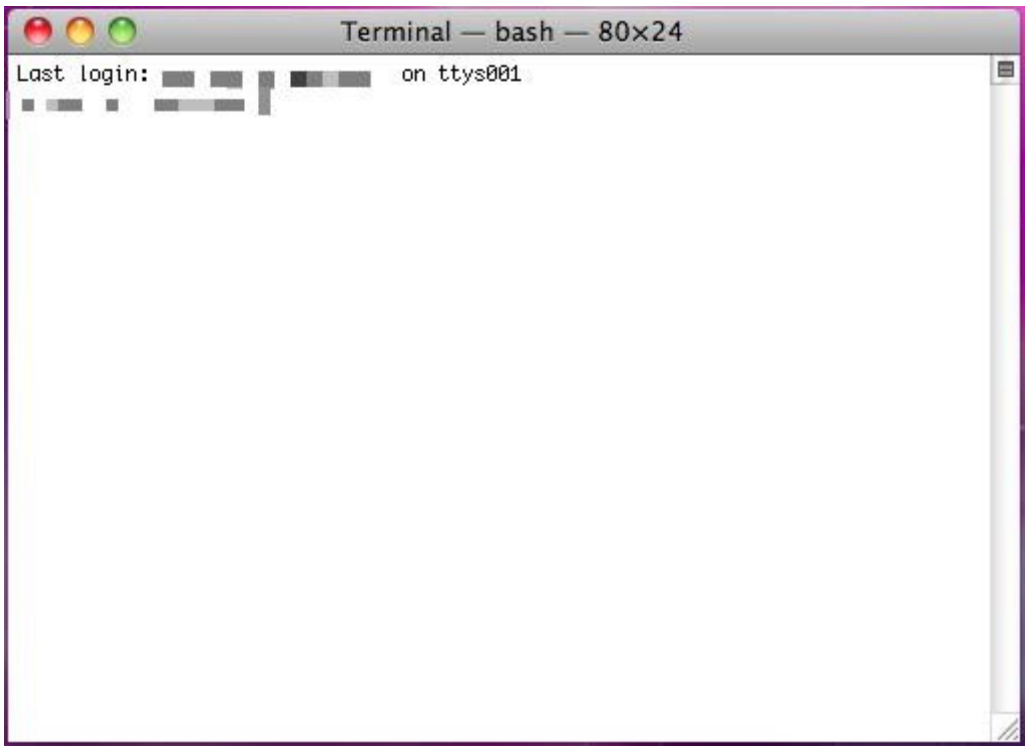
리눅스와 마찬가지로 OSX도 CLI 창을 가지고 있다. OSX에서는 이를 **터미널(Terminal)**이라고 부른다. OSX는 리눅스의 전신인 NetBSD와 FreeBSD UNIX에 기반을 두고 있다. OSX의 GUI와 CLI는 리눅스와 유사하다. 어떤 GUI 툴을 이용하던 다양한 것을 할 수 있다.

터미널 사용법

1. **Spotlight** 아이콘을 클릭하고 **Terminal**을 검색한다.
   돋보기 모양처럼 생긴 Spotlight 아이콘은 주로 화면의 우측 상단에 위치해 있다.



2. 엔터를 누르거나 클릭을 하면 터미널 창을 볼 수 있다.



일반적으로 터미널은 **Applications > Utilities** 아래에 위치한다. 터미널 스타일은 자신이 원하는 데로 조정할 수 있다. 명령어와 콤마 키를 동시에 누르면 터미널의 글자 스타일을 자신이 원하는 데로 조정할 수 있다. 키보드를 이용한 단축키를 사용하면 OSX에서 제공하는 설정 창에 쉽게 갈 수 있다.

명령어와 도구들(OSX)

맥은 bash 쉘을 사용하기 때문에 리눅스에서 사용하는 대부분의 명령어가 동일하게 적용된다.

명령어

이텔릭체로 표기되어 있는 것은 상황에 맞게 원하는 것을 입력하면 된다.

명령	도구
date	현재 날짜를 확인 또는 날짜 설정
time *command*	해당 명령어를 실행하는데 소요되는 시간을 표시
fsck	파일 시스템을 검사하고 결과를 출력한다. 맥OS 10.3 혹은 그 이후 버전들과 같은 저널링이 가능한 (journaled volume) OSX를 쓴다면 기본설정으로 저널링(journaling)을 할 수 있기 때문에 fsck 명령어를 사용할 필요가 없을 것이다.
cat *file*	하나 이상의 텍스트 파일의 내용들을 확인 예시: `cat /etc/passwd`
pwd	현재 설정되어 있는 디렉터리를 확인
hostname	현재 사용하고 있는 컴퓨터의 호스트명을 확인
finger *user*	사용자의 정보를 확인 예시: `finger root`
ls	현재 설정되어 있는 디렉터리 내의 파일과 폴더들을 확인 예시: `ls -la` 다른 디렉터리의 파일과 폴더들을 확인 예시: `ls -la /etc`
cd *directory*	현재 설정되어 있는 디렉터리를 다른 디렉터리로 교체(디렉터리명이 정확하지 않을 시 홈 디렉터리로 교체) “fred” 이름으로 로그인을 한 후 다음과 같이 입력한다. 예시: `cd` /home/fred 디렉터리로 변경된다. 또한, 예시: `cd -` 마지막에 방문한 디렉터리로 변경된다. 예시: `cd /tmp` tmp 디렉터리로 변경

명령	도구
cp *source dest*	하나 이상의 대상을 해당 위치로 복사 예시: `cp /etc/passwd /tmp/bunnies`
rm *file*	해당 파일을 삭제(접속 권한을 가지거나 루트 접속만 가능) 예시: `rm letter.txt`
mv *source dest*	하나 이상의 대상을 해당 위치로 이동 예시: `mv secrets.zip innocent.zip`
mkdir *directory*	자신이 적은 디렉터리를 생성 예시: `mkdir tools`
rmdir *directory*	해당 디렉터리를 삭제(디렉터리 안이 비어있을 경우만 가능) 예시: `rmdir tools` 보너스 문제 : 해당 디렉터리가 비어있지 않을 경우 어떻게 삭제하는가?
find / -name *file*	해당 파일을 검색 예시: `find / –name myfile`
echo *string*	해당 문자를 스크린에 출력 예시: `echo hello`
command > *file*	해당 명령어의 결과를 파일 형식으로 생성(동일한 파일이 있을 경우 원본을 지워버린다.) 예시: `ls > listing.txt`
command >> *file*	해당 명령어의 결과를 파일 형식으로 생성(동일한 파일이 있을 경우 그 아래에 내용을 추가한다.) 예시: `ls >> listing.txt`
man *command*	해당 명령어에 관한 온라인 매뉴얼을 생성 예시: `man ls`

명령어와 도구들에 관한 추가 정보를 알고 싶다면 다음의 명령어를 사용하면 된다.

```
command –h
command --help
man command
help command
info command
```

예를 들어 명령어ls에 관한 정보를 알고 싶다면 다음의 두 가지 선택지가 있다.

```
ls --help
man ls
```

도구

이텔릭체로 표기되어 있는 것은 상황에 맞게 원하는 것을 입력하면 된다.

도구	기능

ping *host*	해당 호스트와의 연결 상태를 확인한다.
	ping 명령어를 통해 ICMP(Internet Control Message Protocol) 핑 패킷을 다른 컴퓨터에 전송해서 해당 호스트가 반응하는지와 반응하는 시간을 측정한다. 호스트명이나 IP주소를 사용할 수 있다.
	
	예제: `ping hackerhighschool.org`
	`ping 216.92.116.13`
	추가기능: `ping -c 100 hackerhighschool.org`
	100개의 핑 패킷을 전송한다.
	`ping -t 216.92.116.13`
	CTRL+C를 입력해 중지할 때까지 계속해서 핑패킷을 전송한다.
	
	더 많은 추가 기능을 확인 하려면 `man ping`을 입력하면 된다.

traceroute host	패킷이 해당 호스트까지 도달하기 위한 경로를 보여준다. **traceroute**는 윈도우의 **tracert**와 동일한 범위를 탐색하지만 사용하는 네트워크 프로토콜은 다르다. traceroute는 UDP(User Datagram Protocol)을 사용하고 tracert는 ICMP(Internet Control Message Protocol)을 사용한다. 동일한 네트워크와 목적지 정보를 사용한다고 해도 traceroute와 tracert로 얻는 결과는 다르다. 둘 다 자신의 컴퓨터에서 목적지까지의 경로를 추적할 수 있다. 각각 얼마나 많은 경로를 거쳐야 하는지 최대 30홉까지 탐색한다. 패킷이 전송되는 것을 통해 해당 컴퓨터의 호스트명을 알 수도 있다. 예시: `traceroute www.hackerhighschool.org` `traceroute 216.92.116.13` 추가기능: 최대 홉 수를 설정할 수 있다. 예시: `traceroute -m 25 www.hackerhighschool.org` DNS 검색을 절약하기 위해 호스트명은 제외하고 IP 주소만 표시할 수 있다. 예시: `traceroute -n 216.92.116.13` 더 많은 추가기능을 확인 하려면 `man traceroute`를 입력하면 된다.

도구	기능
ifconfig	단일 명령어로 작동하며 현재 활성화되어 있는 네트워크 인터페이스 정보를 보여준다 (이더넷, PPP 등). 윈도우의 **ipconfig** 명령어와 유사하다. 추가기능: 보다 상세한 정보를 보여준다.(v는 verbose를 의미한다.) 예시: `ipconfig –v` en1 네트워크 인터페이스 정보만 표시한다. 예시: `ifconfig en1` 해당 네트워크 인터페이스와의 연결을 끊는다. 예시: `ifconfig en1 down` 해당 내트워크 인터페이스와 연결한다. 예시: `ifconfig en1 up` 주의 : 위의 명령어를 사용하기 위해서는 적합한 권한을 가지고 있어야 하기 때문에 명령어 앞에 **sudo**를 추가해야 한다. sudo를 추가했을 때 나오는 암호 입력 창에 암호를 입력하면 적합한 권한을 가지게 된다. 예시: `sudo ifconfig en1 up` 더 많은 추가기능을 확인 하려면 `man ifconfig`를 입력하면 된다.
netstat	네트워크의 상태 정보와 접속한 외부 네트워크의 정보를 보여준다. BSD와 같은 시스템에서는 **netstat**을 이용해 자신의 라우팅 테이블을 확인할 수 있다. 추가기능: 모든 네트워크 연결과 대기 중인 포트를 확인한다. 예시: `netstat –a` 라우팅 테이블을 보여준다. 예시: `netstat –r` IP주소와 포트번호를 숫자 순으로 나열한다. 예시: `netstat –nr` en1 네트워크 인터페이스의 정보를 보여준다. 예시: `netstat –r –ii en1` 더 많은 추가기능을 확인 하려면 `man netstat`을 입력하면 된다.

Exercises

2.12 자신의 컴퓨터의 이름과 IP 주소를 확인해 보자.

2.13 www.hackerhighschool.org까지의 경로와 중간 라우터들의 IP 주소를 확인해 보자.

2.14 윈도우에서 **tracert**를 사용해 www.hackerhighschool.org까지의 경로를 탐색하고 결과물을 **output.txt**파일로 만들어보자.

2.15 리눅스와 OSX에서 traceroute를 사용하여 동일한 네트워크 경로를 탐색하고 **output2OSX.txt**와 **output2linux.txt**파일로 만들어보자.
만들었다면 결과 파일을 잘 살펴보고 다음을 확인해 보자.

1. 탐색된 경로에 차이가 있는가?
2. ***이 포함된 문장을 찾았는가? 무엇을 의미하는가?
3. 동일한 테스트를 한 시간 이후에 다시 해보자. 결과가 달라졌는가?

윈도우, OSX 그리고 리눅스를 위한 기본 명령어 비교표

이텔릭체의 단어들은 반드시 입력해야 한다.

리눅스	맥OS	윈도우
command --help	command --help	command /h, command /?
man *command*	man *command*	help *command*
cp	cp	copy
rm	rm	del
mv	mv	move
mv	mv	ren
more, less, cat	more, less, cat	type
lpr	lpr	print
rm -R	rm -R	deltree
ls	ls	dir
cd	cd	cd
mkdir	mkdir	md
rmdir	rmdir	rd
netstat -r	netstat -r	route print
traceroute	traceroute	tracert
ping	ping	ping
ifconfig	ifconfig	ipconfig

오늘날의 십대들은 SNS, 인터넷 등을 통해 전 세계와 연결되어있다.
그러나 그들은 사기, 신원 위조, 개인정보 유출 등의
인터넷을 통한 공격방식과 이에 대한 대처방법을 잘 알지 못한다.
해커하이스쿨은 이러한 학생들을 위한 지침서가 될 것이다.

해커하이스쿨 프로젝트는 중·고등학생들에게 보안과 개인정보에 대한 인식을 제고하고 향상시킬 수 있는 교육자료이다

해커하이스쿨은 올바른 해커양성을 목적으로하며 이론과 실전파트로 구성되어있다. 해커는 지적능력, 창의성, 논리성을 모두 갖춰야한다. 우리는 학생들에게 일반적인 사이버보안 인식 혹은 IT 기술의 교육뿐만 아니라 어떻게 해커의 자질을 기르고 향상시킬 수 있는지를 가르칠 필요가 있다. 이 프로그램은 보안과 개인정보 인식교육 자료를 무상으로 포함하고 있으며 승인받은 중교등학교 선생님들을 위한 백앤드 서비스도 지원중이다. 다양한 언어로 지원되고 있으며 이 수업은 안전한 인터넷 사용, 웹 개인정보, 인터넷검색, 바이러스나 악성코드를 피하는법 윤리와 법 등을 포함한다.

HHS 프로그램은 ISECOM, 비영리단체, 보안 인식과 전문적인 보안 개발과 승인에 초점을 맞춘 오픈소스리서치 그룹에 의해 개발 되었다.

Hacker Highschool
SECURITY AWARENESS FOR TEENS

ISECOM