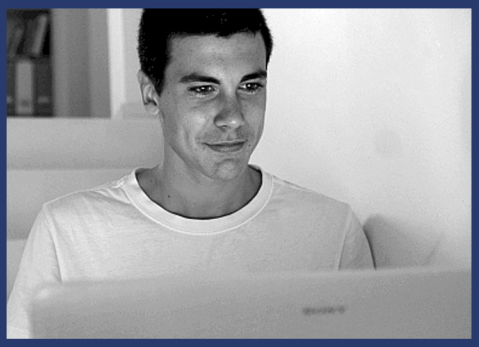
Hacker Highschool

SECURITY AWARENESS FOR TEENS



LESSON 4
PLAYING WITH DAEMONS

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
Chuck Truett, ISECOM
Kim Truett, ISECOM
Marco Ivaldi, ISECOM
Bob Monroe, ISECOM
Jaume Abella, ISECOM
Greg Playle, ISECOM
Simone Onofri, ISECOM
Guiomar Corral, Barcelona
Ashar Iqbal

한국어 번역팀

왕응석 EungSeok Wang
정진우 ZinWoo Jung
박영우 YongHoo Park
최홍선 HongSeon Choe

Introduction

세상에는 수많은 언어들과 방언들이 존재한다. 그 중 몇 가지 언어를 알고 있을 수도 있지만 실제로 세상을 여행하며 만난 사람들에게 자신이 배운 언어를 사용할 확률은 낮다고 할 수 있다.

물론, 수학이나 음악도 언어의 일종이라고 말하는 사람들이 있긴 하다. 하지만 현실적으로 생각해 보자. 그런 언어들을 이용해서 레몬을 곁들인 소다와 아이스크림을 주문할 수 있을까?

만약 그런 언어들을 사용해서 의사소통이 가능한 지역을 여행했다면 ISECOM에 비디오를 보내주길 바란다. 과연 소다를 주문하기 위해 어떤 악기들을 사용해야 하는지 보고 싶다. 어떤 소리가 나는지는 궁금하지 않지만 어떤 일이 벌어지는지는 정말 궁금하다.

수많은 사람들이 날마다 자신들의 언어를 통해 다른 사람들과 소통하고 있다. 이 세상의 모든 사람들이 단일 언어를 사용하지는 않지만 컴퓨터와 네트워크는 단일 언어를 사용한다.

이제부터 우리가 사용할 모델은 **클라이언트-서버 모델(client-server model)**이다. 컴퓨터(**호스트 또는 서버**)는 **서비스**(유닉스에서는 **daemons: disk access and execution monitors**로 불린다.)를 제공한다. 예를 들어 웹서버는 요청이 들어오면 웹 페이지를 제공한다.

하지만 실질적으로 요청 하는 것은 사람이 아니라 **클라이언트**인 웹 브라우저가 한다. 또는 컴퓨터라고 볼 수도 있다. 자신의 컴퓨터는 클라이언트인 동시에 서버가 될 수도 있다. 이런 상부상조가 바로 네트워킹의 묘미라고 할 수 있다.

이러한 메커니즘을 중첩시키면 바로 인터넷이 탄생한다. 수많은 컴퓨터들은 각각의 서비스를 제공한다. 그렇다면 클라이언트가 되기 위해서는 무엇이 필요하고 서버와 클라이언트의 관계를 **뒤바꾸는** 것은 무엇일까?

이제부터 알아보자.

서비스

만약 컴퓨터를 사용한다면 컴퓨터에 유용한 정보들뿐 아니라 쓸모없는 정보들도 자리를 차지하고 있다는 것을 잘 알고 있을 것이다. 또한 수많은 사람들이 컴퓨터를 가지고 있고 그들 역시 유용한 정보들을 가지고 있다는 것을 알고 있을 것이다.

이러한 사람들의 컴퓨터에는 각자의 관심 분야에 해당하는 정보가 축적되어 있다고 짐작할 수 있다. 그렇다면 이러한 정보들을 어디서 가져오는 것일까?

컴퓨터들은 각자의 포트와 프로토콜을 이용해서 서로 통신한다. 하지만 이 과정을 통해 전달되는 이진수 데이터들을 사용자들은 읽을 수 없다. 그렇기 때문에 컴퓨터가 데이터를 가져오고 사용자를 위해 해석한 후 일정한 형태로 제공하도록 하는 방법이 필요하다.

컴퓨터는 데이터를 전달하기 위해 **네트워크 서비스(Network Service)** 또는 간단하게 **서비스(Service)**라 불리는 것을 사용한다. 이러한 서비스들을 통해 사용자들은 웹페이지를 보고, 이메일을 보내거나, 채팅을 할 수 있다. 해당 서비스들은 특정 포트번호에 할당된다.

로컬 컴퓨터(Local Computer)라고 불리는 개인 컴퓨터들은 **클라이언트(Client)**라고 불리는 다양한 프로그램들을 사용해서 자신들이 받아들이는 정보를 해석한다. 사용자는 **서버, 토렌트, P2P**를 통해서 다양한 정보를 받을 수 있다.

물론 자신이 사용하는 컴퓨터를 서버나 서비스 제공자로 만들어서 다른 컴퓨터에 서비스를 제공할 수도 있다. 자신의 컴퓨터에 악성코드가 깔려있다면 자신도 모르는 사이에 몇 가지 서비스가 무단으로 제공되고 있을 수도 있다.

클라이언트에는 웹브라우저, 이메일, 채팅프로그램, 스카이프, 토렌트, RSS등이 있다. 이 프로그램들은 TCP/IP 모델의 **응용프로그램 계층(Application Layer)**의 응용프로그램에 해당한다. 응용프로그램 계층에 도달한 데이터는 하위 계층에서부터 캡슐화, 암호화, 복호화 등의 처리 과정의 거치며 올라와 사용자가 이해할 수 있는 형태로 가공된다.

HTTP와 Web

사람들은 "인터넷"을 **World Wide Web**으로 혼동하는 경우가 있다. World Wide Web은 **웹(Web)**으로 불리며 인터넷 자체가 아니라 이용할 수 있는 서비스들 중 일부분일 뿐이다. 브라우저를 이용해서 웹페이지를 보는 것을 의미한다.

인터넷은 데이터를 세상 곳곳으로 옮기는 컴퓨터, 라우터, 와이어, 케이블, 무선시스템 등으로 이루어져 있다. 웹은 이러한 체계 속의 일부분에 해당한다.

웹은 **HTTP(Hyper Text Transfer Protocol)**과 **웹 브라우저(Web Browser)**라고 불리는 응용프로그램을 사용한다. 다른 컴퓨터에서 전달된 정보는 자신의 컴퓨터가 사용하고 있는 HTTP 프로토콜과 80번 포트를 통해 들어온다. 그리고 웹브라우저는 받은 정보를 해석하고 자신의 컴퓨터에게 전달한다.

모든 브라우저들은 동일하게 제작되어 있지 않다. 브라우저들마다 제공하는 도구와 HTML이 약간씩 다르다. 보안과 프라이버시 문제들도 다른 방식으로 해결된다. 그러니 자신이 사용하는 브라우저의 성능을 파악하고 설정과 플러그인을 어떻게 구성해야 하는지 알고 있어야 보안과 프라이버시의 균형을 유지할 수 있다.

HTTP 프로토콜의 일부분인 **하이퍼텍스트(Hypertext)**는 비선형 구조의 읽기 방식을 지원한다. 일반적으로 사람들의 독서방식은 선형구조로 1페이지를 일고 2페이지로 나가는 순차적인 구조로 진행된다. 하지만 하이퍼텍스트를 이용하면 정보를 비선형방식으로 읽을 수 있다. 사용자는 자신이 원하는 텍스트를 읽기 위해 구간을 건너 뛸 수도 있고 이전 구간으로 돌아오거나 자신이 원하는 구간으로 다시 갈 수도 있다. 순차적으로 정보를 열람할 필요가 없다는 것이 하이퍼텍스트와 일반텍스트의 차이다.

하이퍼텍스트에서 사용하는 단어는 단어 자체뿐 아니라 이미지, 비디오, 음악과 연결되어 있다. 하이퍼텍스트는 웹에만 한정되어 있는 것이 아니다. 대부분의 워드프로세서는 사용자가 작성한 문서를 볼 수 있는 로컬 페이지를 웹이나 HTML에 만들어 준다. 해당 페이지는 자신의 웹브라우저를 통해 볼 수 있지만 다른 컴퓨터에서는 볼 수 없다.

자신만의 웹 페이지를 만드는 것은 어렵지 않다. 가장 쉬운 방법은 OpenOffice/LibreOffice Writer, Microsoft Word, WordPerfect와 같은 워드프로세서 프로그램을 사용하는 것이다. 해당 프로그램들을 사용해서 간단한 웹페이지를 만들고 텍스트, 하이퍼텍스트, 이미지 등을 결합할 수 있다. 이 외에도 Microsoft Notepad, Notepad++, SciTe, emacs등의 다양한 프로그램들이 있다.

하지만 이러한 프로그램들은 **CSS**, **스크립트(Script)**, 애니메이션과 같은 현란하고 다양한 디자인들을 사용하기 힘들다. 웹페이지 디자인을 꾸며주는 프로그램을 구입하기 위해서는 비용을 지불해야 한다. 이러한 프로그램들을 활용하면 보다 풍부한 효과를 웹페이지에 적용할 수 있지만 사용 방법이 훨씬 복잡하다. 하지만 전반적인 작업들을 보다 쉽게 진행할 수 있다. 좀 더 저렴한 방법으로는 HTML 작업과 스크립팅 언어(Scripting Language)에 특화된 텍스트 에디터를 구입하고 HTML문법, 스크립팅(Scripting)과 웹페이지에 사용할 코드를 배우는 것이다.

일단 웹페이지를 만들었다면 인터넷에 올리기 위한 컴퓨터가 필요하다. **ISP(Internet Service Provider)**는 웹 서버에 **웹 포스팅(Web Hosting)**을 제공한다.

여러분은 웹 서버를 자신의 집에서 자신의 컴퓨터로 구동할 수 있지만 몇 가지 해결해야 할 문제점들이 있다. 웹 서버에 저장되어 있는 정보들은 해당 서버가 실행되고 있어야만 이용 가능하다. 자신의 웹 서버가 자신이 잠든 사이에도 구동 되기를 원한다면 컴퓨터의 전원을 항상 켜 놓아야 하고 웹 서버에서 일어나는 각종 문제들을 지속적으로 모니터하고 처리해야 한다. 각종 문제들이란 하드웨어 문제, 악성코드, 외부의 공격, 프로그램 버그와 오류들이 있다. 또한 인터넷 연결을 항상 유지해야 하고 안정성 있고 빠른 서비스를 제공해야 한다. 다양한 문제들과 요구 사항들을 처리하기 위해 많은 사람들이 ISP에 의뢰해서 빠른 인터넷과 고정 IP 주소를 할당 받는다.

웹 호스팅 회사는 접속자가 접속하는 웹 페이지를 회사의 컴퓨터에 보관한다. 호스팅 서비스 신청자 대신에 공격을 받기에 걱정할 필요가 없다. 좋은 웹 호스팅 회사들은 여러 대의 예비 서버들과 백업 정책들을 가지고 있기 때문에 하드웨어 문제로 인한 웹사이트 오류를 걱정할 필요가 없다. 많은 기술자들이 외부의 공격과 프로그램 버그들로부터 서버를 지키고 인터넷 연결을 위한 구간을 다양하게 가지고 있기 때문에 정전 시에도 웹 페이지의 구동이 보장된다. 이러한 서비스들 덕분에 호스팅 서비스 사용자는 웹페이지를 디자인 하고 필요한 정보들을 호스팅 회사의 서버에 올린 후 컴퓨터를 끄고 잠자리에 들 수 있다. 비용을 지불하는 한, 해당 웹페이지는 전 세계 어디에서든지 접속할 수 있다.

무료로 웹 호스팅을 제공하는 회사들도 존재한다. 이런 회사들은 광고를 통해서 수익을 얻기 때문에 해당 회사를 이용한 웹페이지에 접속하면 광고를 의무적으로 봐야 한다. 하지만 구매는 자유이니 비용 걱정을 할 필요는 없다.

Exercises

4.1 웹페이지는 브라우저에 이미지, 비디오 등의 컨텐츠가 어디에 있는지 알려주는 단순한 문서라고 보면 된다. 페이지 소스(Page Source)를 보면 웹페이지가 어떻게 구성되어 있는지 볼 수 있다. 이제 자신이 사용하는 브라우저를 실행해서 ISECOM.ORG에 접속한 후 소스코드를 보자. 소스코드에 'meta'라는 단어들이 적혀있는 태그들이 몇 개 나오는데 첫 번째 관련 문장은 meta-charset="utf-8"라고 적혀있을 것이다. 이 문장이 의 미하는 것은 무엇인가? 중요한 점은 무엇인가?

4.2 메타 태그 3개를 더 찾아보고 무엇을 의미하는지 알아보자. 여러분은 인터넷을 통해서 해당 단어의 뜻을 찾을 수 있을 텐데 검색할 때 사용할 키워드를 아주 세심하게 골라야만 올바른 답을 얻을 수 있다는 것을 명심하자.

4.3 ISECOM.ORG 페이지의 소스를 자신의 컴퓨터에 저장해 보자.

브라우저에 드래그 하면 어떤 일이 발생하는가?
왜 그런 변화가 일어났다고 생각하는가?

4.4 ISECOM.ORG의 페이지 소스를 텍스트 에디터에서 실행하면 단순히 단어와 숫자들의 나열로 보일 것이다. 해당 문장들을 수정하고 저장한 후 웹 브라우저에 드래그 하면 페이지에 변화가 생길 것이다. 삭제한 것은 삭제한 채로 나오고 새로 타이핑 한 문장은 추가되어 나올 것이다. 이제 웹 페이지 소스에 변화를 줘서 자신의 이름을 추가한 후 다른 문장들의 글자들 보다 크고 굵게 만들어 보자.

보다 깊은 이해를 위해 해당 챕터의 마지막 부분에 있는
Feed Your Head: Playing With HTTP 를 참고하자.

이메일 - SMTP, POP, IMAP

인터넷의 두 번째 특징으로는 이메일(E-mail)을 들 수 있다. 여러분은 자신의 컴퓨터에 있는 이메일 클라이언트를 통해 메일 서버에 접속할 수 있다. 이메일 계정을 만들면 **user@domain**와 같은 독특한 아이디를 생성하고 비밀번호를 설정한다.

메일은 두 종류의 서버를 사용한다. 메일을 서로 보내는 서버인 **SMTP(Simple Mail Transfer Protocol)**와 메일을 컴퓨터로 가져오는 **POP(Post Office Protocol)**와 **IMAP(Internet Message Access Protocol)**이 있다.

SMTP 프로토콜은 메일을 보내기 위해 사용한다. SMTP는 보내는 이메일의 메시지 안에는 발신자, 수신자, 제목, 참고인, 메일내용 등 여러 가지 정보를 **포함**한다. 이전의 오래된 버전의 SMTP의 경우는 암호를 설정하지 않고 사용했기 때문에 오고 가는 메일들을 누구든지 볼 수 있었다. 초기의 인터넷은 소규모로 사용되었기 때문에 큰 문제가 되지 않았지만 현재는 **스팸메일**이나 **이메일 스푸핑(Spoofing)**과 같은 피해를 입을 수 있다. 다행히도 현재 대부분의 메일 서버들은 보안기능이 있는 SMTP를 사용하기 때문에 이메일을 사용하기 위해서는 신원을 증명해야 한다.

이후의 레슨에서는 스푸핑이 어떻게 이루어지고 이메일 헤더를 통해 어떻게 찾을 수 있는지 알아보겠다. 이러한 지식은 여러분을 강력하게 무장 시켜줄 것이다.

POP3(Post Office Protocol version 3)는 저장과 분배 기능을 하는 프로토콜이다. POP3는 사용자들이 접속해서 자신의 이메일을 받아갈 때까지 메일을 보관한다. POP3 방식은 메일을 보내고 받는데 적은 시간이 걸리고 오프라인 상태에서도 메일을 읽을 수 있기 때문에 전화모뎀을 이용하는 사람들에게 유리하다.

반면에 **IMAP** 방식에서는 메일을 서버에 저장한다. 많은 회사들이 사용하고 있는 메일서버가 바로 IMAP에 해당한다. IMAP에서는 자신의 메일박스에 여러 폴더들을 만들 수 있고 폴더에 들어있는 메일들을 폴더 간 이동시킬 수도 있다. 사용자가 IMAP 서버에 접속하면 서버는 메일박스에 들어가야 하는 폴더, 수신메일, 삭제한 메일 등을 동기화 시켜준다. 이러한 기능 덕분에 노트북, 키오스크, 스마트폰이나 태블릿 등과 같은 다양한 기기에서도 인터넷을 통해 자신의 메일을 확인할 수 있다. 추가로 사용 중인 컴퓨터에 개인 데이터 파일들을 다운로드 하거나 저장할 수 있다.

하지만 IMAP에는 두 가지 단점이 있다. 첫째, 메일서버를 사용해서 교환해야 하는 정보의 양이 늘어나기 때문에 빠른 접속과 많은 시간이 요구된다. 둘째, 메일용량이 한정되어 있기 때문에 메일서버가 분배하는 메일 박스의 용량 이상으로 메일들을 보관할 수 없다. 만약 허용량을 초과하면 메일을 지우거나 유료 공간을 구매하기 전에는 메일 수신이 불가능하다. 그렇기 때문에 자신의 컴퓨터로 메일을 다운 받거나 받은 메일들을 정리하여 데이터 관리를 꾸준히 해야 한다. 특히 첨부 파일과 함께 온 메일의 경우 더욱 많은 관심이 필요하다. 하지만 요즘과 같이 대용량의 이메일을 무료로 사용할 수 있는 시대에 이러한 걱정은 시간 낭비일 뿐이다. 다만 서버가 해킹 당해 여러 사람들의 이메일이 유출되는 상황은 경계해야 한다.

POP와 IMAP 서버 둘의 계정에 접속하기 위해서는 암호가 필요하다. 둘 다 암호와 많은 데이터를 원본 형태로 보내기 때문에 다른 사람들이 볼 수 있는 취약점을 가지고 있다. 로그인 과정과 메일의 내용을 감추기 위해서는 SSL과 같은 암호화 과정이 필요하다. 이런 과정을 가능케 하기 위해서 많은 이메일 클라이언트가 SSL 체크박스를 사용한다.

이메일 클라이언트에서 '보내기' 버튼을 누르면 두 가지 일이 발생한다. 우선은 사용자의 클라이언트는 POP서버에 로그인을 했음에도 SMTP서버에 로그인을 시도한다. 그 후에 SMTP 프로토콜을 통해 메일을 발송한다.

1990년도 중반에는 이러한 과정이 아주 성가신 절차로 여겨졌다. 사용자들은 POP서버에 자신의 아이디와 암호를 전달하고 난 후에 수신 받은 메일을 다운로드 받는다. 그 후 SMTP서버는 POP서버에게 사용자의 인증절차를 확인하고 난 다음 메일을 발송한다. 시간절약에 도움이 되는 절차이다.

한 가지 알아두어야 할 사항이 있다. 암호가 보호된다는 것이 이메일을 통한 정보전달의 안전성을 보증하지는 않는다. 대부분의 POP 클라이언트들과 서버들은 자신들에게 전달되는 암호를 항시 메일서버와 공유하고 있다. 이 과정에 암호화와 같은 보안시스템은 적용되지 않는다. 이메일을 전달받은 사람들이 암호를 알 수 있는 것은 아니지만 악의적인 목적을 가진 사람들은 빈틈을 노려 암호와 이메일의 내용들을 탈취할 수 있다. 이메일의 보안기능을 강화하기 위한 방법은 이후의 **Lesson 9 : 이메일 해킹**에서 배우게 될 것이다.

Exercises

4.5 자신이 가지고 있는 이메일 계정으로 자신에게 이메일을 보내보자.
그리고 동일한 이메일을 다른 계정에도 보내보자.
두 계정에 동일한 이메일이 도착하는데 얼마나 걸리는가?
시간 차이가 발생하는가?

4.6 자신이 받은 스팸메일들 중 하나를 열어보자.
누가 보냈는지 알 수 있는가?
스팸메일에 숨겨진 정보가 들어있는가?
만약 있다면 어떻게 확인할 수 있는가?

4.7 보내고자 하는 이메일의 발송시간을 조정할 수 있는가?
발송시간 조정을 통해 다른 사람들을 곤란하게 만들 만한 방법이 있는가?

IRC

인터넷의 통제할 수 없는 특성을 보여주는 하나의 지표로 **IRC(Internet Relay Chat)**가 있다. IRC를 이용하면 자신의 의견이나 생각을 누구의 눈치도 보지 않고 말할 수 있다. IRC는 **유즈넷(Usenet)**과 **뉴스그룹(News Group)**으로도 불린다. 뉴스그룹들은 자신들만의 명칭을 가지고 있다.

IRC는 많은 사람들에게 익숙한 채팅방과 매우 유사하다. 몇 가지 다른 점으로는 **네티켓(netiquette)**을 지켜야 한다는 것 외에는 특정한 규칙이 없고 관리하는 사람이 존재하지 않는다는 것이다. IRC를 통해 자신이 찾고자 했던 정보를 발견하거나 전혀 알지 못했던 사실들을 발견할 수 있다.

IRC에서 지켜야 할 규칙들은 다른 채팅방에서 준수하는 것들과 동일하다. 자신의 진짜 이름을 사용하지 말고 전화번호, 주소, 은행계좌와 같은 개인정보를 알려주면 안 된다. 그저 돌아다니면서 즐기면 된다. 하지만 자신이 이용할 수 있는 정보와 파일들에 바이러스가 들어있을 수 있으니 주의해야 하며 무해한 사람들만 있지는 않다는 것을 명심해야 한다.

IRC 서버들 간에 교류되는 정보들은 암호화를 거치지 않기 때문에 주의해야 한다. 다른 IRC 이용자와 전용방을 만들어도 이는 마찬가지다. 닉네임을 사용하는 것으로 지킬 수 있는 프라이버시도 한정되어 있다. 만약 인터넷을 이용한 악의적인 공격을 계획하고 있다면 여러 계정에서 사용하는 닉네임들을 다르게 만드는 것이 좋다. 중복되는 닉네임들을 사용하는 것은 경찰이나 다른 공격자들에게 추적 당할 수 있는 약점이 될 수 있다.

IRC에서 다루는 주제들은 “채널(Channel)"로 대변된다. 수없이 많은 채널들이 존재하고 있고 그

중 많은 곳의 리스트를 아래의 URL주소를 통해 볼 수 있다.

http://www.nic.funet.fi/~irc/channels.html

만약 해당 채널에 관리자가 있다면 규칙을 지키지 않는 내용들을 신고할 수 있고 해당 작성자를 **강제 퇴장** 시킬 수 있다. 특정 인물이 올리는 말이나 메시지가 마음에 들지 않는다면 해당 아이디를 수신거부 할 수 있다.

Exercises

4.8 보안에 관해서 정보를 교류하는 IRC를 찾아보자.
공개방에 참여하기 위해서는 어떻게 해야 하는가?
다른 사람과의 전용방을 만들려면 어떻게 해야 하는가?

4.9 IRC가 사용하는 포트번호는 무엇인가?

4.10 IRC를 통해 파일을 주고받을 수 있다.
어떤 과정을 거쳐야 하는가?
IRC를 통해 파일을 주고받고 싶은가?

4.11 MIME와 SMIME의 가장 큰 차이점은 무엇인가?
S라는 단어를 봤을 때 Secure와 같은 단어가 떠오르는가?

FTP

FTP(File Transfer Protocol)는 일반적으로 20번과 21번 포트를 사용한다. FTP를 통해 두 컴퓨터는 파일들을 주고받을 수 있다. 비용을 내야 사용할 수 있는 FTP 말고도 리눅스 배포판의 ISO와 같은 익명 FTP(Anonymous FTP)를 이용하여 무료로 자료를 전송할 수 있다.

익명 FTP는 한때 자료 교류를 위한 대중적인 방법으로 사용되었다. 익명 FTP 서버들 중에는 비합법적인 서버들도 많지만 합법적으로 운영되는 익명 FTP 서버들 역시 굉장히 많다. 여러분은 인터넷 검색을 통해서 익명 FTP 서비스를 제공하는 서버를 찾을 수 있을 것이다. 하지만 FTP 로그인정보는 암호화 되어 서버에 전달되지 않는다. 암호화와 같은 보안이 적용된 FTP(SFTP) 서버도 존재하지만 수가 매우 드물다.

여러분들은 인터넷 브라우저와 FTP프로토콜을 통해 FTP 서버에 접속할 수 있다. 인터넷 브라우저 외에도 파일관리시스템(File Management Program)이라는 FTP에 특화된 클라이언트가 존재한다. FTP 서버에 로그인 하면 자신의 컴퓨터에서 파일들을 이리 저리 옮기는 것과 똑같은 방법으로 자신의

컴퓨터의 자료들을 업로드 하거나 다운로드 할 수 있다. FTP 서버가 위치한 곳과의 거리가 상당하기 때문에 시간이 좀 더 걸릴 뿐이다.

Exercises

4.12 OSX와 리눅스는 FTP 실행 명령어를 가지고 있다.
FTP에 접속하기 위해서는 명령어창을 열고 다음의 명령어를 입력하면 된다.

```
ftp
```

다음의 명령어를 치면 FTP와 관련된 명령어 리스트를 보여준다.

```
ftp> help
Commands may be abbreviated. Commands are:
!           delete          literal         prompt          send
?           debug           ls              put             status
append      dir             mdelete         pwd             trace
ascii       disconnect      mdir            quit            type
bell        get             mget            quote           user
binary      glob            mkdir           recv            verbose
bye         hash            mls             remotehelp
cd          help            mput            rename
close       lcd             open            rmdir
```

기본 명령어

'ftp.도메인명.이름'의 FTP 서버에 접속한다.

```
ftp> open ftp.domain.name
```

작업 중인 디렉토리와 파일들의 리스트를 보여준다.

```
ftp> ls
```

또는

```
ftp> dir
```

작업 중인 디렉토리를 다른 디렉토리로 바꾼다.

```
ftp> cd newdir
```

FTP에 접속한 원격 컴퓨터의 파일 여러 개를 자신의 컴퓨터로 다운로드 한다.

```
ftp> get filename
```

원격 컴퓨터의 파일 여러 개를 자신의 컴퓨터로 다운로드 한다.

```
ftp> mget file1 file2 file3
```

자신의 컴퓨터에 파일을 원격 컴퓨터에 업로드 한다.

```
ftp> put filename
```

FTP서버와의 접속을 끊는다.

```
ftp> close
```

자신의 FTP클라이언트를 종료한다.

```
ftp> quit
```

단계별 FTP 세션

익명 FTP에 접속하기 위해서는 우선 자신의 FTP 클라이언트를 실행해야 한다.

```
ftp
```

다음의 명령어를 통해 FTP 서버에 접속한다.

```
ftp> open anon.server
```

anon.server대신 실제 서버명을 입력해야 한다.
FTP 서버가 사용자의 접속을 승인하기 위해 사용자의 컴퓨터에 ID를 요청한다.

```
Connected to anon.server.
220 ProFTPD Server (Welcome . . . )
User (anon.server:(none)):
```

대부분의 익명 FTP에서는 anonymous나 ftp를 ID로 사용한다. 해당 ID를 입력하면 FTP서버는 익명 사용자로 인식하고 암호를 어떤 걸로 사용해야 하는지 알려준다.

```
331 Anonymous login ok, send your complete email address as your password.
Password:
```

대부분의 서버에서 입력된 이메일 주소의 진위 여부를 확인하지 않기 때문에 임의의 이메일 주소를 입

력해도 접속할 수 있다. 이런 행동은 네티켓을 위반하는 것이지만 자신의 개인정보를 입력하는 것보다 훨씬 안전한 방법이다. 암호를 입력했다면 서버에서는 자신의 컴퓨터에 환영 메시지를 전달한다.

```
230-
      Welcome to ftp.anon.server, the public ftp server of anon.server. We
      hope you find what you're looking for.
      If you have any problems or questions, please send email to
      ftpadmin@anon.server
      Thanks!
230 Anonymous access granted, restrictions apply.
```

이제부터 여러분은 ls, dir, cd 등의 명령어를 통해 서버에 있는 파일들을 자신의 컴퓨터에 다운받을 수 있다.

Exercises

4.13 연습문제의 예제들을 활용하여 FTP 서버에서 파일을 다운받아 보자.

4.14 인터넷 검색엔진을 통해 '이상한 나라의 엘리스'영화(또는 다른 원하는 영화)를 가지고 있는 익명 FTP서버를 찾고 명령어창을 활용하여 FTP 클라이언트를 실행한 후 해당 파일을 다운받아 보자.

4.15 어떤 FTP 클라이언트가 가장 효과적인가?
클라이언트가 자동 명령어 입력 기능과 GUI를 제공하는가?
명령어창에서 실행할 수 없는 기능이 있는가?

4.16 자신의 컴퓨터를 FTP 서버로 활용할 수 있는가?

텔넷과 SSH

텔넷(Telnet)은 로컬 컴퓨터에서 원거리에 있는 컴퓨터에 다양한 명령을 보낼 수 있도록 해주는 프로토콜이다. 로컬 컴퓨터는 원격 접속을 통해 해당 컴퓨터에서 특정 기능을 수행하고 데이터를 받아올 수 있다. **SSH(Secure Shell)**는 텔넷에 암호화 기능이 추가되어 보안기능이 향상된 프로토콜이다.

윈도우, OSX, 리눅스는 명령어창을 통해 텔넷 클라이언트를 실행할 수 있다. 텔넷을 실행하기 위한 명령어는 다음과 같다.

```
telnet
```

텔넷 서버에 접속하기 위해서는 서버의 관리자에게 계정과 암호를 할당 받아야 한다. 텔넷 프로그램을 통해 할 수 있는 일이 굉장히 많고 그 중 몇 가지는 해당 서버에 타격을 줄 수 있기 때문에 필요한 절차를 거쳐야 한다.

과거의 텔넷은 원거리에서 서버 관리와 사용자 지원을 하기 위해서 사용되었다. 이러한 기능들은 이전 인터넷의 중요한 활용 방법이었지만 지금은 거의 사용되지 않는다.

이 외에도 텔넷은 이메일 수신/발신과 웹페이지의 소스코드 확인과 같은 다양한 기능들을 가지고 있다. 이러한 기능들을 사용하는 것은 물론 합법이지만 얼마든지 오용될 여지가 있다. 텔넷을 이용하면 이메일의 제목과 내용을 확인할 수 있고, 삭제도 가능하다.

텔넷이 가진 많은 취약점들을 보완하기 위해 만들어진 것이 SSH이며 지금도 인터넷에서는 계속해서 취약점들을 찾아내고 있다.

Game On: Command Me

할아버지의 두꺼운 안경에 비치고 있는 모니터 화면은 명령어가 입력되기를 기다리고 있는 것 같았다. 할아버지의 엷은 회색빛 머리칼은 주름진 이마 사이로 내려와 있었다. 할아버지는 키보드를 두드리고 있었고 제이스는 가만히 앉아 할아버지의 키보드 연주를 감상하고 있었다. 할아버지는 미소 지으며 제이스를 쳐다보고는 말했다.
“제이스, 이제부터 너에게 새로운 세상을 보여줄테니 안전벨트를 꼭 매렴.”

제이스는 바닥에 발이 간신히 닿을만한 의자에 앉아 할아버지를 지켜보고 있을 때 근처에 있는 작은 박스에서 갑자기 전화 신호음이 들려왔다. 하얀색 박스는 녹색과 적색의 불빛이 나오고 있었고 신호음은 점점 믹서기에 갈리는 오리 소리처럼 변해갔다. 할아버지는 흥미롭다는 표정을 지으며 컴퓨터 화면에 집중했다. 괴상한 오리 소리는 멈췄고 상자의 불빛은 녹색으로 바뀌어 있었다.
“이걸 보렴”

주로 할아버지가 “이걸 보렴”이라는 말을 할 때는 무언가가 터지거나 연기가 피어오르는 일들이 발생했었다. 아니면 할머니가 미친 듯이 화를 낼만한 일이 발생했다. 제이스는 스릴 넘치는 일을 예견하는 이 마법의 단어를 좋아했다.

검정색으로 가득했던 컴퓨터 화면이 아스키코드 배너에 둘러싸인 “클라인 게시판 시스템(Bulletin Board System)에 오신 것을 환영합니다.”라는 문구를 보여줬다. “드디어 들어왔어.”라며 기뻐하던 할아버지는 8살짜리 제이스에게 하이파이브를 시도했고 몇 센티 빗나간 손은 하마터면 제이스의

얼굴을 칠 뻔 했다. 제이스는 웃음을 터뜨렸고 할아버지도 마찬가지였다.

두 사람은 키보드와 컴퓨터 화면을 번갈아 쳐다보았고 할아버지는 손가락을 마주 비볐다. 제이스는 지금 무슨 일이 벌어지고 있는지 생각하며 할아버지가 키보드에 명령어들을 치는 것을 바라보았다. 할아버지는 머리를 키보드 위에 두고 먹이를 노리는 독수리처럼 머리를 이리저리 움직이며 키보드를 연주하고 있었다. 갑자기 의자에 등을 기댄 할아버지는 깜박 했던 것을 생각해낸 듯이 제이스를 쳐다보고 말했다.
"제이스, 내가 지금 뭘 하고 있는지 설명해 줘야겠구나. 나는 지금 전화선을 이용해서 다른 컴퓨터에 접속하고 있단다. 저 괴상한 소리가 나는 작은 박스는 "모뎀"이라는 건데 디지털 신호와 아날로그 신호를 반대로 변환 시켜주는 장치란다."

기회가 될 때마다 모뎀을 가지고 노는 할아버지 덕분에 이미 제이스는 모뎀에 대해 잘 알고 있었다. 평상시에는 48볼트의 전기가 필요하고 전화신호를 사용할 때는 90볼트가 필요하다는 것도 알고 있다. 아마 웬만한 전문가 보다 더 많은 정보를 알고 있을 것이다. POTS(Plain Old Telephone System)는 할아버지와 제이스 사이에서만 통하는 단어로 할머니는 그게 무엇을 의미하는지 전혀 모르고 있다.

전화선은 다른 사람에게 도청될 수 있지만 전압기를 통해 감지할 수 있는 방법이 있다. 누군가가 도청을 시도하면 전화선의 전압은 급격히 올랐다가 계속해서 미약하게 증가한다. 제이스는 할아버지가 할머니 보다 전압기를 더 좋아하는 것 같다고 생각했다. 밖에 나갈 때는 항상 들고 다녔고 심지어 "발레리"라는 이름까지 지어주었다. 발레리는 제이스와 더불어 할아버지의 절친들 중 한명이다.

제이스는 할아버지가 진행하고 있는 수업에 다시 집중했다. 할아버지는 모뎀이 아날로그 신호인 소리를 디지털 신호로 바꾸는 것에 관해 이야기 하고 있었다.
"내가 접속한 컴퓨터는 컴퓨터에서 제공하는 서비스는 무엇이든지 사용할 수 있도록 내게 허가해 준단다."

제이스는 자신이 처음 들어보는 단어인 "서비스"에 귀를 쫑긋했다.
"할아버지 '서비스'가 무엇을 뜻하는 거죠?"

제이스는 할아버지가 바로 답을 해 줄 것을 기대하며 질문했다.
"아주 좋은 질문이구나."

할아버지는 제이스가 이런 질문을 할 것을 예상하고 있었다.
"네 컴퓨터는 지금 다른 컴퓨터의 네트워크에 접속해 있고 나는 다른 컴퓨터에도 접속할 수 있는 능력이 있단다. 그리고 모뎀은 내가 접속한 컴퓨터들과 통신할 수 있는 기능을 제공하고 컴퓨터들은 파

일, 정보, 대화 상대를 제공해 주지. 이런 컴퓨터들이 제공하는 서비스에는 FTP, 유즈넷, IRC, 텔넷, 이메일 등이 있고.”

제이스는 할아버지의 대답이 별로 만족스럽지 않았고 다음 질문들을 장전하기 시작했다. 곧이어 제이스의 질문들이 속사포같이 터져나왔다.
“FTP가 뭐죠? MIC는요? 텔넷은 어디에 있는거죠? 이메일은 어떤 우표를 사용하죠? 디지털 세계는 색이 존재 하나요? 누가 유즈넷을 발명했죠? 왜 이런 것들을 서비스라고 부르는 거에요? 아기는 어디서 오는 거죠? 젤리는 어디서 만들어지고요?”

할아버지는 연달아 쏟아지는 질문들을 피하기 위해 귀를 가리고 소리쳤다.
“잠깐, 잠깐, 쫌 천천히 말해봐.”

Game Over

DNS

여러분이 친구에게 전화를 하기 위해서는 정확한 전화번호를 알아야 하는 것처럼 다른 컴퓨터에 접속하기 위해서는 컴퓨터 번호를 정확하게 알고 있어야 한다. 이 전 레슨을 기억한다면 컴퓨터 번호가 IP 주소라고 불린다는 것을 알고 있을 것이다.

IP 주소는 컴퓨터에 의해 쉽게 관리될 수 있지만 사람들에게는 **도메인 이름(domain names)**이 훨씬 사용하기 편리하다. 만약 Hacker Highschool 홈페이지에 접속하고 싶다면 웹브라우저의 주소창에 www.hackerhighschool.org를 타이핑하면 된다. 하지만 컴퓨터는 영어로 된 주소를 이해하지 못하기 때문에 IP 주소로 변경해야 한다. 만약 홈페이지가 100여개만 있다면 컴퓨터에 자체적으로 주소 목록을 가지고(**호스트 파일(hosts file)**같은 것) IP 주소를 찾아낼 수 있을 것이다. 하지만 인터넷에서 접속할 수 있는 서버들의 수는 수없이 많고 도메인명에 해당하는 IP 주소도 계속해서 변하기 때문에 다른 방법이 필요하다.

DNS(Domain Name Service)는 도메인명에 해당하는 IP주소를 찾아주거나 그 반대의 기능을 한다. 주소창에 도메인명이 포함된 주소를 입력하면 웹브라우저는 DNS 서버에 접속해서 도메인명에 할당된 IP 주소를 해당 페이지에 접속할 수 있도록 해준다.

하지만 DNS서버에 해당 도메인의 IP 주소가 없다면 다른 DNS서버에 해당 IP 주소를 가지고 있는지 요청하고 IP 주소를 찾아내거나 도메인명이 잘못 됐다는 것을 확인할 때까지 반복한다.

Exercises

4.17 명령어창을 키고 자신의 컴퓨터 IP 주소를 확인해 보자.
어떤 명령어를 사용했는가?
자신의 IP 주소는 무엇인가?

4.18 자신이 접속하는 DNS서버의 IP 주소를 확인해 보자.
어떤 명령어를 사용했는가?
DNS서버의 IP 주소는 무엇인가?

4.19 www.isecom.org를 ping 명령어로 조사해 보자.
어떤 IP 주소가 뜨는가?

4.20 자신이 사용하는 DNS서버를 다른 것으로 바꿀 수 있는가?
가능하다면 다른 DNS서버를 사용할 수 있도록 컴퓨터의 설정을 바꾸고 ping으로 www.isecom.org를 다시 한번 조사해보자.
이전과 차이점이 있는가?
왜 그렇다고 생각하는가?

DHCP

DHCP(Dynamic Host Configuration Protocol)는 로컬 네트워크 서버가 IP 주소를 분배할 수 있도록 해준다. 해당 서버는 일정량의 IP 주소를 할당 받고 컴퓨터가 네트워크에 접속하면 IP 주소를 할당해 주고 컴퓨터가 종료되면 IP 주소를 회수하여 다른 컴퓨터가 사용할 수 있도록 한다.

이는 대규모 네트워크에서 유용하다. 사용자들이 정적 IP 주소를 일일이 할당 받을 필요가 없기 때문에 DHCP를 활용해도 무방하다. 컴퓨터가 네트워크에 접속해서 가장 먼저 하는 것은 DHCP 서버에서 IP 주소를 할당 받는 것이다. 일단 IP 주소를 할당 받으면 컴퓨터는 네트워크의 모든 서비스를 사용할 수 있다.

이제 DHCP에 대해 생각해 보자. 대부분의 무선 네트워크는 DHCP를 제공하고 이는 해당 네트워크에 접근하는 사람들은 누구든지 IP 주소를 받을 수 있다는 뜻이다. 만약 여러분이 카페를 운영하면 상관없지만 보안이 필요한 사무실에서 근무를 한다면 고정 IP 주소가 필요하다. 즉 상황에 따라 다른 기준을 적용해야 한다.

인터넷 연결

굉장히 오래 전에는 모뎀을 통해 인터넷에 접속했었다. 모뎀(Modem)은 비트 정보를 소리로 또는 소

리를 비트 정보로 바꿔주는 **mod**ulating과 **dem**odulating의 앞 글자를 딴 용어이다. 모뎀의 속도는 **보(Baud)**와 **bps**로 나타낸다. 높은 보와 bps도 중요하지만 우선 어떤 용도로 모뎀을 사용할 건지 생각해야 한다. **MUD(Multi User Dungeon)**과 같은 프로그램은 300 보의 모뎀으로 충분하지만 더 넓은 대역폭이 필요한 경우에는 DSL이나 더 빠른 케이블 모뎀을 사용해야 한다.

ISPs

인터넷을 사용하고 싶다고 곧바로 할 수 있는 것은 아니다. 우선 인터넷으로 자신의 컴퓨터를 연결 시켜주는 서버가 필요하다. 서버에 할당되어 있는 업무는 굉장히 광범위하고 24시간 유지되어야 하기 때문에 **ISP(Internet Service Provider)**의 도움을 받아 서버를 사용한다.

ISP는 인터넷 접속점(Point of Presence)을 안정적으로 지속시키고 서버들을 가동시켜 사용자들이 여러 서비스를 이용할 수 있도록 한다. 여러분들 역시 ISP가 하는 일들을 할 수 있다. 자신의 컴퓨터를 메일서버로 운영할 수 있지만 잠깐의 정보교환을 위해 계속해서 네트워크에 접속되어 있어야 하고 전원이 켜져 있어야 한다. ISP는 사용자들이 번거롭고 하기 힘든 일들을 통합적으로 해주며 사용자들이 기다리지 않고 원할 때 서비스를 사용할 수 있도록 해준다. ISP들은 빠른 속도의 컴퓨터들을 사용하여 네트워크 **접속지점(NAP: Network Access Point)**에 접속한다. NAP는 **백본(Backbone)**이라 불리는 초고속망을 통하여 서로간의 연결을 유지한다. 이러한 연결들이 바로 인터넷이다.

POTS

POTS(Plain Old Telephone Service)는 한때 가장 각광받던 인터넷 접속수단이었다. POTS는 속도가 느리다는 단점이 있지만 광범위한 적용성이라는 장점을 가지고 있다. 대부분의 ISP들은 다양한 지역번호를 가지고 있고 많은 사람들이 아직도 유선정화를 가지고 있다. 만약 음향모뎀과 충분한 동전들을 가지고 있다면 공중전화를 통해 인터넷에 접속할 수 있다.

POTS의 속도는 느리다. 가장 빠른 전화모뎀의 속도는 56,600bps로 알려져 있지만 실제 속도는 그렇지 않다. 전력제한으로 인해 다운로드 속도는 53,00bps 정도로 제한되고 실질적인 속도는 이보다 더 느리다. DSL이나 케이블 모뎀에 비교하면 더욱 느리게 느껴질 것이다.

전화 서비스는 어디에서나 이용할 수 있고 POTS를 기반으로 하는 ISP들은 상대적으로 저렴하거나 무료인 인터넷 서비스를 제공한다. 하지만 인터넷으로 영화를 다운받을 때 POTS를 사용한다면 하루 종일 혹은 일주일 내내 전화선을 사용해야 할 것이다. POTS는 친구 사이에 이메일을 보낼 때나 사용하는 게 좋다. 만약 텔넷을 사용한다면 오래된 DOS기반 컴퓨터를 통해서도 인터넷에 접속할 수 있다.

DSL

DSL(Digital Subscriber Line)은 이미 POTS를 위해 깔려있는 선로를 통해 인터넷에 연결할 수 있는 획기적인 방법이다. 가장 큰 장점은 아날로그 모뎀보다 빠르고 지속적인 접속을 보장한다는 것이다. 또한 전화를 사용하고 있을 때도 인터넷을 계속 사용할 수 있다. 단점으로는 전화회사의 스위칭 장비의 거리에 따라 서비스가 제한된다는 점이다. 만약 먼 거리에서 살고 있다면 DSL을 사용하기 어려울 것이다.

케이블 모뎀

케이블 모뎀(Cable Modem)은 기존의 전화선을 사용하지 않고 케이블 회사에서 제공하는 동축선이나 광섬유를 사용한다. DSL과 마찬가지로 케이블 모뎀을 통해 인터넷을 하는 동안에도 자유롭게 전화를 사용할 수 있고 지속적인 인터넷 연결을 보장한다. 그러면서도 속도는 DSL보다 더 빠르다.

케이블 선로는 몇 가지 단점을 가지고 있다. 케이블 선로는 여러 사람들이 공유해서 쓰기 때문에 동시에 사용하는 사람들이 늘어나면 속도가 느려진다. 또한 케이블 연결을 위해서는 케이블 회사에서 우선 선로를 깔아야만 한다. 그리고 가장 심각한 문제는 다른 사람들이 케이블을 통해 전달되는 정보를 탈취할 수 있다는 것이다. 그러니 자신의 정보를 지키기 위해서는 방화벽을 설치해야 한다.

Wimax

와이맥스(Wimax)는 무선 연결방식으로 유선연결이 힘들거나 많은 비용이 드는 장소에서 주로 사용된다. 신호의 세기는 주변 빌딩이나 나무처럼 큰 물체에 영향을 받는다. 와이맥스 버전들 중 일부는 고정 AP(Access Point)가 필요하지만 대부분 버전의 경우 광범위한 장소에서 모바일 접속이 가능하다.

Wifi

와이파이(Wifi)는 ISP를 통해 인터넷에 접속하는 방식이 아니라 집이나 카페, 쇼핑몰 등의 상업시설에서 인터넷접속을 위해 사용하는 무선랜 기술이다. 스마트폰이나 노트북을 통해 많은 사람들이 와이파이를 사용하기 때문에 공격의 타겟이 되기 쉽다. 공공장소에서 와이파이를 사용하는 것은 사람들이 많은 장소에서 헐벗고 있는 것과 같다. 자신을 가리고 다른 사람들이 볼 수 없도록 해야 한다. 후반부 레슨에서는 무선보안에 관해 배우게 될 것이다.

Exercises

4.21 집에서 어떤 방식으로 인터넷에 접속하는가?

4.22 네트워크에서 누구를 만날 수 있는가?

4.23 인터넷 속도가 얼마나 빠른가?
ISP에 전화하지 않고 어떻게 인터넷 속도를 향상시킬 수 있는가?

4.24 이용 중인 ISP가 제공하는 추가 서비스에는 무엇이 있는가?

4.25 자신의 컴퓨터를 통해 제공하는 서비스가 있다면 무엇인가?

Feed Your Head : Playing With HTTP

HTTP는 RFC 1.0과 1.1에서 정의하는 것처럼 TCP/IP 스택의 최상위에 위치해 있다.

1.0과 1.1 사이에는 상당한 변화가 있었다. 확장성, 케싱, 대역폭 최적화, 네트워크 연결 관리, 메시지 전송, 인터넷 주소 변환, 오류감지, 보안, 무결성, 가용성, 인증, 컨텐츠 타협에 상당한 업그레이드가 있었다. 덕분에 웹서버에서 정보를 얻는 것이 보다 효율적으로 변했다.

기본적으로 HTTP는 stateless protocol로서 HTTP 요청(Request)을 서버에 보내고 서버는 그에 응답(Response)한다.

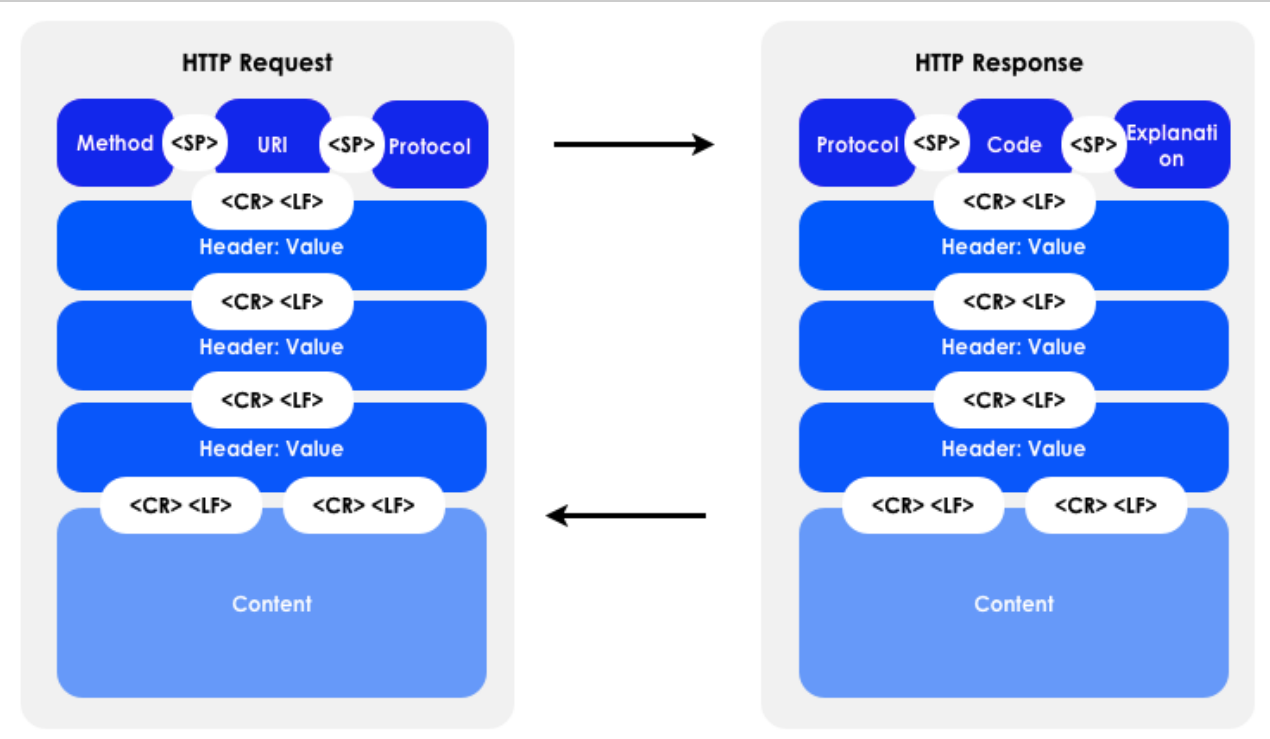


Figure 4.1: HTTP

다양한 정보를 얻기 위해서는 서버에 적합한 명령어를 보내야 하며 그러기 위해선 다음의 도구들이 필요하다.

- netcat: the TCP/IP tool kit
- curl: the HTTP tool kit
- proxy: like OWASP ZAP or Burpsuite free

HHS 통신 구간 HTTP 서버 스니핑

프록시를 사용해서 http://www.hackerhighschool.org에 접속한 후 요청방법(Request Method)을 알아내 보자.

요청문:

```
GET / HTTP/1.1
Host: www.hackerhighschool.org
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.8; rv:11.0)
Gecko/20100101 Firefox/11.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Proxy-Connection: keep-alive
```

응답문:

```
HTTP/1.1 200 OK
Content-Length: 10376
Date: Fri, 03 Feb 2013 09:11:17 GMT
Server: Apache/2.2.22
Last-Modified: Mon, 06 Feb 2013 09:31:18 GMT
ETag: "2f42-4b8485316c580"
Accept-Ranges: bytes
Identity: The Institute for Security and Open Methodologies, The
Institute for Security and Open Methodologies
P3P: Not supported at this time, Not supported at this time
Content-Type: text/html
```

```
Connection: keep-alive

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"[]><html
xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="en-US"
xml:lang="en"><head><meta http-equiv="Content-Type"
content="text/html; charset=UTF-8" /><title>Hacker Highschool -
Security Awareness for Teens</title>
[...]
```

Exercises

4.26 프록시에서 얻은 요청들을 확인해 보자.

4.27 헤더를 통해 무엇을 알 수 있는가?

수동 접속방법

Netcat을 통한 호스트 포트설정을 활용하여 웹서버에 접속할 수 있다.
다음 명령어를 입력하고 엔터를 두 번 눌러보자.

```
    nc www.hackerhighschool.org 80
```

그러면 다음과 같은 문장이 나온다.

```
    GET / HTTP/1.0
```

이에 대한 서버의 응답문은 다음과 같다.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"[]>
<html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="en-US"
xml:lang="en"><head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>ISECOM - Institute for Security and Open Methodologies</title>
<meta name="description" content="Description" />
```

위에서 알 수 있는 것처럼 해당 응답문들은 hackerhighschool.org가 아니라 isecom.org에서 발송되어 온 것 같다. 이유가 무엇이라고 생각하는가?
한 가지 가정을 하자면 HHS와 ISECOM이 동일한 호스트에 의해 운용된다고 볼 수 있다. 이것이 가능한지 살펴보기 위해서 hackerhighschool.org의 IP 주소를 확인해 보자.

```
nslookup www.hackerhighschool.org
[...]
Non-authoritative answer:
www.hackerhighschool.org canonical name = hackerhighschool.org.
Name: hackerhighschool.org
Address: 216.92.116.13
```

이번에는 isecom.org의 IP 주소를 확인해 보자.

```
nslookup isecom.org
[...]
Non-authoritative answer:
Name: isecom.org
Address: 216.92.116.13
```

같은 IP 주소를 사용한다는 것을 알 수 있다. netcat을 통해 호스트 헤더를 추가하고 HTTP 1.1을 사용해 호스트를 확인할 수 있다.

```
GET / HTTP/1.1
Host: www.hackerhighschool.org
HTTP/1.1 200 OK
Content-Length: 10376
Date: Fri, 03 Feb 2013 09:11:17 GMT
Server: Apache/2.2.22
Last-Modified: Mon, 06 Feb 2013 09:31:18 GMT
ETag: "2f42-4b8485316c580"
Accept-Ranges: bytes
Identity: The Institute for Security and Open Methodologies, The
Institute for Security and Open Methodologies
```

```
P3P: Not supported at this time, Not supported at this time
Content-Type: text/html
Connection: keep-alive
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"[]>
<html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="en-US"
xml:lang="en"><head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Hacker Highschool - Security Awareness for Teens</title>
```

요청 방법

HTTP 요청은 다양한 형식으로 사용할 수 있고 이는 요청방법(Request Method)라고 불린다. 주로 사용되는 웹 애플리케이션 요청은 GET과 POST 요청이 있지만 다른 요청 프로토콜들 역시 웹 서버나 애플리케이션 서버에서 사용할 수 있다.
요청 방법의 종류는 다음과 같다.

- **OPTIONS** : 어떤 요청옵션이 지정되었는지 알 수 있다.
  서버를 운용하고 있을 때는 정보에 제한을 걸어야 문제발생을 막을 수 있다.

- **GET** : URL을 통해 직접 정보를 회신 받는다. 예를 들면 다음과 같다.

  ```
  http://www.usairnet.com/cgi-bin/launch/code.cgi?
  Submit=Go&sta=KSAF&state=NM
  ```

  물음표 뒤에 오는 내용들이 전부 요청 데이터이다.
  이런 형식으로 데이터를 전달하는 것은 눈에 잘 뛰기 때문에 위험하다.

- **HEAD** : GET과 유사하지만 서버가 실제 페이지를 전달하지는 않는다.
  해당 요청방법을 통해 접속 확인, 대역폭 최적화, 접속제한 우회기능을 사용할 수 있다. 사실 ACL 실행을 통해 확인할 수 있는 것은 GET 요청 뿐이다. 이런 경우에는 취약점이 발견될 수 있다.

- **POST** : GET요청과 마찬가지로 데이터를 웹 애플리케이션에 보내기 위해 사용한다. 하지만 데이터는 요청 바디에서 전송된다.

- **PUT** : 웹 서버에 자원을 할당하거나 업데이트하기 위해 사용한다.
  이 요청방법은 사용을 금지 시키거나 인증을 통해 사용자를 제한해야 한다. 해당 요청 방법을 통해 취약점을 찾을 가능성이 있다.

- **DELETE** : 웹서버에 있는 자원들을 삭제한다.
  이 요청방법은 사용을 금지 시키거나 인증을 통해 사용자를 제한해야 한다.
  해당 요청 방법을 통해 취약점을 찾을 가능성이 있다.

- **TRACE** : 애플리케이션 계층의 루프백(Loopback)으로 사용되어 메시지를 반사시킨다.
  디버그 방법으로 사용되는 해당 요청방법은 반드시 금지되어야 한다. 생산 환경의 경우 XSS(Cross Site Scripting)를 통해 기밀성이 훼손되거나 취약점이 발견될 수 있다.

- **CONNECT** : 웹서버를 프록시로 사용하게 해준다.
  이 요청방법은 사용을 금지시키거나 인증을 통해 사용자를 제한해야 한다.
  프록시 IP를 사용해서 삼자 서비스(Third Party Service)에 접속할 수 있다.

이 외에도 WebDAV처럼 HTTP에 기반한 요청 방법들이 존재한다.

OPTIONS 활용

Netcat에 다음과 같은 명령문을 입력해 보자.

```
# nc www.hackerhighschool.org 80
```

이번에는 엔터를 두 번 누르지 말고 다음의 명령어를 입력하자.

```
OPTIONS / HTTP/1.1
```

그러면 다음과 같은 응답문을 볼 수 있다.

```
Host: www.hackerhighschool.org
HTTP/1.0 200 OK
Date: Tue, 07 Feb 2013 08:43:38 GMT
Server: Apache/2.2.22
Allow: GET,HEAD,POST,OPTIONS
Identity: The Institute for Security and Open Methodologies, The
```

```
Institute for Security and Open Methodologies
P3P: Not supported at this time, Not supported at this time
Content-Length: 0
Content-Type: text/html
```

HEAD 활용

이번에는 다음 명령어를 입력해 보자.

```
# nc www.hackerhighschool.org 80
HEAD / HTTP/1.1
Host: www.hackerhighschool.org

HTTP/1.0 200 OK
Date: Tue, 07 Feb 2013 08:41:14 GMT
Server: Apache/2.2.22
Last-Modified: Fri, 13 Feb 2013 15:48:14 GMT
ETag: "3e3a-4bd916679ab80"
Accept-Ranges: bytes
Content-Length: 15930
Identity: The Institute for Security and Open Methodologies
P3P: Not supported at this time
Content-Type: text/html
Age: 45
Connection: close
```

CONNECT 활용을 통한 프록시 사용

```
# nc www.hackerhighschool.org 80
CONNECT http://www.isecom.org/ HTTP/1.1
Host: www.hackerhighschool.org
```

Exercises

4.28 Netcat을 사용하여 이번에 공부한 요청 방법들을 HHS 네트워크 서버에서 연습해 보자. 어떤 정보들을 획득할 수 있는가?

Curl 명령어 활용을 통한 HTTP 요청 스크립팅

웹 애플리케이션 테스트는 웹 서버 응답 뿐 아니라 애플리케이션 계층 응답을 통해서도 진행된다. 웹 애플리케이션 취약점은 GET과 POST 한도 변경, 쿠키 변경, 헤더조사를 통해 파악할 수도 있다. 배쉬 스크립팅에 유용한 툴로 **curl** 명령어가 있다. 해당 명령어를 통해 웹페이지에 요청을 보낼 수 있다.

```
# curl http://www.isecom.org
```

위의 명령어는

```
# nc www.isecom.org 80
GET / HTTP/1.1
```

와는 다른 결과를 가져온다.

-v 옵션을 통해 아래와 같은 세밀한 정보를 얻을 수 있다.

```
# curl -v http://www.isecom.org/
* About to connect() to www.isecom.org port 80 (#0)
* Trying 216.92.116.13...
* connected
* Connected to www.isecom.org (216.92.116.13) port 80 (#0)
> GET / HTTP/1.1
> User-Agent: curl/7.26.0
> Host: www.isecom.org
> Accept: */*
>
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
< Date: Tue, 07 Feb 2013 09:29:23 GMT
< Server: Apache/2.2.22
< Last-Modified: Fri, 13 Feb 2013 15:48:14 GMT
< ETag: "3e3a-4bd916679ab80"
< Accept-Ranges: bytes
< Content-Length: 15930
< Identity: The Institute for Security and Open Methodologies
```

```
< P3P: Not supported at this time
< Content-Type: text/html
< Age: 247
< Connection: close
<
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"[]>
<html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="en-US"
xml:lang="en">
[...]
```

Curl 명령어는 자동으로 HTTP version 1.1을 선택하고 호스트 헤더, 유저 에이전트를 추가한다. 해커활동에 필요한 정보들이 많으니 공부해 보자.
Curl 명령어는 옵션들을 사용해서 자기가 원하는 용도로 사용할 수 있다.
모든 옵션을 보기 위해서는 curl –help 명령어를 사용하면 된다.
Netcat과 유사한 옵션들은 다음과 같다.

- **-H** : 해더정보를 추가한다.
- **-X** : 요청방법을 선택한다.
- **-d** : POST 데이터를 추가한다.
- **-i** : 결과에 프로토콜 해더를 포함시킨다.
- **-s** : 스크립팅에 유용한 사일런트 모드(Silent Mode)를 가능하게 한다.

Curl 명령어와 배쉬 스크립팅을 통해 웹 애플리케이션 테스트를 자동화 할 수 있다. Curl과 grep 명령어를 통해 서버에서 원하는 HTTP 헤더를 찾도록 자동화 할 수 있다.

```
# curl –siX HEAD http://www.isecom.org/ | grep "Server:"
Server: Apache/2.2.22
```

Exercises

4.29 위의 스크립트를 수정하여 더 많은 HTTP 헤더정보와 유용한 정보를 검색하도록 만들어보자.

참고문헌과 추가 공부자료

http://www.ietf.org/rfc/rfc1945.txt
http://www.ietf.org/rfc/rfc2616.txt

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